Professional Machine Learning Engineer Sample Questions

The Machine Learning Engineer sample questions will familiarize you with the format of exam questions and example content that may be covered on the exam.

The sample questions do not represent the range of topics or level of difficulty of questions presented on the exam. Performance on the sample questions should not be used to predict your Machine Learning Engineer exam result.

Registration

First Name *

Ashwin

Last Name *

C

Primary Email *

ashwin.a.c@capgemini.com

Recovery Email

ashwinchellappan97@gmail.com

Organization (Employer or School) *

Capgemini

Organization email (an email associated with your current organization)

ashwin.a.c@capgemini.com

Country *

India

Primary Relationship to Google *

Partner

Send me offers, updates and useful tips for getting the most out of Google Cloud training and certification products and services. *

Yes

Question 1

✗ You are developing a proof of concept for a real-time fraud detection model. After undersampling the training set to achieve a 50% fraud rate, you train and tune a tree classifier using area under the curve (AUC) as the metric, and then calibrate the model. You need to share metrics that represent your model's effectiveness with business stakeholders in a way that is easily interpreted. Which approach should you take?

- A. Calculate the AUC on the holdout dataset at a classification threshold of 0.5, and report true positive rate, false positive rate, and false negative rate.
- B. Undersample the minority class to achieve a 50% fraud rate in the holdout set. Plot the confusion matrix at a classification threshold of 0.5, and report precision and recall. ✗
- C. Select all transactions in the holdout dataset. Plot the area under the receiver operating characteristic curve (AUC ROC), and report the F1 score for all available thresholds.
- D. Select all transactions in the holdout dataset. Plot the precision-recall curve with associated average precision, and report the true positive rate, false positive rate, and false negative rate for all available thresholds.

Correct answer

- D. Select all transactions in the holdout dataset. Plot the precision-recall curve with associated average precision, and report the true positive rate, false positive rate, and false negative rate for all available thresholds.

Feedback

A is not correct because you need business directions about the cost of misclassification to define the optimal threshold for both balanced and imbalanced classification.

B is not correct because the holdout dataset needs to represent real-world transactions to have a meaningful model evaluation, and you should never change its distribution.

C is not correct because classes in the holdout dataset are not balanced, so the ROC curve is not appropriate; also, neither F1 score nor ROC curve is recommended for communicating to business stakeholders. The F1 score aggregates precision and recall, but it is important to look at each metric separately to evaluate the model's performance when the cost of misclassification is highly unbalanced between labels.

D is correct because the precision-recall curve is an appropriate metric for imbalanced classification when the output can be set using different thresholds. Presenting the precision-recall curve together with the mentioned rates provides business stakeholders with all the information necessary to evaluate model performance.

https://developers.google.... https://colab.research.goo... https://colab.research.goo...

Question 2

✗ Your organization's marketing team wants to send biweekly scheduled emails to customers that are expected to spend above a variable threshold. This is the first machine learning (ML) use case for the marketing team, and you have been tasked with the implementation. After setting up a new Google Cloud project, you use Vertex AI Workbench to develop model training and batch inference with an XGBoost model on the transactional data stored in Cloud Storage. You want to automate the end-to-end pipeline that will securely provide the predictions to the marketing team, while minimizing cost and code maintenance. What should you do?

- ○ A. Create a scheduled pipeline on Vertex AI Pipelines that accesses the data from Cloud Storage, uses Vertex AI to perform training and batch prediction, and outputs a file in a Cloud Storage bucket that contains a list of all customer emails and expected spending.
- ○ B. Create a scheduled pipeline on Cloud Composer that accesses the data from Cloud Storage, copies the data to BigQuery, uses BigQuery ML to perform training and batch prediction, and outputs a table in BigQuery with customer emails and expected spending.
- ○ C. Create a scheduled notebook on Vertex AI Workbench that accesses the data from Cloud Storage, performs training and batch prediction on the managed notebook instance, and outputs a file in a Cloud Storage bucket that contains a list of all customer emails and expected spending.
- ◉ D. Create a scheduled pipeline on Cloud Composer that accesses the data from Cloud Storage, uses Vertex AI to perform training and batch prediction, and sends an email to the marketing team's Gmail group email with an attachment that contains an encrypted list of all customer emails and expected spending. ✗

Correct answer

- ◉ A. Create a scheduled pipeline on Vertex AI Pipelines that accesses the data from Cloud Storage, uses Vertex AI to perform training and batch prediction, and outputs a file in a Cloud Storage bucket that contains a list of all customer emails and expected spending.

Feedback

A is correct because Vertex AI Pipelines and Cloud Storage are cost-effective and secure solutions. The solution requires the least number of code interactions because the marketing team can update the pipeline and schedule parameters from the Google Cloud console.

B is not correct. Cloud Composer is not a cost-efficient solution for one pipeline because

its environment is always active. In addition, using BigQuery is not the most cost-effective solution.

