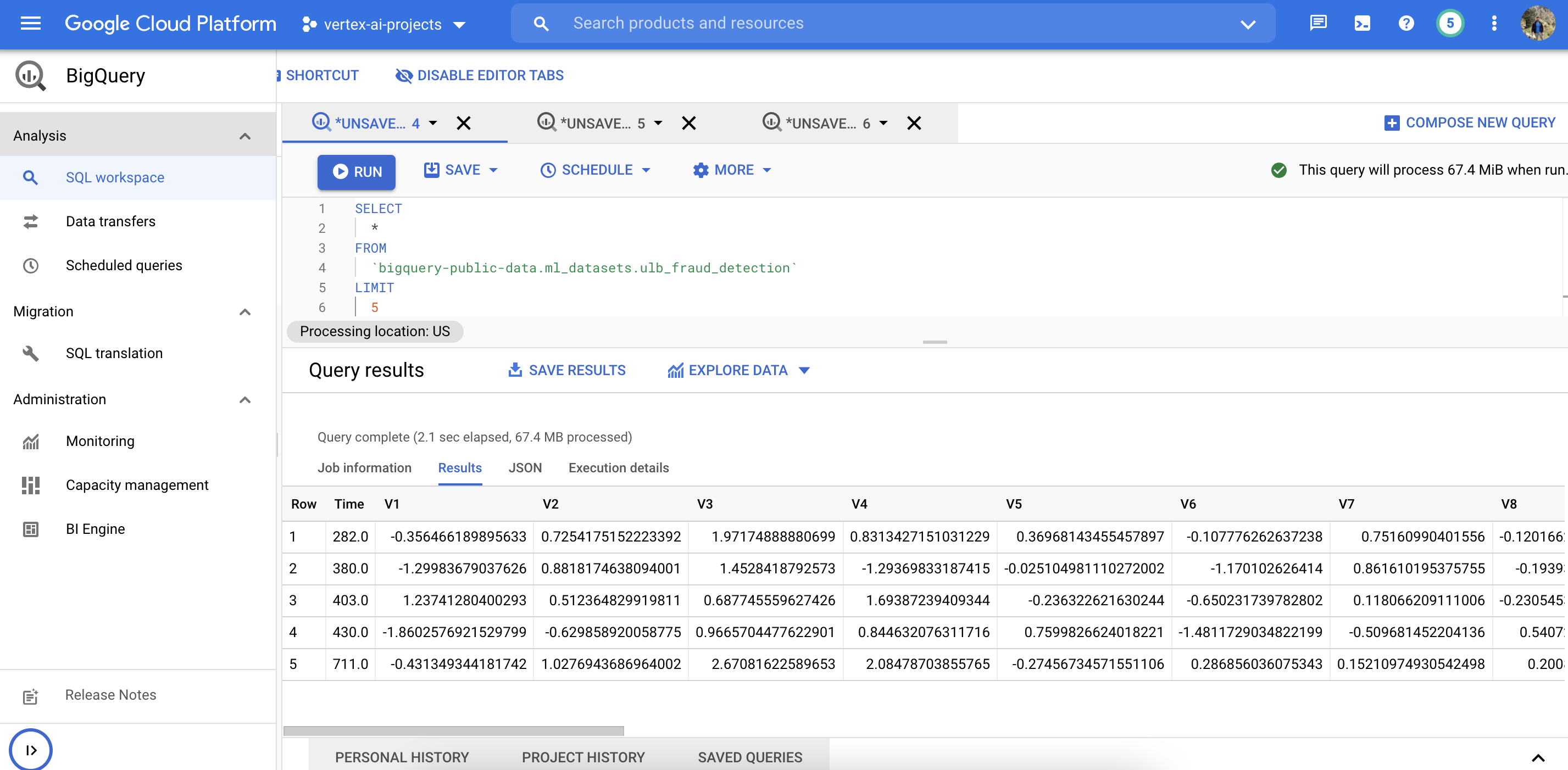
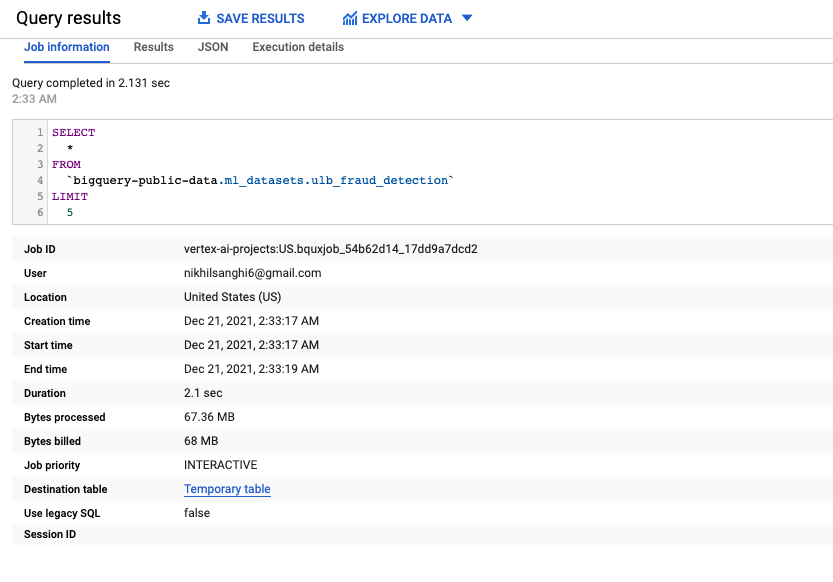
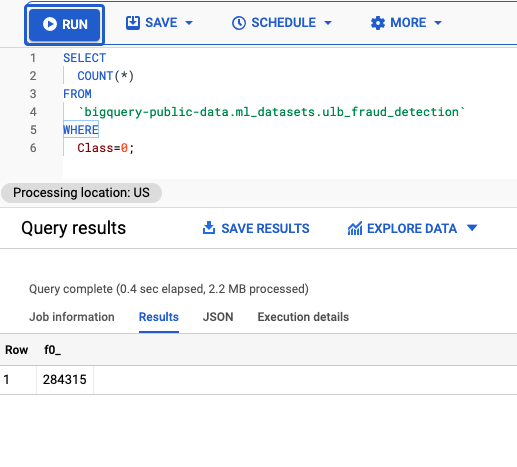
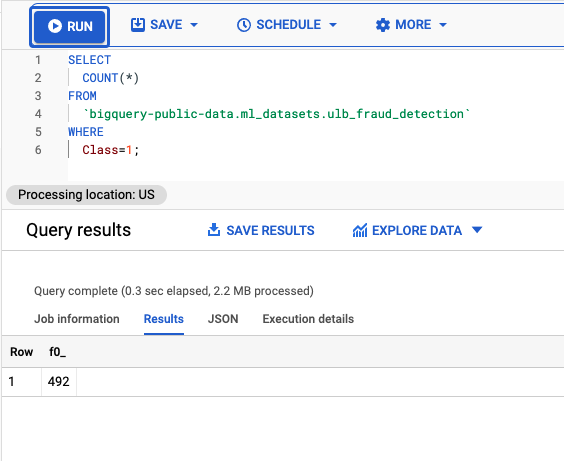
**Fraud Detection on Credit Card Transactions using BigQuery ML on GCP**

