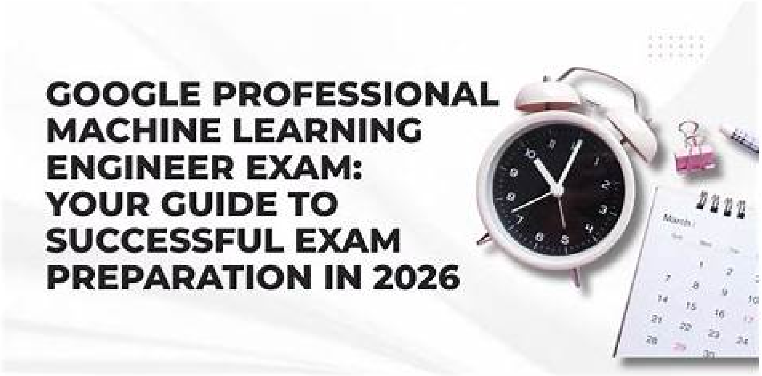
Latest Professional-Machine-Learning-Engineer Exam Preparation & Valid Professional-Machine-Learning-Engineer Dumps



BONUS!!! Download part of TestsDumps Professional-Machine-Learning-Engineer dumps for free: https://drive.google.com/open?id=1YHKEXmldRuY9NdKdjUqv1S3sJZUzC9OZ

For candidates who want to obtain the certification for Professional-Machine-Learning-Engineer exam, passing the exam is necessary. We will help you pass the exam just one time. Professional-Machine-Learning-Engineer training materials are high-quality, since we have experienced experts who are quite familiar with exam center to compile and verify the exam dumps. In addition, we offer you free update for 365 days after payment, and the latest version for Professional-Machine-Learning-Engineer Training Materials will be sent to your email automatically. We have online and offline chat service and if you have any questions for Professional-Machine-Learning-Engineer exam materials, you can have a chat with us.

The third and last format is the Professional-Machine-Learning-Engineer desktop practice exam software form that can be used without an active internet connection. This software works offline on the Windows operating system. The practice exams benefit your preparation because you can attempt them multiple times to improve yourself for the Google Professional Machine Learning Engineer Professional-Cloud-Developercertification test. Our Professional-Machine-Learning-Engineer Exam Dumps are customizable, so you can set the time and questions according to your needs.

>> Latest Professional-Machine-Learning-Engineer Exam Preparation <<

Valid Professional-Machine-Learning-Engineer Dumps & Exam Professional-Machine-Learning-Engineer Details

TestsDumps free update our training materials, which means you will always get the latest Professional-Machine-Learning-Engineer exam training materials. If Professional-Machine-Learning-Engineer exam objectives change, The learning materials TestsDumps provided will follow the change. TestsDumps know the needs of each candidate, we will help you through your Professional-Machine-Learning-Engineer Exam Certification. We help each candidate to pass the exam with best price and highest quality.

Google Professional Machine Learning Engineer Sample Questions (Q278-Q283):

NEW QUESTION # 278
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. 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.
- 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. 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.
- 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

Explanation:
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 # 279
You work for a bank. You have created a custom model to predict whether a loan application should be flagged for human review. The input features are stored in a BigQuery table. The model is performing well and you plan to deploy it to production. Due to compliance requirements the model must provide explanations for each prediction. You want to add this functionality to your model code with minimal effort and provide explanations that are as accurate as possible What should you do?

- A. Upload the custom model to Vertex Al Model Registry and configure feature-based attribution by using sampled Shapley with input baselines.
- B. Create a BigQuery ML deep neural network model, and use the ML. EXPLAIN_PREDICT method with the num_integral_steps parameter.
- C. Update the custom serving container to include sampled Shapley-based explanations in the prediction outputs.
- D. Create an AutoML tabular model by using the BigQuery data with integrated Vertex Explainable Al.

