Product Questions: 283
Version: 10.6

Question: 1

As the lead ML Engineer for your company, you are responsible for building ML models to digitize scanned customer forms. You have developed a TensorFlow model that converts the scanned images into text and stores them in Cloud Storage. You need to use your ML model on the aggregated data collected at the end of each day with minimal manual intervention. What should you do?

A. Use the batch prediction functionality of Al Platform
B. Create a serving pipeline in Compute Engine for prediction
C. Use Cloud Functions for prediction each time a new data point is ingested
D. Deploy the model on Al Platform and create a version of it for online inference.

Answer: A

Explanation:
Batch prediction is the process of using an ML model to make predictions on a large set of data points. Batch prediction is suitable for scenarios where the predictions are not time-sensitive and can be done in batches, such as digitizing scanned customer forms at the end of each day. Batch prediction can also handle large volumes of data and scale up or down the resources as needed. AI Platform provides a batch prediction service that allows users to submit a job with their TensorFlow model and input data stored in Cloud Storage, and receive the output predictions in Cloud Storage as well. This service requires minimal manual intervention and can be automated with Cloud Scheduler or Cloud Functions. Therefore, using the batch prediction functionality of AI Platform is the best option for this use case.
Reference:
Batch prediction overview
Using batch prediction

Question: 2

You work for a global footwear retailer and need to predict when an item will be out of stock based on historical inventory dat
a. Customer behavior is highly dynamic since footwear demand is influenced by many different factors. You want to serve models that are trained on all available data, but track your performance on specific subsets of data before pushing to production. What is the most streamlined and reliable way to perform this validation?
A. Use the TFX ModelValidator tools to specify performance metrics for production readiness

B. Use k-fold cross-validation as a validation strategy to ensure that your model is ready for production.
C. Use the last relevant week of data as a validation set to ensure that your model is performing accurately on current data
D. Use the entire dataset and treat the area under the receiver operating characteristics curve (AUC ROC) as the main metric.

Answer: A

Explanation:
TFX ModelValidator is a tool that allows you to compare new models against a baseline model and evaluate their performance on different metrics and data slices1. You can use this tool to validate your models before deploying them to production and ensure that they meet your expectations and requirements.
k-fold cross-validation is a technique that splits the data into k subsets and trains the model on k-1 subsets while testing it on the remaining subset. This is repeated k times and the average performance is reported2. This technique is useful for estimating the generalization error of a model, but it does not account for the dynamic nature of customer behavior or the potential changes in data distribution over time.
Using the last relevant week of data as a validation set is a simple way to check the model's performance on recent data, but it may not be representative of the entire data or capture the long-term trends and patterns. It also does not allow you to compare the model with a baseline or evaluate it on different data slices.
Using the entire dataset and treating the AUC ROC as the main metric is not a good practice because it does not leave any data for validation or testing. It also assumes that the AUC ROC is the only metric that matters, which may not be true for your business problem. You may want to consider other metrics such as precision, recall, or revenue.

Question: 3

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 I AM 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:
Labels are key-value pairs that can be attached to any AI Platform resource, such as jobs, models, versions, or endpoints1. Labels can help you organize your resources into descriptive categories, such as project, team, environment, or purpose. You can use labels to filter the results when you list or monitor your resources, or to group them for billing or quota purposes2. Using labels is a simple and scalable way to manage your AI Platform resources without creating unnecessary complexity or overhead. Therefore, using labels to organize resources is the best strategy for this use case.
Reference:
Using labels
Filtering and grouping by labels

Question: 4

During batch training of a neural network, you notice that there is an oscillation in the loss. How should you adjust your model to ensure that it converges?

A. Increase the size of the training batch
B. Decrease the size of the training batch
C. Increase the learning rate hyperparameter
D. Decrease the learning rate hyperparameter

Answer: D

Explanation:
Oscillation in the loss during batch training of a neural network means that the model is overshooting the optimal point of the loss function and bouncing back and forth. This can prevent the model from converging to the minimum loss value. One of the main reasons for this phenomenon is that the learning rate hyperparameter, which controls the size of the steps that the model takes along the gradient, is too high. Therefore, decreasing the learning rate hyperparameter can help the model take smaller and more precise steps and avoid oscillation. This is a common technique to improve the stability and performance of neural network training12.
Reference:
Interpreting Loss Curves
Is learning rate the only reason for training loss oscillation after few epochs?

