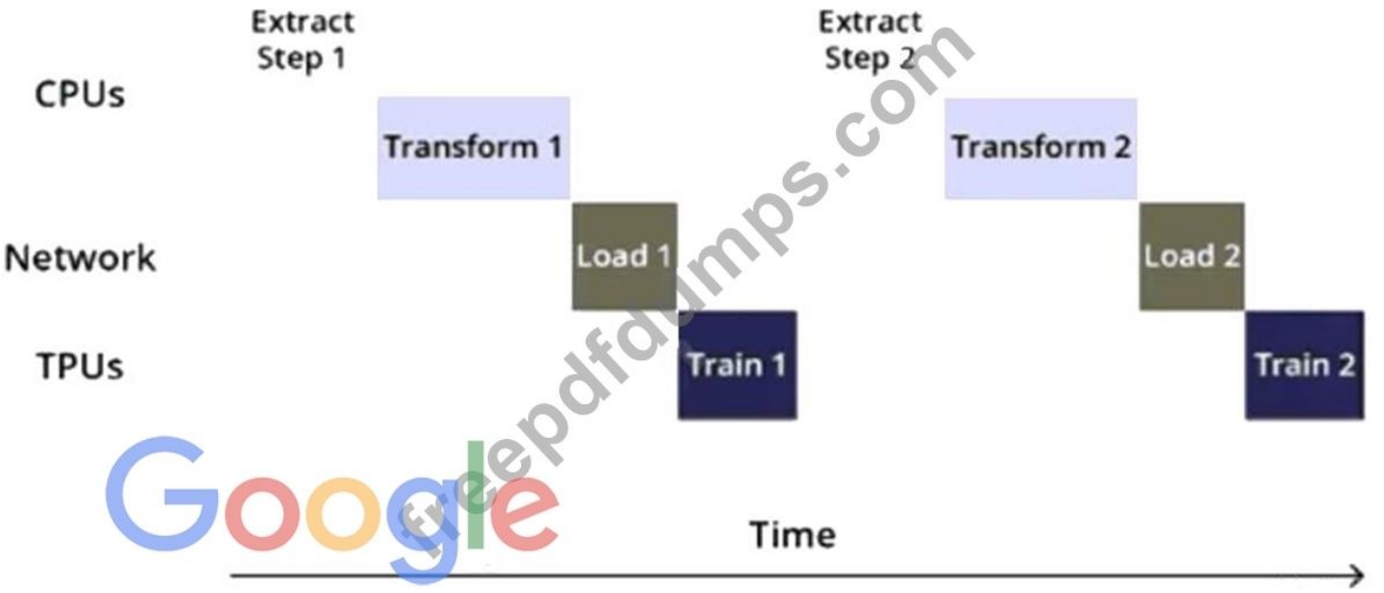
Google.Professional-Machine-Learning-Engineer.v2025-08-30.q138

Exam Code:	Professional-Machine-Learning-Engineer
Exam Name:	Google Professional Machine Learning Engineer
Certification Provider:	Google
Free Question Number:	138
Version:	v2025-08-30
# of views:	107
# of Questions views:	1380
https://www.freepdfdumps.com/Google.Professional-Machine-Learning-Engineer.v2025-08-30.q138.html	

NEW QUESTION: 1
You need to build an ML model for a social media application to predict whether a user's submitted profile photo meets the requirements. The application will inform the user if the picture meets the requirements. How should you build a model to ensure that the application does not falsely accept a non-compliant picture?
A. Use AutoML to optimize the model's recall in order to minimize false negatives.
B. Use AutoML to optimize the model's F1 score in order to balance the accuracy of false positives and false negatives.
C. Use Vertex AI Workbench user-managed notebooks to build a custom model that has three times as many examples of pictures that meet the profile photo requirements.
D. Use Vertex AI Workbench user-managed notebooks to build a custom model that has three times as many examples of pictures that do not meet the profile photo requirements.
Answer: A (LEAVE A REPLY)
Recall is the ratio of true positives to the sum of true positives and false negatives. It measures how well the model can identify all the relevant cases. In this scenario, the relevant cases are the pictures that do not meet the profile photo requirements. Therefore, minimizing false negatives means minimizing the cases where the model incorrectly predicts that a non-compliant picture meets the requirements. By using AutoML to optimize the model's recall, the model will be more likely to reject a non-compliant picture and inform the user accordingly. Reference:
[AutoML Vision] is a service that allows you to train custom ML models for image classification and object detection tasks. You can use AutoML to optimize your model for different metrics, such as recall, precision, or F1 score.
[Recall] is one of the evaluation metrics for ML models. It is defined as TP / (TP + FN), where TP is the number of true positives and FN is the number of false negatives. Recall measures how

well the model can identify all the relevant cases. A high recall means that the model has a low rate of false negatives.

NEW QUESTION: 2
You are training an object detection model using a Cloud TPU v2. Training time is taking longer than expected. Based on this simplified trace obtained with a Cloud TPU profile, what action should you take to decrease training time in a cost-efficient way?



A. Move from Cloud TPU v2 to Cloud TPU v3 and increase batch size.
B. Move from Cloud TPU v2 to 8 NVIDIA V100 GPUs and increase batch size.
C. Rewrite your input function to resize and reshape the input images.
D. Rewrite your input function using parallel reads, parallel processing, and prefetch.
Answer: D (LEAVE A REPLY)
The trace in the question shows that the training time is taking longer than expected. This is likely due to the input function not being optimized. To decrease training time in a cost-efficient way, the best option is to rewrite the input function using parallel reads, parallel processing, and prefetch. This will allow the model to process the data more efficiently and decrease training time.
Reference:
[Cloud TPU Performance Guide]
[Data input pipeline performance guide]

NEW QUESTION: 3
You work for a social media company. You want to create a no-code image classification model for an iOS mobile application to identify fashion accessories You have a labeled dataset in Cloud Storage You need to configure a training workflow that minimizes cost and serves predictions with the lowest possible latency What should you do?
A. Train the model by using AutoML, and register the model in Vertex AI Model Registry Configure your mobile application to send batch requests during prediction.

B. Train the model by using AutoML Edge and export it as a Core ML model Configure your mobile application to use the mlmodel file directly.
C. Train the model by using AutoML Edge and export the model as a TFLite model Configure your mobile application to use the tflite file directly
D. Train the model by using AutoML, and expose the model as a Vertex AI endpoint Configure your mobile application to invoke the endpoint during prediction.
Answer: B (LEAVE A REPLY)
AutoML Edge is a service that allows you to train and deploy custom image classification models for mobile devices12. It supports exporting models as Core ML files, which are compatible with iOS applications3.
Using a Core ML model directly on the device eliminates the need for network requests and reduces prediction latency. It also minimizes the cost of serving predictions, as there is no need to pay for cloud resources or network bandwidth.
Option A is incorrect because sending batch requests during prediction does not reduce latency, as the requests still need to be processed by the cloud service. It also incurs more cost than using a local model on the device.
Option C is incorrect because TFLite models are not compatible with iOS applications. TFLite models are designed for Android and other platforms that support TensorFlow Lite4.
Option D is incorrect because exposing the model as a Vertex AI endpoint requires network requests and cloud resources, which increase latency and cost. It also does not leverage the benefits of AutoML Edge, which is optimized for mobile devices.

NEW QUESTION: 4
You are training a Resnet model on AI Platform using TPUs to visually categorize types of defects in automobile engines. You capture the training profile using the Cloud TPU profiler plugin and observe that it is highly input-bound. You want to reduce the bottleneck and speed up your model training process. Which modifications should you make to the tf .data dataset?
Choose 2 answers
A. Use the interleave option for reading data
B. Reduce the value of the repeat parameter
C. Increase the buffer size for the shuffle option.
D. Set the prefetch option equal to the training batch size
E. Decrease the batch size argument in your transformation
Answer: A,D (LEAVE A REPLY)
The tf.data dataset is a TensorFlow API that provides a way to create and manipulate data pipelines for machine learning. The tf.data dataset allows you to apply various transformations to the data, such as reading, shuffling, batching, prefetching, and interleaving. These transformations can affect the performance and efficiency of the model training process1 One of the common performance issues in model training is input-bound, which means that the model is waiting for the input data to be ready and is not fully utilizing the computational resources. Input-bound can be caused by slow data loading, insufficient parallelism, or large data size. Input-

bound can be detected by using the Cloud TPU profiler plugin, which is a tool that helps you analyze the performance of your model on Cloud TPUs. The Cloud TPU profiler plugin can show you the percentage of time that the TPU cores are idle, which indicates input-bound2 To reduce the input-bound bottleneck and speed up the model training process, you can make some modifications to the tf.data dataset. Two of the modifications that can help are:
Use the interleave option for reading data. The interleave option allows you to read data from multiple files in parallel and interleave their records. This can improve the data loading speed and reduce the idle time of the TPU cores. The interleave option can be applied by using the tf.data.Dataset.interleave method, which takes a function that returns a dataset for each input element, and a number of parallel calls3 Set the prefetch option equal to the training batch size. The prefetch option allows you to prefetch the next batch of data while the current batch is being processed by the model. This can reduce the latency between batches and improve the throughput of the model training. The prefetch option can be applied by using the tf.data.Dataset.prefetch method, which takes a buffer size argument. The buffer size should be equal to the training batch size, which is the number of examples per batch4 The other options are not effective or counterproductive. Reducing the value of the repeat parameter will reduce the number of epochs, which is the number of times the model sees the entire dataset. This can affect the model's accuracy and convergence. Increasing the buffer size for the shuffle option will increase the randomness of the data, but also increase the memory usage and the data loading time. Decreasing the batch size argument in your transformation will reduce the number of examples per batch, which can affect the model's stability and performance.

NEW QUESTION: 5
You work at a gaming startup that has several terabytes of structured data in Cloud Storage. This data includes gameplay time data, user metadata, and game metadat a. You want to build a model that recommends new games to users that requires the least amount of coding. What should you do?
A. Load the data in BigQuery. Use BigQuery ML to train an Autoencoder model.
B. Load the data in BigQuery. Use BigQuery ML to train a matrix factorization model.
C. Read data to a Vertex AI Workbench notebook. Use TensorFlow to train a two-tower model.
D. Read data to a Vertex AI Workbench notebook. Use TensorFlow to train a matrix factorization model.
Answer: B (LEAVE A REPLY)
The best option to build a game recommendation model with the least amount of coding is to use BigQuery ML, which allows you to create and execute machine learning models using standard SQL queries. BigQuery ML supports several types of models, including matrix factorization, which is a common technique for collaborative filtering-based recommendation systems. Matrix factorization models learn latent factors for users and items from the observed ratings, and then use them to predict the ratings for new user-item pairs. BigQuery ML provides a built-in function called ML.RECOMMEND that can generate recommendations for a given user based on a trained matrix factorization model. To use BigQuery ML, you need to load the data in BigQuery, which is

a serverless, scalable, and cost-effective data warehouse. You can use the bq command-line tool, the BigQuery API, or the Cloud Console to load data from Cloud Storage to BigQuery. Alternatively, you can use federated queries to query data directly from Cloud Storage without loading it to BigQuery, but this may incur additional costs and performance overhead. Option A is incorrect because BigQuery ML does not support Autoencoder models, which are a type of neural network that can learn compressed representations of the input data. Autoencoder models are not suitable for recommendation systems, as they do not capture the interactions between users and items. Option C is incorrect because using TensorFlow to train a two-tower model requires more coding than using BigQuery ML. A two-tower model is a type of neural network that learns embeddings for users and items separately, and then combines them with a dot product or a cosine similarity to compute the rating. TensorFlow is a low-level framework that requires you to define the model architecture, the loss function, the optimizer, the training loop, and the evaluation metrics. Moreover, you need to read the data from Cloud Storage to a Vertex AI Workbench notebook, which is an instance of JupyterLab that runs on a Google Cloud virtual machine. This may involve additional steps such as authentication, authorization, and data preprocessing. Option D is incorrect because using TensorFlow to train a matrix factorization model also requires more coding than using BigQuery ML. Although TensorFlow provides some high-level APIs such as Keras and TensorFlow Recommenders that can simplify the model development, you still need to handle the data loading and the model training and evaluation yourself. Furthermore, you need to read the data from Cloud Storage to a Vertex AI Workbench notebook, which may incur additional complexity and costs. Reference:
BigQuery ML documentation
Using matrix factorization with BigQuery ML
Recommendations AI documentation
Loading data into BigQuery
Querying data in Cloud Storage from BigQuery
Vertex AI Workbench documentation
TensorFlow documentation
TensorFlow Recommenders documentation

NEW QUESTION: 6
Your company stores a large number of audio files of phone calls made to your customer call center in an on-premises database. Each audio file is in wav format and is approximately 5 minutes long. You need to analyze these audio files for customer sentiment. You plan to use the Speech-to-Text API. You want to use the most efficient approach. What should you do?
A. 1 Upload the audio files to Cloud Storage
2. Call the speech: longrunningrecognize API endpoint to generate transcriptions
3. Call the predict method of an AutoML sentiment analysis model to analyze the transcriptions
B. 1 Upload the audio files to Cloud Storage
2 Call the speech: longrunningrecognize API endpoint to generate transcriptions.

3 Create a Cloud Function that calls the Natural Language API by using the analyzesentiment method

C. 1 Iterate over your local Tiles in Python

2. Use the Speech-to-Text Python library to create a speech.RecognitionAudio object and set the content to the audio file data

3. Call the speech: recognize API endpoint to generate transcriptions

4. Call the predict method of an AutoML sentiment analysis model to analyze the transcriptions

D. 1 Iterate over your local files in Python

2 Use the Speech-to-Text Python Library to create a speech.RecognitionAudio object, and set the content to the audio file data

3. Call the speech: lengrunningrecognize API endpoint to generate transcriptions

4 Call the Natural Language API by using the analyzesenriment method

Answer: (SHOW ANSWER)

According to the official exam guide1, one of the skills assessed in the exam is to "design, build, and productionalize ML models to solve business challenges using Google Cloud technologies". The Speech-to-Text API2 allows you to convert audio to text by applying powerful neural network models. The Natural Language API3 enables you to analyze text and extract information about the sentiment, entities, and syntax. The Cloud Functions4 service lets you write and deploy code that runs in response to events, such as a Pub/Sub message or an HTTP request. Therefore, option B is the most efficient approach to analyze the audio files for customer sentiment, as it leverages the existing Google Cloud services and avoids unnecessary data processing and model training. The other options are not relevant or optimal for this scenario. Reference:

Professional ML Engineer Exam Guide

Speech-to-Text API

Natural Language API

Cloud Functions

Google Professional Machine Learning Certification Exam 2023

Latest Google Professional Machine Learning Engineer Actual Free Exam Questions

NEW QUESTION: 7

You are building a MLOps platform to automate your company's ML experiments and model retraining. You need to organize the artifacts for dozens of pipelines How should you store the pipelines' artifacts'?

A. Store parameters in Cloud SQL and store the models' source code and binaries in GitHub

B. Store parameters in Cloud SQL store the models' source code in GitHub, and store the models' binaries in Cloud Storage.

C. Store parameters in Vertex ML Metadata store the models' source code in GitHub and store the models' binaries in Cloud Storage.

D. Store parameters in Vertex ML Metadata and store the models source code and binaries in GitHub.

Answer: C (LEAVE A REPLY)

To organize the artifacts for dozens of pipelines, you should store the parameters in Vertex ML Metadata, store the models' source code in GitHub, and store the models' binaries in Cloud Storage. This option has the following advantages:
Vertex ML Metadata is a service that helps you track and manage the metadata of your ML workflows, such as datasets, models, metrics, and parameters1. It can also help you with data lineage, model versioning, and model performance monitoring2.
GitHub is a popular platform for hosting and collaborating on code repositories. It can help you manage the source code of your models, as well as the configuration files, scripts, and notebooks that are part of your ML pipelines3.
Cloud Storage is a scalable and durable object storage service that can store any type of data, including model binaries4. It can also integrate with other services, such as Vertex AI, Cloud Functions, and Cloud Run, to enable easy deployment and serving of your models5.
Reference:
1: Introduction to Vertex ML Metadata | Vertex AI | Google Cloud
2: Manage metadata for ML workflows | Vertex AI | Google Cloud
3: GitHub - Where the world builds software
4: Cloud Storage | Google Cloud
5: Deploying models | Vertex AI | Google Cloud

NEW QUESTION: 8
You recently deployed a scikit-learn model to a Vertex AI endpoint You are now testing the model on live production traffic While monitoring the endpoint. you discover twice as many requests per hour than expected throughout the day You want the endpoint to efficiently scale when the demand increases in the future to prevent users from experiencing high latency What should you do?
A. Deploy two models to the same endpoint and distribute requests among them evenly.
B. Configure an appropriate minReplicaCount value based on expected baseline traffic.
C. Set the target utilization percentage in the autcscalir.gMetricspecs configuration to a higher value
D. Change the model's machine type to one that utilizes GPUs.
Answer: B (LEAVE A REPLY)
The best option for scaling a Vertex AI endpoint efficiently when the demand increases in the future, using a scikit-learn model that is deployed to a Vertex AI endpoint and tested on live production traffic, is to configure an appropriate minReplicaCount value based on expected baseline traffic. This option allows you to leverage the power and simplicity of Vertex AI to automatically scale your endpoint resources according to the traffic patterns. Vertex AI is a unified platform for building and deploying machine learning solutions on Google Cloud. Vertex AI can deploy a trained model to an online prediction endpoint, which can provide low-latency predictions for individual instances. Vertex AI can also provide various tools and services for data analysis, model development, model deployment, model monitoring, and model governance. A minReplicaCount value is a parameter that specifies the minimum number of replicas that the

endpoint must always have, regardless of the load. A minReplicaCount value can help you ensure that the endpoint has enough resources to handle the expected baseline traffic, and avoid high latency or errors. By configuring an appropriate minReplicaCount value based on expected baseline traffic, you can scale your endpoint efficiently when the demand increases in the future. You can set the minReplicaCount value when you deploy the model to the endpoint, or update it later. Vertex AI will automatically scale up or down the number of replicas within the range of the minReplicaCount and maxReplicaCount values, based on the target utilization percentage and the autoscaling metric1.

The other options are not as good as option B, for the following reasons:

Option A: Deploying two models to the same endpoint and distributing requests among them evenly would not allow you to scale your endpoint efficiently when the demand increases in the future, and could increase the complexity and cost of the deployment process. A model is a resource that represents a machine learning model that you can use for prediction. A model can have one or more versions, which are different implementations of the same model. A model version can help you experiment and iterate on your model, and improve the model performance and accuracy. An endpoint is a resource that provides the service endpoint (URL) you use to request the prediction. An endpoint can have one or more deployed models, which are instances of model versions that are associated with physical resources. A deployed model can help you serve online predictions with low latency, and scale up or down based on the traffic. By deploying two models to the same endpoint and distributing requests among them evenly, you can create a load balancing mechanism that can distribute the traffic across the models, and reduce the load on each model. However, deploying two models to the same endpoint and distributing requests among them evenly would not allow you to scale your endpoint efficiently when the demand increases in the future, and could increase the complexity and cost of the deployment process. You would need to write code, create and configure the two models, deploy the models to the same endpoint, and distribute the requests among them evenly. Moreover, this option would not use the autoscaling feature of Vertex AI, which can automatically adjust the number of replicas based on the traffic patterns, and provide various benefits, such as optimal resource utilization, cost savings, and performance improvement2.

Option C: Setting the target utilization percentage in the autoscalingMetricSpecs configuration to a higher value would not allow you to scale your endpoint efficiently when the demand increases in the future, and could cause errors or poor performance. A target utilization percentage is a parameter that specifies the desired utilization level of each replica. A target utilization percentage can affect the speed and accuracy of the autoscaling process. A higher target utilization percentage can help you reduce the number of replicas, but it can also cause high latency, low throughput, or resource exhaustion. By setting the target utilization percentage in the autoscalingMetricSpecs configuration to a higher value, you can increase the utilization level of each replica, and save some resources. However, setting the target utilization percentage in the autoscalingMetricSpecs configuration to a higher value would not allow you to scale your endpoint efficiently when the demand increases in the future, and could cause errors or poor performance. You would need to write code, create and configure the autoscalingMetricSpecs,

and set the target utilization percentage to a higher value. Moreover, this option would not ensure that the endpoint has enough resources to handle the expected baseline traffic, which could cause high latency or errors1.
Option D: Changing the model's machine type to one that utilizes GPUs would not allow you to scale your endpoint efficiently when the demand increases in the future, and could increase the complexity and cost of the deployment process. A machine type is a parameter that specifies the type of virtual machine that the prediction service uses for the deployed model. A machine type can affect the speed and accuracy of the prediction process. A machine type that utilizes GPUs can help you accelerate the computation and processing of the prediction, and handle more prediction requests at the same time. By changing the model's machine type to one that utilizes GPUs, you can improve the prediction performance and efficiency of your model. However, changing the model's machine type to one that utilizes GPUs would not allow you to scale your endpoint efficiently when the demand increases in the future, and could increase the complexity and cost of the deployment process. You would need to write code, create and configure the model, deploy the model to the endpoint, and change the machine type to one that utilizes GPUs. Moreover, this option would not use the autoscaling feature of Vertex AI, which can automatically adjust the number of replicas based on the traffic patterns, and provide various benefits, such as optimal resource utilization, cost savings, and performance improvement2.
Reference:
Configure compute resources for prediction | Vertex AI | Google Cloud
Deploy a model to an endpoint | Vertex AI | Google Cloud

NEW QUESTION: 9
You have trained a deep neural network model on Google Cloud. The model has low loss on the training data, but is performing worse on the validation dat a. You want the model to be resilient to overfitting. Which strategy should you use when retraining the model?
A. Apply a dropout parameter of 0 2, and decrease the learning rate by a factor of 10
B. Apply a L2 regularization parameter of 0.4, and decrease the learning rate by a factor of 10.
C. Run a hyperparameter tuning job on AI Platform to optimize for the L2 regularization and dropout parameters
D. Run a hyperparameter tuning job on AI Platform to optimize for the learning rate, and increase the number of neurons by a factor of 2.
Answer: C (LEAVE A REPLY)
Overfitting occurs when a model tries to fit the training data so closely that it does not generalize well to new data. Overfitting can be caused by having a model that is too complex for the data, such as having too many parameters or layers. Overfitting can lead to poor performance on the validation data, which reflects how the model will perform on unseen data1 To prevent overfitting, one strategy is to use regularization techniques that penalize the complexity of the model and encourage it to learn simpler patterns. Two common regularization techniques for deep neural networks are L2 regularization and dropout. L2 regularization adds a term to the loss function that is proportional to the squared magnitude of the model's weights. This term penalizes large

weights and encourages the model to use smaller weights. Dropout randomly drops out some units in the network during training, which prevents co-adaptation of features and reduces the effective number of parameters. Both L2 regularization and dropout have hyperparameters that control the strength of the regularization effect23 Another strategy to prevent overfitting is to use hyperparameter tuning, which is the process of finding the optimal values for the parameters of the model that affect its performance. Hyperparameter tuning can help find the best combination of hyperparameters that minimize the validation loss and improve the generalization ability of the model. AI Platform provides a service for hyperparameter tuning that can run multiple trials in parallel and use different search algorithms to find the best solution.
Therefore, the best strategy to use when retraining the model is to run a hyperparameter tuning job on AI Platform to optimize for the L2 regularization and dropout parameters. This will allow the model to find the optimal balance between fitting the training data and generalizing to new data. The other options are not as effective, as they either use fixed values for the regularization parameters, which may not be optimal, or they do not address the issue of overfitting at all.
References: 1: Generalization: Peril of Overfitting 2: Regularization for Deep Learning 3: Dropout: A Simple Way to Prevent Neural Networks from Overfitting : [Hyperparameter tuning overview]

NEW QUESTION: 10
You work for a semiconductor manufacturing company. You need to create a real-time application that automates the quality control process High-definition images of each semiconductor are taken at the end of the assembly line in real time. The photos are uploaded to a Cloud Storage bucket along with tabular data that includes each semiconductor's batch number serial number dimensions, and weight You need to configure model training and serving while maximizing model accuracy. What should you do?
A. Use Vertex AI Data Labeling Service to label the images and train an AutoML image classification model.
Deploy the model and configure Pub/Sub to publish a message when an image is categorized into the failing class.
B. Use Vertex AI Data Labeling Service to label the images and train an AutoML image classification model. Schedule a daily batch prediction job that publishes a Pub/Sub message when the job completes.
C. Convert the images into an embedding representation Import this data into BigQuery, and train a BigQuery. ML K-means clustenng model with two clusters Deploy the model and configure Pub/Sub to publish a message when a semiconductor's data is categorized into the failing cluster.
D. Import the tabular data into BigQuery use Vertex AI Data Labeling Service to label the data and train an AutoML tabular classification model Deploy the model and configure Pub/Sub to publish a message when a semiconductor's data is categorized into the failing class.
Answer: A (LEAVE A REPLY)
Vertex AI is a unified platform for building and managing machine learning solutions on Google Cloud. It provides various services and tools for different stages of the machine learning lifecycle, such as data preparation, model training, deployment, monitoring, and experimentation. Vertex AI

Data Labeling Service is a service that allows you to create and manage human-labeled datasets for machine learning. You can use Vertex AI Data Labeling Service to label the images of semiconductors with binary labels, such as "pass" or "fail", based on the quality criteria. You can also use Vertex AI AutoML Image Classification, which is a service that allows you to create and train custom image classification models without writing any code. You can use Vertex AI AutoML Image Classification to train an image classification model on the labeled images of semiconductors, and optimize the model for accuracy. You can also use Vertex AI to deploy the model to an endpoint, which is a service that allows you to serve online predictions from your model. You can configure Pub/Sub, which is a service that allows you to publish and subscribe to messages, to publish a message when an image is categorized into the failing class by the model. You can use the message to trigger an action, such as alerting the quality control team or stopping the production line. This solution can help you create a real-time application that automates the quality control process of semiconductors, and maximizes the model accuracy.
Reference: The answer can be verified from official Google Cloud documentation and resources related to Vertex AI, Vertex AI Data Labeling Service, Vertex AI AutoML Image Classification, and Pub/Sub.
Vertex AI | Google Cloud
Vertex AI Data Labeling Service | Google Cloud
Vertex AI AutoML Image Classification | Google Cloud
Pub/Sub | Google Cloud

NEW QUESTION: 11
You are the Director of Data Science at a large company, and your Data Science team has recently begun using the Kubeflow Pipelines SDK to orchestrate their training pipelines. Your team is struggling to integrate their custom Python code into the Kubeflow Pipelines SDK. How should you instruct them to proceed in order to quickly integrate their code with the Kubeflow Pipelines SDK?
A. Use the func_to_container_op function to create custom components from the Python code.
B. Use the predefined components available in the Kubeflow Pipelines SDK to access Dataproc, and run the custom code there.
C. Package the custom Python code into Docker containers, and use the load_component_from_file function to import the containers into the pipeline.
D. Deploy the custom Python code to Cloud Functions, and use Kubeflow Pipelines to trigger the Cloud Function.
Answer: A (LEAVE A REPLY)
The easiest way to integrate custom Python code into the Kubeflow Pipelines SDK is to use the func_to_container_op function, which converts a Python function into a pipeline component. This function automatically builds a Docker image that executes the Python function, and returns a factory function that can be used to create kfp.dsl.ContainerOp instances for the pipeline. This option has the following benefits:

It allows the data science team to reuse their existing Python code without rewriting it or packaging it into containers manually.
It simplifies the component specification and implementation, as the function signature defines the component interface and the function body defines the component logic.
It supports various types of inputs and outputs, such as primitive types, files, directories, and dictionaries.
The other options are less optimal for the following reasons:
Option B: Using the predefined components available in the Kubeflow Pipelines SDK to access Dataproc, and run the custom code there, introduces additional complexity and cost. This option requires creating and managing Dataproc clusters, which are ephemeral and scalable clusters of Compute Engine instances that run Apache Spark and Apache Hadoop. Moreover, this option requires writing the custom code in PySpark or Hadoop MapReduce, which may not be compatible with the existing Python code.
Option C: Packaging the custom Python code into Docker containers, and using the load_component_from_file function to import the containers into the pipeline, introduces additional steps and overhead. This option requires creating and maintaining Dockerfiles, building and pushing Docker images, and writing component specifications in YAML files. Moreover, this option requires managing the dependencies and versions of the Python code and the Docker images.
Option D: Deploying the custom Python code to Cloud Functions, and using Kubeflow Pipelines to trigger the Cloud Function, introduces additional latency and limitations. This option requires creating and deploying Cloud Functions, which are serverless functions that execute in response to events. Moreover, this option requires invoking the Cloud Functions from the Kubeflow Pipelines using HTTP requests, which can incur network overhead and latency. Additionally, this option is subject to the quotas and limits of Cloud Functions, such as the maximum execution time and memory usage.
Reference:
Building Python function-based components | Kubeflow
Building Python Function-based Components | Kubeflow

NEW QUESTION: 12
You work for a credit card company and have been asked to create a custom fraud detection model based on historical data using AutoML Tables. You need to prioritize detection of fraudulent transactions while minimizing false positives. Which optimization objective should you use when training the model?
A. An optimization objective that minimizes Log loss
B. An optimization objective that maximizes the Precision at a Recall value of 0.50
C. An optimization objective that maximizes the area under the precision-recall curve (AUC PR) value
D. An optimization objective that maximizes the area under the receiver operating characteristic curve (AUC ROC) value

Answer: C (LEAVE A REPLY)

In this scenario, the goal is to create a custom fraud detection model using AutoML Tables. Fraud detection is a type of binary classification problem, where the model needs to predict whether a transaction is fraudulent or not. The optimization objective is a metric that defines how the model is trained and evaluated. AutoML Tables allows you to choose from different optimization objectives for binary classification problems, such as Log loss, Precision at a Recall value, AUC PR, and AUC ROC.

To choose the best optimization objective for fraud detection, we need to consider the characteristics of the problem and the data. Fraud detection is a problem where the positive class (fraudulent transactions) is very rare compared to the negative class (legitimate transactions). This means that the data is highly imbalanced, and the model needs to be sensitive to the minority class. Moreover, fraud detection is a problem where the cost of false negatives (missing a fraudulent transaction) is much higher than the cost of false positives (flagging a legitimate transaction as fraudulent). This means that the model needs to have high recall (the ability to detect all fraudulent transactions) while maintaining high precision (the ability to avoid false alarms).

Given these considerations, the best optimization objective for fraud detection is the one that maximizes the area under the precision-recall curve (AUC PR) value. The AUC PR value is a metric that measures the trade-off between precision and recall for different probability thresholds. A higher AUC PR value means that the model can achieve high precision and high recall at the same time. The AUC PR value is also more suitable for imbalanced data than the AUC ROC value, which measures the trade-off between the true positive rate and the false positive rate. The AUC ROC value can be misleading for imbalanced data, as it can give a high score even if the model has low recall or low precision.

Therefore, option C is the correct answer. Option A is not suitable, as Log loss is a metric that measures the difference between the predicted probabilities and the actual labels, and does not account for the trade-off between precision and recall. Option B is not suitable, as Precision at a Recall value is a metric that measures the precision at a fixed recall level, and does not account for the trade-off between precision and recall at different thresholds. Option D is not suitable, as AUC ROC is a metric that can be misleading for imbalanced data, as explained above.

Reference:

AutoML Tables documentation

Optimization objectives for binary classification

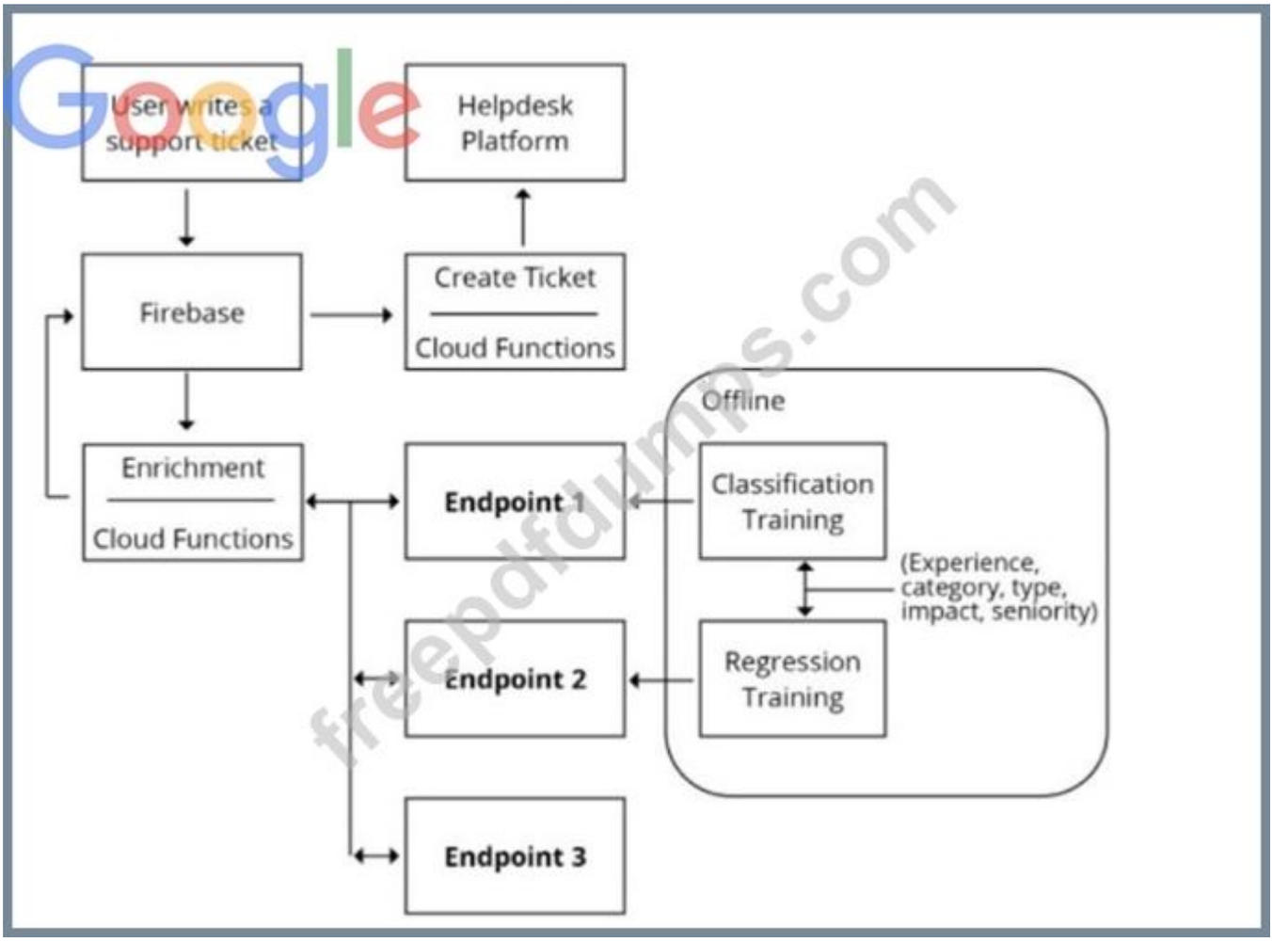
Precision-Recall Curves: How to Easily Evaluate Machine Learning Models in No Time ROC Curves and Area Under the Curve Explained (video)

NEW QUESTION: 13

You are designing an architecture with a serverless ML system to enrich customer support tickets with informative metadata before they are routed to a support agent. You need a set of models to predict ticket priority, predict ticket resolution time, and perform sentiment analysis to help agents

make strategic decisions when they process support requests. Tickets are not expected to have any domain-specific terms or jargon.

The proposed architecture has the following flow:



Which endpoints should the Enrichment Cloud Functions call?

A. 1 Vertex AI. 2 Vertex AI. 3 AutoML Natural Language

B. 1 Vertex AI. 2 Vertex AI. 3 Cloud Natural Language API

C. 1 Vertex AI. 2 Vertex AI. 3 AutoML Vision

D. 1 Cloud Natural Language API. 2 Vertex AI, 3 Cloud Vision API

Answer: B (LEAVE A REPLY)

Vertex AI is a unified platform for building and deploying ML models on Google Cloud. It supports both custom and AutoML models, and provides various tools and services for ML development, such as Vertex Pipelines, Vertex Vizier, Vertex Explainable AI, and Vertex Feature Store. Vertex AI can be used to create models for predicting ticket priority and resolution time, as these are domain-specific tasks that require custom training data and evaluation metrics. Cloud Natural Language API is a pre-trained service that provides natural language understanding capabilities, such as sentiment analysis, entity analysis, syntax analysis, and content classification. Cloud Natural Language API can be used to perform sentiment analysis on the support tickets, as this is a general task that does not require domain-specific knowledge or jargon. The other options are not suitable for the given architecture. AutoML Natural Language and AutoML Vision are services that allow users to create custom natural language and vision models using their own data and

labels. They are not needed for sentiment analysis, as Cloud Natural Language API already provides this functionality. Cloud Vision API is a pre-trained service that provides image analysis capabilities, such as object detection, face detection, text detection, and image labeling. It is not relevant for the support tickets, as they are not expected to have any images. Reference:
Vertex AI documentation
Cloud Natural Language API documentation

NEW QUESTION: 14
You are responsible for building a unified analytics environment across a variety of on-premises data marts. Your company is experiencing data quality and security challenges when integrating data across the servers, caused by the use of a wide range of disconnected tools and temporary solutions. You need a fully managed, cloud-native data integration service that will lower the total cost of work and reduce repetitive work. Some members on your team prefer a codeless interface for building Extract, Transform, Load (ETL) process. Which service should you use?
A. Dataflow
B. Dataprep
C. Apache Flink
D. Cloud Data Fusion
Answer: D (LEAVE A REPLY)
Cloud Data Fusion is a fully managed, cloud-native data integration service that helps users efficiently build and manage ETL/ELT data pipelines. It provides a graphical interface to increase time efficiency and reduce complexity, and allows users to easily create and explore data pipelines using a code-free, point and click visual interface. Cloud Data Fusion also supports a broad range of data sources and formats, including on-premises data marts, and ensures data quality and security by using built-in transformation capabilities and Cloud Data Loss Prevention. Cloud Data Fusion lowers the total cost of ownership by handling performance, scalability, availability, security, and compliance needs automatically. Reference:
Cloud Data Fusion documentation
Cloud Data Fusion overview

NEW QUESTION: 15
You are training and deploying updated versions of a regression model with tabular data by using Vertex AI Pipelines. Vertex AI Training Vertex AI Experiments and Vertex AI Endpoints. The model is deployed in a Vertex AI endpoint and your users call the model by using the Vertex AI endpoint. You want to receive an email when the feature data distribution changes significantly, so you can retrigger the training pipeline and deploy an updated version of your model What should you do?
A. Use Vertex AI Model Monitoring Enable prediction drift monitoring on the endpoint. and specify a notification email.
B. In Cloud Logging, create a logs-based alert using the logs in the Vertex AI endpoint. Configure Cloud Logging to send an email when the alert is triggered.

