EED 363
Applied Machine Learning
Endsem Project Report

Credit Card Fraud Detection

Group No. 26

Member 1 : S Bhanoday
Roll.No- 1810110194

Member 2: Bhavith Dulipalla
Roll.No- 1810110053

Instructor: Dr.M Gopal

Content

Abstract

- Due to cashless transactions every person uses an ATM card and Credit card for transactions. So, Frauds can also be increased.

- Billions of dollars of loss are caused every year by fraudulent credit card transactions. The design of efficient fraud detection algorithms is key for reducing these losses, and more and more algorithms rely on advanced machine learning techniques to assist fraud investigators.

Introduction

- Online shopping is growing day to day. Credit cards are used for purchasing goods and services with the help of virtual card and physical card whereas virtual card for online transactions and physical card for offline transactions.

- In a physical card purchase, the card holder presents his card physically to a merchant for making a payment. If the card holder does not realise loss of card, it can lead to a substantial financial loss to the credit card company. In online payment mode, attackers need only a little information for doing fraudulent transactions (secure code, card number, expiration date etc)

Fraud detection

- Fraud detection is a binary classification problem in which the transaction data is analyzed and classified in “legitimate” or “fraudulent”.
- In this method the legitimate transactions are separated from fraudulent. The idea of this approach is that fraudsters behave differently than the account owner. Thus, models trained on normal transaction data should identify the anomaly.

Dataset

- We took a dataset from an European cardholder from September 2013. This dataset contains the transaction that occurred in two days with 284,807 transactions of which 492 are considered fraud.

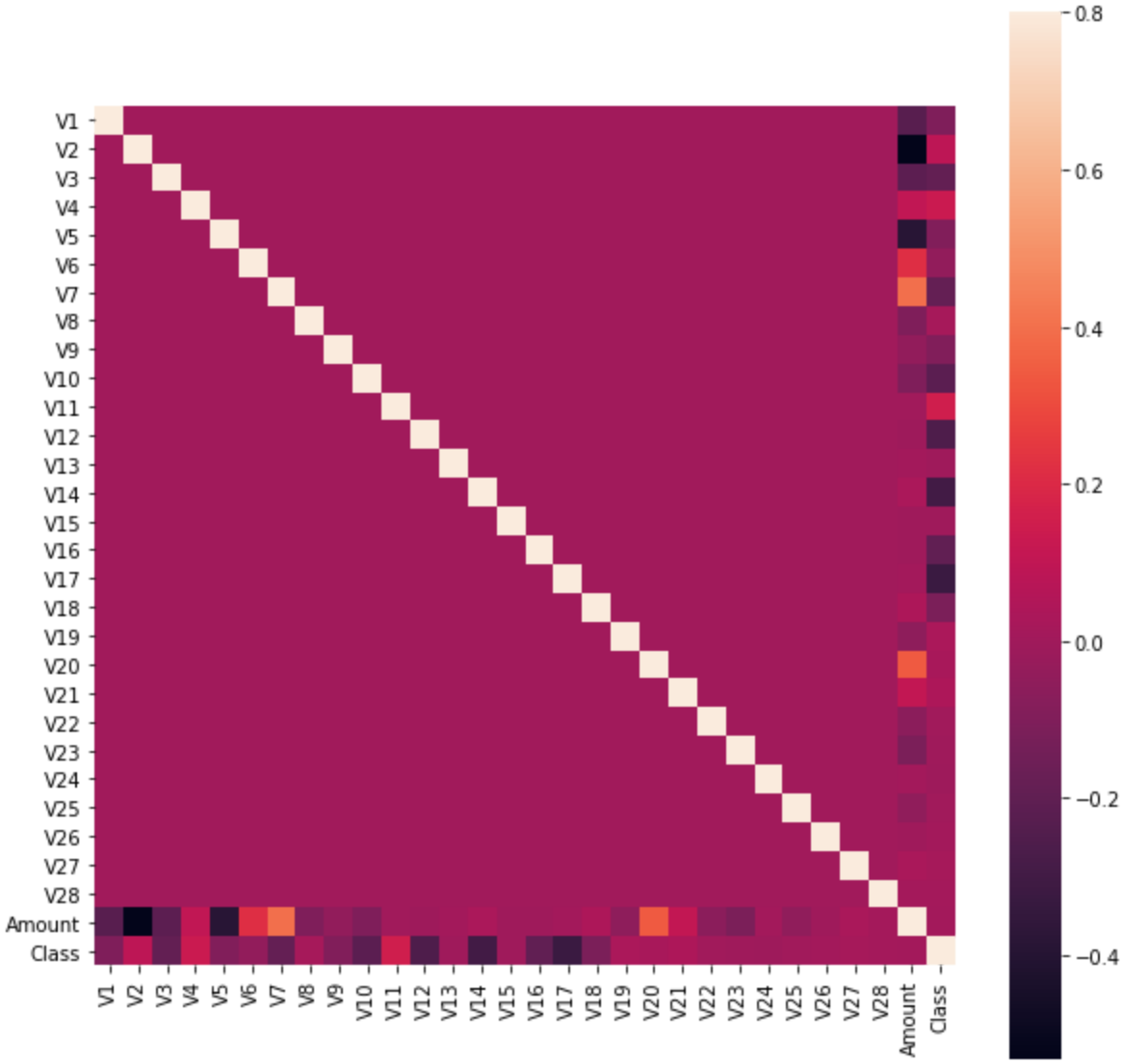
- The dataset is highly imbalanced, the positive class (frauds) account for 0.172% of all transactions. The dataset has been collected and analyzed during a research collaboration of Worldline and the Machine Learning Group of Université Libre de Bruxelles on big data mining and fraud detection.

