PassQueen
a.k.a dumpsqueen.com

Google Professional Machine Learning Engineer

Google Professional-Machine-Learning-Engineer

Version Demo

Total Demo Questions: 38

Total Premium Questions: 389

Buy Premium PDF

https://passqueen.com

support@passqueen.com

Topic Break Down

Topic	No. of Questions
Topic 1, Architecting low-code AI solutions	51
Topic 2, Architecting ML solutions	97
Topic 3, Data preparation and processing	65
Topic 4, Developing ML models	66
Topic 5, Automating and orchestrating ML pipelines	49
Topic 6, Monitoring AI solutions	61
Total	389

QUESTION NO: 1

You are an ML engineer at a bank. You have developed a binary classification model using AutoML Tables to predict whether a customer will make loan payments on time. The output is used to approve or reject loan requests. One customer's loan request has been rejected by your model, and the bank's risks department is asking you to provide the reasons that contributed to the model's decision. What should you do?

A. Use local feature importance from the predictions.

B. Use the correlation with target values in the data summary page.

C. Use the feature importance percentages in the model evaluation page.

D. Vary features independently to identify the threshold per feature that changes the classification.

ANSWER: A

Explanation:

Use local feature importance from the predictions. The bank needs an explanation for one specific rejected loan request, rather than an overall description of model behavior. Local feature importance provides per-prediction feature attributions: it identifies the input features that most influenced the model's score for that particular customer and indicates the direction and relative magnitude of their contribution. This makes it appropriate for communicating the factors behind an individual automated lending decision to the risks department, supporting review, governance, and responsible-use processes. In Vertex AI, the successor service to AutoML Tables, these per-instance attributions are provided through feature attribution methods and can be retrieved for online or batch predictions. For tabular classification models, the attribution values show how each feature affected the prediction relative to a baseline, enabling a decision-specific explanation rather than merely a population-level summary. Google documents feature attributions and their use for explaining individual predictions in the Vertex AI Explainable AI overview and describes how to obtain explanations for tabular models in the online predictions documentation.

QUESTION NO: 2

You want to use a single feature definition for both a Vertex AI training pipeline and an online prediction service. Feature values are updated continuously from streaming events, and the online service requires low-latency feature retrieval. Which actions should you take?

Choose 2 answers

A. Maintain offline historical features for training and online features for low-latency serving.

B. Version shared feature-transformation definitions used by training and serving workflows.

C. Implement separate unversioned feature logic for the training and online services.

D. Retrieve all historical training data synchronously from the online prediction request path.

E. Store streaming feature events only in local memory on serving instances.

ANSWER: A B

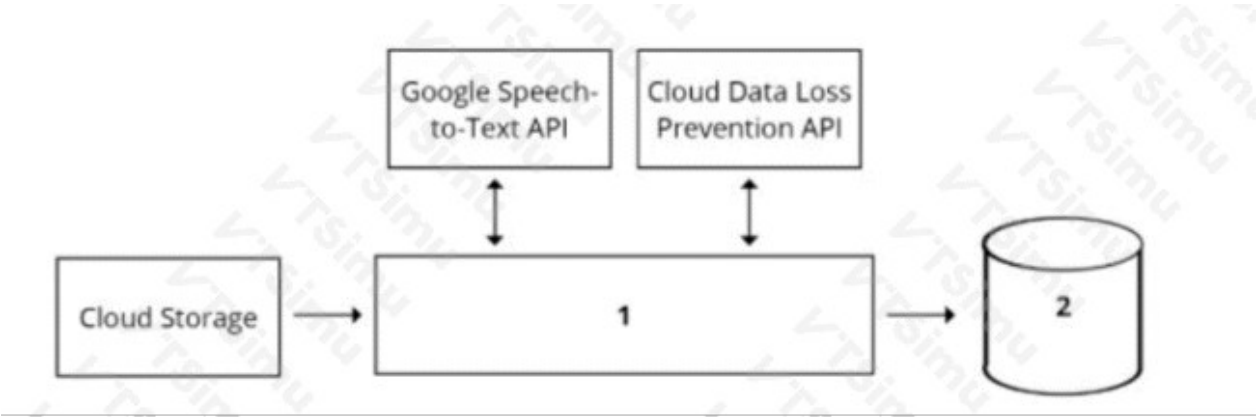
Explanation:

Use a feature store pattern that maintains offline historical feature data for training and online feature values for low-latency serving. This reduces the risk that independently implemented training and serving transformations generate inconsistent values.

Define and version feature transformations in shared pipeline code, then apply that same definition to both offline and online feature generation. Shared definitions help prevent training-serving skew, especially when features have complex aggregation windows or categorical transformations. See the Google Cloud MLOps architecture guidance and the Google Rules of ML guidance on training-serving skew.

QUESTION NO: 3

Your organization's call center has asked you to develop a model that analyzes customer sentiments in each call. The call center receives over one million calls daily, and data is stored in Cloud Storage. The data collected must not leave the region in which the call originated, and no Personally Identifiable Information (PII) can be stored or analyzed. The data science team has a third-party tool for visualization and access which requires a SQL ANSI-2011 compliant interface. You need to select components for data processing and for analytics. How should the data pipeline be designed?



A. 1 = Dataflow, 2 = BigQuery

B. 1 = Pub/Sub, 2 = Datastore

C. 1 = Dataflow, 2 = Cloud SQL

D. 1 = Cloud Function, 2 = Cloud SQL

ANSWER: A

Explanation:

1 = Dataflow, 2 = BigQuery is the appropriate design for this high-volume call-processing pipeline. Dataflow is a fully managed, autoscaling data-processing service built on Apache Beam. It can read the call data from regional Cloud Storage, perform scalable parallel transformations such as transcription enrichment, PII detection and removal, and sentiment-processing preparation, and write only de-identified results to the analytics destination. A Dataflow job can be configured to run in the same region as the source data, helping meet the data-residency requirement. Regional placement must also be applied to the Cloud Storage buckets and all involved services so that call content is not processed across regions.

BigQuery is the appropriate analytics layer because it provides a managed, highly scalable data warehouse for the resulting large daily volume of structured sentiment data. It supports GoogleSQL, which provides an ANSI SQL-compatible interface suitable for third-party SQL-based visualization and access tools. BigQuery datasets can be created in the required location, allowing the sanitized analytics data to remain regional. Together, Dataflow supplies the scalable processing and de-identification stage, while BigQuery supplies governed, SQL-accessible analytics. See the Dataflow regional endpoints documentation and the BigQuery introduction.

QUESTION NO: 4

You have trained an XGBoost model that you plan to deploy on Vertex AI for online prediction. You are now uploading your model to Vertex AI Model Registry, and you need to configure the explanation method that will serve online prediction requests to be returned with minimal latency. You also want to be alerted when feature attributions of the model meaningfully change over time. What should you do?

A. 2 Deploy the model to Vertex AI Endpoints.

B. 2 Deploy the model to Vertex AI Endpoints.

C. 2. Deploy the model to Vertex AI Endpoints.

D. 2. Deploy the model to Vertex AI Endpoints.

E. Configure Sampled Shapley with a low path count for online explanations, and configure Vertex AI Model Monitoring feature-attribution drift alerts.

