DATA SCIENCE PROJECT

FINAL REPORT ON CREDIT CARD FRAUDS

CREDIT CARD FRAUDS:

PREDICTING THAT WHETHER A TRANSACTION IS FRAUD OR NOT

DATA SOURCES

ALL THE DATA HAS BEEN COLLECTED FROM "KAGGLE.COM"

ETL: EXTRACT, TRANSFORM AND LOAD

01

After uploading the dataset in the object storage, I used IBM pandas DataFrame to read the file in ETL format so that it is globally available to in the data storage.

02

This data was having zero missing values. That means I have a not found a single missing value in the dataset

03

I used some basic operations to have some understanding of the data.

	Time	V1	V2	V3	V4	V5	V6	V7	V8	V9	...	V21	V22	V23	V
0	0.0	-1.359807	-0.072781	2.536347	1.378155	-0.338321	0.462388	0.239599	0.098698	0.363787	...	-0.018307	0.277838	-0.110474	0.0669
1	0.0	1.191857	0.266151	0.166480	0.448154	0.060018	-0.082361	-0.078803	0.085102	-0.255425	...	-0.225775	-0.638672	0.101288	-0.339
2	1.0	-1.358354	-1.340163	1.773209	0.379780	-0.503198	1.800499	0.791461	0.247676	-1.514654	...	0.247998	0.771679	0.909412	-0.689
3	1.0	-0.966272	-0.185226	1.792993	-0.863291	-0.010309	1.247203	0.237609	0.377436	-1.387024	...	-0.108300	0.005274	-0.190321	-1.175
4	2.0	-1.158233	0.877737	1.548718	0.403034	-0.407193	0.095921	0.592941	-0.270533	0.817739	...	-0.009431	0.798278	-0.137458	0.141

VIEW OF DATASET

Dataset is in V1, V2,etc. format because to protect the users crucial information. Here in the dataset, "Class" column explains about weather a transaction is fraud or not.