- There are 31 features in this dataset. Due to confidentiality the 28 features, such as V1, V28 is a numerical input variable result of PCA transformation. Other 3 features that were not transformed are “Time”, “Amount”, and “Class”.

- Feature “Time” contains the seconds elapsed between each transaction and the first transaction in the dataset. The feature “Amount” is the transaction amount. Feature “Class” is the response variable and it takes value 1 in case of fraud and 0 otherwise. The dataset is available on Kaggle platform.

Statistics of the dataset

- Correlation Matrices are the essence of under-standing our data. We want to know if there are features that influence heavily in whether a specific transaction is a fraud. The correlation matrix is computed for the data



As we can see, some of our predictors do seem to be correlated with the class variable. Nonetheless, there seem to be relatively little significant correlations for such a big number of variables. This can probably be attributed to two factors:

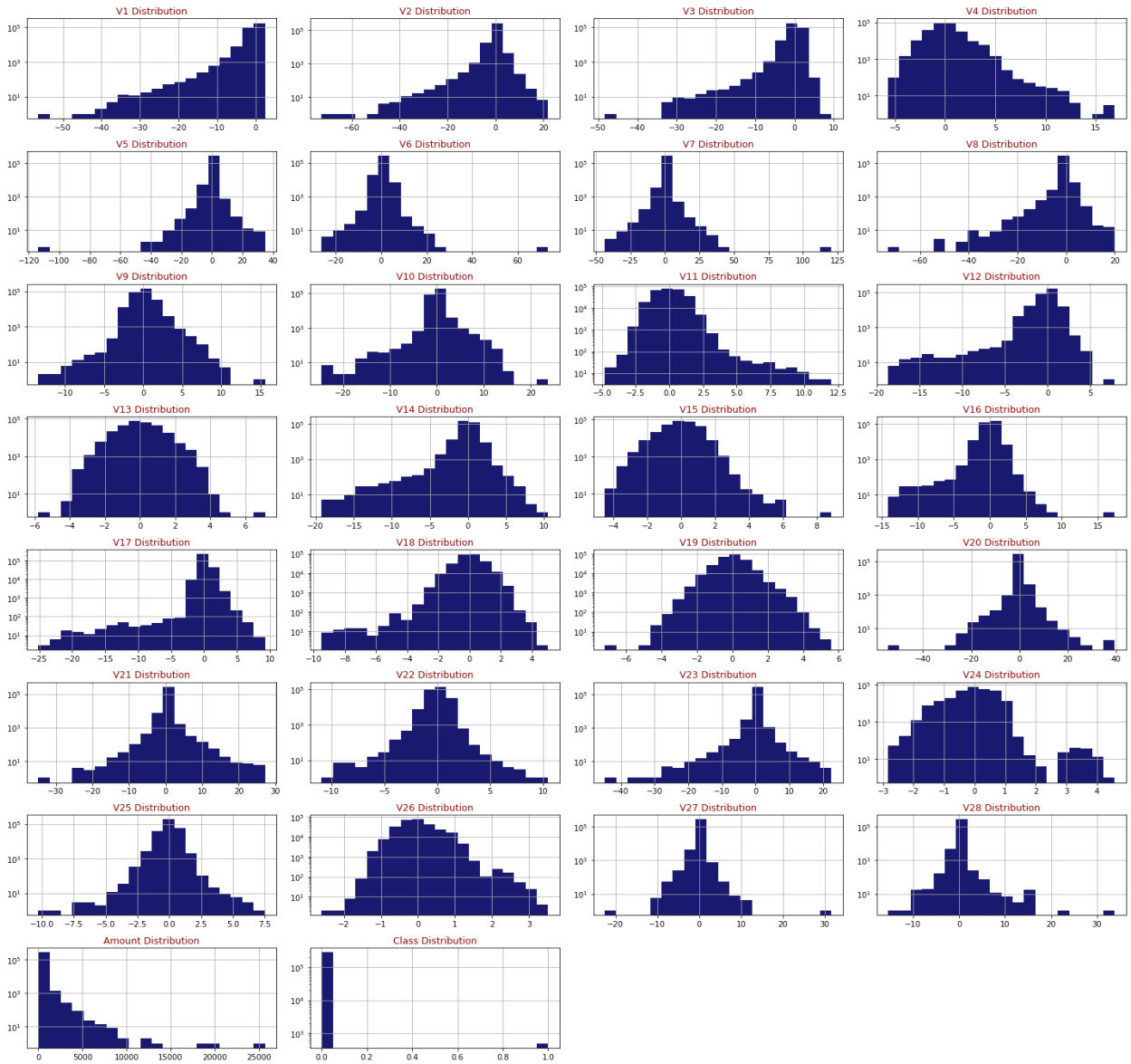
- The data was prepared using a **PCA**, therefore our predictors are principal components.

