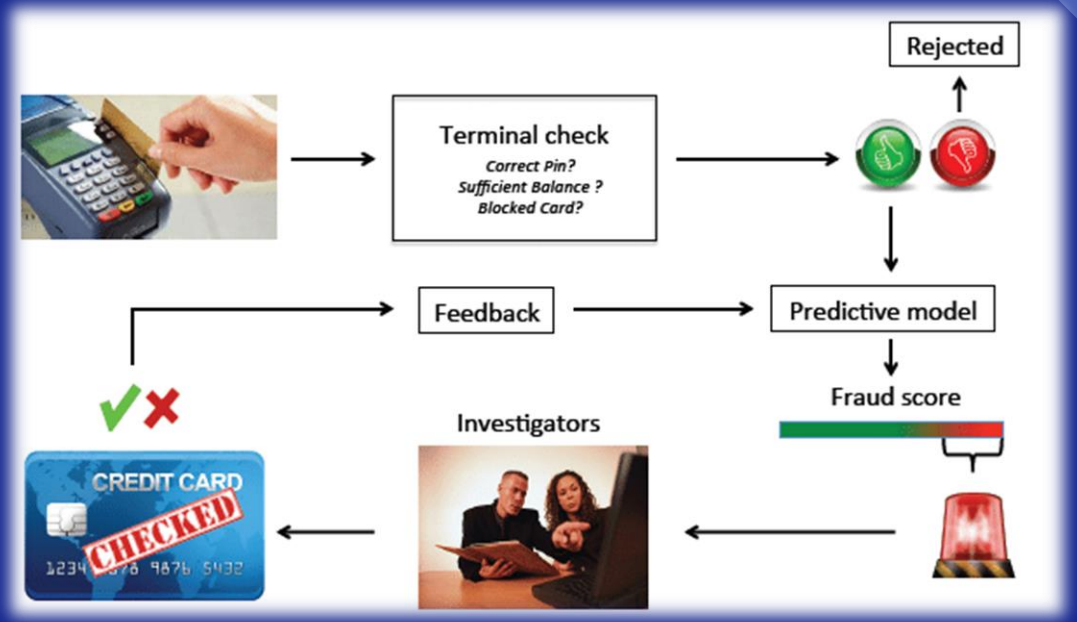
Credit Card Transaction Fraud Detection

Mentored By: Ganesh Sir

Team Members:

Sanu Satyam
Gaurav Srivastava
KrishnaKumar G
Mohit Gupta
Mohammad Yaser
Panchmurthi Bhargavi
Shefali Pandey
Vamsi Kopparthi



Problem Statement

To predict the Credit Card Transaction Fraud using the past data and behavior of the users.

- In the Credit Card Companies, illegitimate credit card usage is a serious problem which results in a need to accurately detect fraudulent transactions vs non-fraudulent transactions.
- Credit card fraud detection is the process of identifying purchase attempts that are fraudulent and rejecting them rather than processing the order
- Here, we are trying to solve the above problem using machine learning. We overcome the problem by creating a binary classifier and experimenting with various ML models to see which predicts better.