C. In Cloud Monitoring create a logs-based metric and a threshold alert for the metric. Configure Cloud Monitoring to send an email when the alert is triggered.
D. Export the container logs of the endpoint to BigQuery Create a Cloud Function to run a SQL query over the exported logs and send an email. Use Cloud Scheduler to trigger the Cloud Function.
Answer: A (LEAVE A REPLY)
Prediction drift is the change in the distribution of feature values or labels over time. It can affect the performance and accuracy of the model, and may require retraining or redeploying the model. Vertex AI Model Monitoring allows you to monitor prediction drift on your deployed models and endpoints, and set up alerts and notifications when the drift exceeds a certain threshold. You can specify an email address to receive the notifications, and use the information to retrigger the training pipeline and deploy an updated version of your model. This is the most direct and convenient way to achieve your goal. Reference:
Vertex AI Model Monitoring
Monitoring prediction drift
Setting up alerts and notifications

NEW QUESTION: 16
You work on a growing team of more than 50 data scientists who all use AI Platform. You are designing a strategy to organize your jobs, models, and versions in a clean and scalable way. Which strategy should you choose?
A. Set up restrictive IAM permissions on the AI Platform notebooks so that only a single user or group can access a given instance.
B. Separate each data scientist's work into a different project to ensure that the jobs, models, and versions created by each data scientist are accessible only to that user.
C. Use labels to organize resources into descriptive categories. Apply a label to each created resource so that users can filter the results by label when viewing or monitoring the resources.
D. Set up a BigQuery sink for Cloud Logging logs that is appropriately filtered to capture information about AI Platform resource usage. In BigQuery, create a SQL view that maps users to the resources they are using
Answer: C (LEAVE A REPLY)
Labels are key-value pairs that you can attach to AI Platform resources such as jobs, models, and versions. Labels can help you organize your resources into descriptive categories that reflect your business needs. For example, you can use labels to indicate the owner, purpose, environment, or status of a resource. You can also use labels to filter the results when you list or monitor your resources on the Google Cloud Console or the Cloud SDK. Using labels can help you manage your resources in a clean and scalable way, without requiring separate projects or restrictive permissions.
Reference:
Using labels to organize AI Platform resources
Creating and managing labels

NEW QUESTION: 17
Your data science team has requested a system that supports scheduled model retraining, Docker containers, and a service that supports autoscaling and monitoring for online prediction requests. Which platform components should you choose for this system?
A. Vertex AI Pipelines and App Engine
B. Vertex AI Pipelines and AI Platform Prediction
C. Cloud Composer, BigQuery ML , and AI Platform Prediction
D. Cloud Composer, AI Platform Training with custom containers, and App Engine
Answer: B (LEAVE A REPLY)
Vertex AI Pipelines and AI Platform Prediction are the platform components that best suit the requirements of the data science team. Vertex AI Pipelines is a service that allows you to orchestrate and automate your machine learning workflows using pipelines. Pipelines are portable and scalable ML workflows that are based on containers. You can use Vertex AI Pipelines to schedule model retraining, use custom containers, and integrate with other Google Cloud services. AI Platform Prediction is a service that allows you to host your trained models and serve online predictions. You can use AI Platform Prediction to deploy models trained on Vertex AI or elsewhere, and benefit from features such as autoscaling, monitoring, logging, and explainability. Reference:
Vertex AI Pipelines
AI Platform Prediction

NEW QUESTION: 18
You developed a Vertex AI pipeline that trains a classification model on data stored in a large BigQuery table. The pipeline has four steps, where each step is created by a Python function that uses the KubeFlow v2 API The components have the following names:

```
dt=datetime.now().strftime("%Y%m%d%H%M%S")
f"export-{dt}.yaml", f"preprocess-{dt}.yaml", f"train-{dt}.yaml",
f"calibrate-{dt}.yaml"
```

You launch your Vertex AI pipeline as the following:

```
job = aip.PipelineJob(
    display_name="my-awesome-pipeline",
    template_path="pipeline.json",
    job_id=f"my-awesome-pipeline-{dt}",
    parameter_values=params,
    enable_caching=True,
    location="europe-west1"
)
```


You perform many model iterations by adjusting the code and parameters of the training step. You observe high costs associated with the development, particularly the data export and preprocessing steps. You need to reduce model development costs. What should you do?

A.

Change the components' YAML filenames to `export.yaml`, `preprocess.yaml`, `f"train-{dt}.yaml"`, `f"calibrate-{dt}.yaml"`.

B.

Add the `{"kubeflow.v1.caching": True}` parameter to the set of `params` provided to your `PipelineJob`.

C.

Move the first step of your pipeline to a separate step, and provide a cached path to Cloud Storage as an input to the main pipeline.

D.

Change the name of the pipeline to `f"my-awesome-pipeline-{dt}"`.

Answer: A (LEAVE A REPLY)

According to the official exam guide1, one of the skills assessed in the exam is to "automate and orchestrate ML pipelines using Cloud Composer". Vertex AI Pipelines2 is a service that allows you to orchestrate your ML workflows using Kubeflow Pipelines SDK v2 or TensorFlow Extended. Vertex AI Pipelines supports execution caching, which means that if you run a pipeline and it reaches a component that has already been run with the same inputs and parameters, the component does not run again. Instead, the component uses the output from the previous run. This can save you time and resources when you are iterating on your pipeline. Therefore, option A is the best way to reduce model development costs, as it enables execution caching for the data export and preprocessing steps, which are likely to be the same for each model iteration. The other options are not relevant or optimal for this scenario. Reference:

Professional ML Engineer Exam Guide

Vertex AI Pipelines

Google Professional Machine Learning Certification Exam 2023
Latest Google Professional Machine Learning Engineer Actual Free Exam Questions

NEW QUESTION: 19
You have recently used TensorFlow to train a classification model on tabular data You have created a Dataflow pipeline that can transform several terabytes of data into training or prediction datasets consisting of TFRecords. You now need to productionize the model, and you want the predictions to be automatically uploaded to a BigQuery table on a weekly schedule. What should you do?
A. Import the model into Vertex AI and deploy it to a Vertex AI endpoint On Vertex AI Pipelines create a pipeline that uses the Dataf lowPythonJobop and the Mcdei3archPredictoc components.
B. Import the model into Vertex AI and deploy it to a Vertex AI endpoint Create a Dataflow pipeline that reuses the data processing logic sends requests to the endpoint and then uploads predictions to a BigQuery table.
C. Import the model into Vertex AI On Vertex AI Pipelines, create a pipeline that uses the DataflowPythonJobOp and the ModelBatchPredictOp components.
D. Import the model into BigQuery Implement the data processing logic in a SQL query On Vertex AI Pipelines create a pipeline that uses the BigqueryQueryJobop and the EigqueryPredictModejobOp components.
Answer: C (LEAVE A REPLY)
Vertex AI is a service that allows you to create and train ML models using Google Cloud technologies. You can use Vertex AI to import the model that you trained with TensorFlow and store it in the Vertex AI Model Registry. The Vertex AI Model Registry is a service that allows you to store and manage your ML models on Google Cloud. You can then use Vertex AI Pipelines to create a pipeline that uses the DataflowPythonJobOp and the ModelBatchPredictOp components. The DataflowPythonJobOp component is a component that allows you to run a Dataflow job using a Python script. Dataflow is a service that allows you to create and run scalable and portable data processing pipelines on Google Cloud. You can use the DataflowPythonJobOp component to reuse the data processing logic that you created for transforming the data into TFRecords. The ModelBatchPredictOp component is a component that allows you to run a batch prediction job using a model from the Vertex AI Model Registry. Batch prediction is a type of prediction that provides high-throughput responses to large batches of input data. You can use the ModelBatchPredictOp component to make predictions using the TFRecords from the DataflowPythonJobOp component and the model from the Vertex AI Model Registry. You can also configure the ModelBatchPredictOp component to automatically upload the predictions to a BigQuery table. BigQuery is a service that allows you to store and query large amounts of data in a scalable and cost-effective way. You can use BigQuery to store and analyze the predictions from your model. You can also schedule the pipeline to run on a weekly basis, so that the predictions are updated regularly. By using Vertex AI, Vertex AI Pipelines, Dataflow, and BigQuery, you can productionize the model and upload the predictions to a BigQuery table on a weekly schedule. Reference:

Vertex AI documentation
Vertex AI Pipelines documentation
Dataflow documentation
BigQuery documentation
Preparing for Google Cloud Certification: Machine Learning Engineer Professional Certificate

NEW QUESTION: 20
You work at a gaming startup that has several terabytes of structured data in Cloud Storage. This data includes gameplay time data user metadata and game metadat a. You want to build a model that recommends new games to users that requires the least amount of coding. What should you do?
A. Load the data in BigQuery Use BigQuery ML to tram an Autoencoder model.
B. Load the data in BigQuery Use BigQuery ML to train a matrix factorization model.
C. Read data to a Vertex AI Workbench notebook Use TensorFlow to train a two-tower model.
D. Read data to a Vertex AI Workbench notebook Use TensorFlow to train a matrix factorization model.
Answer: (SHOW ANSWER)
BigQuery is a serverless data warehouse that allows you to perform SQL queries on large-scale data. BigQuery ML is a feature of BigQuery that enables you to create and execute machine learning models using standard SQL queries. You can use BigQuery ML to train a matrix factorization model, which is a common technique for recommender systems. Matrix factorization models learn the latent factors that represent the preferences of users and the characteristics of items, and use them to predict the ratings or interactions between users and items. You can use the CREATE MODEL statement to create a matrix factorization model in BigQuery ML, and specify the matrix_factorization option as the model type. You can also use the ML.RECOMMEND function to generate recommendations for new games based on the trained model. This solution requires the least amount of coding, as you only need to write SQL queries to train and use the model. Reference: The answer can be verified from official Google Cloud documentation and resources related to BigQuery and BigQuery ML.
BigQuery ML | Google Cloud
Using matrix factorization | BigQuery ML
ML.RECOMMEND function | BigQuery ML

NEW QUESTION: 21
You work for a startup that has multiple data science workloads. Your compute infrastructure is currently on-premises. and the data science workloads are native to PySpark Your team plans to migrate their data science workloads to Google Cloud You need to build a proof of concept to migrate one data science job to Google Cloud You want to propose a migration process that requires minimal cost and effort. What should you do first?
A. Create a n2-standard-4 VM instance and install Java, Scala and Apache Spark dependencies on it.

B. Create a Google Kubemetes Engine cluster with a basic node pool configuration install Java Scala, and Apache Spark dependencies on it.
C. Create a Standard (1 master. 3 workers) Dataproc cluster, and run a Vertex Al Workbench notebook instance on it.
D. Create a Vertex Al Workbench notebook with instance type n2-standard-4.
Answer: C (LEAVE A REPLY)
According to the official exam guide1, one of the skills assessed in the exam is to "design, build, and productionalize ML models to solve business challenges using Google Cloud technologies". Dataproc2 is a fully managed, fast, and easy-to-use service for running Apache Spark and Apache Hadoop clusters on Google Cloud. Dataproc supports PySpark workloads and provides a simple way to migrate your existing Spark jobs to the cloud. You can create a Dataproc cluster with a few clicks or commands, and run your PySpark jobs on it. You can also use Vertex Al Workbench3, a managed notebook service, to create and run PySpark notebooks on Dataproc clusters. This way, you can interactively develop and test your PySpark code on the cloud. Therefore, option C is the best way to build a proof of concept to migrate one data science job to Google Cloud with minimal cost and effort. The other options are not relevant or optimal for this scenario. Reference:
Professional ML Engineer Exam Guide
Dataproc
Vertex Al Workbench
Google Professional Machine Learning Certification Exam 2023
Latest Google Professional Machine Learning Engineer Actual Free Exam Questions

NEW QUESTION: 22
You are developing an image recognition model using PyTorch based on ResNet50 architecture Your code is working fine on your local laptop on a small subsample. Your full dataset has 200k labeled images You want to quickly scale your training workload while minimizing cost. You plan to use 4 V100 GPUs What should you do?
A. Create a Google Kubernetes Engine cluster with a node pool that has 4 V100 GPUs Prepare and submit a TFJob operator to this node pool.
B. Configure a Compute Engine VM with all the dependencies that launches the training Tram your model with Vertex Al using a custom tier that contains the required GPUs.
C. Create a Vertex Al Workbench user-managed notebooks instance with 4 V100 GPUs, and use it to tram your model.
D. Package your code with Setuptools and use a pre-built container. Train your model with Vertex Al using a custom tier that contains the required GPUs.
Answer: D (LEAVE A REPLY)
Vertex Al is a unified platform for building and managing machine learning solutions on Google Cloud. It provides a managed service for training custom models with various frameworks, such as TensorFlow, PyTorch, scikit-learn, and XGBoost. To train your PyTorch model with Vertex Al, you need to package your code with Setuptools, which is a Python tool for creating and

distributing packages. You also need to use a pre-built container, which is a Docker image that contains the dependencies and libraries for your framework. You can choose from a list of pre-built containers provided by Google, or create your own custom container. By using a pre-built container, you can avoid the hassle of installing and configuring the environment for your model. You can also specify a custom tier for your training job, which allows you to select the number and type of GPUs you want to use. You can choose from various GPU options, such as V100, P100, K80, and T4. By using 4 V100 GPUs, you can leverage the high performance and memory capacity of these accelerators to train your model faster and cheaper than using CPUs. This solution requires minimal changes to your code and can scale your training workload efficiently.
Reference:
Vertex AI | Google Cloud
Custom training with pre-built containers | Vertex AI
[Using GPUs | Vertex AI]

NEW QUESTION: 23
You have been asked to build a model using a dataset that is stored in a medium-sized (~10 GB) BigQuery table. You need to quickly determine whether this data is suitable for model development. You want to create a one-time report that includes both informative visualizations of data distributions and more sophisticated statistical analyses to share with other ML engineers on your team. You require maximum flexibility to create your report. What should you do?
A. Use Vertex AI Workbench user-managed notebooks to generate the report.
B. Use the Google Data Studio to create the report.
C. Use the output from TensorFlow Data Validation on Dataflow to generate the report.
D. Use Dataprep to create the report.
Answer: (SHOW ANSWER)
Option A is correct because using Vertex AI Workbench user-managed notebooks to generate the report is the best way to quickly determine whether the data is suitable for model development, and to create a one-time report that includes both informative visualizations of data distributions and more sophisticated statistical analyses to share with other ML engineers on your team. Vertex AI Workbench is a service that allows you to create and use notebooks for ML development and experimentation. You can use Vertex AI Workbench to connect to your BigQuery table, query and analyze the data using SQL or Python, and create interactive charts and plots using libraries such as pandas, matplotlib, or seaborn. You can also use Vertex AI Workbench to perform more advanced data analysis, such as outlier detection, feature engineering, or hypothesis testing, using libraries such as TensorFlow Data Validation, TensorFlow Transform, or SciPy. You can export your notebook as a PDF or HTML file, and share it with your team. Vertex AI Workbench provides maximum flexibility to create your report, as you can use any code or library that you want, and customize the report as you wish.
Option B is incorrect because using Google Data Studio to create the report is not the most flexible way to quickly determine whether the data is suitable for model development, and to create a one-time report that includes both informative visualizations of data distributions and

more sophisticated statistical analyses to share with other ML engineers on your team. Google Data Studio is a service that allows you to create and share interactive dashboards and reports using data from various sources, such as BigQuery, Google Sheets, or Google Analytics. You can use Google Data Studio to connect to your BigQuery table, explore and visualize the data using charts, tables, or maps, and apply filters, calculations, or aggregations to the data. However, Google Data Studio does not support more sophisticated statistical analyses, such as outlier detection, feature engineering, or hypothesis testing, which may be useful for model development. Moreover, Google Data Studio is more suitable for creating recurring reports that need to be updated frequently, rather than one-time reports that are static.
Option C is incorrect because using the output from TensorFlow Data Validation on Dataflow to generate the report is not the most efficient way to quickly determine whether the data is suitable for model development, and to create a one-time report that includes both informative visualizations of data distributions and more sophisticated statistical analyses to share with other ML engineers on your team. TensorFlow Data Validation is a library that allows you to explore, validate, and monitor the quality of your data for ML. You can use TensorFlow Data Validation to compute descriptive statistics, detect anomalies, infer schemas, and generate data visualizations for your data. Dataflow is a service that allows you to create and run scalable data processing pipelines using Apache Beam. You can use Dataflow to run TensorFlow Data Validation on large datasets, such as those stored in BigQuery. However, this option is not very efficient, as it involves moving the data from BigQuery to Dataflow, creating and running the pipeline, and exporting the results. Moreover, this option does not provide maximum flexibility to create your report, as you are limited by the functionalities of TensorFlow Data Validation, and you may not be able to customize the report as you wish.
Option D is incorrect because using Dataprep to create the report is not the most flexible way to quickly determine whether the data is suitable for model development, and to create a one-time report that includes both informative visualizations of data distributions and more sophisticated statistical analyses to share with other ML engineers on your team. Dataprep is a service that allows you to explore, clean, and transform your data for analysis or ML. You can use Dataprep to connect to your BigQuery table, inspect and profile the data using histograms, charts, or summary statistics, and apply transformations, such as filtering, joining, splitting, or aggregating, to the data. However, Dataprep does not support more sophisticated statistical analyses, such as outlier detection, feature engineering, or hypothesis testing, which may be useful for model development. Moreover, Dataprep is more suitable for creating data preparation workflows that need to be executed repeatedly, rather than one-time reports that are static.
Reference:
Vertex AI Workbench documentation
Google Data Studio documentation
TensorFlow Data Validation documentation
Dataflow documentation
Dataprep documentation
[BigQuery documentation]

[pandas documentation]
[matplotlib documentation]
[seaborn documentation]
[TensorFlow Transform documentation]
[SciPy documentation]
[Apache Beam documentation]

NEW QUESTION: 24
You are training a TensorFlow model on a structured data set with 100 billion records stored in several CSV files. You need to improve the input/output execution performance. What should you do?
A. Load the data into BigQuery and read the data from BigQuery.
B. Load the data into Cloud Bigtable, and read the data from Bigtable
C. Convert the CSV files into shards of TFRecords, and store the data in Cloud Storage
D. Convert the CSV files into shards of TFRecords, and store the data in the Hadoop Distributed File System (HDFS)
Answer: (SHOW ANSWER)
The input/output execution performance of a TensorFlow model depends on how efficiently the model can read and process the data from the data source. Reading and processing data from CSV files can be slow and inefficient, especially if the data is large and distributed. Therefore, to improve the input/output execution performance, one should use a more suitable data format and storage system.
One of the best options for improving the input/output execution performance is to convert the CSV files into shards of TFRecords, and store the data in Cloud Storage. TFRecord is a binary data format that can store a sequence of serialized TensorFlow examples. TFRecord has several advantages over CSV, such as:
Faster data loading: TFRecord can be read and processed faster than CSV, as it avoids the overhead of parsing and decoding the text data. TFRecord also supports compression and checksums, which can reduce the data size and ensure data integrity1 Better performance: TFRecord can improve the performance of the model, as it allows the model to access the data in a sequential and streaming manner, and leverage the tf.data API to build efficient data pipelines. TFRecord also supports sharding and interleaving, which can increase the parallelism and throughput of the data processing2 Easier integration: TFRecord can integrate seamlessly with TensorFlow, as it is the native data format for TensorFlow. TFRecord also supports various types of data, such as images, text, audio, and video, and can store the data schema and metadata along with the data3 Cloud Storage is a scalable and reliable object storage service that can store any amount of data. Cloud Storage has several advantages over other storage systems, such as: High availability: Cloud Storage can provide high availability and durability for the data, as it replicates the data across multiple regions and zones, and supports versioning and lifecycle management. Cloud Storage also offers various storage classes, such as Standard, Nearline, Coldline, and Archive, to meet different performance and cost requirements4 Low latency: Cloud

Storage can provide low latency and high bandwidth for the data, as it supports HTTP and HTTPS protocols, and integrates with other Google Cloud services, such as AI Platform, Dataflow, and BigQuery. Cloud Storage also supports resumable uploads and downloads, and parallel composite uploads, which can improve the data transfer speed and reliability5 Easy access: Cloud Storage can provide easy access and management for the data, as it supports various tools and libraries, such as gsutil, Cloud Console, and Cloud Storage Client Libraries. Cloud Storage also supports fine-grained access control and encryption, which can ensure the data security and privacy.
The other options are not as effective or feasible. Loading the data into BigQuery and reading the data from BigQuery is not recommended, as BigQuery is mainly designed for analytical queries on large-scale data, and does not support streaming or real-time data processing. Loading the data into Cloud Bigtable and reading the data from Bigtable is not ideal, as Cloud Bigtable is mainly designed for low-latency and high-throughput key-value operations on sparse and wide tables, and does not support complex data types or schemas. Converting the CSV files into shards of TFRecords and storing the data in the Hadoop Distributed File System (HDFS) is not optimal, as HDFS is not natively supported by TensorFlow, and requires additional configuration and dependencies, such as Hadoop, Spark, or Beam.

NEW QUESTION: 25
You have recently developed a new ML model in a Jupyter notebook. You want to establish a reliable and repeatable model training process that tracks the versions and lineage of your model artifacts. You plan to retrain your model weekly. How should you operationalize your training process?
A. 1. Create an instance of the CustomTrainingJob class with the Vertex AI SDK to train your model.
2. Using the Notebooks API, create a scheduled execution to run the training code weekly.
B. 1. Create an instance of the CustomJob class with the Vertex AI SDK to train your model.
2. Use the Metadata API to register your model as a model artifact.
3. Using the Notebooks API, create a scheduled execution to run the training code weekly.
C. 1. Create a managed pipeline in Vertex AI Pipelines to train your model by using a Vertex AI CustomTrainingJoOp component.
2. Use the ModelUploadOp component to upload your model to Vertex AI Model Registry.
3. Use Cloud Scheduler and Cloud Functions to run the Vertex AI pipeline weekly.
D. 1. Create a managed pipeline in Vertex AI Pipelines to train your model using a Vertex AI HyperParameterTuningJobRunOp component.
2. Use the ModelUploadOp component to upload your model to Vertex AI Model Registry.
3. Use Cloud Scheduler and Cloud Functions to run the Vertex AI pipeline weekly.
Answer: C (LEAVE A REPLY)
The best way to operationalize your training process is to use Vertex AI Pipelines, which allows you to create and run scalable, portable, and reproducible workflows for your ML models. Vertex AI Pipelines also integrates with Vertex AI Metadata, which tracks the provenance, lineage, and

artifacts of your ML models. By using a Vertex AI CustomTrainingJobOp component, you can train your model using the same code as in your Jupyter notebook. By using a ModelUploadOp component, you can upload your trained model to Vertex AI Model Registry, which manages the versions and endpoints of your models. By using Cloud Scheduler and Cloud Functions, you can trigger your Vertex AI pipeline to run weekly, according to your plan. Reference:
Vertex AI Pipelines documentation
Vertex AI Metadata documentation
Vertex AI CustomTrainingJobOp documentation
ModelUploadOp documentation
Cloud Scheduler documentation
[Cloud Functions documentation]

NEW QUESTION: 26
You work for a pet food company that manages an online forum Customers upload photos of their pets on the forum to share with others About 20 photos are uploaded daily You want to automatically and in near real time detect whether each uploaded photo has an animal You want to prioritize time and minimize cost of your application development and deployment What should you do?
A. Send user-submitted images to the Cloud Vision API Use object localization to identify all objects in the image and compare the results against a list of animals.
B. Download an object detection model from TensorFlow Hub. Deploy the model to a Vertex AI endpoint. Send new user-submitted images to the model endpoint to classify whether each photo has an animal.
C. Manually label previously submitted images with bounding boxes around any animals Build an AutoML object detection model by using Vertex AI Deploy the model to a Vertex AI endpoint Send new user-submitted images to your model endpoint to detect whether each photo has an animal.
D. Manually label previously submitted images as having animals or not Create an image dataset on Vertex AI Train a classification model by using Vertex AutoML to distinguish the two classes Deploy the model to a Vertex AI endpoint Send new user-submitted images to your model endpoint to classify whether each photo has an animal.
Answer: (SHOW ANSWER)
Cloud Vision API is a service that allows you to analyze images using pre-trained machine learning models1. You can use Cloud Vision API to perform various tasks, such as face detection, text extraction, logo recognition, and object localization1. Object localization is a feature that allows you to detect multiple objects in an image and draw bounding boxes around them2. You can also get the labels and confidence scores for each detected object2.
By sending user-submitted images to the Cloud Vision API, you can use object localization to identify all objects in the image and compare the results against a list of animals. You can use the OBJECT_LOCALIZATION feature type in the AnnotateImageRequest to request object localization3. You can then use the localizedObjectAnnotations field in the AnnotateImageResponse to get the list of detected objects, their labels, and their confidence

scores. You can compare the labels with a predefined list of animals, such as dogs, cats, birds, etc., and determine whether the image has an animal or not.
This option is the best for your scenario, because it allows you to automatically and in near real time detect whether each uploaded photo has an animal, without requiring any manual labeling, model training, or model deployment. You can also prioritize time and minimize cost of your application development and deployment, as you can use the Cloud Vision API as a ready-to-use service, without needing any machine learning expertise or infrastructure.
The other options are not suitable for your scenario, because they either require manual labeling, model training, or model deployment, which would increase the time and cost of your application development and deployment, or they use object detection models, which are more complex and computationally expensive than object localization models, and are not necessary for your simple task of detecting whether an image has an animal or not.
Reference:
Cloud Vision API | Google Cloud
Object localization | Cloud Vision API | Google Cloud
AnnotateImageRequest | Cloud Vision API | Google Cloud
[AnnotateImageResponse | Cloud Vision API | Google Cloud]

NEW QUESTION: 27
You are an ML engineer at a mobile gaming company. A data scientist on your team recently trained a TensorFlow model, and you are responsible for deploying this model into a mobile application. You discover that the inference latency of the current model doesn't meet production requirements. You need to reduce the inference time by 50%, and you are willing to accept a small decrease in model accuracy in order to reach the latency requirement. Without training a new model, which model optimization technique for reducing latency should you try first?
A. Weight pruning
B. Dynamic range quantization
C. Model distillation
D. Dimensionality reduction
Answer: (SHOW ANSWER)
Dynamic range quantization is a model optimization technique for reducing latency that reduces the numerical precision of the weights and activations of models. This technique can reduce the model size, memory usage, and inference time by up to 4x with negligible accuracy loss. Dynamic range quantization can be applied to a trained TensorFlow model without retraining, and it is suitable for mobile applications that require low latency and power consumption.
Weight pruning, model distillation, and dimensionality reduction are also model optimization techniques for reducing latency, but they have some limitations or drawbacks compared to dynamic range quantization:
Weight pruning works by removing parameters within a model that have only a minor impact on its predictions. Pruned models are the same size on disk, and have the same runtime latency, but

can be compressed more effectively. This makes pruning a useful technique for reducing model download size, but not for reducing inference time.
Model distillation works by training a smaller and simpler model (student) to mimic the behavior of a larger and complex model (teacher). Distilled models can have lower latency and memory usage than the original models, but they require retraining and may not preserve the accuracy of the teacher model.
Dimensionality reduction works by reducing the number of features or dimensions in the input data or the model layers. Dimensionality reduction can improve the computational efficiency and generalization ability of models, but it may also lose some information or introduce noise in the data or the model. Dimensionality reduction also requires retraining or modifying the model architecture.
Reference:
[TensorFlow Model Optimization]
[TensorFlow Model Optimization Toolkit - Post-Training Integer Quantization]
[Model optimization methods to cut latency, adapt to new data]

NEW QUESTION: 28
You trained a model on data stored in a Cloud Storage bucket. The model needs to be retrained frequently in Vertex AI Training using the latest data in the bucket. Data preprocessing is required prior to retraining. You want to build a simple and efficient near-real-time ML pipeline in Vertex AI that will preprocess the data when new data arrives in the bucket. What should you do?
A. Create a pipeline using the Vertex AI SDK. Schedule the pipeline with Cloud Scheduler to preprocess the new data in the bucket. Store the processed features in Vertex AI Feature Store.
B. Create a Cloud Run function that is triggered when new data arrives in the bucket. The function initiates a Vertex AI Pipeline to preprocess the new data and store the processed features in Vertex AI Feature Store.
C. Build a Dataflow pipeline to preprocess the new data in the bucket and store the processed features in BigQuery. Configure a cron job to trigger the pipeline execution.
D. Use the Vertex AI SDK to preprocess the new data in the bucket prior to each model retraining. Store the processed features in BigQuery.
Answer: B (LEAVE A REPLY)
Cloud Run can be triggered on new data arrivals, which makes it ideal for near-real-time processing. The function then initiates the Vertex AI Pipeline for preprocessing and storing features in Vertex AI Feature Store, aligning with the retraining needs. Cloud Scheduler (Option A) is suitable for scheduled jobs, not event-driven triggers. Dataflow (Option C) is better suited for batch processing or ETL rather than ML preprocessing pipelines.

NEW QUESTION: 29
You have a large corpus of written support cases that can be classified into 3 separate categories: Technical Support, Billing Support, or Other Issues. You need to quickly build, test,

and deploy a service that will automatically classify future written requests into one of the categories. How should you configure the pipeline?
A. Use the Cloud Natural Language API to obtain metadata to classify the incoming cases.
B. Use AutoML Natural Language to build and test a classifier. Deploy the model as a REST API.
C. Use BigQuery ML to build and test a logistic regression model to classify incoming requests. Use BigQuery ML to perform inference.
D. Create a TensorFlow model using Google's BERT pre-trained model. Build and test a classifier, and deploy the model using Vertex AI.
Answer: (SHOW ANSWER)
AutoML Natural Language is a service that allows you to quickly build, test and deploy natural language processing (NLP) models without needing to have expertise in NLP or machine learning. You can use it to train a classifier on your corpus of written support cases, and then use the AutoML API to perform classification on new requests. Once the model is trained, it can be deployed as a REST API. This allows the classifier to be integrated into your pipeline and be easily consumed by other systems.

NEW QUESTION: 30
You have a custom job that runs on Vertex Al on a weekly basis The job is Implemented using a proprietary ML workflow that produces the datasets. models, and custom artifacts, and sends them to a Cloud Storage bucket Many different versions of the datasets and models were created Due to compliance requirements, your company needs to track which model was used for making a particular prediction, and needs access to the artifacts for each model. How should you configure your workflows to meet these requirement?
A. Register each model in Vertex Al Model Registry, and use model labels to store the related dataset and model information.
B. Use the Vertex Al Metadata API inside the custom Job to create context, execution, and artifacts for each model, and use events to link them together.
C. Configure a TensorFlow Extended (TFX) ML Metadata database, and use the ML Metadata API.
D. Create a Vertex Al experiment, and enable autologging inside the custom job
Answer: (SHOW ANSWER)

NEW QUESTION: 31
You work for a company that is developing an application to help users with meal planning You want to use machine learning to scan a corpus of recipes and extract each ingredient (e g carrot, rice pasta) and each kitchen cookware (e.g. bowl, pot spoon) mentioned Each recipe is saved in an unstructured text file What should you do?
A. Create a text dataset on Vertex Al for entity extraction Create two entities called ingredient" and cookware" and label at least 200 examples of each entity Train an AutoML entity extraction model to extract occurrences of these entity types Evaluate performance on a holdout dataset.

B. Create a multi-label text classification dataset on Vertex AI Create a test dataset and label each recipe that corresponds to its ingredients and cookware Train a multi-class classification model Evaluate the model's performance on a holdout dataset.
C. Use the Entity Analysis method of the Natural Language API to extract the ingredients and cookware from each recipe Evaluate the model's performance on a prelabeled dataset.
D. Create a text dataset on Vertex AI for entity extraction Create as many entities as there are different ingredients and cookware Train an AutoML entity extraction model to extract those entities Evaluate the models performance on a holdout dataset.
Answer: A (LEAVE A REPLY)
Entity extraction is a natural language processing (NLP) task that involves identifying and extracting specific types of information from text, such as names, dates, locations, etc. Entity extraction can help you analyze a corpus of recipes and extract each ingredient and cookware mentioned in them. Vertex AI is a unified platform for building and managing machine learning solutions on Google Cloud. It provides a service for AutoML entity extraction, which allows you to create and train custom entity extraction models without writing any code. You can use Vertex AI to create a text dataset for entity extraction, and label your data with two entities: "ingredient" and "cookware". You need to label at least 200 examples of each entity type to train an AutoML entity extraction model. You can also use a holdout dataset to evaluate the performance of your model, such as precision, recall, and F1-score. This solution can help you build a machine learning model to scan a corpus of recipes and extract each ingredient and cookware mentioned in them, and use the results to help users with meal planning. Reference:
AutoML Entity Extraction | Vertex AI
Preparing data for AutoML Entity Extraction | Vertex AI

NEW QUESTION: 32
Your data science team has requested a system that supports scheduled model retraining, Docker containers, and a service that supports autoscaling and monitoring for online prediction requests. Which platform components should you choose for this system?
A. Vertex AI Pipelines and App Engine
B. Vertex AI Pipelines, Vertex AI Prediction, and Vertex AI Model Monitoring

C. Cloud Composer, BigQuery ML, and Vertex AI Prediction
D. Cloud Composer, Vertex AI Training with custom containers, and App Engine
Answer: B (LEAVE A REPLY)
Option A is incorrect because Vertex AI Pipelines and App Engine do not meet all the requirements of the system. Vertex AI Pipelines is a service that allows you to create, run, and manage ML workflows using TensorFlow Extended (TFX) components or custom components1. App Engine is a service that allows you to build and deploy scalable web applications using standard or flexible environments2. However, App Engine does not support Docker containers in the standard environment, and does not provide a dedicated service for online prediction and monitoring of ML models3.
Option B is correct because Vertex AI Pipelines, Vertex AI Prediction, and Vertex AI Model Monitoring meet all the requirements of the system. Vertex AI Prediction is a service that allows you to deploy and serve ML models for online or batch prediction, with support for autoscaling and custom containers4. Vertex AI Model Monitoring is a service that allows you to monitor the performance and fairness of your deployed models, and get alerts for any issues or anomalies5.
Option C is incorrect because Cloud Composer, BigQuery ML, and Vertex AI Prediction do not meet all the requirements of the system. Cloud Composer is a service that allows you to create, schedule, and manage workflows using Apache Airflow. BigQuery ML is a service that allows you to create and use ML models within BigQuery using SQL queries. However, BigQuery ML does not support custom containers, and Vertex AI Prediction does not support scheduled model retraining or model monitoring.
Option D is incorrect because Cloud Composer, Vertex AI Training with custom containers, and App Engine do not meet all the requirements of the system. Vertex AI Training is a service that allows you to train ML models using built-in algorithms or custom containers. However, Vertex AI Training does not support online prediction or model monitoring, and App Engine does not support Docker containers in the standard environment or online prediction and monitoring of ML models3.
Reference:
Vertex AI Pipelines overview
App Engine overview
Choosing an App Engine environment
Vertex AI Prediction overview
Vertex AI Model Monitoring overview
[Cloud Composer overview]
[BigQuery ML overview]
[BigQuery ML limitations]
[Vertex AI Training overview]

NEW QUESTION: 33
You work for a gaming company that manages a popular online multiplayer game where teams with 6 players play against each other in 5-minute battles. There are many new players every

day. You need to build a model that automatically assigns available players to teams in real time. User research indicates that the game is more enjoyable when battles have players with similar skill levels. Which business metrics should you track to measure your model's performance? (Choose One Correct Answer)

A. Average time players wait before being assigned to a team

B. Precision and recall of assigning players to teams based on their predicted versus actual ability

C. User engagement as measured by the number of battles played daily per user

D. Rate of return as measured by additional revenue generated minus the cost of developing a new model

Answer: C (LEAVE A REPLY)

The best business metric to track to measure the model's performance is user engagement as measured by the number of battles played daily per user. This metric reflects the main goal of the model, which is to enhance the user experience and satisfaction by creating balanced and fair battles. If the model is successful, it should increase the user retention and loyalty, as well as the word-of-mouth and referrals. This metric is also easy to measure and interpret, as it can be directly obtained from the user activity data.

The other options are not optimal for the following reasons:

A . Average time players wait before being assigned to a team is not a good metric, as it does not capture the quality or outcome of the battles. It only measures the efficiency of the model, which is not the primary objective. Moreover, this metric can be influenced by external factors, such as the availability and demand of players, the network latency, and the server capacity.

B . Precision and recall of assigning players to teams based on their predicted versus actual ability is not a good metric, as it is difficult to measure and interpret. It requires having a reliable and consistent way of estimating the player's ability, which can be subjective and dynamic. It also requires having a ground truth label for each assignment, which can be costly and impractical to obtain. Moreover, this metric does not reflect the user feedback or satisfaction, which is the ultimate goal of the model.

D . Rate of return as measured by additional revenue generated minus the cost of developing a new model is not a good metric, as it is not directly related to the model's performance. It measures the profitability of the model, which is a secondary objective. Moreover, this metric can be affected by many other factors, such as the market conditions, the pricing strategy, the marketing campaigns, and the competition.

Reference:

Professional ML Engineer Exam Guide

Preparing for Google Cloud Certification: Machine Learning Engineer Professional Certificate

Google Cloud launches machine learning engineer certification How to measure user engagement How to choose the right metrics for your machine learning model

NEW QUESTION: 34

You are building a TensorFlow text-to-image generative model by using a dataset that contains billions of images with their respective captions. You want to create a low maintenance,

automated workflow that reads the data from a Cloud Storage bucket collects statistics, splits the dataset into training/validation/test datasets performs data transformations, trains the model using the training/validation datasets. and validates the model by using the test dataset. What should you do?

A. Use the Apache Airflow SDK to create multiple operators that use Dataflow and Vertex AI services Deploy the workflow on Cloud Composer.

B. Use the MLFlow SDK and deploy it on a Google Kubernetes Engine Cluster Create multiple components that use Dataflow and Vertex AI services.

C. Use the Kubeflow Pipelines (KFP) SDK to create multiple components that use Dataflow and Vertex AI services Deploy the workflow on Vertex AI Pipelines.

D. Use the TensorFlow Extended (TFX) SDK to create multiple components that use Dataflow and Vertex AI services Deploy the workflow on Vertex AI Pipelines.

Answer: D (LEAVE A REPLY)

According to the web search results, TensorFlow Extended (TFX) is a platform for building end-to-end machine learning pipelines using TensorFlow1. TFX provides a set of components that can be orchestrated using either the TFX SDK or Kubeflow Pipelines. TFX components can handle different aspects of the pipeline, such as data ingestion, data validation, data transformation, model training, model evaluation, model serving, and more. TFX components can also leverage other Google Cloud services, such as Dataflow2 and Vertex AI3. Dataflow is a fully managed service for running Apache Beam pipelines on Google Cloud. Dataflow handles the provisioning and management of the compute resources, as well as the optimization and execution of the pipelines. Vertex AI is a unified platform for machine learning development and deployment. Vertex AI offers various services and tools for building, managing, and serving machine learning models. Therefore, option D is the best way to create a low maintenance, automated workflow for the given use case, as it allows you to use the TFX SDK to define and execute your pipeline components, and use Dataflow and Vertex AI services to scale and optimize your pipeline. The other options are not relevant or optimal for this scenario. Reference:
TensorFlow Extended
Dataflow
Vertex AI
Google Professional Machine Learning Certification Exam 2023
Latest Google Professional Machine Learning Engineer Actual Free Exam Questions

NEW QUESTION: 35

You were asked to investigate failures of a production line component based on sensor readings. After receiving the dataset, you discover that less than 1% of the readings are positive examples representing failure incidents. You have tried to train several classification models, but none of them converge. How should you resolve the class imbalance problem?

A. Use the class distribution to generate 10% positive examples

B. Use a convolutional neural network with max pooling and softmax activation

C. Downsample the data with upweighting to create a sample with 10% positive examples

D. Remove negative examples until the numbers of positive and negative examples are equal

Answer: (SHOW ANSWER)

The class imbalance problem is a common challenge in machine learning, especially in classification tasks. It occurs when the distribution of the target classes is highly skewed, such that one class (the majority class) has much more examples than the other class (the minority class). The minority class is often the more interesting or important class, such as failure incidents, fraud cases, or rare diseases. However, most machine learning algorithms are designed to optimize the overall accuracy, which can be biased towards the majority class and ignore the minority class. This can result in poor predictive performance, especially for the minority class.

There are different techniques to deal with the class imbalance problem, such as data-level methods, algorithm-level methods, and evaluation-level methods1. Data-level methods involve resampling the original dataset to create a more balanced class distribution. There are two main types of data-level methods: oversampling and undersampling. Oversampling methods increase the number of examples in the minority class, either by duplicating existing examples or by generating synthetic examples. Undersampling methods reduce the number of examples in the majority class, either by randomly removing examples or by using clustering or other criteria to select representative examples. Both oversampling and undersampling methods can be combined with upweighting or downweighting, which assign different weights to the examples according to their class frequency, to further balance the dataset.

For the use case of investigating failures of a production line component based on sensor readings, the best option is to downsample the data with upweighting to create a sample with 10% positive examples. This option involves randomly removing some of the negative examples (the majority class) until the ratio of positive to negative examples is 1:9, and then assigning higher weights to the positive examples to compensate for their low frequency. This option can create a more balanced dataset that can improve the performance of the classification models, while preserving the diversity and representativeness of the original data. This option can also reduce the computation time and memory usage, as the size of the dataset is reduced. Therefore, downsampling the data with upweighting to create a sample with 10% positive examples is the best option for this use case.

Reference:

A Systematic Study of the Class Imbalance Problem in Convolutional Neural Networks

NEW QUESTION: 36

You are an ML engineer on an agricultural research team working on a crop disease detection tool to detect leaf rust spots in images of crops to determine the presence of a disease. These spots, which can vary in shape and size, are correlated to the severity of the disease. You want to develop a solution that predicts the presence and severity of the disease with high accuracy.

What should you do?