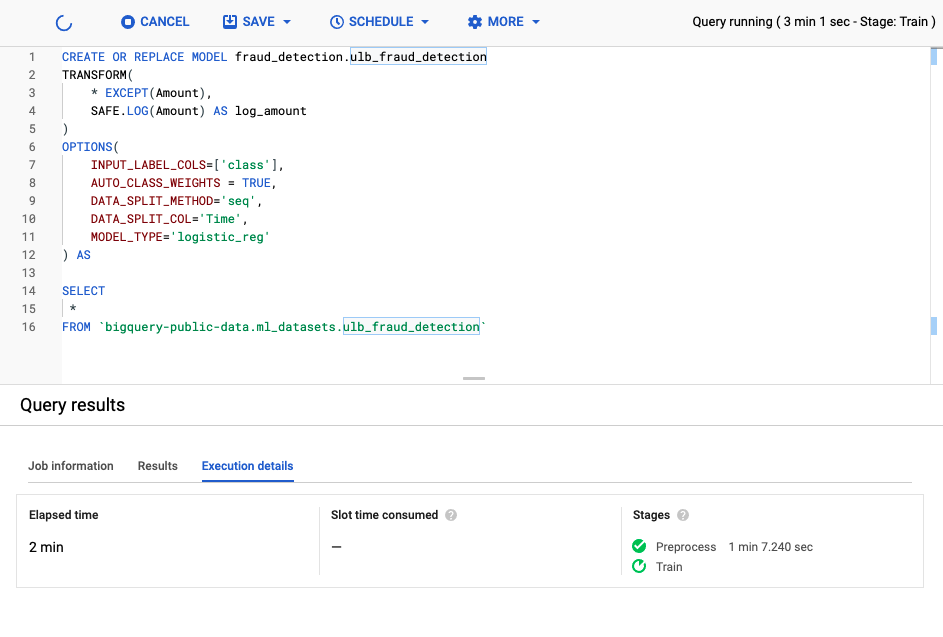
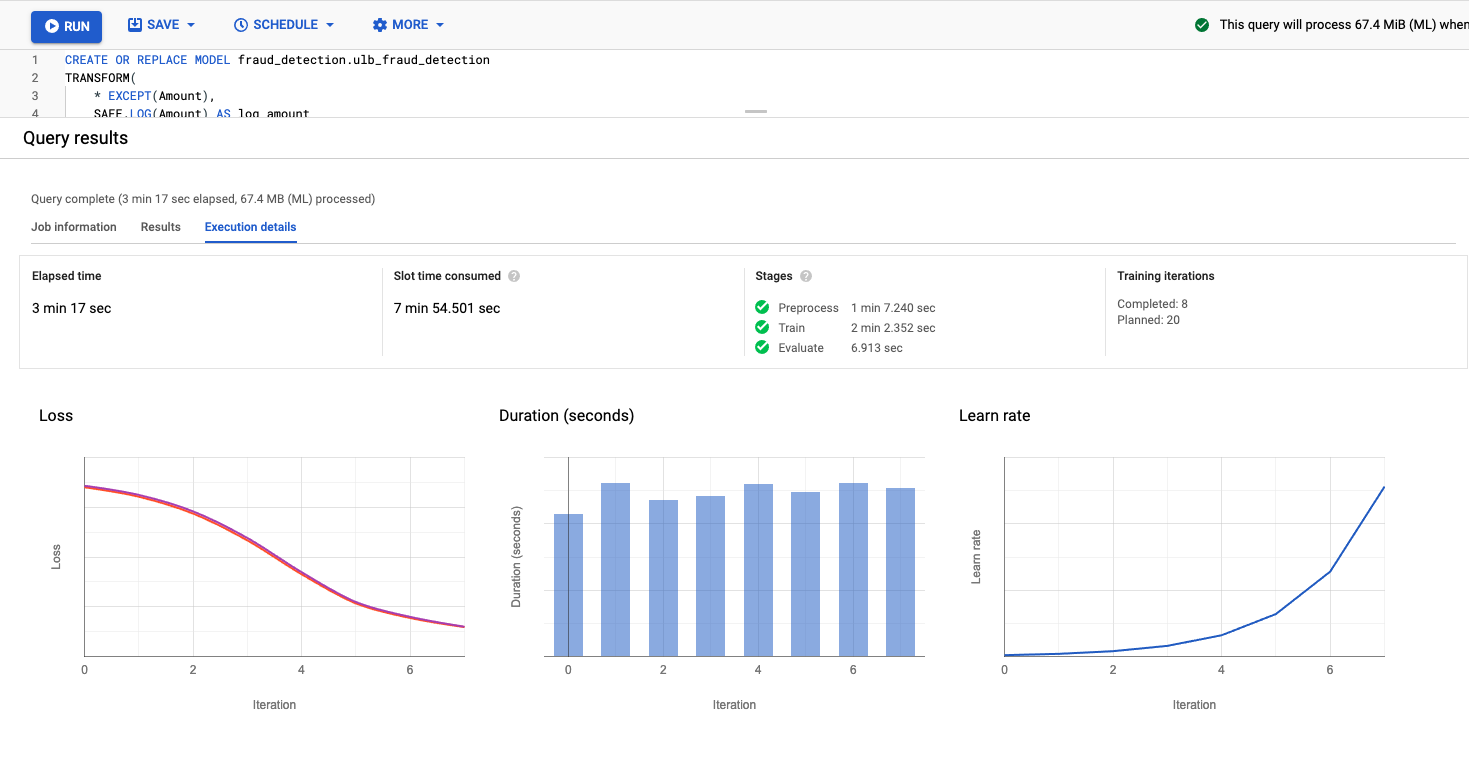