- The huge class imbalance might distort the importance of certain correlations with regards to our class variable.

The below statistics are computed for Dataset

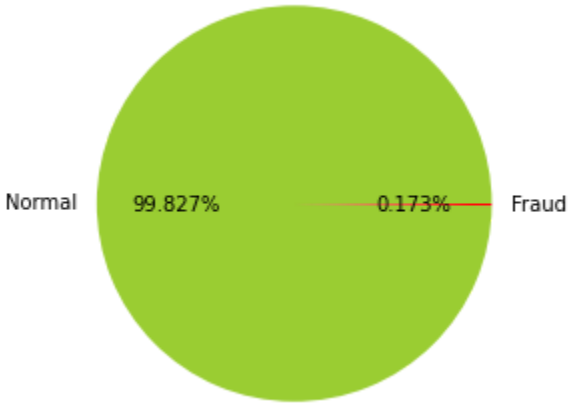
1. Percentage of non-fraudulent data = 99.827%
2. Percentage of fraudulent data = 0.17%
3. Mean of the amount feature = 88 USD5
4. Standard deviation of amount feature = 250.12 USD6
5. 1st Quartile of amountQ1 = 5.6 USD
6. 2nd Quartile of amount Median = 22.0 USD
7. 3rd Quartile of amount Q3 = 77.16 USD

