## Data Engineer Learning Path

* A Data Engineer designs and builds systems that collect and transform the data used to inform business decisions. Once you complete the path, check out the Google Cloud Data Engineer certification to take the next steps in your professional journey.
  + 1. A Tour of Google Cloud Hands-on Labs
  + 2. Google Cloud Big Data and Machine Learning Fundamentals
  + 3. Modernizing Data Lakes and Data Warehouses with Google Cloud
  + 4. Building Batch Data Pipelines on Google Cloud
  + 5. Building Resilient Streaming Analytics Systems on Google Cloud
  + 6. Smart Analytics, Machine Learning, and AI on Google Cloud
  + 7. Serverless Data Processing with Dataflow: Foundations
  + 8. Serverless Data Processing with Dataflow: Develop Pipelines
  + 9. Serverless Data Processing with Dataflow: Operations
  + 10. Create and Manage Cloud Resources
  + 11. Perform Foundational Data, ML, and AI Tasks in Google Cloud
  + 12. Engineer Data in Google Cloud
  + 13. Preparing for the Google Cloud Professional Data Engineer Exam

## 2. Google Cloud Big Data and Machine Learning Fundamentals

* This course introduces the Google Cloud big data and machine learning products and services that support the data-to-AI lifecycle. It explores the processes, challenges, and benefits of building a big data pipeline and machine learning models with Vertex AI on Google Cloud.

### Course Introduction

* + Reading list
    - <https://cloud.google.com/training#learning-paths>
    - <https://cloud.google.com/training/data-engineering-and-analytics>
    - https://cloud.google.com/training/machinelearning-ai

### Big Data and Machine Learning on Google Cloud

* + This section explores the key components of GC's infrastructure. It's here that we introduce many of the big data and machine learning products and services that support the data-to AI lifecycle on GC.

### Compute 5 minutes`- <https://youtu.be/eO289Fj8J-s>

* + - Focusing middle layer
      * Graphical user interface, text, application, chat or text message

        Description automatically generated
    - google offers a range of computing services
      * compute engine,
        + IaaS offering
        + Raw compute, storage and network capabilities
        + maximum flexibility
      * google kubernetes engine (gke) runs containerized apps in a cloud environment
      * app engine a fully managed PaaS offering, bind code to libraries, Focused on application logic
      * Cloud Functions which executes code in response to events, FaaS offering
    - Lets look at an example of a technology that requires a lot of compute power
      * google photos offers a feature called **automatic video stabilization** this takes an unstable video and stabilizes it to minimize movement
      * this feature to work as intended you need the proper data this includes
        + the video itself which is really a large collection of individual images along with
        + time series data on the camera's position
        + in orientation from the onboard gyroscope and
        + motion from the camera lens
      * a short video can require over a billion data points to feed the ml model to create a stabilized version
        + A screenshot of a computer

          Description automatically generated with low confidence
      * the google photos team needed to develop, train and serve a high performing machine learning model on millions of videos that's a large training data set
      * google trains production ML models on a vast network of data centers only to then deploy smaller trained versions of the models to smartphone and PC hardware
    - in 2016 google introduced the tensor processing unit (TPU) are google's custom developed application specific integrated circuits used to accelerate machine learning workloads
      * tpus act as domain specific hardware as opposed to general purpose hardware with cpus and gpus
      * with tpus the computing speed increases more than 200 times this means that instead of waiting 26 hours for results with a single state-of-the-art gpu you'll only need to wait 7.9 minutes for a full cloud tpu v2 pod to deliver the same results
  + Storage 4 minutes - https://youtu.be/Aa5PQGupngU
    - google cloud offers fully managed database and storage services, cloud storage, cloud bigtable, cloud sql, cloud spanner and firestore
    - the goal of these products is to reduce the time and effort needed to store data this means creating an elastic storage bucket directly in a web interface or through a command line
    - google cloud offers relational and non-relational databases in worldwide object storage
    - choosing the right option to store and process data often depends on the data type that needs to be stored in the business need
    - let's start with unstructured versus structured data
      * unstructured data is information stored in a non-tabular form such as documents, images and audio files
        + unstructured data is usually best suited to cloud storage
        + cloud storage has four primary storage classes

Graphical user interface

Description automatically generated

nearline storage examples include data backups long tail multimedia content or data archiving

cold line storage this is also a low cost option for storing infrequently accessed data

archive storage is the lowest cost option used ideally for data archiving, online backup and disaster recovery

* + - * structured data which represents information stored in tables, rows and columns
        + transactional workloads from online transactional processing systems which are used when fast data inserts and updates are required to build row-based records

this is usually to maintain a system snapshot they require relatively standardized queries that impact only a few records

* + - * + analytical workloads from online analytical processing systems which are used when entire data sets need to be read they often require complex queries for example aggregations
        + once you've determined if the workloads are transactional or analytical you'll need to identify whether the data will be accessed using sql or not

Diagram

Description automatically generated

* + - * + firestore is a transactional nosql document-oriented database
        + bigquery is a fully managed google data warehouse solution

lets you analyze petabyte-scale data sets

that can be used to analyze data through sql commands

* + - * + cloud bigtable provides a scalable no-sql solution for analytical workloads, it's best for real-time throughput applications that require only millisecond latency
  + The history of big data and ML products 3 minutes - <https://youtu.be/6W5RdufJtto>
  + Big data and ML product categories 2 minutes - <https://youtu.be/m35XpFf9XGc>
    - google offers a range of big data and machine learning products
    - divided into four general categories along the data to ai workflow
      * ingestion and process
      * storage
      * analytics
      * machine learning
    - ingestion process which include products that are used to digest both real-time and batch data the list includes pub sub, data flow, data proc and cloud data fusion
    - data storage - cloud storage, cloud sql, cloud spanner, cloud bigtable and firestore
    - analytics the major analytics tool is bigquery.
      * you can analyze data and visualize results using google data studio and looker
    - machine learning products include both the ml development platform and the ai solutions
      * the primary product of the ml development platform is vertex ai which includes automl, vertex ai workbench and tensorflow
      * ai solutions are built on the ml development platform and includes state-of-the-art products to meet both horizontal and vertical market needs these include document ai, contact center ai, retail product discovery and healthcare data engine
  + Customer example: Gojek 3 minutes - <https://youtu.be/-who6RBUXSg>
    - Gojek company - how they were able to find success through google cloud's data engineering and machine learning offerings
      * in jakarta traffic congestion is a fact of life for most residents, to minimize delays many rely heavily on motorcycles including motorcycle taxis known as ojx, to travel to and from work or personal engagements
      * gojek has collected data to understand customer behavior and in 2015 launched a mobile application that bundled ride hailing, food delivery and grocery shopping
      * q2 2021 gojek fact sheet the gojek app has been downloaded over 190 million times and they have 2 million driver partners and about 900 000 merchant partners
    - the business has relied heavily on the skills and expertise of the technology team
      * gojek chose to run its applications and data in google cloud
      * gojek's goal is to match the right driver with the right request as quickly as possible
      * in the early days of the app a driver would be pinged every 10 seconds, which meant 6 million pings per minute which turned out to be 8 billion pings per day across their driver partners
      * they generated around 5 terabytes of data each day, leveraging information from this data was vital to meeting their company goals,
    - but gojek faced challenges along the way, let's explore 2 of them to see how google cloud was able to solve them
      * the first challenge was data latency, when they wanted to scale their big data platform they found that most reports were produced one day later , so they couldn't identify problems immediately,
        + to help solve this, go-jek migrated their data pipelines to google cloud, the team started using dataflow for streaming data processing and bigquery for real-time business insights
      * another challenge was quickly determining which location had too many or too few drivers to meet demand.
        + gojek was able to use dataflow to build a streaming event data pipeline.
        + this let driver locations ping pub sub every 30 seconds and dataflow would process the data.
        + the pipeline would aggregate the supply pings from the drivers against the booking requests.
        + this would connect to gojek's notification system to alert drivers where they should go.
        + this process required a system that was able to scale up to handle times of high throughput and then back down again.
        + dataflow was able to automatically manage the number of workers processing the pipeline to meet demand.
        + the gojek team was able to visualize and identify supply and demand issues.
        + they discovered that the areas with the highest discrepancy between supply and demand came from train stations often there were far more booking requests than there were available drivers since
    - using google cloud's big data and machine learning products the gojek team has been able to actively monitor requests to ensure the drivers are in the areas with the highest demand this brings faster bookings for riders and more work for the drivers
      * Graphical user interface, application

        Description automatically generated

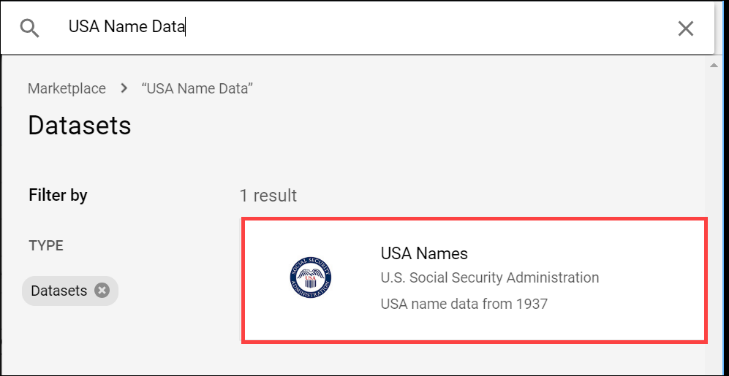
### Exploring a BigQuery Public Dataset 1 hour - https://www.cloudskillsboost.google/course\_sessions/926167/labs/200007

* + - Overview
      * Storing and querying massive datasets can be time consuming and expensive without the right hardware and infrastructure. BigQuery is an enterprise data warehouse that solves this problem by enabling super-fast SQL queries using the processing power of Google's infrastructure. Simply move your data into BigQuery and let us handle the hard work. You can control access to both the project and your data based on your business needs, such as giving others the ability to view or query your data.
      * You access BigQuery through the Cloud Console, the command-line tool, or by making calls to the BigQuery REST API using a variety of client libraries such as Java, .NET, or Python. There are also a variety of third-party tools that you can use to interact with BigQuery, such as visualizing the data or loading the data. In this lab, you access BigQuery using the web UI.
      * You can use the BigQuery web UI in the Cloud Console as a visual interface to complete tasks like running queries, loading data, and exporting data. This hands-on lab shows you how to query tables in a public dataset and how to load sample data into BigQuery through the Cloud Console.
    - Open BigQuery Console
      * Navigation > BigQuery > Click Done.
    - Task 1. Query a public dataset
      * In this task, you load a public dataset, USA Names, into BigQuery, then query the dataset to determine the most common names in the US between 1910 and 2013.
      * Load the USA Names dataset
        + In the left pane, click ADD DATA > Explore public datasets.

Graphical user interface, application

Description automatically generated

* + - * + The Datasets > search USA Names > Click > View dataser



* + - * BigQuery opens in a new browser tab. The project bigquery-public-data is added to your resources and you see the dataset usa\_names listed in the left pane in Resources tree.
        + Graphical user interface, application, table

          Description automatically generated
      * Query the USA Names dataset
        + Query bigquery-public-data.usa\_names.usa\_1910\_2013 for the name and gender of the babies in this dataset, and then list the top 10 names in descending order.

SELECT name, gender, SUM(number) AS total

FROM `bigquery-public-data.usa\_names.usa\_1910\_2013`

GROUP BY name, gender ORDER BY total DESC LIMIT 10

* + - * When the query is valid, the validator also shows the amount of data the query processes when you run it. This helps to determine the cost of running the query.
      * The Query results displays the time elapsed and the data processed by the query.
        + Table

          Description automatically generated
    - Task 2. Create a custom table
      * you create a custom table, load data into it, and then run a query against the table.
      * Download the data to your local computer
        + The file contains data about popular baby names, & it is provided by the US SSA.
        + Download the file <https://www.ssa.gov/OACT/babynames/names.zip>
        + The NationalReadMe.pdf file that describes the dataset. Learn more about the dataset. <https://www.ssa.gov/OACT/babynames/background.html>
        + Open the CSV file named yob2014.txt to see the following 3 columns: name, sex (M or F), and number of children with that name. The file has no header row.
    - Task 3. Create a dataset
      * create a dataset to hold your table, add data to your project, then make the data table you'll query against.
      * Datasets help you control access to tables and views in a project. This lab uses only one table, but you still need a dataset to hold the table.
      * Cloud Console > Explorer section > click your Project ID (it will start with qwiklabs).
        + Graphical user interface, application

          Description automatically generated
      * Click on the three dots next > Create dataset > Enter > Create dataset at the bottom:
        + Graphical user interface, application

          Description automatically generated
        + For Dataset ID, enter babynames.
        + For Data location, choose us (multiple regions in United States).
        + For Default table expiration, leave the default value.
        + For Encryption, leave the default value.
    - Task 4. Load the data into a new table
      * Click babynames in the Explorer section > click Create table.
      * Enter the values > Click Create table at the bottom
      * On the Create table page:
        + For Source, choose Upload from the Create table from: dropdown menu.
        + For Select file, click Browse, navigate to the yob2014.txt file and click Open.
        + For File format, choose CSV from the dropdown menu.
        + For Table name, enter names\_2014.
        + In the Schema section, click the Edit as text toggle and paste the following schema definition in the text box.

name:string,gender:string,count:integer

* + - * Click Create table (at the bottom of the window).
      * Preview the table > select babynames > names\_2014 in the navigation pane.
      * Quick quiz. You need a table to hold the dataset. False
    - Task 5. Query the table
      * This query retrieves the top 5 baby names for US males in 2014.
      * Note: Inside '' it does distinguish upper vs. lower case, therefore make sure to align exactly the names of the dataset and the table you created.
        + SELECT name, count FROM `babynames.names\_2014`
        + WHERE gender = 'M' ORDER BY count DESC LIMIT 5
      * Quick quiz. Check which you can use to access BigQuery.
        + Web UI
        + Command-line tool
        + Third-party tools
        + Make calls to BigQuery REST API

### Quiz

* + - Cloud Storage, Cloud Bigtable, Cloud SQL, Cloud Spanner, and Firestore represent which type of services?
      * Database and storage
    - AutoML, Vertex AI Workbench, and TensorFlow align to which stage of the data-to-AI workflow?
      * Machine learning
    - Which Google hardware innovation tailors architecture to meet the computation needs on a domain, such as the matrix multiplication in machine learning?
      * TPUs (Tensor Processing Units)
    - Which data storage class is best for storing data that needs to be accessed less than once a year, such as online backups and disaster recovery?
      * Archive storage
    - Pub/Sub, Dataflow, Dataproc, and Cloud Data Fusion align to which stage of the data-to-AI workflow?
      * Ingestion and process
    - Compute Engine, Google Kubernetes Engine, App Engine, and Cloud Functions represent which type of services?
      * Compute

### Data Engineering for Streaming Data

* + This section introduces GC's solution to managing streaming data. It examines an end-to-end pipeline, including data ingestion with Pub/Sub, data processing with Dataflow, and data visualization with Looker and Data Studio.