C is not correct because the marketing team would have to enter the Vertex AI Workbench instance to update a pipeline parameter, which does not minimize code interactions.

D is not correct. Cloud Composer is not a cost-efficient solution for one pipeline because its environment is always active. Also, using email to send personally identifiable information (PII) is not a recommended approach.

https://cloud.google.com/... https://cloud.google.com/...

https://cloud.google.com/... https://cloud.google.com/...

Question 3

✗ You have developed a very large network in TensorFlow Keras that is expected to train for multiple days. The model uses only built-in TensorFlow operations to perform training with high-precision arithmetic. You want to update the code to run distributed training using tf.distribute.Strategy and configure a corresponding machine instance in Compute Engine to minimize training time. What should you do?

- ○ A. Select an instance with an attached GPU, and gradually scale up the machine type until the optimal execution time is reached. Add MirroredStrategy to the code, and create the model in the strategy's scope with batch size dependent on the number of replicas.
- ○ B. Create an instance group with one instance with attached GPU, and gradually scale up the machine type until the optimal execution time is reached. Add TF_CONFIG and MultiWorkerMirroredStrategy to the code, create the model in the strategy's scope, and set up data autosharing.
- ○ C. Create a TPU virtual machine, and gradually scale up the machine type until the optimal execution time is reached. Add TPU initialization at the start of the program, define a distributed TPUStrategy, and create the model in the strategy's scope with batch size and training steps dependent on the number of TPUs.
- ◉ D. Create a TPU node, and gradually scale up the machine type until the optimal execution time is reached. Add TPU initialization at the start of the program, define a distributed TPUStrategy, and create the model in the strategy's scope with batch size and training steps dependent on the number of TPUs. ✗

Correct answer

- ◉ B. Create an instance group with one instance with attached GPU, and gradually scale up the machine type until the optimal execution time is reached. Add TF_CONFIG and MultiWorkerMirroredStrategy to the code, create the model in the strategy's scope, and set up data autosharing.

Feedback

A is not correct because it is suboptimal in minimizing execution time for model training. MirroredStrategy only supports multiple GPUs on one instance, which may not be as performant as running on multiple instances.

B is correct because GPUs are the correct hardware for deep learning training with high-precision training, and distributing training with multiple instances will allow maximum flexibility in fine-tuning the accelerator selection to minimize execution time. Note that one worker could still be the best setting if the overhead of synchronizing the gradients across machines is too high, in which case this approach will be equivalent to MirroredStrategy.

C is not correct because TPUs are not recommended for workloads that require high-precision arithmetic, and are recommended for models that train for weeks or months.

D is not correct because TPUs are not recommended for workloads that require high-precision arithmetic, and are recommended for models that train for weeks or months. Also, TPU nodes are not recommended unless required by the application.

https://cloud.google.com/...

https://www.tensorflow.or...

https://www.tensorflow.or...

Question 4

✗ You developed a tree model based on an extensive feature set of user behavioral data. The model has been in production for 6 months. New regulations were just introduced that require anonymizing personally identifiable information (PII), which you have identified in your feature set using the Cloud Data Loss Prevention API. You want to update your model pipeline to adhere to the new regulations while minimizing a reduction in model performance. What should you do?

- ○ A. Redact the features containing PII data, and train the model from scratch.
- ◉ B. Mask the features containing PII data, and tune the model from the last checkpoint. ✗
- ○ C. Use key-based hashes to tokenize the features containing PII data, and train the model from scratch.
- ○ D. Use deterministic encryption to tokenize the features containing PII data, and tune the model from the last checkpoint.

Correct answer

◉ C. Use key-based hashes to tokenize the features containing PII data, and train the model from scratch.

Feedback

A is not correct because removing features from the model does not keep referential integrity by maintaining the original relationship between records, and is likely to cause a drop in performance.

B is not correct because masking does not enforce referential integrity, and a drop in model performance may happen. Also, tuning the existing model is not recommended because the model training on the original dataset may have memorized sensitive information.

C is correct because hashing is an irreversible transformation that ensures anonymization and does not lead to an expected drop in model performance because you keep the same feature set while enforcing referential integrity.

D is not correct because deterministic encryption is reversible, and anonymization requires irreversibility. Also, tuning the existing model is not recommended because the model training on the original dataset may have memorized sensitive information.

https://cloud.google.com/... https://cloud.google.com/...

https://cloud.google.com/... https://cloud.google.com/...

Question 5

✓ You set up a Vertex AI Workbench instance with a TensorFlow Enterprise environment to perform exploratory data analysis for a new use case. Your training and evaluation datasets are stored in multiple partitioned CSV files in Cloud Storage. You want to use TensorFlow Data Validation (TFDV) to explore problems in your data before model tuning. You want to fix these problems as quickly as possible. What should you do?