ANSWER: E

Explanation:

Configure Sampled Shapley with a low path count for online explanations, and configure Vertex AI Model Monitoring feature-attribution drift alerts. Sampled Shapley is a feature-based explanation method suitable for tabular models such as XGBoost. It estimates each input feature's contribution to an individual prediction using sampled paths; reducing the path count reduces the computation required for each explanation request. This makes it appropriate when explanations must be returned alongside online predictions with low latency. The path count is a tunable accuracy-versus-latency setting, so it should be selected at the lowest value that provides sufficiently stable attributions for the application.

After the model is deployed, configure Vertex AI Model Monitoring to monitor feature attribution drift and set notification channels for alerts. Attribution drift measures whether the distribution of feature-importance values produced by the deployed model has shifted meaningfully relative to the baseline. Monitoring this signal helps detect changes in how the model is using features, even when the model artifact itself has not changed. Google documents feature-based explanation configuration, including Sampled Shapley settings, in Configure feature-based explanations. Vertex AI monitoring capabilities and alerting are described in the Model Monitoring overview.

QUESTION NO: 5

You are training a Resnet model on AI Platform using TPUs to visually categorize types of defects in automobile engines. You capture the training profile using the Cloud TPU profiler plugin and observe that it is highly input-bound. You want to reduce the bottleneck and speed up your model training process. Which modifications should you make to the tf .data dataset?

Choose 2 answers

A. Use the interleave option for reading data

B. Reduce the value of the repeat parameter

C. Increase the buffer size for the shuffle option.

D. Use the prefetch option with a buffer size of `tf.data.AUTOTUNE`

E. Decrease the batch size argument in your transformation

ANSWER: A D

Explanation:

An input-bound TPU profile means that the accelerator is waiting for data rather than spending its time performing model computation. The tf.data pipeline should therefore overlap input work with TPU execution and increase parallelism when reading files. "Use the interleave option for reading data" is appropriate because `Dataset.interleave()` can read from multiple input files concurrently, rather than waiting for one file's reads and processing to finish before moving to the next. This is especially useful for image datasets distributed across many files.

"Use the prefetch option with a buffer size of `tf.data.AUTOTUNE`" is also appropriate. Prefetching prepares subsequent dataset elements while the TPU trains on the current element, reducing idle time caused by input latency. With `AUTOTUNE`, TensorFlow tunes the buffer size dynamically based on available resources and pipeline behavior; the prefetch buffer is measured in dataset elements, not inherently in training batch size. Together, parallel file interleaving and asynchronous prefetching improve throughput from storage and CPU preprocessing through device consumption. Google recommends these tf.data performance optimizations for training input pipelines. See TensorFlow: Better performance with the tf.data API and Cloud TPU troubleshooting and performance guidance.

QUESTION NO: 6

You have a functioning end-to-end ML pipeline that involves tuning the hyperparameters of your ML model using AI Platform, and then using the best-tuned parameters for training. Hypertuning is taking longer than expected and is delaying the downstream processes. You want to speed up the tuning job without significantly compromising its effectiveness. Which actions should you take?

Choose 2 answers

A. Decrease the number of parallel trials

B. Decrease the range of floating-point values

C. Set the early stopping parameter to TRUE

D. Change the search algorithm from Bayesian search to random search.

E. Decrease the maximum number of trials during subsequent training phases.

ANSWER: C E

Explanation:

Setting the early stopping parameter to TRUE can reduce tuning-job duration by ending individual trials that are unlikely to produce a competitive result. Hyperparameter tuning monitors the reported objective metric from a trial's intermediate measurements. When the service determines that a trial is underperforming relative to other trials, early stopping avoids spending the remaining training time on that unpromising parameter configuration. This preserves the search effort for trials with better observed potential while reducing unnecessary compute and elapsed time.

Decreasing the maximum number of trials during subsequent training phases directly limits the number of parameter configurations that the tuning service evaluates. Each trial is a separate training execution, so a smaller maximum trial count reduces the total work and allows the pipeline to proceed sooner. When paired with a thoughtfully bounded search space and early stopping, this is a practical way to reduce latency while retaining a useful optimization process and selecting the best result among the completed trials. Google documents both maximum trial configuration and early stopping behavior for hyperparameter tuning in the Vertex AI hyperparameter tuning overview and the early stopping guidance.

QUESTION NO: 7

You are training a custom deep learning model on Vertex AI. The training job can tolerate interruptions because the training application saves checkpoints to Cloud Storage after every epoch. You want to minimize training cost without losing more than one epoch of progress if a worker is interrupted. Which actions should you take?

Choose 2 answers

A. Configure the Vertex AI training job to use Spot VMs

B. Write training checkpoints to Cloud Storage and restore from the latest checkpoint

C. Store checkpoints only on the local disk of each training worker

D. Disable checkpointing because Spot VMs are less expensive than standard VMs

E. Increase the number of training workers after each interruption

ANSWER: A B

Explanation:

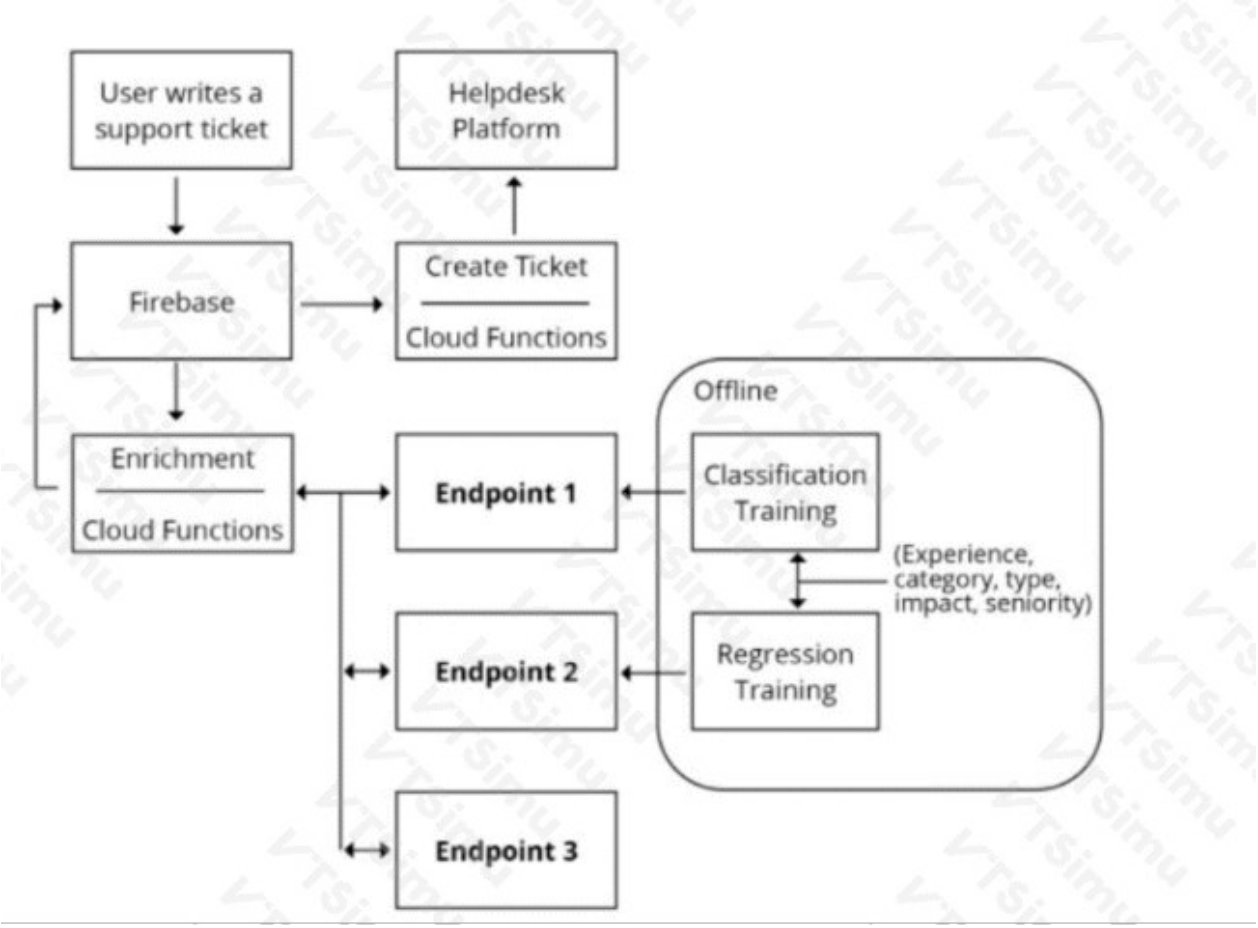
Configure the Vertex AI custom training job to use Spot VMs. Spot VMs provide unused compute capacity at a lower price, but they can be preempted. They are appropriate for fault-tolerant training workloads when the application can resume from saved state.