Histogram for all features of dataset

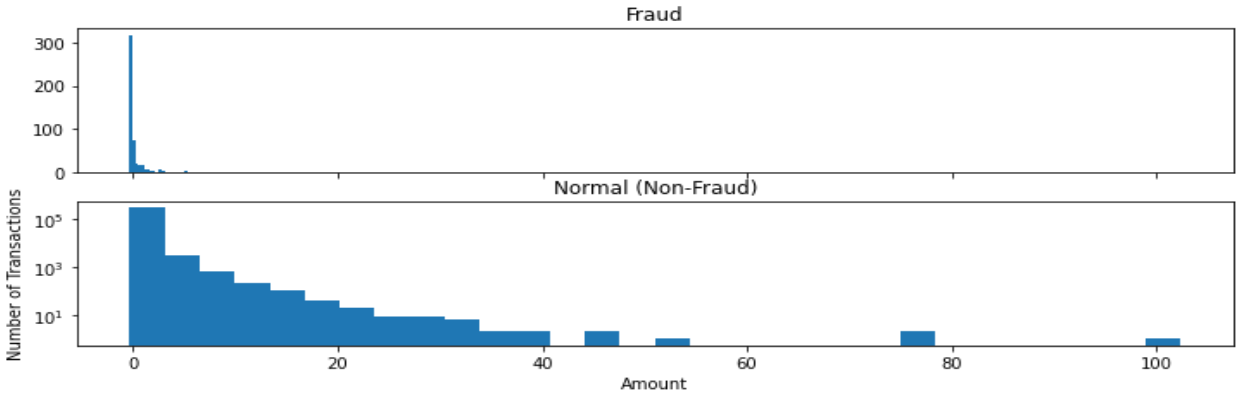


Frequency of fraudulent and non fraudulent data

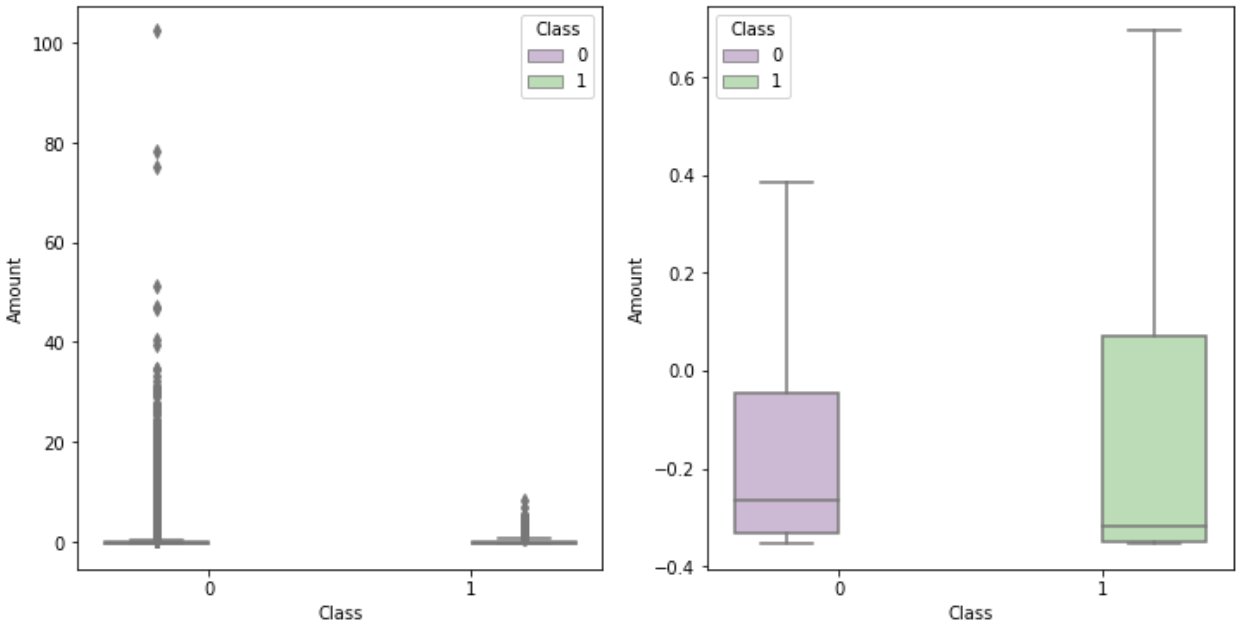