- A. 1. Use TFDV to generate statistics, and use Pandas to infer the schema for the training dataset that has been loaded from Cloud Storage. 2. Visualize both statistics and schema, and manually fix anomalies in the dataset's schema and values.
- B. 1. Use TFDV to generate statistics and infer the schema for the training and evaluation datasets that have been loaded from Cloud Storage by using URI. 2. Visualize statistics for both datasets simultaneously to fix the datasets' values, and fix the training dataset's schema after displaying it together with anomalies in the evaluation dataset. ✓
- C. 1. Use TFDV to generate statistics, and use Pandas to infer the schema for the training dataset that has been loaded from Cloud Storage. 2. Use TFRecordWriter to convert the training dataset into a TFRecord. 3. Visualize both statistics and schema, and manually fix anomalies in the dataset's schema and values.
- D. 1. Use TFDV to generate statistics and infer the schema for the training and evaluation datasets that have been loaded with Pandas. 2. Use TFRecordWriter to convert the training and evaluation datasets into TFRecords. 3. Visualize statistics for both datasets simultaneously to fix the datasets' values, and fix the training dataset's schema after displaying it together with anomalies in the evaluation dataset.

Feedback

A is not correct because you also need to use the evaluation dataset for analysis. If the features do not belong to approximately the same range as the training dataset, the accuracy of the model will be affected.

B is correct because it takes the minimum number of steps to correctly fix problems in the data with TFDV before model tuning. This process involves installing tensorflow_data_validation, loading the training and evaluation datasets directly from Cloud Storage, and fixing schema and values for both. Note that the schema is only stored for the training set because it is expected to match at evaluation.

C is not correct because transforming into TFRecord is an unnecessary step. Also, you need to use the evaluation dataset for analysis. If the features do not belong to approximately the same range as the training dataset, the accuracy of the model will be affected.

D is not correct because transforming into TFRecord is an unnecessary step.

https://www.tensorflow.or… https://cloud.google.com/…

https://cloud.google.com/…

Question 6

✗ You have developed a simple feedforward network on a very wide dataset. You trained the model with mini-batch gradient descent and L1 regularization. During training, you noticed the loss steadily decreasing before moving back to the top at a very sharp angle and starting to oscillate. You want to fix this behavior with minimal changes to the model. What should you do?

- ○ A. Shuffle the data before training, and iteratively adjust the batch size until the loss improves.
- ○ B. Explore the feature set to remove NaNs and clip any noisy outliers. Shuffle the data before retraining.
- ○ C. Switch from L1 to L2 regularization, and iteratively adjust the L2 penalty until the loss improves.
- ◉ D. Adjust the learning rate to exponentially decay with a larger decrease at the step where the loss jumped, and iteratively adjust the initial learning rate until the loss improves. ✗

Correct answer

◉ B. Explore the feature set to remove NaNs and clip any noisy outliers. Shuffle the data before retraining.

Feedback

A is not correct because divergence due to repetitive behavior in the data typically shows a loss that starts oscillating after some steps but does not jump back to the top.

B is correct because a large increase in loss is typically caused by anomalous values in the input data that cause NaN traps or exploding gradients.

C is not correct because L2 is not clearly a better solution than L1 regularization for wide models. L1 helps with sparsity, and L2 helps with collinearity.

D is not correct because a learning rate schedule that is not tuned typically shows a loss that starts oscillating after some steps but does not jump back to the top.

https://developers.google.... https://developers.google....

https://developers.google....

Question 7

✓ You trained a neural network on a small normalized wide dataset. The model performs well without overfitting, but you want to improve how the model pipeline processes the features because they are not all expected to be relevant for the prediction. You want to implement changes that minimize model complexity while maintaining or improving the model's offline performance. What should you do?

- A. Keep the original feature set, and add L1 regularization to the loss function.
- B. Use principal component analysis (PCA), and select the first n components that explain 99% of the variance.
- C. Perform correlation analysis. Remove features that are highly correlated to one another and features that are not correlated to the target. ✓
- D. Ensure that categorical features are one-hot encoded and that continuous variables are binned, and create feature crosses for a subset of relevant features.

Feedback

A is not correct because, although the approach lets you reduce RAM requirements by pushing the weights for meaningless features to 0, regularization tends to cause the training error to increase. Consequently, the model performance is expected to decrease.

B is not correct because PCA is an unsupervised approach, and it is a valid method of feature selection only if the most important variables are the ones that also have the most variation. This is usually not true, and disregarding the last few components is likely to decrease model performance.

C is correct because removing irrelevant features reduces model complexity and is expected to boost performance by removing noise.

D is not correct because this approach can make the model converge faster but it increases model RAM requirements, and it is not expected to boost model performance because neural networks inherently learn feature crosses.

https://developers.google.... https://cloud.google.com/... https://developers.google....

Question 8

✗ You trained a model in a Vertex AI Workbench notebook that has good validation RMSE. You defined 20 parameters with the associated search spaces that you plan to use for model tuning. You want to use a tuning approach that maximizes tuning job speed. You also want to optimize cost, reproducibility, model performance, and scalability where possible if they do not affect speed. What should you do?