### Big data challenges 2 minutes - <https://youtu.be/yPy7Yb0Jdjc>

* + - building scalable and reliable pipelines is a core responsibility of data engineers
    - in modern organizations data engineers and data scientists are facing four major challenges
      * variety, volume, velocity and veracity
      * 1st data could come in from a variety of different sources and in various formats
        + imagine hundreds of thousands of sensors for self-driving cars on roads around the world the data is returned in various formats such as number image or even audio
      * volume of data that can vary from gigabytes to petabytes
        + whether your pipeline code and infrastructure can scale with those changes or whether it will grind to a halt or even crash
      * velocity data often needs to be processed in near real time as soon as it reaches the system
        + need a way to handle data that - arrives late, has bad data in the message or needs to be transformed
      * veracity which refers to the data quality
  + Message-oriented architecture 4 minutes - <https://youtu.be/76szWPgPf84>
    - data ingestion which is where large amounts of streaming data are received (a single structured db or data stream from a 1000/million different events that are all happening asynchronously
      * For example, data from iot applications, send out location data every 30 seconds or temperature sensors around a data center to help optimize heating and cooling.
      * These iot devices present new challenges to data ingestion which can be summarized in 4 pts
        + 1st is that data can be streamed from many different methods and devices

many of which might not talk to each other and might be sending bad or delayed data

* + - * + 2nd is that it can be hard to distribute event messages to the right subscribers

event messages are notifications a method is needed to collect the streaming messages that come from iot sensors and broadcast them to the subscribers as needed

* + - * + 3rd is that data can arrive quickly and at high volumes

services must be able to support this

* + - * + 4th is ensuring services are reliable secure and perform as expected
    - google cloud has a tool to handle distributed message-oriented architectures at scale and that's pub sub - short for publisher subscriber or published messages to subscribers
      * pubsub is a distributed messaging service that can receive messages from a variety of device streams such as gaming events, iot devices and application streams
      * let's explore the end-to-end pub sub architecture,
        + upstream source data comes in from devices all over the globe and is ingested into pub sub which is the first point of contact within the system.
        + pub sub reads, stores and broadcasts to any subscribers of this data topic that new messages are available.
        + as a subscriber of pub sub dataflow can ingest and transform those messages in an elastic streaming pipeline and output the results into an analytics data warehouse like bigquery.
        + finally you can connect a data visualization tool like looker or data studio to visualize and monitor the results of a pipeline or an ai or ml tool such as vertex ai to explore the data to uncover business insights or help with predictions
      * A picture containing diagram

        Description automatically generated
  + Designing streaming pipelines with Apache Beam 2 minutes - https://youtu.be/AiqeMxpq1Ik
    - after messages have been captured from the streaming input sources you need a way to pipe that data into a data warehouse for analysis this is where dataflow comes in
    - dataflow creates a pipeline to process both streaming data and batch data
      * process in this case refers to the steps to extract, transform and load (etl) data
      * when building a data pipeline data engineers often encounter challenges related to coding the pipeline design and implementing and serving the pipeline at scale
      * during the pipeline design phase there are a few questions to consider
        + will the pipeline code be compatible with both batch and streaming data or will it need to be refactored
        + will the pipeline code SDK being used have all the transformations mid-flight aggregations and windowing and
        + be able to handle late data
        + are there existing templates or solutions that should be referenced
      * a popular solution for pipeline design is apache beam.
        + it's an open source unified programming model to define and execute data processing pipelines including etl , batch and stream processing
        + apache beam is unified which means it uses a single programming model for both batch and streaming data
        + it's portable which means it can work on multiple execution environments like dataflow and apache spark , among others
        + it's extensible which means it allows you to write and share your own connectors and transformation libraries
        + apache beam provides pipeline templates, so you don't need to build a pipeline from nothing and
        + it can write pipelines in java, python or go
        + the apache beam sdk is a collection of software development tools in one installable package it provides a variety of libraries for transformations and data connectors to sources and sinks
        + apache beam creates a model representation from your code that's portable across many runners. runners pass off your model for execution on a variety of different possible engines with dataflow being a popular choice
  + Implementing streaming pipelines on Cloud Dataflow 3 minutes - https://youtu.be/fzP6mUK\_Tfw
    - the next step is to identify an execution engine to implement those pipelines
      * Graphical user interface, text, application, chat or text message

        Description automatically generated
    - when choosing an execution engine for your pipeline code it may be helpful to consider the following questions
      * how much maintenance overhead is involved
      * is the infrastructure reliable
      * how is the pipeline scaling handled
      * how can the pipeline be monitored
      * is the pipeline locked into a specific service provider
    - this brings us to dataflow
      * dataflow is a fully managed service for executing apache beam pipelines within the google cloud ecosystem
      * dataflow handles much of the complexity relating to infrastructure setup and maintenance and is built on google's infrastructure
      * this allows for reliable auto scaling to meet data pipeline demands
      * Graphical user interface, application

        Description automatically generated
      * dataflow is serverless and no ops which means no operations but
        + what does that mean exactly a no-ops environment is one that doesn't require management from an operations team because m,m,s are automated
        + serverless computing is a cloud computing execution model, this is when google cloud for example manages infrastructure tasks on behalf of the users this includes tasks like rp,pt,pr
      * using a serverless and nopps solution like dataflow means that you can spend more time analyzing the insights from your data sets and less time provisioning resources to ensure that your pipeline will successfully complete its next cycles
      * it's designed to be low maintenance
    - let's explore the tasks dataflow performs
      * + Graphical user interface, application

          Description automatically generated
      * when a job is received it starts by optimizing a pipeline model's execution graph to remove any inefficiencies,
      * next it schedules out distributed work to new workers and scales as needed ,
      * after that it auto heals any worker faults,
      * from there it automatically rebalances efforts to most efficiently use its workers and
      * finally it outputs data to produce a result, bigquery is one of many options that data can be outputted
    - dataflow templates can be broken down into three categories
      * streaming templates are for processing continuous or real-time data
        + for example pub sub to bigquery, pubs up to cloud storage, data stream to bigquery and pubsub to mongodb
      * batch templates are for processing bulk data or batch load data
        + for example bigquery to cloud storage, bigtable to cloud storage, cloud storage to bigquery and cloud spanner to cloud storage
      * finally utility templates address activities related to bulk compression, deletion and conversion
  + Visualization with Looker 3 minutes - https://youtu.be/duQzCa51AlY
    - to help create an environment where stakeholders can easily interact with and visualize data google cloud offers two solutions looker and google data studio
    - looker supports bigquery as well as more than 60 different types of sql database products commonly referred to as dialects
      * + A picture containing graphical user interface

          Description automatically generated
      * it allows developers to define a semantic modeling layer on top of databases using looker modeling language or lookml
      * lookml
        + defines logic and permissions independent from a specific database or a sql language
        + which frees a data engineer from interacting with individual databases to focus more on business logic across an organization
      * the looker platform is 100 web-based which makes it easy to integrate into existing workflows and share with multiple teams at an organization
      * there's also a looker api which can be used to embed looker reports in other applications
  + Visualization with Data Studio 1 minute - https://youtu.be/8BJy0IttYSo
    - data studio is integrated into bigquery which makes data visualization possible with just a few clicks
    - data studio integration is the google cloud billing dashboard
    - to create a data studio dashboard
      * first choose a template you can start with either a pre-built template or a blank report
      * second link the dashboard to a data source this might come from bigquery, a local file or a google application like google sheets or google analytics or a combination of any of these sources and
      * third explore your dashboard

### Creating a Streaming Data Pipeline for a Real-Time Dashboard with Dataflow 1 hour

* + - Overview
      * In this lab, you own a fleet of New York City taxi cabs and are looking to monitor how well your business is doing in real-time. You will build a streaming data pipeline to capture taxi revenue, passenger count, ride status, and much more and visualize the results in a management dashboard.
    - Task 1. Source a public Pub/Sub topic and create a BigQuery dataset
      * Pub/Sub is an asynchronous global messaging service. By decoupling senders and receivers, it allows for secure and highly available communication between independently written applications. Pub/Sub delivers low-latency, durable messaging.
      * In Pub/Sub, publisher applications and subscriber applications connect with one another through the use of a shared string called a topic. A publisher application creates and sends messages to a topic. Subscriber applications create a subscription to a topic to receive messages from it.
      * Google maintains a few public Pub/Sub streaming data topics for labs like this one. We'll be using the NYC Taxi & Limousine Commission’s open dataset. https://data.cityofnewyork.us/
      * BigQuery is a serverless data warehouse. Tables in BigQuery are organized into datasets. In this lab, messages published into Pub/Sub will be aggregated and stored in BigQuery.
      * To create a new BigQuery dataset:
      * Option 1: The command-line tool
        + Open Cloud Shell and run the below command to create the taxirides dataset.

bq mk taxirides

* + - * + Run this command to create the taxirides.realtime table (empty schema that you will stream into later).

bq mk \

--time\_partitioning\_field timestamp \

--schema ride\_id:string,point\_idx:integer,latitude:float,longitude:float,\

timestamp:timestamp,meter\_reading:float,meter\_increment:float,ride\_status:string,\

passenger\_count:integer -t taxirides.realtime

* + - * Option 2: The BigQuery Console UI
        + Navigation > Big Data > BigQuery > Create dataset.
        + Set the Dataset ID as taxirides, leave all the other fields > click CREATE DATASET.
        + Click the taxirides dataset > Click CREATE TABLE. Name the table realtime
        + For the schema, click Edit as text and paste in the below:

ride\_id:string,

point\_idx:integer,

latitude:float,

longitude:float,

timestamp:timestamp,

meter\_reading:float,

meter\_increment:float,

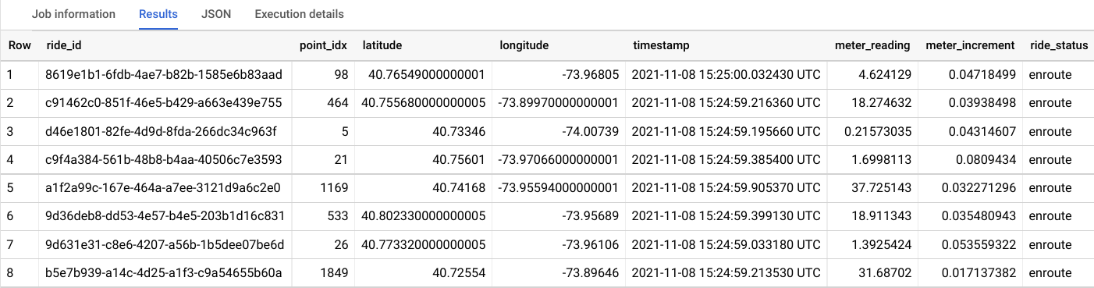
ride\_status:string,

passenger\_count:integer

* + - * Under Partition and cluster settings, select the timestamp option for the Partitioning field. Click the CREATE TABLE button.
    - Task 2. Create a Cloud Storage bucket
      * Cloud Storage allows world-wide storage and retrieval of any amount of data at any time. You can use Cloud Storage for a range of scenarios including serving website content, storing data for archival and disaster recovery, or distributing large data objects to users via direct download. In this lab, you use Cloud Storage to provide working space for your Dataflow pipeline.
      * Navigation > Cloud Storage > CREATE BUCKET > name (GCP Project ID) > Continue.
      * For Location type, click Multi-region if it is not already selected. Click CREATE.
    - Task 3. Set up a Dataflow Pipeline
      * Dataflow is a serverless way to carry out data analysis. In this lab, you set up a streaming data pipeline to read sensor data from Pub/Sub, compute the maximum temperature within a time window, and write this out to BigQuery.
      * Restart the connection to the Dataflow API.
      * Navigation > API > search Dataflow API > Manage > Disable API > Click Enable.
      * To create a new streaming pipeline:
        + Navigation > Dataflow > CREATE JOB FROM TEMPLATE.
        + Enter streaming-taxi-pipeline as the Job name for your Dataflow job.
        + Under Dataflow template, select the Pub/Sub Topic to BigQuery template.
        + Under Input Pub/Sub topic, enter projects/pubsub-public-data/topics/taxirides-realtime
        + Under BigQuery output table, enter <myprojectid>:taxirides.realtime
        + Under Temporary location, enter gs://<mybucket>/tmp/.
        + Click Show Optional Parameters and input the following values as listed below:

