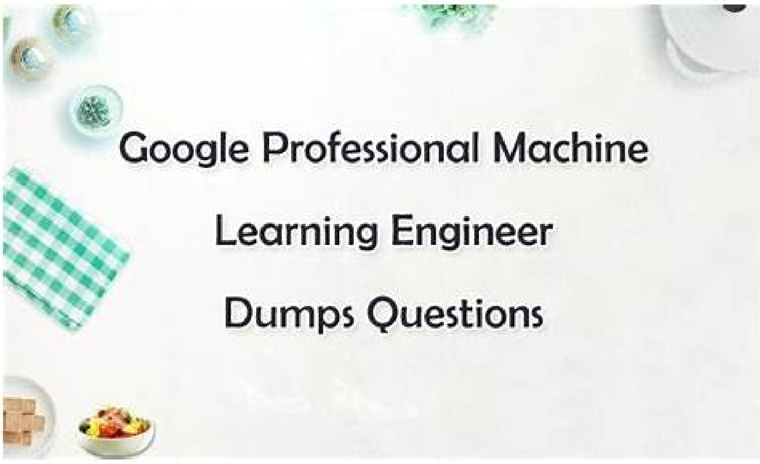
Free PDF Quiz 2026 Google Pass-Sure Professional-Machine-Learning-Engineer Dumps Questions



2026 Latest ExamCost Professional-Machine-Learning-Engineer PDF Dumps and Professional-Machine-Learning-Engineer Exam Engine Free Share: https://drive.google.com/open?id=1GgVllsU_surfHrDATOTE9D5VQ2f_HN5V

Elaborately designed and developed Professional-Machine-Learning-Engineer test guide as well as good learning support services are the key to assisting our customers to realize their dreams. Our Professional-Machine-Learning-Engineer study braindumps have a variety of self-learning and self-assessment functions to detect learners' study outcomes, and the statistical reporting function of our Professional-Machine-Learning-Engineer test guide is designed for students to figure out their weaknesses and tackle the causes, thus seeking out specific methods dealing with them. Our Professional-Machine-Learning-Engineer Exam Guide have also set a series of explanation about the complicated parts certificated by the syllabus and are based on the actual situation to stimulate exam circumstance in order to provide you a high-quality and high-efficiency user experience.

What is the duration, language, and format of Professional Machine Learning Engineer - Google

- Type of Questions: Multiple choice (MCQs), multiple answers
- No negative marking for wrong answers
- Language of Exam: English, Japanese, Korean
- Duration of Exam: 120 minutes

The Google Professional Machine Learning Engineer certification exam is a great way for professionals to demonstrate their knowledge and skills in machine learning. It is also an excellent way for employers to identify and hire qualified candidates with advanced skills in machine learning. Google Professional Machine Learning Engineer certification exam is recognized by Google and is a valuable asset for professionals who want to advance their career in machine learning.

>> Professional-Machine-Learning-Engineer Dumps Questions <<

Make {Useful Study Notes} With Google Professional-Machine-Learning-Engineer PDF Questions

The privacy protection of users is an eternal issue in the internet age. Many illegal websites will sell users' privacy to third parties, resulting in many buyers are reluctant to believe strange websites. But you don't need to worry about it at all when buying our Professional-Machine-Learning-Engineer Learning Engine. We assure you that we will never sell users' information on the Professional-Machine-Learning-Engineer exam questions because it is damaging our own reputation. And we will help you on the Professional-Machine-Learning-Engineer study materials if you have any question.

Google Professional Machine Learning Engineer Certification Exam is a challenging and rewarding test that requires a deep understanding of machine learning principles and their practical application. Professional-Machine-Learning-Engineer Exam is designed to test the candidate's ability to design, implement, and deploy machine learning models using Google Cloud Platform. It covers a broad range of topics, including data preprocessing, model training, hyperparameter tuning, and model evaluation.

Google Professional Machine Learning Engineer Sample Questions (Q159-Q164):

NEW QUESTION # 159
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. 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.
- C. Remove all sensitive data fields, and ask the data science team to build their models using non-sensitive data.
- D. Use principal component analysis (PCA) to reduce the four sensitive fields to one PCA vector.

Answer: B

Explanation:
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.
References:
* 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 # 160

You recently deployed a pipeline in Vertex Al Pipelines that trains and pushes a model to a Vertex Al endpoint to serve real-time traffic. You need to continue experimenting and iterating on your pipeline to improve model performance. You plan to use Cloud Build for CI/CD You want to quickly and easily deploy new pipelines into production and you want to minimize the chance that the new pipeline implementations will break in production. What should you do?