A. Create an object detection model that can localize the rust spots.

B. Develop an image segmentation ML model to locate the boundaries of the rust spots.

C. Develop a template matching algorithm using traditional computer vision libraries.
D. Develop an image classification ML model to predict the presence of the disease.
Answer: B (LEAVE A REPLY)
The best option for developing a solution that predicts the presence and severity of the disease with high accuracy is to develop an image segmentation ML model to locate the boundaries of the rust spots. Image segmentation is a technique that partitions an image into multiple regions, each corresponding to a different object or semantic category. Image segmentation can be used to detect and localize the rust spots in the images of crops, and measure their shape and size. This information can then be used to determine the presence and severity of the disease, as the rust spots are correlated to the disease symptoms. Image segmentation can also handle the variability of the rust spots, as it does not rely on predefined templates or thresholds. Image segmentation can be implemented using deep learning models, such as U-Net, Mask R-CNN, or DeepLab, which can learn from large-scale datasets and achieve high accuracy and robustness. The other options are not as suitable for developing a solution that predicts the presence and severity of the disease with high accuracy, because:
Creating an object detection model that can localize the rust spots would only provide the bounding boxes of the rust spots, not their exact boundaries. This would result in less precise measurements of the shape and size of the rust spots, and might affect the accuracy of the disease prediction. Object detection models are also more complex and computationally expensive than image segmentation models, as they have to perform both classification and localization tasks.
Developing a template matching algorithm using traditional computer vision libraries would require manually designing and selecting the templates for the rust spots, which might not capture the diversity and variability of the rust spots. Template matching algorithms are also sensitive to noise, occlusion, rotation, and scale changes, and might fail to detect the rust spots in different scenarios. Template matching algorithms are also less accurate and robust than deep learning models, as they do not learn from data.
Developing an image classification ML model to predict the presence of the disease would only provide a binary or categorical output, not the location or severity of the disease. Image classification models are also less informative and interpretable than image segmentation models, as they do not provide any spatial information or visual explanation for the prediction. Image classification models might also suffer from class imbalance or mislabeling issues, as the presence of the disease might not be consistent or clear across the images. Reference:
Image Segmentation | Computer Vision | Google Developers
Crop diseases and pests detection based on deep learning: a review | Plant Methods | Full Text
Using Deep Learning for Image-Based Plant Disease Detection Computer Vision, IoT and Data Fusion for Crop Disease Detection Using ...
On Using Artificial Intelligence and the Internet of Things for Crop ...
Crop Disease Detection Using Machine Learning and Computer Vision

NEW QUESTION: 37

You are training a custom language model for your company using a large dataset. You plan to use the ReductionServer strategy on Vertex AI. You need to configure the worker pools of the distributed training job. What should you do?

A. Configure the machines of the first two worker pools to have GPUs and to use a container image where your training code runs Configure the third worker pool to have GPUs: and use the reduction server container image.

B. Configure the machines of the first two worker pools to have GPUs and to use a container image where your training code runs. Configure the third worker pool to use the reductionserver container image without accelerators, and choose a machine type that prioritizes bandwidth.

C. Configure the machines of the first two worker pools to have TPUs and to use a container image where your training code runs Configure the third worker pool without accelerators, and use the reductionserver container image without accelerators and choose a machine type that prioritizes bandwidth.

D. Configure the machines of the first two pools to have TPUs. and to use a container image where your training code runs Configure the third pool to have TPUs: and use the reductionserver container image.

Answer: B (LEAVE A REPLY)

According to the web search results, Reduction Server is a faster GPU all-reduce algorithm developed at Google that uses a dedicated set of reducers to aggregate gradients from workers12. Reducers are lightweight CPU VM instances that are significantly cheaper than GPU VMs2. Therefore, the third worker pool should not have any accelerators, and should use a machine type that has high network bandwidth to optimize the communication between workers and reducers2. TPUs are not supported by Reduction Server, so the first two worker pools should have GPUs and use a container image that contains the training code12. The reduction-server container image is provided by Google and should be used for the third worker pool2.

NEW QUESTION: 38

While performing exploratory data analysis on a dataset, you find that an important categorical feature has 5% null values. You want to minimize the bias that could result from the missing values. How should you handle the missing values?

A. Remove the rows with missing values, and upsample your dataset by 5%.

B. Replace the missing values with the feature's mean.

C. Replace the missing values with a placeholder category indicating a missing value.

D. Move the rows with missing values to your validation dataset.

Answer: C (LEAVE A REPLY)

The best option for handling missing values in a categorical feature is to replace them with a placeholder category indicating a missing value. This is a type of imputation, which is a method of estimating the missing values based on the observed data. Imputing the missing values with a placeholder category preserves the information that the data is missing, and avoids introducing bias or distortion in the feature distribution. It also allows the machine learning model to learn

from the missingness pattern, and potentially use it as a predictor for the target variable. The other options are not suitable for handling missing values in a categorical feature, because:
Removing the rows with missing values and upsampling the dataset by 5% would reduce the size of the dataset and potentially lose important information. It would also introduce sampling bias and overfitting, as the upsampling process would create duplicate or synthetic observations that do not reflect the true population.
Replacing the missing values with the feature's mean would not make sense for a categorical feature, as the mean is a numerical measure that does not capture the mode or frequency of the categories. It would also create a new category that does not exist in the original data, and might confuse the machine learning model.
Moving the rows with missing values to the validation dataset would compromise the validity and reliability of the model evaluation, as the validation dataset would not be representative of the test or production data. It would also reduce the amount of data available for training the model, and might introduce leakage or inconsistency between the training and validation datasets.
Reference:
Imputation of missing values
Effective Strategies to Handle Missing Values in Data Analysis
How to Handle Missing Values of Categorical Variables?
Google Cloud launches machine learning engineer certification
Google Professional Machine Learning Engineer Certification
Professional ML Engineer Exam Guide
Preparing for Google Cloud Certification: Machine Learning Engineer Professional Certificate

NEW QUESTION: 39
You are an ML engineer at a bank. You have developed a binary classification model using AutoML Tables to predict whether a customer will make loan payments on time. The output is used to approve or reject loan requests. One customer's loan request has been rejected by your model, and the bank's risks department is asking you to provide the reasons that contributed to the model's decision. What should you do?
A. Use local feature importance from the predictions.
B. Use the correlation with target values in the data summary page.
C. Use the feature importance percentages in the model evaluation page.
D. Vary features independently to identify the threshold per feature that changes the classification.
Answer: (SHOW ANSWER)
Option A is correct because using local feature importance from the predictions is the best way to provide the reasons that contributed to the model's decision for a specific customer's loan request. Local feature importance is a measure of how much each feature affects the prediction for a given instance, relative to the average prediction for the dataset1. AutoML Tables provides local feature importance values for each prediction, which can be accessed using the Vertex AI

SDK for Python or the Cloud Console2. By using local feature importance, you can explain why the model rejected the loan request based on the customer's data.
Option B is incorrect because using the correlation with target values in the data summary page is not a good way to provide the reasons that contributed to the model's decision for a specific customer's loan request. The correlation with target values is a measure of how much each feature is linearly related to the target variable for the entire dataset, not for a single instance3. The data summary page in AutoML Tables shows the correlation with target values for each feature, as well as other statistics such as mean, standard deviation, and histogram4. However, these statistics are not useful for explaining the model's decision for a specific customer, as they do not account for the interactions between features or the non-linearity of the model.
Option C is incorrect because using the feature importance percentages in the model evaluation page is not a good way to provide the reasons that contributed to the model's decision for a specific customer's loan request. The feature importance percentages are a measure of how much each feature affects the overall accuracy of the model for the entire dataset, not for a single instance5. The model evaluation page in AutoML Tables shows the feature importance percentages for each feature, as well as other metrics such as precision, recall, and confusion matrix. However, these metrics are not useful for explaining the model's decision for a specific customer, as they do not reflect the individual contribution of each feature for a given prediction.
Option D is incorrect because varying features independently to identify the threshold per feature that changes the classification is not a feasible way to provide the reasons that contributed to the model's decision for a specific customer's loan request. This method involves changing the value of one feature at a time, while keeping the other features constant, and observing how the prediction changes. However, this method is not practical, as it requires making multiple prediction requests, and may not capture the interactions between features or the non-linearity of the model.
Reference:
Local feature importance
Getting local feature importance values
Correlation with target values
Data summary page
Feature importance percentages
[Model evaluation page]
[Varying features independently]

NEW QUESTION: 40
You are an ML engineer at a large grocery retailer with stores in multiple regions. You have been asked to create an inventory prediction model. Your models features include region, location, historical demand, and seasonal popularity. You want the algorithm to learn from new inventory data on a daily basis. Which algorithms should you use to build the model?
A. Classification
B. Reinforcement Learning

C. Recurrent Neural Networks (RNN)
D. Convolutional Neural Networks (CNN)
Answer: B (LEAVE A REPLY)
Reinforcement learning is a machine learning technique that enables an agent to learn from its own actions and feedback in an environment. Reinforcement learning does not require labeled data or explicit rules, but rather relies on trial and error and reward and punishment mechanisms to optimize the agent's behavior and achieve a goal. Reinforcement learning can be used to solve complex and dynamic problems that involve sequential decision making and adaptation to changing situations1.
For the use case of creating an inventory prediction model for a large grocery retailer with stores in multiple regions, reinforcement learning is a suitable algorithm to use. This is because the problem involves multiple factors that affect the inventory demand, such as region, location, historical demand, and seasonal popularity, and the inventory manager needs to make optimal decisions on how much and when to order, store, and distribute the products. Reinforcement learning can help the inventory manager to learn from the new inventory data on a daily basis, and adjust the inventory policy accordingly. Reinforcement learning can also handle the uncertainty and variability of the inventory demand, and balance the trade-off between overstocking and understocking2.
The other options are not as suitable as option B, because they are not designed to handle sequential decision making and adaptation to changing situations. Option A, classification, is a machine learning technique that assigns a label to an input based on predefined categories. Classification can be used to predict the inventory demand for a single product or a single period, but it cannot optimize the inventory policy over multiple products and periods. Option C, recurrent neural networks (RNN), are a type of neural network that can process sequential data, such as text, speech, or time series. RNN can be used to model the temporal patterns and dependencies of the inventory demand, but they cannot learn from feedback and rewards. Option D, convolutional neural networks (CNN), are a type of neural network that can process spatial data, such as images, videos, or graphs. CNN can be used to extract features and patterns from the inventory data, but they cannot optimize the inventory policy over multiple actions and states.
Therefore, option B, reinforcement learning, is the best answer for this question.
Reference:
Reinforcement learning - Wikipedia
Reinforcement Learning for Inventory Optimization

NEW QUESTION: 41
You are an ML engineer at a global car manufacturer. You need to build an ML model to predict car sales in different cities around the world. Which features or feature crosses should you use to train city-specific relationships between car type and number of sales?
A. Three individual features binned latitude, binned longitude, and one-hot encoded car type
B. One feature obtained as an element-wise product between latitude, longitude, and car type

C. One feature obtained as an element-wise product between binned latitude, binned longitude, and one-hot encoded car type
D. Two feature crosses as a element-wise product the first between binned latitude and one-hot encoded car type, and the second between binned longitude and one-hot encoded car type
Answer: (SHOW ANSWER)
A feature cross is a synthetic feature that is obtained by combining two or more existing features, usually by taking their product or concatenation. A feature cross can help to capture the nonlinear and interaction effects between the original features, and improve the predictive performance of the model. A feature cross can be applied to different types of features, such as numeric, categorical, or geospatial features1.
For the use case of building an ML model to predict car sales in different cities around the world, the best option is to use one feature obtained as an element-wise product between binned latitude, binned longitude, and one-hot encoded car type. This option involves creating a feature cross that combines three individual features: binned latitude, binned longitude, and one-hot encoded car type. Binning is a technique that transforms a continuous numeric feature into a discrete categorical feature by dividing its range into equal intervals, or bins. One-hot encoding is a technique that transforms a categorical feature into a binary vector, where each element corresponds to a possible category, and has a value of 1 if the feature belongs to that category, and 0 otherwise. By applying binning and one-hot encoding to the latitude, longitude, and car type features, the feature cross can capture the city-specific relationships between car type and number of sales, as each combination of bins and car types can represent a different city and its preference for a certain car type. For example, the feature cross can learn that a city with a latitude bin of [40, 50], a longitude bin of [-80, -70], and a car type of SUV has a higher number of sales than a city with a latitude bin of [-10, 0], a longitude bin of [10, 20], and a car type of sedan. Therefore, using one feature obtained as an element-wise product between binned latitude, binned longitude, and one-hot encoded car type is the best option for this use case.
Reference:
Feature Crosses | Machine Learning Crash Course

NEW QUESTION: 42
You have trained a DNN regressor with TensorFlow to predict housing prices using a set of predictive features. Your default precision is tf.float64, and you use a standard TensorFlow estimator; estimator tf.estimator.DNNRegressor(feature_columns[YOUR_LIST_OF_FEATURES], hidden_units-[1024, 512, 256], dropoutNone) Your model performs well, but Just before deploying it to production, you discover that your current serving latency is 10ms @ 90 percentile and you currently serve on CPUs. Your production requirements expect a model latency of 8ms @ 90 percentile. You are willing to accept a small decrease in performance in order to reach the latency requirement Therefore your plan is to improve latency while evaluating how much the model's prediction decreases. What should you first try to quickly lower the serving latency?

A. Increase the dropout rate to 0.8 in_PREDICT mode by adjusting the TensorFlow Serving parameters
B. Increase the dropout rate to 0.8 and retrain your model.
C. Switch from CPU to GPU serving
D. Apply quantization to your SavedModel by reducing the floating point precision to tf.float16.
Answer: D (LEAVE A REPLY)
Quantization is a technique that reduces the numerical precision of the weights and activations of a neural network, which can improve the inference speed and reduce the memory footprint of the model1.
Reducing the floating point precision from tf.float64 to tf.float16 can potentially halve the latency and memory usage of the model, while having minimal impact on the accuracy2.
Increasing the dropout rate to 0.8 in either mode would not affect the latency, but would likely degrade the performance of the model significantly, as dropout is a regularization technique that randomly drops out units during training to prevent overfitting3.
Switching from CPU to GPU serving may or may not improve the latency, depending on the hardware specifications and the model complexity, but it would also incur additional costs and complexity for deployment4

NEW QUESTION: 43
You have been asked to develop an input pipeline for an ML training model that processes images from disparate sources at a low latency. You discover that your input data does not fit in memory. How should you create a dataset following Google-recommended best practices?
A. Create a tf.data.Dataset.prefetch transformation
B. Convert the images to tf .Tensor Objects, and then run Dataset. from_tensor_slices{).
C. Convert the images to tf .Tensor Objects, and then run tf. data. Dataset. from_tensors ().
D. Convert the images Into TFRecords, store the images in Cloud Storage, and then use the tf. data API to read the images for training
Answer: (SHOW ANSWER)
An input pipeline is a way to prepare and feed data to a machine learning model for training or inference. An input pipeline typically consists of several steps, such as reading, parsing, transforming, batching, and prefetching the data. An input pipeline can improve the performance and efficiency of the model, as it can handle large and complex datasets, optimize the data processing, and reduce the latency and memory usage1.
For the use case of developing an input pipeline for an ML training model that processes images from disparate sources at a low latency, the best option is to convert the images into TFRecords, store the images in Cloud Storage, and then use the tf.data API to read the images for training. This option involves using the following components and techniques:
TFRecords: TFRecords is a binary file format that can store a sequence of data records, such as images, text, or audio. TFRecords can help to compress, serialize, and store the data efficiently, and reduce the data loading and parsing time. TFRecords can also support data sharding and interleaving, which can improve the data throughput and parallelism2.

Cloud Storage: Cloud Storage is a service that allows you to store and access data on Google Cloud. Cloud Storage can help to store and manage large and distributed datasets, such as images from different sources, and provide high availability, durability, and scalability. Cloud Storage can also integrate with other Google Cloud services, such as Compute Engine, AI Platform, and Dataflow3.
tf.data API: tf.data API is a set of tools and methods that allow you to create and manipulate data pipelines in TensorFlow. tf.data API can help to read, transform, batch, and prefetch the data efficiently, and optimize the data processing for performance and memory. tf.data API can also support various data sources and formats, such as TFRecords, CSV, JSON, and images.
By using these components and techniques, the input pipeline can process large datasets of images from disparate sources that do not fit in memory, and provide low latency and high performance for the ML training model. Therefore, converting the images into TFRecords, storing the images in Cloud Storage, and using the tf.data API to read the images for training is the best option for this use case.
Reference:
Build TensorFlow input pipelines | TensorFlow Core
TFRecord and tf.Example | TensorFlow Core
Cloud Storage documentation | Google Cloud
[tf.data: Build TensorFlow input pipelines | TensorFlow Core]

NEW QUESTION: 44
You are creating a deep neural network classification model using a dataset with categorical input values. Certain columns have a cardinality greater than 10,000 unique values. How should you encode these categorical values as input into the model?
A. Convert each categorical value into an integer value.
B. Convert the categorical string data to one-hot hash buckets.
C. Map the categorical variables into a vector of boolean values.
D. Convert each categorical value into a run-length encoded string.
Answer: B (LEAVE A REPLY)
Option A is incorrect because converting each categorical value into an integer value is not a good way to encode categorical values with high cardinality. This method implies an ordinal relationship between the categories, which may not be true. For example, assigning the values 1, 2, and 3 to the categories "red", "green", and "blue" does not make sense, as there is no inherent order among these colors1.
Option B is correct because converting the categorical string data to one-hot hash buckets is a suitable way to encode categorical values with high cardinality. This method uses a hash function to map each category to a fixed-length vector of binary values, where only one element is 1 and the rest are 0. This method preserves the sparsity and independence of the categories, and reduces the dimensionality of the input space2.
Option C is incorrect because mapping the categorical variables into a vector of boolean values is not a valid way to encode categorical values with high cardinality. This method implies that each

category can be represented by a combination of true/false values, which may not be possible for a large number of categories. For example, if there are 10,000 categories, then there are 2^10,000 possible combinations of boolean values, which is impractical to store and process3. Option D is incorrect because converting each categorical value into a run-length encoded string is not a useful way to encode categorical values with high cardinality. This method compresses a string by replacing consecutive repeated characters with the character and the number of repetitions. For example, "AAAABBBCC" becomes "A4B3C2". This method does not reduce the dimensionality of the input space, and does not preserve the semantic meaning of the categories4.
Reference:
Encoding categorical features
One-hot hash buckets
Boolean vector
Run-length encoding

NEW QUESTION: 45
You work on the data science team at a manufacturing company. You are reviewing the company's historical sales data, which has hundreds of millions of records. For your exploratory data analysis, you need to calculate descriptive statistics such as mean, median, and mode; conduct complex statistical tests for hypothesis testing; and plot variations of the features over time You want to use as much of the sales data as possible in your analyses while minimizing computational resources. What should you do?
A. Spin up a Vertex AI Workbench user-managed notebooks instance and import the dataset Use this data to create statistical and visual analyses
B. Visualize the time plots in Google Data Studio. Import the dataset into Vertex AI Workbench user-managed notebooks Use this data to calculate the descriptive statistics and run the statistical analyses
C. Use BigQuery to calculate the descriptive statistics. Use Vertex AI Workbench user-managed notebooks to visualize the time plots and run the statistical analyses.
Answer: (SHOW ANSWER)
D Use BigQuery to calculate the descriptive statistics, and use Google Data Studio to visualize the time plots. Use Vertex AI Workbench user-managed notebooks to run the statistical analyses.
Explanation:
The best option for analyzing large and complex datasets while minimizing computational resources is to use a combination of BigQuery and Vertex AI Workbench. BigQuery is a serverless, scalable, and cost-effective data warehouse that can perform fast and interactive queries on petabytes of data. BigQuery can calculate descriptive statistics such as mean, median, and mode by using SQL functions such as AVG, PERCENTILE_CONT, and MODE. Vertex AI Workbench is a managed service that provides an integrated development environment for data science and machine learning. Vertex AI Workbench allows users to create and run Jupyter notebooks on Google Cloud, and access various tools and libraries for data visualization and

statistical analysis. Vertex AI Workbench can connect to BigQuery and use the results of the queries to create time plots and run statistical tests for hypothesis testing. By using BigQuery and Vertex AI Workbench, users can leverage the power and flexibility of Google Cloud to perform exploratory data analysis on large and complex datasets. Reference:
Preparing for Google Cloud Certification: Machine Learning Engineer, Course 2: Data Engineering for ML on Google Cloud, Week 1: Introduction to Data Engineering for ML Google Cloud Professional Machine Learning Engineer Exam Guide, Section 1: Architecting low-code ML solutions, 1.1 Developing ML models by using BigQuery ML Official Google Cloud Certified Professional Machine Learning Engineer Study Guide, Chapter 3: Data Engineering for ML, Section 3.2: BigQuery for ML

NEW QUESTION: 46
You are designing an ML recommendation model for shoppers on your company's ecommerce website. You will use Recommendations AI to build, test, and deploy your system. How should you develop recommendations that increase revenue while following best practices?
A. Use the "Other Products You May Like" recommendation type to increase the click-through rate
B. Use the "Frequently Bought Together' recommendation type to increase the shopping cart size for each order.
C. Import your user events and then your product catalog to make sure you have the highest quality event stream
D. Because it will take time to collect and record product data, use placeholder values for the product catalog to test the viability of the model.
Answer: (SHOW ANSWER)
Recommendations AI is a service that allows users to build, test, and deploy personalized product recommendations for their ecommerce websites. It uses Google's deep learning models to learn from user behavior and product data, and generate high-quality recommendations that can increase revenue, click-through rate, and customer satisfaction. One of the best practices for using Recommendations AI is to choose the right recommendation type for the business objective. The "Frequently Bought Together" recommendation type shows products that are often purchased together with the current product, and encourages users to add more items to their shopping cart. This can increase the average order value and the revenue for each transaction. The other options are not as effective or feasible for this objective. The "Other Products You May Like" recommendation type shows products that are similar to the current product, and may increase the click-through rate, but not necessarily the shopping cart size. Importing the user events and then the product catalog is not a recommended order, as it may cause data inconsistency and missing recommendations. The product catalog should be imported first, and then the user events. Using placeholder values for the product catalog is not a viable option, as it will not produce meaningful recommendations or reflect the real performance of the model. Reference:
Recommendations AI documentation

Choosing a recommendation type
Importing data to Recommendations AI

NEW QUESTION: 47
You are working on a prototype of a text classification model in a managed Vertex AI Workbench notebook. You want to quickly experiment with tokenizing text by using a Natural Language Toolkit (NLTK) library. How should you add the library to your Jupyter kernel?
A. Install the NLTK library from a terminal by using the pip install nltk command.
B. Write a custom Dataflow job that uses NLTK to tokenize your text and saves the output to Cloud Storage.
C. Create a new Vertex AI Workbench notebook with a custom image that includes the NLTK library.
D. Install the NLTK library from a Jupyter cell by using the! pip install nltk -user command.
Answer: D (LEAVE A REPLY)
NLTK is a Python library that provides a set of tools for natural language processing, such as tokenization, stemming, tagging, parsing, and sentiment analysis. Tokenization is a process of breaking a text into smaller units, such as words or sentences. You can use NLTK to quickly experiment with tokenizing text in a managed Vertex AI Workbench notebook. A Vertex AI Workbench notebook is a web-based interactive environment that allows you to write and execute Python code on Google Cloud. You can install the NLTK library from a Jupyter cell by using the ! pip install nltk --user command. This command uses the pip package manager to install the NLTK library for the current user. By installing the NLTK library from a Jupyter cell, you can avoid the hassle of opening a terminal or creating a custom image for your notebook. Reference:
NLTK documentation
Vertex AI Workbench documentation
Preparing for Google Cloud Certification: Machine Learning Engineer Professional Certificate

NEW QUESTION: 48

You have trained a text classification model in TensorFlow using AI Platform. You want to use the trained model for batch predictions on text data stored in BigQuery while minimizing computational overhead. What should you do?

A. Export the model to BigQuery ML.

B. Deploy and version the model on AI Platform.

C. Use Dataflow with the SavedModel to read the data from BigQuery

D. Submit a batch prediction job on AI Platform that points to the model location in Cloud Storage.

Answer: D (LEAVE A REPLY)

This answer is correct because it allows you to use the trained TensorFlow model for batch predictions on text data stored in BigQuery without any additional processing or overhead. AI Platform provides a service for running batch prediction jobs that can take input data from BigQuery or Cloud Storage and write the output to BigQuery or Cloud Storage. You can use the SavedModel format to export your TensorFlow model to Cloud Storage and then submit a batch prediction job that points to the model location and the input data location. AI Platform will handle the scaling and distribution of the prediction requests and return the results in the specified output location. Reference:

[AI Platform: Batch prediction overview]

[AI Platform: Exporting a SavedModel for prediction]

NEW QUESTION: 49

Your company manages an application that aggregates news articles from many different online sources and sends them to users. You need to build a recommendation model that will suggest articles to readers that are similar to the articles they are currently reading. Which approach should you use?

A. Create a collaborative filtering system that recommends articles to a user based on the user's past behavior.

B. Encode all articles into vectors using word2vec, and build a model that returns articles based on vector similarity.

C. Build a logistic regression model for each user that predicts whether an article should be recommended to a user.

D. Manually label a few hundred articles, and then train an SVM classifier based on the manually classified articles that categorizes additional articles into their respective categories.

Answer: B (LEAVE A REPLY)

Option A is incorrect because creating a collaborative filtering system that recommends articles to a user based on the user's past behavior is not the best approach to suggest articles that are similar to the articles they are currently reading. Collaborative filtering is a method of recommendation that uses the ratings or preferences of other users to predict the preferences of a target user1. However, this method does not consider the content or features of the articles, and may not be able to find articles that are similar in terms of topic, style, or sentiment.

Option B is correct because encoding all articles into vectors using word2vec, and building a model that returns articles based on vector similarity is a suitable approach to suggest articles

that are similar to the articles they are currently reading. Word2vec is a technique that learns low-dimensional and dense representations of words from a large corpus of text, such that words that are semantically similar have similar vectors2. By applying word2vec to the articles, we can obtain vector representations of the articles that capture their meaning and usage. Then, we can use a similarity measure, such as cosine similarity, to find articles that have similar vectors to the current article3.
Option C is incorrect because building a logistic regression model for each user that predicts whether an article should be recommended to a user is not a feasible approach to suggest articles that are similar to the articles they are currently reading. Logistic regression is a supervised learning method that models the probability of a binary outcome (such as recommend or not) based on some input features (such as user profile or article content)4. However, this method requires a large amount of labeled data for each user, which may not be available or scalable. Moreover, this method does not directly measure the similarity between articles, but rather the likelihood of a user's preference.
Option D is incorrect because manually labeling a few hundred articles, and then training an SVM classifier based on the manually classified articles that categorizes additional articles into their respective categories is not an effective approach to suggest articles that are similar to the articles they are currently reading. SVM (support vector machine) is a supervised learning method that finds a hyperplane that separates the data into different classes (such as news categories) with the maximum margin5. However, this method also requires a large amount of labeled data, which may be costly and time-consuming to obtain. Moreover, this method does not account for the fine-grained similarity between articles within the same category, or the cross-category similarity between articles from different categories.
Reference:
Collaborative filtering
Word2vec
Cosine similarity
Logistic regression
SVM

NEW QUESTION: 50
You work at an organization that maintains a cloud-based communication platform that integrates conventional chat, voice, and video conferencing into one platform. The audio recordings are stored in Cloud Storage. All recordings have an 8 kHz sample rate and are more than one minute long. You need to implement a new feature in the platform that will automatically transcribe voice call recordings into a text for future applications, such as call summarization and sentiment analysis. How should you implement the voice call transcription feature following Google-recommended best practices?
A. Upsample the audio recordings to 16 kHz. and transcribe the audio by using the Speech-to-Text API with asynchronous recognition.

B. Upsample the audio recordings to 16 kHz. and transcribe the audio by using the Speech-to-Text API with synchronous recognition.
C. Use the original audio sampling rate, and transcribe the audio by using the Speech-to-Text API with asynchronous recognition.
D. Use the original audio sampling rate, and transcribe the audio by using the Speech-to-Text API with synchronous recognition.
Answer: A (LEAVE A REPLY)

NEW QUESTION: 51
You are an ML engineer at a manufacturing company. You need to build a model that identifies defects in products based on images of the product taken at the end of the assembly line. You want your model to preprocess the images with lower computation to quickly extract features of defects in products. Which approach should you use to build the model?
A. Reinforcement learning
B. Recommender system
C. Recurrent Neural Networks (RNN)
D. Convolutional Neural Networks (CNN)
Answer: D (LEAVE A REPLY)
Option A is incorrect because reinforcement learning is not a suitable approach to build a model that identifies defects in products based on images of the product taken at the end of the assembly line. Reinforcement learning is a type of machine learning that learns from its own actions and rewards, rather than from labeled data or explicit feedback1. Reinforcement learning is more suitable for problems that involve sequential decision making, such as games, robotics, or control systems1. However, defect detection is a problem that involves image classification or segmentation, which requires supervised learning, not reinforcement learning.
Option B is incorrect because a recommender system is not a relevant approach to build a model that identifies defects in products based on images of the product taken at the end of the assembly line. A recommender system is a system that suggests items or actions to users based on their preferences, behavior, or context2. A recommender system is more suitable for problems that involve personalization, such as e-commerce, entertainment, or social media2. However, defect detection is a problem that involves image classification or segmentation, which requires supervised learning, not recommender system.
Option C is incorrect because recurrent neural networks (RNN) are not the most efficient approach to build a model that identifies defects in products based on images of the product taken at the end of the assembly line. RNNs are a type of neural networks that can process sequential data, such as text, speech, or video, by maintaining a hidden state that captures the temporal dependencies3. RNNs are more suitable for problems that involve natural language processing, speech recognition, or video analysis3. However, defect detection is a problem that involves image classification or segmentation, which does not require temporal dependencies, but rather spatial dependencies. Moreover, RNNs are computationally expensive and prone to vanishing or exploding gradients4.

Option D is correct because convolutional neural networks (CNN) are the best approach to build a model that identifies defects in products based on images of the product taken at the end of the assembly line. CNNs are a type of neural networks that can process image data, by applying convolutional filters that extract local features and reduce the dimensionality of the data5. CNNs are more suitable for problems that involve image classification, object detection, or segmentation5. CNNs can preprocess the images with lower computation to quickly extract features of defects in products, by using techniques such as pooling, dropout, or batch normalization6.
Reference:
Reinforcement learning
Recommender system
Recurrent neural network
Vanishing and exploding gradients
Convolutional neural network
CNN techniques
[Defect detection]
[Image classification]
[Image segmentation]

NEW QUESTION: 52
You work for a retail company. You have been asked to develop a model to predict whether a customer will purchase a product on a given day. Your team has processed the company's sales data, and created a table with the following rows:
* Customer_id
* Product_id
* Date
* Days_since_last_purchase (measured in days)
* Average_purchase_frequency (measured in 1/days)
* Purchase (binary class, if customer purchased product on the Date)
You need to interpret your models results for each individual prediction. What should you do?
A. Create a BigQuery table Use BigQuery ML to build a boosted tree classifier Inspect the partition rules of the trees to understand how each prediction flows through the trees.
B. Create a Vertex AI tabular dataset Train an AutoML model to predict customer purchases Deploy the model to a Vertex AI endpoint and enable feature attributions Use the "explain" method to get feature attribution values for each individual prediction.
C. Create a BigQuery table Use BigQuery ML to build a logistic regression classification model Use the values of the coefficients of the model to interpret the feature importance with higher values corresponding to more importance.
D. Create a Vertex AI tabular dataset Train an AutoML model to predict customer purchases Deploy the model to a Vertex AI endpoint. At each prediction enable L1 regularization to detect non-informative features.

Answer: (SHOW ANSWER)
According to the official exam guide1, one of the skills assessed in the exam is to "explain the predictions of a trained model". Vertex AI provides feature attributions using Shapley Values, a cooperative game theory algorithm that assigns credit to each feature in a model for a particular outcome2. Feature attributions can help you understand how the model calculates the predictions and debug or optimize the model accordingly. You can use AutoML for Tabular Data to generate and query local feature attributions3. The other options are not relevant or optimal for this scenario. Reference:
Professional ML Engineer Exam Guide
Feature attributions for classification and regression
AutoML for Tabular Data
Google Professional Machine Learning Certification Exam 2023
Latest Google Professional Machine Learning Engineer Actual Free Exam Questions

NEW QUESTION: 53
You are profiling the performance of your TensorFlow model training time and notice a performance issue caused by inefficiencies in the input data pipeline for a single 5 terabyte CSV file dataset on Cloud Storage. You need to optimize the input pipeline performance. Which action should you try first to increase the efficiency of your pipeline?
A. Preprocess the input CSV file into a TFRecord file.
B. Randomly select a 10 gigabyte subset of the data to train your model.
C. Split into multiple CSV files and use a parallel interleave transformation.
D. Set the reshuffle_each_iteration parameter to true in the tf.data.Dataset.shuffle method.
Answer: A (LEAVE A REPLY)
According to the web search results, the TFRecord format is a recommended way to store large amounts of data efficiently and improve the performance of the data input pipeline123. The TFRecord format is a binary format that can be compressed and serialized, which reduces the I/O overhead and the memory footprint of the data1. The tf.data API provides tools to create and read TFRecord files easily1.
The other options are not as effective as option A. Option B would reduce the amount of data available for training and might affect the model accuracy. Option C would still require reading from a single CSV file at a time, which might not utilize the full bandwidth of the remote storage. Option D would only affect the order of the data elements, not the speed of reading them.

NEW QUESTION: 54
You have recently trained a scikit-learn model that you plan to deploy on Vertex AI. This model will support both online and batch prediction. You need to preprocess input data for model inference. You want to package the model for deployment while minimizing additional code What should you do?
A. 1 Upload your model to the Vertex AI Model Registry by using a prebuilt scikit-learn prediction container

2 Deploy your model to Vertex AI Endpoints, and create a Vertex AI batch prediction job that uses the instanceConfig.inscanceType setting to transform your input data

B. 1 Wrap your model in a custom prediction routine (CPR). and build a container image from the CPR local model

2 Upload your sci-kit learn model container to Vertex AI Model Registry

3 Deploy your model to Vertex AI Endpoints, and create a Vertex AI batch prediction job

C. 1. Create a custom container for your sci-kit learn model,

2 Define a custom serving function for your model

3 Upload your model and custom container to Vertex AI Model Registry

4 Deploy your model to Vertex AI Endpoints, and create a Vertex AI batch prediction job

D. 1 Create a custom container for your sci-kit learn model.

2 Upload your model and custom container to Vertex AI Model Registry

3 Deploy your model to Vertex AI Endpoints, and create a Vertex AI batch prediction job that uses the instanceConfig. instanceType setting to transform your input data

Answer: (SHOW ANSWER)

The best option for deploying a scikit-learn model on Vertex AI with minimal additional code is to wrap the model in a custom prediction routine (CPR) and build a container image from the CPR local model. Upload your scikit-learn model container to Vertex AI Model Registry. Deploy your model to Vertex AI Endpoints, and create a Vertex AI batch prediction job. This option allows you to leverage the power and simplicity of Google Cloud to deploy and serve a scikit-learn model that supports both online and batch prediction. Vertex AI is a unified platform for building and deploying machine learning solutions on Google Cloud. Vertex AI can deploy a trained scikit-learn model to an online prediction endpoint, which can provide low-latency predictions for individual instances. Vertex AI can also create a batch prediction job, which can provide high-throughput predictions for a large batch of instances. A custom prediction routine (CPR) is a Python script that defines the logic for preprocessing the input data, running the prediction, and postprocessing the output data. A CPR can help you customize the prediction behavior of your model, and handle complex or non-standard data formats. A CPR can also help you minimize the additional code, as you only need to write a few functions to implement the prediction logic. A container image is a package that contains the model, the CPR, and the dependencies. A container image can help you standardize and simplify the deployment process, as you only need to upload the container image to Vertex AI Model Registry, and deploy it to Vertex AI Endpoints. By wrapping the model in a CPR and building a container image from the CPR local model, uploading the scikit-learn model container to Vertex AI Model Registry, deploying the model to Vertex AI Endpoints, and creating a Vertex AI batch prediction job, you can deploy a scikit-learn model on Vertex AI with minimal additional code1.

The other options are not as good as option B, for the following reasons:

Option A: Uploading your model to the Vertex AI Model Registry by using a prebuilt scikit-learn prediction container, deploying your model to Vertex AI Endpoints, and creating a Vertex AI batch prediction job that uses the instanceConfig.instanceType setting to transform your input data would not allow you to preprocess the input data for model inference, and could cause errors or

poor performance. A prebuilt scikit-learn prediction container is a container image that is provided by Google Cloud, and contains the scikit-learn framework and the dependencies. A prebuilt scikit-learn prediction container can help you deploy a scikit-learn model without writing any code, but it also limits your customization options. A prebuilt scikit-learn prediction container can only handle standard data formats, such as JSON or CSV, and cannot perform any preprocessing or postprocessing on the input or output data. If your input data requires any transformation or normalization before running the prediction, you cannot use a prebuilt scikit-learn prediction container. The instanceConfig.instanceType setting is a parameter that determines the machine type and the accelerator type for the batch prediction job. The instanceConfig.instanceType setting can help you optimize the performance and the cost of the batch prediction job, but it cannot help you transform your input data2.

Option C: Creating a custom container for your scikit-learn model, defining a custom serving function for your model, uploading your model and custom container to Vertex AI Model Registry, and deploying your model to Vertex AI Endpoints, and creating a Vertex AI batch prediction job would require more skills and steps than using a CPR and a container image. A custom container is a container image that contains the model, the dependencies, and a web server. A custom container can help you customize the prediction behavior of your model, and handle complex or non-standard data formats. A custom serving function is a Python function that defines the logic for running the prediction on the model. A custom serving function can help you implement the prediction logic of your model, and handle complex or non-standard data formats. However, creating a custom container and defining a custom serving function would require more skills and steps than using a CPR and a container image. You would need to write code, build and test the container image, configure the web server, and implement the prediction logic. Moreover, creating a custom container and defining a custom serving function would not allow you to preprocess the input data for model inference, as the custom serving function only runs the prediction on the model3.

Option D: Creating a custom container for your scikit-learn model, uploading your model and custom container to Vertex AI Model Registry, deploying your model to Vertex AI Endpoints, and creating a Vertex AI batch prediction job that uses the instanceConfig.instanceType setting to transform your input data would not allow you to preprocess the input data for model inference, and could cause errors or poor performance. A custom container is a container image that contains the model, the dependencies, and a web server. A custom container can help you customize the prediction behavior of your model, and handle complex or non-standard data formats. However, creating a custom container would require more skills and steps than using a CPR and a container image. You would need to write code, build and test the container image, and configure the web server. The instanceConfig.instanceType setting is a parameter that determines the machine type and the accelerator type for the batch prediction job. The instanceConfig.instanceType setting can help you optimize the performance and the cost of the batch prediction job, but it cannot help you transform your input data23.

Reference:

Preparing for Google Cloud Certification: Machine Learning Engineer, Course 3: Production ML Systems, Week 2: Serving ML Predictions Google Cloud Professional Machine Learning Engineer Exam Guide, Section 3: Scaling ML models in production, 3.1 Deploying ML models to production Official Google Cloud Certified Professional Machine Learning Engineer Study Guide, Chapter 6: Production ML Systems, Section 6.2: Serving ML Predictions Custom prediction routines Using pre-built containers for prediction Using custom containers for prediction

NEW QUESTION: 55
You are building a linear model with over 100 input features, all with values between -1 and 1. You suspect that many features are non-informative. You want to remove the non-informative features from your model while keeping the informative ones in their original form. Which technique should you use?
A. Use Principal Component Analysis to eliminate the least informative features.
B. Use L1 regularization to reduce the coefficients of uninformative features to 0.
C. After building your model, use Shapley values to determine which features are the most informative.
D. Use an iterative dropout technique to identify which features do not degrade the model when removed.
Answer: B (LEAVE A REPLY)
L1 regularization, also known as Lasso regularization, adds the sum of the absolute values of the model's coefficients to the loss function1. It encourages sparsity in the model by shrinking some coefficients to precisely zero2. This way, L1 regularization can perform feature selection and remove the non-informative features from the model while keeping the informative ones in their original form. Therefore, using L1 regularization is the best technique for this use case.
Reference:
Regularization in Machine Learning - GeeksforGeeks
Regularization in Machine Learning (with Code Examples) - Dataquest
L1 And L2 Regularization Explained & Practical How To Examples
L1 and L2 as Regularization for a Linear Model

NEW QUESTION: 56
You work as an ML researcher at an investment bank and are experimenting with the Gemini large language model (LLM). You plan to deploy the model for an internal use case and need full control of the model's underlying infrastructure while minimizing inference time. Which serving configuration should you use for this task?
A. Deploy the model on a Google Kubernetes Engine (GKE) cluster manually by creating a custom YAML manifest.
B. Deploy the model on a Vertex AI endpoint manually by creating a custom inference container.
C. Deploy the model on a Google Kubernetes Engine (GKE) cluster using the deployment options in Model Garden.
D. Deploy the model on a Vertex AI endpoint using one-click deployment in Model Garden.

Answer: A (LEAVE A REPLY)

NEW QUESTION: 57
You work for a company that sells corporate electronic products to thousands of businesses worldwide. Your company stores historical customer data in BigQuery. You need to build a model that predicts customer lifetime value over the next three years. You want to use the simplest approach to build the model. What should you do?
A. Access BigQuery Studio in the Google Cloud console. Run the create model statement in the SQL editor to create an ARIMA model.
B. Create a Vertex AI Workbench notebook. Use IPython magic to run the create model statement to create an ARIMA model.
C. Access BigQuery Studio in the Google Cloud console. Run the create model statement in the SQL editor to create an AutoML regression model.
D. Create a Vertex AI Workbench notebook. Use IPython magic to run the create model statement to create an AutoML regression model.
Answer: (SHOW ANSWER)
* BigQuery ML allows you to build and run machine learning models using SQL queries directly within BigQuery, which is one of the simplest approaches because it doesn't require setting up an external environment like Vertex AI or managing infrastructure.
* AutoML regression is more appropriate for predicting customer lifetime value (CLV) compared to ARIMA, which is typically used for time series forecasting (e.g., sales over time, stock prices, etc.). CLV prediction involves understanding complex relationships between customer behavior and value, which is best captured by a regression model.
* Using BigQuery Studio and running a CREATE MODEL statement to build an AutoML regression model offers the simplicity you're looking for because it automates much of the feature engineering, model selection, and hyperparameter tuning.
* The other options involving ARIMA models (A and B) are not appropriate for CLV, and setting up a Vertex AI Workbench notebook (D) introduces unnecessary complexity for this task.