Question: 5

You are building a linear model with over 100 input features, all with values between -1 and 1. You suspect that many features are non-informative. You want to remove the non-informative features from your model while keeping the informative ones in their original form. Which technique should you use?

A. Use Principal Component Analysis to eliminate the least informative features.
B. Use L1 regularization to reduce the coefficients of uninformative features to 0.
C. After building your model, use Shapley values to determine which features are the most

informative.
D. Use an iterative dropout technique to identify which features do not degrade the model when removed.

Answer: B

Explanation:
L1 regularization, also known as Lasso regularization, adds the sum of the absolute values of the model's coefficients to the loss function1. It encourages sparsity in the model by shrinking some coefficients to precisely zero2. This way, L1 regularization can perform feature selection and remove the non-informative features from the model while keeping the informative ones in their original form. Therefore, using L1 regularization is the best technique for this use case.
Reference:
Regularization in Machine Learning - GeeksforGeeks
Regularization in Machine Learning (with Code Examples) - Dataquest
L1 And L2 Regularization Explained & Practical How To Examples
L1 and L2 as Regularization for a Linear Model

Question: 6

Your team has been tasked with creating an ML solution in Google Cloud to classify support requests for one of your platforms. You analyzed the requirements and decided to use TensorFlow to build the classifier so that you have full control of the model's code, serving, and deployment. You will use Kubeflow pipelines for the ML platform. To save time, you want to build on existing resources and use managed services instead of building a completely new model. How should you build the classifier?

A. Use the Natural Language API to classify support requests
B. Use AutoML Natural Language to build the support requests classifier
C. Use an established text classification model on AI Platform to perform transfer learning
D. Use an established text classification model on AI Platform as-is to classify support requests

Answer: C

Explanation:
Transfer learning is a technique that leverages the knowledge and weights of a pre-trained model and adapts them to a new task or domain1. Transfer learning can save time and resources by avoiding training a model from scratch, and can also improve the performance and generalization of the model by using a larger and more diverse dataset2. AI Platform provides several established text classification models that can be used for transfer learning, such as BERT, ALBERT, or XLNet3. These models are based on state-of-the-art natural language processing techniques and can handle various text classification tasks, such as sentiment analysis, topic classification, or spam detection4. By using one of these models on AI Platform, you can customize the model's code, serving, and deployment, and use Kubeflow pipelines for the ML platform. Therefore, using an established text classification model on AI Platform to perform transfer learning is the best option for this use case.
Reference:

Transfer Learning - Machine Learning's Next Frontier
A Comprehensive Hands-on Guide to Transfer Learning with Real-World Applications in Deep Learning
Text classification models
Text Classification with Pre-trained Models in TensorFlow

Question: 7

Your team is working on an NLP research project to predict political affiliation of authors based on articles they have written. You have a large training dataset that is structured like this:

```
AuthorA:Political Party A
    TextA1: [SentenceA11, SentenceA12, SentenceA13, ...]
    TextA2: [SentenceA21, SentenceA22, SentenceA23, ...]
    ...
AuthorB:Political Party B
    TextB1: [SentenceB11, SentenceB12, SentenceB13, ...]
    TextB2: [SentenceB21, SentenceB22, SentenceB23, ...]
    ...
AuthorC:Political Party B
    TextC1: [SentenceC11, SentenceC12, SentenceC13, ...]
    TextC2: [SentenceC21, SentenceC22, SentenceC23, ...]
    ...
AuthorD:Political Party A
    TextD1: [SentenceD11, SentenceD12, SentenceD13, ...]
    TextD2: [SentenceD21, SentenceD22, SentenceD23, ...]
    ...
...
```

You followed the standard 80%-10%-10% data distribution across the training, testing, and evaluation subsets. How should you distribute the training examples across the train-test-eval subsets while maintaining the 80-10-10 proportion?

A)

Distribute texts randomly across the train-test-eval subsets:
Train set: [TextA1, TextB2, ...]
Test set: [TextA2, TextC1, TextD2, ...]
Eval set: [TextB1, TextC2, TextD1, ...]

B)

Distribute authors randomly across the train-test-eval subsets: (*)
Train set: [TextA1, TextA2, TextD1, TextD2, ...]
Test set: [TextB1, TextB2, ...]
Eval set: [TextC1, TextC2, ...]

C)
Distribute sentences randomly across the train-test-eval subsets:
Train set: [SentenceA11, SentenceA21, Sentence B11, SentenceB21, SentenceC11, SentenceD21, ...]
Test set: [SentenceA12, SentenceA22, Sentence B12, SentenceC22, SentenceC12, SentenceD22, ...]
Eval set: [SentenceA13, SentenceA23, Sentence B13, SentenceC23, SentenceC13, SentenceD31, ...]