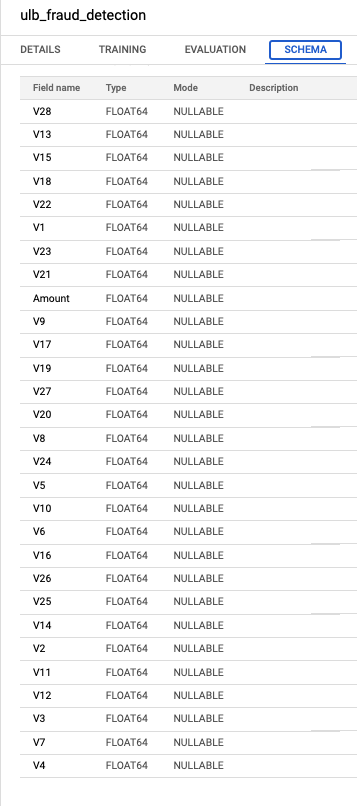
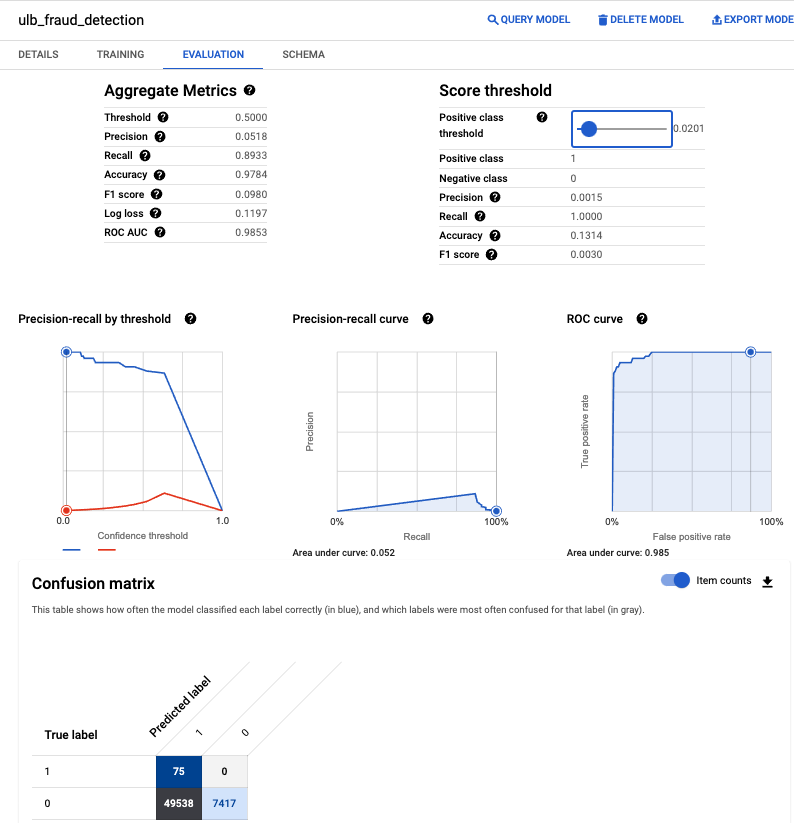