- (●) A. Set up a cell to run a hyperparameter tuning job using Vertex AI Vizier with val_rmse specified as the metric in the study configuration. ✗
- ( ) B. Using a dedicated Python library such as Hyperopt or Optuna, configure a cell to run a local hyperparameter tuning job with Bayesian optimization.
- ( ) C. Refactor the notebook into a parametrized and dockerized Python script, and push it to Container Registry. Use the UI to set up a hyperparameter tuning job in Vertex AI. Use the created image and include Grid Search as an algorithm.
- ( ) D. Refactor the notebook into a parametrized and dockerized Python script, and push it to Container Registry. Use the command line to set up a hyperparameter tuning job in Vertex AI. Use the created image and include Random Search as an algorithm where maximum trial count is equal to parallel trial count.

Correct answer

- (●) D. Refactor the notebook into a parametrized and dockerized Python script, and push it to Container Registry. Use the command line to set up a hyperparameter tuning job in Vertex AI. Use the created image and include Random Search as an algorithm where maximum trial count is equal to parallel trial count.

Feedback

A is not correct because Vertex AI Vizier should be used for systems that do not have a known objective function or are too costly to evaluate using the objective function. Neither applies to the specified use case. Vizier requires sequential trials and does not optimize for cost or tuning time.

B is not correct because Bayesian optimization can converge in fewer iterations than the other algorithms but not necessarily in a faster time because trials are dependent and thus require sequentiality. Also, running tuning locally does not optimize for reproducibility and scalability.

C is not correct because Grid Search is a brute-force approach and it is not feasible to fully parallelize. Because you need to try all hyperparameter combinations, that is an exponential number of trials with respect to the number of hyperparameters, Grid Search is inefficient for high spaces in time, cost, and computing power.

D is correct because Random Search can limit the search iterations on time and parallelize

all trials so that the execution time of the tuning job corresponds to the longest training produced by your hyperparameter combination. This approach also optimizes for the other mentioned metrics.

https://cloud.google.com/...

https://cloud.google.com/...

https://google-cloud-pipeli...

Question 9

✓ You trained a deep model for a regression task. The model predicts the expected sale price for a house based on features that are not guaranteed to be independent. You want to evaluate your model by defining a baseline approach and selecting an evaluation metric for comparison that detects high variance in the model. What should you do?

- A. Use a heuristic that predicts the mean value as the baseline, and compare the trained model's mean absolute error against the baseline.
- B. Use a linear model trained on the most predictive features as the baseline, and compare the trained model's root mean squared error against the baseline.
- C. Determine the maximum acceptable mean absolute percentage error (MAPE) as the baseline, and compare the model's MAPE against the baseline.
- (●) D. Use a simple neural network with one fully connected hidden layer as the baseline, and compare the trained model's mean squared error against the baseline. ✓

Feedback

A is not correct because always predicting the mean value is not expected to be a strong baseline; house prices could assume a wide range of values. Also, mean absolute error is not the best metric to detect variance because it gives the same weight to all errors.

B is not correct because a linear model is not expected to perform well with multicollinearity. Also, root mean squared error does not penalize high variance as much as mean squared error because the root operation reduces the importance of higher values.

C is not correct because, while defining a threshold for acceptable performance is a good practice for blessing models, a baseline should aim to test statistically a model's ability to learn by comparing it to a less complex data-driven approach. Also, this approach does not detect high variance in the model.

D is correct because a one-layer neural network can handle collinearity and is a good baseline. The mean square error is a good metric because it gives more weight to errors with larger absolute values than to errors with smaller absolute values.

https://developers.google.... https://cloud.google.com/... https://developers.google....

Question 10

✗ You designed a 5-billion-parameter language model in TensorFlow Keras that used autotuned tf.data to load the data in memory. You created a distributed training job in Vertex AI with tf.distribute.MirroredStrategy, and set the large_model_v100 machine for the primary instance. The training job fails with the following error:

"The replica 0 ran out of memory with a non-zero status of 9."

You want to fix this error without vertically increasing the memory of the replicas. What should you do?

- ○ A. Keep MirroredStrategy. Increase the number of attached V100 accelerators until the memory error is resolved.
- ○ B. Switch to ParameterServerStrategy, and add a parameter server worker pool with large_model_v100 instance type.
- ◉ C. Switch to tf.distribute.MultiWorkerMirroredStrategy with Reduction Server. Increase the number of workers until the memory error is resolved. ✗
- ○ D. Switch to a custom distribution strategy that uses TF_CONFIG to equally split model layers between workers. Increase the number of workers until the memory error is resolved.

Correct answer

- ◉ D. Switch to a custom distribution strategy that uses TF_CONFIG to equally split model layers between workers. Increase the number of workers until the memory error is resolved.

Feedback

A is not correct because MirroredStrategy is a data-parallel approach. This approach is not expected to fix the error because the memory issues in the primary replica are caused by the size of the model itself.