D)
Distribute paragraphs of texts (i.e., chunks of consecutive sentences) across the train-test-eval subsets:
Train set: [SentenceA11, SentenceA12, Sentence D11, SentenceD12, ...]
Test set: [SentenceA13, SentenceB13, Sentence B21, SentenceD23, SentenceC12, SentenceD13, ...]
Eval set: [SentenceA11, SentenceA22, Sentence B13, SentenceD22, SentenceC23, SentenceD11, ...]

A. Option A
B. Option B
C. Option C
D. Option D

Answer: C

Explanation:
The best way to distribute the training examples across the train-test-eval subsets while maintaining the 80-10-10 proportion is to use option C. This option ensures that each subset contains a balanced and representative sample of the different classes (Democrat and Republican) and the different authors. This way, the model can learn from a diverse and comprehensive set of articles and avoid overfitting or underfitting. Option C also avoids the problem of data leakage, which occurs when the same author appears in more than one subset, potentially biasing the model and inflating its performance. Therefore, option C is the most suitable technique for this use case.

Question: 8

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

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

Sheets file, and query the results using the Google Sheets API

Answer: C

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

Question: 9

You are an ML engineer at a bank that has a mobile application. Management has asked you to build an ML-based biometric authentication for the app that verifies a customer's identity based on their fingerprint. Fingerprints are considered highly sensitive personal information and cannot be downloaded and stored into the bank databases. Which learning strategy should you recommend to train and deploy this ML model?

A. Differential privacy
B. Federated learning
C. MD5 to encrypt data
D. Data Loss Prevention API

Answer: B

Explanation:
Federated learning is a machine learning technique that enables organizations to train AI models on decentralized data without centralizing or sharing it1. It allows data privacy, continual learning, and better performance on end-user devices2. Federated learning works by sending the model parameters to the devices, where they are updated locally on the device's data, and then aggregating the updated parameters on a central server to form a global model3. This way, the data

never leaves the device and the model can learn from a large and diverse dataset.
Federated learning is suitable for the use case of building an ML-based biometric authentication for the bank's mobile app that verifies a customer's identity based on their fingerprint. Fingerprints are considered highly sensitive personal information and cannot be downloaded and stored into the bank databases. By using federated learning, the bank can train and deploy an ML model that can recognize fingerprints without compromising the data privacy of the customers. The model can also adapt to the variations and changes in the fingerprints over time and improve its accuracy and reliability. Therefore, federated learning is the best learning strategy for this use case.

Question: 10

You are building a linear regression model on BigQuery ML to predict a customer's likelihood of purchasing your company's products. Your model uses a city name variable as a key predictive component. In order to train and serve the model, your data must be organized in columns. You want to prepare your data using the least amount of coding while maintaining the predictable variables. What should you do?

A. Create a new view with BigQuery that does not include a column with city information
B. Use Dataprep to transform the state column using a one-hot encoding method, and make each city a column with binary values.
C. Use Cloud Data Fusion to assign each city to a region labeled as 1, 2, 3, 4, or 5r and then use that number to represent the city in the model.
D. Use TensorFlow to create a categorical variable with a vocabulary list Create the vocabulary file, and upload it as part of your model to BigQuery ML.

Answer: B

Explanation:
One-hot encoding is a technique that converts categorical variables into numerical variables by creating dummy variables for each possible category. Each dummy variable has a value of 1 if the original variable belongs to that category, and 0 otherwise1. One-hot encoding can help linear regression models to capture the effect of different categories on the target variable without imposing any ordinal relationship among them2. Dataprep is a service that allows you to explore, clean, and transform your data for analysis and machine learning. You can use Dataprep to apply one-hot encoding to your city name variable and make each city a column with binary values3. This way, you can prepare your data using the least amount of coding while maintaining the predictive variables. Therefore, using Dataprep to transform the state column using a one-hot encoding method is the best option for this use case.
Reference:
One Hot Encoding: A Beginner's Guide
One-Hot Encoding in Linear Regression Models
Dataprep documentation

Question: 11

You work for a toy manufacturer that has been experiencing a large increase in demand. You need to build an ML model to reduce the amount of time spent by quality control inspectors checking for product defects. Faster defect detection is a priority. The factory does not have reliable Wi-Fi. Your company wants to implement the new ML model as soon as possible. Which model should you use?