Pie Chart of Normal and Fraud data



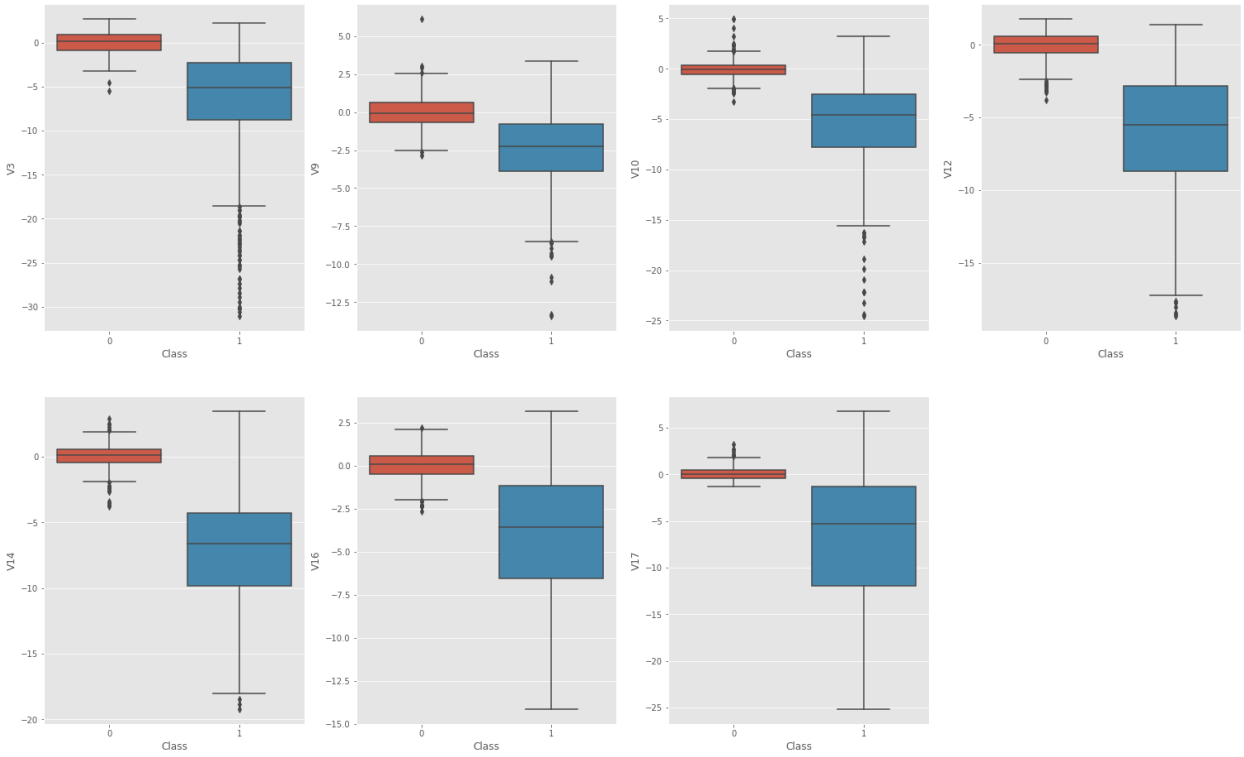
Number of transactions vs amount for fraud and normal data