NEW QUESTION: 58
You work for a retailer that sells clothes to customers around the world. You have been tasked with ensuring that ML models are built in a secure manner. Specifically, you need to protect sensitive customer data that might be used in the models. You have identified four fields containing sensitive data that are being used by your data science team: AGE, IS_EXISTING_CUSTOMER, LATITUDE_LONGITUDE, and SHIRT_SIZE. What should you do with the data before it is made available to the data science team for training purposes?
A. Tokenize all of the fields using hashed dummy values to replace the real values.
B. Use principal component analysis (PCA) to reduce the four sensitive fields to one PCA vector.
C. Coarsen the data by putting AGE into quantiles and rounding LATITUDE_LONGTTUDE into single precision. The other two fields are already as coarse as possible.

D. Remove all sensitive data fields, and ask the data science team to build their models using non-sensitive data.

Answer: (SHOW ANSWER)

The best option for protecting sensitive customer data that might be used in the ML models is to coarsen the data by putting AGE into quantiles and rounding LATITUDE_LONGITUDE into single precision. This option has the following advantages:

It preserves the utility and relevance of the data for the ML models, as the coarsened data still captures the essential information and patterns that the models need to learn. For example, putting AGE into quantiles can group the customers into different age ranges, which can be useful for predicting their preferences or behavior. Rounding LATITUDE_LONGITUDE into single precision can reduce the precision of the location data, but still retain the general geographic region of the customers, which can be useful for personalizing the recommendations or offers.

It reduces the risk of exposing the personal or private information of the customers, as the coarsened data makes it harder to identify or re-identify the individual customers from the data. For example, putting AGE into quantiles can hide the exact age of the customers, which can be considered sensitive or confidential. Rounding LATITUDE_LONGITUDE into single precision can obscure the exact location of the customers, which can be considered sensitive or confidential.

The other options are less optimal for the following reasons:

Option A: Tokenizing all of the fields using hashed dummy values to replace the real values eliminates the utility and relevance of the data for the ML models, as the tokenized data loses all the information and patterns that the models need to learn. For example, tokenizing AGE using hashed dummy values can make the data meaningless and irrelevant, as the models cannot learn anything from the random tokens. Tokenizing LATITUDE_LONGITUDE using hashed dummy values can make the data meaningless and irrelevant, as the models cannot learn anything from the random tokens.

Option B: Using principal component analysis (PCA) to reduce the four sensitive fields to one PCA vector reduces the utility and relevance of the data for the ML models, as the PCA vector may not capture all the information and patterns that the models need to learn. For example, using PCA to reduce AGE, IS_EXISTING_CUSTOMER, LATITUDE_LONGITUDE, and SHIRT_SIZE to one PCA vector can lose some information or introduce noise in the data, as the PCA vector is a linear combination of the original features, which may not reflect their true relationship or importance. Moreover, using PCA to reduce the four sensitive fields to one PCA vector may not reduce the risk of exposing the personal or private information of the customers, as the PCA vector may still be reversible or linkable to the original data, depending on the amount of variance explained by the PCA vector and the availability of the PCA transformation matrix.

Option D: Removing all sensitive data fields, and asking the data science team to build their models using non-sensitive data reduces the utility and relevance of the data for the ML models, as the non-sensitive data may not contain enough information and patterns that the models need to learn. For example, removing AGE, IS_EXISTING_CUSTOMER, LATITUDE_LONGITUDE, and SHIRT_SIZE from the data can make the data insufficient and unrepresentative, as the models may not be able to learn the factors that influence the customers' preferences or

behavior. Moreover, removing all sensitive data fields from the data may not be necessary or feasible, as the data protection legislation may allow the use of sensitive data for the ML models, as long as the data is processed in a secure and ethical manner, and the customers' consent and rights are respected.
Reference:
Protecting Sensitive Data and AI Models with Confidential Computing | NVIDIA Technical Blog Training machine learning models from sensitive data | Fast Data Science Securing ML applications. Model security and protection - Medium Security of AI/ML systems, ML model security | Cossack Labs Vulnerabilities, security and privacy for machine learning models

NEW QUESTION: 59
You are developing an ML model in a Vertex AI Workbench notebook. You want to track artifacts and compare models during experimentation using different approaches. You need to rapidly and easily transition successful experiments to production as you iterate on your model implementation. What should you do?
A. 1 Initialize the Vertex SDK with the name of your experiment Log parameters and metrics for each experiment, and attach dataset and model artifacts as inputs and outputs to each execution.
2 After a successful experiment create a Vertex AI pipeline.
B. 1. Initialize the Vertex SDK with the name of your experiment Log parameters and metrics for each experiment, save your dataset to a Cloud Storage bucket and upload the models to Vertex AI Model Registry.
2 After a successful experiment create a Vertex AI pipeline.
C. 1 Create a Vertex AI pipeline with parameters you want to track as arguments to your Pipeline Job Use the Metrics. Model, and Dataset artifact types from the Kubeflow Pipelines DSL as the inputs and outputs of the components in your pipeline.
2. Associate the pipeline with your experiment when you submit the job.
D. 1 Create a Vertex AI pipeline Use the Dataset and Model artifact types from the Kubeflow Pipelines. DSL as the inputs and outputs of the components in your pipeline.
2. In your training component use the Vertex AI SDK to create an experiment run Configure the log_params and log_metrics functions to track parameters and metrics of your experiment.
Answer: A (LEAVE A REPLY)
Vertex AI is a unified platform for building and managing machine learning solutions on Google Cloud. It provides various services and tools for different stages of the machine learning lifecycle, such as data preparation, model training, deployment, monitoring, and experimentation. Vertex AI Workbench is an integrated development environment (IDE) that allows you to create and run Jupyter notebooks on Google Cloud. You can use Vertex AI Workbench to develop your ML model in Python, using libraries such as TensorFlow, PyTorch, scikit-learn, etc. You can also use the Vertex SDK, which is a Python client library for Vertex AI, to track artifacts and compare models during experimentation. You can use the aiplatform.init function to initialize the Vertex SDK with the name of your experiment. You can use the aiplatform.start_run and aiplatform.end_run functions to create and close an experiment run. You can use the

aiplatform.log_params and aiplatform.log_metrics functions to log the parameters and metrics for each experiment run. You can also use the aiplatform.log_datasets and aiplatform.log_model functions to attach the dataset and model artifacts as inputs and outputs to each experiment run. These functions allow you to record and store the metadata and artifacts of your experiments, and compare them using the Vertex AI Experiments UI. After a successful experiment, you can create a Vertex AI pipeline, which is a way to automate and orchestrate your ML workflows. You can use the aiplatform.PipelineJob class to create a pipeline job, and specify the components and dependencies of your pipeline. You can also use the aiplatform.CustomContainerTrainingJob class to create a custom container training job, and use the run method to run the job as a pipeline component. You can use the aiplatform.Model.deploy method to deploy your model as a pipeline component. You can also use the aiplatform.Model.monitor method to monitor your model as a pipeline component. By creating a Vertex AI pipeline, you can rapidly and easily transition successful experiments to production, and reuse and share your ML workflows. This solution requires minimal changes to your code, and leverages the Vertex AI services and tools to streamline your ML development process. Reference: The answer can be verified from official Google Cloud documentation and resources related to Vertex AI, Vertex AI Workbench, Vertex SDK, and Vertex AI pipelines.
Vertex AI | Google Cloud
Vertex AI Workbench | Google Cloud
Vertex SDK for Python | Google Cloud
Vertex AI pipelines | Google Cloud

NEW QUESTION: 60
You work for a gaming company that develops massively multiplayer online (MMO) games. You built a TensorFlow model that predicts whether players will make in-app purchases of more than $10 in the next two weeks. The model's predictions will be used to adapt each user's game experience. User data is stored in BigQuery. How should you serve your model while optimizing cost, user experience, and ease of management?
A. Import the model into BigQuery ML. Make predictions using batch reading data from BigQuery, and push the data to Cloud SQL
B. Deploy the model to Vertex AI Prediction. Make predictions using batch reading data from Cloud Bigtable, and push the data to Cloud SQL.
C. Embed the model in the mobile application. Make predictions after every in-app purchase event is published in Pub/Sub, and push the data to Cloud SQL.
D. Embed the model in the streaming Dataflow pipeline. Make predictions after every in-app purchase event is published in Pub/Sub, and push the data to Cloud SQL.
Answer: (SHOW ANSWER)
The best option to serve the model while optimizing cost, user experience, and ease of management is to deploy the model to Vertex AI Prediction, which is a managed service that can scale up or down according to the demand and provide low latency and high availability. Vertex AI Prediction can also handle TensorFlow models natively, without requiring any additional steps or

conversions. By using batch prediction, the model can process large volumes of data efficiently and periodically, without affecting the user experience. The data can be read from Cloud Bigtable, which is a scalable and performant NoSQL database that can store user data in a flexible schema. The predictions can then be pushed to Cloud SQL, which is a fully managed relational database that can store the predictions in a structured format and enable easy querying and analysis. This option also simplifies the management of the model and the data, as it leverages the existing Google Cloud services and does not require any additional infrastructure or code.
The other options are not optimal for the following reasons:
A . Importing the model into BigQuery ML is not a good option, as it requires converting the TensorFlow model into a format that BigQuery ML can understand, which can introduce errors and reduce the performance. Moreover, BigQuery ML is not designed for serving real-time predictions, but rather for training and evaluating models using SQL queries. Reading and writing data from BigQuery and Cloud SQL can also incur additional costs and latency, as they are both relational databases that require schema definition and data transformation.
C . Embedding the model in the mobile application is not a good option, as it increases the size and complexity of the application, and requires updating the application every time the model changes. Moreover, it exposes the model to the users, which can pose security and privacy risks, as well as potential misuse or abuse. Additionally, it does not leverage the benefits of the cloud, such as scalability, reliability, and performance.
D . Embedding the model in the streaming Dataflow pipeline is not a good option, as it requires building and maintaining a custom pipeline that can handle the model inference and data processing. This can increase the development and operational costs and complexity, as well as the potential for errors and failures. Moreover, it does not take advantage of the batch prediction feature of Vertex AI Prediction, which can optimize the resource utilization and cost efficiency.
Reference:
Professional ML Engineer Exam Guide
Preparing for Google Cloud Certification: Machine Learning Engineer Professional Certificate
Google Cloud launches machine learning engineer certification Vertex AI Prediction documentation Cloud Bigtable documentation Cloud SQL documentation

NEW QUESTION: 61
You work on a data science team at a bank and are creating an ML model to predict loan default risk. You have collected and cleaned hundreds of millions of records worth of training data in a BigQuery table, and you now want to develop and compare multiple models on this data using TensorFlow and Vertex AI. You want to minimize any bottlenecks during the data ingestion state while considering scalability. What should you do?
A. Use the BigQuery client library to load data into a dataframe, and use tf.data.Dataset.from_tensor_slices() to read it.
B. Export data to CSV files in Cloud Storage, and use tf.data.TextLineDataset() to read them.
C. Convert the data into TFRecords, and use tf.data.TFRecordDataset() to read them.
D. Use TensorFlow I/O's BigQuery Reader to directly read the data.

Answer: D (LEAVE A REPLY)

The best option for developing and comparing multiple models on a large-scale BigQuery table using TensorFlow and Vertex AI is to use TensorFlow I/O's BigQuery Reader to directly read the data. This option has the following advantages:

It minimizes any bottlenecks during the data ingestion stage, as the BigQuery Reader can stream data from BigQuery to TensorFlow in parallel and in batches, without loading the entire table into memory or disk. The BigQuery Reader can also perform data transformations and filtering using SQL queries, reducing the need for additional preprocessing steps in TensorFlow.

It leverages the scalability and performance of BigQuery, as the BigQuery Reader can handle hundreds of millions of records worth of training data efficiently and reliably. BigQuery is a serverless, fully managed, and highly scalable data warehouse that can run complex queries over petabytes of data in seconds.

It simplifies the integration with Vertex AI, as the BigQuery Reader can be used with both custom and pre-built TensorFlow models on Vertex AI. Vertex AI is a unified platform for machine learning that provides various tools and features for data ingestion, data labeling, data preprocessing, model training, model tuning, model deployment, model monitoring, and model explainability.

The other options are less optimal for the following reasons:

Option A: Using the BigQuery client library to load data into a dataframe, and using tf.data.Dataset.from_tensor_slices() to read it, introduces memory and performance issues. This option requires loading the entire BigQuery table into a Pandas dataframe, which can consume a lot of memory and cause out-of-memory errors. Moreover, using tf.data.Dataset.from_tensor_slices() to read the dataframe can be slow and inefficient, as it creates one slice per row of the dataframe, resulting in a large number of small tensors.

Option B: Exporting data to CSV files in Cloud Storage, and using tf.data.TextLineDataset() to read them, introduces additional steps and complexity. This option requires exporting the BigQuery table to one or more CSV files in Cloud Storage, which can take a long time and consume a lot of storage space. Moreover, using tf.data.TextLineDataset() to read the CSV files can be slow and error-prone, as it requires parsing and decoding each line of text, handling missing values and invalid data, and applying data transformations and validations.

Option C: Converting the data into TFRecords, and using tf.data.TFRecordDataset() to read them, introduces additional steps and complexity. This option requires converting the BigQuery table into one or more TFRecord files, which are binary files that store serialized TensorFlow examples. This can take a long time and consume a lot of storage space. Moreover, using tf.data.TFRecordDataset() to read the TFRecord files requires defining and parsing the schema of the TensorFlow examples, which can be tedious and error-prone.

Reference:

[TensorFlow I/O documentation]

[BigQuery documentation]

[Vertex AI documentation]

NEW QUESTION: 62
You developed a Transformer model in TensorFlow to translate text Your training data includes millions of documents in a Cloud Storage bucket. You plan to use distributed training to reduce training time. You need to configure the training job while minimizing the effort required to modify code and to manage the clusters configuration. What should you do?
A. Create a Vertex AI custom training job with GPU accelerators for the second worker pool Use tf .distribute.MultiWorkerMirroredStrategy for distribution.
B. Create a Vertex AI custom distributed training job with Reduction Server Use N1 high-memory machine type instances for the first and second pools, and use N1 high-CPU machine type instances for the third worker pool.
C. Create a training job that uses Cloud TPU VMs Use tf.distribute.TPUStrategy for distribution.
D. Create a Vertex AI custom training job with a single worker pool of A2 GPU machine type instances Use tf .distribute.MirroredStraregy for distribution.
Answer: C (LEAVE A REPLY)
According to the official exam guide1, one of the skills assessed in the exam is to "configure and optimize model training jobs". Cloud TPU VMs2 are a new way to access Cloud TPUs directly on the TPU host machines, offering a simpler and more flexible user experience. Cloud TPU VMs are optimized for ML model training and can reduce training time and cost. You can use Cloud TPU VMs to train Transformer models in TensorFlow by using the tf.distribute.TPUStrategy3, which handles the distribution of computations across the TPU cores. The other options are not relevant or optimal for this scenario. Reference:
Professional ML Engineer Exam Guide
Cloud TPU VMs
Distributed training with TPUStrategy
Google Professional Machine Learning Certification Exam 2023
Latest Google Professional Machine Learning Engineer Actual Free Exam Questions

NEW QUESTION: 63
You work for an online retail company that is creating a visual search engine. You have set up an end-to-end ML pipeline on Google Cloud to classify whether an image contains your company's

product. Expecting the release of new products in the near future, you configured a retraining functionality in the pipeline so that new data can be fed into your ML models. You also want to use AI Platform's continuous evaluation service to ensure that the models have high accuracy on your test data set. What should you do?

A. Keep the original test dataset unchanged even if newer products are incorporated into retraining

B. Extend your test dataset with images of the newer products when they are introduced to retraining

C. Replace your test dataset with images of the newer products when they are introduced to retraining.

D. Update your test dataset with images of the newer products when your evaluation metrics drop below a pre-decided threshold.

Answer: B (LEAVE A REPLY)

The test dataset is used to evaluate the performance of the ML model on unseen data. It should reflect the distribution of the data that the model will encounter in production. Therefore, if the retraining data includes new products, the test dataset should also be extended with images of those products to ensure that the model can generalize well to them. Keeping the original test dataset unchanged or replacing it entirely with images of the new products would not capture the diversity of the data that the model needs to handle. Updating the test dataset only when the evaluation metrics drop below a threshold would be reactive rather than proactive, and might result in poor user experience if the model fails to recognize the new products. Reference:
Continuous evaluation documentation
Preparing and using test sets

NEW QUESTION: 64

Your data science team is training a PyTorch model for image classification based on a pre-trained RestNet model. You need to perform hyperparameter tuning to optimize for several parameters. What should you do?

A. Convert the model to a Keras model, and run a Keras Tuner job.

B. Run a hyperparameter tuning job on AI Platform using custom containers.

C. Create a Kuberflow Pipelines instance, and run a hyperparameter tuning job on Katib.

D. Convert the model to a TensorFlow model, and run a hyperparameter tuning job on AI Platform.

Answer: B (LEAVE A REPLY)

AI Platform supports hyperparameter tuning for PyTorch models using custom containers. This allows you to use any Python dependencies and libraries that are not included in the pre-built AI Platform Training runtime versions. You can also use a pre-trained model such as ResNet as a base for your custom model. To run a hyperparameter tuning job on AI Platform using custom containers, you need to do the following steps:

Create a Dockerfile that defines the container image for your training application. The Dockerfile should install PyTorch and any other dependencies, copy your training code and configuration files, and set the entrypoint for the container.
Build the container image and push it to Container Registry or another accessible registry.
Create a YAML file that defines the configuration for your hyperparameter tuning job. The YAML file should specify the container image URI, the training input and output paths, the hyperparameters to tune, the metric to optimize, and the tuning algorithm and budget.
Submit the hyperparameter tuning job to AI Platform using the gcloud command-line tool or the AI Platform Training API.
Reference:
Hyperparameter tuning overview
Using custom containers
PyTorch on AI Platform Training

NEW QUESTION: 65
You work for a hospital that wants to optimize how it schedules operations. You need to create a model that uses the relationship between the number of surgeries scheduled and beds used You want to predict how many beds will be needed for patients each day in advance based on the scheduled surgeries You have one year of data for the hospital organized in 365 rows The data includes the following variables for each day
* Number of scheduled surgeries
* Number of beds occupied
* Date
You want to maximize the speed of model development and testing What should you do?
A. Create a BigQuery table Use BigQuery ML to build a regression model, with number of beds as the target variable and number of scheduled surgeries and date features (such as day of week) as the predictors
B. Create a BigQuery table Use BigQuery ML to build an ARIMA model, with number of beds as the target variable and date as the time variable.
C. Create a Vertex AI tabular dataset Tram an AutoML regression model, with number of beds as the target variable and number of scheduled minor surgeries and date features (such as day of the week) as the predictors
D. Create a Vertex AI tabular dataset Train a Vertex AI AutoML Forecasting model with number of beds as the target variable, number of scheduled surgeries as a covariate, and date as the time variable.
Answer: D (LEAVE A REPLY)
According to the official exam guide1, one of the skills assessed in the exam is to "design, build, and productionalize ML models to solve business challenges using Google Cloud technologies". Vertex AI AutoML Forecasting2 is a service that allows you to train and deploy custom time-series forecasting models for batch prediction. Vertex AI AutoML Forecasting simplifies the model development process by providing a graphical user interface and a no-code approach. You can

use Vertex AI AutoML Forecasting to train a model by using your tabular data, and specify the target variable, the covariates, and the time variable. Vertex AI AutoML Forecasting automatically handles the feature engineering, model selection, and hyperparameter tuning. Therefore, option D is the best way to maximize the speed of model development and testing for the given use case. The other options are not relevant or optimal for this scenario. Reference:
Professional ML Engineer Exam Guide
Vertex AI AutoML Forecasting
Google Professional Machine Learning Certification Exam 2023
Latest Google Professional Machine Learning Engineer Actual Free Exam Questions

NEW QUESTION: 66
You have a functioning end-to-end ML pipeline that involves tuning the hyperparameters of your ML model using AI Platform, and then using the best-tuned parameters for training. Hypertuning is taking longer than expected and is delaying the downstream processes. You want to speed up the tuning job without significantly compromising its effectiveness. Which actions should you take?
Choose 2 answers
A. Decrease the number of parallel trials
B. Decrease the range of floating-point values
C. Set the early stopping parameter to TRUE
D. Change the search algorithm from Bayesian search to random search.
E. Decrease the maximum number of trials during subsequent training phases.
Answer: C,E (LEAVE A REPLY)
Hyperparameter tuning is the process of finding the optimal values for the parameters of a machine learning model that affect its performance. AI Platform provides a service for hyperparameter tuning that can run multiple trials in parallel and use different search algorithms to find the best combination of hyperparameters. However, hyperparameter tuning can be time-consuming and costly, especially if the search space is large and the model training is complex. Therefore, it is important to optimize the tuning job to reduce the time and resources required. One way to speed up the tuning job is to set the early stopping parameter to TRUE. This means that the tuning service will automatically stop trials that are unlikely to perform well based on the intermediate results. This can save time and resources by avoiding unnecessary computations for trials that are not promising. The early stopping parameter can be set in the trainingInput.hyperparameters field of the training job request1 Another way to speed up the tuning job is to decrease the maximum number of trials during subsequent training phases. This means that the tuning service will use fewer trials to refine the search space after the initial phase. This can reduce the time required for the tuning job to converge to the optimal solution. The maximum number of trials can be set in the trainingInput.hyperparameters.maxTrials field of the training job request1 The other options are not effective ways to speed up the tuning job. Decreasing the number of parallel trials will reduce the concurrency of the tuning job and increase the overall time required. Decreasing the range of floating-point values will reduce the diversity of

the search space and may miss some optimal solutions. Changing the search algorithm from Bayesian search to random search will reduce the efficiency of the tuning job and may require more trials to find the best solution1 References: 1: Hyperparameter tuning overview

NEW QUESTION: 67
You work on the data science team for a multinational beverage company. You need to develop an ML model to predict the company's profitability for a new line of naturally flavored bottled waters in different locations. You are provided with historical data that includes product types, product sales volumes, expenses, and profits for all regions. What should you use as the input and output for your model?
A. Use latitude, longitude, and product type as features. Use profit as model output.
B. Use latitude, longitude, and product type as features. Use revenue and expenses as model outputs.
C. Use product type and the feature cross of latitude with longitude, followed by binning, as features. Use profit as model output.
D. Use product type and the feature cross of latitude with longitude, followed by binning, as features. Use revenue and expenses as model outputs.
Answer: C (LEAVE A REPLY)
Option A is incorrect because using latitude, longitude, and product type as features, and using profit as model output is not the best way to develop an ML model to predict the company's profitability for a new line of naturally flavored bottled waters in different locations. This option does not capture the interaction between latitude and longitude, which may affect the profitability of the product. For example, the same product may have different profitability in different regions, depending on the climate, culture, or preferences of the customers. Moreover, this option does not account for the granularity of the location data, which may be too fine or too coarse for the model. For example, using the exact coordinates of a city may not be meaningful, as the profitability may vary within the city, or using the country name may not be informative, as the profitability may vary across the country.
Option B is incorrect because using latitude, longitude, and product type as features, and using revenue and expenses as model outputs is not a suitable way to develop an ML model to predict the company's profitability for a new line of naturally flavored bottled waters in different locations. This option has the same drawbacks as option A, as it does not capture the interaction between latitude and longitude, or account for the granularity of the location data. Moreover, this option does not directly predict the profitability of the product, which is the target variable of interest. Instead, it predicts the revenue and expenses of the product, which are intermediate variables that depend on other factors, such as the price, the cost, or the demand of the product. To obtain the profitability, we would need to subtract the expenses from the revenue, which may introduce errors or uncertainties in the prediction.
Option C is correct because using product type and the feature cross of latitude with longitude, followed by binning, as features, and using profit as model output is a good way to develop an ML model to predict the company's profitability for a new line of naturally flavored bottled waters in

different locations. This option captures the interaction between latitude and longitude, which may affect the profitability of the product, by creating a feature cross of these two features. A feature cross is a synthetic feature that combines the values of two or more features into a single feature1. This option also accounts for the granularity of the location data, by binning the feature cross into discrete buckets. Binning is a technique that groups continuous values into intervals, which can reduce the noise and complexity of the data2. Moreover, this option directly predicts the profitability of the product, which is the target variable of interest, by using it as the model output.
Option D is incorrect because using product type and the feature cross of latitude with longitude, followed by binning, as features, and using revenue and expenses as model outputs is not a valid way to develop an ML model to predict the company's profitability for a new line of naturally flavored bottled waters in different locations. This option has the same advantages as option C, as it captures the interaction between latitude and longitude, and accounts for the granularity of the location data, by creating a feature cross and binning it. However, this option does not directly predict the profitability of the product, which is the target variable of interest, but rather predicts the revenue and expenses of the product, which are intermediate variables that depend on other factors, as explained in option B.
Reference:
Feature cross
Binning
[Profitability]
[Revenue and expenses]
[Latitude and longitude]
[Product type]

NEW QUESTION: 68
You are training models in Vertex AI by using data that spans across multiple Google Cloud Projects You need to find track, and compare the performance of the different versions of your models Which Google Cloud services should you include in your ML workflow?
A. Dataplex. Vertex AI Feature Store and Vertex AI TensorBoard
B. Vertex AI Pipelines, Vertex AI Feature Store, and Vertex AI Experiments
C. Dataplex. Vertex AI Experiments, and Vertex AI ML Metadata
D. Vertex AI Pipelines: Vertex AI Experiments and Vertex AI Metadata
Answer: B (LEAVE A REPLY)
Vertex AI Pipelines is a service that allows you to orchestrate and automate your machine learning (ML) workflows using pipelines1. A pipeline is a description of an ML workflow, including all of the components in the workflow, how the components are connected as a graph, and the runtime parameters that the pipeline accepts1. Vertex AI Pipelines helps you manage the end-to-end lifecycle of your ML projects, from data preprocessing to model deployment1.
Vertex AI Feature Store is a service that enables you to serve, share, and reuse ML features across different models and projects2. A feature is a measurable property or characteristic of an

entity, such as the age of a person or the price of a product2. Vertex AI Feature Store helps you reduce data duplication, ensure data consistency, and improve model performance2.
Vertex AI Experiments is a service that helps you track and compare the performance of different versions of your models3. You can use Vertex AI Experiments to run multiple training jobs with different hyperparameters, architectures, or data sources, and then compare the results using metrics, visualizations, and reports3. Vertex AI Experiments helps you identify the best model for your use case and optimize your model performance3. Reference:
Vertex AI Pipelines | Google Cloud
Vertex AI Feature Store | Google Cloud
Vertex AI Experiments | Google Cloud

NEW QUESTION: 69
You are developing a recommendation engine for an online clothing store. The historical customer transaction data is stored in BigQuery and Cloud Storage. You need to perform exploratory data analysis (EDA), preprocessing and model training. You plan to rerun these EDA, preprocessing, and training steps as you experiment with different types of algorithms. You want to minimize the cost and development effort of running these steps as you experiment. How should you configure the environment?
A. Create a Vertex AI Workbench user-managed notebook using the default VM instance, and use the %%bigquery magic commands in Jupyter to query the tables.
B. Create a Vertex AI Workbench managed notebook to browse and query the tables directly from the JupyterLab interface.
C. Create a Vertex AI Workbench user-managed notebook on a Dataproc Hub. and use the %%bigquery magic commands in Jupyter to query the tables.
D. Create a Vertex AI Workbench managed notebook on a Dataproc cluster, and use the spark-bigquery-connector to access the tables.
Answer: (SHOW ANSWER)
Cost-effectiveness: User-managed notebooks in Vertex AI Workbench allow you to leverage pre-configured virtual machines with reasonable resource allocation, keeping costs lower compared to options involving managed notebooks or Dataproc clusters.
Development flexibility: User-managed notebooks offer full control over the environment, allowing you to install additional libraries or dependencies needed for your specific EDA, preprocessing, and model training tasks. This flexibility is crucial while experimenting with different algorithms.
BigQuery integration: The %%bigquery magic commands provide seamless integration with BigQuery within the Jupyter Notebook environment. This enables efficient querying and exploration of customer transaction data stored in BigQuery directly from the notebook, streamlining the workflow.
Other options and why they are not the best fit:
B . Managed notebook: While managed notebooks offer an easier setup, they might have limited customization options, potentially hindering your ability to install specific libraries or tools.

C . Dataproc Hub: Dataproc Hub focuses on running large-scale distributed workloads, and it might be overkill for your scenario involving exploratory analysis and experimentation with different algorithms. Additionally, it could incur higher costs compared to a user-managed notebook.
D . Dataproc cluster with spark-bigquery-connector: Similar to option C, using a Dataproc cluster with the spark-bigquery-connector would be more complex and potentially more expensive than using %%bigquery magic commands within a user-managed notebook for accessing BigQuery data.
Reference:
https://cloud.google.com/vertex-ai/docs/workbench/instances/bigquery
https://cloud.google.com/vertex-ai-notebooks

NEW QUESTION: 70
You built a deep learning-based image classification model by using on-premises dat a. You want to use Vertex AI to deploy the model to production Due to security concerns you cannot move your data to the cloud. You are aware that the input data distribution might change over time You need to detect model performance changes in production. What should you do?
A. Use Vertex Explainable AI for model explainability Configure feature-based explanations.
B. Use Vertex Explainable AI for model explainability Configure example-based explanations.
C. Create a Vertex AI Model Monitoring job. Enable training-serving skew detection for your model.
D. Create a Vertex AI Model Monitoring job. Enable feature attribution skew and dnft detection for your model.
Answer: C (LEAVE A REPLY)
Vertex AI Model Monitoring is a service that allows you to monitor the performance and quality of your ML models in production. You can use Vertex AI Model Monitoring to detect changes in the input data distribution, the prediction output distribution, or the model accuracy over time. Training-serving skew detection is a feature of Vertex AI Model Monitoring that compares the statistics of the data used for training the model and the data used for serving the model. If there is a significant difference between the two data distributions, it indicates that the model might be outdated or inaccurate. By enabling training-serving skew detection for your model, you can detect model performance changes in production and trigger retraining or redeployment of your model as needed. This way, you can ensure that your model is always up-to-date and accurate, without moving your data to the cloud. Reference:
Vertex AI Model Monitoring documentation
Training-serving skew detection documentation
Preparing for Google Cloud Certification: Machine Learning Engineer Professional Certificate

NEW QUESTION: 71

You have trained a model by using data that was preprocessed in a batch Dataflow pipeline Your use case requires real-time inference. You want to ensure that the data preprocessing logic is applied consistently between training and serving. What should you do?

A. Perform data validation to ensure that the input data to the pipeline is the same format as the input data to the endpoint.

B. Refactor the transformation code in the batch data pipeline so that it can be used outside of the pipeline Use the same code in the endpoint.

C. Refactor the transformation code in the batch data pipeline so that it can be used outside of the pipeline Share this code with the end users of the endpoint.

D. Batch the real-time requests by using a time window and then use the Dataflow pipeline to preprocess the batched requests. Send the preprocessed requests to the endpoint.

Answer: B (LEAVE A REPLY)

According to the official exam guide1, one of the skills assessed in the exam is to "design, build, and productionalize ML models to solve business challenges using Google Cloud technologies". Dataflow2 is a fully managed, fast, and easy-to-use service for running Apache Spark and Apache Hadoop clusters on Google Cloud. Dataflow supports both batch and streaming data processing pipelines. However, if your use case requires real-time inference, you need to ensure that the data preprocessing logic is applied consistently between training and serving. One way to achieve this is to refactor the transformation code in the batch data pipeline so that it can be used outside of the pipeline, and use the same code in the endpoint. This way, you can avoid data skew and drift issues that might arise from using different preprocessing methods for training and serving. Therefore, option B is the best way to ensure the data preprocessing logic is applied consistently between training and serving. The other options are not relevant or optimal for this scenario. Reference:

Professional ML Engineer Exam Guide

Dataflow

Google Professional Machine Learning Certification Exam 2023

Latest Google Professional Machine Learning Engineer Actual Free Exam Questions

NEW QUESTION: 72

You are developing a mode! to detect fraudulent credit card transactions. You need to prioritize detection because missing even one fraudulent transaction could severely impact the credit card holder. You used AutoML to tram a model on users' profile information and credit card transaction dat a. After training the initial model, you notice that the model is failing to detect many fraudulent transactions. How should you adjust the training parameters in AutoML to improve model performance?

Choose 2 answers

A. Increase the score threshold.

B. Decrease the score threshold.

C. Add more positive examples to the training set.

D. Add more negative examples to the training set.

E. Reduce the maximum number of node hours for training.

Answer: B,C (LEAVE A REPLY)

The best options for adjusting the training parameters in AutoML to improve model performance are to decrease the score threshold and add more positive examples to the training set. These options can help increase the detection rate of fraudulent transactions, which is the priority for this use case. The score threshold is a parameter that determines the minimum probability score that a prediction must have to be classified as positive. Decreasing the score threshold can increase the recall of the model, which is the proportion of actual positive cases that are correctly identified. Increasing the recall can help reduce the number of false negatives, which are fraudulent transactions that are missed by the model. However, decreasing the score threshold can also decrease the precision of the model, which is the proportion of positive predictions that are actually correct. Decreasing the precision can increase the number of false positives, which are legitimate transactions that are flagged as fraudulent by the model. Therefore, there is a trade-off between recall and precision, and the optimal score threshold depends on the business objective and the cost of errors1. Adding more positive examples to the training set can help balance the data distribution and improve the model performance. Positive examples are the instances that belong to the target class, which in this case are fraudulent transactions. Negative examples are the instances that belong to the other class, which in this case are legitimate transactions. Fraudulent transactions are usually rare and imbalanced compared to legitimate transactions, which can cause the model to be biased towards the majority class and fail to learn the characteristics of the minority class. Adding more positive examples can help the model learn more features and patterns of the fraudulent transactions, and increase the detection rate2.

The other options are not as good as options B and C, for the following reasons:

Option A: Increasing the score threshold would decrease the detection rate of fraudulent transactions, which is the opposite of the desired outcome. Increasing the score threshold would decrease the recall of the model, which is the proportion of actual positive cases that are correctly identified. Decreasing the recall would increase the number of false negatives, which are fraudulent transactions that are missed by the model. Increasing the score threshold would increase the precision of the model, which is the proportion of positive predictions that are actually correct. Increasing the precision would decrease the number of false positives, which are legitimate transactions that are flagged as fraudulent by the model. However, in this use case, the cost of false negatives is much higher than the cost of false positives, so increasing the score threshold is not a good option1.

Option D: Adding more negative examples to the training set would not improve the model performance, and could worsen the data imbalance. Negative examples are the instances that belong to the other class, which in this case are legitimate transactions. Legitimate transactions are usually abundant and dominant compared to fraudulent transactions, which can cause the model to be biased towards the majority class and fail to learn the characteristics of the minority class. Adding more negative examples would exacerbate this problem, and decrease the detection rate of the fraudulent transactions2.

Option E: Reducing the maximum number of node hours for training would not improve the model performance, and could limit the model optimization. Node hours are the units of computation that are used to train an AutoML model. The maximum number of node hours is a parameter that determines the upper limit of node hours that can be used for training. Reducing the maximum number of node hours would reduce the training time and cost, but also the model quality and accuracy. Reducing the maximum number of node hours would limit the number of iterations, trials, and evaluations that the model can perform, and prevent the model from finding the optimal hyperparameters and architecture3.
Reference:
Preparing for Google Cloud Certification: Machine Learning Engineer, Course 5: Responsible AI, Week 4: Evaluation Google Cloud Professional Machine Learning Engineer Exam Guide, Section 2: Developing high-quality ML models, 2.2 Handling imbalanced data Official Google Cloud Certified Professional Machine Learning Engineer Study Guide, Chapter 4: Low-code ML Solutions, Section 4.3: AutoML Understanding the score threshold slider Handling imbalanced data sets in machine learning AutoML Vision pricing

NEW QUESTION: 73
You are developing an ML model to identify your company s products in images. You have access to over one million images in a Cloud Storage bucket. You plan to experiment with different TensorFlow models by using Vertex AI Training You need to read images at scale during training while minimizing data I/O bottlenecks What should you do?
A. Load the images directly into the Vertex AI compute nodes by using Cloud Storage FUSE Read the images by using the tf .data.Dataset.from_tensor_slices function.
B. Create a Vertex AI managed dataset from your image data Access the aip_training_data_uri environment variable to read the images by using the tf. data. Dataset. list_flies function.
C. Convert the images to TFRecords and store them in a Cloud Storage bucket Read the TFRecords by using the tf. ciata.TFRecordDataset function.
D. Store the URLs of the images in a CSV file Read the file by using the tf.data.experomental.CsvDataset function.
Answer: C (LEAVE A REPLY)
TFRecords are a binary file format that can store large amounts of data efficiently. By converting the images to TFRecords and storing them in a Cloud Storage bucket, you can reduce the data size and improve the data transfer speed. You can then read the TFRecords by using the tf.data.TFRecordDataset function, which creates a dataset of tensors from the TFRecord files. This way, you can read images at scale during training while minimizing data I/O bottlenecks.
Reference:
TFRecord documentation
tf.data.TFRecordDataset documentation
Preparing for Google Cloud Certification: Machine Learning Engineer Professional Certificate

NEW QUESTION: 74

You work on an operations team at an international company that manages a large fleet of on-premises servers located in few data centers around the world. Your team collects monitoring data from the servers, including CPU/memory consumption. When an incident occurs on a server, your team is responsible for fixing it. Incident data has not been properly labeled yet. Your management team wants you to build a predictive maintenance solution that uses monitoring data from the VMs to detect potential failures and then alerts the service desk team. What should you do first?

A. Train a time-series model to predict the machines' performance values. Configure an alert if a machine's actual performance values significantly differ from the predicted performance values.

B. Implement a simple heuristic (e.g., based on z-score) to label the machines' historical performance data. Train a model to predict anomalies based on this labeled dataset.

C. Develop a simple heuristic (e.g., based on z-score) to label the machines' historical performance data. Test this heuristic in a production environment.

D. Hire a team of qualified analysts to review and label the machines' historical performance data. Train a model based on this manually labeled dataset.

Answer: (SHOW ANSWER)

Option A is incorrect because training a time-series model to predict the machines' performance values, and configuring an alert if a machine's actual performance values significantly differ from the predicted performance values, is not the best way to build a predictive maintenance solution that uses monitoring data from the VMs to detect potential failures and then alerts the service desk team. This option assumes that the performance values follow a predictable pattern, which may not be the case for complex systems. Moreover, this option does not use any historical incident data, which may contain useful information for identifying failures. Furthermore, this option does not involve any model evaluation or validation, which are essential steps for ensuring the quality and reliability of the model.

Option B is correct because implementing a simple heuristic (e.g., based on z-score) to label the machines' historical performance data, and training a model to predict anomalies based on this labeled dataset, is a reasonable way to build a predictive maintenance solution that uses monitoring data from the VMs to detect potential failures and then alerts the service desk team. This option uses a simple and fast method to label the historical performance data, which is necessary for supervised learning. A z-score is a measure of how many standard deviations a value is away from the mean of a distribution1. By using a z-score, we can label the performance values that are unusually high or low as anomalies, which may indicate failures. Then, we can train a model to learn the patterns of normal and anomalous performance values, and use it to predict anomalies on new data. We can also evaluate and validate the model using metrics such as precision, recall, or F1-score, and compare it with other models or methods.