A. AutoML Vision model
B. AutoML Vision Edge mobile-versatile-1 model
C. AutoML Vision Edge mobile-low-latency-1 model
D. AutoML Vision Edge mobile-high-accuracy-1 model

Answer: C

Explanation:
AutoML Vision Edge is a service that allows you to create custom image classification and object detection models that can run on edge devices, such as mobile phones, tablets, or IoT devices1. AutoML Vision Edge offers four types of models that vary in size, accuracy, and latency: mobile-versatile-1, mobile-low-latency-1, mobile-high-accuracy-1, and mobile-core-ml-low-latency-12. Each model has its own trade-offs and use cases, depending on the device specifications and the application requirements.
For the use case of building an ML model to reduce the amount of time spent by quality control inspectors checking for product defects, the best model to use is the AutoML Vision Edge mobile-low-latency-1 model. This model is optimized for fast inference on mobile devices, with a latency of less than 50 milliseconds on a Pixel 1 phone2. Faster defect detection is a priority for the toy manufacturer, and the factory does not have reliable Wi-Fi, so a low-latency model that can run on the device without internet connection is ideal. The mobile-low-latency-1 model also has a small size of less than 4 MB, which makes it easy to deploy and update2. The mobile-low-latency-1 model has a slightly lower accuracy than the mobile-high-accuracy-1 model, but it is still suitable for most image classification tasks2. Therefore, the AutoML Vision Edge mobile-low-latency-1 model is the best option for this use case.
Reference:
AutoML Vision Edge documentation
AutoML Vision Edge model types

Question: 12

You are going to train a DNN regression model with Keras APIs using this code:

```
model = tf.keras.Sequential()
model.add(tf.keras.layers.Dense(
    256,
    use_bias=True,
    activation='relu',
    kernel_initializer=None,
    kernel_regularizer=None,
    input_shape=(500,)))
model.add(tf.keras.layers.Dropout(rate=0.25))
model.add(tf.keras.layers.Dense(
    128, use_bias=True,
    activation='relu',
    kernel_initializer='uniform',
    kernel_regularizer='l2'))
model.add(tf.keras.layers.Dropout(rate=0.25))
model.add(tf.keras.layers.Dense(
    2, use_bias=False,
    activation='softmax'))
model.compile(loss='mse')
```

How many trainable weights does your model have? (The arithmetic below is correct.)

A. 501*256+257*128+2 161154
B. 500*256+256*128+128*2 161024
C. 501*256+257*128+128*2161408
D. 500*256*0 25+256*128*0 25+128*2 40448

Answer: B

Explanation:
The number of trainable weights in a DNN regression model with Keras APIs can be calculated by multiplying the number of input units by the number of output units for each layer, and adding the number of bias units for each layer. The bias units are usually equal to the number of output units, except for the last layer, which does not have bias units if the activation function is softmax1. In this code, the model has three layers: a dense layer with 256 units and relu activation, a dropout layer with 0.25 rate, and a dense layer with 2 units and softmax activation. The input shape is 500.

Therefore, the number of trainable weights is:
For the first layer: 500 input units * 256 output units + 256 bias units 128256
For the second layer: The dropout layer does not have any trainable weights, as it only randomly sets some of the input units to zero to prevent overfitting2.
For the third layer: 256 input units * 2 output units + 0 bias units 512
The total number of trainable weights is 128256 + 512 161024. Therefore, the correct answer is B.
Reference:
How to calculate the number of parameters for a Convolutional Neural Network?
Dropout (keras.io)

Question: 13

You recently joined a machine learning team that will soon release a new project. As a lead on the project, you are asked to determine the production readiness of the ML components. The team has already tested features and data, model development, and infrastructure. Which additional readiness check should you recommend to the team?

A. Ensure that training is reproducible
B. Ensure that all hyperparameters are tuned
C. Ensure that model performance is monitored
D. Ensure that feature expectations are captured in the schema

Answer: C

Explanation:
Monitoring model performance is an essential part of production readiness, as it allows the team to detect and address any issues that may arise after deployment, such as data drift, model degradation, or errors.
Other Options:
A . Ensuring that training is reproducible is important for model development, but not necessarily for production readiness. Reproducibility helps the team to track and compare different experiments, but it does not guarantee that the model will perform well in production.
B . Ensuring that all hyperparameters are tuned is also important for model development, but not sufficient for production readiness. Hyperparameter tuning helps the team to find the optimal configuration for the model, but it does not account for the dynamic and changing nature of the production environment.
D . Ensuring that feature expectations are captured in the schema is a part of testing features and data, which the team has already done. The schema defines the expected format, type, and range of the features, and helps the team to validate and preprocess the data.

