Fraud Detection on Credit Card Transactions using BigQuery ML on GCP

Dataset

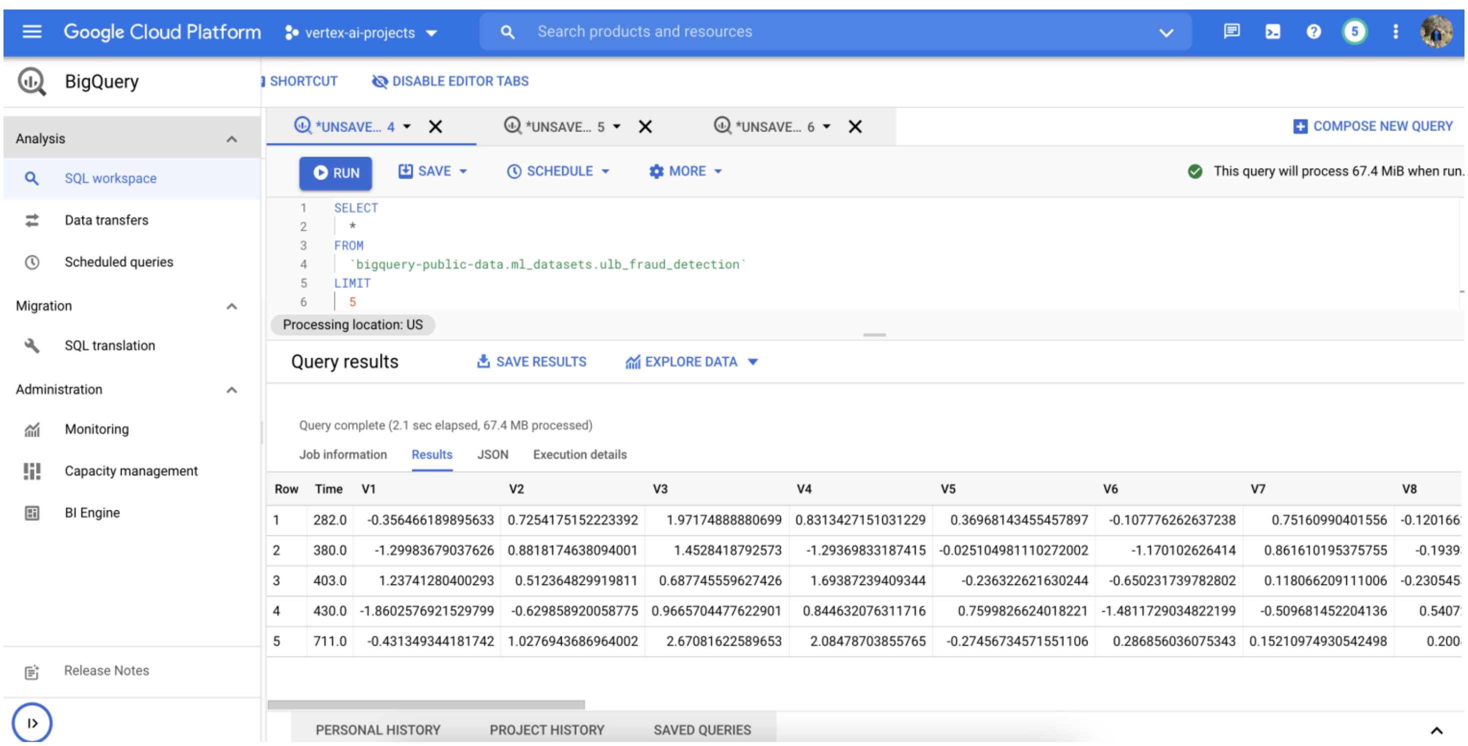
Data used (also available as public dataset on bigquery) -
https://github.com/jbrownlee/Datasets/blob/d20fcb6402ae34e653d4513b00f39257bb37ed7f/creditcard.csv.zip

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)