Max workers: 2

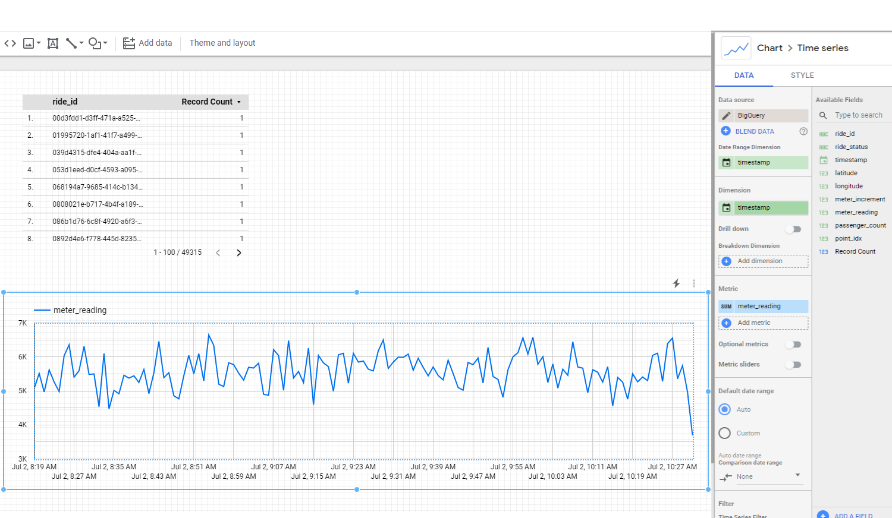
Number of workers: 2

* + - * + Click the RUN JOB button.
      * A new streaming job has started! You can now see a visual representation of the data pipeline.
      * Note: If the dataflow job fails for the first time then re-create a new job template with new job name and run the job.
    - Task 4. Analyze the taxi data using BigQuery
      * To analyze the data as it is streaming:
      * Navigation > BigQuery :
        + SELECT \* FROM taxirides.realtime LIMIT 10
      * If no records are returned, wait another minute and re-run the above query (Dataflow takes 3-5 minutes to setup the stream). You will receive a similar output:
        + 
    - Task 5. Perform aggregations on the stream for reporting
      * Copy and paste the below query and click RUN.
        + WITH streaming\_data AS (
        + SELECT
        + timestamp, TIMESTAMP\_TRUNC(timestamp, HOUR, 'UTC') AS hour,
        + TIMESTAMP\_TRUNC(timestamp, MINUTE, 'UTC') AS minute,
        + TIMESTAMP\_TRUNC(timestamp, SECOND, 'UTC') AS second,
        + ride\_id, latitude, longitude, meter\_reading, ride\_status, passenger\_count
        + FROM taxirides.realtime
        + WHERE ride\_status = 'dropoff' ORDER BY timestamp DESC LIMIT 1000
        + )
        + # calculate aggregations on stream for reporting:
        + SELECT
        + ROW\_NUMBER() OVER() AS dashboard\_sort, minute,
        + COUNT(DISTINCT ride\_id) AS total\_rides,
        + SUM(meter\_reading) AS total\_revenue,
        + SUM(passenger\_count) AS total\_passengers
        + FROM streaming\_data GROUP BY minute, timestamp
      * Note: Ensure dataflow is registering data in BigQuery before proceeding to the next task.
      * The result shows key metrics by the minute for every taxi drop-off.
    - Task 6. Stop the Dataflow job
      * Dataflow > Click the streaming-taxi-pipeline > STOP > Cancel > STOP JOB.
      * This will free up resources for your project.
    - Task 7. Create a real-time dashboard
      * Open this Google Data Studio link https://datastudio.google.com/
      * On the Reports page, in the Start with a Template section, click the [+] Blank Report template.
      * Welcome page, click Get started > Check the checkbox to acknowledge.
      * Select No to all the questions, then click Continue. Switch back to the BigQuery Console.
      * Click EXPLORE DATA > Explore with Data Studio in BigQuery page.
      * Click GET STARTED, then click AUTHORIZE.
      * Specify the below settings:
        + Chart type: Combo chart
        + Date range Dimension: dashboard\_sort
        + Dimension: dashboard\_sort
        + Drill Down: dashboard\_sort (Make sure that Drill down option is turned ON)
        + Metric: SUM() total\_rides, SUM() total\_passengers, SUM() total\_revenue
        + Sort: dashboard\_sort, Ascending (latest rides first)
      * Your chart should look similar to this:
        + 
      * Note: Visualizing data at a minute-level granularity is currently not supported in Data Studio as a timestamp. This is why we created our own dashboard\_sort dimension.
      * When you're happy with your dashboard, click Save to save this data source.
      * Whenever anyone visits your dashboard, it will be up-to-date with the latest transactions. You can try it yourself by clicking on the Refresh button near the Save button.
    - Task 8. Create a time series dashboard
      * Click this Google Data Studio link to open Data Studio <https://datastudio.google.com/>
      * On the Reports page, in the Start with a Template section, click the [+] Blank Report template.
      * A new, empty report opens with Add data to report.
      * From the list of Google Connectors, select the BigQuery tile.
      * Under CUSTOM QUERY, click qwiklabs-gcp-xxxxxxx > Enter Custom Query, add the following query.
        + SELECT \* FROM taxirides.realtime
        + WHERE ride\_status='dropoff'
      * Click Add > ADD TO REPORT.
      * Create a time series chart
        + In the Data panel, scroll down to the bottom right and click ADD A FIELD. Click All Fields on the left corner.
        + Change the field timestamp type to Date & Time > Date Hour Minute (YYYYMMDDhhmm).
        + Click Continue and then click Done. Click Add a chart.
        + Choose Time series chart.
        + Position the chart in the bottom left corner - in the blank space.
        + In the Data panel on the right, change the following:

Dimension: timestamp

Metric: meter\_reading(SUM)

* + - * + Your time series chart should look similar to this:



* + - * + Note: if Dimension is timestamp(Date), then click on calendar icon next to timestamp(Date), and select type to Date & Time > Date Hour Minute.

### Quiz

* + - Select the correct streaming data workflow.
      * Ingest the streaming data, process the data, and visualize the results.
    - Which Google Cloud product is a distributed messaging service that is designed to ingest messages from multiple device streams such as gaming events, IoT devices, and application streams?
      * Pub/Sub
    - Which Google Cloud product acts as an execution engine to process and implement data processing pipelines?
      * Dataflow
    - Due to several data types and sources, big data often has many data dimensions. This can introduce data inconsistencies and uncertainties. Which type of challenge might this present to data engineers?
      * Veracity
    - When you build scalable and reliable pipelines, data often needs to be processed in near-real time, as soon as it reaches the system. Which type of challenge might this present to data engineers?
      * Velocity
  + Reading list

### Big Data with BigQuery

* + This section introduces learners to BigQuery, Google's fully-managed, serverless data warehouse. It also explores BigQuery ML, and the processes and key commands that are used to build custom machine learning models.

### Storage and analytics 3 minutes - https://youtu.be/9puClccSJX4

* + - bigquery provides two services in one it's both
      * a fully managed storage facility to load and store data sets and
      * also a fast sql-based analytical engine
    - the two services are connected by google's high-speed internal network
      * it's the super fast network that allows bigquery to scale both storage and compute independently based on demand
    - let's look at how bigquery manages the storage and metadata for data sets
      * bigquery can ingest data sets from various sources including
        + internal data which is data saved directly in bigquery,
        + external data, multi-cloud data and public data sets
      * after the data is stored in bigquery it's fully managed and is automatically
        + replicated, backed up and set to auto scale
    - three basic patterns to load data into bigquery
      * the first is a batch load where source data is loaded into a bigquery table in a single batch operation
      * the second is streaming where smaller batches of data are streamed continuously so that the data is available for querying in near real time
      * and the third is generated data where sql statements are used to insert rows into an existing table or to write the results of a query to a table
    - the analytics features that are available in bigquery, bigquery supports
      * ad-hoc analysis using standard sql, the bigquery sql dialect
      * geospatial analytics using geography data types and standard sql geography functions
      * building machine learning models using bigquery ml
      * building rich interactive business intelligence dashboards using bigquery bi engine
      * by default bquery runs interactive queries which means that the queries are executed as needed
      * bigquery also offers batch queries where each query is queued on your behalf and the query starts when idle resources are available usually within a few minutes
  + BigQuery demo - San Francisco bike share 11 minutes - https://youtu.be/g4h27DwojIs
  + Introduction to BigQuery ML 4 minutes - <https://youtu.be/nhL_9jRvIFk>
    - if you've worked with ml models before you know that building and training them can be very time intensive
      * you must first export data from your data store into an ide such as jupyter notebook or google collab and
      * then transform the data and perform your feature engineering steps before you can feed it into a training model
      * then finally you need to build the model in tensorflow or similar library and train it locally on a computer or on a virtual machine
      * to improve the model performance
    - now you can create and execute ML models on your structured data sets in bigquery using sql, there are two steps needed to start
      * create a model with a sql statement, here the bike share data set as an example
      * write a sql prediction query and invoke ml.predict
    - bigquery ml was designed to be simple like building a model in two steps that simplicity extends to defining the machine learning hyper parameters
      * which let you tune the model to achieve the best training result
      * hyperparameters are the settings apply to a model before the training starts like a learning rate with
    - bigquery ml you can either manually control the hyper parameters or hand it to bigquery starting with a default hyperparameter setting and then automatic tuning
    - when using a structured data set in bigquery ml you need to choose the appropriate model type choosing which type of ml model depends on your business goal and the data sets
    - bigquery supports supervised and unsupervised models
      * supervised models are task driven and identify a goal
      * alternatively unsupervised models are data driven and identify a pattern
      * within a supervised model if your goal is to classify data like whether an email is spam use logistic regression, if your goal is to predict a number like shoe sales for the next three months use linear regression
      * within an unsupervised model if your goal is to identify patterns or clusters and then determine the best way to group them like grouping random photos of flowers into categories you should use cluster analysis
      * A picture containing Teams

        Description automatically generated
      * A picture containing table

        Description automatically generated
      * Text

        Description automatically generated
  + Using BigQuery ML to predict customer lifetime value 4 minutes - <https://youtu.be/cU9WU_MZMTQ>
    - now that you're familiar with the types of ml models available to choose from high quality data must be used to teach the models what they need to learn
    - the best way to learn the key concepts of machine learning on structured data sets is through an example
    - this scenario we'll predict customer lifetime value with a model
    - lifetime value or ltv is a common metric in marketing used to estimate how much revenue or profit you can expect from a customer given their history and customers with similar patterns
      * Graphical user interface, application

        Description automatically generated
    - we'll use a google analytics ecommerce data set from google's own merchandise store that sells branded items like t-shirts and jackets
    - Table

      Description automatically generated
    - the goal is to identify high-value customers and bring them to our store with special promotions and incentives
    - having explored the available fields you may find some useful in determining whether a customer is high value based on their behavior on our website
    - these fields include customer lifetime page views, total visits, average time spent on the site, total revenue brought in and e-commerce transactions on the site
    - remember that in machine learning you feed in columns of data and let the model figure out the relationship to best predict the label
    - it may turn out that some of the columns weren't useful at all to the model in predicting the outcome
    - we have some data we can prepare to feed it into the model incidentally to keep this example simple we're only using 7 records but we'd need tens of thousands of records to train a model effectively
    - before we feed the data into the model we first need to define our data and columns in the language that data scientists and other ml professionals use
    - using the google merchandise store example a record or row in the data set is called an example an observation or an instance
    - 
    - a label because it comes from historical data this is what you need to train the model on in order to predict future data depending on what you want to predict
      * Table

        Description automatically generated
    - a label can either be a numeric variable which requires a linear regression model or a categorical variable which requires a logistic regression model
      * Table

        Description automatically generated
    - for example if we know that a customer who has made transactions in the past and spends a lot of time on our website often turns out to have high lifetime revenue we could use revenue as the label and predict the same for newer customers with that same spending trajectory
    - this means forecasting a number so we can use a linear regression as a starting point to model
    - labels could also be categorical variables like binary values such as high value customer or not to predict a categorical variable
    - use a logistic regression model knowing what you're trying to predict such as a class or a number will greatly influence the type of model you'll use
    - but all the other data columns in the data table called features or potential features
      * Table