Question: 14

You recently designed and built a custom neural network that uses critical dependencies specific to your organization's framework. You need to train the model using a managed training service on

Google Cloud. However, the ML framework and related dependencies are not supported by AI Platform Training. Also, both your model and your data are too large to fit in memory on a single machine. Your ML framework of choice uses the scheduler, workers, and servers distribution structure. What should you do?

A. Use a built-in model available on AI Platform Training
B. Build your custom container to run jobs on AI Platform Training
C. Build your custom containers to run distributed training jobs on AI Platform Training
D. Reconfigure your code to a ML framework with dependencies that are supported by AI Platform Training

Answer: C

Explanation:
AI Platform Training is a service that allows you to run your machine learning training jobs on Google Cloud using various features, model architectures, and hyperparameters. You can use AI Platform Training to scale up your training jobs, leverage distributed training, and access specialized hardware such as GPUs and TPUs1. AI Platform Training supports several pre-built containers that provide different ML frameworks and dependencies, such as TensorFlow, PyTorch, scikit-learn, and XGBoost2. However, if the ML framework and related dependencies that you need are not supported by the pre-built containers, you can build your own custom containers and use them to run your training jobs on AI Platform Training3.
Custom containers are Docker images that you create to run your training application. By using custom containers, you can specify and pre-install all the dependencies needed for your application, and have full control over the code, serving, and deployment of your model4. Custom containers also enable you to run distributed training jobs on AI Platform Training, which can help you train large-scale and complex models faster and more efficiently5. Distributed training is a technique that splits the training data and computation across multiple machines, and coordinates them to update the model parameters. AI Platform Training supports two types of distributed training: parameter server and collective all-reduce. The parameter server architecture consists of a set of workers that perform the computation, and a set of servers that store and update the model parameters. The collective all-reduce architecture consists of a set of workers that perform the computation and synchronize the model parameters among themselves. Both architectures also have a scheduler that coordinates the workers and servers.
For the use case of training a custom neural network that uses critical dependencies specific to your organization's framework, the best option is to build your custom containers to run distributed training jobs on AI Platform Training. This option allows you to use the ML framework and dependencies of your choice, and train your model on multiple machines without having to manage the infrastructure. Since your ML framework of choice uses the scheduler, workers, and servers distribution structure, you can use the parameter server architecture to run your distributed training job on AI Platform Training. You can specify the number and type of machines, the custom container image, and the training application arguments when you submit your training job. Therefore, building your custom containers to run distributed training jobs on AI Platform Training is the best option for this use case.
Reference:
AI Platform Training documentation
Pre-built containers for training
Custom containers for training

Custom containers overview | Vertex AI | Google Cloud
Distributed training overview
[Types of distributed training]
[Distributed training architectures]
[Using custom containers for training with the parameter server architecture]

Question: 15

You are an ML engineer in the contact center of a large enterprise. You need to build a sentiment analysis tool that predicts customer sentiment from recorded phone conversations. You need to identify the best approach to building a model while ensuring that the gender, age, and cultural differences of the customers who called the contact center do not impact any stage of the model development pipeline and results. What should you do?

A. Extract sentiment directly from the voice recordings
B. Convert the speech to text and build a model based on the words
C. Convert the speech to text and extract sentiments based on the sentences
D. Convert the speech to text and extract sentiment using syntactical analysis

Answer: C

Explanation:
Sentiment analysis is the process of identifying and extracting the emotions, opinions, and attitudes expressed in a text or speech. Sentiment analysis can help businesses understand their customers' feedback, satisfaction, and preferences. There are different approaches to building a sentiment analysis tool, depending on the input data and the output format. Some of the common approaches are:
Extracting sentiment directly from the voice recordings: This approach involves using acoustic features, such as pitch, intensity, and prosody, to infer the sentiment of the speaker. This approach can capture the nuances and subtleties of the vocal expression, but it also requires a large and diverse dataset of labeled voice recordings, which may not be easily available or accessible. Moreover, this approach may not account for the semantic and contextual information of the speech, which can also affect the sentiment.
Converting the speech to text and building a model based on the words: This approach involves using automatic speech recognition (ASR) to transcribe the voice recordings into text, and then using lexical features, such as word frequency, polarity, and valence, to infer the sentiment of the text. This approach can leverage the existing text-based sentiment analysis models and tools, but it also introduces some challenges, such as the accuracy and reliability of the ASR system, the ambiguity and variability of the natural language, and the loss of the acoustic information of the speech.
Converting the speech to text and extracting sentiments based on the sentences: This approach involves using ASR to transcribe the voice recordings into text, and then using syntactic and semantic features, such as sentence structure, word order, and meaning, to infer the sentiment of the text. This approach can capture the higher-level and complex aspects of the natural language, such as negation, sarcasm, and irony, which can affect the sentiment. However, this approach also requires more sophisticated and advanced natural language processing techniques, such as parsing, dependency analysis, and semantic role labeling, which may not be readily available or easy to implement.
Converting the speech to text and extracting sentiment using syntactical analysis: This approach