Write checkpoints to durable Cloud Storage and resume training from the most recent checkpoint after an interruption. Local worker disks can be lost when a Spot VM is preempted, so checkpoints must be written to persistent storage. See the Vertex AI custom training compute configuration documentation and the Compute Engine Spot VMs documentation.

QUESTION NO: 8

You are designing an architecture with a serverless ML system to enrich customer support tickets with informative metadata before they are routed to a support agent. You need a set of models to predict ticket priority, predict ticket resolution time, and perform sentiment analysis to help agents make strategic decisions when they process support requests. Tickets are not expected to have any domain-specific terms or jargon.

The proposed architecture has the following flow:



Which endpoints should the Enrichment Cloud Functions call?

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

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

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

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

ANSWER: B

Explanation:

1 = Vertex AI. 2 = Vertex AI. 3 = Cloud Natural Language API is correct. Predicting ticket priority and estimated resolution time are business-specific supervised-learning tasks. The models need to learn from the organization's historical support-ticket data, including the ticket content and the associated priority or time-to-resolution labels. After these custom models are trained and deployed, Vertex AI online prediction endpoints provide a managed, scalable HTTPS interface that the enrichment Cloud Functions can invoke for each incoming ticket.

Sentiment analysis, by contrast, is a standard natural-language processing task for which Google provides a pretrained model through the Cloud Natural Language API. The API can return document sentiment directly, without requiring the team to collect labels or train and host a separate model. The statement that tickets do not contain domain-specific jargon supports use of this general-purpose pretrained language capability. Cloud Functions can therefore orchestrate all three

synchronous enrichment calls, write the resulting metadata back to the ticket record, and allow the downstream routing workflow to use the predictions.

Google documents deployment and online inference through Vertex AI online predictions and describes sentiment analysis in the Cloud Natural Language API documentation.

QUESTION NO: 9

Your company stores a large number of audio files of phone calls made to your customer call center in an on-premises database. Each audio file is in wav format and is approximately 5 minutes long. You need to analyze these audio files for customer sentiment. You plan to use the Speech-to-Text API. You want to use the most efficient approach. What should you do?

A. 1 Upload the audio files to Cloud Storage2. Call the speech: longrunningrecognize API endpoint to generate transcriptions3. Call the predict method of an AutoML sentiment analysis model to analyze the transcriptions

B. 1 Upload the audio files to Cloud Storage2 Call the speech: longrunningrecognize API endpoint to generate transcriptions.3 Create a Cloud Function that calls the Natural Language API by using the analyzesentiment method

C. 1 Iterate over your local Tiles in Python2. Use the Speech-to-Text Python library to create a speech.RecognitionAudio object and set the content to the audio file data3. Call the speech: recognize API endpoint to generate transcriptions4. Call the predict method of an AutoML sentiment analysis model to analyze the transcriptions

D. 1 Iterate over your local files in Python2 Use the Speech-to-Text Python Library to create a speech.RecognitionAudio object, and set the content to the audio file data3. Call the speech: lengrunningrecognize API endpoint to generate transcriptions4 Call the Natural Language API by using the analyzesenriment method

ANSWER: B

Explanation:

"1 Upload the audio files to Cloud Storage 2 Call the speech: longrunningrecognize API endpoint to generate transcriptions. 3 Create a Cloud Function that calls the Natural Language API by using the analyzesentiment method" is the most efficient managed-service design. First, Cloud Storage provides durable, scalable object storage and lets Speech-to-Text access audio through a Cloud Storage URI rather than requiring the application to load and transmit every audio file itself. Because each call is approximately five minutes long and there are many files, asynchronous long-running recognition is appropriate: it submits recognition work as an operation that can be monitored or handled after completion without holding a synchronous request open. Google documents long-running recognition specifically for audio longer than one minute and for batch-style transcription workflows.

After transcription, the Natural Language API's sentiment analysis capability can analyze the resulting text and return an overall sentiment score and magnitude. A Cloud Function supplies lightweight event-driven orchestration, allowing the transcription output to trigger the sentiment-analysis step without operating servers or building and training a custom sentiment model. This uses purpose-built Google Cloud APIs for both stages and scales with the file-processing workload. See the Speech-to-Text asynchronous recognition documentation and the Natural Language sentiment analysis documentation.

QUESTION NO: 10

You are developing a model to help your company create more targeted online advertising campaigns. You need to create a dataset that you will use to train the model. You want to avoid creating or reinforcing unfair bias in the model. What should you do?

Choose 2 answers

A. Include a comprehensive set of demographic features.

B. include only the demographic groups that most frequently interact with advertisements.

C. Collect a random sample of production traffic to build the training dataset.

D. Collect a stratified sample of production traffic to build the training dataset.

E. Conduct fairness tests across sensitive categories and demographics on the trained model.

ANSWER: C E

Explanation:

Collect a random sample of production traffic to build the training dataset because training data should reflect the population and real-world inputs on which the advertising model will operate. Sampling from production traffic helps avoid constructing a dataset from a narrow, selectively chosen audience and provides a sound basis for assessing whether outcomes generalize across the actual user population. Google's guidance on data sampling describes random sampling as a way to select rows from a dataset without systematically favoring particular records or segments. See BigQuery table sampling documentation.