Answer: A

Explanation:
The best option for adding explanations to your model code with minimal effort and providing explanations that are as accurate as possible is to upload the custom model to Vertex AI Model Registry and configure feature-based attribution by using sampled Shapley with input baselines. This option allows you to leverage the power and simplicity of Vertex Explainable AI to generate feature attributions for each prediction, and understand how each feature contributes to the model output. Vertex Explainable AI is a service that can help you understand and interpret predictions made by your machine learning models, natively integrated with a number of Google's products and services. Vertex Explainable AI can provide feature-based and example-based explanations to provide better understanding of model decision making. Feature-based explanations are explanations that show how much each feature in the input influenced the prediction.
Feature-based explanations can help you debug and improve model performance, build confidence in the predictions, and understand when and why things go wrong. Vertex Explainable AI supports various feature attribution methods, such as sampled Shapley, integrated gradients, and XRAI. Sampled Shapley is a feature attribution method that is based on the Shapley value, which is a concept from game theory that measures how much each player in a cooperative game contributes to the total payoff. Sampled Shapley approximates the Shapley value for each feature by sampling different subsets of features, and computing the marginal contribution of each feature to the prediction. Sampled Shapley can provide accurate and consistent feature attributions, but it can also be computationally expensive. To reduce the computation cost, you can use input baselines, which are reference inputs that are used to compare with the actual inputs. Input baselines can help you define the starting point or the default state of the features, and calculate the feature attributions relative to the input baselines. By uploading the custom model to Vertex AI Model Registry and configuring feature-based attribution by using sampled Shapley with input baselines, you can add explanations to your model code with minimal effort and provide explanations that are as accurate as possible1.
The other options are not as good as option C, for the following reasons:
* Option A: Creating an AutoML tabular model by using the BigQuery data with integrated Vertex Explainable AI would require more skills and steps than uploading the custom model to Vertex AI Model Registry and configuring feature-based attribution by

using sampled Shapley with input baselines. AutoML tabular is a service that can automatically build and train machine learning models for structured or tabular data. AutoML tabular can use BigQuery as the data source, and provide feature-based explanations by using integratedgradients as the feature attribution method. However, creating an AutoML tabular model by using the BigQuery data with integrated Vertex Explainable AI would require more skills and steps than uploading the custom model to Vertex AI Model Registry and configuring feature-based attribution by using sampled Shapley with input baselines. You would need to create a new AutoML tabular model, import the BigQuery data, configure the model settings, train and evaluate the model, and deploy the model. Moreover, this option would not use your existing custom model, which is already performing well, but create a new model, which may not have the same performance or behavior as your custom model2.

* Option B: Creating a BigQuery ML deep neural network model, and using the ML.EXPLAIN_PREDICT method with the num_integral_steps parameter would not allow you to deploy the model to production, and could provide less accurate explanations than using sampled Shapley with input baselines. BigQuery ML is a service that can create and train machine learning models by using SQL queries on BigQuery. BigQuery ML can create a deep neural network model, which is a type of machine learning model that consists of multiple layers of neurons, and can learn complex patterns and relationships from the data. BigQuery ML can also provide feature-based explanations by using the ML.EXPLAIN_PREDICT method, which is a SQL function that returns the feature attributions for each prediction. The ML.EXPLAIN_PREDICT method uses integrated gradients as the feature attribution method, which is a method that calculates the average gradient of the prediction output with respect to the feature values along the path from the input baseline to the input. The num_integral_steps parameter is a parameter that determines the number of steps along the path from the input baseline to the input. However, creating a BigQuery ML deep neural network model, and using the ML.EXPLAIN_PREDICT method with the num_integral_steps parameter would not allow you to deploy the model to production, and could provide less accurate explanations than using sampled Shapley with input baselines. BigQuery ML does not support deploying the model to Vertex AI

* Endpoints, which is a service that can provide low-latency predictions for individual instances.

BigQuery ML only supports batch prediction, which is a service that can provide high-throughput predictions for a large batch of instances. Moreover, integrated gradients can provide less accurate and consistent explanations than sampled Shapley, as integrated gradients can be sensitive to the choice of the input baseline and the num_integral_steps parameter3.

* Option D: Updating the custom serving container to include sampled Shapley-based explanations in the prediction outputs would require more skills and steps than uploading the custom model to Vertex AI Model Registry and configuring feature-based attribution by using sampled Shapley with input baselines. A custom serving container is a container image that contains the model, the dependencies, and a web server. A custom serving container can help you customize the prediction behavior of your model, and handle complex or non-standard data formats. However, updating the custom serving container to include sampled Shapley-based explanations in the prediction outputs would require more skills and steps than uploading the custom model to Vertex AI Model Registry and configuring feature-based attribution by using sampled Shapley with input baselines. You would need to write code, implement the sampled Shapley algorithm, build and test the container image, and upload and deploy the container image. Moreover, this option would not leverage the power and simplicity of Vertex Explainable AI, which can provide feature-based explanations natively integrated with Vertex AI services4.