involves using ASR to transcribe the voice recordings into text, and then using syntactical analysis, such as part-of-speech tagging, phrase chunking, and constituency parsing, to infer the sentiment of the text. This approach can identify the grammatical and structural elements of the natural language, such as nouns, verbs, adjectives, and clauses, which can indicate the sentiment. However, this approach may not account for the pragmatic and contextual information of the speech, such as the speaker's intention, tone, and situation, which can also influence the sentiment.
For the use case of building a sentiment analysis tool that predicts customer sentiment from recorded phone conversations, the best approach is to convert the speech to text and extract sentiments based on the sentences. This approach can balance the trade-offs between the accuracy, complexity, and feasibility of the sentiment analysis tool, while ensuring that the gender, age, and cultural differences of the customers who called the contact center do not impact any stage of the model development pipeline and results. This approach can also handle different types and levels of sentiment, such as polarity (positive, negative, or neutral), intensity (strong or weak), and emotion (anger, joy, sadness, etc.). Therefore, converting the speech to text and extracting sentiments based on the sentences is the best approach for this use case.

Question: 16

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:
AI Platform Notebooks is a service that provides managed Jupyter notebooks for data science and machine learning. You can use AI Platform Notebooks to create, run, and share your code and analysis in a collaborative and interactive environment1. BigQuery is a service that allows you to analyze large-scale and complex data using SQL queries. You can use BigQuery to stream, store, and query your data in a fast and cost-effective way2. Pandas is a popular Python library that provides data structures and tools for data analysis and manipulation. You can use pandas to create, manipulate, and visualize dataframes, which are tabular data structures with rows and columns3.
AI Platform Notebooks provides a cell magic, %%bigquery, that allows you to run SQL queries on BigQuery data and ingest the results as a pandas dataframe. A cell magic is a special command that applies to the whole cell in a Jupyter notebook. The %%bigquery cell magic can take various arguments, such as the name of the destination dataframe, the name of the destination table in

BigQuery, the project ID, and the query parameters4. By using the %%bigquery cell magic, you can query the data in BigQuery with minimal code and manipulate the results with pandas in AI Platform Notebooks. This is the most convenient and efficient way to achieve your goal.
The other options are not as good as option A, because they involve more steps, more code, and more manual effort. Option B requires you to export your table as a CSV file from BigQuery to Google Drive, and then use the Google Drive API to ingest the file into your notebook instance. This option is cumbersome and time-consuming, as it involves moving the data across different services and formats. Option C requires you to download your table from BigQuery as a local CSV file, and then upload it to your AI Platform notebook instance. This option is also inefficient and impractical, as it involves downloading and uploading large files, which can take a long time and consume a lot of bandwidth. Option D requires you to use a bash cell in your AI Platform notebook to export the table as a CSV file to Cloud Storage, and then copy the data into the notebook. This option is also complex and unnecessary, as it involves using different commands and tools to move the data around.
Therefore, option A is the best option for this use case.
Reference:
AI Platform Notebooks documentation
BigQuery documentation
pandas documentation
Using Jupyter magics to query BigQuery data

Question: 17

You have trained a model on a dataset that required computationally expensive preprocessing operations. You need to execute the same preprocessing at prediction time. You deployed the model on AI Platform for high-throughput online prediction. Which architecture should you use?

A. • Validate the accuracy of the model that you trained on preprocessed data
- Create a new model that uses the raw data and is available in real time
- Deploy the new model onto AI Platform for online prediction

B. • Send incoming prediction requests to a Pub/Sub topic
- Transform the incoming data using a Dataflow job
- Submit a prediction request to AI Platform using the transformed data
- Write the predictions to an outbound Pub/Sub queue

C. • Stream incoming prediction request data into Cloud Spanner
- Create a view to abstract your preprocessing logic.
- Query the view every second for new records
- Submit a prediction request to AI Platform using the transformed data
- Write the predictions to an outbound Pub/Sub queue.