Conduct fairness tests across sensitive categories and demographics on the trained model because representative collection alone does not establish equitable model behavior. Model evaluation must inspect outcomes for relevant groups, such as comparative error rates, prediction distributions, and performance metrics, to identify disparities that may result in unfair treatment. TensorFlow Fairness Indicators supports evaluating commonly used fairness metrics across slices of data, enabling teams to detect and monitor group-level differences throughout the model lifecycle. See TensorFlow Fairness Indicators documentation. Together, production-based sampling and group-aware fairness evaluation reduce the risk of introducing or perpetuating bias in targeted advertising decisions.

QUESTION NO: 11

You are building a Vertex AI Pipeline for a loan-default model. The pipeline receives a new training dataset every week from BigQuery. You want the pipeline to stop before training if a required feature is missing, a feature type changes, or the rate of missing values exceeds an approved threshold. Which actions should you take?

Choose 2 answers

A. Define a schema for the expected training features and their constraints.

B. Run TensorFlow Data Validation against the incoming dataset before training.

C. Use a larger training machine type to tolerate malformed records.

D. Randomly split the incoming data into training and validation datasets before checking it.

E. Enable Vertex AI Model Monitoring only after the model is deployed.

ANSWER: A B

Explanation:

Define a schema that specifies the expected features, their data types, and relevant constraints. A schema provides the explicit data contract against which incoming examples can be compared. It can identify conditions such as missing required features and incompatible type changes.

Run TensorFlow Data Validation (TFDV) to generate statistics and validate the incoming dataset against the schema before training. TFDV can detect anomalies, including unexpected schema differences and constraint violations such as excessive missing values. A pipeline can use this validation result as a gate so invalid data does not reach training. See TensorFlow Data Validation and the TensorFlow Data Validation schema guide.

QUESTION NO: 12

You are building a model to predict daily temperatures. You split the data randomly and then transformed the training and test datasets. Temperature data for model training is uploaded hourly. During testing, your model performed with 97% accuracy; however, after deploying to production, the model's accuracy dropped to 66%. How can you make your production model more accurate?

A. Normalize the data for the training, and test datasets as two separate steps.

B. Split the training and test data based on time rather than a random split to avoid leakage

C. Add more data to your test set to ensure that you have a fair distribution and sample for testing

D. Apply data transformations before splitting, and cross-validate to make sure that the transformations are applied to both the training and test sets.

ANSWER: B

Explanation:

Split the training and test data based on time rather than a random split to avoid leakage is correct because temperature prediction is a time-dependent forecasting problem. In production, a prediction for a given day can use only information available before that prediction time. A random split can place observations from later periods into the training dataset while earlier observations appear in the test dataset. This lets the evaluation benefit from future temporal patterns and produces an overly optimistic offline result, such as the reported accuracy. The production system cannot access that future information, so its measured accuracy is substantially lower.

A chronological split instead trains on an earlier contiguous time period and evaluates on a later holdout period. This simulates the actual deployment workflow: historical hourly uploads are used to predict future daily temperatures. The same approach should be used for validation and model selection, ideally with rolling or walk-forward validation windows when sufficient history exists. Any learned transformation parameters, such as normalization statistics, should be fit using only each training window and then applied to its corresponding later validation or test window. This preserves the boundary between past and future data and makes offline evaluation a reliable estimate of production performance. Google's ML guidance emphasizes preventing training-serving skew, while Vertex AI forecasting guidance describes preparing time-series data for forecasting workflows. See Google Rules of ML and Vertex AI forecasting data preparation documentation.

QUESTION NO: 13

You are an AI architect at a popular photo-sharing social media platform. Your organization's content moderation team currently scans images uploaded by users and removes explicit images manually. You want to implement an AI service to automatically prevent users from uploading explicit images. What should you do?

A. Develop a custom TensorFlow model in a Vertex AI Workbench instance. Train the model on a dataset of manually labeled images. Deploy the model to a Vertex AI endpoint. Run periodic batch inference to identify inappropriate uploads and report them to the content moderation team.

B. Train an image clustering model using TensorFlow in a Vertex AI Workbench instance. Deploy this model to a Vertex AI endpoint and configure it for online inference. Run this model each time a new image is uploaded to identify and block inappropriate uploads.

C. Create a dataset using manually labeled images. Ingest this dataset into AutoML. Train an image classification model and deploy it to a Vertex AI endpoint. Integrate this endpoint with the image upload process to identify and block inappropriate uploads. Monitor predictions and periodically retrain the model.

D. Send a copy of every user-uploaded image to a Cloud Storage bucket. Configure a Cloud Run function that triggers the Cloud Vision API to detect explicit content each time a new image is uploaded. Report the classifications to the content moderation team for review.

ANSWER: D

Explanation:

Send a copy of every user-uploaded image to a Cloud Storage bucket. Configure a Cloud Run function that triggers the Cloud Vision API to detect explicit content each time a new image is uploaded. Report the classifications to the content moderation team for review. This approach uses Cloud Vision SafeSearch detection, a managed, pre-trained capability specifically intended to evaluate image attributes such as adult, racy, violent, and medical content. The API returns likelihood values for these categories, allowing the upload workflow to apply defined moderation thresholds. For example, the service can prevent an image from being published when its adult-content likelihood exceeds the organization's acceptable threshold, while routing borderline cases to human reviewers. Cloud Run is appropriate for implementing the event-driven processing and policy logic around the managed API, and Cloud Storage provides a durable event source for uploaded image objects. Because the model is already trained for this general moderation task, the organization can implement scalable automated screening without collecting, labeling, training, evaluating, and operating a custom image-classification model. Google documents SafeSearch detection and its likelihood-based responses in the Cloud Vision SafeSearch

detection guide. Cloud Storage events can be used to invoke Cloud Run services through Cloud Storage triggers for Cloud Run.

QUESTION NO: 14

You work for an online retail company that is creating a visual search engine. You have set up an end-to-end ML pipeline on Google Cloud to classify whether an image contains your company's product. Expecting the release of new products in the near future, you configured a retraining functionality in the pipeline so that new data can be fed into your ML models. You also want to use AI Platform's continuous evaluation service to ensure that the models have high accuracy on your test data set. What should you do?

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

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

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

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

ANSWER: B

Explanation:

Extend your test dataset with images of the newer products when they are introduced to retraining. Continuous evaluation is useful only when its evaluation data continues to represent the production task and the range of classes or product appearances that the model is expected to recognize. As the retailer introduces products, these images constitute a meaningful expansion of the prediction domain. Adding representative, correctly labeled examples to the held-out test set lets each retrained model be assessed for accuracy across both the established catalog and the newly introduced products. This produces an evaluation signal that is comparable over time while also measuring whether the updated model meets the actual business requirement.

The added test examples should remain separate from the examples used to train or tune the model, so that the evaluation remains unbiased. In practice, the pipeline should version the datasets and evaluation results, allowing the team to identify changes in quality as the catalog evolves. Google’s guidance for monitoring and evaluating models emphasizes using representative evaluation data and continuously checking model quality; Vertex AI Model Monitoring also supports ongoing quality monitoring for deployed models. See Vertex AI model evaluation documentation and Vertex AI Model Monitoring overview.