Option C is incorrect because developing a simple heuristic (e.g., based on z-score) to label the machines' historical performance data, and testing this heuristic in a production environment, is not a safe way to build a predictive maintenance solution that uses monitoring data from the VMs to detect potential failures and then alerts the service desk team. This option does not involve any model training or evaluation, which are essential steps for ensuring the quality and reliability of

the solution. Moreover, this option does not test the heuristic on a separate dataset, such as a validation or test set, before deploying it to production, which may lead to errors or failures in the production environment.
Option D is incorrect because hiring a team of qualified analysts to review and label the machines' historical performance data, and training a model based on this manually labeled dataset, is not a feasible way to build a predictive maintenance solution that uses monitoring data from the VMs to detect potential failures and then alerts the service desk team. This option may produce high-quality labels, but it is also costly, time-consuming, and prone to human errors or biases. Moreover, this option may not scale well with large or complex datasets, which may require more analysts or more time to label.
Reference:
Z-score
[Predictive maintenance]
[Anomaly detection]
[Time-series analysis]
[Model evaluation]

NEW QUESTION: 75
You need to deploy a scikit-learn classification model to production. The model must be able to serve requests 24/7 and you expect millions of requests per second to the production application from 8 am to 7 pm. You need to minimize the cost of deployment What should you do?
A. Deploy an online Vertex AI prediction endpoint Set the max replica count to 1
B. Deploy an online Vertex AI prediction endpoint Set the max replica count to 100
C. Deploy an online Vertex AI prediction endpoint with one GPU per replica Set the max replica count to 1.
D. Deploy an online Vertex AI prediction endpoint with one GPU per replica Set the max replica count to 100.
Answer: (SHOW ANSWER)
The best option for deploying a scikit-learn classification model to production is to deploy an online Vertex AI prediction endpoint and set the max replica count to 100. This option allows you to leverage the power and scalability of Google Cloud to serve requests 24/7 and handle millions of requests per second. Vertex AI is a unified platform for building and deploying machine learning solutions on Google Cloud. Vertex AI can deploy a trained scikit-learn model to an online prediction endpoint, which can provide low-latency predictions for individual instances. An online prediction endpoint consists of one or more replicas, which are copies of the model that run on virtual machines. The max replica count is a parameter that determines the maximum number of replicas that can be created for the endpoint. By setting the max replica count to 100, you can enable the endpoint to scale up to 100 replicas when the traffic increases, and scale down to zero replicas when the traffic decreases. This can help minimize the cost of deployment, as you only pay for the resources that you use. Moreover, you can use the autoscaling algorithm option to optimize the scaling behavior of the endpoint based on the latency and utilization metrics1.

The other options are not as good as option B, for the following reasons:
Option A: Deploying an online Vertex AI prediction endpoint and setting the max replica count to 1 would not be able to serve requests 24/7 and handle millions of requests per second. Setting the max replica count to 1 would limit the endpoint to only one replica, which can cause performance issues and service disruptions when the traffic increases. Moreover, setting the max replica count to 1 would prevent the endpoint from scaling down to zero replicas when the traffic decreases, which can increase the cost of deployment, as you pay for the resources that you do not use1.
Option C: Deploying an online Vertex AI prediction endpoint with one GPU per replica and setting the max replica count to 1 would not be able to serve requests 24/7 and handle millions of requests per second, and would increase the cost of deployment. Adding a GPU to each replica would increase the computational power of the endpoint, but it would also increase the cost of deployment, as GPUs are more expensive than CPUs. Moreover, setting the max replica count to 1 would limit the endpoint to only one replica, which can cause performance issues and service disruptions when the traffic increases, and prevent the endpoint from scaling down to zero replicas when the traffic decreases1. Furthermore, scikit-learn models do not benefit from GPUs, as scikit-learn is not optimized for GPU acceleration2.
Option D: Deploying an online Vertex AI prediction endpoint with one GPU per replica and setting the max replica count to 100 would be able to serve requests 24/7 and handle millions of requests per second, but it would increase the cost of deployment. Adding a GPU to each replica would increase the computational power of the endpoint, but it would also increase the cost of deployment, as GPUs are more expensive than CPUs. Setting the max replica count to 100 would enable the endpoint to scale up to 100 replicas when the traffic increases, and scale down to zero replicas when the traffic decreases, which can help minimize the cost of deployment. However, scikit-learn models do not benefit from GPUs, as scikit-learn is not optimized for GPU acceleration2. Therefore, using GPUs for scikit-learn models would be unnecessary and wasteful.
Reference:
Preparing for Google Cloud Certification: Machine Learning Engineer, Course 3: Production ML Systems, Week 2: Serving ML Predictions Google Cloud Professional Machine Learning Engineer Exam Guide, Section 3: Scaling ML models in production, 3.1 Deploying ML models to production Official Google Cloud Certified Professional Machine Learning Engineer Study Guide, Chapter 6: Production ML Systems, Section 6.2: Serving ML Predictions Online prediction Scaling online prediction scikit-learn FAQ

NEW QUESTION: 76
You manage a team of data scientists who use a cloud-based backend system to submit training jobs. This system has become very difficult to administer, and you want to use a managed service instead. The data scientists you work with use many different frameworks, including Keras, PyTorch, theano. Scikit-team, and custom libraries. What should you do?
A. Use the AI Platform custom containers feature to receive training jobs using any framework
B. Configure Kubeflow to run on Google Kubernetes Engine and receive training jobs through TFJob

C. Create a library of VM images on Compute Engine; and publish these images on a centralized repository
D. Set up Slurm workload manager to receive jobs that can be scheduled to run on your cloud infrastructure.
Answer: A (LEAVE A REPLY)
A cloud-based backend system is a system that runs on a cloud platform and provides services or resources to other applications or users. A cloud-based backend system can be used to submit training jobs, which are tasks that involve training a machine learning model on a given dataset using a specific framework and configuration1 However, a cloud-based backend system can also have some drawbacks, such as:
High maintenance: A cloud-based backend system may require a lot of administration and management, such as provisioning, scaling, monitoring, and troubleshooting the cloud resources and services. This can be time-consuming and costly, and may distract from the core business objectives2 Low flexibility: A cloud-based backend system may not support all the frameworks and libraries that the data scientists need to use for their training jobs. This can limit the choices and capabilities of the data scientists, and affect the quality and performance of their models3 Poor integration: A cloud-based backend system may not integrate well with other cloud services or tools that the data scientists need to use for their machine learning workflows, such as data processing, model deployment, or model monitoring. This can create compatibility and interoperability issues, and reduce the efficiency and productivity of the data scientists.
Therefore, it may be better to use a managed service instead of a cloud-based backend system to submit training jobs. A managed service is a service that is provided and operated by a third-party provider, and offers various benefits, such as:
Low maintenance: A managed service handles the administration and management of the cloud resources and services, and abstracts away the complexity and details of the underlying infrastructure. This can save time and money, and allow the data scientists to focus on their core tasks2 High flexibility: A managed service can support multiple frameworks and libraries that the data scientists need to use for their training jobs, and allow them to customize and configure their training environments and parameters. This can enhance the choices and capabilities of the data scientists, and improve the quality and performance of their models3 Easy integration: A managed service can integrate seamlessly with other cloud services or tools that the data scientists need to use for their machine learning workflows, and provide a unified and consistent interface and experience. This can solve the compatibility and interoperability issues, and increase the efficiency and productivity of the data scientists.
One of the best options for using a managed service to submit training jobs is to use the AI Platform custom containers feature to receive training jobs using any framework. AI Platform is a Google Cloud service that provides a platform for building, deploying, and managing machine learning models. AI Platform supports various machine learning frameworks, such as TensorFlow, PyTorch, scikit-learn, and XGBoost, and provides various features, such as hyperparameter tuning, distributed training, online prediction, and model monitoring.

The AI Platform custom containers feature allows the data scientists to use any framework or library that they want for their training jobs, and package their training application and dependencies as a Docker container image. The data scientists can then submit their training jobs to AI Platform, and specify the container image and the training parameters. AI Platform will run the training jobs on the cloud infrastructure, and handle the scaling, logging, and monitoring of the training jobs. The data scientists can also use the AI Platform features to optimize, deploy, and manage their models.
The other options are not as suitable or feasible. Configuring Kubeflow to run on Google Kubernetes Engine and receive training jobs through TFJob is not ideal, as Kubeflow is mainly designed for TensorFlow-based training jobs, and does not support other frameworks or libraries. Creating a library of VM images on Compute Engine and publishing these images on a centralized repository is not optimal, as Compute Engine is a low-level service that requires a lot of administration and management, and does not provide the features and integrations of AI Platform. Setting up Slurm workload manager to receive jobs that can be scheduled to run on your cloud infrastructure is not relevant, as Slurm is a tool for managing and scheduling jobs on a cluster of nodes, and does not provide a managed service for training jobs.

NEW QUESTION: 77
You are implementing a batch inference ML pipeline in Google Cloud. The model was developed by using TensorFlow and is stored in SavedModel format in Cloud Storage. You need to apply the model to a historical dataset that is stored in a BigQuery table. You want to perform inference with minimal effort. What should you do?
A. Import the TensorFlow model by using the create model statement in BigQuery ML. Apply the historical data to the TensorFlow model.
B. Export the historical data to Cloud Storage in Avro format. Configure a Vertex AI batch prediction job to generate predictions for the exported data.
C. Export the historical data to Cloud Storage in CSV format. Configure a Vertex AI batch prediction job to generate predictions for the exported data.
D. Configure and deploy a Vertex AI endpoint. Use the endpoint to get predictions from the historical data in BigQuery.

Answer: (SHOW ANSWER)

* Vertex AI batch prediction is the most appropriate and efficient way to apply a pre-trained model like TensorFlow's SavedModel to a large dataset, especially for batch processing.
* The Vertex AI batch prediction job works by exporting your dataset (in this case, historical data from BigQuery) to a suitable format (like Avro or CSV) and then processing it in Cloud Storage where the model is stored.
* Avro format is recommended for large datasets as it is highly efficient for data storage and is optimized for read/write operations in Google Cloud, which is why option B is correct.
* Option A suggests using BigQuery ML for inference, but it does not support running arbitrary TensorFlow models directly within BigQuery ML. Hence, BigQuery ML is not a valid option for this particular task.
* Option C (exporting to CSV) is a valid alternative but is less efficient compared to Avro in terms of performance.

NEW QUESTION: 78

You have created a Vertex AI pipeline that automates custom model training You want to add a pipeline component that enables your team to most easily collaborate when running different executions and comparing metrics both visually and programmatically. What should you do?

A. Add a component to the Vertex AI pipeline that logs metrics to a BigQuery table Query the table to compare different executions of the pipeline Connect BigQuery to Looker Studio to visualize metrics.

B. Add a component to the Vertex AI pipeline that logs metrics to a BigQuery table Load the table into a pandas DataFrame to compare different executions of the pipeline Use Matplotlib to visualize metrics.

C. Add a component to the Vertex AI pipeline that logs metrics to Vertex ML Metadata Use Vertex AI Experiments to compare different executions of the pipeline Use Vertex AI TensorBoard to visualize metrics.

D. Add a component to the Vertex AI pipeline that logs metrics to Vertex ML Metadata Load the Vertex ML Metadata into a pandas DataFrame to compare different executions of the pipeline. Use Matplotlib to visualize metrics.

Answer: C (LEAVE A REPLY)

Vertex AI Experiments is a managed service that allows you to track, compare, and manage experiments with Vertex AI. You can use Vertex AI Experiments to record the parameters, metrics, and artifacts of each pipeline run, and compare them in a graphical interface. Vertex AI TensorBoard is a tool that lets you visualize the metrics of your models, such as accuracy, loss, and learning curves. By logging metrics to Vertex ML Metadata and using Vertex AI Experiments and TensorBoard, you can easily collaborate with your team and find the best model configuration for your problem. Reference: Vertex AI Pipelines: Metrics visualization and run comparison using the KFP SDK, Track, compare, manage experiments with Vertex AI Experiments, Vertex AI Pipelines

NEW QUESTION: 79
You are developing a custom image classification model in Python. You plan to run your training application on Vertex AI Your input dataset contains several hundred thousand small images You need to determine how to store and access the images for training. You want to maximize data throughput and minimize training time while reducing the amount of additional code. What should you do?
A. Store image files in Cloud Storage and access them directly.
B. Store image files in Cloud Storage and access them by using serialized records.
C. Store image files in Cloud Filestore, and access them by using serialized records.
D. Store image files in Cloud Filestore and access them directly by using an NFS mount point.
Answer: B (LEAVE A REPLY)
Cloud Storage is a scalable and cost-effective storage service for any type of data. By storing image files in Cloud Storage, you can access them from anywhere and avoid the overhead of managing your own storage infrastructure. However, accessing image files directly from Cloud Storage can be slow and inefficient, especially for large-scale training. A better option is to use serialized records, such as TFRecord or Apache Avro, which are binary formats that store multiple images and their labels in a single file. Serialized records can improve the data throughput and reduce the network latency, as well as enable data compression and sharding. You can use TensorFlow or Apache Beam APIs to create and read serialized records from Cloud Storage. This solution requires minimal code changes and can speed up your training time significantly. Reference:
Cloud Storage | Google Cloud
TFRecord and tf.Example | TensorFlow Core
Apache Avro 1.10.2 Specification
Using Apache Beam with Cloud Storage | Cloud Storage

NEW QUESTION: 80
You work at a bank You have a custom tabular ML model that was provided by the bank's vendor. The training data is not available due to its sensitivity. The model is packaged as a Vertex AI Model serving container which accepts a string as input for each prediction instance. In each string the feature values are separated by commas. You want to deploy this model to production for online predictions, and monitor the feature distribution over time with minimal effort What should you do?
A. 1 Upload the model to Vertex AI Model Registry and deploy the model to a Vertex Ai endpoint.
2. Create a Vertex AI Model Monitoring job with feature drift detection as the monitoring objective, and provide an instance schema.
B. 1 Upload the model to Vertex AI Model Registry and deploy the model to a Vertex AI endpoint.
2 Create a Vertex AI Model Monitoring job with feature skew detection as the monitoring objective and provide an instance schema.
C. 1 Refactor the serving container to accept key-value pairs as input format.
2. Upload the model to Vertex AI Model Registry and deploy the model to a Vertex AI endpoint.

3. Create a Vertex AI Model Monitoring job with feature drift detection as the monitoring objective.

D. 1 Refactor the serving container to accept key-value pairs as input format.
2 Upload the model to Vertex AI Model Registry and deploy the model to a Vertex AI endpoint.
3. Create a Vertex AI Model Monitoring job with feature skew detection as the monitoring objective.

Answer: A (LEAVE A REPLY)

The best option for deploying a custom tabular ML model to production for online predictions, and monitoring the feature distribution over time with minimal effort, using a model that was provided by the bank's vendor, the training data is not available due to its sensitivity, and the model is packaged as a Vertex AI Model serving container which accepts a string as input for each prediction instance, is to upload the model to Vertex AI Model Registry and deploy the model to a Vertex AI endpoint, create a Vertex AI Model Monitoring job with feature drift detection as the monitoring objective, and provide an instance schema. This option allows you to leverage the power and simplicity of Vertex AI to serve and monitor your model with minimal code and configuration. Vertex AI is a unified platform for building and deploying machine learning solutions on Google Cloud. Vertex AI can deploy a trained model to an online prediction endpoint, which can provide low-latency predictions for individual instances. Vertex AI can also provide various tools and services for data analysis, model development, model deployment, model monitoring, and model governance. A Vertex AI Model Registry is a resource that can store and manage your models on Vertex AI. A Vertex AI Model Registry can help you organize and track your models, and access various model information, such as model name, model description, and model labels. A Vertex AI Model serving container is a resource that can run your custom model code on Vertex AI. A Vertex AI Model serving container can help you package your model code and dependencies into a container image, and deploy the container image to an online prediction endpoint. A Vertex AI Model serving container can accept various input formats, such as JSON, CSV, or TFRecord. A string input format is a type of input format that accepts a string as input for each prediction instance. A string input format can help you encode your feature values into a single string, and separate them by commas. By uploading the model to Vertex AI Model Registry and deploying the model to a Vertex AI endpoint, you can serve your model for online predictions with minimal code and configuration. You can use the Vertex AI API or the gcloud command-line tool to upload the model to Vertex AI Model Registry, and provide the model name, model description, and model labels. You can also use the Vertex AI API or the gcloud command-line tool to deploy the model to a Vertex AI endpoint, and provide the endpoint name, endpoint description, endpoint labels, and endpoint resources. A Vertex AI Model Monitoring job is a resource that can monitor the performance and quality of your deployed models on Vertex AI. A Vertex AI Model Monitoring job can help you detect and diagnose issues with your models, such as data drift, prediction drift, training/serving skew, or model staleness. Feature drift is a type of model monitoring metric that measures the difference between the distributions of the features used to train the model and the features used to serve the model over time. Feature drift can indicate that the online data is changing over time, and the model performance is degrading. By creating a Vertex AI Model Monitoring job with feature drift detection as the monitoring objective,

and providing an instance schema, you can monitor the feature distribution over time with minimal effort. You can use the Vertex AI API or the gcloud command-line tool to create a Vertex AI Model Monitoring job, and provide the monitoring objective, the monitoring frequency, the alerting threshold, and the notification channel. You can also provide an instance schema, which is a JSON file that describes the features and their types in the prediction input data. An instance schema can help Vertex AI Model Monitoring parse and analyze the string input format, and calculate the feature distributions and distance scores1.

The other options are not as good as option A, for the following reasons:

Option B: Uploading the model to Vertex AI Model Registry and deploying the model to a Vertex AI endpoint, creating a Vertex AI Model Monitoring job with feature skew detection as the monitoring objective, and providing an instance schema would not help you monitor the changes in the online data over time, and could cause errors or poor performance. Feature skew is a type of model monitoring metric that measures the difference between the distributions of the features used to train the model and the features used to serve the model at a given point in time. Feature skew can indicate that the model is not trained on the representative data, or that the data is changing over time. By creating a Vertex AI Model Monitoring job with feature skew detection as the monitoring objective, and providing an instance schema, you can monitor the feature distribution at a given point in time with minimal effort. However, uploading the model to Vertex AI Model Registry and deploying the model to a Vertex AI endpoint, creating a Vertex AI Model Monitoring job with feature skew detection as the monitoring objective, and providing an instance schema would not help you monitor the changes in the online data over time, and could cause errors or poor performance. You would need to use the Vertex AI API or the gcloud command-line tool to upload the model to Vertex AI Model Registry, deploy the model to a Vertex AI endpoint, create a Vertex AI Model Monitoring job, and provide an instance schema. Moreover, this option would not monitor the feature drift, which is a more direct and relevant metric for measuring the changes in the online data over time, and the model performance and quality1.

Option C: Refactoring the serving container to accept key-value pairs as input format, uploading the model to Vertex AI Model Registry and deploying the model to a Vertex AI endpoint, creating a Vertex AI Model Monitoring job with feature drift detection as the monitoring objective would require more skills and steps than uploading the model to Vertex AI Model Registry and deploying the model to a Vertex AI endpoint, creating a Vertex AI Model Monitoring job with feature drift detection as the monitoring objective, and providing an instance schema. A key-value pair input format is a type of input format that accepts a key-value pair as input for each prediction instance. A key-value pair input format can help you specify the feature names and values in a JSON object, and separate them by colons. By refactoring the serving container to accept key-value pairs as input format, uploading the model to Vertex AI Model Registry and deploying the model to a Vertex AI endpoint, creating a Vertex AI Model Monitoring job with feature drift detection as the monitoring objective, you can serve and monitor your model with minimal code and configuration. You can write code to refactor the serving container to accept key-value pairs as input format, and use the Vertex AI API or the gcloud command-line tool to upload the model to Vertex AI Model Registry, deploy the model to a Vertex AI endpoint, and create a Vertex AI Model

Monitoring job. However, refactoring the serving container to accept key-value pairs as input format, uploading the model to Vertex AI Model Registry and deploying the model to a Vertex AI endpoint, creating a Vertex AI Model Monitoring job with feature drift detection as the monitoring objective would require more skills and steps than uploading the model to Vertex AI Model Registry and deploying the model to a Vertex AI endpoint, creating a Vertex AI Model Monitoring job with feature drift detection as the monitoring objective, and providing an instance schema. You would need to write code, refactor the serving container, upload the model to Vertex AI Model Registry, deploy the model to a Vertex AI endpoint, and create a Vertex AI Model Monitoring job. Moreover, this option would not use the instance schema, which is a JSON file that can help Vertex AI Model Monitoring parse and analyze the string input format, and calculate the feature distributions and distance scores1.
Option D: Refactoring the serving container to accept key-value pairs as input format, uploading the model to Vertex AI Model Registry and deploying the model to a Vertex AI endpoint, creating a Vertex AI Model Monitoring job with feature skew detection as the monitoring objective would require more skills and steps than uploading the model to Vertex AI Model Registry and deploying the model to a Vertex AI endpoint, creating a Vertex AI Model Monitoring job with feature drift detection as the monitoring objective, and providing an instance schema, and would not help you monitor the changes in the online data over time, and could cause errors or poor performance. Feature skew is a type of model monitoring metric that measures the difference between the distributions of the features used to train the model and the features used to serve the model at a given point in time. Feature skew can indicate that the model is not trained on the representative data, or that the data is changing over time. By creating a Vertex AI Model Monitoring job with feature skew detection as the monitoring objective, you can monitor the feature distribution at a given point in time with minimal effort. However, refactoring the serving container to accept key-value pairs as input format, uploading the model to Vertex AI Model Registry and deploying the model to a Vertex AI endpoint, creating a Vertex AI Model Monitoring job with feature skew detection as the monitoring objective would require more skills and steps than uploading the model to Vertex AI Model Registry and deploying the model to a Vertex AI endpoint, creating a Vertex AI Model Monitoring job with feature drift detection as the monitoring objective, and providing an instance schema, and would not help you monitor the changes in the online data over time, and could cause errors or poor performance. You would need to write code, refactor the serving container, upload the model to Vertex AI Model Registry, deploy the model to a Vertex AI endpoint, and create a Vertex AI Model Monitoring job. Moreover, this option would not monitor the feature drift, which is a more direct and relevant metric for measuring the changes in the online data over time, and the model performance and quality1.
Reference:
Using Model Monitoring | Vertex AI | Google Cloud

NEW QUESTION: 81
You work with a team of researchers to develop state-of-the-art algorithms for financial analysis. Your team develops and debugs complex models in TensorFlow. You want to maintain the ease

of debugging while also reducing the model training time. How should you set up your training environment?

A. Configure a v3-8 TPU VM SSH into the VM to tram and debug the model.

B. Configure a v3-8 TPU node Use Cloud Shell to SSH into the Host VM to train and debug the model.

C. Configure a M-standard-4 VM with 4 NVIDIA P100 GPUs SSH into the VM and use Parameter Server Strategy to train the model.

D. Configure a M-standard-4 VM with 4 NVIDIA P100 GPUs SSH into the VM and use MultiWorkerMirroredStrategy to train the model.

Answer: A (LEAVE A REPLY)

A TPU VM is a virtual machine that has direct access to a Cloud TPU device. TPU VMs provide a simpler and more flexible way to use Cloud TPUs, as they eliminate the need for a separate host VM and network setup. TPU VMs also support interactive debugging tools such as TensorFlow Debugger (tfdbg) and Python Debugger (pdb), which can help researchers develop and troubleshoot complex models. A v3-8 TPU VM has 8 TPU cores, which can provide high performance and scalability for training large models. SSHing into the TPU VM allows the user to run and debug the TensorFlow code directly on the TPU device, without any network overhead or data transfer issues. Reference:

1: TPU VMs Overview

2: TPU VMs Quickstart

3: Debugging TensorFlow Models on Cloud TPUs

NEW QUESTION: 82

You need to develop an image classification model by using a large dataset that contains labeled images in a Cloud Storage Bucket. What should you do?

A. Use Vertex AI Pipelines with the Kubeflow Pipelines SDK to create a pipeline that reads the images from Cloud Storage and trains the model.

B. Use Vertex AI Pipelines with TensorFlow Extended (TFX) to create a pipeline that reads the images from Cloud Storage and trams the model.

C. Import the labeled images as a managed dataset in Vertex AI: and use AutoML to tram the model.

D. Convert the image dataset to a tabular format using Dataflow Load the data into BigQuery and use BigQuery ML to tram the model.

Answer: (SHOW ANSWER)

The best option for developing an image classification model by using a large dataset that contains labeled images in a Cloud Storage bucket is to import the labeled images as a managed dataset in Vertex AI and use AutoML to train the model. This option allows you to leverage the power and simplicity of Google Cloud to create and deploy a high-quality image classification model with minimal code and configuration. Vertex AI is a unified platform for building and deploying machine learning solutions on Google Cloud. Vertex AI can create a managed dataset from a Cloud Storage bucket that contains labeled images, which can be used to train an AutoML

model. AutoML is a service that can automatically build and optimize machine learning models for various tasks, such as image classification, object detection, natural language processing, and tabular data analysis. AutoML can handle the complex aspects of machine learning, such as feature engineering, model architecture, hyperparameter tuning, and model evaluation. AutoML can also evaluate, deploy, and monitor the image classification model, and provide online or batch predictions. By using Vertex AI and AutoML, users can develop an image classification model by using a large dataset with ease and efficiency.
The other options are not as good as option C, for the following reasons:
Option A: Using Vertex AI Pipelines with the Kubeflow Pipelines SDK to create a pipeline that reads the images from Cloud Storage and trains the model would require more skills and steps than using Vertex AI and AutoML. Vertex AI Pipelines is a service that can orchestrate machine learning workflows using Vertex AI. Vertex AI Pipelines can run preprocessing and training steps on custom Docker images, and evaluate, deploy, and monitor the machine learning model. Kubeflow Pipelines SDK is a Python library that can create and run pipelines on Vertex AI Pipelines or on Kubeflow, an open-source platform for machine learning on Kubernetes. However, using Vertex AI Pipelines and Kubeflow Pipelines SDK would require writing code, building Docker images, defining pipeline components and steps, and managing the pipeline execution and artifacts. Moreover, Vertex AI Pipelines and Kubeflow Pipelines SDK are not specialized for image classification, and users would need to use other libraries or frameworks, such as TensorFlow or PyTorch, to build and train the image classification model.
Option B: Using Vertex AI Pipelines with TensorFlow Extended (TFX) to create a pipeline that reads the images from Cloud Storage and trains the model would require more skills and steps than using Vertex AI and AutoML. TensorFlow Extended (TFX) is a framework that can create and run end-to-end machine learning pipelines on TensorFlow, a popular library for building and training deep learning models. TFX can preprocess the data, train and evaluate the model, validate and push the model, and serve the model for online or batch predictions. However, using Vertex AI Pipelines and TFX would require writing code, building Docker images, defining pipeline components and steps, and managing the pipeline execution and artifacts. Moreover, TFX is not optimized for image classification, and users would need to use other libraries or tools, such as TensorFlow Data Validation, TensorFlow Transform, and TensorFlow Hub, to handle the image data and the model architecture.
Option D: Converting the image dataset to a tabular format using Dataflow, loading the data into BigQuery, and using BigQuery ML to train the model would not handle the image data properly and could result in a poor model performance. Dataflow is a service that can create scalable and reliable pipelines to process large volumes of data from various sources. Dataflow can preprocess the data by using Apache Beam, a programming model for defining and executing data processing workflows. BigQuery is a serverless, scalable, and cost-effective data warehouse that can perform fast and interactive queries on large datasets. BigQuery ML is a service that can create and train machine learning models by using SQL queries on BigQuery. However, converting the image data to a tabular format would lose the spatial and semantic information of the images, which are essential for image classification. Moreover, BigQuery ML is not

specialized for image classification, and users would need to use other tools or techniques, such as feature hashing, embedding, or one-hot encoding, to handle the categorical features.

NEW QUESTION: 83
You work for a company that sells corporate electronic products to thousands of businesses worldwide. Your company stores historical customer data in BigQuery. You need to build a model that predicts customer lifetime value over the next three years. You want to use the simplest approach to build the model and you want to have access to visualization tools. What should you do?
A. Create a Vertex AI Workbench notebook to perform exploratory data analysis. Use IPython magics to create a new BigQuery table with input features Use the BigQuery console to run the create model statement Validate the results by using the ml. evaluate and ml. predict statements.
B. Run the create model statement from the BigQuery console to create an AutoML model Validate the results by using the ml. evaluate and ml. predict statements.
C. Create a Vertex AI Workbench notebook to perform exploratory data analysis and create input features Save the features as a CSV file in Cloud Storage Import the CSV file as a new BigQuery table Use the BigQuery console to run the create model statement Validate the results by using the ml. evaluate and ml. predict statements.
D. Create a Vertex AI Workbench notebook to perform exploratory data analysis Use IPython magics to create a new BigQuery table with input features, create the model and validate the results by using the create model, ml. evaluates, and ml. predict statements.
Answer: (SHOW ANSWER)
BigQuery is a service that allows you to store and query large amounts of data in a scalable and cost-effective way. You can use BigQuery to build a model that predicts customer lifetime value over the next three years, by using the create model statement. The create model statement is a SQL command that allows you to create and train an ML model using your data in BigQuery. You can use the create model statement to create an AutoML model, which is a type of model that automatically selects the best features and architecture for your data. By using an AutoML model, you can use the simplest approach to build the model, without writing any code or performing any feature engineering. You can also use the ml.evaluate and ml.predict statements to validate the results of your model. The ml.evaluate statement is a SQL command that allows you to evaluate the performance and quality of your model using various metrics. The ml.predict statement is a SQL command that allows you to make predictions using your model and new data. You can also use the BigQuery console to access visualization tools, such as charts and graphs, to explore and analyze your data and model results. By using the BigQuery console, the create model statement, and the ml.evaluate and ml.predict statements, you can build and validate a model that predicts customer lifetime value over the next three years, and have access to visualization tools.
Reference:
BigQuery documentation
create model statement documentation
ml.evaluate statement documentation

ml.predict statement documentation
Preparing for Google Cloud Certification: Machine Learning Engineer Professional Certificate

NEW QUESTION: 84
You are developing a training pipeline for a new XGBoost classification model based on tabular data The data is stored in a BigQuery table You need to complete the following steps
1. Randomly split the data into training and evaluation datasets in a 65/35 ratio
2. Conduct feature engineering
3 Obtain metrics for the evaluation dataset.
4 Compare models trained in different pipeline executions
How should you execute these steps'?
A. 1 Using Vertex AI Pipelines, add a component to divide the data into training and evaluation sets, and add another component for feature engineering
2. Enable auto logging of metrics in the training component.
3 Compare pipeline runs in Vertex AI Experiments
B. 1 Using Vertex AI Pipelines, add a component to divide the data into training and evaluation sets, and add another component for feature engineering
2 Enable autologging of metrics in the training component
3 Compare models using the artifacts lineage in Vertex ML Metadata
C. 1 In BigQuery ML. use the create model statement with bocstzd_tree_classifier as the model type and use BigQuery to handle the data splits.
2 Use a SQL view to apply feature engineering and train the model using the data in that view
3. Compare the evaluation metrics of the models by using a SQL query with the ml. training_infc statement.
D. 1 In BigQuery ML use the create model statement with boosted_tree_classifier as the model type, and use BigQuery to handle the data splits.
2 Use ml transform to specify the feature engineering transformations, and train the model using the data in the table
Answer: (SHOW ANSWER)
' 3. Compare the evaluation metrics of the models by using a SQL query with the ml. training_info statement.
Explanation:
Vertex AI Pipelines is a service that allows you to create and run scalable and portable ML pipelines on Google Cloud. You can use Vertex AI Pipelines to add a component to divide the data into training and evaluation sets, and add another component for feature engineering. A component is a self-contained piece of code that performs a specific task in the pipeline. You can use the built-in components provided by Vertex AI Pipelines, or create your own custom components. By using Vertex AI Pipelines, you can orchestrate and automate your ML workflow, and track the provenance and lineage of your data and models. You can also enable autologging of metrics in the training component, which is a feature that automatically logs the metrics from your XGBoost model to Vertex AI Experiments. Vertex AI Experiments is a service that allows you

to track, compare, and optimize your ML experiments on Google Cloud. You can use Vertex AI Experiments to monitor the training progress, visualize the metrics, and analyze the results of your model. You can also compare models using the artifacts lineage in Vertex ML Metadata. Vertex ML Metadata is a service that stores and manages the metadata of your ML artifacts, such as datasets, models, metrics, and executions. You can use Vertex ML Metadata to view the artifacts lineage, which is a graph that shows the relationships and dependencies among the artifacts. By using the artifacts lineage, you can compare the performance and quality of different models trained in different pipeline executions, and identify the best model for your use case. By using Vertex AI Pipelines, Vertex AI Experiments, and Vertex ML Metadata, you can execute the steps required for developing a training pipeline for a new XGBoost classification model based on tabular data stored in a BigQuery table. Reference:
Vertex AI Pipelines documentation
Vertex AI Experiments documentation
Vertex ML Metadata documentation
Preparing for Google Cloud Certification: Machine Learning Engineer Professional Certificate

NEW QUESTION: 85
You are the lead ML engineer on a mission-critical project that involves analyzing massive datasets using Apache Spark. You need to establish a robust environment that allows your team to rapidly prototype Spark models using Jupyter notebooks. What is the fastest way to achieve this?
A. Configure a Compute Engine instance with Spark and use Jupyter notebooks.
B. Set up a Dataproc cluster with Spark and use Jupyter notebooks.
C. Set up a Vertex AI Workbench instance with a Spark kernel.
D. Use Colab Enterprise with a Spark kernel.
Answer: (SHOW ANSWER)
Dataproc provides a managed Spark environment and integrates with Jupyter notebooks, ideal for large datasets and rapid prototyping. It reduces setup time compared to manual Spark configurations on Compute Engine or Vertex AI. Colab Enterprise is more suitable for small-scale prototyping rather than extensive Spark-based analysis.

NEW QUESTION: 86
You recently trained a XGBoost model that you plan to deploy to production for online inference Before sending a predict request to your model's binary you need to perform a simple data preprocessing step This step exposes a REST API that accepts requests in your internal VPC Service Controls and returns predictions You want to configure this preprocessing step while minimizing cost and effort What should you do?
A. Store a pickled model in Cloud Storage Build a Flask-based app packages the app in a custom container image, and deploy the model to Vertex AI Endpoints.
B. Build a Flask-based app. package the app and a pickled model in a custom container image, and deploy the model to Vertex AI Endpoints.

C. Build a custom predictor class based on XGBoost Predictor from the Vertex AI SDK. package it and a pickled model in a custom container image based on a Vertex built-in image, and deploy the model to Vertex AI Endpoints.
D. Build a custom predictor class based on XGBoost Predictor from the Vertex AI SDK and package the handler in a custom container image based on a Vertex built-in container image Store a pickled model in Cloud Storage and deploy the model to Vertex AI Endpoints.
Answer: (SHOW ANSWER)
Option A is not the best answer because it requires storing the pickled model in Cloud Storage, which may incur additional cost and latency for loading the model. It also requires building a Flask-based app, which may not be necessary for a simple data preprocessing step.
Option B is not the best answer because it requires building a Flask-based app, which may not be necessary for a simple data preprocessing step. It also requires packaging the app and the pickled model in a custom container image, which may increase the size and complexity of the image.
Option C is not the best answer because it requires packaging the pickled model in a custom container image, which may increase the size and complexity of the image. It also does not leverage the Vertex built-in container image, which may provide some optimizations and integrations for XGBoost models.
Option D is the best answer because it leverages the Vertex built-in container image, which may provide some optimizations and integrations for XGBoost models. It also allows storing the pickled model in Cloud Storage, which may reduce the size and complexity of the image. It also allows building a custom predictor class based on XGBoost Predictor from the Vertex AI SDK, which may simplify the data preprocessing step and the prediction logic.

NEW QUESTION: 87
You work as an ML researcher at an investment bank and are experimenting with the Gemini large language model (LLM). You plan to deploy the model for an internal use case and need full control of the model's underlying infrastructure while minimizing inference time. Which serving configuration should you use for this task?
A. Deploy the model on a Vertex AI endpoint using one-click deployment in Model Garden.
B. Deploy the model on a Google Kubernetes Engine (GKE) cluster manually by creating a custom YAML manifest.
C. Deploy the model on a Vertex AI endpoint manually by creating a custom inference container.
D. Deploy the model on a Google Kubernetes Engine (GKE) cluster using the deployment options in Model Garden.
Answer: (SHOW ANSWER)
Deploying the model on GKE with a custom YAML manifest allows maximum control over infrastructure and latency, aligning with the need for low inference time and internal model use. Vertex AI's one-click deployment (Option A) limits control, and deploying on Vertex AI (Option C) doesn't allow for as much customization as a GKE setup.

NEW QUESTION: 88
You have recently created a proof-of-concept (POC) deep learning model. You are satisfied with the overall architecture, but you need to determine the value for a couple of hyperparameters. You want to perform hyperparameter tuning on Vertex AI to determine both the appropriate embedding dimension for a categorical feature used by your model and the optimal learning rate. You configure the following settings:
For the embedding dimension, you set the type to INTEGER with a minValue of 16 and maxValue of 64.
For the learning rate, you set the type to DOUBLE with a minValue of 10e-05 and maxValue of 10e-02.
You are using the default Bayesian optimization tuning algorithm, and you want to maximize model accuracy. Training time is not a concern. How should you set the hyperparameter scaling for each hyperparameter and the maxParallelTrials?
A. Use UNIT_LINEAR_SCALE for the embedding dimension, UNIT_LOG_SCALE for the learning rate, and a large number of parallel trials.
B. Use UNIT_LINEAR_SCALE for the embedding dimension, UNIT_LOG_SCALE for the learning rate, and a small number of parallel trials.
C. Use UNIT_LOG_SCALE for the embedding dimension, UNIT_LINEAR_SCALE for the learning rate, and a large number of parallel trials.
D. Use UNIT_LOG_SCALE for the embedding dimension, UNIT_LINEAR_SCALE for the learning rate, and a small number of parallel trials.
Answer: (SHOW ANSWER)
The best option for performing hyperparameter tuning on Vertex AI to determine the appropriate embedding dimension and the optimal learning rate is to use UNIT_LINEAR_SCALE for the embedding dimension, UNIT_LOG_SCALE for the learning rate, and a large number of parallel trials. This option has the following advantages:
It matches the appropriate scaling type for each hyperparameter, based on their range and distribution. The embedding dimension is an integer hyperparameter that varies linearly between 16 and 64, so using UNIT_LINEAR_SCALE makes sense. The learning rate is a double hyperparameter that varies exponentially between 10e-05 and 10e-02, so using UNIT_LOG_SCALE is more suitable.
It maximizes the exploration of the hyperparameter space, by using a large number of parallel trials. Since training time is not a concern, using more trials can help find the best combination of hyperparameters that maximizes model accuracy. The default Bayesian optimization tuning algorithm can efficiently sample the hyperparameter space and converge to the optimal values.
The other options are less optimal for the following reasons:
Option B: Using UNIT_LINEAR_SCALE for the embedding dimension, UNIT_LOG_SCALE for the learning rate, and a small number of parallel trials, reduces the exploration of the hyperparameter space, by using a small number of parallel trials. Since training time is not a concern, using fewer trials can miss some potentially good combinations of hyperparameters that

maximize model accuracy. The default Bayesian optimization tuning algorithm can benefit from more trials to sample the hyperparameter space and converge to the optimal values.
Option C: Using UNIT_LOG_SCALE for the embedding dimension, UNIT_LINEAR_SCALE for the learning rate, and a large number of parallel trials, mismatches the appropriate scaling type for each hyperparameter, based on their range and distribution. The embedding dimension is an integer hyperparameter that varies linearly between 16 and 64, so using UNIT_LOG_SCALE is not suitable. The learning rate is a double hyperparameter that varies exponentially between 10e-05 and 10e-02, so using UNIT_LINEAR_SCALE makes less sense.
Option D: Using UNIT_LOG_SCALE for the embedding dimension, UNIT_LINEAR_SCALE for the learning rate, and a small number of parallel trials, combines the drawbacks of option B and option C. It mismatches the appropriate scaling type for each hyperparameter, based on their range and distribution, and reduces the exploration of the hyperparameter space, by using a small number of parallel trials.
Reference:
[Vertex AI: Hyperparameter tuning overview]
[Vertex AI: Configuring the hyperparameter tuning job]

NEW QUESTION: 89
You developed a custom model by using Vertex AI to forecast the sales of your company s products based on historical transactional data You anticipate changes in the feature distributions and the correlations between the features in the near future You also expect to receive a large volume of prediction requests You plan to use Vertex AI Model Monitoring for drift detection and you want to minimize the cost. What should you do?
A. Use the features for monitoring Set a monitoring- frequency value that is higher than the default.
B. Use the features for monitoring Set a prediction-sampling-rare value that is closer to 1 than 0.
C. Use the features and the feature attributions for monitoring. Set a monitoring-frequency value that is lower than the default.
D. Use the features and the feature attributions for monitoring Set a prediction-sampling-rate value that is closer to 0 than 1.
Answer: D (LEAVE A REPLY)
The best option for using Vertex AI Model Monitoring for drift detection and minimizing the cost is to use the features and the feature attributions for monitoring, and set a prediction-sampling-rate value that is closer to 0 than 1. This option allows you to leverage the power and flexibility of Google Cloud to detect feature drift in the input predict requests for custom models, and reduce the storage and computation costs of the model monitoring job. Vertex AI Model Monitoring is a service that can track and compare the results of multiple machine learning runs. Vertex AI Model Monitoring can monitor the model's prediction input data for feature skew and drift. Feature drift occurs when the feature data distribution in production changes over time. If the original training data is not available, you can enable drift detection to monitor your models for feature drift. Vertex AI Model Monitoring uses TensorFlow Data Validation (TFDV) to calculate the distributions and

distance scores for each feature, and compares them with a baseline distribution. The baseline distribution is the statistical distribution of the feature's values in the training data. If the training data is not available, the baseline distribution is calculated from the first 1000 prediction requests that the model receives. If the distance score for a feature exceeds an alerting threshold that you set, Vertex AI Model Monitoring sends you an email alert. However, if you use a custom model, you can also enable feature attribution monitoring, which can provide more insights into the feature drift. Feature attribution monitoring analyzes the feature attributions, which are the contributions of each feature to the prediction output. Feature attribution monitoring can help you identify the features that have the most impact on the model performance, and the features that have the most significant drift over time. Feature attribution monitoring can also help you understand the relationship between the features and the prediction output, and the correlation between the features1. The prediction-sampling-rate is a parameter that determines the percentage of prediction requests that are logged and analyzed by the model monitoring job. Using a lower prediction-sampling-rate can reduce the storage and computation costs of the model monitoring job, but also the quality and validity of the data. Using a lower prediction-sampling-rate can introduce sampling bias and noise into the data, and make the model monitoring job miss some important features or patterns of the data. However, using a higher prediction-sampling-rate can increase the storage and computation costs of the model monitoring job, and also the amount of data that needs to be processed and analyzed. Therefore, there is a trade-off between the prediction-sampling-rate and the cost and accuracy of the model monitoring job, and the optimal prediction-sampling-rate depends on the business objective and the data characteristics2. By using the features and the feature attributions for monitoring, and setting a prediction-sampling-rate value that is closer to 0 than 1, you can use Vertex AI Model Monitoring for drift detection and minimize the cost.