D. • Send incoming prediction requests to a Pub/Sub topic
- Set up a Cloud Function that is triggered when messages are published to the Pub/Sub topic.
- Implement your preprocessing logic in the Cloud Function
- Submit a prediction request to AI Platform using the transformed data
- Write the predictions to an outbound Pub/Sub queue

Answer: D

Explanation:
Option A is incorrect because creating a new model that uses the raw data and is available in real time would require retraining the model and deploying it again, which is not efficient or scalable.
Option B is incorrect because using a Dataflow job to transform the incoming data would introduce unnecessary latency and complexity for online prediction, which requires fast and simple processing.
Option C is incorrect because using Cloud Spanner to stream and query the incoming data would incur high costs and overhead for online prediction, which does not need a relational database.
Option D is correct because using a Cloud Function to preprocess the data and submit a prediction request to Al Platform is a simple and scalable solution for online prediction, which leverages the serverless and event-driven features of Cloud Functions.

Question: 18

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:
When building a model to predict daily temperatures, it is important to split the training and test data based on time rather than a random split. This is because temperature data is likely to have temporal dependencies and patterns, such as seasonality, trends, and cycles. If the data is split randomly, there is a risk of data leakage, which occurs when information from the future is used to train or validate the model. Data leakage can lead to overfitting and unrealistic performance estimates, as the model may learn from data that it should not have access to. By splitting the data based on time, such as using the most recent data as the test set and the older data as the training set, the model can be evaluated on how well it can forecast future temperatures based on past data, which is the realistic scenario in production. Therefore, splitting the data based on time rather than a random split is the best way to make the production model more accurate.

Question: 19

You have a demand forecasting pipeline in production that uses Dataflow to preprocess raw data prior to model training and prediction. During preprocessing, you employ Z-score normalization on data stored in BigQuery and write it back to BigQuery. New training data is added every week. You

want to make the process more efficient by minimizing computation time and manual intervention. What should you do?

A. Normalize the data using Google Kubernetes Engine
B. Translate the normalization algorithm into SQL for use with BigQuery
C. Use the normalizer_fn argument in TensorFlow's Feature Column API
D. Normalize the data with Apache Spark using the Dataproc connector for BigQuery

Answer: B

Explanation:
Z-score normalization is a technique that transforms the values of a numeric variable into standardized units, such that the mean is zero and the standard deviation is one. Z-score normalization can help to compare variables with different scales and ranges, and to reduce the effect of outliers and skewness. The formula for z-score normalization is:
z (x - mu) / sigma
where x is the original value, mu is the mean of the variable, and sigma is the standard deviation of the variable.
Dataflow is a service that allows you to create and run data processing pipelines on Google Cloud. You can use Dataflow to preprocess raw data prior to model training and prediction, such as applying z-score normalization on data stored in BigQuery. However, using Dataflow for this task may not be the most efficient option, as it involves reading and writing data from and to BigQuery, which can be time-consuming and costly. Moreover, using Dataflow requires manual intervention to update the pipeline whenever new training data is added.
A more efficient way to perform z-score normalization on data stored in BigQuery is to translate the normalization algorithm into SQL and use it with BigQuery. BigQuery is a service that allows you to analyze large-scale and complex data using SQL queries. You can use BigQuery to perform z-score normalization on your data using SQL functions such as AVG(), STDDEV_POP(), and OVER(). For example, the following SQL query can normalize the values of a column called temperature in a table called weather:
SELECT (temperature - AVG(temperature) OVER ()) / STDDEV_POP(temperature) OVER () AS normalized_temperature FROM weather;
By using SQL to perform z-score normalization on BigQuery, you can make the process more efficient by minimizing computation time and manual intervention. You can also leverage the scalability and performance of BigQuery to handle large and complex datasets. Therefore, translating the normalization algorithm into SQL for use with BigQuery is the best option for this use case.

Question: 20

You were asked to investigate failures of a production line component based on sensor readings. After receiving the dataset, you discover that less than 1% of the readings are positive examples representing failure incidents. You have tried to train several classification models, but none of them converge. How should you resolve the class imbalance problem?