        Description automatically generatedeach column of data is like a cooking ingredient you can use from the kitchen pantry too many ingredients however can ruin a dish
    - the process of sifting through data can be time consuming understanding the quality of the data in each column and working with teams to get the most features or more history is often the hardest part of any ml project
    - you can even combine or transform feature columns in a process called feature engineering if you've ever created calculated fields in sql you've already executed the basics of feature engineering
      * Table

        Description automatically generated
    - also bigquery ml does much of the hard work for you like
      * + automatically one hot encoding categorical values , is a method of converting categorical data to numeric data to prepare it for model training
        + from there bigquery ml automatically splits the data set into training data and evaluation data
    - and finally there is predicting and future data
      * let's say new data comes in that you don't have a label for so you don't know whether it's for a high-value customer you do however have a rich history of labeled examples for you to train a model on so if we train a model on the known historical data and are happy with the performance then we can use it to predict our future data sets
        + Table

          Description automatically generated
        + Table

          Description automatically generated
  + BigQuery ML project phases 2 minutes - <https://youtu.be/7nntdeBrmb0>
    - let's explore the key phases of a machine learning project
      * phase 1 you ETL data into bigquery if it isn't there already
        + if you're already using other google products like youtube for example look out for easy connectors to get the data into bigquery before you build your own pipeline
        + you can enrich your existing data warehouse with other data sources by using sql joins
      * phase 2 you select and pre-process features
        + you can use sql to create the training data set for the model to learn
        + from you'll recall that bigquery ml does some of the pre-processing for you like one-hot encoding of your categorical variables
        + one hot encoding converts your categorical data into numeric data that is required by a training model
      * phase 3 you create the model inside bigquery
        + Text

          Description automatically generated
        + this is done by using the create model command (specify a name, the model type) and pass it in a sql query with your training data set from there you can run the query
        + phase 4 after your model is trained you can execute an ml.evaluate query to evaluate the performance of the trained model on your evaluation data set

Graphical user interface, application

Description automatically generated

it's here that you can analyze loss metrics like a root mean squared error for forecasting models and area under the curve, accuracy, precision and recall for classification models we'll explore these metrics later in the course

* + - * + final phase 5 when you're happy with your model performance you can then use it to make predictions

Graphical user interface, text, application

Description automatically generated

to do so invoke the ml.predict command on your newly trained model to return with predictions and the model's confidence in those predictions

with the results your label field will have predicted added to the field name this is your model's prediction for that label

* + BigQuery ML key commands 3 minutes - <https://youtu.be/TpgtM3egpBk>
    - create a model with just the create model command if you want to overwrite an existing model use the create or replace model command
      * models have options which you can specify, the model type is required
      * Graphical user interface, text, application

        Description automatically generated
    - you can inspect what the model learned with the ml.weights command and filtering on an input column. the output of ml.weights is a numerical value and each feature has a weight from -1 to 1.
      * + Graphical user interface, text, application, email

          Description automatically generated
      * the value indicates how important the feature is for predicting the result or label
      * if the number is closer to zero the feature isn't important for the prediction
      * however if the number is closer to negative one or one then the feature is more important for predicting the result
    - to evaluate the model's performance you can run an ml.evaluate command against a trained model you get different performance metrics depending on the model type you choose
      * + Graphical user interface, text, application, email

          Description automatically generated
    - and if you want to make batch predictions you can use the ml.predict command on a trained model and pass through the data set you want to make the prediction on
      * Graphical user interface, text, application

        Description automatically generated
    - list of bigquery ml commands for supervised models
      * LABELS: first in bigquery ml you need a field in your training data set titled label or you need to specify which field or fields your labels are using as the input label columns in your model options
      * Features: second your model features are the data columns that are part of your select statement after your create model statement after a model is trained you can use the ml.feature info command to get statistics and metrics about the column for additional analysis
      * Model Object: next is the model object itself this is an object created in bigquery that resides in your bigquery data set you train many different models which will all be objects stored under your bigquery data set much like your tables and views model objects can display information for when it was last updated or how many training runs it completed
      * Model Types: creating a new model is as easy as writing create model choosing a type and passing in a training data set again if you're predicting on a numeric field such as next year's sales consider linear regression for forecasting if it's a discrete class like high medium low or spam or not spam consider using logistic regression for classification
      * Training Progress: while the model is running and even after it's complete you can view training progress with ml.training info
      * Training weights: as mentioned earlier you can inspect weights to see what the model learned about the importance of each feature as it relates to the label you're predicting the importance is indicated by the weight of each feature
      * Evaluation: you can see how well the model performed against its evaluation data set by using ml.evaluate and
      * Prediction: lastly getting predictions is as simple as writing ml.predict and referencing your model name and prediction data set

### Predicting Visitor Purchases with a Classification Model with BigQuery ML 1 hour 10 minutes - <https://www.cloudskillsboost.google/course_sessions/926167/labs/200031>

* + - Overview
      * BigQuery ML (BigQuery machine learning) is a feature in BigQuery where data analysts can create, train, evaluate, and predict with machine learning models with minimal coding.
      * The Google Analytics Sample Ecommerce dataset that has millions of Google Analytics records for the Google Merchandise Store loaded into BigQuery. In this lab, you will use this data to run some typical queries that businesses would want to know about their customers' purchasing habits.
    - Objectives
      * Use BigQuery to find public datasets
      * Query and explore the ecommerce dataset
      * Create a training and evaluation dataset to be used for batch prediction
      * Create a classification (logistic regression) model in BigQuery ML
      * Evaluate the performance of your machine learning model
      * Predict and rank the probability that a visitor will make a purchase
    - Open BigQuery Console
      * Navigation menu > BigQuery > Welcome msg > Done
    - Access the course dataset
      * Once BigQuery is open, open data-to-insights project in a new browser tab to bring this project into your BigQuery projects panel. -https://console.cloud.google.com/bigquery?p=data-to-insights&d=ecommerce&t=web\_analytics&page=table
      * The field definitions for the data-to-insights ecommerce dataset are here. Keep the link open in a new tab for reference. https://support.google.com/analytics/answer/3437719?hl=en
    - Task 1. Explore ecommerce data
      * Scenario: Your data analyst team exported the Google Analytics logs for an ecommerce website into BigQuery and created a new table of all the raw ecommerce visitor session data for you to explore. Using this data, you'll try to answer a few questions.
      * Question: Out of the total visitors who visited our website, what % made a purchase?
        + Click the query EDITOR.

#standardSQL

WITH visitors AS(

SELECT

COUNT(DISTINCT fullVisitorId) AS total\_visitors

FROM `data-to-insights.ecommerce.web\_analytics`

),

purchasers AS(

SELECT

COUNT(DISTINCT fullVisitorId) AS total\_purchasers

FROM `data-to-insights.ecommerce.web\_analytics`

WHERE totals.transactions IS NOT NULL

)

SELECT

total\_visitors, total\_purchasers,

total\_purchasers / total\_visitors AS conversion\_rate

FROM visitors, purchasers

* + - * + The result: 2.69%
      * Question: What are the top 5 selling products?

SELECT

p.v2ProductName, p.v2ProductCategory,

SUM(p.productQuantity) AS units\_sold,

ROUND(SUM(p.localProductRevenue/1000000),2) AS revenue

FROM `data-to-insights.ecommerce.web\_analytics`,

UNNEST(hits) AS h,

UNNEST(h.product) AS p

GROUP BY 1, 2 ORDER BY revenue DESCLIMIT 5;

* + - * + The result:
* Row v2ProductName v2ProductCategory units\_sold revenue
* 1 Nest® Learning Thermostat 3rd Gen-USA - Stainless Steel Nest-USA 17651 870976.95
* 2 Nest® Cam Outdoor Security Camera - USA Nest-USA 16930 684034.55
* 3 Nest® Cam Indoor Security Camera - USA Nest-USA 14155 548104.47
* 4 Nest® Protect Smoke + CO White Wired Alarm-USA Nest-USA 6394 178937.6
* 5 Nest® Protect Smoke + CO White Battery Alarm-USA Nest-USA 6340 178572.4
  + - * Question: How many visitors bought on subsequent visits to the website?

# visitors who bought on a return visit (could have bought on first as well

WITH all\_visitor\_stats AS (

SELECT

fullvisitorid, # 741,721 unique visitors

IF(COUNTIF(totals.transactions > 0 AND totals.newVisits IS NULL) > 0, 1, 0) AS will\_buy\_on\_return\_visit

FROM `data-to-insights.ecommerce.web\_analytics`

GROUP BY fullvisitorid

)

SELECT

COUNT(DISTINCT fullvisitorid) AS total\_visitors,

will\_buy\_on\_return\_visit

FROM all\_visitor\_stats

GROUP BY will\_buy\_on\_return\_visit

* + - * + The results:
        + Row total\_visitors will\_buy\_on\_return\_visit
        + 1 729848 0
        + 2 11873 1
      * Analyzing the results, you can see that (11873 / 729848) = 1.6% of total visitors will return and purchase from the website. This includes the subset of visitors who bought on their very first session and then came back and bought again.
      * What are some of the reasons a typical ecommerce customer will browse but not buy until a later visit? Choose all that could apply.
        + The customer wants to comparison shop on other sites before making a purchase decision.
        + The customer is waiting for products to go on sale or other promotion
        + The customer is doing additional research
      * This behavior is very common for luxury goods where significant up-front research and comparison is required by the customer before deciding (think car purchases) but also true to a lesser extent for the merchandise on this site (t-shirts, accessories, etc).
      * In the world of online marketing, identifying and marketing to these future customers based on the characteristics of their first visit will increase conversion rates and reduce the outflow to competitor sites.
    - Task 2. Select features and create your training dataset
      * Now you will create a Machine Learning model in BigQuery to predict whether or not a new user is likely to purchase in the future. Identifying these high-value users can help your marketing team target them with special promotions and ad campaigns to ensure a conversion while they comparison shop between visits to your ecommerce site.
      * Google Analytics captures a wide variety of dimensions and measures about a user's visit on this ecommerce website. Browse the complete list of fields here <https://support.google.com/analytics/answer/3437719?hl=en> and then preview the demo dataset <https://bigquery.cloud.google.com/table/data-to-insights:ecommerce.web_analytics?tab=preview> to find useful features that will help a machine learning model understand the relationship between data about a visitor's first time on your website and whether they will return and make a purchase.
      * Your team decides to test whether these two fields are good inputs for your classification model:
        + totals.bounces (whether the visitor left the website immediately)
        + totals.timeOnSite (how long the visitor was on our website)
      * What are the risks of only using the above two fields?
        + Whether a user bounces is highly correlated with their time on site (e.g. 0 seconds)
        + Only using time spent on the site ignores other potential useful columns (features)
      * Machine learning is only as good as the training data that is fed into it. If there isn't enough information for the model to determine and learn the relationship between your input features and your label (in this case, whether the visitor bought in the future) then you will not have an accurate model. While training a model on just these two fields is a start, you will see if they're good enough to produce an accurate model.

SELECT

\* EXCEPT(fullVisitorId)

FROM

# features

(SELECT

fullVisitorId,

IFNULL(totals.bounces, 0) AS bounces,

IFNULL(totals.timeOnSite, 0) AS time\_on\_site

FROM

`data-to-insights.ecommerce.web\_analytics`

WHERE

totals.newVisits = 1)

JOIN

(SELECT

fullvisitorid,

IF(COUNTIF(totals.transactions > 0 AND totals.newVisits IS NULL) > 0, 1, 0) AS will\_buy\_on\_return\_visit

FROM

`data-to-insights.ecommerce.web\_analytics`

GROUP BY fullvisitorid)

USING (fullVisitorId)

ORDER BY time\_on\_site DESC LIMIT 10;

* + - * + Results:
        + Row bounces time\_on\_site will\_buy\_on\_return\_visit
        + 1 0 15047 0
        + 2 0 12136 0
        + 3 0 11201 0
        + 4 0 10046 0
        + 5 0 9974 0
        + 6 0 9564 0
        + 7 0 9520 0
        + 8 0 9275 1
        + 9 0 9138 0
        + 10 0 8872 0
      * Which fields are the model features? What is the label (correct answer)?
        + The features are bounces and time\_on\_site. The label is will\_buy\_on\_return\_visit
      * Which fields are known after a visitor's first session? (Check all that apply)
        + bounces
        + visitId
        + time\_on\_site
      * Which field isn't known until later in the future after their first session?
        + will\_buy\_on\_return\_visit
      * Discussion: will\_buy\_on\_return\_visit is not known after the first visit. Again, you're predicting for a subset of users who returned to your website and purchased. Since you don't know the future at prediction time, you cannot say with certainty whether a new visitor comes back and purchases. The value of building a ML model is to get the probability of future purchase based on the data gleaned about their first session.
      * Question: Looking at the initial data results, do you think time\_on\_site and bounces will be a good indicator of whether the user will return and purchase or not?
        + Answer: It's often too early to tell before training and evaluating the model, but at first glance out of the top 10 time\_on\_site, only 1 customer returned to buy, which isn't very promising. Let's see how well the model does.
    - Task 3. Create a BigQuery dataset to store models
      * Next, create a new BigQuery dataset which will also store your ML models.
      * select Create Dataset > In the Create Dataset dialog: For Dataset ID, type ecommerce.
      * Leave the other values at their defaults. Click Create dataset.
    - Task 4. Select a BigQuery ML model type and specify options
      * Now that you have your initial features selected, you are now ready to create your first ML model in BigQuery.
      * There are the two model types to choose from:

|  |  |  |  |
| --- | --- | --- | --- |
| **Model** | **Model Type** | **Label Data type** | **Example** |
| Forecasting | linear\_reg | Numeric value (typically an integer or floating point) | Forecast sales figures for next year given historical sales data. |
| Classification | logistic\_reg | 0 or 1 for binary classification | Classify an email as spam or not spam given the context. |