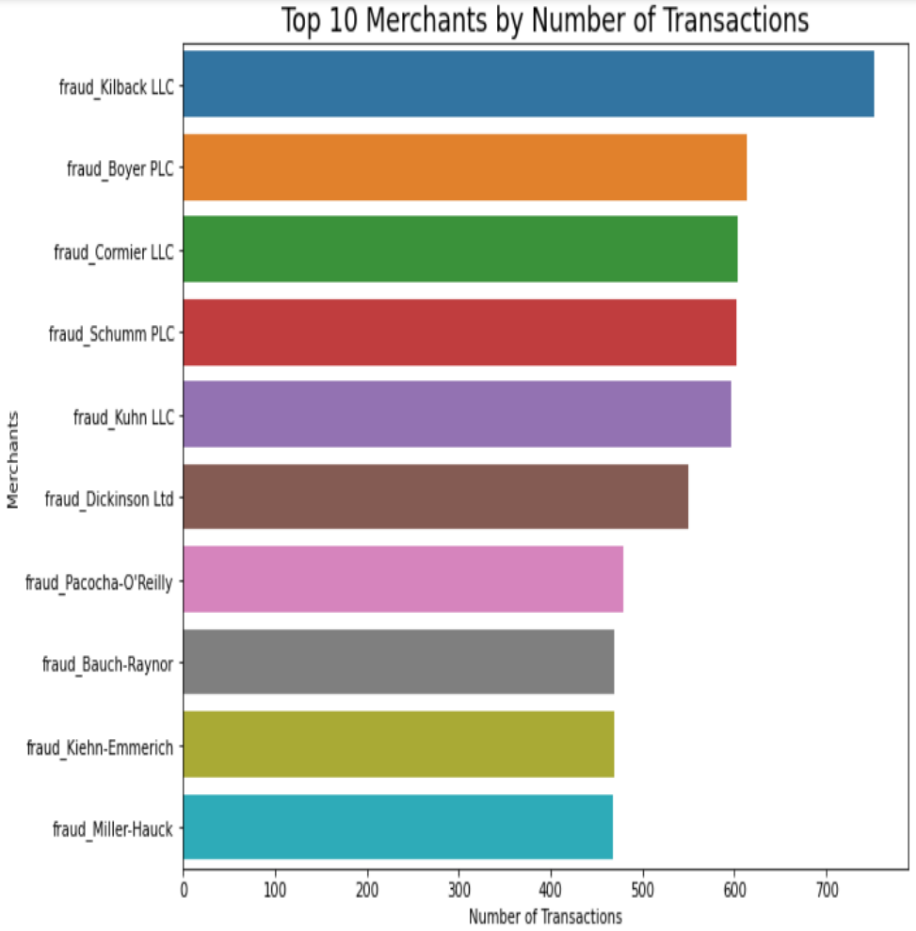
About Dataset & EDA

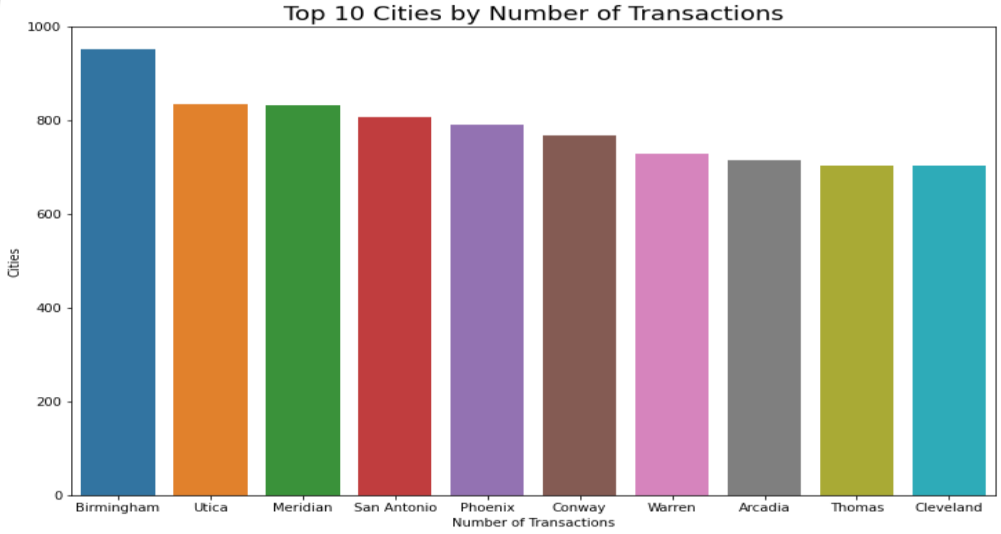
- The Dataset has been collected from Kaggle having more than 2 Lac Rows. And the Data is Unbalanced.
- The Dataset has 209651 rows and 20 columns. It has no duplicates and null values.
- **Exploratory Data Analysis**, is the act of analyzing a dataset to understand the main statistical characteristics with visual and statistical methods.

