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 *

Soma

Last Name *

S

Primary Email *

soma.bss@gmail.com

Recovery Email

soma.bss@gmail.com

Organization (Employer or School) *

Fiserv

Organization email (an email associated with your current organization)

Country * Dropdown

United States

Primary Relationship to Google * Dropdown

Other - Student

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

No

Question 1

✓ 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 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.

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 2

✗ 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 autosharding.
- 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 autosharding.

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 3

✓ 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.

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 4

✓ You need to train an object detection model to identify bounding boxes around Post-it Notes® in an image. Post-it Notes can have a variety of background colors and shapes. You have a dataset with 1000 images with a maximum size of 1.4MB and a CSV file containing annotations stored in Cloud Storage. You want to select a training method that reliably detects Post-it Notes of any relative size in the image and that minimizes the time to train a model. What should you do?

- A. Use the Cloud Vision API in Vertex AI with OBJECT_LOCALIZATION type, and filter the detected objects that match the Post-it Note category only.
- B. Upload your dataset into Vertex AI. Use Vertex AI AutoML Vision Object Detection with accuracy as the optimization metric, early stopping enabled, and no training budget specified. ✓
- C. Write a Python training application that trains a custom vision model on the training set. Autopackage the application, and configure a custom training job in Vertex AI.
- D. Write a Python training application that performs transfer learning on a pre-trained neural network. Autopackage the application, and configure a custom training job in Vertex AI.

Feedback

A is not correct because the object detection capability of the Cloud Vision API confidently detects large objects within the image and is not the best option to reliably detect sticky notes of any relative size in the image.

B is correct because AutoML is a codeless solution that minimizes time to train and develop the model, and it is capable of detecting bounding boxes up to one percent the length of a side of an image.

C is not correct because creating a custom training job requires more development time than using AutoML does. The extra flexibility of custom training is not required because AutoML achieves state-of-the-art performance even on tiny objects (8-32 pixels). Additionally, training a model from scratch is not expected to be as performant as transfer learning.

D is not correct because creating a custom training job requires more development time than using AutoML does. The extra flexibility of custom training is not required because AutoML achieves state-of-the-art performance even on tiny objects (8-32 pixels).

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

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

Question 5

✓ You used Vertex AI Workbench notebooks to build a model in TensorFlow. The notebook i) loads data from Cloud Storage, ii) uses TensorFlow Transform to pre-process data, iii) uses built-in TensorFlow operators to define a sequential Keras model, iv) trains and evaluates the model with *model.fit()* on the notebook instance, and v) saves the trained model to Cloud Storage for serving. You want to orchestrate the model retraining pipeline to run on a weekly schedule while minimizing cost and implementation effort. What should you do?

- A. Add relevant parameters to the notebook cells and set a recurring run in Vertex AI Workbench.
- B. Use TensorFlow Extended (TFX) with Google Cloud executors to define your pipeline, and automate the pipeline to run on Cloud Composer.
- (●) C. Use Kubeflow Pipelines SDK with Google Cloud executors to define your pipeline, and use Vertex AI pipelines to automate the pipeline to run. ✓
- D. Separate each cell in the notebook into a containerised application and use Cloud Workflows to launch each application.

Feedback

A is not correct because Vertex AI Workbench does not provide alerts, and you would have to log in every week to check the pipeline run status. This does not minimize monitoring steps.

B is not correct because Cloud Composer does not provide ML-specific monitoring capabilities. Also, unless many pipelines are hosted in Cloud Composer, this solution is not the most cost-efficient.

C is correct because using the Kubeflow Pipelines SDK is the best practice to orchestrate AI pipelines with modular steps.

D is not correct because this approach requires more effort and does not follow best practices given that Vertex AI pipelines is the more suitable product to run modular containerised AI pipeline steps.

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

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

Question 6

✓ 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 Vertex AI Feature Store to automatically import features from BigQuery, and serve them to the model. 2. Deploy the prediction model as a custom Vertex AI endpoint, 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.

Feedback

A is correct because using Vertex AI Feature Store with BigQuery prioritizes low latency, scalability, requires minimal maintenance, and facilitates integration with other Vertex AI services as a fully managed solution.

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

Question 7

✓ 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 assure the quality of the pipeline results as quickly as possible so you can determine whether to deploy the trained model. What should you do?

- (●) A. Use Vertex AI TensorBoard to check whether the training metrics converge to typical values. Verify pipeline input configuration and steps have the expected values. ✓
- ( ) B. Upgrade to the latest version of the Vertex SDK and re-run the pipeline.
- ( ) 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 8

✗ 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 9

✓ You recently developed a classification model that predicts which customers will be repeat customers. Before deploying the model, you perform post-training analysis on multiple data slices and discover that the model is under-predicting for users who are more than 60 years old. You want to remove age bias while maintaining similar offline performance. What should you do?

- A. Perform correlation analysis on the training feature set against the age column, and remove features that are highly correlated with age from the training and evaluation sets.
- B. Review the data distribution for each feature against the bucketized age column for the training and evaluation sets, and introduce preprocessing to even irregular feature distributions. ✓
- C. Configure the model to support explainability, and modify the input-baselines to include min and max age ranges.
- D. Apply a calibration layer at post-processing that matches the prediction distributions of users below and above 60 years old.