- A. Set up a CI/CD pipeline that builds and tests your source code and then deploys built artifacts into a pre-production environment. After a successful pipeline run in the pre-production environment deploy the pipeline to production
- B. Set up a CI/CD pipeline that builds your source code and then deploys built artifacts into a pre-production environment Run unit tests in the pre-production environment If the tests are successful deploy the pipeline to production.
- C. Set up a CI/CD pipeline that builds and tests your source code If the tests are successful use the Google Cloud console to upload the built container to Artifact Registry and upload the compiled pipeline to Vertex Al Pipelines.
- D. Set up a CI/CD pipeline that builds and tests your source code and then deploys built arrets into a pre-production environment After a successful pipeline run in the pre-production environment, rebuild the source code, and deploy the artifacts to production

Answer: A

Explanation:
The best option for continuing experimenting and iterating on your pipeline to improve model performance, using Cloud Build for CI/CD, and deploying new pipelines into production quickly and easily, is to set up a CI/CD pipeline that builds and tests your source code and then deploys built artifacts into a pre-production environment. After a successful pipeline run in the pre-production environment, deploy the pipeline to production. This option allows you to leverage the power and simplicity of Cloud Build to automate, monitor, and manage your pipeline development and deployment workflow. Cloud Build is a service that can create and run continuous integration and continuous delivery (CI/CD) pipelines on Google Cloud. Cloud Build can build your source code, run unit tests, and deploy built artifacts to various Google Cloud services, such as Vertex AI Pipelines, Vertex AI Endpoints, and Artifact Registry. A CI/CD pipeline is a workflow that can automate the process of building, testing, and deploying software. A CI/CD pipeline can help you improve the quality and reliability of your software, accelerate the development and delivery cycle, and reduce the manual effort and errors. A pre-production environment is an environment that can simulate the production environment, but is isolated from the real users and data. A pre-production environment can help you test and validate your software before deploying it to production, and catch any bugs or issues that may affect the user experience or the system performance. By setting up a CI/CD pipeline that builds and tests your source code and then deploys built artifacts into a pre-production environment, you can ensure that your pipeline code is consistent and error-free, and that your pipeline artifacts are compatible and functional. After a successful pipeline run in the pre-production environment, you can deploy the pipeline to production, which is the environment where your software is accessible and usable by the real users and data. By deploying the pipeline to production after a successful pipeline run in the pre-production environment, you can minimize the chance that the new pipeline implementations will break in production, and ensure that your software meets the user expectations and requirements1.
The other options are not as good as option C, for the following reasons:
Option A: Setting up a CI/CD pipeline that builds and tests your source code, and if the tests are successful, using the Google Cloud console to upload the built container to Artifact Registry and upload the compiled pipeline to Vertex AI Pipelines would not allow you to deploy new pipelines into production quickly and easily, and could increase the manual effort and errors. The Google Cloud console is a web-based user interface that can help you access and manage various Google Cloud services, such as Artifact Registry and Vertex AI Pipelines. Artifact Registry is a service that can store and manage your container images and other artifacts on Google Cloud. Artifact Registry can help you upload and organize your container images, and track the image versions and metadata. 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. However, setting up a CI/CD pipeline that builds and tests your source code, and if the tests are successful, using the Google Cloud console to upload the built container to Artifact Registry and upload the compiled pipeline to Vertex AI Pipelines would not allow you to deploy new pipelines into production quickly and easily, and could increase the manual effort and errors. You would need to write code, create and run the CI/CD pipeline, use the Google Cloud console to upload the built container to Artifact Registry, and use the Google Cloud console to upload the compiled pipeline to Vertex AI Pipelines. Moreover, this option would not use a pre-production environment to test and validate your pipeline before deploying it to production, which could increase the chance that the new pipeline implementations will break in production1.
Option B: Setting up a CI/CD pipeline that builds your source code and then deploys built artifacts into a pre-production environment, running unit tests in the pre-production environment, and if the tests are successful, deploying the pipeline to production would not allow you to test and validate your pipeline before deploying it to production, and could cause errors or poor performance. A unit test is a type of test that can verify the functionality and correctness of a small and isolated unit of code, such as a function or a class. A unit test can help you debug and improve your code quality, and catch any bugs or issues that may affect the code logic or output. However, setting up a CI/CD pipeline that builds your source code and then deploys built artifacts into a pre-production environment, running unit tests in the pre-production environment, and if the tests are successful, deploying the pipeline to production would not allow you to test and validate your pipeline before deploying it to production, and could cause errors or poor performance. You would need to write code, create and run the CI/CD pipeline, deploy the built artifacts to the pre-production