* + - * Note: There are many additional model types used in Machine Learning (like Neural Networks and decision trees) and available using libraries like TensorFlow. At the time of writing, BigQuery ML supports the two listed above.
      * Which model type should you choose that will buy or won't buy?
        + Classification model (like logistic\_reg etc.)
      * Enter the following query to create a model and specify model options:
        + CREATE OR REPLACE MODEL `ecommerce.classification\_model`
        + OPTIONS
        + (
        + model\_type='logistic\_reg',
        + labels = ['will\_buy\_on\_return\_visit']
        + )
        + AS
        + #standardSQL
        + SELECT
        + \* EXCEPT(fullVisitorId)
        + FROM
        + # features
        + (SELECT
        + fullVisitorId, IFNULL(totals.bounces, 0) AS bounces,
        + IFNULL(totals.timeOnSite, 0) AS time\_on\_site
        + FROM
        + `data-to-insights.ecommerce.web\_analytics`
        + WHERE
        + totals.newVisits = 1
        + AND date BETWEEN '20160801' AND '20170430') # train on first 9 months
        + JOIN
        + (SELECT
        + fullvisitorid,
        + IF(COUNTIF(totals.transactions > 0 AND totals.newVisits IS NULL) > 0, 1, 0) AS will\_buy\_on\_return\_visit
        + FROM
        + `data-to-insights.ecommerce.web\_analytics`
        + GROUP BY fullvisitorid) USING (fullVisitorId) ;
      * Next, click Run to train your model. Wait for the model to train (5 - 10 minutes).
      * Note: You cannot feed all of your available data to the model during training since you need to save some unseen data points for model evaluation and testing. To accomplish this, add a WHERE clause condition is being used to filter and train on only the first 9 months of session data in your 12 month dataset.
      * After your model is trained, you will see the message "This statement created a new model named qwiklabs-gcp-xxxxxxxxx:ecommerce.classification\_model".
      * Click Go to model.
      * Look inside the ecommerce dataset and confirm classification\_model now appears.
      * Next, you will evaluate the performance of the model against new unseen evaluation data.
    - Task 5. Evaluate classification model performance
      * Select your performance criteria
      * For classification problems in ML, you want to minimize the False Positive Rate (predict that the user will return and purchase and they don't) and maximize the True Positive Rate (predict that the user will return and purchase and they do).
      * This relationship is visualized with a ROC (Receiver Operating Characteristic) curve like the one shown here, where you try to maximize the area under the curve or AUC:
        + Chart, line chart

          Description automatically generated
      * In BigQuery ML, roc\_auc is simply a queryable field when evaluating your trained ML model.
      * Now that training is complete, you can evaluate how well the model performs by running this query using ML.EVALUATE:

SELECT

roc\_auc,

CASE

WHEN roc\_auc > .9 THEN 'good'

WHEN roc\_auc > .8 THEN 'fair'

WHEN roc\_auc > .7 THEN 'not great'

ELSE 'poor' END AS model\_quality

FROM

ML.EVALUATE(MODEL ecommerce.classification\_model, (

SELECT

\* EXCEPT(fullVisitorId)

FROM

# features

(SELECT

fullVisitorId,

IFNULL(totals.bounces, 0) AS bounces,

IFNULL(totals.timeOnSite, 0) AS time\_on\_site

FROM

`data-to-insights.ecommerce.web\_analytics`

WHERE

totals.newVisits = 1

AND date BETWEEN '20170501' AND '20170630') # eval on 2 months

JOIN

(SELECT

fullvisitorid,

IF(COUNTIF(totals.transactions > 0 AND totals.newVisits IS NULL) > 0, 1, 0) AS will\_buy\_on\_return\_visit

FROM

`data-to-insights.ecommerce.web\_analytics`

GROUP BY fullvisitorid) USING (fullVisitorId)));

* + - * You should see the following result:
        + Row roc\_auc model\_quality
        + 1 0.724588 not great
      * After evaluating your model you get a roc\_auc of 0.72, which shows that the model has not great predictive power. Since the goal is to get the area under the curve as close to 1.0 as possible, there is room for improvement.
    - Task 6. Improve model performance with feature engineering
      * As was hinted at earlier, there are many more features in the dataset that may help the model better understand the relationship between a visitor's first session and the likelihood that they will purchase on a subsequent visit.
      * Add some new features and create a second machine learning model called classification\_model\_2:
        + How far the visitor got in the checkout process on their first visit
        + Where the visitor came from (traffic source: organic search, referring site etc.)
        + Device category (mobile, tablet, desktop)
        + Geographic information (country)
      * Create this second model by running the below query:

CREATE OR REPLACE MODEL `ecommerce.classification\_model\_2`

OPTIONS

(model\_type='logistic\_reg', labels = ['will\_buy\_on\_return\_visit']) AS

WITH all\_visitor\_stats AS (

SELECT

fullvisitorid,

IF(COUNTIF(totals.transactions > 0 AND totals.newVisits IS NULL) > 0, 1, 0) AS will\_buy\_on\_return\_visit

FROM `data-to-insights.ecommerce.web\_analytics`

GROUP BY fullvisitorid

)

# add in new features

SELECT \* EXCEPT(unique\_session\_id) FROM (

SELECT

CONCAT(fullvisitorid, CAST(visitId AS STRING)) AS unique\_session\_id,

# labels

will\_buy\_on\_return\_visit,

MAX(CAST(h.eCommerceAction.action\_type AS INT64)) AS latest\_ecommerce\_progress,

# behavior on the site

IFNULL(totals.bounces, 0) AS bounces,

IFNULL(totals.timeOnSite, 0) AS time\_on\_site,

totals.pageviews,

# where the visitor came from

trafficSource.source, trafficSource.medium, channelGrouping,

# mobile or desktop

device.deviceCategory,

# geographic

IFNULL(geoNetwork.country, "") AS country

FROM `data-to-insights.ecommerce.web\_analytics`,

UNNEST(hits) AS h

JOIN all\_visitor\_stats USING(fullvisitorid)

WHERE 1=1

# only predict for new visits

AND totals.newVisits = 1

AND date BETWEEN '20160801' AND '20170430' # train 9 months

GROUP BY

unique\_session\_id, will\_buy\_on\_return\_visit, bounces, time\_on\_site,

totals.pageviews, trafficSource.source, trafficSource.medium,

channelGrouping, device.deviceCategory, country);

* + - * Note: You are still training on the same first 9 months of data, even with this new model. It's important to have the same training dataset so you can be certain a better model output is attributable to better input features and not new or different training data.
      * A key new feature that was added to the training dataset query is the maximum checkout progress each visitor reached in their session, which is recorded in the field hits.eCommerceAction.action\_type. If you search for that field in the field definitions you will see the field mapping of 6 = Completed Purchase.
      * As an aside, the web analytics dataset has nested and repeated fields like ARRAYS which need to be broken apart into separate rows in your dataset. This is accomplished by using the UNNEST() function, which you can see in the above query.
      * Wait for the new model to finish training (5-10 minutes).
      * Evaluate this new model to see if there is better predictive power by running the below query:
    - #standardSQL
    - SELECT
    - roc\_auc,
    - CASE
    - WHEN roc\_auc > .9 THEN 'good'
    - WHEN roc\_auc > .8 THEN 'fair'
    - WHEN roc\_auc > .7 THEN 'not great'
    - ELSE 'poor' END AS model\_quality
    - FROM
    - ML.EVALUATE(MODEL ecommerce.classification\_model\_2, (
    - WITH all\_visitor\_stats AS (
    - SELECT
    - fullvisitorid,
    - IF(COUNTIF(totals.transactions > 0 AND totals.newVisits IS NULL) > 0, 1, 0) AS will\_buy\_on\_return\_visit
    - FROM `data-to-insights.ecommerce.web\_analytics`
    - GROUP BY fullvisitorid
    - )
    - # add in new features
    - SELECT \* EXCEPT(unique\_session\_id) FROM (
    - SELECT
    - CONCAT(fullvisitorid, CAST(visitId AS STRING)) AS unique\_session\_id,
    - # labels
    - will\_buy\_on\_return\_visit,
    - MAX(CAST(h.eCommerceAction.action\_type AS INT64)) AS latest\_ecommerce\_progress,
    - # behavior on the site
    - IFNULL(totals.bounces, 0) AS bounces,
    - IFNULL(totals.timeOnSite, 0) AS time\_on\_site,
    - totals.pageviews,
    - # where the visitor came from
    - trafficSource.source, trafficSource.medium, channelGrouping,
    - # mobile or desktop
    - device.deviceCategory,
    - # geographic
    - IFNULL(geoNetwork.country, "") AS country
    - FROM `data-to-insights.ecommerce.web\_analytics`,
    - UNNEST(hits) AS h
    - JOIN all\_visitor\_stats USING(fullvisitorid)
    - WHERE 1=1
    - # only predict for new visits
    - AND totals.newVisits = 1
    - AND date BETWEEN '20170501' AND '20170630' # eval 2 months
    - GROUP BY
    - unique\_session\_id, will\_buy\_on\_return\_visit, bounces, time\_on\_site,
    - totals.pageviews, trafficSource.source, trafficSource.medium,
    - channelGrouping, device.deviceCategory, country ) ));
      * (Output)
        + Row roc\_auc model\_quality
        + 1 0.910382 good
      * With this new model you now get a roc\_auc of 0.91 which is significantly better than the first model.
      * Now that you have a trained model, time to make some predictions.
    - Task 7. Predict which new visitors will come back and purchase
      * Next you will write a query to predict which new visitors will come back and make a purchase.
      * Run the prediction query below which uses the improved classification model to predict the probability that a first-time visitor to the Google Merchandise Store will make a purchase in a later visit:
    - SELECT
    - \*
    - FROM
    - ml.PREDICT(MODEL `ecommerce.classification\_model\_2`,
    - (
    - WITH all\_visitor\_stats AS (
    - SELECT
    - fullvisitorid,
    - IF(COUNTIF(totals.transactions > 0 AND totals.newVisits IS NULL) > 0, 1, 0) AS will\_buy\_on\_return\_visit
    - FROM `data-to-insights.ecommerce.web\_analytics`
    - GROUP BY fullvisitorid
    - )
    - SELECT
    - CONCAT(fullvisitorid, '-',CAST(visitId AS STRING)) AS unique\_session\_id,
    - # labels
    - will\_buy\_on\_return\_visit,
    - MAX(CAST(h.eCommerceAction.action\_type AS INT64)) AS latest\_ecommerce\_progress,
    - # behavior on the site
    - IFNULL(totals.bounces, 0) AS bounces,
    - IFNULL(totals.timeOnSite, 0) AS time\_on\_site,
    - totals.pageviews,
    - # where the visitor came from
    - trafficSource.source, trafficSource.medium, channelGrouping,
    - # mobile or desktop
    - device.deviceCategory,
    - # geographic
    - IFNULL(geoNetwork.country, "") AS country
    - FROM `data-to-insights.ecommerce.web\_analytics`,
    - UNNEST(hits) AS h
    - JOIN all\_visitor\_stats USING(fullvisitorid)
    - WHERE
    - # only predict for new visits
    - totals.newVisits = 1
    - AND date BETWEEN '20170701' AND '20170801' # test 1 month
    - GROUP BY
    - unique\_session\_id, will\_buy\_on\_return\_visit, bounces, time\_on\_site, totals.pageviews, trafficSource.source, trafficSource.medium, channelGrouping, device.deviceCategory, country ) )
    - ORDER BY predicted\_will\_buy\_on\_return\_visit DESC;
      * The predictions are made in the last 1 month (out of 12 months) of the dataset.
      * Your model will now output the predictions it has for those July 2017 ecommerce sessions. You can see three newly added fields:
        + predicted\_will\_buy\_on\_return\_visit: whether the model thinks the visitor will buy later (1 = yes)
        + predicted\_will\_buy\_on\_return\_visit\_probs.label: the binary classifier for yes / no
        + predicted\_will\_buy\_on\_return\_visit\_probs.prob: the confidence the model has in it's prediction (1 = 100%)
    - Table

      Description automatically generated
    - Results
      * Of the top 6% of first-time visitors (sorted in decreasing order of predicted probability), more than 6% make a purchase in a later visit.
      * These users represent nearly 50% of all first-time visitors who make a purchase in a later visit.
      * Overall, only 0.7% of first-time visitors make a purchase in a later visit.
      * Targeting the top 6% of first-time increases marketing ROI by 9x vs targeting them all!
    - Additional information
      * roc\_auc is just one of the performance metrics available during model evaluation. Also available are accuracy, precision, and recall. Knowing which performance metric to rely on is highly dependent on what your overall objective or goal is.
    - Congratulations!
      * You created a machine learning model using just SQL.
    - Challenge
      * Summary
        + In the previous two tasks you saw the power of feature engineering at work in improving our models performance. However, we still may be able to improve our performance by exploring other model types. For classification problems, BigQuery ML also supports the following model types:

Deep Neural Networks .

