Project Summary:

The purpose of this project is to analyze the Google Analytics 360 dataset that is provided by Google and develop a classification model that can predict whether a hit session will yield an “add to cart” action based on various factors, with a business goal to increase the return on advertisement expenses.

This dataset is a collection of hit sessions that took place in the Google store, an ecommerce store, between August 2016 - August 2017. Some of the information included in this dataset are traffic source, contents data such as behavior of the users, and transactional data.

Vertai Workbench setup:

To set up a workbench in Vertex Ai, in Google Cloud Platform go to **Vertex AI** and navigate to **Workbench.** Click **CREATE NEW** and fill in the name, region, and zone. Leave everything else default. The configurations for this project’s instance is as below:

Name: khue-capstone
Region: us-east1
Zone: us-east1-b
Machine Type: e2-standard-4 (Efficient Instance: 4 vCPUs, 16 GB RAM)
GPU: None

Data Ingestion:

For the sake of time and cost, let's focus the training on just one month worth of data. For this model training, we are focusing on the month November, 2016. For future training, change the dates in the **WHERE clause** to the dates you want.

```
WHERE
  _TABLE_SUFFIX BETWEEN '20161101' AND '20161130'
```

There are columns that are nested as records, and within those nested columns there are more nested records, in which some are repeated records meaning they can appear in multiple rows. The reason for that is because each hit session could have multiple hits across different pages and products.

To select each and every unique session will result in too much data, therefore we need to aggregated some of this columns, for example if you look at this query:

```
SELECT
   CONCAT(fullVisitorId, visitId) AS session_id,
   MAX(CASE hit.eCommerceAction.action_type = '3' WHEN TRUE THEN 1 ELSE 0 END) AS add_to_cart
```

Note: for a full query, refer to the queries.sql file.

Our target is the ‘add to cart’ action which is within the hits nested columns. Here we set a condition to group the value 3 as 1 and any other values as 0, then we take the max out, resulting in a target column with just 0s and 1s.

The other columns we want to include are:

Date: date the session took place
Hits: total of hits during the session
Page_views: total pages viewed during session
Bounces: values are 1 and null, 1 means the session was bounced, and null means it’s not
Time_on_site: total time session was on the site
Hour: hour the session was on the site
Minute: minute the session was on the site
Device: the type of device the hit came from, either Mobile, Tablet, or Desktop
Sub_continent: e.g South America, North America, etc.
Country: country where the hit came from
Product_category: category of the product
product _name: name of product
product _price: local price of the product, times 10^6. E.g 35.56 will appear as 35560000

The reason we came down only to a few columns was because a lot of the features or fields were null or were not available on this particular dataset. Again, the goal of this project is to determine whether a hit will yield an “add to cart” action or not. If we look from a business perspective, we would want to focus on the features or factors that are more tangible or easier to understand. For example, we selected products because if a certain product tends to yield more “add to cart” we would want to direct more fundings toward it. The same logic can be applied to product price and sub continent.

Model Deployment:

After the data is processed, structured, and uploaded to Vertex AI, a model can be trained using AutoML. Refer to the Data Analysis Report to learn more about how the data was preprocessed and uploaded to Vertex AI. Follow the steps below to train and deploy the model to an endpoint for online predictions.

1. In **Vertex AI**, under **Model Development** click on **Training**. Then click on **Train New Model**.

2. For the **Dataset field** choose the dataset that you upload to Vertex AI, and for **Objective** choose Classification. Under **model training method** check **AutoML** and click Continue.

3. In the next section, check **Train new model**. Fill in the **Name** and pick the appropriate **Target column** which in this case would be **“add_to_cart”**, then click Continue.

4. Skip **Add features** and continue.

5. For the **Training Options** section leave everything default and continue.

6. For **Compute and Pricing** type in 1 in the Maximum node hours input. Then click on **START TRAINING.**

7. The training can take anywhere between 1 - 3 hours to finish training. Once the training is completed you can move onto the next step.

8. Under **Deploy and Use** click on **Online Prediction**. Enter a name and continue. Choose the model name(model we just trained), Version 1, and continue. Leave everything in **Model monitoring** default and continue. For **Monitoring objectives**, check Vertex AI dataset and pick the dataset that was trained for the model. Add the target column “add to cart”. Finally click **CREATE** to deploy the model to an endpoint.

To make a prediction request, navigate to the online-prediction.ipynb file. Change the values in the **instance_dict** to the values you want to predict and run the codes.

Redeployment:

To redeploy the model, repeat step 8, if the model is not trained repeat steps 1-8.

Project Cost:

Workbench Instance: $115.20 monthly, about $0.158 hourly (created Feb 14, 2024, ran up to Feb 20th, 2014 resulting in about 135 hours)

AutoML training: $21.252 per node hour (two models was trained using 1 max node hour each)

Model Prediction: $0.0926 per node hour (1 model was deployed to endpoint for predictions)

Workbench: 0.158 x 135 = $21.33
AutoML: 21.252 x 2 = $42.50
Model Prediction: 0.0926 x 1 = $0.09

Total = $63.92