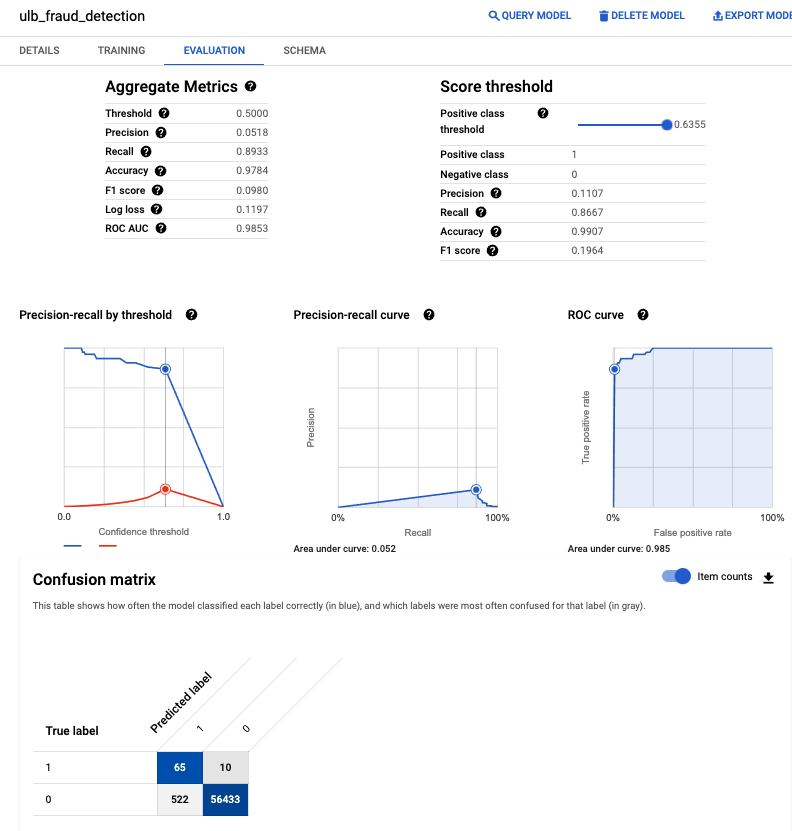
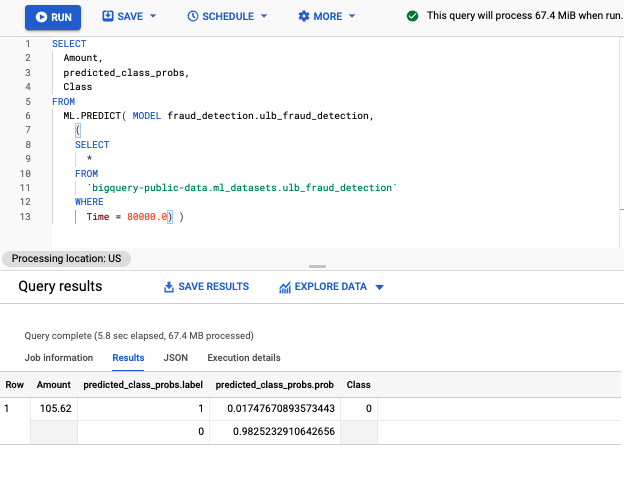
**Dataset**