Boosted Decision Trees (XGBoost) .

AutoML Tables Models .

Importing Custom TensorFlow Models .

* + - * Task
        + Though our linear classification (logistic regression) model performed well after feature engineering, it may be too simple of a model to fully capture the relationship between the features and the label. Using the same dataset and labels as you did in Task 6 to create the model ecommerce.classification\_model\_2, your challenge is to create a XGBoost Classifier.
        + Hint : Use following options for Boosted\_Tree\_Classifier:

1. L2\_reg = 0.1

2. num\_parallel\_tree = 8

3. max\_tree\_depth = 10

* + - * + You may need to look at the documentation linked above to see the exact syntax. The model will take around 7 minutes to train. The solution can be found in the solution section below if you need help writing the query.
      * Solution:
        + This is the solution that you require in order to create a XGBoost Classifier.

CREATE OR REPLACE MODEL `ecommerce.classification\_model\_3`

OPTIONS

(model\_type='BOOSTED\_TREE\_CLASSIFIER' , l2\_reg = 0.1, num\_parallel\_tree = 8, max\_tree\_depth = 10,

labels = ['will\_buy\_on\_return\_visit']) AS

WITH all\_visitor\_stats AS (

SELECT

fullvisitorid,

IF(COUNTIF(totals.transactions > 0 AND totals.newVisits IS NULL) > 0, 1, 0) AS will\_buy\_on\_return\_visit

FROM `data-to-insights.ecommerce.web\_analytics`

GROUP BY fullvisitorid

)

# add in new features

SELECT \* EXCEPT(unique\_session\_id) FROM (

SELECT

CONCAT(fullvisitorid, CAST(visitId AS STRING)) AS unique\_session\_id,

# labels

will\_buy\_on\_return\_visit,

MAX(CAST(h.eCommerceAction.action\_type AS INT64)) AS latest\_ecommerce\_progress,

# behavior on the site

IFNULL(totals.bounces, 0) AS bounces,

IFNULL(totals.timeOnSite, 0) AS time\_on\_site,

totals.pageviews,

# where the visitor came from

trafficSource.source, trafficSource.medium, channelGrouping,

# mobile or desktop

device.deviceCategory,

# geographic

IFNULL(geoNetwork.country, "") AS country

FROM `data-to-insights.ecommerce.web\_analytics`,

UNNEST(hits) AS h

JOIN all\_visitor\_stats USING(fullvisitorid)

WHERE 1=1

# only predict for new visits

AND totals.newVisits = 1

AND date BETWEEN '20160801' AND '20170430' # train 9 months

GROUP BY

unique\_session\_id, will\_buy\_on\_return\_visit, bounces,

time\_on\_site, totals.pageviews, trafficSource.source,

trafficSource.medium, channelGrouping,

device.deviceCategory, country

);

* + - * + Let us now evaluate our model and see how we did.
    - #standardSQL
    - SELECT
    - roc\_auc,
    - CASE
    - WHEN roc\_auc > .9 THEN 'good'
    - WHEN roc\_auc > .8 THEN 'fair'
    - WHEN roc\_auc > .7 THEN 'not great'
    - ELSE 'poor' END AS model\_quality
    - FROM
    - ML.EVALUATE(MODEL ecommerce.classification\_model\_3, (
    - WITH all\_visitor\_stats AS (
    - SELECT
    - fullvisitorid,
    - IF(COUNTIF(totals.transactions > 0 AND totals.newVisits IS NULL) > 0, 1, 0) AS will\_buy\_on\_return\_visit
    - FROM `data-to-insights.ecommerce.web\_analytics`
    - GROUP BY fullvisitorid
    - )
    - # add in new features
    - SELECT \* EXCEPT(unique\_session\_id) FROM (
    - SELECT
    - CONCAT(fullvisitorid, CAST(visitId AS STRING)) AS unique\_session\_id,
    - # labels
    - will\_buy\_on\_return\_visit,
    - MAX(CAST(h.eCommerceAction.action\_type AS INT64)) AS latest\_ecommerce\_progress,
    - # behavior on the site
    - IFNULL(totals.bounces, 0) AS bounces,
    - IFNULL(totals.timeOnSite, 0) AS time\_on\_site,
    - totals.pageviews,
    - # where the visitor came from
    - trafficSource.source,
    - trafficSource.medium,
    - channelGrouping,
    - # mobile or desktop
    - device.deviceCategory,
    - # geographic
    - IFNULL(geoNetwork.country, "") AS country
    - FROM `data-to-insights.ecommerce.web\_analytics`,
    - UNNEST(hits) AS h
    - JOIN all\_visitor\_stats USING(fullvisitorid)
    - WHERE 1=1
    - # only predict for new visits
    - AND totals.newVisits = 1
    - AND date BETWEEN '20170501' AND '20170630' # eval 2 months
    - GROUP BY
    - unique\_session\_id,
    - will\_buy\_on\_return\_visit,
    - bounces,
    - time\_on\_site,
    - totals.pageviews,
    - trafficSource.source,
    - trafficSource.medium,
    - channelGrouping,
    - device.deviceCategory,
    - country
    - )
    - ));
      * Our roc\_auc has increased by about .02 to around .94!
      * Note : Your exact values will differ due to the randomness involved in the training process.
      * It’s a small change in the roc\_auc, but note that since 1 is a perfect roc\_auc, it gets more difficult to improve the metric the closer to 1 it gets.
      * This is a great example of how easy it is in BigQuery ML to try out different model types with different options to see how they perform. We were able to use a much more complex model type by only changing one line of SQL.
      * One may reasonably ask “Where did the choices for these options come from?”, and the answer is experimentation! When you are trying to find the best model type for your problems, then one has to experiment with different sets of options in a process known as hyperparameter tuning.
      * Let’s finish up by generating predictions with our improved model and see how they compare to those we generated before. By using a Boosted tree classifier model, you can observe a slight improvement of 0.2 in our ROC AUC compared to the previous model. The query below will predict which new visitors will come back and make a purchase.
    - SELECT
    - \*
    - FROM
    - ml.PREDICT(MODEL `ecommerce.classification\_model\_3`,
    - (
    - WITH all\_visitor\_stats AS (
    - SELECT
    - fullvisitorid,
    - IF(COUNTIF(totals.transactions > 0 AND totals.newVisits IS NULL) > 0, 1, 0) AS will\_buy\_on\_return\_visit
    - FROM `data-to-insights.ecommerce.web\_analytics`
    - GROUP BY fullvisitorid
    - )
    - SELECT
    - CONCAT(fullvisitorid, '-',CAST(visitId AS STRING)) AS unique\_session\_id,
    - # labels
    - will\_buy\_on\_return\_visit,
    - MAX(CAST(h.eCommerceAction.action\_type AS INT64)) AS latest\_ecommerce\_progress,
    - # behavior on the site
    - IFNULL(totals.bounces, 0) AS bounces,
    - IFNULL(totals.timeOnSite, 0) AS time\_on\_site,
    - totals.pageviews,
    - # where the visitor came from
    - trafficSource.source,
    - trafficSource.medium,
    - channelGrouping,
    - # mobile or desktop
    - device.deviceCategory,
    - # geographic
    - IFNULL(geoNetwork.country, "") AS country
    - FROM `data-to-insights.ecommerce.web\_analytics`,
    - UNNEST(hits) AS h
    - JOIN all\_visitor\_stats USING(fullvisitorid)
    - WHERE
    - # only predict for new visits
    - totals.newVisits = 1
    - AND date BETWEEN '20170701' AND '20170801' # test 1 month
    - GROUP BY
    - unique\_session\_id,
    - will\_buy\_on\_return\_visit,
    - bounces,
    - time\_on\_site,
    - totals.pageviews,
    - trafficSource.source,
    - trafficSource.medium,
    - channelGrouping,
    - device.deviceCategory,
    - country
    - )
    - )
    - ORDER BY
    - predicted\_will\_buy\_on\_return\_visit DESC;
      * + A picture containing graphical user interface

          Description automatically generated
      * The output now shows a classification model that can better predict the probability that a first-time visitor to the Google Merchandise Store will make a purchase in a later visit. By comparing the result above with the previous model shown in Task 7, you can see the confidence the model has in its predictions is more accurate when compared to the logistic\_regression model type.

### Quiz

* + - You want to use machine learning to identify whether an email is spam. Which should you use?
      * Supervised learning, logistic regression
    - Data has been loaded into BigQuery, and the features have been selected and preprocessed. What should happen next when you use BigQuery ML to develop a machine learning model?
      * Create the ML model inside BigQuery.
    - You want to use machine learning to group random photos into similar groups. Which should you use?
      * Unsupervised learning, cluster analysis
    - Which pattern describes source data that is moved into a BigQuery table in a single operation?
      * Batch load
    - BigQuery is a fully managed data warehouse. What does “fully managed” refer to?
      * BigQuery manages the underlying structure for you.
    - In a supervised machine learning model, what provides historical data that can be used to predict future data?
      * Labels
    - Which two services does BigQuery provide?
      * Storage and analytics
    - Which BigQuery feature leverages geography data types and standard SQL geography functions to analyze a data set?
      * Geospatial analysis
  + Reading list

### Machine Learning Options on Google Cloud

* + This section explores four different options to build machine learning models on Google Cloud. It also introduces Vertex AI, Google's unified platform for building and managing the lifecycle of ML projects.
  + Options to build ML models 3 minutes- <https://youtu.be/k7KCxdBProQ>
    - google cloud offers four options for building machine learning models
      * 1. bigquery ml, is a tool for using sql queries to create and execute ML models in bigquery
        + if you already have your data in bigquery and your problems fit the predefined ml models this could be your choice
      * 2. pre-built apis which are APIs this option lets you leverage ML models that have already been built and trained by google
        + so you don't have to build your own ML models if you don't have enough training data or sufficient ML expertise in-house
      * 3. automl which is a no code solution so you can build your own ML models on vertex ai through a point-and-click interface and
      * 4. custom training through which you can code your very own ML environment the training and the deployment which gives you flexibility and provides full control over the entire process
    - let's compare the four options to help you decide which one to use for building your ml model data type
      * Table

        Description automatically generated
      * can experiment with hyper parameters using bigquery ml and custom
      * time to train the model pre-built apis require no time to train a model because they directly use pre-built models from google
      * Graphical user interface, application

        Description automatically generated
      * if your business users or developers have little ml experience using pre-built apis is likely the best choice
  + Pre-built APIs 2 minutes - https://youtu.be/ugJ0SBjbomU
    - good ML models require lots of high quality training data, if you don't have pre-built apis are a great place to start
      * the speech-to-text api converts audio to text for data processing
      * the cloud natural language api recognizes parts of speech called entities and sentiment
      * the cloud translation api converts text from one language to another
      * the text-to-speech api converts text into high quality voice audio
      * the vision api works with and recognizes content in static images and
      * the video intelligence api recognizes motion and action in video
    - google has already done a lot of work to train these models using google data sets for example
      * the vision api is based on google's image data sets
      * the speech-to-text api is trained on youtube captions and
      * the translation api is built on google translations
    - let's try out the vision api cloud.google.com/vision
      * Graphical user interface, application, website