References:

* 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

* Vertex Explainable AI

* AutoML Tables

* BigQuery ML

* Using custom containers for prediction

NEW QUESTION # 280

You are training a Resnet model on Al 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. Set the prefetch option equal to the training batch size
- B. Decrease the batch size argument in your transformation
- C. Increase the buffer size for the shuffle option.
- D. Reduce the value of the repeat parameter
- E. Use the interleave option for reading data

Answer: A,B

Explanation:
https://towardsdatascience.com/overcoming-data-preprocessing-bottlenecks-with-tensorflow-data-service-nvidia-dali-and-other-d6321917f851

NEW QUESTION # 281
You have developed a BigQuery ML model that predicts customer churn and deployed the model to Vertex Al Endpoints. You want to automate the retraining of your model by using minimal additional code when model feature values change. You also want to minimize the number of times that your model is retrained to reduce training costs. What should you do?

- A. 1. Create a Vertex Al Model Monitoring job configured to monitor training/serving skew
  2. Configure alert monitoring to publish a message to a Pub/Sub queue when a monitoring alert is detected
  3. Use a Cloud Function to monitor the Pub/Sub queue, and trigger retraining in BigQuery.
- B. 1. Enable request-response logging on Vertex Al Endpoints.
  2 Schedule a TensorFlow Data Validation job to monitor prediction drift
  3. Execute model retraining if there is significant distance between the distributions.
- C. 1. Enable request-response logging on Vertex Al Endpoints
  2. Schedule a TensorFlow Data Validation job to monitor training/serving skew
  3. Execute model retraining if there is significant distance between the distributions
- D. 1 Create a Vertex Al Model Monitoring job configured to monitor prediction drift.
  2. Configure alert monitoring to publish a message to a Pub/Sub queue when a monitonng alert is detected.
  3. Use a Cloud Function to monitor the Pub/Sub queue, and trigger retraining in BigQuery

Answer: D

Explanation:
The best option for automating the retraining of your model by using minimal additional code when model feature values change, and minimizing the number of times that your model is retrained to reduce training costs, is to create a Vertex AI Model Monitoring job configured to monitor prediction drift, configure alert monitoring to publish a message to a Pub/Sub queue when a monitoring alert is detected, and use a Cloud Function to monitor the Pub/Sub queue, and trigger retraining in BigQuery. This option allows you to leverage the power and simplicity of Vertex AI, Pub/Sub, and Cloud Functions to monitor your model performance and retrain your model when needed. 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 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. Prediction drift is a type of model monitoring metric that measures the difference between the distributions of the predictions generated by the model on the training data and the predictions generated by the model on the online data. Prediction drift can indicate that the model performance is degrading, or that the online data is changing over time. By creating a Vertex AI Model Monitoring job configured to monitor prediction drift, you can track the changes in the model predictions, and compare them with the expected predictions. Alert monitoring is a feature of Vertex AI Model Monitoring that can notify you when a monitoring metric exceeds a predefined threshold. Alert monitoring can help you set up rules and conditions for triggering alerts, and choose the notification channel for receiving alerts. Pub/Sub is a service that can provide reliable and scalable messaging and event streaming on Google Cloud. Pub/Sub can help you publish and subscribe to messages, and deliver them to various Google Cloud services, such as Cloud Functions. A Pub/Sub queue is a resource that can hold messages that are published to a Pub/Sub topic. A Pub/Sub queue can help you store and manage messages, and ensure that they are delivered to the subscribers. By configuring alert monitoring to publish a message to a Pub/Sub queue when a monitoring alert is detected, you can send a notification to a Pub/Sub topic, and trigger a downstream action based on the alert. Cloud Functions is a service that can run your stateless code in response to events on Google Cloud. Cloud Functions can help you create and execute functions without provisioning or managing servers, and pay only for the resources you use. A Cloud Function is a resource that can execute a piece of code in response to an event, such as a Pub/Sub message. A Cloud Function can help you perform various tasks, such as data processing, data transformation, or data analysis. BigQuery is a service that can store and query large-scale data on Google Cloud. BigQuery can help you analyze your data by using SQL queries, and perform various tasks, such as data exploration, data transformation, or data visualization. BigQuery ML is a feature of BigQuery that can create and execute machine learning models in BigQuery by using SQL queries.
BigQuery ML can help you build and train various types of models,such as linear regression, logistic regression, k-means clustering, matrix factorization, and deep neural networks. By using a Cloud Function to monitor the Pub/Sub queue, and trigger retraining in BigQuery, you can automate the retraining of your model by using minimal additional code when model feature values change. You can write a Cloud Function that listens to the Pub/Sub queue, and executes a SQL query to retrain your model in BigQuery ML when a prediction drift alert is received. By retraining your model in BigQuery ML, you can update your model parameters and improve your model performance and accuracy1.