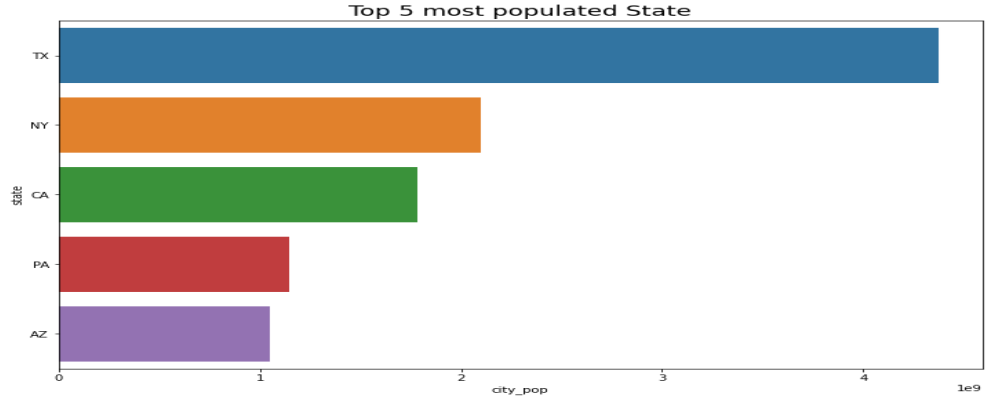
```
data.describe()
```

	cc_num	amt	lat	long	city_pop	unix_time	merch_lat	merch_long	is_fraud
count	2.096510e+05	209651.000000	209651.000000	209651.000000	2.096510e+05	2.096510e+05	209651.000000	209651.000000	209651.000000
mean	4.146945e+17	89.583376	38.549583	-90.236515	8.904593e+04	1.358516e+09	38.548503	-90.237149	0.046034
std	1.305438e+18	212.000962	5.072616	13.761965	3.011289e+05	1.817933e+07	5.107879	13.771945	0.209558
min	6.041621e+10	1.000000	20.027100	-165.672300	2.300000e+01	1.325378e+09	19.027804	-166.669638	0.000000
25%	1.800360e+14	9.930000	34.668900	-96.798000	7.430000e+02	1.342961e+09	34.756059	-96.904540	0.000000
50%	3.519610e+15	49.180000	39.371600	-87.476900	2.456000e+03	1.356951e+09	39.375763	-87.459875	0.000000
75%	4.635330e+15	87.930000	41.948800	-80.175200	2.032800e+04	1.374411e+09	41.966234	-80.261122	0.000000
max	4.992350e+18	27119.770000	66.693300	-67.950300	2.906700e+06	1.388534e+09	67.510267	-66.956540	1.000000