QUESTION NO: 15

You have a large corpus of written support cases that can be classified into 3 separate categories: Technical Support, Billing Support, or Other Issues. You need to quickly build, test, and deploy a service that will automatically classify future written requests into one of the categories. How should you configure the pipeline?

A. Use the Cloud Natural Language API to obtain metadata to classify the incoming cases.

B. Use AutoML Natural Language to build and test a classifier. Deploy the model as a REST API.

C. Use BigQuery ML to build and test a logistic regression model to classify incoming requests. Use BigQuery ML to perform inference.

D. Create a TensorFlow model using Google’s BERT pre-trained model. Build and test a classifier, and deploy the model using Vertex AI.

ANSWER: B

Explanation:

Use AutoML Natural Language to build and test a classifier. Deploy the model as a REST API. This approach is designed for supervised text classification when labeled examples already belong to a finite set of mutually exclusive categories. The historical support cases provide the training data, with each case labeled Technical Support, Billing Support, or Other Issues. An AutoML text-classification workflow automates much of the model-development process, including training and evaluation, allowing the team to produce a useful classifier quickly without designing and maintaining a custom natural-language model architecture.

After selecting the trained model based on its evaluation results, serving it through an online prediction endpoint enables an application or support-intake service to submit the text of each new request and receive a predicted category. This supports the requirement to deploy a classification service rather than merely analyze text in an offline environment. In the current Google Cloud product organization, AutoML text classification capabilities are available in Vertex AI, which provides managed training and online prediction endpoints. Google documents the AutoML text-classification workflow at Vertex AI text classification overview and endpoint deployment and online prediction at Deploy a model to an endpoint.

QUESTION NO: 16

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

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

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

C. Develop a template matching algorithm using traditional computer vision libraries.

D. Develop an image classification ML model to predict the presence of the disease.

ANSWER: B

Explanation:

Develop an image segmentation ML model to locate the boundaries of the rust spots. is the appropriate approach because disease severity depends not only on whether rust is present, but also on the spatial extent and morphology of the affected regions. Image segmentation assigns a class label to individual pixels, producing a mask that precisely identifies each rust-affected area and its boundary. From these masks, the research team can calculate measurements such as total diseased-pixel area, proportion of the leaf affected, number of separate lesions, and lesion shapes. Those features can support a severity score while also providing a direct disease-presence prediction.

This pixel-level representation is especially useful when spots have irregular outlines and variable sizes, because it captures the actual affected tissue rather than a coarse image-level label. The resulting masks are also interpretable: agronomists can review the highlighted regions and verify that the model is measuring symptoms rather than unrelated image features. Google Cloud describes image segmentation as identifying and classifying pixels in an image, including use cases that require detailed delineation of objects or regions. See the Vertex AI image segmentation overview and the Vertex AI guidance for preparing image-segmentation data.

QUESTION NO: 17

You have trained a deep neural network model on Google Cloud. The model has low loss on the training data, but is performing worse on the validation data. You want the model to be resilient to overfitting. Which strategy should you use when retraining the model?

A. Apply a dropout parameter of 0.2, and decrease the learning rate by a factor of 10

B. Apply a L2 regularization parameter of 0.4, and decrease the learning rate by a factor of 10.

C. Run a hyperparameter tuning job on AI Platform to optimize for the L2 regularization and dropout parameters

D. Run a hyperparameter tuning job on AI Platform to optimize for the learning rate, and increase the number of neurons by a factor of 2.

ANSWER: C

Explanation:

Run a hyperparameter tuning job on AI Platform to optimize for the L2 regularization and dropout parameters. The gap between low training loss and poorer validation performance is the characteristic signal that the network is fitting details of the training set that do not generalize. L2 regularization and dropout are both directly intended to control model complexity during training: L2 adds a penalty for large weights, while dropout randomly omits units during training so the model cannot depend too heavily on particular activations. Their effective values depend on the architecture, dataset size, feature distribution, and training setup, so selecting a fixed value without validation-based experimentation is not a robust approach. A hyperparameter tuning job evaluates candidate parameter combinations against a validation objective and can identify the regularization/dropout balance that improves generalization for this specific model. Google Cloud's hyperparameter tuning workflow is designed to run multiple trials and select values using the metric reported by the training application. TensorFlow also describes dropout and regularization as techniques for reducing overfitting. See Vertex AI hyperparameter tuning overview and TensorFlow: Overfit and underfit.

QUESTION NO: 18

You work on a growing team of more than 50 data scientists who all use AI Platform. You are designing a strategy to organize your jobs, models, and versions in a clean and scalable way. Which strategy should you choose?

A. Set up restrictive IAM permissions on the AI Platform notebooks so that only a single user or group can access a given instance.

B. Separate each data scientist's work into a different project to ensure that the jobs, models, and versions created by each data scientist are accessible only to that user.

C. Use labels to organize resources into descriptive categories. Apply a label to each created resource so that users can filter the results by label when viewing or monitoring the resources.

D. Set up a BigQuery sink for Cloud Logging logs that is appropriately filtered to capture information about AI Platform resource usage. In BigQuery, create a SQL view that maps users to the resources they are using.

ANSWER: C

Explanation:

Use labels to organize resources into descriptive categories. Apply a label to each created resource so that users can filter the results by label when viewing or monitoring the resources. Labels are key-value metadata that can be attached to AI Platform Prediction resources, including training jobs, models, and model versions. This provides a lightweight, scalable organizational structure without requiring a separate project or isolated environment for every individual data scientist. For a large team, a consistent label taxonomy—such as `owner`, `team`, `environment`, `cost_center`, or `use_case`—lets users locate related resources efficiently in the console and through API-based workflows. Labels also support resource governance and operational visibility because teams can filter and group resources according to the metadata most relevant to their process. Applying the labels when resources are created ensures that the organization remains reliable as the number of jobs, models, and versions grows. Google documents labels as the intended mechanism for organizing AI Platform Prediction resources into categories for viewing and monitoring. See AI Platform Prediction resource labels and Google Cloud labels overview.

QUESTION NO: 19

You are evaluating a binary classification model that identifies potentially fraudulent transactions. Fraud represents less than 0.2% of transactions, and investigators can review only a limited number of alerts each day. You need to select a model and decision threshold that identify as much fraud as possible while keeping alert volume within the investigation capacity. Which actions should you take?

Choose 2 answers

A. Use a precision-recall curve to compare candidate models

B. Select a threshold using the predicted alert volume, precision, and recall at each threshold

C. Select the model with the highest overall accuracy

D. Use a threshold of 0.5 because the model produces probability scores

E. Balance the evaluation dataset by removing negative transactions before calculating all metrics

ANSWER: A B

Explanation:

Use a precision-recall curve to evaluate the model because precision and recall are more informative than accuracy when the positive class is rare. The curve shows how precision and recall change as the classification threshold changes.