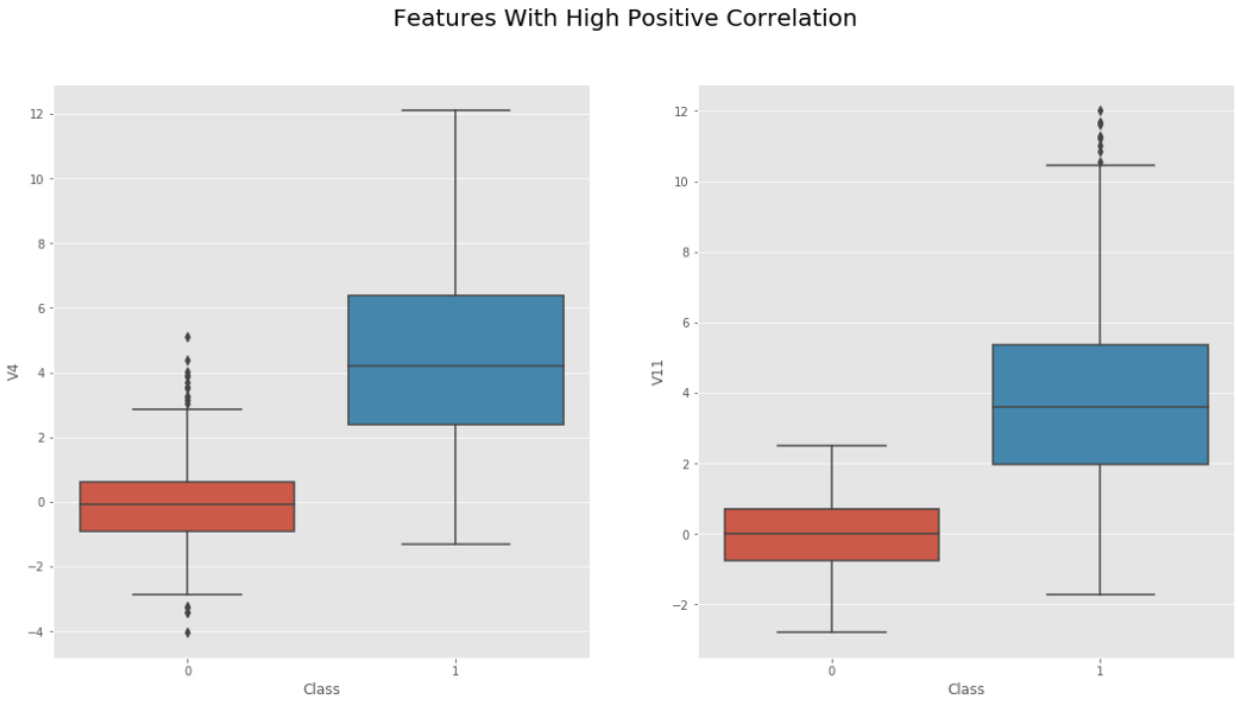


Amount vs class



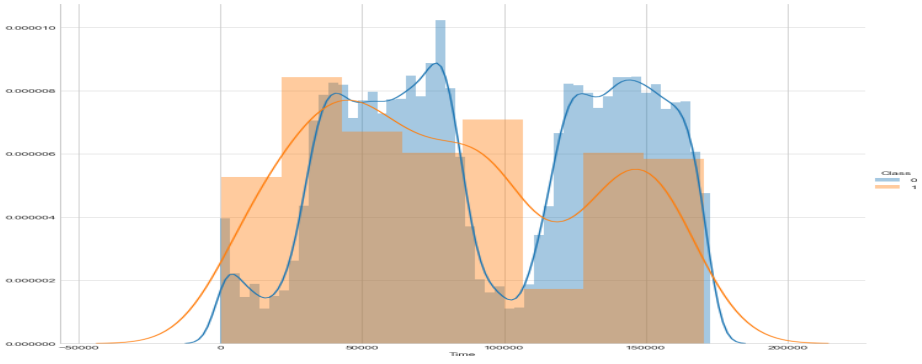
Features With High Negative Correlation





Modifications on Dataset

- As we know before, **features V1-V28 have been transformed by PCA and scaled already**. Whereas features ”Time” and ”Amount” have not. 'Time' contains the seconds elapsed between each transaction and the first transaction in the dataset.



- The above figure shows the time at which genuine and fraud transactions occurred. There is a heavy overlap of genuine and fraud

transactions throughout the time and there is no clear distinction. It is just showing the elapsed time, nothing much. **So, we will drop this variable**.

- And considering that we will analyze “Amount” with other V1-V28, they should better be scaled before we train our model using various algorithms. This is necessary because one feature, which is expressed in a very high magnitude(number), may affect the prediction a lot more than an equally important feature. The scaling removes bias towards time and amount. Certain types like Naive Bayes, do not require feature scaling, because it works in a differ-ent manner. But, in general, algorithms that exploit distances or similarities(e.g. in form of scalar product) between data samples, such as k-NN andSVM, often require feature scaling. There are many methods for scaling.

- The Standard Scaler is not used as the ”Amount” feature is not normally distributed. **So, we normalised “Amount” using a standard scaler to fit between -1 to 1. Now all features are in the same range.**

- We have used K-fold cross validation in logistic regression. We split the data into 5 groups and performed the cross validation method.

Methodology

- We first downloaded the appropriate dataset. Now, we separate the dataset into train and test data (70:30). Using this train data we train the machine using various algorithms. As mentioned above, the data is already pre-processed using PCA.

- This being a classification problem, we used certain algorithms like **Logistic Regression, K- Nearest Neighbors, Support vector machine (SVM), Decision trees, Naive bayes, Random forest and Random forest with over sampling of data**. As the data is skewed (i.e heavily imbalanced data), based on our knowledge we preferred using cost sensitive learning rather than using methods like oversampling or undersampling.

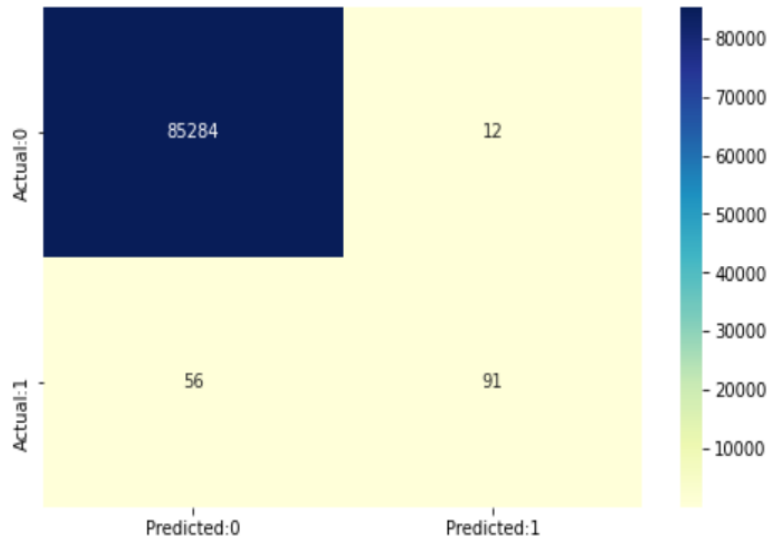
- For some of the algorithms we used the library functions such as pandas, numpy, sklearn etc. After the model is trained using the train data we predict for the unseen test data (30% of original data) which was initially separated from the original dataset.

