**CREDIT CARD FRAUD DETECTION**

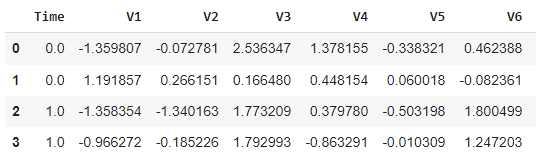
Shyam Modi

**Understanding the Problem**

* For financial banks around the world, identifying credit card fraud is crucial. Banks have to spot potential fraud so that their costumers don't pay for goods they haven't purchased. This is important so that the customer doesn't lose money and the bank does not lose its customers.
* The credit limit in credit cards helps in making purchases even when we don't have the amount at that time. But these features can be misused by cyber attackers to commit fraudulent transactions. Hence, the ultimate goal is to tackle this situation by building generalized classification models which tracks the pattern of all the transactions to classify and distinguish fraud transactions from the normal ones.
* This is a binary classification problem.

**About the Dataset**

* The dataset consists of 31 feature columns and 284807 rows. Due to security issues, 28 of the features are the result of the PCA transformation. 'Time' and 'Amount are the only columns that were not modified with PCA. The output variable is named 'Class'.
* There are no null/missing values in this dataset. All feature columns are float64 or int64 values. The output variable is heavily imbalanced.



**Visualizing distributions (with time)**

* Understanding data with respect to time and frequency. However, the dataset is imbalanced. Only 0.17% of transactions are fraudulent. Distribution of fraud and normal transactions.

**Chart, histogram

Description automatically generated**

**Chart, histogram

Description automatically generated**

**Train-Test Split**Dividing data and sampling training/testing set to balance the output variable.

**Graphical user interface, text, application, email

Description automatically generated**

**Algorithms Used For Classification**

1. Logistic Regression
2. Decision Tree
3. Random Forest
4. Ensembling techniques – Stacking, Boosting
5. Naïve Bayes
6. KNN

**Top performing models**

1. **AdaBoost**

**Table

Description automatically generated**

1. **Gradient Boosting**

**Table

Description automatically generated**

1. **Random Forest**

****

**Table

Description automatically generated**

**Chart, histogram

Description automatically generated**

1. **Stacking (with logistic, random forest and decision tree)**

**Text

Description automatically generated**

**Table

Description automatically generated**

**Summary**

* Accuracy of **ADABoost** model is: 0.9320345102452291
* Recall value : 0.9035612134852534
* F1 score: 0.93
* Accuracy of **Gradient Boosting** model is: 0.93
* Recall value: 0.86
* F1 score: 0.92
* Stacking and Random forest algorithms also performed well after some hyper-parameter tuning.

**Why KNN would perform poorly in this dataset**

* KNN does not work well with large dataset: In large datasets, the cost of calculating the distance between the new point and each existing point is huge which degrades the performance of the algorithm. Does not work well with high dimensions or noise in data.
* Focus was on getting optimal Recall score (reducing False-negatives) as machine predicting not fraud, but actual value being fraud would be a major concern for the company and would lead to a loss of customers. It also makes sense to compare F1 scores to get the model with the best score on jointly precision and recall.

**Conclusion**

* In this machine learning project, a binary classifier was implemented using various classification algorithms to predict potential fraud transactions. Through this project, several techniques were applied to address feature selection, check correlation and treat major class imbalance problem.
* 28 features columns out of 31 are the result of the PCA transformation. The output variable is named 'Class'. Visualizing data distribution with respect to time and frequency was necessary. However, the dataset was heavily imbalanced. Only 0.17% of all transactions were fraudulent**.**

**Results**

* Looking at model results, the best accuracy on the test set was achieved by the boosting technique algorithms followed by Logistic Regression. Boosting is an ensemble meta-algorithm that converts weak learners to strong ones. Important features were showcased while using Decision tree and Random Forest.