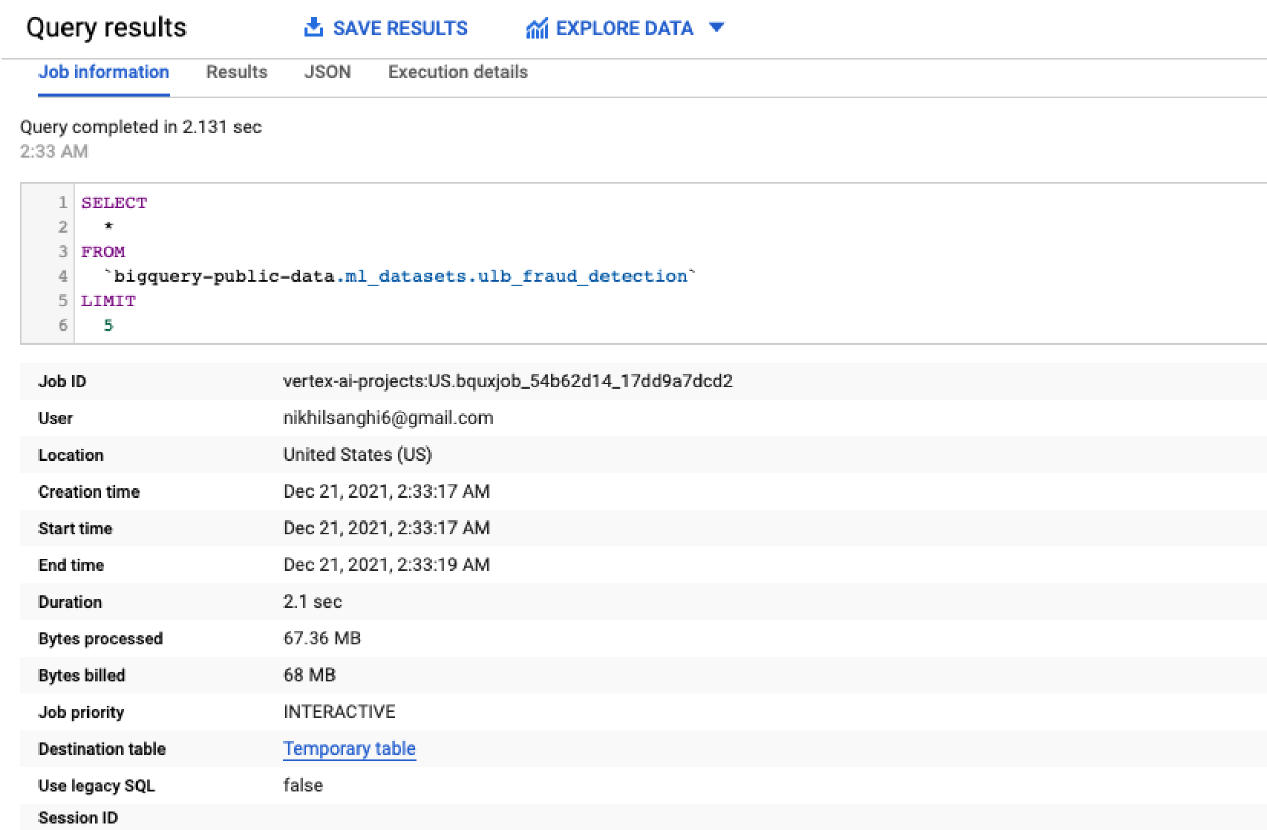
Ingesting the data-

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




```
SELECT
  *
FROM
  `bigquery-public-data.ml_datasets.ulb_fraud_detection`
LIMIT
  5
```