- We build a confusion matrix using the actual data and the predicted data. Computed ROC plot to examine the threshold values, area under curve and depending on the F1 score, tp rate, fp rate, precision, recall and ROC curves as our evaluation metrics we decided the best fit model.

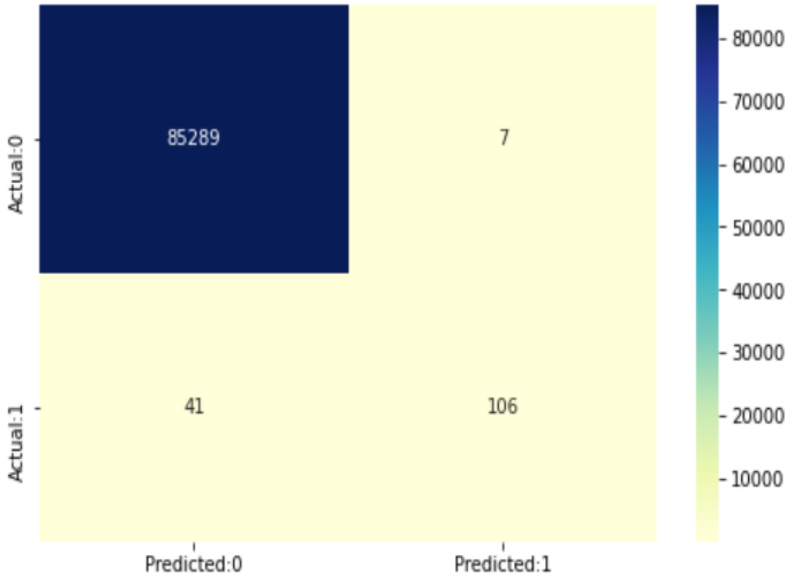
Results