Select a threshold by evaluating the expected number of predicted positive transactions, precision, and recall at candidate thresholds. This allows the team to choose a threshold that fits the daily investigator capacity while maximizing the rate at which actual fraud is identified. See the Google Machine Learning Crash Course: Precision and recall.

QUESTION NO: 20

Your team retrains a custom model on Vertex AI each month. The team must be able to reproduce any production model by identifying the exact training container, source revision, dataset version, hyperparameters, and evaluation metrics used for the training run. Which actions should you take?

Choose 2 answers

A. Reference an immutable container image digest when submitting each training job

B. Use Vertex AI Experiments to log training parameters, metrics, and artifacts for each run

C. Overwrite the same Cloud Storage object whenever a new training dataset is available

D. Use the latest container image tag so that every run uses the newest dependencies

E. Store only the final model evaluation metric in a spreadsheet

ANSWER: A B

Explanation:

Use immutable container image references, such as an Artifact Registry image digest, for training jobs. A mutable tag such as `latest` can refer to different image contents at different times, while an immutable digest identifies the exact training environment used by a run.

Use Vertex AI Experiments to log parameters, metrics, and artifacts associated with each training run. An experiment run can record the configuration and outcomes needed to compare runs and establish training lineage. The dataset version and source revision should also be recorded as run parameters or artifacts. See the Vertex AI Experiments overview and Artifact Registry container image documentation.

QUESTION NO: 21

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. Decrease the score threshold.

C. Add more positive examples to the training set.

D. Add more negative examples to the training set.

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

ANSWER: B C

Explanation:

Decrease the score threshold. is appropriate because the stated business objective is to avoid missed fraud. In a binary classifier, the score threshold determines how much predicted confidence is required before a transaction is classified as fraudulent. Lowering that threshold causes more transactions to be flagged, increasing recall: a greater proportion of the truly fraudulent transactions are detected. This directly reduces false negatives, which are the especially costly errors in this scenario. The resulting operational trade-off is that more legitimate transactions can be reviewed or temporarily blocked, but this is aligned with prioritizing detection.

Add more positive examples to the training set. is also appropriate because fraud is typically a rare target class. Providing additional representative, accurately labeled fraudulent transactions gives AutoML more evidence from which to learn the patterns associated with fraud. Improving the representation of the positive class can improve the model's ability to identify fraud during evaluation and production use. Google's classification guidance describes choosing a threshold based on the business costs of false positives and false negatives, while Vertex AI documentation emphasizes evaluating classification models with metrics such as recall and using representative training data. See Vertex AI classification evaluation overview and Google Machine Learning Crash Course: Thresholding.

QUESTION NO: 22

You are analyzing customer data for a healthcare organization that is stored in Cloud Storage. The data contains personally identifiable information (PII) You need to perform data exploration and preprocessing while ensuring the security and privacy of sensitive fields What should you do?

A. Use the Cloud Data Loss Prevention (DLP) API to de-identify the PI! before performing data exploration and preprocessing.

B. Use customer-managed encryption keys (CMEK) to encrypt the Pll data at rest and decrypt the Pll data during data exploration and preprocessing.

C. Use a VM inside a VPC Service Controls security perimeter to perform data exploration and preprocessing.

D. Use Google-managed encryption keys to encrypt the Pll data at rest, and decrypt the Pll data during data exploration and preprocessing.

ANSWER: A

Explanation:

Use the Cloud Data Loss Prevention (DLP) API to de-identify the PII before performing data exploration and preprocessing. De-identification transforms sensitive values so that analysts and preprocessing systems can work with useful data while reducing exposure of direct identifiers. Sensitive fields can be masked, replaced, generalized, or transformed using techniques such as tokenization, date shifting, and cryptographic hashing. For healthcare data, Cloud DLP can also inspect content using built-in and custom detectors to locate PII and other sensitive data before applying the appropriate transformation. This supports a privacy-by-design workflow: retain only the information needed for exploratory analysis and feature preparation, while keeping identifiable values unavailable in the working dataset. Cloud DLP supports reversible transformations, including format-preserving encryption and surrogate-token replacement, when an approved, tightly controlled re-identification process is genuinely required. The source data should remain protected with Cloud Storage IAM and other applicable security controls, but de-identifying the data is the measure that directly minimizes disclosure of sensitive fields during analysis. See the Sensitive Data Protection de-identification documentation and the de-identification concepts guide.

QUESTION NO: 23

You are investigating the root cause of a misclassification error made by one of your models. You used Vertex AI Pipelines to tram and deploy the model. The pipeline reads data from BigQuery. creates a copy of the data in Cloud Storage in TFRecord format trains the model in Vertex AI Training on that copy, and deploys the model to a Vertex AI endpoint. You have identified the specific version of that model that misclassified: and you need to recover the data this model was trained on. How should you find that copy of the data ' ?

A. Use Vertex AI Feature Store Modify the pipeline to use the feature store; and ensure that all training data is stored in it Search the feature store for the data used for the training.

B. Use the lineage feature of Vertex AI Metadata to find the model artifact Determine the version of the model and identify the step that creates the data copy, and search in the metadata for its location.

C. Use the logging features in the Vertex AI endpoint to determine the timestamp of the models deployment Find the pipeline run at that timestamp Identify the step that creates the data copy; and search in the logs for its location.

D. Find the job ID in Vertex AI Training corresponding to the training for the model Search in the logs of that job for the data used for the training.

ANSWER: B

Explanation:

Use the lineage feature of Vertex AI Metadata to find the model artifact Determine the version of the model and identify the step that creates the data copy, and search in the metadata for its location. Vertex AI Pipelines automatically records pipeline execution metadata, including the artifacts consumed and produced by pipeline steps and the relationships between those artifacts. This lineage is designed for precisely this kind of reproducibility and debugging task: starting with the known deployed model version, trace backward through the model artifact and its training execution to the input dataset artifact. The data-copy component's output artifact contains the Cloud Storage URI for the TFRecord copy that was produced during that particular pipeline run. This preserves the connection to the immutable, run-specific training-data copy rather than relying on deployment timing or potentially incomplete application logs. Vertex AI Model Registry and metadata lineage allow teams to inspect how a model was created, including its associated pipeline executions, inputs, outputs, and upstream artifacts. Reviewing this provenance enables recovery of the exact dataset copy used to train the misclassifying model and supports an auditable root-cause investigation. See Vertex AI pipeline lineage documentation and Vertex ML Metadata documentation.

QUESTION NO: 24

Your team uses Vertex AI Pipelines to train and evaluate several candidate models each week. Only a model that meets a minimum recall requirement and improves precision compared with the currently deployed model should be eligible for deployment. Which actions should you take?

Choose 2 answers

A. Add a pipeline evaluation component that calculates recall and precision on a held-out dataset

B. Add a conditional deployment task that runs only when the required metric criteria are met

C. Deploy every candidate model and compare production metrics later

D. Use training accuracy as the only condition for deployment

E. Replace the held-out evaluation dataset with the training dataset before model promotion

ANSWER: A B

Explanation:

Add an evaluation component that calculates recall and precision on a held-out evaluation dataset. The component should write the measured metrics as pipeline artifacts so subsequent steps can use them for an automated decision.