The other options are not as good as option C, for the following reasons:
* Option A: Enabling request-response logging on Vertex AI Endpoints, scheduling a TensorFlow Data Validation job to monitor prediction drift, and executing model retraining if there is significant distance between the distributions would require more skills and steps than creating a Vertex AI Model Monitoring job configured to monitor prediction drift, configuring alert monitoring to publish a message to a Pub/Sub queue when a monitoring alert is detected, and using a Cloud Function to monitor the Pub/Sub queue, and trigger retraining in BigQuery. Request-response logging is a feature of Vertex AI Endpoints that can record the requests and responses that are sent to and from the online prediction endpoint. Request-response logging can help you collect and analyze the online prediction data, and troubleshoot any issues with your model. TensorFlow Data Validation is a tool that can analyze and validate your data for machine learning. TensorFlow Data Validation can help you explore, understand, and clean your data, and detect various data issues, such as data drift, data skew, or data anomalies.
Prediction drift is a type of data issue that measures the difference between the distributions of the
* predictions generated by the model on the training data and the predictions generated by the model on the online data. Prediction drift can indicate that the model performance is degrading, or that the online data is changing over time. By enabling request-response logging on Vertex AI Endpoints, and scheduling a TensorFlow Data Validation job to monitor prediction drift, you can collect and analyze the online prediction data, and compare the distributions of the predictions. However, enabling request-response logging on Vertex AI Endpoints, scheduling a TensorFlow Data Validation job to monitor prediction drift, and executing model retraining if there is significant distance between the distributions would require more skills and steps than creating a Vertex AI Model Monitoring job configured to monitor prediction drift, configuring alert monitoring to publish a message to a Pub/Sub queue when a monitoring alert is detected, and using a Cloud Function to monitor the Pub/Sub queue, and trigger retraining in BigQuery. You would need to write code, enable and configure the request-response logging, create and run the TensorFlow Data Validation job, define and measure the distance between the distributions, and execute the model retraining. Moreover, this option would not automate the retraining of your model, as you would need to manually check the prediction drift and trigger the retraining2.
* Option B: Enabling request-response logging on Vertex AI Endpoints, scheduling a TensorFlow Data Validation job to monitor training/serving skew, and executing model retraining if there is significant distance between the distributions would not help you monitor the changes in the model feature values, and could cause errors or poor performance. Training/serving skew is a type of data issue that measures the difference between the distributions of the features used to train the model and the features used to serve the model. Training/serving skew can indicate that the model is not trained on the representative data, orthat the data is changing over time. By enabling request-response logging on Vertex AI Endpoints, and scheduling a TensorFlow Data Validation job to monitor training/serving skew, you can collect and analyze the online prediction data, and compare the distributions of the features. However, enabling request-response logging on Vertex AI Endpoints, scheduling a TensorFlow Data Validation job to monitor training/serving skew, and executing model retraining if there is significant distance between the distributions would not help you monitor the changes in the model feature values, and could cause errors or poor performance. You would need to write code, enable and configure the request-response logging, create and run the TensorFlow Data Validation job, define and measure the distance between the distributions, and execute the model retraining. Moreover, this option would not monitor the prediction drift, which is a more direct and relevant metric for measuring the model performance and quality2.
* Option D: Creating a Vertex AI Model Monitoring job configured to monitor training/serving skew, configuring alert monitoring to publish a message to a Pub/Sub queue when a monitoring alert is detected, and using a Cloud Function to monitor the Pub/Sub queue, and trigger retraining in BigQuery would not help you monitor the changes in the model feature values, and could cause errors or poor performance. Training/serving skew is a type of data issue that measures the difference between the distributions of the features used to train the model and the features used to serve the model.
Training/serving 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 configured to monitor training/serving skew, you can track the changes in the model features, and compare them with the expected features. However, creating a Vertex AI Model Monitoring job configured to monitor training/serving skew, configuring alert monitoring to publish a message to a Pub/Sub queue when a monitoring alert is detected, and using a Cloud Function to monitor the Pub/Sub queue, and trigger retraining in BigQuery would not help you monitor the changes in the model feature values, and could cause errors or poor performance. You would need to write code, create and configure the Vertex AI Model Monitoring job, configure the alert monitoring, create and configure the Pub/Sub queue, and write a Cloud Function to trigger the retraining. Moreover, this option would not monitor the prediction drift, which is a more direct and relevant metric for measuring the model performance and quality1.
References:
* Preparing for Google Cloud Certification: Machine Learning Engineer, Course 3: Production ML Systems, Week 4: ML Governance
* Google Cloud Professional Machine Learning Engineer Exam Guide, Section 3: Scaling ML models in production