B is not correct because the parameter server alleviates some workload from the primary replica by coordinating the shared model state between the workers, but it still requires the whole model to be shared with workers. This approach is not expected to fix the error because the memory issues in the primary replica are caused by the size of the model itself.

C is not correct because MultiWorkerMirroredStrategy is a data-parallel approach. This approach is not expected to fix the error because the memory issues in the primary replica are caused by the size of the model itself. Reduction Server increases throughput and reduces latency of communication, but it does not help with memory issues.

D is correct because this is an example of a model-parallel approach that splits the model between workers. You can use TensorFlow Mesh to implement this. This approach is expected to fix the error because the memory issues in the primary replica are caused by the size of the model itself.

https://cloud.google.com/...

https://cloud.google.com/...

https://cloud.google.com/...

https://cloud.google.com/...

https://github.com/tensor...

Question 11

✗ You need to develop an online model prediction service that accesses pre-computed near-real-time features and returns a customer churn probability value. The features are saved in BigQuery and updated hourly using a scheduled query. You want this service to be low latency and scalable and require minimal maintenance. What should you do?

- A. 1. Configure a Cloud Function that exports features from BigQuery to Memorystore. 2. Use Memorystore to perform feature lookup. Deploy the model as a custom prediction endpoint in Vertex AI, and enable automatic scaling.
- B. 1. Configure a Cloud Function that exports features from BigQuery to Memorystore. 2. Use a custom container on Google Kubernetes Engine to deploy a service that performs feature lookup from Memorystore and performs inference with an in-memory model.
- C. 1. Configure a Cloud Function that exports features from BigQuery to Vertex AI Feature Store. 2. Use the online service API from Vertex AI Feature Store to perform feature lookup. Deploy the model as a custom prediction endpoint in Vertex AI, and enable automatic scaling. ✗
- D. 1. Configure a Cloud Function that exports features from BigQuery to Vertex AI Feature Store. 2. Use a custom container on Google Kubernetes Engine to deploy a service that performs feature lookup from Vertex AI Feature Store's online serving API and performs inference with an in-memory model.

Correct answer

- A. 1. Configure a Cloud Function that exports features from BigQuery to Memorystore. 2. Use Memorystore to perform feature lookup. Deploy the model as a custom prediction endpoint in Vertex AI, and enable automatic scaling.

Feedback

A is correct because this approach creates a fully managed autoscalable service that minimizes maintenance while providing low latency with the use of Memorystore.

B is not correct because feature lookup and model inference can be performed in Cloud Function, and using Google Kubernetes Engine increases maintenance.

C is not correct because Vertex AI Feature Store is not as low-latency as Memorystore.

D is not correct because feature lookup and model inference can be performed in Cloud Function, and using Google Kubernetes Engine increases maintenance. Also, Vertex AI Feature Store is not as low-latency as Memorystore.

https://cloud.google.com/...

https://cloud.google.com/...

https://cloud.google.com/...

Question 12

✓ You are logged into the Vertex AI Pipeline UI and noticed that an automated production TensorFlow training pipeline finished three hours earlier than a typical run. You do not have access to production data for security reasons, but you have verified that no alert was logged in any of the ML system's monitoring systems and that the pipeline code has not been updated recently. You want to debug the pipeline as quickly as possible so you can determine whether to deploy the trained model. What should you do?

- (●) A. Navigate to Vertex AI Pipelines, and open Vertex AI TensorBoard. Check whether the training regime and metrics converge. ✓
- ( ) B. Access the Pipeline run analysis pane from Vertex AI Pipelines, and check whether the input configuration and pipeline steps have the expected values.
- ( ) C. Determine the trained model's location from the pipeline's metadata in Vertex ML Metadata, and compare the trained model's size to the previous model.
- ( ) D. Request access to production systems. Get the training data's location from the pipeline's metadata in Vertex ML Metadata, and compare data volumes of the current run to the previous run.

Feedback

A is correct because TensorBoard provides a compact and complete overview of training metrics such as loss and accuracy over time. If the training converges with the model's expected accuracy, the model can be deployed.

B is not correct because checking input configuration is a good test, but it is not sufficient to ensure that model performance is acceptable. You can access logs and outputs for each pipeline step to review model performance, but it would involve more steps than using TensorBoard.

C is not correct because model size is a good indicator of health but does not provide a complete overview to make sure that the model can be safely deployed. Note that the pipeline's metadata can also be accessed directly from Vertex AI Pipelines.

D is not correct because data is the most probable cause of this behavior, but it is not the only possible cause. Also, access requests could take a long time and are not the most secure option. Note that the pipeline's metadata can also be accessed directly from Vertex AI Pipelines.

https://cloud.google.com/...

https://cloud.google.com/...

https://cloud.google.com/...

Question 13

✗ You recently developed a custom ML model that was trained in Vertex AI on a post-processed training dataset stored in BigQuery. You used a Cloud Run container to deploy the prediction service. The service performs feature lookup and pre-processing and sends a prediction request to a model endpoint in Vertex AI. You want to configure a comprehensive monitoring solution for training-serving skew that requires minimal maintenance. What should you do?