A. Use the class distribution to generate 10% positive examples
B. Use a convolutional neural network with max pooling and softmax activation
C. Downsample the data with upweighting to create a sample with 10% positive examples

D. Remove negative examples until the numbers of positive and negative examples are equal

Answer: C

Explanation:
The class imbalance problem is a common challenge in machine learning, especially in classification tasks. It occurs when the distribution of the target classes is highly skewed, such that one class (the majority class) has much more examples than the other class (the minority class). The minority class is often the more interesting or important class, such as failure incidents, fraud cases, or rare diseases. However, most machine learning algorithms are designed to optimize the overall accuracy, which can be biased towards the majority class and ignore the minority class. This can result in poor predictive performance, especially for the minority class.
There are different techniques to deal with the class imbalance problem, such as data-level methods, algorithm-level methods, and evaluation-level methods1. Data-level methods involve resampling the original dataset to create a more balanced class distribution. There are two main types of data-level methods: oversampling and undersampling. Oversampling methods increase the number of examples in the minority class, either by duplicating existing examples or by generating synthetic examples. Undersampling methods reduce the number of examples in the majority class, either by randomly removing examples or by using clustering or other criteria to select representative examples. Both oversampling and undersampling methods can be combined with upweighting or downweighting, which assign different weights to the examples according to their class frequency, to further balance the dataset.
For the use case of investigating failures of a production line component based on sensor readings, the best option is to downsample the data with upweighting to create a sample with 10% positive examples. This option involves randomly removing some of the negative examples (the majority class) until the ratio of positive to negative examples is 1:9, and then assigning higher weights to the positive examples to compensate for their low frequency. This option can create a more balanced dataset that can improve the performance of the classification models, while preserving the diversity and representativeness of the original data. This option can also reduce the computation time and memory usage, as the size of the dataset is reduced. Therefore, downsampling the data with upweighting to create a sample with 10% positive examples is the best option for this use case.
Reference:
A Systematic Study of the Class Imbalance Problem in Convolutional Neural Networks

Question: 21

You need to design a customized deep neural network in Keras that will predict customer purchases based on their purchase history. You want to explore model performance using multiple model architectures, store training data, and be able to compare the evaluation metrics in the same dashboard. What should you do?

A. Create multiple models using AutoML Tables
B. Automate multiple training runs using Cloud Composer
C. Run multiple training jobs on AI Platform with similar job names
D. Create an experiment in Kubeflow Pipelines to organize multiple runs

Answer: D

Explanation:
Kubeflow Pipelines is a service that allows you to create and run machine learning workflows on Google Cloud using various features, model architectures, and hyperparameters. You can use Kubeflow Pipelines to scale up your workflows, leverage distributed training, and access specialized hardware such as GPUs and TPUs1. An experiment in Kubeflow Pipelines is a workspace where you can try different configurations of your pipelines and organize your runs into logical groups. You can use experiments to compare the performance of different models and track the evaluation metrics in the same dashboard2.
For the use case of designing a customized deep neural network in Keras that will predict customer purchases based on their purchase history, the best option is to create an experiment in Kubeflow Pipelines to organize multiple runs. This option allows you to explore model performance using multiple model architectures, store training data, and compare the evaluation metrics in the same dashboard. You can use Keras to build and train your deep neural network models, and then package them as pipeline components that can be reused and combined with other components. You can also use Kubeflow Pipelines SDK to define and submit your pipelines programmatically, and use Kubeflow Pipelines UI to monitor and manage your experiments. Therefore, creating an experiment in Kubeflow Pipelines to organize multiple runs is the best option for this use case.
Reference:
Kubeflow Pipelines documentation
Experiment | Kubeflow

Question: 22

Your team needs to build a model that predicts whether images contain a driver's license, passport, or credit card. The data engineering team already built the pipeline and generated a dataset composed of 10,000 images with driver's licenses, 1,000 images with passports, and 1,000 images with credit cards. You now have to train a model with the following label map: ['driversjicense', 'passport', 'credit_card']. Which loss function should you use?

A. Categorical hinge
B. Binary cross-entropy
C. Categorical cross-entropy
D. Sparse categorical cross-entropy

Answer: C

Explanation:
Categorical cross-entropy is a loss function that is suitable for multi-class classification problems, where the target variable has more than two possible values. Categorical cross-entropy measures the difference between the true probability distribution of the target classes and the predicted probability distribution of the model. It is defined as:
L - sum(y_i * log(p_i))
where y_i is the true probability of class i, and p_i is the predicted probability of class i. Categorical cross-entropy penalizes the model for making incorrect predictions, and encourages the model to