The other options are not as good as option D, for the following reasons:

Option A: Using the features for monitoring and setting a monitoring-frequency value that is higher than the default would not enable feature attribution monitoring, and could increase the cost of the model monitoring job. The monitoring-frequency is a parameter that determines how often the model monitoring job analyzes the logged prediction requests and calculates the distributions and distance scores for each feature. Using a higher monitoring-frequency can increase the frequency and timeliness of the model monitoring job, but also the computation costs of the model monitoring job. Moreover, using the features for monitoring would not enable feature attribution monitoring, which can provide more insights into the feature drift and the model performance1.

Option B: Using the features for monitoring and setting a prediction-sampling-rate value that is closer to 1 than 0 would not enable feature attribution monitoring, and could increase the cost of the model monitoring job. The prediction-sampling-rate is a parameter that determines the percentage of prediction requests that are logged and analyzed by the model monitoring job. Using a higher prediction-sampling-rate can increase the quality and validity of the data, but also the storage and computation costs of the model monitoring job. Moreover, using the features for

monitoring would not enable feature attribution monitoring, which can provide more insights into the feature drift and the model performance12.
Option C: Using the features and the feature attributions for monitoring and setting a monitoring-frequency value that is lower than the default would enable feature attribution monitoring, but could reduce the frequency and timeliness of the model monitoring job. The monitoring-frequency is a parameter that determines how often the model monitoring job analyzes the logged prediction requests and calculates the distributions and distance scores for each feature. Using a lower monitoring-frequency can reduce the computation costs of the model monitoring job, but also the frequency and timeliness of the model monitoring job. This can make the model monitoring job less responsive and effective in detecting and alerting the feature drift1.
Reference:
Preparing for Google Cloud Certification: Machine Learning Engineer, Course 3: Production ML Systems, Week 4: Evaluation Google Cloud Professional Machine Learning Engineer Exam Guide, Section 3: Scaling ML models in production, 3.3 Monitoring ML models in production Official Google Cloud Certified Professional Machine Learning Engineer Study Guide, Chapter 6: Production ML Systems, Section 6.3: Monitoring ML Models Using Model Monitoring
Understanding the score threshold slider

NEW QUESTION: 90
You are training a deep learning model for semantic image segmentation with reduced training time. While using a Deep Learning VM Image, you receive the following error: The resource 'projects/deeplearning-platforn/zones/europe-west4-c/acceleratorTypes/nvidia-tesla-k80' was not found. What should you do?
A. Ensure that you have GPU quota in the selected region.
B. Ensure that the required GPU is available in the selected region.
C. Ensure that you have preemptible GPU quota in the selected region.
D. Ensure that the selected GPU has enough GPU memory for the workload.
Answer: B (LEAVE A REPLY)
The error message indicates that the selected GPU type (nvidia-tesla-k80) is not available in the selected region (europe-west4-c). This can happen when the GPU type is not supported in the region, or when the GPU quota is exhausted in the region. To avoid this error, you should ensure that the required GPU is available in the selected region before creating a Deep Learning VM Image. You can use the following steps to check the GPU availability and quota:
To check the GPU availability, you can use the gcloud compute accelerator-types list command with the --filter flag to specify the GPU type and the region. For example, to check the availability of nvidia-tesla-k80 in europe-west4-c, you can run:
gcloud compute accelerator-types list --filter"namenvidia-tesla-k80 AND zone:europe-west4-c" If the command returns an empty result, it means that the GPU type is not supported in the region. You can either choose a different GPU type or a different region that supports the GPU type. You can use the same command without the --filter flag to list all the available GPU types and regions. For example, to list all the available GPU types in europe-west4-c, you can run:

gcloud compute accelerator-types list --filter"zone:europe-west4-c"
To check the GPU quota, you can use the gcloud compute regions describe command with the --format flag to specify the region and the quota metric. For example, to check the quota for nvidia-tesla-k80 in europe-west4-c, you can run:
gcloud compute regions describe europe-west4-c --format"value(quotas.NVIDIA_K80_GPUS)" If the command returns a value of 0, it means that the GPU quota is exhausted in the region. You can either request more quota from Google Cloud or choose a different region that has enough quota for the GPU type.
Reference:
Troubleshooting | Deep Learning VM Images | Google Cloud
Checking GPU availability
Checking GPU quota

NEW QUESTION: 91
You work for a food product company. Your company's historical sales data is stored in BigQuery You need to use Vertex AI's custom training service to train multiple TensorFlow models that read the data from BigQuery and predict future sales You plan to implement a data preprocessing algorithm that performs min-max scaling and bucketing on a large number of features before you start experimenting with the models. You want to minimize preprocessing time, cost and development effort How should you configure this workflow?
A. Write the transformations into Spark that uses the spark-bigquery-connector and use Dataproc to preprocess the data.
B. Write SQL queries to transform the data in-place in BigQuery.
C. Add the transformations as a preprocessing layer in the TensorFlow models.
D. Create a Dataflow pipeline that uses the BigQueryIO connector to ingest the data process it and write it back to BigQuery.
Answer: (SHOW ANSWER)
The best option for configuring the workflow is to add the transformations as a preprocessing layer in the TensorFlow models. This option allows you to leverage the power and simplicity of TensorFlow to preprocess and transform the data with simple Python code. TensorFlow is a framework for building and training machine learning models. TensorFlow provides various tools and libraries for data analysis and machine learning. A preprocessing layer is a type of layer in TensorFlow that can perform data preprocessing and feature engineering operations on the input data. A preprocessing layer can help you customize the data transformation and preprocessing logic, and handle complex or non-standard data formats. A preprocessing layer can also help you minimize the preprocessing time, cost, and development effort, as you only need to write a few lines of code to implement the preprocessing layer, and you do not need to create any intermediate data sources or pipelines. By adding the transformations as a preprocessing layer in the TensorFlow models, you can use Vertex AI's custom training service to train multiple TensorFlow models that read the data from BigQuery and predict future sales1.
The other options are not as good as option C, for the following reasons:

Option A: Writing the transformations into Spark that uses the spark-bigquery-connector and using Dataproc to preprocess the data would require more skills and steps than using a preprocessing layer in TensorFlow. Spark is a framework for distributed data processing and machine learning. Spark can read and write data from BigQuery by using the spark-bigquery-connector, which is a library that allows Spark to communicate with BigQuery. Dataproc is a service that can create and manage Spark clusters on Google Cloud. Dataproc can help you run Spark jobs on Google Cloud, and scale the clusters according to the workload. However, writing the transformations into Spark that uses the spark-bigquery-connector and using Dataproc to preprocess the data would require more skills and steps than using a preprocessing layer in TensorFlow. You would need to write code, create and configure the Spark cluster, install and import the spark-bigquery-connector, load and preprocess the data, and write the data back to BigQuery. Moreover, this option would create an intermediate data source in BigQuery, which can increase the storage and computation costs2.
Option B: Writing SQL queries to transform the data in-place in BigQuery would not allow you to use Vertex AI's custom training service to train multiple TensorFlow models that read the data from BigQuery and predict future sales. BigQuery is a service that can perform data analysis and machine learning by using SQL queries. BigQuery can perform data transformation and preprocessing by using SQL functions and clauses, such as MIN, MAX, CASE, and TRANSFORM. BigQuery can also perform machine learning by using BigQuery ML, which is a feature that can create and train machine learning models by using SQL queries. However, writing SQL queries to transform the data in-place in BigQuery would not allow you to use Vertex AI's custom training service to train multiple TensorFlow models that read the data from BigQuery and predict future sales. Vertex AI's custom training service is a service that can run your custom machine learning code on Vertex AI. Vertex AI's custom training service can support various machine learning frameworks, such as TensorFlow, PyTorch, and scikit-learn. Vertex AI's custom training service cannot support SQL queries, as SQL is not a machine learning framework. Therefore, if you want to use Vertex AI's custom training service, you cannot use SQL queries to transform the data in-place in BigQuery3.
Option D: Creating a Dataflow pipeline that uses the BigQueryIO connector to ingest the data, process it, and write it back to BigQuery would require more skills and steps than using a preprocessing layer in TensorFlow. Dataflow is a service that can create and run data processing and machine learning pipelines on Google Cloud. Dataflow can read and write data from BigQuery by using the BigQueryIO connector, which is a library that allows Dataflow to communicate with BigQuery. Dataflow can perform data transformation and preprocessing by using Apache Beam, which is a framework for distributed data processing and machine learning. However, creating a Dataflow pipeline that uses the BigQueryIO connector to ingest the data, process it, and write it back to BigQuery would require more skills and steps than using a preprocessing layer in TensorFlow. You would need to write code, create and configure the Dataflow pipeline, install and import the BigQueryIO connector, load and preprocess the data, and write the data back to BigQuery. Moreover, this option would create an intermediate data source in BigQuery, which can increase the storage and computation costs4.

Reference:
Preparing for Google Cloud Certification: Machine Learning Engineer, Course 3: Production ML Systems, Week 2: Serving ML Predictions Google Cloud Professional Machine Learning Engineer Exam Guide, Section 2: Developing ML models, 2.1 Developing ML models by using TensorFlow Official Google Cloud Certified Professional Machine Learning Engineer Study Guide, Chapter 4: Developing ML Models, Section 4.1: Developing ML Models by Using TensorFlow TensorFlow Preprocessing Layers Spark and BigQuery Dataproc BigQuery ML Dataflow and BigQuery Apache Beam

NEW QUESTION: 92
You are an ML engineer at a travel company. You have been researching customers' travel behavior for many years, and you have deployed models that predict customers' vacation patterns. You have observed that customers' vacation destinations vary based on seasonality and holidays; however, these seasonal variations are similar across years. You want to quickly and easily store and compare the model versions and performance statistics across years. What should you do?
A. Store the performance statistics in Cloud SQL. Query that database to compare the performance statistics across the model versions.
B. Create versions of your models for each season per year in Vertex AI. Compare the performance statistics across the models in the Evaluate tab of the Vertex AI UI.
C. Store the performance statistics of each pipeline run in Kubeflow under an experiment for each season per year. Compare the results across the experiments in the Kubeflow UI.
D. Store the performance statistics of each version of your models using seasons and years as events in Vertex ML Metadata. Compare the results across the slices.
Answer: D (LEAVE A REPLY)
Option A is incorrect because Cloud SQL is a relational database service that is not designed for storing and comparing model performance statistics. It would require writing complex SQL queries to perform the comparison, and it would not provide any visualization or analysis tools.
Option B is incorrect because Vertex AI does not support creating versions of models for each season per year. Vertex AI models are versioned based on the training data and

hyperparameters, not on external factors such as seasonality or holidays. Moreover, the Evaluate tab of the Vertex AI UI only shows the performance metrics of a single model version, not across multiple versions.
Option C is incorrect because Kubeflow is a different platform than Vertex AI, and it does not integrate well with Vertex AI Pipelines. Kubeflow experiments are used to group pipeline runs that share a common goal or objective, not to compare performance statistics across different seasons or years. Kubeflow UI does not provide any tools to compare the results across the experiments, and it would require switching between different platforms to access the data.
Option D is correct because Vertex ML Metadata is a service that allows storing and tracking metadata associated with machine learning workflows, such as models, datasets, metrics, and events. Events are user-defined labels that can be used to group or slice the metadata for analysis. By using seasons and years as events, you can easily store and compare the performance statistics of each version of your models across different time periods. Vertex ML Metadata also provides tools to visualize and analyze the metadata, such as the ML Metadata Explorer and the What-If Tool.

NEW QUESTION: 93
You want to rebuild your ML pipeline for structured data on Google Cloud. You are using PySpark to conduct data transformations at scale, but your pipelines are taking over 12 hours to run. To speed up development and pipeline run time, you want to use a serverless tool and SQL syntax. You have already moved your raw data into Cloud Storage. How should you build the pipeline on Google Cloud while meeting the speed and processing requirements?
A. Use Data Fusion's GUI to build the transformation pipelines, and then write the data into BigQuery
B. Convert your PySpark into SparkSQL queries to transform the data and then run your pipeline on Dataproc to write the data into BigQuery.
C. Ingest your data into Cloud SQL convert your PySpark commands into SQL queries to transform the data, and then use federated queries from BigQuery for machine learning
D. Ingest your data into BigQuery using BigQuery Load, convert your PySpark commands into BigQuery SQL queries to transform the data, and then write the transformations to a new table
Answer: (SHOW ANSWER)
BigQuery is a serverless, scalable, and cost-effective data warehouse that allows users to run SQL queries on large volumes of data. BigQuery Load is a tool that can ingest data from Cloud Storage into BigQuery tables. BigQuery SQL is a dialect of SQL that supports many of the same functions and operations as PySpark, such as window functions, aggregate functions, joins, and subqueries. By using BigQuery Load and BigQuery SQL, you can rebuild your ML pipeline for structured data on Google Cloud without having to manage any servers or clusters, and with faster performance and lower cost than using PySpark on Dataproc. You can also use BigQuery ML to create and evaluate ML models using SQL commands. Reference:
BigQuery documentation
BigQuery Load documentation

BigQuery SQL reference
BigQuery ML documentation

NEW QUESTION: 94
You are investigating the root cause of a misclassification error made by one of your models. You used Vertex AI Pipelines to tram and deploy the model. The pipeline reads data from BigQuery. creates a copy of the data in Cloud Storage in TFRecord format trains the model in Vertex AI Training on that copy, and deploys the model to a Vertex AI endpoint. You have identified the specific version of that model that misclassified: and you need to recover the data this model was trained on. How should you find that copy of the data'?
A. Use Vertex AI Feature Store Modify the pipeline to use the feature store; and ensure that all training data is stored in it Search the feature store for the data used for the training.
B. Use the lineage feature of Vertex AI Metadata to find the model artifact Determine the version of the model and identify the step that creates the data copy, and search in the metadata for its location.
C. Use the logging features in the Vertex AI endpoint to determine the timestamp of the models deployment Find the pipeline run at that timestamp Identify the step that creates the data copy; and search in the logs for its location.
D. Find the job ID in Vertex AI Training corresponding to the training for the model Search in the logs of that job for the data used for the training.
Answer: (SHOW ANSWER)
Option A is not the best answer because it requires modifying the pipeline to use the Vertex AI Feature Store, which may not be feasible or necessary for recovering the data that the model was trained on. The Vertex AI Feature Store is a service that helps you manage, store, and serve feature values for your machine learning models1, but it is not designed for storing the raw data or the TFRecord files.
Option B is the best answer because it leverages the lineage feature of Vertex AI Metadata, which is a service that helps you track and manage the metadata of your machine learning workflows, such as datasets, models, metrics, and parameters2. The lineage feature allows you to view the relationships and dependencies among the artifacts and executions in your pipeline, and trace back the origin and history of any artifact3. By using the lineage feature, you can find the model artifact, determine the version of the model, identify the step that creates the data copy, and search in the metadata for its location.
Option C is not the best answer because it relies on the logging features in the Vertex AI endpoint, which may not be accurate or reliable for finding the data copy. The logging features in the Vertex AI endpoint help you monitor and troubleshoot the online predictions made by your deployed models, but they do not provide information about the training data or the pipeline steps4. Moreover, the timestamp of the model deployment may not match the timestamp of the pipeline run, as there may be delays or errors in the deployment process.
Option D is not the best answer because it requires finding the job ID in Vertex AI Training, which may not be easy or straightforward. Vertex AI Training is a service that helps you train your

custom models on Google Cloud, but it does not provide a direct way to link the training job to the model version or the pipeline run. Moreover, searching in the logs of the job may not reveal the location of the data copy, as the logs may only contain information about the training process and the metrics.
Reference:
1: Introduction to Vertex AI Feature Store | Vertex AI | Google Cloud
2: Introduction to Vertex AI Metadata | Vertex AI | Google Cloud
3: View lineage for ML workflows | Vertex AI | Google Cloud
4: Monitor online predictions | Vertex AI | Google Cloud
[5]: Train custom models | Vertex AI | Google Cloud

NEW QUESTION: 95
You want to train an AutoML model to predict house prices by using a small public dataset stored in BigQuery. You need to prepare the data and want to use the simplest most efficient approach. What should you do?
A. Write a query that preprocesses the data by using BigQuery and creates a new table Create a Vertex AI managed dataset with the new table as the data source.
B. Use Dataflow to preprocess the data Write the output in TFRecord format to a Cloud Storage bucket.
C. Write a query that preprocesses the data by using BigQuery Export the query results as CSV files and use those files to create a Vertex AI managed dataset.
D. Use a Vertex AI Workbench notebook instance to preprocess the data by using the pandas library Export the data as CSV files, and use those files to create a Vertex AI managed dataset.
Answer: (SHOW ANSWER)
The simplest and most efficient approach for preparing the data for AutoML is to use BigQuery and Vertex AI. BigQuery is a serverless, scalable, and cost-effective data warehouse that can perform fast and interactive queries on large datasets. BigQuery can preprocess the data by using SQL functions such as filtering, aggregating, joining, transforming, and creating new features. The preprocessed data can be stored in a new table in BigQuery, which can be used as the data source for Vertex AI. Vertex AI is a unified platform for building and deploying machine learning solutions on Google Cloud. Vertex AI can create a managed dataset from a BigQuery table, which can be used to train an AutoML model. Vertex AI can also evaluate, deploy, and monitor the AutoML model, and provide online or batch predictions. By using BigQuery and Vertex AI, users can leverage the power and simplicity of Google Cloud to train an AutoML model to predict house prices.
The other options are not as simple or efficient as option A, for the following reasons:
Option B: Using Dataflow to preprocess the data and write the output in TFRecord format to a Cloud Storage bucket would require more steps and resources than using BigQuery and Vertex AI. Dataflow is a service that can create scalable and reliable pipelines to process large volumes of data from various sources. Dataflow can preprocess the data by using Apache Beam, a programming model for defining and executing data processing workflows. TFRecord is a binary

file format that can store sequential data efficiently. However, using Dataflow and TFRecord would require writing code, setting up a pipeline, choosing a runner, and managing the output files. Moreover, TFRecord is not a supported format for Vertex AI managed datasets, so the data would need to be converted to CSV or JSONL files before creating a Vertex AI managed dataset.
Option C: Writing a query that preprocesses the data by using BigQuery and exporting the query results as CSV files would require more steps and storage than using BigQuery and Vertex AI. CSV is a text file format that can store tabular data in a comma-separated format. Exporting the query results as CSV files would require choosing a destination Cloud Storage bucket, specifying a file name or a wildcard, and setting the export options. Moreover, CSV files can have limitations such as size, schema, and encoding, which can affect the quality and validity of the data. Exporting the data as CSV files would also incur additional storage costs and reduce the performance of the queries.
Option D: Using a Vertex AI Workbench notebook instance to preprocess the data by using the pandas library and exporting the data as CSV files would require more steps and skills than using BigQuery and Vertex AI. Vertex AI Workbench is a service that provides an integrated development environment for data science and machine learning. Vertex AI Workbench allows users to create and run Jupyter notebooks on Google Cloud, and access various tools and libraries for data analysis and machine learning. Pandas is a popular Python library that can manipulate and analyze data in a tabular format. However, using Vertex AI Workbench and pandas would require creating a notebook instance, writing Python code, installing and importing pandas, connecting to BigQuery, loading and preprocessing the data, and exporting the data as CSV files. Moreover, pandas can have limitations such as memory usage, scalability, and compatibility, which can affect the efficiency and reliability of the data processing.
Reference:
Preparing for Google Cloud Certification: Machine Learning Engineer, Course 2: Data Engineering for ML on Google Cloud, Week 1: Introduction to Data Engineering for ML Google Cloud Professional Machine Learning Engineer Exam Guide, Section 1: Architecting low-code ML solutions, 1.3 Training models by using AutoML Official Google Cloud Certified Professional Machine Learning Engineer Study Guide, Chapter 4: Low-code ML Solutions, Section 4.3: AutoML BigQuery Vertex AI Dataflow TFRecord CSV Vertex AI Workbench Pandas

NEW QUESTION: 96
You have written unit tests for a Kubeflow Pipeline that require custom libraries. You want to automate the execution of unit tests with each new push to your development branch in Cloud Source Repositories. What should you do?
A. Write a script that sequentially performs the push to your development branch and executes the unit tests on Cloud Run
B. Using Cloud Build, set an automated trigger to execute the unit tests when changes are pushed to your development branch.

C. Set up a Cloud Logging sink to a Pub/Sub topic that captures interactions with Cloud Source Repositories Configure a Pub/Sub trigger for Cloud Run, and execute the unit tests on Cloud Run.
D. Set up a Cloud Logging sink to a Pub/Sub topic that captures interactions with Cloud Source Repositories. Execute the unit tests using a Cloud Function that is triggered when messages are sent to the Pub/Sub topic
Answer: B (LEAVE A REPLY)
Cloud Build is a service that executes your builds on Google Cloud Platform infrastructure. Cloud Build can import source code from Cloud Source Repositories, Cloud Storage, GitHub, or Bitbucket, execute a build to your specifications, and produce artifacts such as Docker containers or Java archives1 Cloud Build allows you to set up automated triggers that start a build when changes are pushed to a source code repository. You can configure triggers to filter the changes based on the branch, tag, or file path2 To automate the execution of unit tests for a Kubeflow Pipeline that require custom libraries, you can use Cloud Build to set an automated trigger to execute the unit tests when changes are pushed to your development branch in Cloud Source Repositories. You can specify the steps of the build in a YAML or JSON file, such as installing the custom libraries, running the unit tests, and reporting the results. You can also use Cloud Build to build and deploy the Kubeflow Pipeline components if the unit tests pass3 The other options are not recommended or feasible. Writing a script that sequentially performs the push to your development branch and executes the unit tests on Cloud Run is not a good practice, as it does not leverage the benefits of Cloud Build and its integration with Cloud Source Repositories. Setting up a Cloud Logging sink to a Pub/Sub topic that captures interactions with Cloud Source Repositories and using a Pub/Sub trigger for Cloud Run or Cloud Function to execute the unit tests is unnecessarily complex and inefficient, as it adds extra steps and latency to the process. Cloud Run and Cloud Function are also not designed for executing unit tests, as they have limitations on the memory, CPU, and execution time45

NEW QUESTION: 97
You work at a leading healthcare firm developing state-of-the-art algorithms for various use cases You have unstructured textual data with custom labels You need to extract and classify various medical phrases with these labels What should you do?
A. Use the Healthcare Natural Language API to extract medical entities.
B. Use a BERT-based model to fine-tune a medical entity extraction model.
C. Use AutoML Entity Extraction to train a medical entity extraction model.
D. Use TensorFlow to build a custom medical entity extraction model.
Answer: B (LEAVE A REPLY)
Medical entity extraction is a task that involves identifying and classifying medical terms or concepts from unstructured textual data, such as electronic health records, clinical notes, or research papers. Medical entity extraction can help with various use cases, such as information retrieval, knowledge discovery, decision support, and data analysis1.

One possible approach to perform medical entity extraction is to use a BERT-based model to fine-tune a medical entity extraction model. BERT (Bidirectional Encoder Representations from Transformers) is a pre-trained language model that can capture the contextual information from both left and right sides of a given token2. BERT can be fine-tuned on a specific downstream task, such as medical entity extraction, by adding a task-specific layer on top of the pre-trained model and updating the model parameters with a small amount of labeled data3.
A BERT-based model can achieve high performance on medical entity extraction by leveraging the large-scale pre-training on general-domain corpora and the fine-tuning on domain-specific data. For example, Nesterov and Umerenkov4 proposed a novel method of doing medical entity extraction from electronic health records as a single-step multi-label classification task by fine-tuning a transformer model pre-trained on a large EHR dataset. They showed that their model can achieve human-level quality for most frequent entities.
Reference:
1: Medical Named Entity Recognition from Un-labelled Medical Records based on Pre-trained Language Models and Domain Dictionary | Data Intelligence | MIT Press
2: BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding
3: Fine-tuning BERT for Medical Entity Extraction
4: Distantly supervised end-to-end medical entity extraction from electronic health records with human-level quality

NEW QUESTION: 98
You have trained an XGBoost model that you plan to deploy on Vertex AI for online prediction. You are now uploading your model to Vertex AI Model Registry, and you need to configure the explanation method that will serve online prediction requests to be returned with minimal latency. You also want to be alerted when feature attributions of the model meaningfully change over time. What should you do?
A. 1 Specify sampled Shapley as the explanation method with a path count of 5.
2 Deploy the model to Vertex AI Endpoints.
3. Create a Model Monitoring job that uses prediction drift as the monitoring objective.
B. 1 Specify Integrated Gradients as the explanation method with a path count of 5.
2 Deploy the model to Vertex AI Endpoints.
3. Create a Model Monitoring job that uses prediction drift as the monitoring objective.
C. 1. Specify sampled Shapley as the explanation method with a path count of 50.
2. Deploy the model to Vertex AI Endpoints.
3. Create a Model Monitoring job that uses training-serving skew as the monitoring objective.
D. 1 Specify Integrated Gradients as the explanation method with a path count of 50.
2. Deploy the model to Vertex AI Endpoints.
3 Create a Model Monitoring job that uses training-serving skew as the monitoring objective.
Answer: A (LEAVE A REPLY)
Sampled Shapley is a fast and scalable approximation of the Shapley value, which is a game-theoretic concept that measures the contribution of each feature to the model prediction. Sampled

Shapley is suitable for online prediction requests, as it can return feature attributions with minimal latency. The path count parameter controls the number of samples used to estimate the Shapley value, and a lower value means faster computation. Integrated Gradients is another explanation method that computes the average gradient along the path from a baseline input to the actual input. Integrated Gradients is more accurate than Sampled Shapley, but also more computationally intensive. Therefore, it is not recommended for online prediction requests, especially with a high path count. Prediction drift is the change in the distribution of feature values or labels over time. It can affect the performance and accuracy of the model, and may require retraining or redeploying the model. Vertex AI Model Monitoring allows you to monitor prediction drift on your deployed models and endpoints, and set up alerts and notifications when the drift exceeds a certain threshold. You can specify an email address to receive the notifications, and use the information to retrigger the training pipeline and deploy an updated version of your model. This is the most direct and convenient way to achieve your goal. Training-serving skew is the difference between the data used for training the model and the data used for serving the model. It can also affect the performance and accuracy of the model, and may indicate data quality issues or model staleness. Vertex AI Model Monitoring allows you to monitor training-serving skew on your deployed models and endpoints, and set up alerts and notifications when the skew exceeds a certain threshold. However, this is not relevant to the question, as the question is about the feature attributions of the model, not the data distribution. Reference:
Vertex AI: Explanation methods
Vertex AI: Configuring explanations
Vertex AI: Monitoring prediction drift
Vertex AI: Monitoring training-serving skew

NEW QUESTION: 99
As the lead ML Engineer for your company, you are responsible for building ML models to digitize scanned customer forms. You have developed a TensorFlow model that converts the scanned images into text and stores them in Cloud Storage. You need to use your ML model on the aggregated data collected at the end of each day with minimal manual intervention. What should you do?
A. Use the batch prediction functionality of AI Platform
B. Create a serving pipeline in Compute Engine for prediction
C. Use Cloud Functions for prediction each time a new data point is ingested
D. Deploy the model on AI Platform and create a version of it for online inference.
Answer: A (LEAVE A REPLY)
Batch prediction is the process of using an ML model to make predictions on a large set of data points. Batch prediction is suitable for scenarios where the predictions are not time-sensitive and can be done in batches, such as digitizing scanned customer forms at the end of each day. Batch prediction can also handle large volumes of data and scale up or down the resources as needed. AI Platform provides a batch prediction service that allows users to submit a job with their TensorFlow model and input data stored in Cloud Storage, and receive the output predictions in

Cloud Storage as well. This service requires minimal manual intervention and can be automated with Cloud Scheduler or Cloud Functions. Therefore, using the batch prediction functionality of AI Platform is the best option for this use case.
Reference:
Batch prediction overview
Using batch prediction

NEW QUESTION: 100
You work for a large hotel chain and have been asked to assist the marketing team in gathering predictions for a targeted marketing strategy. You need to make predictions about user lifetime value (LTV) over the next 30 days so that marketing can be adjusted accordingly. The customer dataset is in BigQuery, and you are preparing the tabular data for training with AutoML Tables. This data has a time signal that is spread across multiple columns. How should you ensure that AutoML fits the best model to your data?
A. Manually combine all columns that contain a time signal into an array Allow AutoML to interpret this array appropriately Choose an automatic data split across the training, validation, and testing sets
B. Submit the data for training without performing any manual transformations Allow AutoML to handle the appropriate transformations Choose an automatic data split across the training, validation, and testing sets
C. Submit the data for training without performing any manual transformations, and indicate an appropriate column as the Time column Allow AutoML to split your data based on the time signal provided, and reserve the more recent data for the validation and testing sets
D. Submit the data for training without performing any manual transformations Use the columns that have a time signal to manually split your data Ensure that the data in your validation set is from 30 days after the data in your training set and that the data in your testing set is from 30 days after your validation set
Answer: C (LEAVE A REPLY)
This answer is correct because it allows AutoML Tables to handle the time signal in the data and split the data accordingly. This ensures that the model is trained on the historical data and evaluated on the more recent data, which is consistent with the prediction task. AutoML Tables can automatically detect and handle temporal features in the data, such as date, time, and duration. By specifying the Time column, AutoML Tables can also perform time-series forecasting and use the time signal to generate additional features, such as seasonality and trend.
Reference:
[AutoML Tables: Preparing your training data]
[AutoML Tables: Time-series forecasting]

NEW QUESTION: 101
You work for an online retailer. Your company has a few thousand short lifecycle products. Your company has five years of sales data stored in BigQuery. You have been asked to build a model

that will make monthly sales predictions for each product. You want to use a solution that can be implemented quickly with minimal effort. What should you do?

A. Use Prophet on Vertex AI Training to build a custom model.

B. Use Vertex AI Forecast to build a NN-based model.

C. Use BigQuery ML to build a statistical AR1MA_PLUS model.

D. Use TensorFlow on Vertex AI Training to build a custom model.

Answer: C (LEAVE A REPLY)

According to the web search results, BigQuery ML1 is a service that allows you to create and execute machine learning models in BigQuery using SQL queries. BigQuery ML supports various types of models, such as linear regression, logistic regression, k-means clustering, matrix factorization, deep neural networks, and time series forecasting1. ARIMA_PLUS2 is a statistical model for time series forecasting that is built in to BigQuery ML. ARIMA_PLUS stands for AutoRegressive Integrated Moving Average with eXogenous regressors. ARIMA_PLUS models the relationship between a target variable and its past values, as well as other external factors that might influence the target variable. ARIMA_PLUS can handle multiple time series, seasonality, holidays, and missing values2. Therefore, option C is the best way to use a solution that can be implemented quickly with minimal effort for the given use case, as it allows you to use SQL queries to build and run a forecasting model in BigQuery without moving the data or writing custom code. The other options are not relevant or optimal for this scenario. Reference:
BigQuery ML
ARIMA_PLUS
Google Professional Machine Learning Certification Exam 2023
Latest Google Professional Machine Learning Engineer Actual Free Exam Questions

NEW QUESTION: 102

You work at a large organization that recently decided to move their ML and data workloads to Google Cloud. The data engineering team has exported the structured data to a Cloud Storage bucket in Avro format. You need to propose a workflow that performs analytics, creates features, and hosts the features that your ML models use for online prediction How should you configure the pipeline?

A. Ingest the Avro files into Cloud Spanner to perform analytics Use a Dataflow pipeline to create the features and store them in BigQuery for online prediction.

B. Ingest the Avro files into BigQuery to perform analytics Use a Dataflow pipeline to create the features, and store them in Vertex AI Feature Store for online prediction.

C. Ingest the Avro files into BigQuery to perform analytics Use BigQuery SQL to create features and store them in a separate BigQuery table for online prediction.

D. Ingest the Avro files into Cloud Spanner to perform analytics. Use a Dataflow pipeline to create the features. and store them in Vertex AI Feature Store for online prediction.

Answer: B (LEAVE A REPLY)

BigQuery is a service that allows you to store and query large amounts of data in a scalable and cost-effective way. You can use BigQuery to ingest the Avro files from the Cloud Storage bucket

and perform analytics on the structured data. Avro is a binary file format that can store complex data types and schemas. You can use the bq load command or the BigQuery API to load the Avro files into a BigQuery table. You can then use SQL queries to analyze the data and generate insights. Dataflow is a service that allows you to create and run scalable and portable data processing pipelines on Google Cloud. You can use Dataflow to create the features for your ML models, such as transforming, aggregating, and encoding the data. You can use the Apache Beam SDK to write your Dataflow pipeline code in Python or Java. You can also use the built-in transforms or custom transforms to apply the feature engineering logic to your data. Vertex AI Feature Store is a service that allows you to store and manage your ML features on Google Cloud. You can use Vertex AI Feature Store to host the features that your ML models use for online prediction. Online prediction is a type of prediction that provides low-latency responses to individual or small batches of input data. You can use the Vertex AI Feature Store API to write the features from your Dataflow pipeline to a feature store entity type. You can then use the Vertex AI Feature Store online serving API to read the features from the feature store and pass them to your ML models for online prediction. By using BigQuery, Dataflow, and Vertex AI Feature Store, you can configure a pipeline that performs analytics, creates features, and hosts the features that your ML models use for online prediction. Reference:
BigQuery documentation
Dataflow documentation
Vertex AI Feature Store documentation
Preparing for Google Cloud Certification: Machine Learning Engineer Professional Certificate

NEW QUESTION: 103
You are building a real-time prediction engine that streams files which may contain Personally Identifiable Information (PII) to Google Cloud. You want to use the Cloud Data Loss Prevention (DLP) API to scan the files. How should you ensure that the PII is not accessible by unauthorized individuals?
A. Stream all files to Google CloudT and then write the data to BigQuery Periodically conduct a bulk scan of the table using the DLP API.
B. Stream all files to Google Cloud, and write batches of the data to BigQuery While the data is being written to BigQuery conduct a bulk scan of the data using the DLP API.
C. Create two buckets of data Sensitive and Non-sensitive Write all data to the Non-sensitive bucket Periodically conduct a bulk scan of that bucket using the DLP API, and move the sensitive data to the Sensitive bucket
D. Create three buckets of data: Quarantine, Sensitive, and Non-sensitive Write all data to the Quarantine bucket.
E. Periodically conduct a bulk scan of that bucket using the DLP API, and move the data to either the Sensitive or Non-Sensitive bucket
Answer: (SHOW ANSWER)
The Cloud DLP API is a service that allows users to inspect, classify, and de-identify sensitive data. It can be used to scan data in Cloud Storage, BigQuery, Cloud Datastore, and Cloud

Pub/Sub. The best way to ensure that the PII is not accessible by unauthorized individuals is to use a quarantine bucket to store the data before scanning it with the DLP API. This way, the data is isolated from other applications and users until it is classified and moved to the appropriate bucket. The other options are not as secure or efficient, as they either expose the data to BigQuery before scanning, or scan the data after writing it to a non-sensitive bucket. Reference:
Cloud DLP documentation
Scanning and classifying Cloud Storage files

NEW QUESTION: 104
Your team has a model deployed to a Vertex AI endpoint You have created a Vertex AI pipeline that automates the model training process and is triggered by a Cloud Function. You need to prioritize keeping the model up-to-date, but also minimize retraining costs. How should you configure retraining'?
A. Configure Pub/Sub to call the Cloud Function when a sufficient amount of new data becomes available.
B. Configure a Cloud Scheduler job that calls the Cloud Function at a predetermined frequency that fits your team's budget.
C. Enable model monitoring on the Vertex AI endpoint Configure Pub/Sub to call the Cloud Function when anomalies are detected.
D. Enable model monitoring on the Vertex AI endpoint Configure Pub/Sub to call the Cloud Function when feature drift is detected.
Answer: D (LEAVE A REPLY)
According to the official exam guide1, one of the skills assessed in the exam is to "configure and optimize model monitoring jobs". Vertex AI Model Monitoring documentation states that "model monitoring helps you detect when your model's performance degrades over time due to changes in the data that your model receives or returns" and that "you can configure model monitoring to send notifications to Pub/Sub when it detects anomalies or drift in your model's predictions"2. Therefore, enabling model monitoring on the Vertex AI endpoint and configuring Pub/Sub to call the Cloud Function when feature drift is detected would help you keep the model up-to-date and minimize retraining costs. The other options are not relevant or optimal for this scenario.
Reference:
Professional ML Engineer Exam Guide
Vertex AI Model Monitoring
Google Professional Machine Learning Certification Exam 2023
Latest Google Professional Machine Learning Engineer Actual Free Exam Questions

NEW QUESTION: 105
You work for a bank and are building a random forest model for fraud detection. You have a dataset that includes transactions, of which 1% are identified as fraudulent. Which data transformation strategy would likely improve the performance of your classifier?
A. Write your data in TFRecords.