Feedback

A is not correct because this approach could lead to large drops in offline performance.

B is correct because this approach compensates for bias directly in the data by enhancing the data distribution of users above 60 years old. Some useful preprocessing steps could be filling null values, bucketizing, clipping outliers, sampling, or even collecting new data.

C is not correct because modifying input baselines will only adjust explainability of model features and not offline model performance.

D is not correct because this approach could add unconscious or implicit bias to the label. This approach is not recommended because it is a brittle solution that fixes the symptom rather than the cause.

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

Question 10

✗ 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 reduce latency while minimizing the offline performance drop and modifications to the model 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. Apply knowledge distillation to train a new, smaller "student" model that mimics the behavior of the larger, fine-tuned 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... https://cloud.google.com/...

Question 11

✗ 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. 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.

Correct answer

- D. Introduce L2 regularization to fix the validation loss.

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 12

✗ 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 13

✗ You work as an analyst at a large banking firm. You are developing a robust, scalable ML pipeline to train several regression and classification models. Your primary focus for the pipeline is model interpretability. You want to productionize the pipeline as quickly as possible. What should you do?

- (●) A. Use Tabular Workflow for Wide & Deep through Vertex AI Pipelines to jointly train wide linear models and deep neural networks. ✗
- ( ) B. Use Cloud Composer to build the training pipelines for custom deep learning-based models.
- ( ) C. Use Google Kubernetes Engine to build a custom training pipeline for XGBoost-based models.
- ( ) D. Use Tabular Workflow for TabNet through Vertex AI Pipelines to train attention-based models.

Correct answer

- (●) D. Use Tabular Workflow for TabNet through Vertex AI Pipelines to train attention-based models.

Feedback

A is not correct because though Tabular Workflows for Wide & Deep is capable of handling classification and regression pipelines, it's optimized for memorization and generalization, and in general deep learning-based models are not preferred for interpretability.

B is not correct because Cloud Composer is not the right tool to build an ML pipeline quickly, and in general deep learning-based models are not preferred for interpretability.

C is not correct because building a pipeline on Google Kubernetes Engine would take a long time.

D is correct because TabNet uses sequential attention that promotes model interpretability and Tabular Workflows is a set of integrated, fully managed, and scalable pipelines for end-to-end ML with tabular data for regression and classification.

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

Question 14

✓ You are developing a custom image classification model in Python. You plan to run your training application on Vertex AI. Your input dataset contains several hundred thousand small images. You need to determine how to store and access the images for training. You want to maximize data throughput and minimize training time while reducing the amount of additional code. What should you do?

- ○ Store image files in Cloud Storage, and access them directly.
- ○ Store image files in Cloud Storage, and access them by using serialized records.
- ◉ Store image files in Cloud Filestore, and access them by using serialized records. ✓
- ○ Store image files in Cloud Filestore, and access them directly by using an NFS mount point.

Feedback

A is not correct because Cloud Storage is not optimized for accessing lots of small files, as there is overhead in establishing the connections to retrieve each file.

B is not correct because although accessing a large archive via serialized records (TFRecords, WebDatasets) is faster than small files, it's still slower than using Filestore.

C is correct because Filestore is faster than Cloud Storage for accessing files, and serialized records are faster for feeding training pipelines than individual files.

D is not correct because although Filestore is faster than Cloud Storage for accessing files, serialized records are still faster than individual file I/O.

https://github.com/webda... https://cloud.google.com/... https://cloud.google.com/... https://cloud.google.com/...

Question 15

✗ Your company manages an ecommerce website. You developed an ML model that recommends additional products to users in near real time based on items currently in the user's cart. The workflow will include the following processes:

1. The website will send a Pub/Sub message with the relevant data, and then receive a message with the prediction from Pub/Sub.

2. Predictions will be stored in BigQuery.

3. The model will be stored in a Cloud Storage bucket and will be updated frequently.

You want to minimize prediction latency and the effort required to update the model. How should you reconfigure the architecture?

- ○ Write a Cloud Function that loads the model into memory for prediction. Configure the function to be triggered when messages are sent to Pub/Sub.
- ○ Expose the model as a Vertex AI endpoint. Write a custom DoFn in a Dataflow job that calls the endpoint for prediction.
- ○ Use the RunInference API with WatchFilePattern in a Dataflow job that wraps around the model and serves predictions.(*)
- ◉ Create a pipeline in Vertex AI Pipelines that performs preprocessing, prediction, and postprocessing. Configure the pipeline to be triggered by a Cloud Function when messages are sent to Pub/Sub. ✗

Correct answer

- ◉ Use the RunInference API with WatchFilePattern in a Dataflow job that wraps around the model and serves predictions.(*)

Feedback

A is not correct because Cloud Functions will run into limitations based on request rate and model size.

B is not correct because exposing the model as an endpoint adds to the total latency.

C is correct because the RunInference API with a locally loaded model minimizes the prediction latency and makes model updates seamless.

D is not correct because provisioning Vertex AI Pipelines adds to the total latency.

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