Add a conditional deployment step that executes only when the candidate satisfies the minimum recall threshold and the required precision improvement. This creates an automated promotion gate and prevents a model from being deployed

solely because training completed successfully. See the Vertex AI Pipelines overview and the Kubeflow Pipelines control flow documentation.

QUESTION NO: 25

Your organization needs to reduce cost for a nightly prediction workload that scores 80 million records from BigQuery. Predictions do not need to be available immediately, and the output must be written to Cloud Storage for downstream processing. Which actions should you take?

Choose 2 answers

A. Run a Vertex AI batch prediction job using the nightly dataset.

B. Schedule the prediction workflow after the nightly data refresh completes.

C. Deploy the model to an online endpoint with enough replicas to process all records simultaneously.

D. Invoke the endpoint once for every record from a Cloud Function.

E. Download the BigQuery data to a local workstation before running predictions.

ANSWER: A B

Explanation:

Use a Vertex AI batch prediction job because the workload is a large, bounded collection of records that can be processed asynchronously. Batch prediction avoids the cost of maintaining continuously provisioned online-serving capacity for a workload that runs only once per night.

Schedule the job through an orchestration service such as Vertex AI Pipelines or Cloud Scheduler with a workflow invocation. Scheduling provides repeatable, automated execution after the nightly source data is ready and removes the need for manual job submission. See the Vertex AI batch prediction documentation and the Vertex AI Pipelines overview.

QUESTION NO: 26

You work for an advertising company and want to understand the effectiveness of your company's latest advertising campaign. You have streamed 500 MB of campaign data into BigQuery. You want to query the table, and then manipulate the results of that query with a pandas dataframe in an AI Platform notebook. What should you do?

A. Use AI Platform Notebooks' BigQuery cell magic to query the data, and ingest the results as a pandas dataframe

B. Export your table as a CSV file from BigQuery to Google Drive, and use the Google Drive API to ingest the file into your notebook instance

C. Download your table from BigQuery as a local CSV file, and upload it to your AI Platform notebook instance Use pandas. read_csv to ingest the file as a pandas dataframe

D. From a bash cell in your AI Platform notebook, use the bq extract command to export the table as a CSV file to Cloud Storage, and then use gsutii cp to copy the data into the notebook Use pandas. read_csv to ingest the file as a pandas dataframe

ANSWER: A

Explanation:

Use AI Platform Notebooks' BigQuery cell magic to query the data, and ingest the results as a pandas dataframe. This is the direct, notebook-native workflow for running a SQL query in BigQuery and loading its result set into a pandas DataFrame for interactive analysis. The BigQuery IPython magic extension supports SQL execution through `%%bigquery` and can assign the query output directly to a Python variable, making the returned data immediately available for pandas operations such as filtering, aggregation, visualization, and feature preparation. For a 500 MB dataset, querying in BigQuery first is especially appropriate because BigQuery performs the SQL processing server-side and only the query results need to be brought into

the notebook environment. This avoids unnecessary manual export, download, upload, and file-parsing steps. The notebook identity must have permission to run the query and access the relevant BigQuery resources; if the BigQuery Storage API is used for higher-throughput result downloads, the appropriate API and permissions must also be enabled. See the BigQuery query documentation and the BigQuery IPython magics reference.

QUESTION NO: 27

You are developing an image recognition model using PyTorch based on ResNet50 architecture. Your code is working fine on your local laptop on a small subsample. Your full dataset has 200k labeled images You want to quickly scale your training workload while minimizing cost. You plan to use 4 V100 GPUs. What should you do? (Choose Correct Answer and Give References and Explanation)

A. Configure a Compute Engine VM with all the dependencies that launches the training. Train your model with Vertex AI using a custom tier that contains the required GPUs.

B. Package your code with Setuptools and use a pre-built PyTorch container. Train your model with Vertex AI using a custom machine configuration containing the required GPUs.

C. Create a Vertex AI Workbench user-managed notebooks instance with 4 V100 GPUs, and use it to train your model.

D. Create a Google Kubernetes Engine cluster with a node pool that has 4 V100 GPUs. Prepare and submit a TFJob operator to this node pool.

ANSWER: B

Explanation:

Package your code with Setuptools and use a pre-built PyTorch container. Train your model with Vertex AI using a custom machine configuration containing the required GPUs. This approach is appropriate because Vertex AI custom training provides managed infrastructure for running training applications without requiring the team to operate a long-running virtual machine, notebook environment, or Kubernetes cluster. A Python source distribution created with Setuptools provides a portable, repeatable way to submit the PyTorch training application and its entry point to the managed training service.

Vertex AI supplies pre-built training containers for supported frameworks, including PyTorch. Using one avoids the operational work of building, securing, versioning, and maintaining a framework image when the standard runtime meets the application's needs. The custom job worker-pool configuration can then specify a compatible machine type, the NVIDIA Tesla V100 accelerator type, and an accelerator count of four. This allocates GPU capacity for the job only while it runs, which supports rapid scale-out while avoiding the idle-resource costs associated with keeping self-managed compute available. The training code should use PyTorch multi-GPU execution, such as distributed data parallelism, to use all four GPUs effectively. See Vertex AI pre-built training containers and Configure compute resources for Vertex AI training.

QUESTION NO: 28

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

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

B. Use Vertex AI default data split.

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

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

ANSWER: A

Explanation:

Use Vertex AI manual split, using the store name feature to assign one store for each set. The deployment objective is to predict sales for a store that was not represented during model development. Therefore, the evaluation data must represent that situation: rows from a given store should be kept together in one split rather than distributed among training, validation, and test data. Assigning splits by store name prevents the model from benefiting during evaluation from patterns specific to a store that it has already seen in training, such as its baseline demand, local customer behavior, or operational characteristics. This produces a more meaningful estimate of how well the model can generalize to the newly opened store. Vertex AI supports manual data splitting through a split column, allowing the data producer to explicitly designate records for training, validation, and test use. In this case, create a split-assignment column based on the store identifier and use it as the manual split column when training. This applies a group-aware evaluation design while retaining the existing managed tabular dataset. See Google Cloud's Prepare tabular data documentation and Vertex AI data split documentation.

QUESTION NO: 29

You want to train an AutoML model to predict house prices by using a small public dataset stored in BigQuery. You need to prepare the data and want to use the simplest most efficient approach. What should you do?

A. Write a query that preprocesses the data by using BigQuery and creates a new table. Create a Vertex AI managed dataset with the new table as the data source.

B. Use Dataflow to preprocess the data. Write the output in TFRecord format to a Cloud Storage bucket.

C. Write a query that preprocesses the data by using BigQuery. Export the query results as CSV files and use those files to create a Vertex AI managed dataset.

D. Use a Vertex AI Workbench notebook instance to preprocess the data by using the pandas library. Export the data as CSV files, and use those files to create a Vertex AI managed dataset.