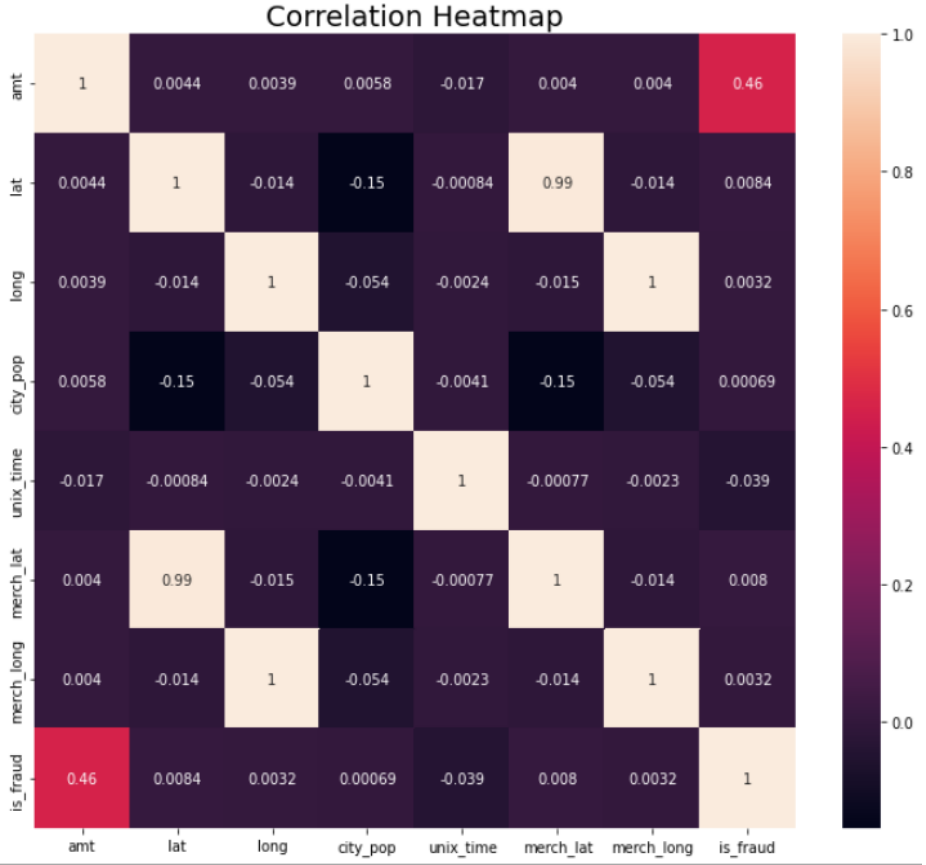
Visualization

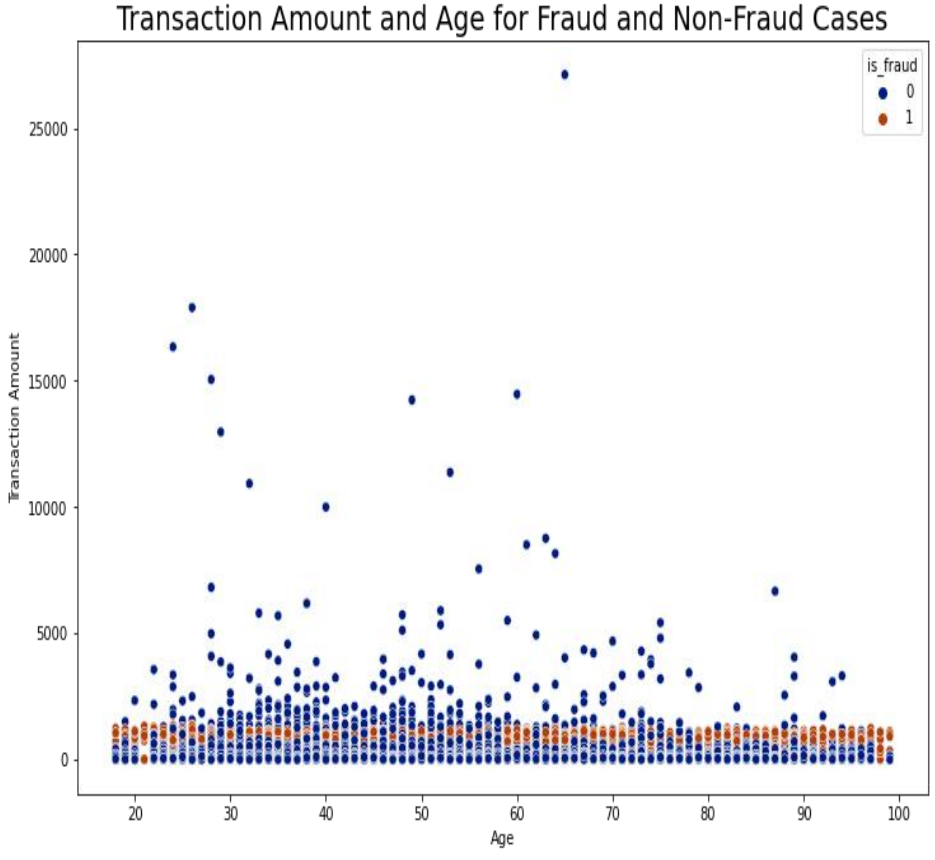






Visualization





Label Encoding & Handling Class Imbalance

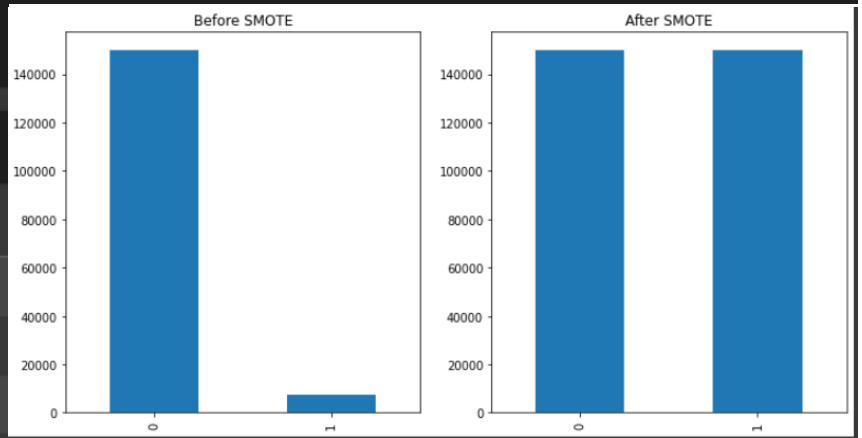
- Label Encoding refers to converting the labels into a numeric form so as to convert them into the machine-readable form.
- Imbalanced data set will lead algorithms to get good results by returning the majority. That will be a problem if you are interested in the minority more.

```
from sklearn.preprocessing import LabelEncoder
le=LabelEncoder()
for i in data.columns:
        if data[i].dtypes==("object"):
            label=le.fit_transform(data[i])
            data[i]=label
```

```
data.head()
```

	category	gender	amt	lat	long	city_pop	merch_lat	merch_long	age	is_fraud
0	4	0	212.75	37.9931	-100.9893	2691	38.862183	-101.234087	30	0
1	4	0	83.07	48.3400	-122.3456	85	48.682111	-122.719904	39	0
2	3	0	16.16	39.8936	-79.7856	328	39.222743	-78.839099	40	0

```
from imblearn.over_sampling import SMOTE
sm = SMOTE()
X_train_new, y_train_new = sm.fit_resample(X_train, y_train.ravel())
```



ML Model