B. Z-normalize all the numeric features.
C. Oversample the fraudulent transaction 10 times.
D. Use one-hot encoding on all categorical features.
Answer: C (LEAVE A REPLY)
Oversampling is a technique for dealing with imbalanced datasets, where the majority class dominates the minority class. It balances the distribution of classes by increasing the number of samples in the minority class. Oversampling can improve the performance of a classifier by reducing the bias towards the majority class and increasing the sensitivity to the minority class.
In this case, the dataset includes transactions, of which 1% are identified as fraudulent. This means that the fraudulent transactions are the minority class and the non-fraudulent transactions are the majority class. A random forest model trained on this dataset might have a low recall for the fraudulent transactions, meaning that it might miss many of them and fail to detect fraud. This could have a high cost for the bank and its customers.
One way to overcome this problem is to oversample the fraudulent transactions 10 times, meaning that each fraudulent transaction is duplicated 10 times in the training dataset. This would increase the proportion of fraudulent transactions from 1% to about 10%, making the dataset more balanced. This would also make the random forest model more aware of the patterns and features that distinguish fraudulent transactions from non-fraudulent ones, and thus improve its accuracy and recall for the minority class.
For more information about oversampling and other techniques for imbalanced data, see the following references:
Random Oversampling and Undersampling for Imbalanced Classification
Exploring Oversampling Techniques for Imbalanced Datasets

NEW QUESTION: 106
You work for an advertising company and want to understand the effectiveness of your company's latest advertising campaign. You have streamed 500 MB of campaign data into BigQuery. You want to query the table, and then manipulate the results of that query with a pandas dataframe in an AI Platform notebook. What should you do?
A. Use AI Platform Notebooks' BigQuery cell magic to query the data, and ingest the results as a pandas dataframe
B. Export your table as a CSV file from BigQuery to Google Drive, and use the Google Drive API to ingest the file into your notebook instance
C. Download your table from BigQuery as a local CSV file, and upload it to your AI Platform notebook instance Use pandas. read_csv to ingest the file as a pandas dataframe
D. From a bash cell in your AI Platform notebook, use the bq extract command to export the table as a CSV file to Cloud Storage, and then use gsutii cp to copy the data into the notebook Use pandas. read_csv to ingest the file as a pandas dataframe
Answer: A (LEAVE A REPLY)
AI Platform Notebooks is a service that provides managed Jupyter notebooks for data science and machine learning. You can use AI Platform Notebooks to create, run, and share your code

and analysis in a collaborative and interactive environment1. BigQuery is a service that allows you to analyze large-scale and complex data using SQL queries. You can use BigQuery to stream, store, and query your data in a fast and cost-effective way2. Pandas is a popular Python library that provides data structures and tools for data analysis and manipulation. You can use pandas to create, manipulate, and visualize dataframes, which are tabular data structures with rows and columns3.
AI Platform Notebooks provides a cell magic, %%bigquery, that allows you to run SQL queries on BigQuery data and ingest the results as a pandas dataframe. A cell magic is a special command that applies to the whole cell in a Jupyter notebook. The %%bigquery cell magic can take various arguments, such as the name of the destination dataframe, the name of the destination table in BigQuery, the project ID, and the query parameters4. By using the %%bigquery cell magic, you can query the data in BigQuery with minimal code and manipulate the results with pandas in AI Platform Notebooks. This is the most convenient and efficient way to achieve your goal.
The other options are not as good as option A, because they involve more steps, more code, and more manual effort. Option B requires you to export your table as a CSV file from BigQuery to Google Drive, and then use the Google Drive API to ingest the file into your notebook instance. This option is cumbersome and time-consuming, as it involves moving the data across different services and formats. Option C requires you to download your table from BigQuery as a local CSV file, and then upload it to your AI Platform notebook instance. This option is also inefficient and impractical, as it involves downloading and uploading large files, which can take a long time and consume a lot of bandwidth. Option D requires you to use a bash cell in your AI Platform notebook to export the table as a CSV file to Cloud Storage, and then copy the data into the notebook. This option is also complex and unnecessary, as it involves using different commands and tools to move the data around. Therefore, option A is the best option for this use case.
Reference:
AI Platform Notebooks documentation
BigQuery documentation
pandas documentation
Using Jupyter magics to query BigQuery data

NEW QUESTION: 107
You work on a team that builds state-of-the-art deep learning models by using the TensorFlow framework. Your team runs multiple ML experiments each week which makes it difficult to track the experiment runs. You want a simple approach to effectively track, visualize and debug ML experiment runs on Google Cloud while minimizing any overhead code. How should you proceed?
A. Set up Vertex AI Experiments to track metrics and parameters Configure Vertex AI TensorBoard for visualization.
B. Set up a Cloud Function to write and save metrics files to a Cloud Storage Bucket Configure a Google Cloud VM to host TensorBoard locally for visualization.
C. Set up a Vertex AI Workbench notebook instance Use the instance to save metrics data in a Cloud Storage bucket and to host TensorBoard locally for visualization.
D. Set up a Cloud Function to write and save metrics files to a BigQuery table. Configure a Google Cloud VM to host TensorBoard locally for visualization.
Answer: A (LEAVE A REPLY)
Vertex AI Experiments is a service that allows you to track, compare, and optimize your ML experiments on Google Cloud. You can use Vertex AI Experiments to log metrics and parameters from your TensorFlow models, and then visualize them in Vertex AI TensorBoard. Vertex AI TensorBoard is a managed service that provides a web interface for viewing and debugging your ML experiments. You can use Vertex AI TensorBoard to compare different runs, inspect model graphs, analyze scalars, histograms, images, and more. By using Vertex AI Experiments and Vertex AI TensorBoard, you can simplify your ML experiment tracking and visualization workflow, and avoid the overhead of setting up and maintaining your own Cloud Functions, Cloud Storage buckets, or VMs. Reference:
[Vertex AI Experiments documentation]
[Vertex AI TensorBoard documentation]
Preparing for Google Cloud Certification: Machine Learning Engineer Professional Certificate

NEW QUESTION: 108
You recently developed a deep learning model using Keras, and now you are experimenting with different training strategies. First, you trained the model using a single GPU, but the training process was too slow. Next, you distributed the training across 4 GPUs using tf.distribute.MirroredStrategy (with no other changes), but you did not observe a decrease in training time. What should you do?
A. Distribute the dataset with tf.distribute.Strategy.experimental_distribute_dataset
B. Create a custom training loop.
C. Use a TPU with tf.distribute.TPUStrategy.
D. Increase the batch size.
Answer: (SHOW ANSWER)
Option A is incorrect because distributing the dataset with tf.distribute.Strategy.experimental_distribute_dataset is not the most effective way to decrease

the training time. This method allows you to distribute your dataset across multiple devices or machines, by creating a tf.data.Dataset instance that can be iterated over in parallel1. However, this option may not improve the training time significantly, as it does not change the amount of data or computation that each device or machine has to process. Moreover, this option may introduce additional overhead or complexity, as it requires you to handle the data sharding, replication, and synchronization across the devices or machines1.
Option B is incorrect because creating a custom training loop is not the easiest way to decrease the training time. A custom training loop is a way to implement your own logic for training your model, by using low-level TensorFlow APIs, such as tf.GradientTape, tf.Variable, or tf.function2. A custom training loop may give you more flexibility and control over the training process, but it also requires more effort and expertise, as you have to write and debug the code for each step of the training loop, such as computing the gradients, applying the optimizer, or updating the metrics2. Moreover, a custom training loop may not improve the training time significantly, as it does not change the amount of data or computation that each device or machine has to process.
Option C is incorrect because using a TPU with tf.distribute.TPUStrategy is not a valid way to decrease the training time. A TPU (Tensor Processing Unit) is a custom hardware accelerator designed for high-performance ML workloads3. A tf.distribute.TPUStrategy is a distribution strategy that allows you to distribute your training across multiple TPUs, by creating a tf.distribute.TPUStrategy instance that can be used with high-level TensorFlow APIs, such as Keras4. However, this option is not feasible, as Vertex AI Training does not support TPUs as accelerators for custom training jobs5. Moreover, this option may require significant code changes, as TPUs have different requirements and limitations than GPUs.
Option D is correct because increasing the batch size is the best way to decrease the training time. The batch size is a hyperparameter that determines how many samples of data are processed in each iteration of the training loop. Increasing the batch size may reduce the training time, as it reduces the number of iterations needed to train the model, and it allows each device or machine to process more data in parallel. Increasing the batch size is also easy to implement, as it only requires changing a single hyperparameter. However, increasing the batch size may also affect the convergence and the accuracy of the model, so it is important to find the optimal batch size that balances the trade-off between the training time and the model performance.
Reference:
tf.distribute.Strategy.experimental_distribute_dataset
Custom training loop
TPU overview
tf.distribute.TPUStrategy
Vertex AI Training accelerators
[TPU programming model]
[Batch size and learning rate]
[Keras overview]
[tf.distribute.MirroredStrategy]
[Vertex AI Training overview]

[TensorFlow overview]

NEW QUESTION: 109
You work for a public transportation company and need to build a model to estimate delay times for multiple transportation routes. Predictions are served directly to users in an app in real time. Because different seasons and population increases impact the data relevance, you will retrain the model every month. You want to follow Google-recommended best practices. How should you configure the end-to-end architecture of the predictive model?
A. Configure Kubeflow Pipelines to schedule your multi-step workflow from training to deploying your model.
B. Use a model trained and deployed on BigQuery ML and trigger retraining with the scheduled query feature in BigQuery
C. Write a Cloud Functions script that launches a training and deploying job on Ai Platform that is triggered by Cloud Scheduler
D. Use Cloud Composer to programmatically schedule a Dataflow job that executes the workflow from training to deploying your model
Answer: A (LEAVE A REPLY)
The end-to-end architecture of the predictive model for estimating delay times for multiple transportation routes should be configured using Kubeflow Pipelines. Kubeflow Pipelines is a platform for building and deploying scalable, portable, and reusable machine learning pipelines on Kubernetes. Kubeflow Pipelines allows you to orchestrate your multi-step workflow from data preparation, model training, model evaluation, model deployment, and model serving. Kubeflow Pipelines also provides a user interface for managing and tracking your pipeline runs, experiments, and artifacts1 Using Kubeflow Pipelines has several advantages for this use case: Full automation: You can define your pipeline as a Python script that specifies the steps and dependencies of your workflow, and use the Kubeflow Pipelines SDK to compile and upload your pipeline to the Kubeflow Pipelines service. You can also use the Kubeflow Pipelines UI to create, run, and monitor your pipeline2 Scalability: You can leverage the power of Kubernetes to scale your pipeline components horizontally and vertically, and use distributed training frameworks such as TensorFlow or PyTorch to train your model on multiple nodes or GPUs3 Portability: You can package your pipeline components as Docker containers that can run on any Kubernetes cluster, and use the Kubeflow Pipelines SDK to export and import your pipeline packages across different environments4 Reusability: You can reuse your pipeline components across different pipelines, and share your components with other users through the Kubeflow Pipelines Component Store. You can also use pre-built components from the Kubeflow Pipelines library or other sources5 Schedulability: You can use the Kubeflow Pipelines UI or the Kubeflow Pipelines SDK to schedule recurring pipeline runs based on cron expressions or intervals. For example, you can schedule your pipeline to run every month to retrain your model on the latest data.
The other options are not as suitable for this use case. Using a model trained and deployed on BigQuery ML is not recommended, as BigQuery ML is mainly designed for simple and quick machine learning tasks on large-scale data, and does not support complex models or custom

code. Writing a Cloud Functions script that launches a training and deploying job on AI Platform is not ideal, as Cloud Functions has limitations on the memory, CPU, and execution time, and does not provide a user interface for managing and tracking your pipeline. Using Cloud Composer to programmatically schedule a Dataflow job that executes the workflow from training to deploying your model is not optimal, as Dataflow is mainly designed for data processing and streaming analytics, and does not support model serving or monitoring.

NEW QUESTION: 110
You have been tasked with deploying prototype code to production. The feature engineering code is in PySpark and runs on Dataproc Serverless. The model training is executed by using a Vertex AI custom training job. The two steps are not connected, and the model training must currently be run manually after the feature engineering step finishes. You need to create a scalable and maintainable production process that runs end-to-end and tracks the connections between steps. What should you do?
A. Create a Vertex AI Workbench notebook Use the notebook to submit the Dataproc Serverless feature engineering job Use the same notebook to submit the custom model training job Run the notebook cells sequentially to tie the steps together end-to-end
B. Create a Vertex AI Workbench notebook Initiate an Apache Spark context in the notebook, and run the PySpark feature engineering code Use the same notebook to run the custom model training job in TensorFlow Run the notebook cells sequentially to tie the steps together end-to-end
C. Use the Kubeflow pipelines SDK to write code that specifies two components
- The first is a Dataproc Serverless component that launches the feature engineering job
- The second is a custom component wrapped in the creare_cusrora_rraining_job_from_ccraponent Utility that launches the custom model training job.
D. Create a Vertex AI Pipelines job to link and run both components Use the Kubeflow pipelines SDK to write code that specifies two components
- The first component initiates an Apache Spark context that runs the PySpark feature engineering code
- The second component runs the TensorFlow custom model training code Create a Vertex AI Pipelines job to link and run both components
Answer: (SHOW ANSWER)
The best option for creating a scalable and maintainable production process that runs end-to-end and tracks the connections between steps, using prototype code to production, feature engineering code in PySpark that runs on Dataproc Serverless, and model training that is executed by using a Vertex AI custom training job, is to use the Kubeflow pipelines SDK to write code that specifies two components. The first is a Dataproc Serverless component that launches the feature engineering job. The second is a custom component wrapped in the create_custom_training_job_from_component utility that launches the custom model training job. This option allows you to leverage the power and simplicity of Kubeflow pipelines to orchestrate and automate your machine learning workflows on Vertex AI. Kubeflow pipelines is a platform that

can build, deploy, and manage machine learning pipelines on Kubernetes. Kubeflow pipelines can help you create reusable and scalable pipelines, experiment with different pipeline versions and parameters, and monitor and debug your pipelines. Kubeflow pipelines SDK is a set of Python packages that can help you build and run Kubeflow pipelines. Kubeflow pipelines SDK can help you define pipeline components, specify pipeline parameters and inputs, and create pipeline steps and tasks. A component is a self-contained set of code that performs one step in a pipeline, such as data preprocessing, model training, or model evaluation. A component can be created from a Python function, a container image, or a prebuilt component. A custom component is a component that is not provided by Kubeflow pipelines, but is created by the user to perform a specific task. A custom component can be wrapped in a utility function that can help you create a Vertex AI custom training job from the component. A custom training job is a resource that can run your custom training code on Vertex AI. A custom training job can help you train various types of models, such as linear regression, logistic regression, k-means clustering, matrix factorization, and deep neural networks. By using the Kubeflow pipelines SDK to write code that specifies two components, the first is a Dataproc Serverless component that launches the feature engineering job, and the second is a custom component wrapped in the create_custom_training_job_from_component utility that launches the custom model training job, you can create a scalable and maintainable production process that runs end-to-end and tracks the connections between steps. You can write code that defines the two components, their inputs and outputs, and their dependencies. You can then use the Kubeflow pipelines SDK to create a pipeline that runs the two components in sequence, and submit the pipeline to Vertex AI Pipelines for execution. By using Dataproc Serverless component, you can run your PySpark feature engineering code on Dataproc Serverless, which is a service that can run Spark batch workloads without provisioning and managing your own cluster. By using custom component wrapped in the create_custom_training_job_from_component utility, you can run your custom model training code on Vertex AI, which is a unified platform for building and deploying machine learning solutions on Google Cloud1.

The other options are not as good as option C, for the following reasons:

Option A: Creating a Vertex AI Workbench notebook, using the notebook to submit the Dataproc Serverless feature engineering job, using the same notebook to submit the custom model training job, and running the notebook cells sequentially to tie the steps together end-to-end would require more skills and steps than using the Kubeflow pipelines SDK to write code that specifies two components, the first is a Dataproc Serverless component that launches the feature engineering job, and the second is a custom component wrapped in the create_custom_training_job_from_component utility that launches the custom model training job. Vertex AI Workbench is a service that can provide managed notebooks for machine learning development and experimentation. Vertex AI Workbench can help you create and run JupyterLab notebooks, and access various tools and frameworks, such as TensorFlow, PyTorch, and JAX. By creating a Vertex AI Workbench notebook, using the notebook to submit the Dataproc Serverless feature engineering job, using the same notebook to submit the custom model training job, and running the notebook cells sequentially to tie the steps together end-to-end, you can

create a production process that runs end-to-end and tracks the connections between steps. You can write code that submits the Dataproc Serverless feature engineering job and the custom model training job to Vertex AI, and run the code in the notebook cells. However, creating a Vertex AI Workbench notebook, using the notebook to submit the Dataproc Serverless feature engineering job, using the same notebook to submit the custom model training job, and running the notebook cells sequentially to tie the steps together end-to-end would require more skills and steps than using the Kubeflow pipelines SDK to write code that specifies two components, the first is a Dataproc Serverless component that launches the feature engineering job, and the second is a custom component wrapped in the create_custom_training_job_from_component utility that launches the custom model training job. You would need to write code, create and configure the Vertex AI Workbench notebook, submit the Dataproc Serverless feature engineering job and the custom model training job, and run the notebook cells. Moreover, this option would not use the Kubeflow pipelines SDK, which can simplify the pipeline creation and execution process, and provide various features, such as pipeline parameters, pipeline metrics, and pipeline visualization2.

Option B: Creating a Vertex AI Workbench notebook, initiating an Apache Spark context in the notebook, and running the PySpark feature engineering code, using the same notebook to run the custom model training job in TensorFlow, and running the notebook cells sequentially to tie the steps together end-to-end would not allow you to use Dataproc Serverless to run the feature engineering job, and could increase the complexity and cost of the production process. Apache Spark is a framework that can perform large-scale data processing and machine learning. Apache Spark can help you run various tasks, such as data ingestion, data transformation, data analysis, and data visualization. PySpark is a Python API for Apache Spark. PySpark can help you write and run Spark code in Python. An Apache Spark context is a resource that can initialize and configure the Spark environment. An Apache Spark context can help you create and manage Spark objects, such as SparkSession, SparkConf, and SparkContext. By creating a Vertex AI Workbench notebook, initiating an Apache Spark context in the notebook, and running the PySpark feature engineering code, using the same notebook to run the custom model training job in TensorFlow, and running the notebook cells sequentially to tie the steps together end-to-end, you can create a production process that runs end-to-end and tracks the connections between steps. You can write code that initiates an Apache Spark context and runs the PySpark feature engineering code, and runs the custom model training job in TensorFlow, and run the code in the notebook cells. However, creating a Vertex AI Workbench notebook, initiating an Apache Spark context in the notebook, and running the PySpark feature engineering code, using the same notebook to run the custom model training job in TensorFlow, and running the notebook cells sequentially to tie the steps together end-to-end would not allow you to use Dataproc Serverless to run the feature engineering job, and could increase the complexity and cost of the production process. You would need to write code, create and configure the Vertex AI Workbench notebook, initiate and configure the Apache Spark context, run the PySpark feature engineering code, and run the custom model training job in TensorFlow. Moreover, this option would not use Dataproc Serverless, which is a service that can run Spark batch workloads without provisioning and

managing your own cluster, and provide various benefits, such as autoscaling, dynamic resource allocation, and serverless billing2.
Option D: Creating a Vertex AI Pipelines job to link and run both components, using the Kubeflow pipelines SDK to write code that specifies two components, the first component initiates an Apache Spark context that runs the PySpark feature engineering code, and the second component runs the TensorFlow custom model training code, would not allow you to use Dataproc Serverless to run the feature engineering job, and could increase the complexity and cost of the production process. Vertex AI Pipelines is a service that can run Kubeflow pipelines on Vertex AI. Vertex AI Pipelines can help you create and manage machine learning pipelines, and integrate with various Vertex AI services, such as Vertex AI Workbench, Vertex AI Training, and Vertex AI Prediction. A Vertex AI Pipelines job is a resource that can execute a pipeline on Vertex AI Pipelines. A Vertex AI Pipelines job can help you run your pipeline steps and tasks, and monitor and debug your pipeline execution. By creating a Vertex AI Pipelines job to link and run both components, using the Kubeflow pipelines SDK to write code that specifies two components, the first component initiates an Apache Spark context that runs the PySpark feature engineering code, and the second component runs the TensorFlow custom model training code, you can create a scalable and maintainable production process that runs end-to-end and tracks the connections between steps. You can write code that defines the two components, their inputs and outputs, and their dependencies. You can then use the Kubeflow pipelines SDK to create a pipeline that runs the two components in sequence, and submit the pipeline to Vertex AI Pipelines for execution. However, creating a Vertex AI Pipelines job to link and run both components, using the Kubeflow pipelines SDK to write code that specifies two components, the first component initiates an Apache Spark context that runs the PySpark feature engineering code,

NEW QUESTION: 111
Your team needs to build a model that predicts whether images contain a driver's license, passport, or credit card. The data engineering team already built the pipeline and generated a dataset composed of 10,000 images with driver's licenses, 1,000 images with passports, and 1,000 images with credit cards. You now have to train a model with the following label map: ['driversjicense', 'passport', 'credit_card']. Which loss function should you use?
A. Categorical hinge
B. Binary cross-entropy
C. Categorical cross-entropy
D. Sparse categorical cross-entropy
Answer: C (LEAVE A REPLY)
Categorical cross-entropy is a loss function that is suitable for multi-class classification problems, where the target variable has more than two possible values. Categorical cross-entropy measures the difference between the true probability distribution of the target classes and the predicted probability distribution of the model. It is defined as:
L - sum(y_i * log(p_i))

where y_i is the true probability of class i, and p_i is the predicted probability of class i. Categorical cross-entropy penalizes the model for making incorrect predictions, and encourages the model to assign high probabilities to the correct classes and low probabilities to the incorrect classes.
For the use case of building a model that predicts whether images contain a driver's license, passport, or credit card, categorical cross-entropy is the appropriate loss function to use. This is because the problem is a multi-class classification problem, where the target variable has three possible values: ['drivers_license', 'passport', 'credit_card']. The label map is a list that maps the class names to the class indices, such that 'drivers_license' corresponds to index 0, 'passport' corresponds to index 1, and 'credit_card' corresponds to index 2. The model should output a probability distribution over the three classes for each image, and the categorical cross-entropy loss function should compare the output with the true labels. Therefore, categorical cross-entropy is the best loss function for this use case.

NEW QUESTION: 112
Your work for a textile manufacturing company. Your company has hundreds of machines and each machine has many sensors. Your team used the sensory data to build hundreds of ML models that detect machine anomalies Models are retrained daily and you need to deploy these models in a cost-effective way. The models must operate 24/7 without downtime and make sub millisecond predictions. What should you do?
A. Deploy a Dataflow batch pipeline and a Vertex AI Prediction endpoint.
B. Deploy a Dataflow batch pipeline with the RunInference API. and use model refresh.
C. Deploy a Dataflow streaming pipeline and a Vertex AI Prediction endpoint with autoscaling.
D. Deploy a Dataflow streaming pipeline with the RunInference API and use automatic model refresh.
Answer: D (LEAVE A REPLY)
A Dataflow streaming pipeline is a cost-effective way to process large volumes of real-time data from sensors. The RunInference API is a Dataflow transform that allows you to run online predictions on your streaming data using your ML models. By using the RunInference API, you can avoid the latency and cost of using a separate prediction service. The automatic model refresh feature enables you to update your models in the pipeline without redeploying the pipeline. This way, you can ensure that your models are always up-to-date and accurate. By deploying a Dataflow streaming pipeline with the RunInference API and using automatic model refresh, you can achieve sub-millisecond predictions, 24/7 availability, and low operational overhead for your ML models. Reference:
Dataflow documentation
RunInference API documentation
Automatic model refresh documentation
Preparing for Google Cloud Certification: Machine Learning Engineer Professional Certificate

NEW QUESTION: 113

You received a training-serving skew alert from a Vertex AI Model Monitoring job running in production. You retrained the model with more recent training data, and deployed it back to the Vertex AI endpoint but you are still receiving the same alert. What should you do?

A. Update the model monitoring job to use a lower sampling rate.

B. Update the model monitoring job to use the more recent training data that was used to retrain the model.

C. Temporarily disable the alert Enable the alert again after a sufficient amount of new production traffic has passed through the Vertex AI endpoint.

D. Temporarily disable the alert until the model can be retrained again on newer training data Retrain the model again after a sufficient amount of new production traffic has passed through the Vertex AI endpoint

Answer: B (LEAVE A REPLY)

The best option for resolving the training-serving skew alert is to update the model monitoring job to use the more recent training data that was used to retrain the model. This option can help align the baseline distribution of the model monitoring job with the current distribution of the production data, and eliminate the false positive alerts. Model Monitoring is a service that can track and compare the results of multiple machine learning runs. Model Monitoring can monitor the model's prediction input data for feature skew and drift. Training-serving skew occurs when the feature data distribution in production deviates from the feature data distribution used to train the model. If the original training data is available, you can enable skew detection to monitor your models for training-serving skew. Model Monitoring uses TensorFlow Data Validation (TFDV) to calculate the distributions and distance scores for each feature, and compares them with a baseline distribution. The baseline distribution is the statistical distribution of the feature's values in the training data. If the distance score for a feature exceeds an alerting threshold that you set, Model Monitoring sends you an email alert. However, if you retrain the model with more recent training data, and deploy it back to the Vertex AI endpoint, the baseline distribution of the model monitoring job may become outdated and inconsistent with the current distribution of the production data. This can cause the model monitoring job to generate false positive alerts, even if the model performance is not deteriorated. To avoid this problem, you need to update the model monitoring job to use the more recent training data that was used to retrain the model. This can help the model monitoring job to recalculate the baseline distribution and the distance scores, and compare them with the current distribution of the production data. This can also help the model monitoring job to detect any true positive alerts, such as a sudden change in the production data that causes the model performance to degrade1.

The other options are not as good as option B, for the following reasons:

Option A: Updating the model monitoring job to use a lower sampling rate would not resolve the training-serving skew alert, and could reduce the accuracy and reliability of the model monitoring job. The sampling rate is a parameter that determines the percentage of prediction requests that are logged and analyzed by the model monitoring job. Using a lower sampling rate can reduce the storage and computation costs of the model monitoring job, but also the quality and validity of the data. Using a lower sampling rate can introduce sampling bias and noise into the data, and

make the model monitoring job miss some important features or patterns of the data. Moreover, using a lower sampling rate would not address the root cause of the training-serving skew alert, which is the mismatch between the baseline distribution and the current distribution of the production data2.
Option C: Temporarily disabling the alert, and enabling the alert again after a sufficient amount of new production traffic has passed through the Vertex AI endpoint, would not resolve the training-serving skew alert, and could expose the model to potential risks and errors. Disabling the alert would stop the model monitoring job from sending email notifications when the distance score for a feature exceeds the alerting threshold, but it would not stop the model monitoring job from calculating and comparing the distributions and distance scores. Therefore, disabling the alert would not address the root cause of the training-serving skew alert, which is the mismatch between the baseline distribution and the current distribution of the production data. Moreover, disabling the alert would prevent the model monitoring job from detecting any true positive alerts, such as a sudden change in the production data that causes the model performance to degrade. This can expose the model to potential risks and errors, and affect the user satisfaction and trust1.
Option D: Temporarily disabling the alert until the model can be retrained again on newer training data, and retraining the model again after a sufficient amount of new production traffic has passed through the Vertex AI endpoint, would not resolve the training-serving skew alert, and could cause unnecessary costs and efforts. Disabling the alert would stop the model monitoring job from sending email notifications when the distance score for a feature exceeds the alerting threshold, but it would not stop the model monitoring job from calculating and comparing the distributions and distance scores. Therefore, disabling the alert would not address the root cause of the training-serving skew alert, which is the mismatch between the baseline distribution and the current distribution of the production data. Moreover, disabling the alert would prevent the model monitoring job from detecting any true positive alerts, such as a sudden change in the production data that causes the model performance to degrade. This can expose the model to potential risks and errors, and affect the user satisfaction and trust. Retraining the model again on newer training data would create a new model version, but it would not update the model monitoring job to use the newer training data as the baseline distribution. Therefore, retraining the model again on newer training data would not resolve the training-serving skew alert, and could cause unnecessary costs and efforts1.
Reference:
Preparing for Google Cloud Certification: Machine Learning Engineer, Course 3: Production ML Systems, Week 4: Evaluation Google Cloud Professional Machine Learning Engineer Exam Guide, Section 3: Scaling ML models in production, 3.3 Monitoring ML models in production Official Google Cloud Certified Professional Machine Learning Engineer Study Guide, Chapter 6: Production ML Systems, Section 6.3: Monitoring ML Models Using Model Monitoring Understanding the score threshold slider Sampling rate

NEW QUESTION: 114

You deployed an ML model into production a year ago. Every month, you collect all raw requests that were sent to your model prediction service during the previous month. You send a subset of these requests to a human labeling service to evaluate your model's performance. After a year, you notice that your model's performance sometimes degrades significantly after a month, while other times it takes several months to notice any decrease in performance. The labeling service is costly, but you also need to avoid large performance degradations. You want to determine how often you should retrain your model to maintain a high level of performance while minimizing cost. What should you do?

A. Train an anomaly detection model on the training dataset, and run all incoming requests through this model. If an anomaly is detected, send the most recent serving data to the labeling service.

B. Identify temporal patterns in your model's performance over the previous year. Based on these patterns, create a schedule for sending serving data to the labeling service for the next year.

C. Compare the cost of the labeling service with the lost revenue due to model performance degradation over the past year. If the lost revenue is greater than the cost of the labeling service, increase the frequency of model retraining; otherwise, decrease the model retraining frequency.

D. Run training-serving skew detection batch jobs every few days to compare the aggregate statistics of the features in the training dataset with recent serving data. If skew is detected, send the most recent serving data to the labeling service.

Answer: D (LEAVE A REPLY)

The best option for determining how often to retrain your model to maintain a high level of performance while minimizing cost is to run training-serving skew detection batch jobs every few days. Training-serving skew refers to the discrepancy between the distributions of the features in the training dataset and the serving data. This can cause the model to perform poorly on the new data, as it is not representative of the data that the model was trained on. By running training-serving skew detection batch jobs, you can monitor the changes in the feature distributions over time, and identify when the skew becomes significant enough to affect the model performance. If skew is detected, you can send the most recent serving data to the labeling service, and use the labeled data to retrain your model. This option has the following benefits:

It allows you to retrain your model only when necessary, based on the actual data changes, rather than on a fixed schedule or a heuristic. This can save you the cost of the labeling service and the retraining process, and also avoid overfitting or underfitting your model.

It leverages the existing tools and frameworks for training-serving skew detection, such as TensorFlow Data Validation (TFDV) and Vertex Data Labeling. TFDV is a library that can compute and visualize descriptive statistics for your datasets, and compare the statistics across different datasets. Vertex Data Labeling is a service that can label your data with high quality and low latency, using either human labelers or automated labelers.

It integrates well with the MLOps practices, such as continuous integration and continuous delivery (CI/CD), which can automate the workflow of running the skew detection jobs, sending the data to the labeling service, retraining the model, and deploying the new model version.

The other options are less optimal for the following reasons:

Option A: Training an anomaly detection model on the training dataset, and running all incoming requests through this model, introduces additional complexity and overhead. This option requires building and maintaining a separate model for anomaly detection, which can be challenging and time-consuming. Moreover, this option requires running the anomaly detection model on every request, which can increase the latency and resource consumption of the prediction service. Additionally, this option may not capture the subtle changes in the feature distributions that can affect the model performance, as anomalies are usually defined as rare or extreme events.
Option B: Identifying temporal patterns in your model's performance over the previous year, and creating a schedule for sending serving data to the labeling service for the next year, introduces additional assumptions and risks. This option requires analyzing the historical data and model performance, and finding the patterns that can explain the variations in the model performance over time. However, this can be difficult and unreliable, as the patterns may not be consistent or predictable, and may depend on various factors that are not captured by the data. Moreover, this option requires creating a schedule based on the past patterns, which may not reflect the future changes in the data or the environment. This can lead to either sending too much or too little data to the labeling service, resulting in either wasted cost or degraded performance.
Option C: Comparing the cost of the labeling service with the lost revenue due to model performance degradation over the past year, and adjusting the frequency of model retraining accordingly, introduces additional challenges and trade-offs. This option requires estimating the cost of the labeling service and the lost revenue due to model performance degradation, which can be difficult and inaccurate, as they may depend on various factors that are not easily quantifiable or measurable. Moreover, this option requires finding the optimal balance between the cost and the performance, which can be subjective and variable, as different stakeholders may have different preferences and expectations. Furthermore, this option may not account for the potential impact of the model performance degradation on other aspects of the business, such as customer satisfaction, retention, or loyalty.

NEW QUESTION: 115
You work at a subscription-based company. You have trained an ensemble of trees and neural networks to predict customer churn, which is the likelihood that customers will not renew their yearly subscription. The average prediction is a 15% churn rate, but for a particular customer the model predicts that they are 70% likely to churn. The customer has a product usage history of 30%, is located in New York City, and became a customer in 1997. You need to explain the difference between the actual prediction, a 70% churn rate, and the average prediction. You want to use Vertex Explainable AI. What should you do?
A. Train local surrogate models to explain individual predictions.
B. Configure sampled Shapley explanations on Vertex Explainable AI.
C. Configure integrated gradients explanations on Vertex Explainable AI.
D. Measure the effect of each feature as the weight of the feature multiplied by the feature value.
Answer: B (LEAVE A REPLY)

Option A is incorrect because training local surrogate models to explain individual predictions is not a feature of Vertex Explainable AI, but rather a general technique for interpreting black-box models. Local surrogate models are simpler models that approximate the behavior of the original model around a specific input1.
Option B is correct because configuring sampled Shapley explanations on Vertex Explainable AI is a way to explain the difference between the actual prediction and the average prediction for a given input. Sampled Shapley explanations are based on the Shapley value, which is a game-theoretic concept that measures how much each feature contributes to the prediction2. Vertex Explainable AI supports sampled Shapley explanations for tabular data, such as customer churn3.
Option C is incorrect because configuring integrated gradients explanations on Vertex Explainable AI is not suitable for explaining the difference between the actual prediction and the average prediction for a given input. Integrated gradients explanations are based on the idea of computing the gradients of the prediction with respect to the input features along a path from a baseline input to the actual input4. Vertex Explainable AI supports integrated gradients explanations for image and text data, but not for tabular data3.
Option D is incorrect because measuring the effect of each feature as the weight of the feature multiplied by the feature value is not a valid way to explain the difference between the actual prediction and the average prediction for a given input. This method assumes that the model is linear and additive, which is not the case for an ensemble of trees and neural networks. Moreover, this method does not account for the interactions between features or the non-linearity of the model5.
Reference:
Local surrogate models
Shapley value
Vertex Explainable AI overview
Integrated gradients
Feature importance

NEW QUESTION: 116
You are building a TensorFlow model for a financial institution that predicts the impact of consumer spending on inflation globally. Due to the size and nature of the data, your model is long-running across all types of hardware, and you have built frequent checkpointing into the training process. Your organization has asked you to minimize cost. What hardware should you choose?
A. A Vertex AI Workbench user-managed notebooks instance running on an n1-standard-16 with 4 NVIDIA P100 GPUs
B. A Vertex AI Workbench user-managed notebooks instance running on an n1-standard-16 with an NVIDIA P100 GPU
C. A Vertex AI Workbench user-managed notebooks instance running on an n1-standard-16 with a non-preemptible v3-8 TPU

D. A Vertex AI Workbench user-managed notebooks instance running on an n1-standard-16 with a preemptible v3-8 TPU

Answer: D (LEAVE A REPLY)

The best hardware to choose for your model while minimizing cost is a Vertex AI Workbench user-managed notebooks instance running on an n1-standard-16 with a preemptible v3-8 TPU. This hardware configuration can provide you with high performance, scalability, and efficiency for your TensorFlow model, as well as low cost and flexibility for your long-running and checkpointing process. The v3-8 TPU is a cloud tensor processing unit (TPU) device, which is a custom ASIC chip designed by Google to accelerate ML workloads. It can handle large and complex models and datasets, and offer fast and stable training and inference. The n1-standard-16 is a general-purpose VM that can support the CPU and memory requirements of your model, as well as the data preprocessing and postprocessing tasks. By choosing a preemptible v3-8 TPU, you can take advantage of the lower price and availability of the TPU devices, as long as you can tolerate the possibility of the device being reclaimed by Google at any time. However, since you have built frequent checkpointing into your training process, you can resume your model from the last saved state, and avoid losing any progress or data. Moreover, you can use the Vertex AI Workbench user-managed notebooks to create and manage your notebooks instances, and leverage the integration with Vertex AI and other Google Cloud services.

The other options are not optimal for the following reasons:

A . A Vertex AI Workbench user-managed notebooks instance running on an n1-standard-16 with 4 NVIDIA P100 GPUs is not a good option, as it has higher cost and lower performance than the v3-8 TPU. The NVIDIA P100 GPUs are the previous generation of GPUs from NVIDIA, which have lower performance, scalability, and efficiency than the latest NVIDIA A100 GPUs or the TPUs. They also have higher price and lower availability than the preemptible TPUs, which can increase the cost and complexity of your solution.

B . A Vertex AI Workbench user-managed notebooks instance running on an n1-standard-16 with an NVIDIA P100 GPU is not a good option, as it has higher cost and lower performance than the v3-8 TPU. It also has less GPU memory and compute power than the option with 4 NVIDIA P100 GPUs, which can limit the size and complexity of your model, and affect the training and inference speed and quality.

C . A Vertex AI Workbench user-managed notebooks instance running on an n1-standard-16 with a non-preemptible v3-8 TPU is not a good option, as it has higher cost and lower flexibility than the preemptible v3-8 TPU. The non-preemptible v3-8 TPU has the same performance, scalability, and efficiency as the preemptible v3-8 TPU, but it has higher price and lower availability, as it is reserved for your exclusive use. Moreover, since your model is long-running and checkpointing, you do not need the guarantee of the device not being reclaimed by Google, and you can benefit from the lower cost and higher availability of the preemptible v3-8 TPU.

Reference:

Professional ML Engineer Exam Guide

Preparing for Google Cloud Certification: Machine Learning Engineer Professional Certificate Google Cloud launches machine learning engineer certification Cloud TPU Vertex AI Workbench user-managed notebooks Preemptible VMs NVIDIA Tesla P100 GPU

NEW QUESTION: 117
You built and manage a production system that is responsible for predicting sales numbers. Model accuracy is crucial, because the production model is required to keep up with market changes. Since being deployed to production, the model hasn't changed; however the accuracy of the model has steadily deteriorated. What issue is most likely causing the steady decline in model accuracy?
A. Poor data quality
B. Lack of model retraining
C. Too few layers in the model for capturing information
D. Incorrect data split ratio during model training, evaluation, validation, and test
Answer: B (LEAVE A REPLY)
Model retraining is the process of updating an existing machine learning model with new data and parameters to improve its performance and accuracy. Model retraining is essential for maintaining the relevance and validity of the model, especially when the data or the environment changes over time. Model retraining can help to avoid or reduce the effects of model degradation, which is the phenomenon of the model's predictive performance decreasing as it is tested on new datasets within rapidly evolving environments1.
For the use case of predicting sales numbers, model accuracy is crucial, because the production model is required to keep up with market changes. Market changes can affect the demand, supply, price, and preference of the products, and thus influence the sales numbers. If the model is not retrained with new data that reflects the market changes, it may become outdated and inaccurate, and fail to capture the patterns and trends of the sales numbers. Therefore, the most likely issue that is causing the steady decline in model accuracy is the lack of model retraining.
The other options are not as likely as option B, because they are not directly related to the model's ability to adapt to market changes. Option A, poor data quality, may affect the model's accuracy, but it is not a specific cause of model degradation over time. Option C, too few layers in the model for capturing information, may affect the model's complexity and expressiveness, but it is not a specific cause of model degradation over time. Option D, incorrect data split ratio during model training, evaluation, validation, and test, may affect the model's generalization and validation, but it is not a specific cause of model degradation over time. Therefore, option B, lack of model retraining, is the best answer for this question.
Reference:
Beware Steep Decline: Understanding Model Degradation In Machine Learning Models

NEW QUESTION: 118
Your company needs to generate product summaries for vendors. You evaluated a foundation model from Model Garden for text summarization but found that the summaries do not align with

your company's brand voice. How should you improve this LLM-based summarization model to better meet your business objectives?

A. Increase the model's temperature parameter.

B. Fine-tune the model using a company-specific dataset.

C. Tune the token output limit in the response.

D. Replace the pre-trained model with another model in Model Garden.

Answer: B (LEAVE A REPLY)

Fine-tuning the model with a company-specific dataset aligns the model outputs with the brand voice, making it better suited for the company's objectives. Adjusting the temperature (Option A) affects randomness rather than content style, and changing token limits (Option C) does not impact tone. Replacing the model (Option D) is inefficient without guarantees of better alignment.

NEW QUESTION: 119

During batch training of a neural network, you notice that there is an oscillation in the loss. How should you adjust your model to ensure that it converges?

A. Increase the size of the training batch

B. Decrease the size of the training batch

C. Increase the learning rate hyperparameter

D. Decrease the learning rate hyperparameter

Answer: D (LEAVE A REPLY)

Oscillation in the loss during batch training of a neural network means that the model is overshooting the optimal point of the loss function and bouncing back and forth. This can prevent the model from converging to the minimum loss value. One of the main reasons for this phenomenon is that the learning rate hyperparameter, which controls the size of the steps that the model takes along the gradient, is too high. Therefore, decreasing the learning rate hyperparameter can help the model take smaller and more precise steps and avoid oscillation. This is a common technique to improve the stability and performance of neural network training12.

Reference:

Interpreting Loss Curves

Is learning rate the only reason for training loss oscillation after few epochs?

NEW QUESTION: 120

You need to analyze user activity data from your company's mobile applications. Your team will use BigQuery for data analysis, transformation, and experimentation with ML algorithms. You need to ensure real-time ingestion of the user activity data into BigQuery. What should you do?

A. Configure Pub/Sub to stream the data into BigQuery.

B. Run an Apache Spark streaming job on Dataproc to ingest the data into BigQuery.

C. Run a Dataflow streaming job to ingest the data into BigQuery.

D. Configure Pub/Sub and a Dataflow streaming job to ingest the data into BigQuery,

Answer: (SHOW ANSWER)

The best option to ensure real-time ingestion of the user activity data into BigQuery is to run a Dataflow streaming job to ingest the data into BigQuery. Dataflow is a fully managed service that can handle both batch and stream processing of data, and can integrate seamlessly with BigQuery and other Google Cloud services. Dataflow can also use Apache Beam as the programming model, which provides a unified and portable API for developing data pipelines. By using Dataflow, you can avoid the complexity and overhead of managing your own infrastructure, and focus on the logic and transformation of your data. Dataflow can also handle various types of data, such as structured, unstructured, or binary data, and can apply windowing, aggregation, and other operations on the data streams.
The other options are not optimal for the following reasons:
A . Configuring Pub/Sub to stream the data into BigQuery is not a good option, as Pub/Sub is a messaging service that can publish and subscribe to data streams, but cannot perform any transformation or processing on the data. Pub/Sub can be used as a source or a sink for Dataflow, but not as a standalone solution for ingesting data into BigQuery.
B . Running an Apache Spark streaming job on Dataproc to ingest the data into BigQuery is not a good option, as it requires setting up and managing your own cluster of virtual machines, which can increase the cost and complexity of your solution. Moreover, Apache Spark is not natively integrated with BigQuery, and requires using connectors or intermediate storage to write data to BigQuery, which can introduce latency and inefficiency.
D . Configuring Pub/Sub and a Dataflow streaming job to ingest the data into BigQuery is not a bad option, but it is not necessary, as Dataflow can directly read data from the mobile applications without using Pub/Sub as an intermediary. Using Pub/Sub can add an extra layer of abstraction and reliability, but it can also increase the cost and complexity of your solution, and introduce some delay in the data ingestion.
Reference:
Professional ML Engineer Exam Guide
Preparing for Google Cloud Certification: Machine Learning Engineer Professional Certificate
Google Cloud launches machine learning engineer certification Dataflow documentation BigQuery documentation

NEW QUESTION: 121
You are developing a model to identify traffic signs in images extracted from videos taken from the dashboard of a vehicle. You have a dataset of 100 000 images that were cropped to show one out of ten different traffic signs. The images have been labeled accordingly for model training and are stored in a Cloud Storage bucket You need to be able to tune the model during each training run. How should you train the model?
A. Train a model for object detection by using Vertex AI AutoML.
B. Train a model for image classification by using Vertex AI AutoML.
C. Develop the model training code for object detection and tram a model by using Vertex AI custom training.