ANSWER: A

Explanation:

Write a query that preprocesses the data by using BigQuery and creates a new table. Create a Vertex AI managed dataset with the new table as the data source. This is the simplest and most efficient approach because the source data is already in BigQuery, so SQL can perform the necessary preparation in place without creating an additional processing pipeline or exporting intermediate files. For a house-price regression model, a query can select the training label, filter invalid or incomplete records, join relevant tables if needed, derive features, and materialize the resulting training-ready data in a BigQuery table. BigQuery is serverless, so this avoids managing compute infrastructure for a small public dataset.

Vertex AI supports creating tabular datasets from BigQuery sources. This provides a direct, managed path from the prepared table into AutoML training while retaining the schema and avoiding Cloud Storage file-export steps. Keeping the prepared data in BigQuery also makes the workflow easier to reproduce: the SQL query defines the transformations, and the resulting table is a stable source for model development. Google documents BigQuery as a supported source for Vertex AI tabular datasets and describes preparing tabular data for training in its Vertex AI tabular data preparation guide. The direct integration is also covered in the Vertex AI BigQuery dataset documentation.

QUESTION NO: 30

You are preparing a tabular dataset in BigQuery for a churn model. A customer can have multiple monthly records, and you want to predict whether the customer will churn during the following month. You need to prevent leakage while creating training and evaluation data. Which actions should you take?

Choose 2 answers

A. Split the dataset so that evaluation records are from a later time period than training records.

B. Calculate features by using only information available before each prediction timestamp.

C. Randomly distribute all customer-month records across the training and evaluation datasets.

D. Include customer activity from the month after the prediction timestamp as an input feature.

E. Fit normalization statistics independently on the training and evaluation datasets.

ANSWER: A B

Explanation:

Use a time-based split in which all evaluation records occur after the training period. This simulates the production task, where predictions are made from past information for a future outcome. It prevents later customer behavior from being used to assess earlier predictions.

Calculate every feature using only data available at the prediction timestamp. For example, a feature representing support-ticket count must include tickets opened before the prediction date, not tickets created during the following month when churn is being measured. This avoids target leakage and makes offline metrics more representative of production performance. See the Google Rules of ML and the Vertex AI time-series data preparation documentation.

QUESTION NO: 31

You need to quickly build and train a model to predict the sentiment of customer reviews with custom categories without writing code. You do not have enough data to train a model from scratch. The resulting model should have high predictive performance. Which service should you use?

A. AutoML Natural Language

B. Cloud Natural Language API

C. AI Hub pre-made Jupyter Notebooks

D. AI Platform Training built-in algorithms

ANSWER: A

Explanation:

AutoML Natural Language is the appropriate service because it is designed for creating custom text-classification models through a managed, no-code workflow. For customer-review sentiment with business-specific categories, you can provide labeled examples for the categories that matter to the organization rather than being limited to a fixed, general-purpose sentiment vocabulary. The service automatically handles core model-development activities, including training, evaluation, and deployment, allowing a model to be built quickly without writing training code.

It is also a strong fit when the available labeled dataset is not sufficient for building a high-performing language model from scratch. AutoML text classification uses transfer learning: it starts from language knowledge learned during pretraining and adapts that capability to the supplied labeled data. This generally requires less task-specific data and training effort than developing and training a custom NLP architecture yourself, while still producing a model tailored to the review domain and custom labels. Google now provides this capability in Vertex AI AutoML for text classification; “AutoML Natural Language” is the legacy product name. See the Vertex AI AutoML training overview and text classification data preparation documentation.

QUESTION NO: 32

You have deployed a churn model to a Vertex AI endpoint. The model's labels are available 60 days after a prediction, but you want to detect input-data drift immediately and notify the operations team when drift exceeds an approved threshold. Which actions should you take?

Choose 2 answers

A. Configure Vertex AI Model Monitoring with baseline feature distributions from the training data

B. Create Cloud Monitoring alerting policies for monitoring-threshold violations

C. Wait for 60-day labels before monitoring production feature distributions

D. Use only training loss to detect changes in production input data

E. Retrain the model after every online prediction request

ANSWER: A B

Explanation:

Configure Vertex AI Model Monitoring with an appropriate baseline distribution from the training data. Model Monitoring can compare production feature distributions with baseline distributions and detect drift before delayed ground-truth labels are available.

Configure alerting through Cloud Monitoring so that the operations team is notified when monitoring results exceed the defined threshold. This enables the team to investigate changes in production inputs and determine whether retraining or data-pipeline remediation is required. See the Vertex AI Model Monitoring overview and the Cloud Monitoring alerting documentation.

QUESTION NO: 33

You are training a Resnet model on AI Platform using TPUs to visually categorize types of defects in automobile engines. You capture the training profile using the Cloud TPU profiler plugin and observe that it is highly input-bound. You want to reduce the bottleneck and speed up your model training process. Which modifications should you make to the tf .data dataset?

Choose 2 answers

A. Use the interleave option for reading data

B. Reduce the value of the repeat parameter

C. Increase the buffer size for the shuffle option.

D. Use the prefetch option with `tf.data.AUTOTUNE`

E. Decrease the batch size argument in your transformation

ANSWER: A D

Explanation:

An input-bound TPU profile means that accelerator cores spend time waiting for the next training batch rather than executing model computations. The input pipeline should therefore overlap data reading, preprocessing, and model execution as much as possible. **Use the interleave option for reading data** is appropriate because interleaving can read from multiple input files concurrently. This improves throughput when data is split across files, reduces delays caused by sequential file access, and keeps batches flowing to the TPU more consistently.

Use the prefetch option with `tf.data.AUTOTUNE` is also appropriate because prefetching prepares future dataset elements while the TPU processes the current batch. TensorFlow can tune the prefetch buffer automatically according to available resources and pipeline behavior. Prefetching operates on the elements produced by the preceding dataset transformation; when batching occurs before prefetching, those elements are complete batches. Together, parallel interleaving and autotuned prefetching are standard `tf.data` performance techniques for reducing host-side input latency and increasing accelerator utilization. See the TensorFlow guidance on optimizing input-pipeline performance and the Dataset.interleave API reference.

QUESTION NO: 34

You have a regulated credit-risk model that serves online predictions from Vertex AI. Auditors require the team to identify the features that contributed most to an individual prediction and to retain the model version and input data version used for each deployment. Which actions should you take?

Choose 2 answers

A. Enable Vertex Explainable AI to obtain feature attributions for online predictions.

B. Version and retain model, data, code, evaluation, and deployment metadata.

C. Remove all feature names from the training dataset before deployment.

D. Use only aggregate accuracy metrics to explain individual decisions.

E. Retrain the model whenever an auditor requests a prediction explanation.

ANSWER: A B

Explanation:

Enable Vertex Explainable AI for the deployed model to obtain feature-attribution values for individual predictions. Feature attributions help explain how input features influenced a particular prediction, which supports review of high-impact model decisions.

Record and version the model artifact, training code, training dataset, feature definitions, evaluation results, and deployment metadata. This creates the lineage necessary to reproduce a released model and trace a prediction system back to the data and configuration from which it was produced. See the Vertex Explainable AI overview and the Vertex ML Metadata documentation.