Credit card fraud detection using Random Forest Classifier, shows a powerful technique, once the models receive huge quantities of new data every day. Although we have reached good results in the model.

Benefits:

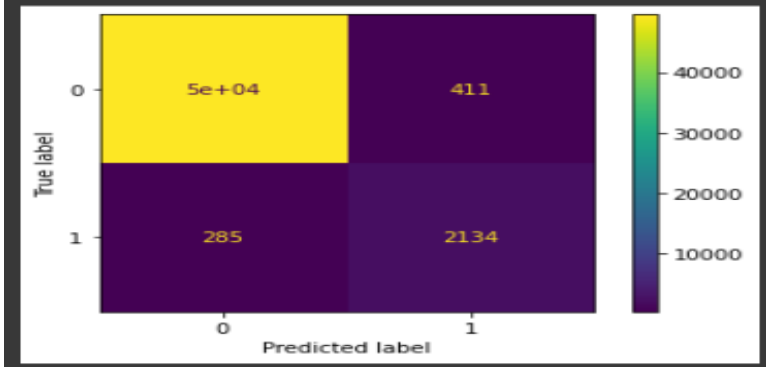
- Reduction in number of Fraud Detection
- User can safely use his credit/debit card for online transaction.
- Added layer of security

```
The Accuracy Score is:  0.986720851697098
The Classification Report is:
              precision    recall  f1-score   support

           0       0.99      0.99      0.99     49994
           1       0.84      0.88      0.86      2419

    accuracy                           0.99     52413
   macro avg       0.92      0.94      0.93     52413
weighted avg       0.99      0.99      0.99     52413

The Confusion Matrix is:
```




```
#Saving the model in pickle file

pickle.dump(RF, open('RF_model', 'wb'))
```

```
#Opening and Testing the Pickle Model
pickled_model = pickle.load(open('/content/RF_model', 'rb'))
pickled_model.predict(X_test[:10])
```

```
array([0, 0, 0, 0, 0, 0, 1, 0, 0, 0])
```

THANK YOU