- A. Create a Model Monitoring job for the Vertex AI endpoint that uses the training data in BigQuery to perform training-serving skew detection and uses email to send alerts. When an alert is received, use the console to diagnose the issue.
- B. Update the model hosted in Vertex AI to enable request-response logging. Create a Data Studio dashboard that compares training data and logged data for potential training-serving skew and uses email to send a daily scheduled report.
- C. Create a Model Monitoring job for the Vertex AI endpoint that uses the training data in BigQuery to perform training-serving skew detection and uses Cloud Logging to send alerts. Set up a Cloud Function to initiate model retraining that is triggered when an alert is logged.
- D. Update the model hosted in Vertex AI to enable request-response logging. Schedule a daily DataFlow Flex job that uses Tensorflow Data Validation to detect training-serving skew and uses Cloud Logging to send alerts. Set up a Cloud Function to initiate model retraining that is triggered when an alert is logged. ✗

Correct answer

- A. Create a Model Monitoring job for the Vertex AI endpoint that uses the training data in BigQuery to perform training-serving skew detection and uses email to send alerts. When an alert is received, use the console to diagnose the issue.

Feedback

A is correct because Vertex AI Model Monitoring is a fully managed solution for monitoring training-serving skew that, by definition, requires minimal maintenance. Using the console for diagnostics is recommended for a comprehensive monitoring solution because there could be multiple causes for the skew that require manual review.

B is not correct because this solution does not minimize maintenance. It involves multiple custom components that require additional updates for any schema change.

C is not correct because a model retrain does not necessarily fix skew. For example, differences in pre-processing logic between training and prediction can also cause skew.

D is not correct because this solution does not minimize maintenance. It involves multiple components that require additional updates for any schema change. Also, a model retrain does not necessarily fix skew. For example, differences in pre-processing logic between training and prediction can also cause skew.

https://cloud.google.com/... https://cloud.google.com/...

Question 14

✗ You have a historical data set of the sale price of 10,000 houses and the 10 most important features resulting from principal component analysis (PCA). You need to develop a model that predicts whether a house will sell at one of the following equally distributed price ranges: 200-300k, 300-400k, 400-500k, 500-600k, or 600-700k. You want to use the simplest algorithmic and evaluative approach. What should you do?

- ○ A. Define a one-vs-one classification task where each price range is a categorical label. Use F1 score as the metric.
- ○ B. Define a multi-class classification task where each price range is a categorical label. Use accuracy as the metric.
- ○ C. Define a regression task where the label is the sale price represented as an integer. Use mean absolute error as the metric.
- ◉ D. Define a regression task where the label is the average of the price range that corresponds to the house sale price represented as an integer. Use root mean squared error as the metric. ✗

Correct answer

- ◉ B. Define a multi-class classification task where each price range is a categorical label. Use accuracy as the metric.

Feedback

A is not correct because this approach is more complex than the classification approach suggested in response B. F1 score is not useful with equally distributed labels, and one-vs-one classification is used for multi-label classification, but the use case would require only one label to be correct.

B is correct because the use case is an ordinal classification task which is most simply solved using multi-class classification. Accuracy as a metric is the best match for a use case with discrete and balanced labels.

C is not correct because regression is not the recommended approach when solving an ordinal classification task with a small number of discrete values. This specific regression approach adds complexity in comparison to the regression approach suggested in response D because it uses the exact sale price to predict a range. Finally, the mean absolute error would not be the recommended metric because it gives the same penalty for errors of any magnitude.

D is not correct because regression is not the recommended approach when solving an ordinal classification task with a small number of discrete values. This specific regression approach would be recommended in comparison to the regression approach suggested in

response C because it uses a less complex label and a recommended metric to minimize variance and bias.

https://cloud.google.com/... https://cloud.google.com/...

https://www.tensorflow.or... https://www.tensorflow.or...

Question 15

✗ You downloaded a TensorFlow language model pre-trained on a proprietary dataset by another company, and you tuned the model with Vertex AI Training by replacing the last layer with a custom dense layer. The model achieves the expected offline accuracy; however, it exceeds the required online prediction latency by 20ms. You want to optimize the model to reduce latency while minimizing the offline performance drop before deploying the model to production. What should you do?

- ○ A. Apply post-training quantization on the tuned model, and serve the quantized model.
- ◉ B. Use quantization-aware training to tune the pre-trained model on your dataset, and serve the quantized model. ✗
- ○ C. Use pruning to tune the pre-trained model on your dataset, and serve the pruned model after stripping it of training variables.
- ○ D. Use clustering to tune the pre-trained model on your dataset, and serve the clustered model after stripping it of training variables.

Correct answer

- ◉ A. Apply post-training quantization on the tuned model, and serve the quantized model.

Feedback

A is correct because post-training quantization is the recommended option for reducing model latency when re-training is not possible. Post-training quantization can minimally decrease model performance.