NEW QUESTION # 282
Your data science team needs to rapidly experiment with various features, model architectures, and hyperparameters. They need to track the accuracy metrics for various experiments and use an API to query the metrics over time. What should they use to track and report their experiments while minimizing manual effort?

- A. Use Al Platform Notebooks to execute the experiments. Collect the results in a shared Google Sheets file, and query the results using the Google Sheets API
- B. Use Kubeflow Pipelines to execute the experiments Export the metrics file, and query the results using the Kubeflow Pipelines API.
- C. Use Al Platform Training to execute the experiments Write the accuracy metrics to Cloud Monitoring, and query the results using the Monitoring API.
- D. Use Al Platform Training to execute the experiments Write the accuracy metrics to BigQuery, and query the results using the BigQueryAPI.

Answer: C

Explanation:
AI Platform Training is a service that allows you to run your machine learning experiments on Google Cloud using various features, model architectures, and hyperparameters. You can use AI Platform Training to scale up your experiments, leverage distributed training, and access specialized hardware such as GPUs and TPUs1.
Cloud Monitoring is a service that collects and analyzes metrics, logs, and traces from Google Cloud, AWS, and other sources. You can use Cloud Monitoring to create dashboards, alerts, and reports based on your data2. The Monitoring API is an interface that allows you to programmatically access and manipulate your monitoring data3.
By using AI Platform Training and Cloud Monitoring, you can track and report your experiments while minimizing manual effort. You can write the accuracy metrics from your experiments to Cloud Monitoring using the AI Platform Training Python package4. You can then query the results using the Monitoring API and compare the performance of different experiments. You can also visualize the metrics in the Cloud Console or create custom dashboards and alerts5. Therefore, using AI Platform Training and Cloud Monitoring is the best option for this use case.
References:
* AI Platform Training documentation
* Cloud Monitoring documentation
* Monitoring API overview
* Using Cloud Monitoring with AI Platform Training
* Viewing evaluation metrics

NEW QUESTION # 283
......

We will free provide you part of the exercises of Google Certification Professional-Machine-Learning-Engineer Exam on the Internet to let you try to test our product's quality. After your trail you will find TestsDumps's exercises is the most comprehensive one and is what you want to.

Valid Professional-Machine-Learning-Engineer Dumps: https://www.testsdumps.com/Professional-Machine-Learning-Engineer_real-exam-dumps.html

Hurry to get an international Professional-Machine-Learning-Engineer certificate, The development of our Professional-Machine-Learning-Engineer exam bootcamp come a long way and form three versions right now of great usefulness, which is full of useful knowledge and materials for your exercise and review, No matter you are a green-hand or have little knowledge about Professional-Machine-Learning-Engineer training pdf dumps, You can have a try before you decide to buy our Professional-Machine-Learning-Engineer easy pass test.

Techniques discussed include internal referrals, campus recruiting, Professional-Machine-Learning-Engineer Exam Simulator Fee agencies, newspaper and other print media ads, job fairs, Internet ads, and perhaps one of the least considered, acquisitions.

Google - Updated Latest Professional-Machine-Learning-Engineer Exam Preparation

Chapter One: Nuts and Bolts of Digital Imaging, Hurry to get an international Professional-Machine-Learning-Engineer certificate, The development of our Professional-Machine-Learning-Engineer exam bootcampcome a long way and form three versions right now Professional-Machine-Learning-Engineer of great usefulness, which is full of useful knowledge and materials for your exercise and review.

No matter you are a green-hand or have little knowledge about Professional-Machine-Learning-Engineer training pdf dumps, You can have a try before you decide to buy our Professional-Machine-Learning-Engineer easy pass test.

All details of the Professional-Machine-Learning-Engineer exam questions are developed to aim squarely at improving your chance of success.

What's more, part of that TestsDumps Professional-Machine-Learning-Engineer dumps now are free: https://drive.google.com/open?id=1YHKEXmldRuY9NdKdjUqv1S3sJZUzC9OZ