D. Develop the model training code for image classification and train a model by using Vertex AI custom training.

Answer: D (LEAVE A REPLY)

Image classification is a task where the model assigns a label to an image based on its content, such as "stop sign" or "speed limit"1. Object detection is a task where the model locates and identifies multiple objects in an image, and draws bounding boxes around them2. Since your dataset consists of images that were cropped to show one out of ten different traffic signs, you are dealing with an image classification problem, not an object detection problem. Therefore, you need to train a model for image classification, not object detection.

Vertex AI AutoML is a service that allows you to train and deploy high-quality ML models with minimal effort and machine learning expertise3. You can use Vertex AI AutoML to train a model for image classification by uploading your images and labels to a Vertex AI dataset, and then launching an AutoML training job4. However, Vertex AI AutoML does not allow you to tune the model during each training run, as it automatically selects the best model architecture and hyperparameters for your data4.

Vertex AI custom training is a service that allows you to train and deploy your own custom ML models using your own code and frameworks5. You can use Vertex AI custom training to train a model for image classification by writing your own model training code, such as using TensorFlow or PyTorch, and then creating and running a custom training job. Vertex AI custom training allows you to tune the model during each training run, as you can specify the model architecture and hyperparameters in your code, and use Vertex AI Hyperparameter Tuning to optimize them .

Therefore, the best option for your scenario is to develop the model training code for image classification and train a model by using Vertex AI custom training.

Reference:

Image classification | TensorFlow Core

Object detection | TensorFlow Core

Introduction to Vertex AI AutoML | Google Cloud

AutoML Vision | Google Cloud

Introduction to Vertex AI custom training | Google Cloud

[Custom training with TensorFlow | Vertex AI | Google Cloud]

[Hyperparameter tuning overview | Vertex AI | Google Cloud]

NEW QUESTION: 122
You are developing a model to help your company create more targeted online advertising campaigns. You need to create a dataset that you will use to train the model. You want to avoid creating or reinforcing unfair bias in the model. What should you do?
Choose 2 answers
A. Include a comprehensive set of demographic features.
B. include only the demographic groups that most frequently interact with advertisements.
C. Collect a random sample of production traffic to build the training dataset.
D. Collect a stratified sample of production traffic to build the training dataset.
E. Conduct fairness tests across sensitive categories and demographics on the trained model.
Answer: (SHOW ANSWER)
To avoid creating or reinforcing unfair bias in the model, you should collect a representative sample of production traffic to build the training dataset, and conduct fairness tests across sensitive categories and demographics on the trained model. A representative sample is one that reflects the true distribution of the population, and does not over- or under-represent any group. A random sample is a simple way to obtain a representative sample, as it ensures that every data point has an equal chance of being selected. A stratified sample is another way to obtain a representative sample, as it ensures that every subgroup has a proportional representation in the sample. However, a stratified sample requires prior knowledge of the subgroups and their sizes, which may not be available or easy to obtain. Therefore, a random sample is a more feasible option in this case. A fairness test is a way to measure and evaluate the potential bias and discrimination of the model, based on different categories and demographics, such as age, gender, race, etc. A fairness test can help you identify and mitigate any unfair outcomes or impacts of the model, and ensure that the model treats all groups fairly and equitably. A fairness test can be conducted using various methods and tools, such as confusion matrices, ROC curves, fairness indicators, etc. Reference: The answer can be verified from official Google Cloud documentation and resources related to data sampling and fairness testing.
Sampling data | BigQuery
Fairness Indicators | TensorFlow
What-if Tool | TensorFlow

NEW QUESTION: 123
You work for a retail company. You have created a Vertex AI forecast model that produces monthly item sales predictions. You want to quickly create a report that will help to explain how the model calculates the predictions. You have one month of recent actual sales data that was not included in the training dataset. How should you generate data for your report?
A. Create a batch prediction job by using the actual sales data Compare the predictions to the actuals in the report.

B. Create a batch prediction job by using the actual sates data and configure the job settings to generate feature attributions. Compare the results in the report.
C. Generate counterfactual examples by using the actual sales data Create a batch prediction job using the actual sales data and the counterfactual examples Compare the results in the report.
D. Train another model by using the same training dataset as the original and exclude some columns. Using the actual sales data create one batch prediction job by using the new model and another one with the original model Compare the two sets of predictions in the report.
Answer: B (LEAVE A REPLY)
According to the official exam guide1, one of the skills assessed in the exam is to "explain the predictions of a trained model". Vertex AI provides feature attributions using Shapley Values, a cooperative game theory algorithm that assigns credit to each feature in a model for a particular outcome2. Feature attributions can help you understand how the model calculates the predictions and debug or optimize the model accordingly. You can use Forecasting with AutoML or Tabular Workflow for Forecasting to generate and query local feature attributions2. The other options are not relevant or optimal for this scenario. Reference:
Professional ML Engineer Exam Guide
Feature attributions for forecasting
Google Professional Machine Learning Certification Exam 2023
Latest Google Professional Machine Learning Engineer Actual Free Exam Questions

NEW QUESTION: 124
You are developing an ML model that predicts the cost of used automobiles based on data such as location, condition model type color, and engine-'battery efficiency. The data is updated every night Car dealerships will use the model to determine appropriate car prices. You created a Vertex AI pipeline that reads the data splits the data into training/evaluation/test sets performs feature engineering trains the model by using the training dataset and validates the model by using the evaluation dataset. You need to configure a retraining workflow that minimizes cost What should you do?
A. Compare the training and evaluation losses of the current run If the losses are similar, deploy the model to a Vertex AI endpoint Configure a cron job to redeploy the pipeline every night.
B. Compare the training and evaluation losses of the current run If the losses are similar deploy the model to a Vertex AI endpoint with training/serving skew threshold model monitoring When the model monitoring threshold is tnggered redeploy the pipeline.
C. Compare the results to the evaluation results from a previous run If the performance improved deploy the model to a Vertex AI endpoint Configure a cron job to redeploy the pipeline every night.
D. Compare the results to the evaluation results from a previous run If the performance improved deploy the model to a Vertex AI endpoint with training/serving skew threshold model monitoring. When the model monitoring threshold is triggered, redeploy the pipeline.
Answer: B (LEAVE A REPLY)

Comparing the training and evaluation losses of the current run is a good way to check if the model is overfitting or underfitting. If the losses are similar, it means that the model is generalizing well and can be deployed to a Vertex AI endpoint. Vertex AI endpoint is a service that allows you to serve your ML models online and scale them automatically. By using a training/serving skew threshold model monitoring, you can detect if there is a significant difference between the data used for training and the data used for serving. This can indicate that the model is becoming stale or inaccurate over time. When the model monitoring threshold is triggered, it means that the model needs to be retrained with the latest data. By redeploying the pipeline, you can automate the retraining process and update the model with the new data. This way, you can minimize the cost of retraining and ensure that your model is always up-to-date and accurate. Reference:
Vertex AI documentation
Vertex AI endpoint documentation
Model monitoring documentation
Preparing for Google Cloud Certification: Machine Learning Engineer Professional Certificate

NEW QUESTION: 125
You are an ML engineer at a regulated insurance company. You are asked to develop an insurance approval model that accepts or rejects insurance applications from potential customers. What factors should you consider before building the model?
A. Redaction, reproducibility, and explainability
B. Traceability, reproducibility, and explainability
C. Federated learning, reproducibility, and explainability
D. Differential privacy federated learning, and explainability
Answer: B (LEAVE A REPLY)
Before building an insurance approval model, an ML engineer should consider the factors of traceability, reproducibility, and explainability, as these are important aspects of responsible AI and fairness in a regulated domain. Traceability is the ability to track the provenance and lineage of the data, models, and decisions throughout the ML lifecycle. It helps to ensure the quality, reliability, and accountability of the ML system, and to comply with the regulatory and ethical standards. Reproducibility is the ability to recreate the same results and outcomes using the same data, models, and parameters. It helps to verify the validity, consistency, and robustness of the ML system, and to debug and improve the performance. Explainability is the ability to understand and interpret the logic, behavior, and outcomes of the ML system. It helps to increase the transparency, trust, and confidence of the ML system, and to identify and mitigate any potential biases, errors, or risks. The other options are not as relevant or comprehensive as this option. Redaction is the process of removing sensitive or confidential information from the data or documents, but it is not a factor that the ML engineer should consider before building the model, as it is more related to the data preparation and protection. Federated learning is a technique that allows training ML models on decentralized data without transferring the data to a central server, but it is not a factor that the ML engineer should consider before building the model, as it is more related to the model architecture and privacy preservation. Differential privacy is a method that

adds noise to the data or the model outputs to protect the individual privacy of the data subjects, but it is not a factor that the ML engineer should consider before building the model, as it is more related to the model evaluation and deployment. Reference:
Responsible AI documentation
Traceability documentation
Reproducibility documentation
Explainability documentation

NEW QUESTION: 126
You are creating a social media app where pet owners can post images of their pets. You have one million user uploaded images with hashtags. You want to build a comprehensive system that recommends images to users that are similar in appearance to their own uploaded images.
What should you do?
A. Download a pretrained convolutional neural network, and fine-tune the model to predict hashtags based on the input images. Use the predicted hashtags to make recommendations.
B. Retrieve image labels and dominant colors from the input images using the Vision API. Use these properties and the hashtags to make recommendations.
C. Use the provided hashtags to create a collaborative filtering algorithm to make recommendations.
D. Download a pretrained convolutional neural network, and use the model to generate embeddings of the input images. Measure similarity between embeddings to make recommendations.
Answer: D (LEAVE A REPLY)
The best option to build a comprehensive system that recommends images to users that are similar in appearance to their own uploaded images is to download a pretrained convolutional neural network (CNN), and use the model to generate embeddings of the input images. Embeddings are low-dimensional representations of high-dimensional data that capture the essential features and semantics of the data. By using a pretrained CNN, you can leverage the knowledge learned from large-scale image datasets, such as ImageNet, and apply it to your own domain. A pretrained CNN can be used as a feature extractor, where the output of the last hidden layer (or any intermediate layer) is taken as the embedding vector for the input image. You can then measure the similarity between embeddings using a distance metric, such as cosine similarity or Euclidean distance, and recommend images that have the highest similarity scores to the user's uploaded image. Option A is incorrect because downloading a pretrained CNN and fine-tuning the model to predict hashtags based on the input images may not capture the visual similarity of the images, as hashtags may not reflect the appearance of the images accurately. For example, two images of different breeds of dogs may have the same hashtag #dog, but they may not look similar to each other. Moreover, fine-tuning the model may require additional data and computational resources, and it may not generalize well to new images that have different or missing hashtags. Option B is incorrect because retrieving image labels and dominant colors from the input images using the Vision API may not capture the visual similarity of the images, as

labels and colors may not reflect the fine-grained details of the images. For example, two images of the same breed of dog may have different labels and colors depending on the background, lighting, and angle of the image. Moreover, using the Vision API may incur additional costs and latency, and it may not be able to handle custom or domain-specific labels. Option C is incorrect because using the provided hashtags to create a collaborative filtering algorithm may not capture the visual similarity of the images, as collaborative filtering relies on the ratings or preferences of users, not the features of the images. For example, two images of different animals may have similar ratings or preferences from users, but they may not look similar to each other. Moreover, collaborative filtering may suffer from the cold start problem, where new images or users that have no ratings or preferences cannot be recommended. Reference:
Image similarity search with TensorFlow
Image embeddings documentation
Pretrained models documentation
Similarity metrics documentation

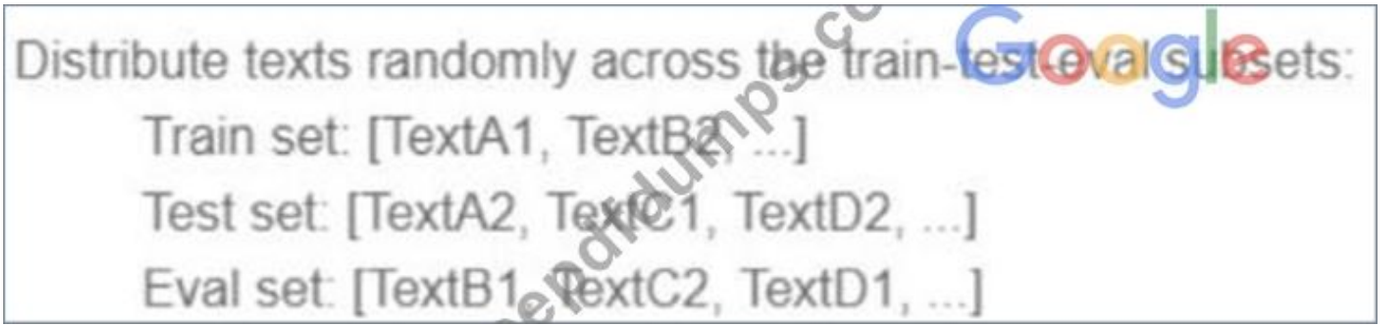
NEW QUESTION: 127
You are developing an ML pipeline using Vertex AI Pipelines. You want your pipeline to upload a new version of the XGBoost model to Vertex AI Model Registry and deploy it to Vertex AI End points for online inference. You want to use the simplest approach. What should you do?
A. Use the Vertex AI REST API within a custom component based on a vertex-ai/prediction/xgboost-cpu image.
B. Use the Vertex AI ModelEvaluationOp component to evaluate the model.
C. Use the Vertex AI SDK for Python within a custom component based on a python: 3.10 Image.
D. Chain the Vertex AI ModelUploadOp and ModelDeployop components together.
Answer: (SHOW ANSWER)
According to the web search results, Vertex AI Pipelines is a serverless orchestrator for running ML pipelines, using either the KFP SDK or TFX1. Vertex AI Pipelines provides a set of prebuilt components that can be used to perform common ML tasks, such as training, evaluation, deployment, and more2. Vertex AI ModelUploadOp and ModelDeployOp are two such components that can be used to upload a new version of the XGBoost model to Vertex AI Model Registry and deploy it to Vertex AI Endpoints for online inference3. Therefore, option D is the best way to use the simplest approach for the given use case, as it only requires chaining two prebuilt components together. The other options are not relevant or optimal for this scenario. Reference:
Vertex AI Pipelines
Google Cloud Pipeline Components
Vertex AI ModelUploadOp and ModelDeployOp
Google Professional Machine Learning Certification Exam 2023
Latest Google Professional Machine Learning Engineer Actual Free Exam Questions

NEW QUESTION: 128

Your team is working on an NLP research project to predict political affiliation of authors based on articles they have written. You have a large training dataset that is structured like this:

You followed the standard 80%-10%-10% data distribution across the training, testing, and evaluation subsets. How should you distribute the training examples across the train-test-eval subsets while maintaining the 80-10-10 proportion?

A.



B.

C.

Distribute sentences randomly across the train-test-eval subsets:

Train set: [SentenceA11, SentenceA21, Sentence B11, SentenceB21, SentenceC11, SentenceD21, ...]
Test set: [SentenceA12, SentenceA22, Sentence B12, SentenceC22, SentenceC12, SentenceD22, ...]
Eval set: [SentenceA13, SentenceA23, Sentence B13, SentenceC23, SentenceC13, SentenceD31, ...]

D.

Distribute paragraphs of texts (i.e., chunks of consecutive sentences) across the train-test-eval subsets:

Train set: [SentenceA11, SentenceA12, Sentence D11, SentenceD12, ...]
Test set: [SentenceA13, SentenceB13, Sentence B21, SentenceD23, SentenceC12, SentenceD13, ...]
Eval set: [SentenceA11, SentenceA22, Sentence B13, SentenceD22, SentenceC23, SentenceD11, ...]

Answer: C (LEAVE A REPLY)

The best way to distribute the training examples across the train-test-eval subsets while maintaining the 80-10-10 proportion is to use option C. This option ensures that each subset contains a balanced and representative sample of the different classes (Democrat and Republican) and the different authors. This way, the model can learn from a diverse and comprehensive set of articles and avoid overfitting or underfitting. Option C also avoids the problem of data leakage, which occurs when the same author appears in more than one subset, potentially biasing the model and inflating its performance. Therefore, option C is the most suitable technique for this use case.

NEW QUESTION: 129

You developed a BigQuery ML linear regressor model by using a training dataset stored in a BigQuery table. New data is added to the table every minute. You are using Cloud Scheduler and Vertex AI Pipelines to automate hourly model training, and use the model for direct inference. The feature preprocessing logic includes quantile bucketization and MinMax scaling on data received in the last hour. You want to minimize storage and computational overhead. What should you do?

A. Create a component in the Vertex AI Pipelines directed acyclic graph (DAG) to calculate the required statistics, and pass the statistics on to subsequent components.

B. Preprocess and stage the data in BigQuery prior to feeding it to the model during training and inference.

C. Create SQL queries to calculate and store the required statistics in separate BigQuery tables that are referenced in the CREATE MODEL statement.

D. Use the TRANSFORM clause in the CREATE MODEL statement in the SQL query to calculate the required statistics.

Answer: (SHOW ANSWER)

The best option to minimize storage and computational overhead is to use the TRANSFORM clause in the CREATE MODEL statement in the SQL query to calculate the required statistics. The TRANSFORM clause allows you to specify feature preprocessing logic that applies to both training and prediction. The preprocessing logic is executed in the same query as the model creation, which avoids the need to create and store intermediate tables. The TRANSFORM clause also supports quantile bucketization and MinMax scaling, which are the preprocessing steps required for this scenario. Option A is incorrect because creating a component in the Vertex

AI Pipelines DAG to calculate the required statistics may increase the computational overhead, as the component needs to run separately from the model creation. Moreover, the component needs to pass the statistics to subsequent components, which may increase the storage overhead. Option B is incorrect because preprocessing and staging the data in BigQuery prior to feeding it to the model may also increase the storage and computational overhead, as you need to create and maintain additional tables for the preprocessed data. Moreover, you need to ensure that the preprocessing logic is consistent for both training and inference. Option C is incorrect because creating SQL queries to calculate and store the required statistics in separate BigQuery tables may also increase the storage and computational overhead, as you need to create and maintain additional tables for the statistics. Moreover, you need to ensure that the statistics are updated regularly to reflect the new data. Reference:
BigQuery ML documentation
Using the TRANSFORM clause
Feature preprocessing with BigQuery ML

NEW QUESTION: 130
You work for a company that provides an anti-spam service that flags and hides spam posts on social media platforms. Your company currently uses a list of 200,000 keywords to identify suspected spam posts. If a post contains more than a few of these keywords, the post is identified as spam. You want to start using machine learning to flag spam posts for human review. What is the main advantage of implementing machine learning for this business case?
A. Posts can be compared to the keyword list much more quickly.
B. New problematic phrases can be identified in spam posts.
C. A much longer keyword list can be used to flag spam posts.
D. Spam posts can be flagged using far fewer keywords.
Answer: (SHOW ANSWER)
The main advantage of implementing machine learning for this business case is that new problematic phrases can be identified in spam posts. This is because machine learning can learn from the data and the feedback, and adapt to the changing patterns and trends of spam posts. Machine learning can also capture the semantic and contextual meaning of the posts, and not just rely on the presence or absence of keywords. By using machine learning, you can improve the accuracy and coverage of your anti-spam service, and detect new and emerging types of spam posts that may not be captured by the keyword list.
The other options are not advantages of implementing machine learning for this business case for the following reasons:
A . Posts can be compared to the keyword list much more quickly is not an advantage, as it does not improve the quality or effectiveness of the anti-spam service. It only improves the efficiency of the service, which is not the primary objective. Moreover, machine learning may not necessarily be faster than the keyword list, depending on the complexity and size of the model and the data.
C . A much longer keyword list can be used to flag spam posts is not an advantage, as it does not address the limitations or challenges of the keyword list approach. It only increases the size and

complexity of the keyword list, which can make it harder to maintain and update. Moreover, a longer keyword list may not improve the accuracy or coverage of the anti-spam service, as it may introduce more false positives or false negatives, or miss new and emerging types of spam posts.
D . Spam posts can be flagged using far fewer keywords is not an advantage, as it does not reflect the capabilities or benefits of machine learning. It only reduces the size and complexity of the keyword list, which can make it easier to maintain and update. However, using fewer keywords may not improve the accuracy or coverage of the anti-spam service, as it may lose some information or meaning of the posts, or miss some types of spam posts.
Reference:
Professional ML Engineer Exam Guide
Preparing for Google Cloud Certification: Machine Learning Engineer Professional Certificate
Google Cloud launches machine learning engineer certification Machine Learning for Spam Detection Spam Detection Using Machine Learning

NEW QUESTION: 131
You are building an ML model to detect anomalies in real-time sensor dat a. You will use Pub/Sub to handle incoming requests. You want to store the results for analytics and visualization. How should you configure the pipeline?

A. 1 Dataflow, 2 - AI Platform, 3 BigQuery
B. 1 DataProc, 2 AutoML, 3 Cloud Bigtable
C. 1 BigQuery, 2 AutoML, 3 Cloud Functions
D. 1 BigQuery, 2 AI Platform, 3 Cloud Storage
Answer: A (LEAVE A REPLY)
Dataflow is a fully managed service for executing Apache Beam pipelines that can process streaming or batch data1.
AI Platform is a unified platform that enables you to build and run machine learning applications across Google Cloud2.
BigQuery is a serverless, highly scalable, and cost-effective cloud data warehouse designed for business agility3.
These services are suitable for building an ML model to detect anomalies in real-time sensor data, as they can handle large-scale data ingestion, preprocessing, training, serving, storage, and visualization. The other options are not as suitable because:
DataProc is a service for running Apache Spark and Apache Hadoop clusters, which are not optimized for streaming data processing4.

AutoML is a suite of machine learning products that enables developers with limited machine learning expertise to train high-quality models specific to their business needs5. However, it does not support custom models or real-time predictions.
Cloud Bigtable is a scalable, fully managed NoSQL database service for large analytical and operational workloads. However, it is not designed for ad hoc queries or interactive analysis.
Cloud Functions is a serverless execution environment for building and connecting cloud services. However, it is not suitable for storing or visualizing data.
Cloud Storage is a service for storing and accessing data on Google Cloud. However, it is not a data warehouse and does not support SQL queries or visualization tools.

NEW QUESTION: 132
You need to design an architecture that serves asynchronous predictions to determine whether a particular mission-critical machine part will fail. Your system collects data from multiple sensors from the machine. You want to build a model that will predict a failure in the next N minutes, given the average of each sensor's data from the past 12 hours. How should you design the architecture?
A. 1. HTTP requests are sent by the sensors to your ML model, which is deployed as a microservice and exposes a REST API for prediction
2. Your application queries a Vertex AI endpoint where you deployed your model.
3. Responses are received by the caller application as soon as the model produces the prediction.
B. 1. Events are sent by the sensors to Pub/Sub, consumed in real time, and processed by a Dataflow stream processing pipeline.
2. The pipeline invokes the model for prediction and sends the predictions to another Pub/Sub topic.
3. Pub/Sub messages containing predictions are then consumed by a downstream system for monitoring.
C. 1. Export your data to Cloud Storage using Dataflow.
2. Submit a Vertex AI batch prediction job that uses your trained model in Cloud Storage to perform scoring on the preprocessed data.
3. Export the batch prediction job outputs from Cloud Storage and import them into Cloud SQL.
D. 1. Export the data to Cloud Storage using the BigQuery command-line tool
2. Submit a Vertex AI batch prediction job that uses your trained model in Cloud Storage to perform scoring on the preprocessed data.
3. Export the batch prediction job outputs from Cloud Storage and import them into BigQuery.
Answer: B (LEAVE A REPLY)
Reasoning: The question asks for a design that serves asynchronous predictions to determine whether a machine part will fail. This means that the predictions do not need to be returned immediately to the sensors, but can be processed in batches and sent to a downstream system for monitoring. Option B is the only one that uses a streaming data pipeline with Pub/Sub and Dataflow, which can handle real-time data ingestion, processing, and prediction. Option B also

invokes the model for prediction, which is required by the question. The other options either use synchronous predictions (option A), batch predictions (options C and D), or do not invoke the model for prediction (option D).

NEW QUESTION: 133
You are an ML engineer in the contact center of a large enterprise. You need to build a sentiment analysis tool that predicts customer sentiment from recorded phone conversations. You need to identify the best approach to building a model while ensuring that the gender, age, and cultural differences of the customers who called the contact center do not impact any stage of the model development pipeline and results. What should you do?
A. Extract sentiment directly from the voice recordings
B. Convert the speech to text and build a model based on the words
C. Convert the speech to text and extract sentiments based on the sentences
D. Convert the speech to text and extract sentiment using syntactical analysis
Answer: (SHOW ANSWER)
Sentiment analysis is the process of identifying and extracting the emotions, opinions, and attitudes expressed in a text or speech. Sentiment analysis can help businesses understand their customers' feedback, satisfaction, and preferences. There are different approaches to building a sentiment analysis tool, depending on the input data and the output format. Some of the common approaches are:
Extracting sentiment directly from the voice recordings: This approach involves using acoustic features, such as pitch, intensity, and prosody, to infer the sentiment of the speaker. This approach can capture the nuances and subtleties of the vocal expression, but it also requires a large and diverse dataset of labeled voice recordings, which may not be easily available or accessible. Moreover, this approach may not account for the semantic and contextual information of the speech, which can also affect the sentiment.
Converting the speech to text and building a model based on the words: This approach involves using automatic speech recognition (ASR) to transcribe the voice recordings into text, and then using lexical features, such as word frequency, polarity, and valence, to infer the sentiment of the text. This approach can leverage the existing text-based sentiment analysis models and tools, but it also introduces some challenges, such as the accuracy and reliability of the ASR system, the ambiguity and variability of the natural language, and the loss of the acoustic information of the speech.
Converting the speech to text and extracting sentiments based on the sentences: This approach involves using ASR to transcribe the voice recordings into text, and then using syntactic and semantic features, such as sentence structure, word order, and meaning, to infer the sentiment of the text. This approach can capture the higher-level and complex aspects of the natural language, such as negation, sarcasm, and irony, which can affect the sentiment. However, this approach also requires more sophisticated and advanced natural language processing techniques, such as parsing, dependency analysis, and semantic role labeling, which may not be readily available or easy to implement.

Converting the speech to text and extracting sentiment using syntactical analysis: This approach involves using ASR to transcribe the voice recordings into text, and then using syntactical analysis, such as part-of-speech tagging, phrase chunking, and constituency parsing, to infer the sentiment of the text. This approach can identify the grammatical and structural elements of the natural language, such as nouns, verbs, adjectives, and clauses, which can indicate the sentiment. However, this approach may not account for the pragmatic and contextual information of the speech, such as the speaker's intention, tone, and situation, which can also influence the sentiment.
For the use case of building a sentiment analysis tool that predicts customer sentiment from recorded phone conversations, the best approach is to convert the speech to text and extract sentiments based on the sentences. This approach can balance the trade-offs between the accuracy, complexity, and feasibility of the sentiment analysis tool, while ensuring that the gender, age, and cultural differences of the customers who called the contact center do not impact any stage of the model development pipeline and results. This approach can also handle different types and levels of sentiment, such as polarity (positive, negative, or neutral), intensity (strong or weak), and emotion (anger, joy, sadness, etc.). Therefore, converting the speech to text and extracting sentiments based on the sentences is the best approach for this use case.

NEW QUESTION: 134
You have created multiple versions of an ML model and have imported them to Vertex AI Model Registry. You want to perform A/B testing to identify the best-performing model using the simplest approach. What should you do?
A. Split incoming traffic among separate Cloud Run instances of deployed models. Monitor the performance of each version using Cloud Monitoring.
B. Split incoming traffic to distribute prediction requests among the versions. Monitor the performance of each version using Looker Studio dashboards that compare logged data for each version.
C. Split incoming traffic among Google Kubernetes Engine (GKE) clusters and use Traffic Director to distribute prediction requests to different versions. Monitor the performance of each version using Cloud Monitoring.
D. Split incoming traffic to distribute prediction requests among the versions. Monitor the performance of each version using Vertex AI's built-in monitoring tools.
Answer: D (LEAVE A REPLY)
Vertex AI Model Registry supports traffic splitting and built-in monitoring, making A/B testing seamless. This approach eliminates the need for additional monitoring tools and infrastructure overhead. Cloud Run and GKE solutions (Options A and C) add unnecessary complexity, while Looker Studio (Option B) requires additional configuration for monitoring.

NEW QUESTION: 135

You are an ML engineer at a global shoe store. You manage the ML models for the company's website. You are asked to build a model that will recommend new products to the user based on their purchase behavior and similarity with other users. What should you do?

A. Build a classification model

B. Build a knowledge-based filtering model

C. Build a collaborative-based filtering model

D. Build a regression model using the features as predictors

Answer: C (LEAVE A REPLY)

A recommender system is a type of machine learning system that suggests relevant items to users based on their preferences and behavior. Recommender systems are widely used in e-commerce, media, and entertainment industries to enhance user experience and increase revenue1 There are different types of recommender systems that use different filtering methods to generate recommendations. The most common types are:

Content-based filtering: This method uses the features of the items and the users to find the similarity between them. For example, a content-based recommender system for movies may use the genre, director, cast, and ratings of the movies, and the preferences, demographics, and history of the users, to recommend movies that are similar to the ones the user liked before2

Collaborative filtering: This method uses the feedback and ratings of the users to find the similarity between them and the items. For example, a collaborative filtering recommender system for books may use the ratings of the users for different books, and recommend books that are liked by other users who have similar ratings to the target user3 Hybrid method: This method combines content-based and collaborative filtering methods to overcome the limitations of each method and improve the accuracy and diversity of the recommendations. For example, a hybrid recommender system for music may use both the features of the songs and the artists, and the ratings and listening habits of the users, to recommend songs that match the user's taste and preferences4 Deep learning-based: This method uses deep neural networks to learn complex and non-linear patterns from the data and generate recommendations. Deep learning-based recommender systems can handle large-scale and high-dimensional data, and incorporate various types of information, such as text, images, audio, and video. For example, a deep learning-based recommender system for fashion may use the images and descriptions of the products, and the profiles and feedback of the users, to recommend products that suit the user's style and preferences.

For the use case of building a model that will recommend new products to the user based on their purchase behavior and similarity with other users, the best option is to build a collaborative-based filtering model. This is because collaborative filtering can leverage the implicit feedback and ratings of the users to find the items that are most likely to interest them. Collaborative filtering can also help discover new products that the user may not be aware of, and increase the diversity and serendipity of the recommendations3 The other options are not as suitable for this use case. Building a classification model or a regression model using the features as predictors is not a good idea, as these models are not designed for recommendation tasks, and may not capture the preferences and behavior of the users. Building a knowledge-based filtering model is not relevant,

as this method uses the explicit knowledge and requirements of the users to find the items that meet their criteria, and does not rely on the purchase behavior or similarity with other users.

NEW QUESTION: 136
You are a lead ML engineer at a retail company. You want to track and manage ML metadata in a centralized way so that your team can have reproducible experiments by generating artifacts. Which management solution should you recommend to your team?
A. Store your tf.logging data in BigQuery.
B. Manage all relational entities in the Hive Metastore.
C. Store all ML metadata in Google Cloud's operations suite.
D. Manage your ML workflows with Vertex ML Metadata.
Answer: (SHOW ANSWER)
Vertex ML Metadata is a service that lets you track and manage the metadata produced by your ML workflows in a centralized way. It helps you have reproducible experiments by generating artifacts that represent the data, parameters, and metrics used or produced by your ML system. You can also analyze the lineage and performance of your ML artifacts using Vertex ML Metadata.
Some of the benefits of using Vertex ML Metadata are:
It captures your ML system's metadata as a graph, where artifacts and executions are nodes, and events are edges that link them as inputs or outputs.
It allows you to create contexts to group sets of artifacts and executions together, such as experiments, runs, or projects.
It supports querying and filtering the metadata using the Vertex AI SDK for Python or REST commands.
It integrates with other Vertex AI services, such as Vertex AI Pipelines and Vertex AI Experiments, to automatically log metadata and artifacts.
The other options are not suitable for tracking and managing ML metadata in a centralized way.
Option A: Storing your tf.logging data in BigQuery is not enough to capture the full metadata of your ML system, such as the artifacts and their lineage. BigQuery is a data warehouse service that is mainly used for analytics and reporting, not for metadata management.
Option B: Managing all relational entities in the Hive Metastore is not a good solution for ML metadata, as it is designed for storing metadata of Hive tables and partitions, not for ML artifacts and executions. Hive Metastore is a component of the Apache Hive project, which is a data warehouse system for querying and analyzing large datasets stored in Hadoop.
Option C: Storing all ML metadata in Google Cloud's operations suite is not a feasible option, as it is a set of tools for monitoring, logging, tracing, and debugging your applications and infrastructure, not for ML metadata. Google Cloud's operations suite does not provide the features and integrations that Vertex ML Metadata offers for ML workflows.

NEW QUESTION: 137

You work for a retail company. You have a managed tabular dataset in Vertex AI that contains sales data from three different stores. The dataset includes several features such as store name and sale timestamp. You want to use the data to train a model that makes sales predictions for a new store that will open soon You need to split the data between the training, validation, and test sets What approach should you use to split the data?

A. Use Vertex AI manual split, using the store name feature to assign one store for each set.

B. Use Vertex AI default data split.

C. Use Vertex AI chronological split and specify the sales timestamp feature as the time vanable.

D. Use Vertex AI random split assigning 70% of the rows to the training set, 10% to the validation set, and 20% to the test set.

Answer: (SHOW ANSWER)

The best option for splitting the data between the training, validation, and test sets, using a managed tabular dataset in Vertex AI that contains sales data from three different stores, is to use Vertex AI default data split. This option allows you to leverage the power and simplicity of Vertex AI to automatically and randomly split your data into the three sets by percentage. Vertex AI is a unified platform for building and deploying machine learning solutions on Google Cloud. Vertex AI can support various types of models, such as linear regression, logistic regression, k-means clustering, matrix factorization, and deep neural networks. Vertex AI can also provide various tools and services for data analysis, model development, model deployment, model monitoring, and model governance. A default data split is a data split method that is provided by Vertex AI, and does not require any user input or configuration. A default data split can help you split your data into the training, validation, and test sets by using a random sampling method, and assign a fixed percentage of the data to each set. A default data split can help you simplify the data split process, and works well in most cases. A training set is a subset of the data that is used to train the model, and adjust the model parameters. A training set can help you learn the relationship between the input features and the target variable, and optimize the model performance. A validation set is a subset of the data that is used to validate the model, and tune the model hyperparameters. A validation set can help you evaluate the model performance on unseen data, and avoid overfitting or underfitting. A test set is a subset of the data that is used to test the model, and provide the final evaluation metrics. A test set can help you assess the model

performance on new data, and measure the generalization ability of the model. By using Vertex AI default data split, you can split your data into the training, validation, and test sets by using a random sampling method, and assign the following percentages of the data to each set1:

Set	Text	Image	Video
Training	80%	80%	80%
Validation	10%	10%	N/A
Test	10%	10%	20%

The other options are not as good as option B, for the following reasons:
Option A: Using Vertex AI manual split, using the store name feature to assign one store for each set would not allow you to split your data into representative and balanced sets, and could cause errors or poor performance. A manual split is a data split method that allows you to control how your data is split into sets, by using the ml_use label or the data filter expression. A manual split can help you customize the data split logic, and handle complex or non-standard data formats. A store name feature is a feature that indicates the name of the store where the sales data was collected. A store name feature can help you identify the source of the data, and group the data by store. However, using Vertex AI manual split, using the store name feature to assign one store for each set would not allow you to split your data into representative and balanced sets, and could cause errors or poor performance. You would need to write code, create and configure the ml_use label or the data filter expression, and assign one store for each set. Moreover, this option would not ensure that the data in each set has the same distribution and characteristics as the data in the whole dataset, which could prevent you from learning the general pattern of the data, and cause bias or variance in the model2.
Option C: Using Vertex AI chronological split and specifying the sales timestamp feature as the time variable would not allow you to split your data into representative and balanced sets, and could cause errors or poor performance. A chronological split is a data split method that allows you to split your data into sets based on the order of the data. A chronological split can help you preserve the temporal dependency and sequence of the data, and avoid data leakage. A sales timestamp feature is a feature that indicates the date and time when the sales data was collected. A sales timestamp feature can help you track the changes and trends of the data over time, and capture the seasonality and cyclicality of the data. However, using Vertex AI chronological split and specifying the sales timestamp feature as the time variable would not allow you to split your data into representative and balanced sets, and could cause errors or poor performance. You would need to write code, create and configure the time variable, and split the data by the order of the time variable. Moreover, this option would not ensure that the data in each set has the same distribution and characteristics as the data in the whole dataset, which could prevent you from learning the general pattern of the data, and cause bias or variance in the model3.
Option D: Using Vertex AI random split, assigning 70% of the rows to the training set, 10% to the validation set, and 20% to the test set would not allow you to use the default data split method that is provided by Vertex AI, and could increase the complexity and cost of the data split

process. A random split is a data split method that allows you to split your data into sets by using a random sampling method, and assign a custom percentage of the data to each set. A random split can help you split your data into representative and balanced sets, and avoid data leakage. However, using Vertex AI random split, assigning 70% of the rows to the training set, 10% to the validation set, and 20% to the test set would not allow you to use the default data split method that is provided by Vertex AI, and could increase the complexity and cost of the data split process. You would need to write code, create and configure the random split method, and assign the custom percentages to each set. Moreover, this option would not use the default data split method that is provided by Vertex AI, which can simplify the data split process, and works well in most cases1.
Reference:
About data splits for AutoML models | Vertex AI | Google Cloud
Manual split for unstructured data
Mathematical split

NEW QUESTION: 138
While running a model training pipeline on Vertex AI, you discover that the evaluation step is failing because of an out-of-memory error. You are currently using TensorFlow Model Analysis (TFMA) with a standard Evaluator TensorFlow Extended (TFX) pipeline component for the evaluation step. You want to stabilize the pipeline without downgrading the evaluation quality while minimizing infrastructure overhead. What should you do?
A. Add tfma.MetricsSpec () to limit the number of metrics in the evaluation step.
B. Migrate your pipeline to Kubeflow hosted on Google Kubernetes Engine, and specify the appropriate node parameters for the evaluation step.
C. Include the flag -runnerDataflowRunner in beam_pipeline_args to run the evaluation step on Dataflow.
D. Move the evaluation step out of your pipeline and run it on custom Compute Engine VMs with sufficient memory.
Answer: (SHOW ANSWER)
The best option to stabilize the pipeline without downgrading the evaluation quality while minimizing infrastructure overhead is to use Dataflow as the runner for the evaluation step. Dataflow is a fully managed service for executing Apache Beam pipelines that can scale up and down according to the workload. Dataflow can handle large-scale, distributed data processing tasks such as model evaluation, and it can also integrate with Vertex AI Pipelines and TensorFlow Extended (TFX). By using the flag -runnerDataflowRunner in beam_pipeline_args, you can instruct the Evaluator component to run the evaluation step on Dataflow, instead of using the default DirectRunner, which runs locally and may cause out-of-memory errors. Option A is incorrect because adding tfma.MetricsSpec() to limit the number of metrics in the evaluation step may downgrade the evaluation quality, as some important metrics may be omitted. Moreover, reducing the number of metrics may not solve the out-of-memory error, as the evaluation step may still consume a lot of memory depending on the size and complexity of the data and the

model. Option B is incorrect because migrating the pipeline to Kubeflow hosted on Google Kubernetes Engine (GKE) may increase the infrastructure overhead, as you need to provision, manage, and monitor the GKE cluster yourself. Moreover, you need to specify the appropriate node parameters for the evaluation step, which may require trial and error to find the optimal configuration. Option D is incorrect because moving the evaluation step out of the pipeline and running it on custom Compute Engine VMs may also increase the infrastructure overhead, as you need to create, configure, and delete the VMs yourself. Moreover, you need to ensure that the VMs have sufficient memory for the evaluation step, which may require trial and error to find the optimal machine type. Reference:
Dataflow documentation
Using DataflowRunner
Evaluator component documentation
Configuring the Evaluator component