B is not correct because tuning the whole model on the custom dataset only will cause a drop in offline performance.

C is not correct because tuning the whole model on the custom dataset only will cause a drop in offline performance. Also, pruning helps in compressing model size, but it is expected to provide less latency improvements than quantization.

D is not correct because tuning the whole model on the custom dataset only will cause a drop in offline performance. Also, clustering helps in compressing model size, but it does not reduce latency.

https://cloud.google.com/...

https://www.tensorflow.or...

https://www.tensorflow.or...

Question 16

✗ You developed a model for a classification task where the minority class appears in 10% of the data set. You ran the training on the original imbalanced data set and have checked the resulting model performance. The confusion matrix indicates that the model did not learn the minority class. You want to improve the model performance while minimizing run time and keeping the predictions calibrated. What should you do?

- ○ A. Update the weights of the classification function to penalize misclassifications of the minority class.
- ○ B. Tune the classification threshold, and calibrate the model with isotonic regression on the validation set.
- ◉ C. Upsample the minority class in the training set, and update the weight of the upsampled class by the same sampling factor. ✗
- ○ D. Downsample the majority class in the training set, and update the weight of the downsampled class by the same sampling factor.

Correct answer

- ◉ D. Downsample the majority class in the training set, and update the weight of the downsampled class by the same sampling factor.

Feedback

A is not correct because this approach does not guarantee calibrated predictions and does not improve training run time.

B is not correct because this approach increases run time by adding threshold tuning and calibration on top of model training.

C is not correct because upsampling increases training run time by providing more data samples during training.

D is correct because downsampling with upweighting improves performance on the minority class while speeding up convergence and keeping the predictions calibrated.

- https://developers.google.…
- https://colab.research.goo…
- https://colab.research.goo…
- https://developers.google.…

Question 17

✗ You have a dataset that is split into training, validation, and test sets. All the sets have similar distributions. You have sub-selected the most relevant features and trained a neural network in TensorFlow. TensorBoard plots show the training loss oscillating around 0.9, with the validation loss higher than the training loss by 0.3. You want to update the training regime to maximize the convergence of both losses and reduce overfitting. What should you do?

- ○ A. Decrease the learning rate to fix the validation loss, and increase the number of training epochs to improve the convergence of both losses.
- ◉ B. Decrease the learning rate to fix the validation loss, and increase the number and dimension of the layers in the network to improve the convergence of both losses. ✗
- ○ C. Introduce L1 regularization to fix the validation loss, and increase the learning rate and the number of training epochs to improve the convergence of both losses.
- ○ D. Introduce L2 regularization to fix the validation loss, and increase the number and dimension of the layers in the network to improve the convergence of both losses.

Correct answer

- ◉ D. Introduce L2 regularization to fix the validation loss, and increase the number and dimension of the layers in the network to improve the convergence of both losses.

Feedback

A is not correct because changing the learning rate does not reduce overfitting. Increasing the number of training epochs is not expected to improve the losses significantly.

B is not correct because changing the learning rate does not reduce overfitting.

C is not correct because increasing the number of training epochs is not expected to improve the losses significantly, and increasing the learning rate could also make the model training unstable. L1 regularization could be used to stabilize the learning, but it is not expected to be particularly helpful because only the most relevant features have been used for training.

D is correct because L2 regularization prevents overfitting. Increasing the model's complexity boosts the predictive ability of the model, which is expected to optimize loss convergence when underfitting.

https://developers.google.... https://developers.google....

https://developers.google.... https://cloud.google.com/...

https://www.tensorflow.or... https://www.tensorflow.or...

https://cloud.google.com/...

Question 18

✗ You recently used Vertex AI Prediction to deploy a custom-trained model in production. The automated re-training pipeline made available a new model version that passed all unit and infrastructure tests. You want to define a rollout strategy for the new model version that guarantees an optimal user experience with zero downtime. What should you do?

- (●) A. Release the new model version in the same Vertex AI endpoint. Use traffic splitting in Vertex AI Prediction to route a small random subset of requests to the new version and, if the new version is successful, gradually route the remaining traffic to it. ✗
- ( ) B. Release the new model version in a new Vertex AI endpoint. Update the application to send all requests to both Vertex AI endpoints, and log the predictions from the new endpoint. If the new version is successful, route all traffic to the new application.
- ( ) C. Deploy the current model version with an Istio resource in Google Kubernetes Engine, and route production traffic to it. Deploy the new model version, and use Istio to route a small random subset of traffic to it. If the new version is successful, gradually route the remaining traffic to it.
- ( ) D. Install Seldon Core and deploy an Istio resource in Google Kubernetes Engine. Deploy the current model version and the new model version using the multi-armed bandit algorithm in Seldon to dynamically route requests between the two versions before eventually routing all traffic over to the best-performing version.

Correct answer

- (●) B. Release the new model version in a new Vertex AI endpoint. Update the application to send all requests to both Vertex AI endpoints, and log the predictions from the new endpoint. If the new version is successful, route all traffic to the new application.