environment, run the unit tests in the pre-production environment, and deploy the pipeline to production. Moreover, this option would not run the pipeline in the pre-production environment, which could prevent you from testing and validating the pipeline functionality and compatibility, and catching any bugs or issues that may affect the pipeline workflow or output1.
Option D: Setting up a CI/CD pipeline that builds and tests your source code and then deploys built artifacts into a pre-production environment, after a successful pipeline run in the pre-production environment, rebuilding the source code, and deploying the artifacts to production would not allow you to deploy new pipelines into production quickly and easily, and could increase the complexity and cost of the pipeline development and deployment. Rebuilding the source code is a process that can recompile and repackage the source code into executable artifacts, such as container images and pipeline files. Rebuilding the source code can help you incorporate any changes or updates that may have occurred in the source code, and ensure that the artifacts are consistent and up-to-date. However, setting up a CI/CD pipeline that builds and tests your source code and then deploys built artifacts into a pre-production environment, after a successful pipeline run in the pre-production environment, rebuilding the source code, and deploying the artifacts to production would not allow you to deploy new pipelines into production quickly and easily, and could increase the complexity and cost of the pipeline development and deployment. You would need to write code, create and run the CI/CD pipeline, deploy the built artifacts to the pre-production environment, run the pipeline in the pre-production environment, rebuild the source code, and deploy the artifacts to production. Moreover, this option would increase the pipeline development and deployment time, as rebuilding the source code can be a time-consuming and resource-intensive process1.
Reference:
Preparing for Google Cloud Certification: Machine Learning Engineer, Course 3: Production ML Systems, Week 3: MLOps Google Cloud Professional Machine Learning Engineer Exam Guide, Section 3: Scaling ML models in production, 3.2 Automating ML workflows Official Google Cloud Certified Professional Machine Learning Engineer Study Guide, Chapter 6: Production ML Systems, Section 6.4: Automating ML Workflows Cloud Build Vertex AI Pipelines Artifact Registry Pre-production environment

NEW QUESTION # 161
Your organization wants to make its internal shuttle service route more efficient. The shuttles currently stop at all pick-up points across the city every 30 minutes between 7 am and 10 am. The development team has already built an application on Google Kubernetes Engine that requires users to confirm their presence and shuttle station one day in advance. What approach should you take?

- A. 1. Build a tree-based regression model that predicts how many passengers will be picked up at each shuttle station.
  2. Dispatch an appropriately sized shuttle and provide the map with the required stops based on the prediction.
- B. 1. Build a reinforcement learning model with tree-based classification models that predict the presence of passengers at shuttle stops as agents and a reward function around a distance-based metric
  2. Dispatch an appropriately sized shuttle and provide the map with the required stops based on the simulated outcome.
- C. 1. Build a tree-based classification model that predicts whether the shuttle should pick up passengers at each shuttle station.
  2. Dispatch an available shuttle and provide the map with the required stops based on the prediction
- D. 1. Define the optimal route as the shortest route that passes by all shuttle stations with confirmed attendance at the given time under capacity constraints.
  2 Dispatch an appropriately sized shuttle and indicate the required stops on the map

Answer: A

Explanation:
This answer is correct because it uses a regression model to estimate the number of passengers at each shuttle station, which is a continuous variable. A tree-based regression model can handle both numerical and categorical features, such as the time of day, the location of the station, and the weather conditions. Based on the predicted number of passengers, the organization can dispatch a shuttle that has enough capacity and provide a map that shows the required stops. This way, the organization can optimize the shuttle service route and reduce the waiting time and fuel consumption. Reference:
[Tree-based regression models]

NEW QUESTION # 162
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 data. 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. Reduce the maximum number of node hours for training.
* C. Decrease the score threshold.
* D. Add more negative examples to the training set.
* E. Add more positive examples to the training set.

Answer: C,E

Explanation:
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.
References:
* 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 # 163
You work for a retail company. You have a managed tabular dataset in Vertex Al 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 Al default data split.
* B. Use Vertex Al chronological split and specify the sales timestamp feature as the time vanable.

- C. Use Vertex Al random split assigning 70% of the rows to the training set, 10% to the validation set, and 20% to the test set.
- D. Use Vertex Al manual split, using the store name feature to assign one store for each set.

Answer: A

Explanation:
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 datathat 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 splitmethod 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.
References:
* About data splits for AutoML models | Vertex AI | Google Cloud
* Manual split for unstructured data
* Mathematical split

NEW QUESTION # 164
......

2026 Latest ExamCost Professional-Machine-Learning-Engineer PDF Dumps and Professional-Machine-Learning-Engineer Exam Engine Free Share: https://drive.google.com/open?id=1GgVllsU_surfHrDATOTE9D5VQ2f_HN5V