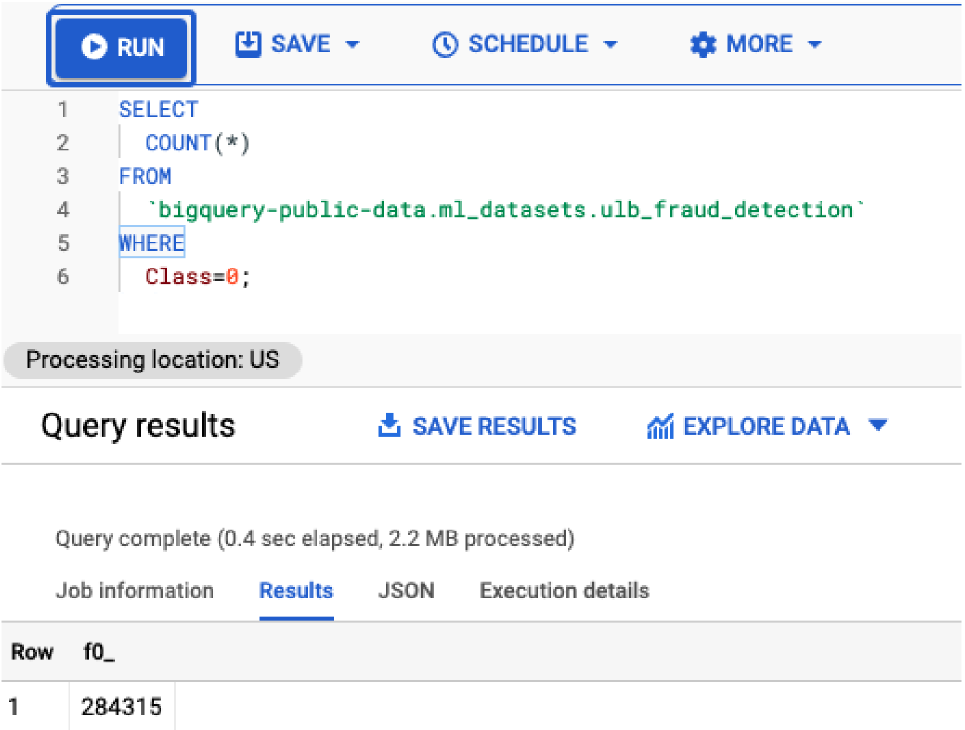
Job ID	vertex-ai-projects:US.bquxjob_54b62d14_17dd9a7dcd2
User	nikhilsanghi6@gmail.com
Location	United States (US)
Creation time	Dec 21, 2021, 2:33:17 AM
Start time	Dec 21, 2021, 2:33:17 AM
End time	Dec 21, 2021, 2:33:19 AM
Duration	2.1 sec
Bytes processed	67.36 MB
Bytes billed	68 MB
Job priority	INTERACTIVE
Destination table	Temporary table
Use legacy SQL	false
Session ID	

Data exploration

Let's check the data distribution-

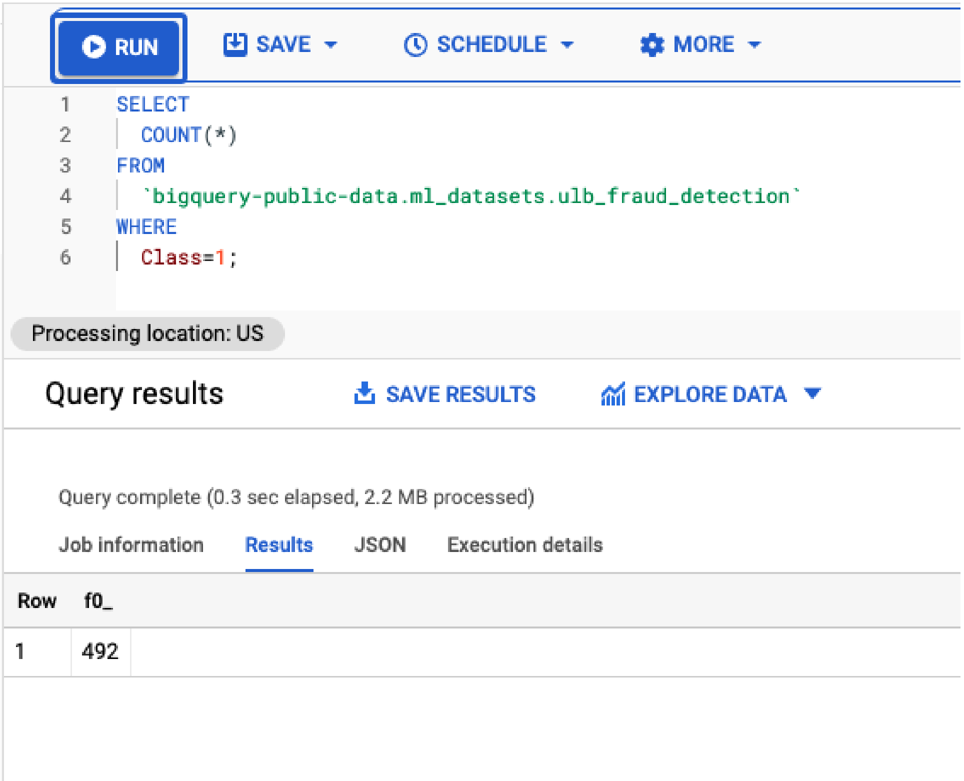
Query = “SELECT COUNT(*) FROM `bigquery-public-data.ml_datasets.ulb_fraud_detection` where Class=0;”