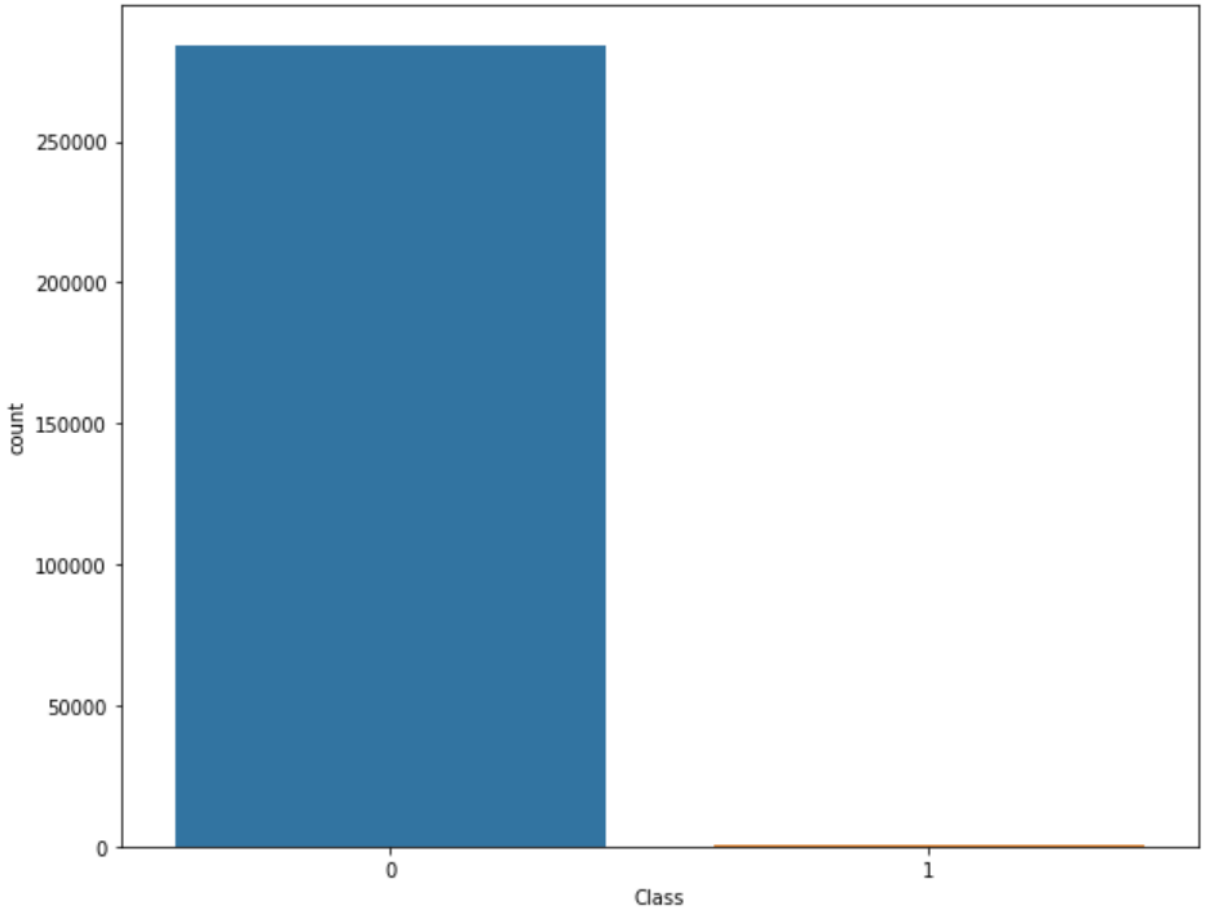
EXPLORATORY DATA ANALYSIS

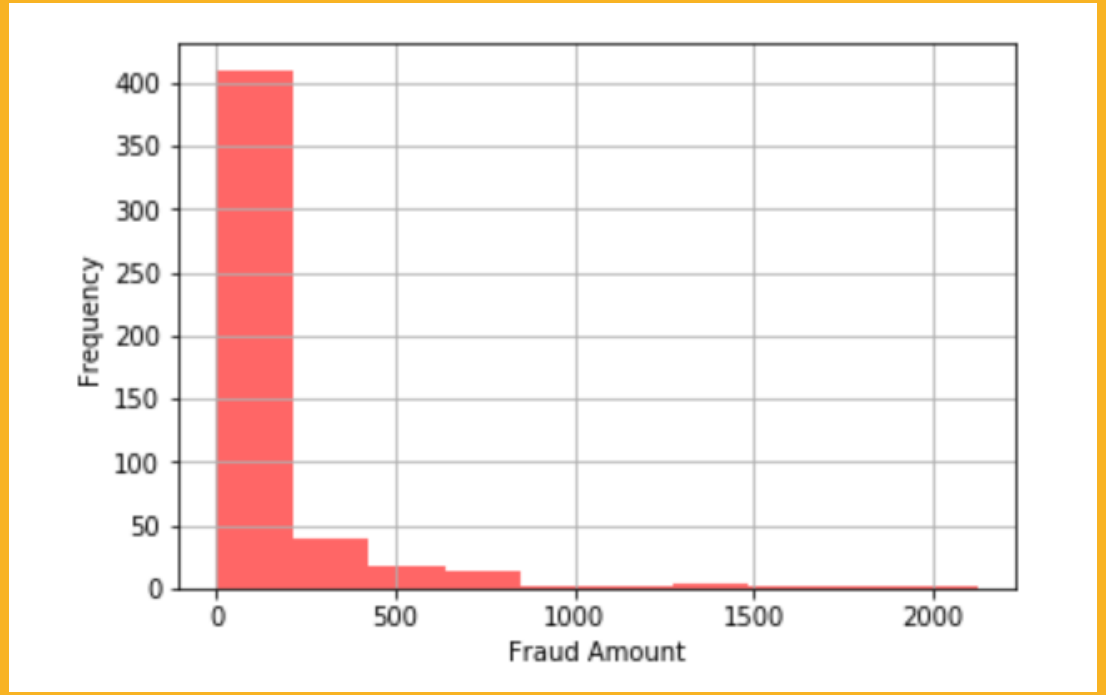
- Here I found that the data is imbalanced. Here is the result for value of fraud transactions and genuine transactions
- This data set is taken to stop the charges to users who don't make any purchase.

```
: 0    284315
  1       492
  Name: Class, dtype: int64
```

	Time	V1	V2	V3
count	284807.000000	2.848070e+05	2.848070e+05	2.848070e+05
mean	94813.859575	3.919560e-15	5.688174e-16	-8.769071e-15
std	47488.145955	1.958696e+00	1.651309e+00	1.516255e+00
min	0.000000	-5.640751e+01	-7.271573e+01	-4.832559e+01
25%	54201.500000	-9.203734e-01	-5.985499e-01	-8.903648e-01
50%	84692.000000	1.810880e-02	6.548556e-02	1.798463e-01
75%	139320.500000	1.315642e+00	8.037239e-01	1.027196e+00
max	172792.000000	2.454930e+00	2.205773e+01	9.382558e+00

PLOT TO SHOW THE DATA IMBALANCE





MAXIMUM AMOUNT OF FRAUD AMOUNT

HERE I DON'T UNDERSTAND WHY ARE THEY TREATING AMOUNT ZERO AS A FRAUD TRANSACTION.

LINEAR REGRESSION

- Linear regression was performed to check the relationship between different variables and transaction is fraud or genuine.