Feedback

A is not correct because canary deployments may affect user experience, even if on a small subset of users.

B is correct because shadow deployments minimize the risk of affecting user experience while ensuring zero downtime.

C is not correct because canary deployments may affect user experience, even if on a small subset of users. This approach is a less managed alternative to response A and could cause downtime when moving between services.

D is not correct because the multi-armed bandit approach may affect user experience,

even if on a small subset of users. This approach could cause downtime when moving between services.

https://cloud.google.com/...
https://cloud.google.com/...
https://cloud.google.com/...
https://cloud.google.com/...
https://docs.seldon.io/pro...

Question 19

✗ You trained a model for sentiment analysis in TensorFlow Keras, saved it in SavedModel format, and deployed it with Vertex AI Predictions as a custom container. You selected a random sentence from the test set, and used a REST API call to send a prediction request. The service returned the error:

"Could not find matching concrete function to call loaded from the SavedModel. Got: Tensor("inputs:0", shape=(None,), dtype=string). Expected: TensorSpec(shape=(None, None), dtype=tf.int64, name='inputs')".

You want to update the model's code and fix the error while following Google-recommended best practices. What should you do?

- (●) A. Combine all preprocessing steps in a function, and call the function on the string input before requesting the model's prediction on the processed input. ✗
- ( ) B. Combine all preprocessing steps in a function, and update the default serving signature to accept a string input wrapped into the preprocessing function call.
- ( ) C. Create a custom layer that performs all preprocessing steps, and update the Keras model to accept a string input followed by the custom preprocessing layer.
- ( ) D. Combine all preprocessing steps in a function, and update the Keras model to accept a string input followed by a Lambda layer wrapping the preprocessing function.

Correct answer

- (●) B. Combine all preprocessing steps in a function, and update the default serving signature to accept a string input wrapped into the preprocessing function call.

Feedback

A is not correct because duplicating the preprocessing adds unnecessary dependencies between the training and serving code and could cause training-serving skew.

B is correct because this approach efficiently updates the model while ensuring no training-serving skew.

C is not correct because this approach adds unnecessary complexity. Because you update the model directly, you will need to re-train the model.

D is not correct because this approach adds unnecessary complexity. Because you update the model directly, you will need to re-train the model. Note that using Lambda layers over custom layers is recommended for simple operations or quick experimentation only.

https://cloud.google.com/...

https://www.tensorflow.or...

https://www.tensorflow.or...

https://developers.google....

Question 20

✓ You used Vertex AI Workbench user-managed notebooks to develop a TensorFlow model. The model pipeline accesses data from Cloud Storage, performs feature engineering and training locally, and outputs the trained model in Vertex AI Model Registry. The end-to-end pipeline takes 10 hours on the attached optimized instance type. You want to introduce model and data lineage for automated re-training runs for this pipeline only while minimizing the cost to run the pipeline. What should you do?

- A. 1. Use the Vertex AI SDK to create an experiment for the pipeline runs, and save metadata throughout the pipeline. 2. Configure a scheduled recurring execution for the notebook. 3. Access data and model metadata in Vertex ML Metadata.
- B. 1. Use the Vertex AI SDK to create an experiment, launch a custom training job in Vertex training service with the same instance type configuration as the notebook, and save metadata throughout the pipeline. 2. Configure a scheduled recurring execution for the notebook. 3. Access data and model metadata in Vertex ML Metadata.
- C. 1. Create a Cloud Storage bucket to store metadata. 2. Write a function that saves data and model metadata by using TensorFlow ML Metadata in one time-stamped subfolder per pipeline run. 3. Configure a scheduled recurring execution for the notebook. 4. Access data and model metadata in Cloud Storage. ✓
- D. 1. Refactor the pipeline code into a TensorFlow Extended (TFX) pipeline. 2. Load the TFX pipeline in Vertex AI Pipelines, and configure the pipeline to use the same instance type configuration as the notebook. 3. Use Cloud Scheduler to configure a recurring execution for the pipeline. 4. Access data and model metadata in Vertex AI Pipelines.

Feedback

A is not correct because a managed solution does not minimize running costs, and Vertex AI ML Metadata is more managed than Cloud Storage.

B is not correct because a managed solution does not minimize running costs, and this approach introduces Vertex training service with Vertex ML Metadata, which are both managed services.

C is correct because this approach minimizes running costs by being self-managed. This approach is recommended to minimize running costs only for simple use cases such as deploying one pipeline only. When optimizing for maintenance and development costs or scaling to more than one pipeline or performing experimentation, using Vertex ML Metadata and Vertex AI Pipelines are recommended.

D is not correct because a managed solution does not minimize running costs, and this approach introduces Vertex AI Pipelines, which is a fully managed service.

https://cloud.google.com/...
https://cloud.google.com/...
https://cloud.google.com/...
https://cloud.google.com/...
https://cloud.google.com/...

This form was created inside of Google.com. Privacy & Terms

Google Forms