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).

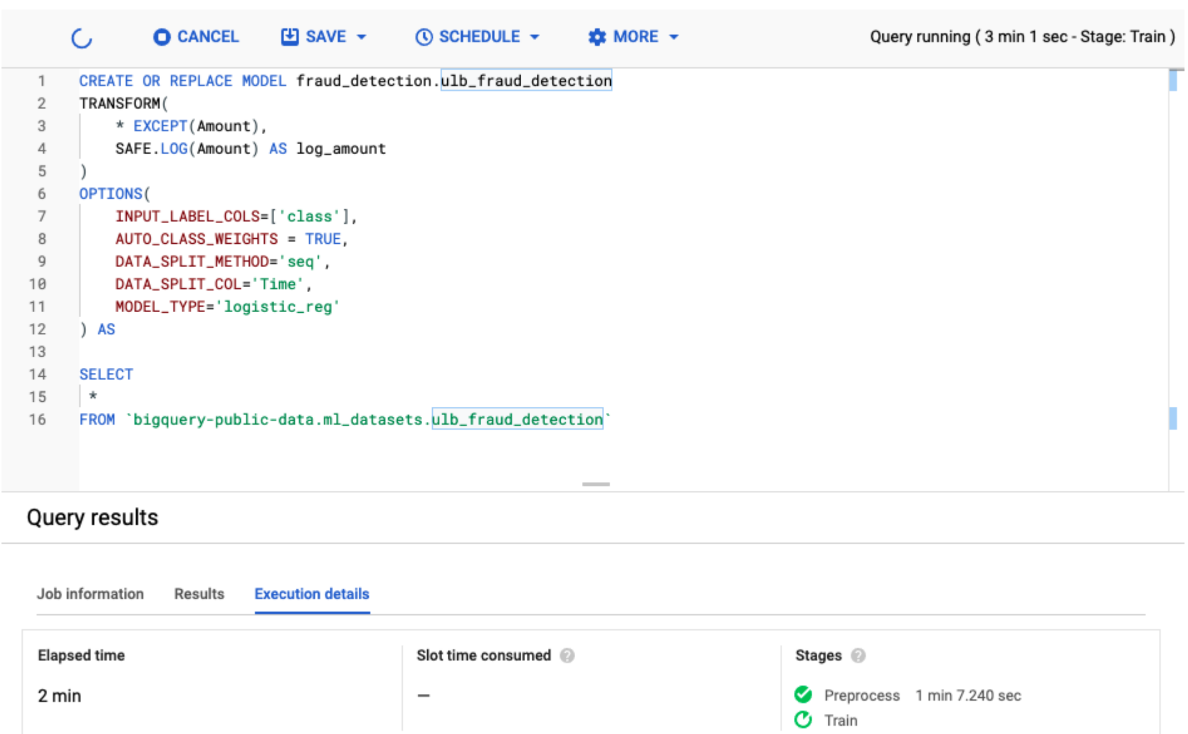
Options -
This needs the parameters we need to define for the model to be trained.