        Description automatically generated
  + AutoML 6 minutes - automated machine learning - https://youtu.be/3xSwkTOInO4
    - let's briefly look, the training and deploying ml models can be extremely time consuming because you need to repeatedly add new data and features, try different models and tune parameters to achieve the best result to solve this problem use automl
    - the goal was to AutoML so data scientists didn't have to start the process from scratch but how could this be done
      * well ML is similar to human learning it all starts with gathering the right information
    - automl two technologies are vital
      * 1. transfer learning - you build a knowledge base in the field (ex gathering lots of books to create a library)
        + It’s a powerful technique that lets people with smaller data sets or less computational power achieve state-of-the-art results by taking advantage of pre-trained models that have been trained on similar larger data sets
      * 2. neural architect search , is to find the optimal model for the relevant project (ex finding the best book in the library to help you learn what you need to)
    - automl supports four types of data - image, tabular, text and video
      * upload your data into automl it can come from cloud storage, bigquery or even your local machine
      * from there inform automl of the problems you want to solve
      * some problems may sound similar to those mentioned in pre-built apis
        + the major difference is that pre-built apis use pre-built machine learning models but automl uses custom built models
      * use your own data to train the ML model and then apply the trained model to predict your goal
    - image data, can use a model to analyze image data
      * classification model - return a list of content categories that apply to the image
        + like, train a model that classifies images as containing a dog or not, dogs by breed
      * object detection model - return annotations that consist of a label and bounding box location for each object found in an image
        + like, train a model to find the location of the dogs in image data
    - tabular data
      * regression model - return a numeric value
        + train a model to estimate a house's value or rental price based on a set of factors such as location , size of the house and number of bedrooms
      * classification model - return a list of categories
        + train a model to classify different types of land into high, median and low potentials for commercial real estate
      * forecasting model - can use multiple rows of time-dependent tabular data from the past to predict a series of numeric values in the future
        + use the historical plus the economic data to predict what the housing market will look like in the next five years
    - text data
      * classification model - return a list of categories that's applied to the text found in the data
        + classify customer questions and comments to different categories and then redirect them to corresponding departments
      * entity extraction model can be used to inspect text data for known entities referenced in the data and label those entities in the text
        + label a social media post in terms of predefined entities such as time, location and topic this can help with online search similar to the concept of a hashtag but created by a machine
      * sentiment analysis model can be used to inspect text data and identify the prevailing emotional opinion within it
        + especially to determine a writer's comment as positive negative or neutral
    - video data
      * classification model - return a list of categorized shots and segments
        + train a model that analyzes video data to identify whether the video is of a soccer baseball, basketball or football game
      * object tracking model - return a list of shots and segments where these objects were detected
        + train a model that analyzes video data from soccer games to identify and track the ball
      * action recognition model - return a list of categorized actions with the moments the actions happened
        + train a model that analyzes video data to identify the action moments involving a soccer goal, a golf swing, a touchdown or a high five
  + Custom training 1 minute - https://youtu.be/IfnA77ea5R0
    - if you want to code your ML model you can use this option by building a custom training solution with vertex ai workbench
      * workbench is a single development environment for the entire data science workflow from exploring to training and then deploying a ML model with code
    - determine what environment you want your ml training code to use , 2 options
      * a pre-built container or a custom container, imagine that a container is a kitchen
        + a pre-built container would represent a fully furnished room with cabinets and appliances which represent the dependencies that includes all the cookware which represents the libraries you need to make a meal

so if your ml training needs a platform like tensorflow, pytorch, scikit-learn, xgboost and python code to work with the platform a pre-built container is probably your best solution

* + - * a custom container alternatively is like an empty room with no cabinets appliances or cookware you define the exact tools you need to complete the job
  + Vertex AI 3 minutes - https://youtu.be/c2NaWuWxjkA
    - Google developed
    - Timeline

      Description automatically generated
    - Traditional challenges
      * determining how to handle large quantities of data
      * determining the right machine learning model to train the data
      * harnessing the required amount of computing power
    - production challenges
      * scalability, monitoring and CI/CD or deployment
      * these challenges can make projects fail
    - ease of use challenges
      * many tools on the market require advanced coding skills
      * which can take a data scientist's focus away from model configuration
      * and without a unified workflow
      * data scientists often have difficulties finding tools
    - google's solution to many of the production and ease of use challenges is vertex ai
      * a unified platform that brings all the components of the machine learning ecosystem and workflow together
      * a unified platform mean in the case of vertex ai, having one digital experience to create, deploy and manage models over time and at scale
      * for example
      * Timeline

        Description automatically generated with medium confidence
    - vertex ai allows users to build ML models with either automl (a codeless solution) or custom training (a code-based solution)
    - summarized vertex - seamless, scalable, sustainable, speedy
  + AI Solutions 2 minutes - https://youtu.be/2sYDr9FcPc8
    - look at google cloud's AI solution portfolio
      * Graphical user interface, website

        Description automatically generated document ai uses computer vision and optical character recognition along with natural language processing to create pre-trained models to extract information from documents
        + the goal is to increase the speed and accuracy of document processing to help organizations make better decisions faster while reducing costs
      * contact center ai (ccai) - to improve customer service in contact centers through the use of AI
        + it can help automate simple interactions, assist human agents, unlock caller insights and provide information to answer customer questions
      * retail product discovery which gives retailers the ability to provide google quality search and recommendations on their own digital properties helping to increase conversions and reduce search abandonment
      * google cloud healthcare data engine which generates healthcare insights and analytics with one end-to-end solution
      * lending.ai which aims to transform the home loan experience for borrowers and lenders by automating mortgage document processing
      * learn more from cloud.google.com/solutions/ai
  + Quiz
    - You work for a global hotel chain that has recently loaded some guest data into BigQuery. You have experience writing SQL and want to leverage machine learning to help predict guest trends for the next few months. Which option is best?
      * BigQuery ML
    - You work for a video production company and want to use machine learning to categorize event footage, but want to train your own ML model. Which option can help you get started?
      * Pre-built APIs
    - Which Google Cloud product lets users create, deploy, and manage machine learning models in one unified platform?
      * Vertex AI
    - Your company has a lot of data, and you want to train your own machine model to see what insights ML can provide. Due to resource constraints, you require a codeless solution. Which option is best?
      * AutoML
    - Which code-based solution offered with Vertex AI gives data scientists full control over the development environment and process?
      * Custom training
  + Reading list

### The Machine Learning Workflow with Vertex AI

* + This section focuses on the three key phases--data preparation, model training, and model preparation--of the machine learning workflow in Vertex AI. Learners get the opportunity to practice building a machine learning model with AutoML.
    - Graphical user interface

      Description automatically generated with medium confidence

### automl workflow - 1st stage Data preparation 3 minutes, <https://youtu.be/_ZLS-wg9TVU> , 2 steps

* + - upload data in the vertex ai user interface
      * provide a meaningful name for the data
      * select the data type and objective
        + 4 types of data - image, tabular, text and video
        + to select the correct data type and objective you
      * checking data requirements (included a link)
      * add labels to the data
      * Graphical user interface, application, Teams

        Description automatically generated
        + if you want a model to distinguish a cat from a dog you must first provide sample images that are tagged or labeled either cat or dog
      * upload the data
        + uploaded from a local source, bigquery or cloud storage
    - and then prepare the data for model training with feature engineering
      * preparing a meal your data is like your ingredients such as carrots, onions and tomatoes, before you start cooking you need to peel the carrots, chop the onions and rinse the tomatoes
        + Chart

          Description automatically generated
      * this is what feature engineering is like the data must be processed before the model starts training,
      * a feature refers to a factor that contributes to the prediction
        + it's an independent variable in statistics or a column in a table
        + preparing features can be both challenging and tedious
        + vertex ai has a function called feature store, is a centralized repository to organize, store and serve ML features

aggregates & update features

* + - * + benefits of vertex ai feature store

features are shareable for training or serving tasks

features are reusable

features are scalable

features are easy to use

* + Model training 3 minutes - <https://youtu.be/aPN7Boph8i8>, 2 stpes
    - model training which would be like cooking the recipe
    - model evaluation which is when we taste how good the meal, is this process might be iterative
    - Diagram

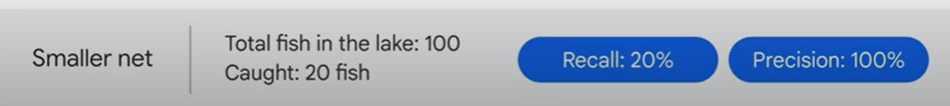
      Description automatically generated
      * ML is a subset of AI
    - difference between supervised and unsupervised learning
      * Diagram

        Description automatically generated with low confidence
    - google cloud provides 4 ML options
      * automl and pre-built apis you don't need to specify a ML model instead you'll define your objective such as text translation or image detection then on the back end google will select the best model to meet your business goal
      * bigquery ml and custom training you'll need to specify which model you want to train your data on and assign something called hyper parameters
        + hyper parameters as user-defined knobs in a machine that helps guide the ML process
        + for example one parameter is a learning rate which is how fast you want the machine to learn
        + with automl you don't need to worry about adjusting these hyperparameter knobs because the tuning happens automatically on the back end
        + this is largely done by a neural architect search which finds the best fit model by comparing the performance against thousands of other models
  + Model evaluation 4 minutes
    - while we're experimenting with a recipe we need to keep tasting it constantly to make sure it meets our expectations
    - this is the model evaluation portion of the model training stage
    - vertex ai provides extensive evaluation metrics to help determine a model's performance among the metrics are two sets of measurements
      * 1st confusion matrix - for example recall and precision
        + It’s a specific performance measurement for ML classification problems
        + it's a table with combinations of predicted and actual values
        + to keep things simple we can assume the output includes only two classes
        + Graphical user interface, application

          Description automatically generated
        + Graphical user interface, text, application

          Description automatically generated
        + Graphical user interface, text, chat or text message

          Description automatically generated
        + A picture containing chart

          Description automatically generated
        + 
      * gmail separates emails into two categories spam and not spam
        + if the goal is to catch as many potential spam emails as possible gmail may want to prioritize recall
        + in contrast if the goal is to only catch the messages that were definitely spam without blocking other emails gmail may want to prioritize precision
        + in vertex ai the platform visualizes the precision in the recall curve so they can be adjusted based on the problem that needs solving
      * 2nd feature importance
        + Chart

          Description automatically generated in virtuxii feature importance is displayed through a bar chart to illustrate how each feature contributes to a prediction

the longer the bar or the larger the numerical value associated with a feature the more important

it is this information helps decide which features are included in a machine learning model to predict the goal

* + - * + feature importance is just one example of vertex ai's comprehensive machine learning functionality called explainable ai explainable ai is a set of tools and frameworks to help understand and interpret predictions made by machine learning models
  + Model deployment and monitoring 3 minutes
    - the recipes are ready and now it's time to serve the meal
    - this represents the final stage of the ML workflow model serving, consists of two steps
      * 1. model deployment which we can compare to serving the meal to a hungry customer
      * 2. model monitoring which we can compare to checking with the wait staff to ensure that the restaurant is operating efficiently
    - ML operations or mlaps play a big role
      * mlapse combines ML development with operations and applies similar principles from devops to machine learning models which is short for development and operations
      * mlaps aims to solve production challenges related to ML, in this case this refers to building an integrated ML system and operating it in production
      * practicing mlaps - continuous integration, continuous training and continuous delivery
    - 3 options to deploy a machine learning model
      * 1. deploy to an endpoint
        + making instant recommendations based on a user's browsing habits whenever they're online
      * 2. deploy using batch prediction
        + sending out new ads every other week based on the user's recent purchasing behavior and what's currently popular on the market
      * 3. deploy using offline prediction
    - model monitoring
      * the backbone of ml ops on vertex ai is a tool called vertex ai pipelines
        + it automates, monitors and governs ML systems by orchestrating the workflow in a serverless manner
      * imagine, vertex ai pipelines is displaying the production data on screen if something goes wrong it automatically triggers warnings based on a predefined threshold
      * with vertex ai workbench
        + you can define your own pipeline
        + you can do this with pre-built pipeline components which means that you primarily need to specify how the pipeline is put together using components as building blocks

### Vertex AI: Predicting Loan Risk with AutoML 2 hours 31 minutes

* + - Overview
      * In this lab, you use Vertex AI to train and serve a machine learning model to predict loan risk with a tabular dataset.
    - Objectives
      * Upload a dataset to Vertex AI.
      * Train a machine learning model with AutoML.
      * Evaluate the model performance.
      * Deploy the model to an endpoint.
      * Get predictions.
    - Introduction to Vertex AI
      * This lab uses Vertex AI, the unified AI platform on Google Cloud to train and deploy a ML model. Vertex AI offers two options on one platform to build a ML model: a codeless solution with AutoML and a code-based solution with Custom Training using Vertex Workbench. You use AutoML in this lab.
      * In this lab you build a ML model to determine whether a particular customer will repay a loan.
    - Task 1: Prepare the training data
      * The initial Vertex AI dashboard illustrates the major stages to train and deploy a ML model: prepare the training data, train the model, and get predictions. Later, the dashboard displays your recent activities, such as the recent datasets, models, predictions, endpoints, and notebook instances.
      * Create a dataset
        + Navigation > Vertex AI > Create dataset

On the Datasets page, give the dataset a name. LoadRisk

For the data type and objective, click Tabular, and then select Regression/classification.

* + - * + Click Create.
      * Upload data
        + Three options to import data in Vertex AI:

Upload a local file from your computer.

Select files from Cloud Storage.

Select data from BigQuery.

* + - * + For convenience, the dataset is already uploaded to Cloud Storage.
        + For the data source, select Select CSV files from Cloud Storage.
        + For Import file path, enter

spls/cbl455/loan\_risk.csv

* + - * + Click Continue.
      * (Optional) Generate statistics
        + To see the descriptive statistics for each column of your dataset, click Generate statistics . Generating the statistics might take a few minutes, especially the first time.
        + When the statistics are ready, click each column name to display analytical charts.