FEATURE ENGINEERING

- Here I used f1 score to test the performance accuracy
- Logistic Regression to check some to check the relationships between dependant variables and other non dependant variables.
- Recall Score is used to check the true positives and false negatives.
-

- Our F1 score and Recall is score is low, so again using them but with some different variables.
- These are the earlier and after random forest

Earlier Score

```
#Predict on test dataset
Lr_pred = Lr.predict(x_t
#check the accuracy
accuracy_score(Lr_pred,y
```

```
0.9990051847430451
```

```
from sklearn.metrics imp
f1_score(y_test,Lr_pred)
```

```
0.6530612244897959
```

```
recall_score(y_test,Lr_p
```

```
0.5925925925925926
```

After Score

```
random_forest_pred = r
accuracy_score(random_
```

```
/opt/conda/envs/Pythor
mators will change frc
  "10 in version 0.20
```

```
2]: 0.9994382219725431
```

```
3]: f1_score(random_forest
```

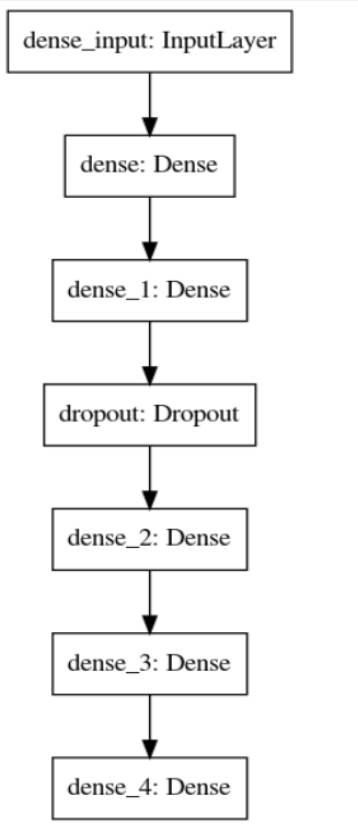
```
3]: 0.8032786885245902
```

```
4]: recall_score(random_fc
```

```
4]: 0.8990825688073395
```

APPLYING LAYERS ON SEQUENTIAL MODEL

Layers used by diagram



Layers with the parameters

Layer (type)	Output Shape	Param #
dense (Dense)	(None, 16)	496
dense_1 (Dense)	(None, 24)	408
dropout (Dropout)	(None, 24)	0
dense_2 (Dense)	(None, 20)	500
dense_3 (Dense)	(None, 24)	504
dense_4 (Dense)	(None, 1)	25

Total params: 1,933
Trainable params: 1,933
Non-trainable params: 0

MODEL TESTING RESULT

```
model.compile(optimizer="adam",loss="binary_crossentropy",metrics=['accuracy'])
model.fit(x_train,y_train,batch_size=15,epochs=3)
```

```
WARNING:tensorflow:From /opt/conda/envs/Python36/lib/python3.6/site-packages/tensorflow/python/
(from tensorflow.python.ops.math_ops) is deprecated and will be removed in a future version.
Instructions for updating:
Use tf.cast instead.
Epoch 1/3
199364/199364 [==============================] - 434s 2ms/sample - loss: 0.4326 - acc: 0.9728
Epoch 2/3
199364/199364 [==============================] - 405s 2ms/sample - loss: 0.0288 - acc: 0.9982
Epoch 3/3
199364/199364 [==============================] - 455s 2ms/sample - loss: 0.0288 - acc: 0.9982
```

```
score = model.evaluate(x_test,y_test)
```

```
85443/85443 [==============================] - 30s 352us/sample - loss: 0.0255 - acc: 0.9984
```

CONCLUSION

- Though the data is imbalanced but still it is giving good accuracy.
- Accuracy metric is not best metric to use when evaluating imbalanced class as it can be mislead for the classification.
- Following are some metrics give us the good insights on the imbalanced dataset
- 1. **Confusion Metrics :** confusion metrics show the clearly classification of the predicted class vs actual class. We can also see how many data point wrongly classified.
- 2. **Precision :** We can get the precision by number of all positive classified value divided by all positive predicted value. It's measure the classifier's exactness. Low precision indicates the high number of false positive.
- 3. **Recall :** Recall is a metric that quantifies the number of correct positive predictions made out of all positive predictions that could have been made. Unlike precision that only comments on the correct positive predictions out of all positive predictions, recall provides an indication of missed positive predictions.
- 4. **F1-score :** The F1-measure, which weights precision and recall equally, is the variant most often used when learning from imbalanced data.
- 5. **Classification Report :** All above mentioned things are auto-generated in the classification report

CONCLUSION

- Majority of the transactions are genuine according to dataset
- With the use of random forest classifier our results and accuracy for different measures improved
- Now our model is ready to predict that whether any transaction is genuine or fraud.