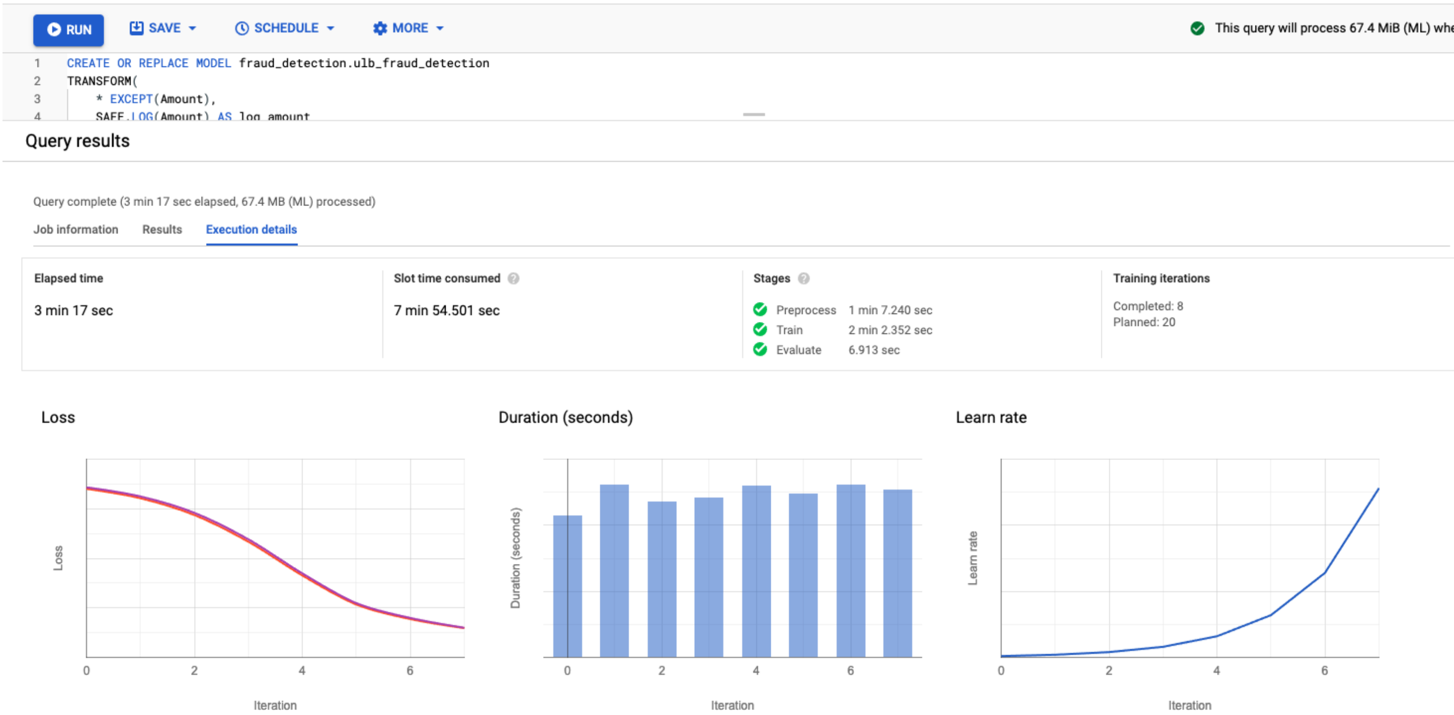
Input_label_cols refer to the prediction column which is “class” in our case.

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 -

ulb_fraud_detection

DETAILS TRAINING EVALUATION SCHEMA

Field name	Type	Mode	Description
V28	FLOAT64	NULLABLE	
V13	FLOAT64	NULLABLE	
V15	FLOAT64	NULLABLE	
V18	FLOAT64	NULLABLE	
V22	FLOAT64	NULLABLE	
V1	FLOAT64	NULLABLE	
V23	FLOAT64	NULLABLE	
V21	FLOAT64	NULLABLE	
Amount	FLOAT64	NULLABLE	
V9	FLOAT64	NULLABLE	
V17	FLOAT64	NULLABLE	
V19	FLOAT64	NULLABLE	
V27	FLOAT64	NULLABLE	
V20	FLOAT64	NULLABLE	
V8	FLOAT64	NULLABLE	
V24	FLOAT64	NULLABLE	
V5	FLOAT64	NULLABLE	
V10	FLOAT64	NULLABLE	
V6	FLOAT64	NULLABLE	
V16	FLOAT64	NULLABLE	
V26	FLOAT64	NULLABLE	
V25	FLOAT64	NULLABLE	
V14	FLOAT64	NULLABLE	
V2	FLOAT64	NULLABLE	
V11	FLOAT64	NULLABLE	
V12	FLOAT64	NULLABLE	
V3	FLOAT64	NULLABLE	
V7	FLOAT64	NULLABLE	
V4	FLOAT64	NULLABLE	