Data used (also available as public dataset on bigquery) -https://github.com/jbrownlee/Datasets/blob/d20fcb6402ae34e653d4513b00f39257bb37ed7f/creditcard.csv.zip  
  
Dataset holds 28 feature which are obtained after PCA.  
Feature time shows the time elapsed between first and that specific transaction.  
Feature amount refers to the transaction amount.  
Feature class refers to the transaction being fraudulent or not (1 for fraud and 0 otherwise)

**Ingesting the data**-  
  
Query - “SELECT \* FROM `bigquery-public-data.ml\_datasets.ulb\_fraud\_detection` LIMIT 5”  
  
This query fetches the dataset stored in the GCP public repository and shows us the first 5 rows.  
  
  
  
  
  
  
We can see the job description as well   
  
  
  


**Data exploration**  
  
Let's check the data distribution-  
  
Query = “SELECT COUNT(\*) FROM `bigquery-public-data.ml\_datasets.ulb\_fraud\_detection`   
where Class=0;”  
  
This query fetches all the rows which are non-fraudulent and returns the total count.  
  
  
  
  
  
Query = “SELECT COUNT(\*) FROM `bigquery-public-data.ml\_datasets.ulb\_fraud\_detection`   
where Class=1;”  
  
This query fetches all the rows which are fraudulent and returns the total count.  
  
  
  
  
**Observation** -   
There are 492 fraudulent Transactions out of 284315, which makes this dataset highly imbalanced.  
  
Only 0.17% Fraudulent transactions.

**Model Building**  
  
Query = “  
  
CREATE OR REPLACE MODEL fraud\_detection.ulb\_fraud\_detection   
TRANSFORM(  
 \* EXCEPT(Amount),  
 SAFE.LOG(Amount) AS log\_amount  
)  
OPTIONS(  
 INPUT\_LABEL\_COLS=['class'],  
 AUTO\_CLASS\_WEIGHTS = TRUE,  
 DATA\_SPLIT\_METHOD='seq',  
 DATA\_SPLIT\_COL='Time',  
 MODEL\_TYPE='logistic\_reg'  
) AS  
  
SELECT   
 \*  
FROM `bigquery-public-data.ml\_datasets.ulb\_fraud\_detection`

”  
  
for more info - <https://cloud.google.com/bigquery-ml/docs/reference/standard-sql/bigqueryml-syntax-create>This query creates a new model or replaces an already existing model, transforms the input variables except the amount feature column, and safely logs the amount column. (Safe makes sure that a null value is returned instead of any possible error).  
   
**Options** -   
This needs the parameters we need to define for the model to be trained.  
  
**Input\_label\_cols** refer to the prediction column which is “class” in our case.  
  
**Auto\_class\_weights** refer to the weights assigned to different classes in our prediction column. It is set to True for imbalanced dataset such as ours.  
  
**Model\_type** refer to the training algorithm used which is “Logistic Regression” in our case.  
  
**Data\_split\_type** refer to the split between training and testing data.  
  
  
  
   
  
  
  
The model training takes approximately **1 min 7.24 secs** to **preprocess** the data and **2 mins 2.352** secs to **train** and finally **6.913** secs to **evaluate** the predictions.  
  
We notice that the **loss decreases** with every iteration and the **learning rate increases** with each iteration  
  
Schema of the data set -   
  
  
  
**Model Evaluation** -   
  
Observation -   
At ” 0 ” Positive threshold we observe that the only 75 prediction are right.  
0 precision with 100 % recall, which is totally undesirable.  
  
Our model have a right balance between precision and recall depending upon or business use case.  
  
Now changing the positive thresholds->  
  
  
  
  
  
Changing the Positive threshold to 2 % doesn't change much of the metrics, there are few predictions which fall in the right category.  
  
  
  
Changing the positive threshold to 20%, we notice a significant of the predictions falling in the categories.  
  
  
  
  
  
**Observation( Evaluation) -**   
  
**Further increasing the Positive Threshold value to 0.6355 we obtain the maximum precision value of 0.11 with recall 0.866 and accuracy of 0.99 with a decent usable f1 score of 0.19.**  
  
  
**There are 522 wrong predictions as fraudulent transactions and has 10 transactions which are predicted as non-fraudulent which were actually fraudulent which is more serious case, since we don't want any transaction that is fraudulent to go un-noticed, it will okay for us to flag transaction as fraudulent and then investigate (522) rather than flagging 10 cases as non-fraudulent for our banking use case.**  
  
**This threshold can change the values of model predicted positive values depending upon the business use case, in some cases we need more precision such as our case and in some we need more recall.**  
  
**We also can notice that the area under roc curve gives us the value of 0.985 which gives us good estimate of model performance.**  
  
  
Predicting class for a specific row with timestamp.  
Class column indicates the actual class of the example and the predicted\_class\_probs\_label indicates the predicted label for that transaction example. This shows that the probability for the transaction to be fraudulent is is 0.017 % while the probability for the transaction to be non-fraudulent is 0.98 %, which is correct in this case.  
  
  
**RESULT -**   
**We created a Logistic Regression model using BIgQuery ML to detect Fraudulent transactions among a highly imbalanced dataset.**