Chart

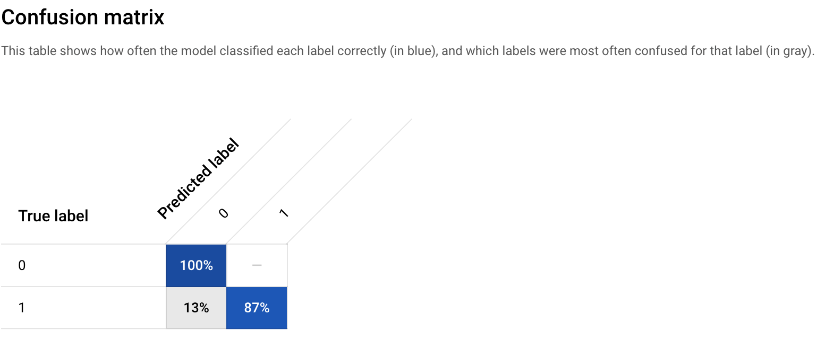
Description automatically generated

* + - Task 2: Train your model
      * With a dataset uploaded, you're ready to train a model to predict whether a customer will repay the loan.
        + Click Train new model.
      * Training method
        + The dataset is called LoanRisk.
        + For Objective, select Classification. Select classification instead of regression because you are predicting a distinct number (whether a customer will repay a loan: 0 for repay, 1 for default/not repay) instead of a continuous number.
        + Click Continue.
      * Model details
        + Specify the name of the model and the target column.
        + Give the model a name, such as LoanRisk.
        + For Target column, select Default .
        + (Optional) Explore Advanced options to determine how to assign the training vs. testing data and specify the encryption.
        + Click Continue.
      * Training options
        + Specify which columns you want to include in the training model. For example, ClientID might be irrelevant to predict loan risk.
        + Click the minus sign on the ClientID row to exclude it from the training model.
        + (Optional) Explore Advanced options to select different optimization objectives. For more information about optimization objectives for tabular AutoML models, see https://cloud.google.com/vertex-ai/docs/training/tabular-opt-obj.
        + Click Continue.
      * Compute and pricing
        + For Budget, which represents the number of node hours for training, enter 1. Training your AutoML model for 1 compute hour is typically a good start for understanding whether there is a relationship between the features and label you've selected. From there, you can modify your features and train for more time to improve model performance.
        + Leave early stopping enabled.
        + Click Start training.
      * Depending on the data size and the training method, the training can take from a few minutes to a couple of hours. Normally you would receive an email from Google Cloud when the training job is complete. However, in the Qwiklabs environment, you will not receive an email.
      * To save the waiting for the model training, you download a pre-trained model in task 5 to get predictions in task 6. This pre-trained model is the training result following the same steps from task 1 to task 2.
    - Task 3: Evaluate the model performance (demonstration only)
      * Veretex AI provides many metrics to evaluate the model performance. You focus on three:
        + Precision/Recall curve
        + Confusion Matrix
        + Feature Importance
      * If you had a model trained, you could navigate to the Models tab in Vertex AI.
        + 1. Navigate to the Models.
        + 2. Click on the model you just trained.
        + 3. Browse the Evaluate tab.
      * However in this lab, you canskip this step since you use a pre-trained model.
      * The Precision/Recall curve

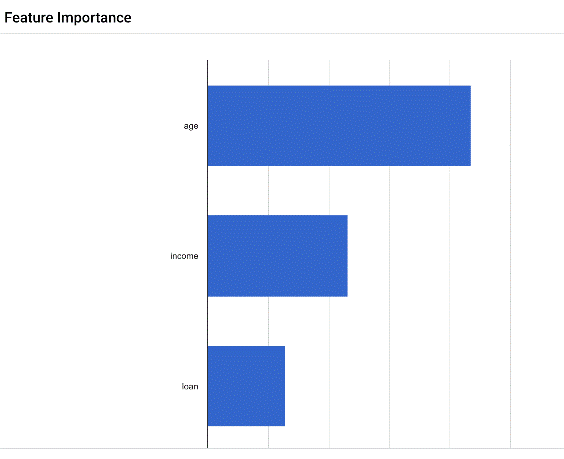
Chart

Description automatically generated

* + - * + The confidence threshold determines how a ML model counts the positive cases. A higher threshold increases the precision, but decreases recall. A lower threshold decreases the precision, but increases recall. You can manually adjust the threshold to observe its impact on precision and recall and find the best tradeoff point between the two to meet your business needs.
      * The confusion matrix
        + A confusion matrix tells you the percentage of examples from each class in your test set that your model predicted correctly.

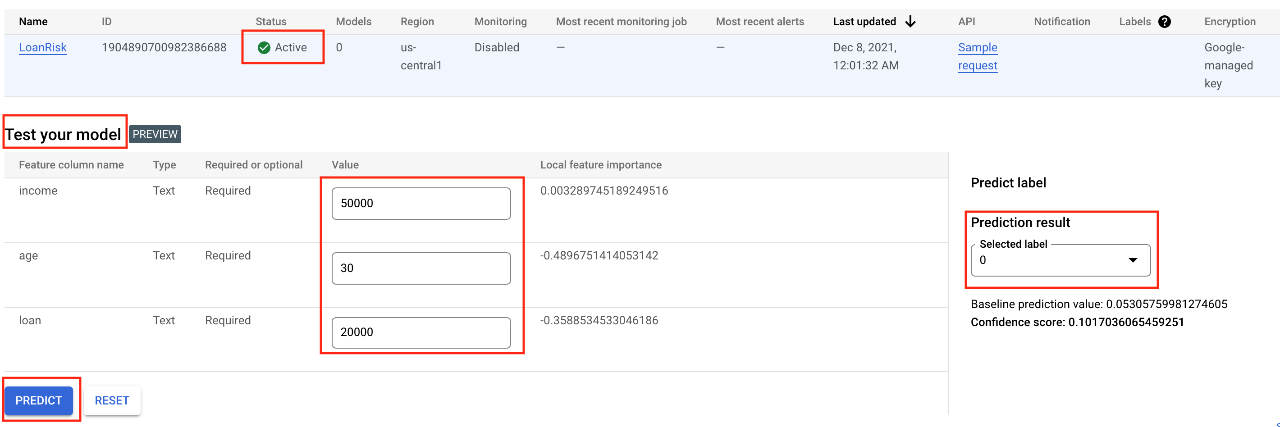


* + - * + The confusion matrix shows that your initial model is able to predict 100% of the repay examples and 87% of the default examples in your test set correctly, which is not too bad.
        + You can improve the precentage by adding more examples (more data), engineering new features, and changing the training method, etc.
      * The feature importance
        + In Vertex AI, feature importance is displayed through a bar chart to illustrate how each feature contributes to a prediction. The longer the bar, or the larger the numerical value associated with a feature, the more important it is.



* + - * + These feature importance values could be used to help you improve your model and have more confidence in its predictions. You might decide to remove the least important features next time you train a model or to combine two of the more significant features into a feature cross to see if this improves model performance.
        + Feature importance is just one example of Vertex AI’s comprehensive machine learning functionality called Explainable AI. Explainable AI is a set of tools and frameworks to help understand and interpret predictions made by machine learning models.
    - Task 4: Deploy the model (demonstration only)
      * You will not deploy the model to an endpoint because the model training can take an hour. Here you can review the steps you would perform in a production environment.
      * Now that you have a trained model, the next step is to create an endpoint in Vertex. A model resource in Vertex can have multiple endpoints associated with it, and you can split traffic between endpoints.
      * Create and define an endpoint
        + On your model page, on the Deploy and test tab, click Deploy to endpoint.
        + For Endpoint name, enter a name for your endpoint, such as LoanRisk.
        + Click Continue.
      * Model settings and monitoring
        + Leave the traffic splitting settings as-is.
        + As the machine type for your model deployment, under Machine type, select n1-standard-8, 8 vCPUs, 30 GiB memory.
        + Leave the remaining settings as-is.
        + Click Deploy.
      * Your endpoint will take a few minutes to deploy. When it is completed, a green check mark will appear next to the name.
      * Now you're ready to get predictions on your deployed model.
    - Task 5: SML Bearer Token
      * Retrieve your Bearer Token
        + To allow the pipeline to authenticate, and be authorized to call the endpoint to get the predictions, you will need to provide your Bearer Token.
        + Follow the instructions below to get your token. If you have issues getting the Bearer Token, this can be due to cookies in the incognito window. If this is happening to you, try this step in a non-incognito window.
        + Log in to https://gsp-auth-kjyo252taq-uc.a.run.app/
        + When logging in, use your student email address and password.
        + Click the Copy button. This will copy a very long token to your clipboard.
        + This token will only be available for about 60 seconds, so copy and and move on to the next steps.
        + If you have issues getting the Bearer Token, this can be due to cookies in the incognito window - try in a non-incognito window.
    - Task 6: Get predictions
      * In this section, use the Shared Machine Learning (SML) service to work with an existing trained model.

|  |  |
| --- | --- |
| **ENVIRONMENT VARIABLE** | **VALUE** |
| AUTH\_TOKEN | Use the value from the previous section |
| ENDPOINT | https://sml-api-vertex-kjyo252taq-uc.a.run.app/vertex/predict/tabular\_classification |
| INPUT\_DATA\_FILE | INPUT-JSON |

* + - * To use the trained model, you will need to create some environment variables.
      * Open a Cloud Shell window.
      * Replace INSERT\_SML\_BEARER\_TOKEN with the bearer token value from the previous section:
        + AUTH\_TOKEN="INSERT\_SML\_BEARER\_TOKEN"
      * Download the lab assets:
        + gsutil cp gs://spls/cbl455/cbl455.tar.gz .
      * Extract the lab assets:
        + tar -xvf cbl455.tar.gz
      * Create an ENDPOINT environment variable:
        + ENDPOINT="https://sml-api-vertex-kjyo252taq-uc.a.run.app/vertex/predict/tabular\_classification"
      * Create a INPUT\_DATA\_FILE environment variable:
        + INPUT\_DATA\_FILE="INPUT-JSON"
      * After the lab assets are extracted, take a moment to review the contents. The INPUT-JSON file is used to provide Vertex AI with the model data required.
      * Alter this file to generate custom predictions. The smlproxy application is used to communicate with the backend.
      * The file INPUT-JSON is composed of the following values:
        + age ClientID income loan
        + 40.77 997 44964.01 3944.22
      * Test the SML Service by passing the parameters specified in the environment variables:
      * Perform a request to the SML service:
        + ./smlproxy tabular \
        + -a $AUTH\_TOKEN \
        + -e $ENDPOINT \
        + -d $INPUT\_DATA\_FILE
      * This query should result in a response similar to this:
        + SML Tabular HTTP Response:
        + 2022/01/10 15:04:45 {"model\_class":"0","model\_score":0.9999981}
      * Alter the INPUT-JSON file to test a new scenario:
        + age ClientID income loan
        + 30.00 998 50000.00 20000.00
      * Test the SML Service by passing the parameters specified in the environment variables:
      * Edit the file INPUT-JSON and replace the original values.
      * Perform a request to the SML service:
        + ./smlproxy tabular \
        + -a $AUTH\_TOKEN \
        + -e $ENDPOINT \
        + -d $INPUT\_DATA\_FILE
      * In this case, assuming that the person's income is 50,000, age 30, and loan 20,000, the model predicts that this person will repay the loan.
      * This query should result in a response similar to this::
        + SML Tabular HTTP Response:
        + 2022/01/10 15:04:45 {"model\_class":"0","model\_score":0.9999981}
      * If you use the Google Cloud Console, the following image illustrates how the same action could be performed:
        + 
      * You can now use Vertex AI to:
        + Upload a dataset.
        + Train a model with AutoML.
        + Evaluate the model performance.
        + Deploy the trained AutoML model to an endpoint.
        + Get predictions.

### Lab recap: Predicting loan risk with AutoML 3 minutes <https://youtu.be/EgjGbwg73Ck>

* + - review the results of the lab starting with the confusion matrix
      * + A picture containing timeline

          Description automatically generated
      * the true positives were 100% - this represents the percentage of people the model predicted would repay their loan who actually did pay it back
      * the true negatives were 87% - this represents the percentage of people the model predicted would not repay their loan who indeed did not pay it
      * the false negatives were zero this represents the percentage of people the model predicted would not repay their loan but who actually did pay it back
      * the false positives were 13 percent this represents the percentage of people the model predicted would repay their loan but who actually did not pay it back
        + Table

          Description automatically generated with medium confidence
      * how high or low they need to be really depends on the business goals you're looking to achieve
    - review the precision recall curve from the automl lab the confidence threshold
      * determines how a machine learning model counts the positive cases
      * Graphical user interface, application, Teams

        Description automatically generated
      * Graphical user interface, application

        Description automatically generated
      * Graphical user interface, application, Teams

        Description automatically generated

### Quiz

* + - A hospital uses Google’s machine learning technology to help pre-diagnose cancer by feeding historical patient medical data to the model. The goal is to identify as many potential cases as possible. Which metric should the model focus on?
      * Recall
    - A farm uses Google’s machine learning technology to detect defective apples in their crop, such as those that are irregular in size or have scratches. The goal is to identify only the apples that are actually bad so that no good apples are wasted. Which metric should the model focus on?
      * Precision
    - Which Vertex AI tool automates, monitors, and governs machine learning systems by orchestrating the workflow in a serverless manner?
      * Vertex AI Pipelines
    - Select the correct machine learning workflow.
      * Data preparation, model training, model servin
    - Which stage of the machine learning workflow includes model evaluation?
      * Model training
    - Which stage of the machine learning workflow includes feature engineering?
      * Data preparation
  + Reading list