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.

ulb_fraud_detection

QUERY MODEL | DELETE MODEL | EXPORT MODE

DETAILS | TRAINING | EVALUATION | SCHEMA

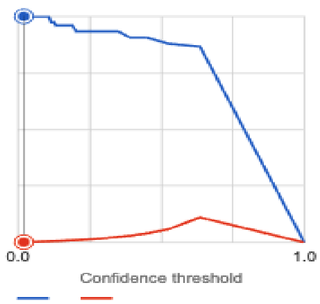
Aggregate Metrics

Threshold	0.5000
Precision	0.0518
Recall	0.8933
Accuracy	0.9784
F1 score	0.0980
Log loss	0.1197
ROC AUC	0.9853

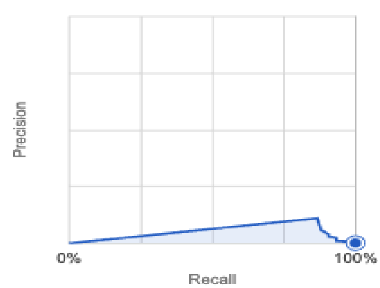
Score threshold

Positive class threshold	0.0201
Positive class	1
Negative class	0
Precision	0.0015
Recall	1.0000
Accuracy	0.1314
F1 score	0.0030

Precision-recall by threshold

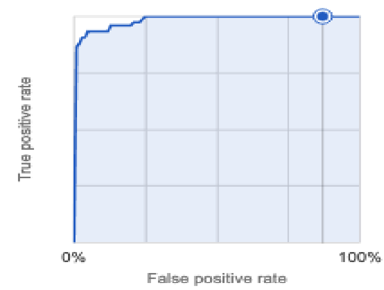


Precision-recall curve



Area under curve: 0.052

ROC curve



Area under curve: 0.985

Confusion matrix

Item counts

This table shows how often the model classified each label correctly (in blue), and which labels were most often confused for that label (in gray).

True label \ Predicted label	1	0
1	75	0
0	49538	7417

Changing the positive threshold to 20%, we notice a significant of the predictions falling in the categories.

ulb_fraud_detection

QUERY MODEL | DELETE MODEL | EXPORT MODEL

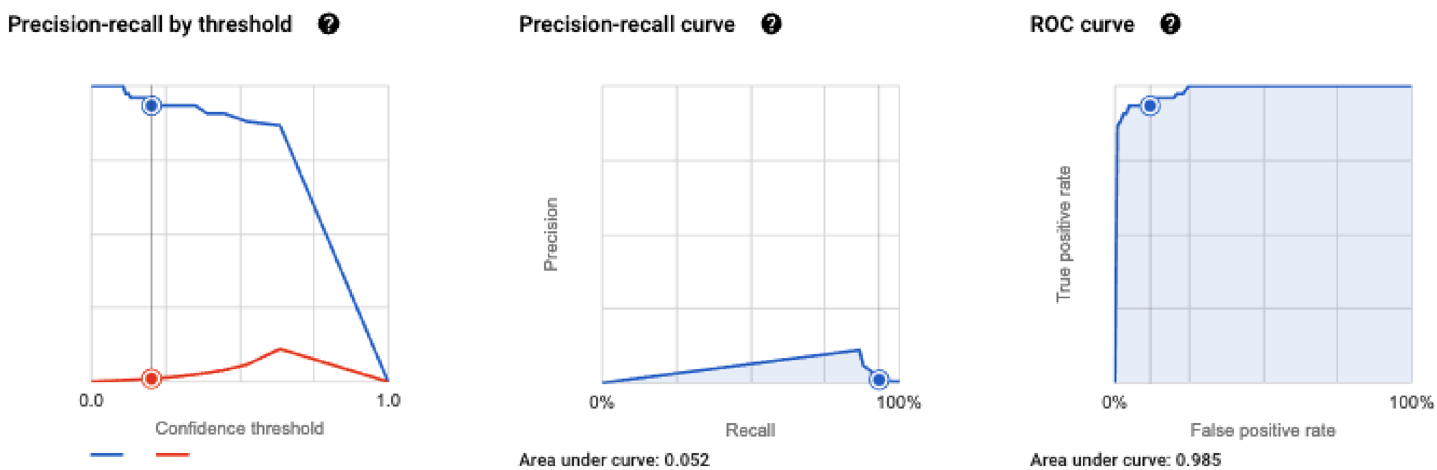
DETAILS | TRAINING | **EVALUATION** | SCHEMA

Aggregate Metrics

Threshold	0.5000
Precision	0.0518
Recall	0.8933
Accuracy	0.9784
F1 score	0.0980
Log loss	0.1197
ROC AUC	0.9853

Score threshold

Positive class threshold	0.2028
Positive class	1
Negative class	0
Precision	0.0102
Recall	0.9333
Accuracy	0.8812
F1 score	0.0202



Confusion matrix

Item counts

This table shows how often the model classified each label correctly (in blue), and which labels were most often confused for that label (in gray).

True label \ Predicted label	1	0
1	70	5
0	6770	50185

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.

ulb_fraud_detection

QUERY MODEL DELETE MODEL EXPORT MODE

DETAILS TRAINING EVALUATION SCHEMA

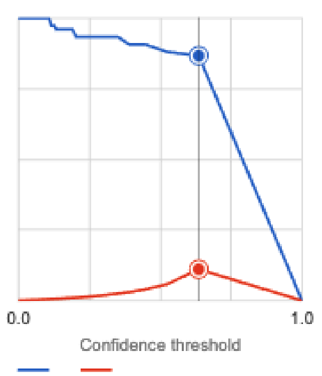
Aggregate Metrics

Threshold	0.5000
Precision	0.0518
Recall	0.8933
Accuracy	0.9784
F1 score	0.0980
Log loss	0.1197
ROC AUC	0.9853

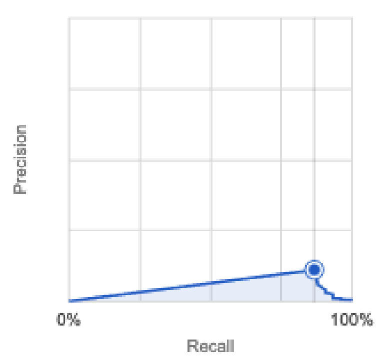
Score threshold

Positive class threshold	0.6355
Positive class	1
Negative class	0
Precision	0.1107
Recall	0.8667
Accuracy	0.9907
F1 score	0.1964

Precision-recall by threshold

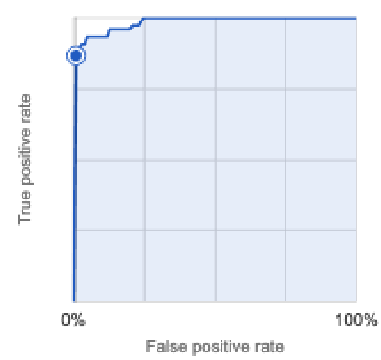


Precision-recall curve



Area under curve: 0.052

ROC curve



Area under curve: 0.985

Confusion matrix

Item counts

This table shows how often the model classified each label correctly (in blue), and which labels were most often confused for that label (in gray).

True label \ Predicted label	1	0
1	65	10
0	522	56433

RUN SAVE SCHEDULE MORE This query will process 67.4 MiB when run.

```
SELECT
  Amount,
  predicted_class_probs,
  Class
FROM
  ML.PREDICT( MODEL fraud_detection.ulb_fraud_detection,
    (
    SELECT
      *
    FROM
      `bigquery-public-data.ml_datasets.ulb_fraud_detection`
    WHERE
      Time = 80000.0) )
```

Processing location: US

Query results SAVE RESULTS EXPLORE DATA

Query complete (5.8 sec elapsed, 67.4 MB processed)

Job information Results JSON Execution details

Row	Amount	predicted_class_probs.label	predicted_class_probs.prob	Class
1	105.62	1	0.01747670893573443	0
		0	0.9825232910642656	

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.