Confusion Matrices on test data

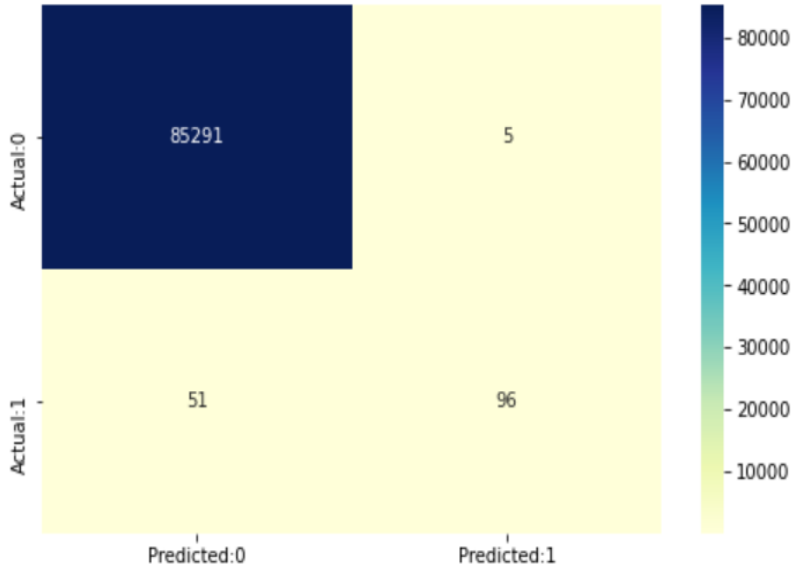
- Logistic regression



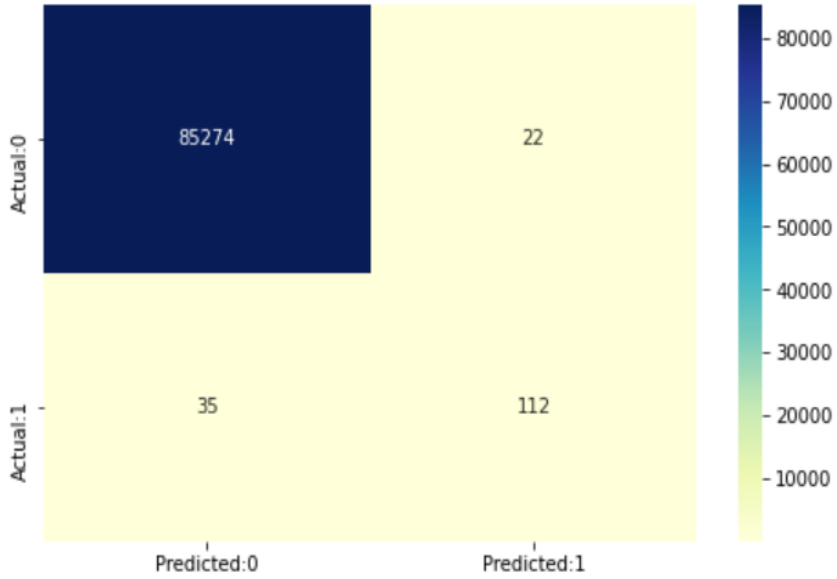
- K- nearest Neighbours (KNN)



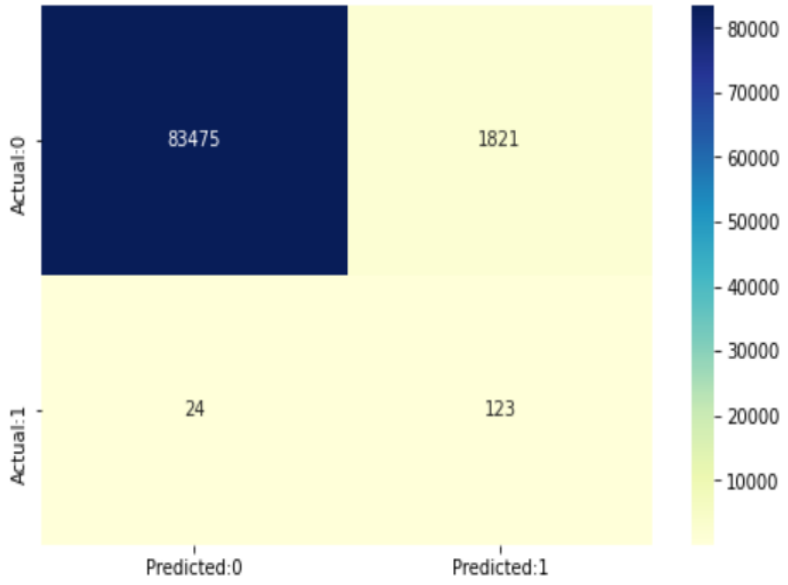
- Support Vector Machine (SVM)



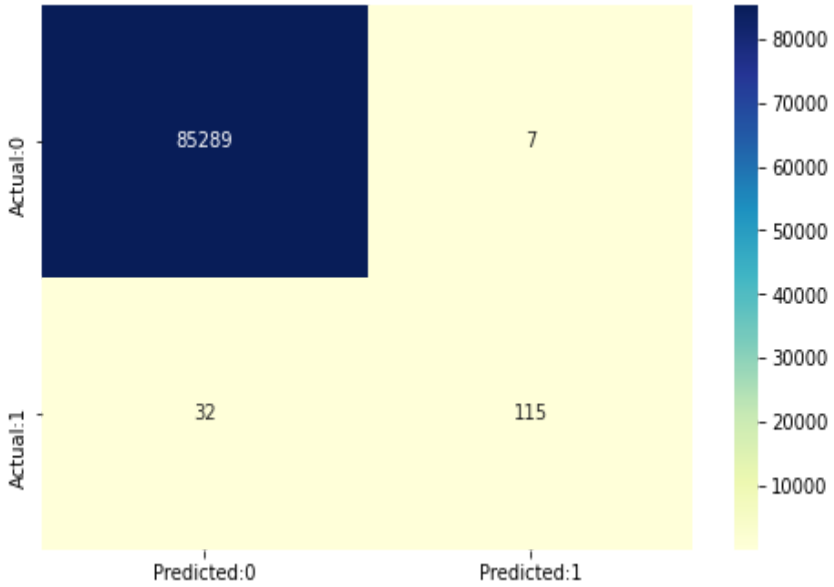
- Decision Trees



- Naive Bayes



- Random Forest



Tp rate, Fp rate

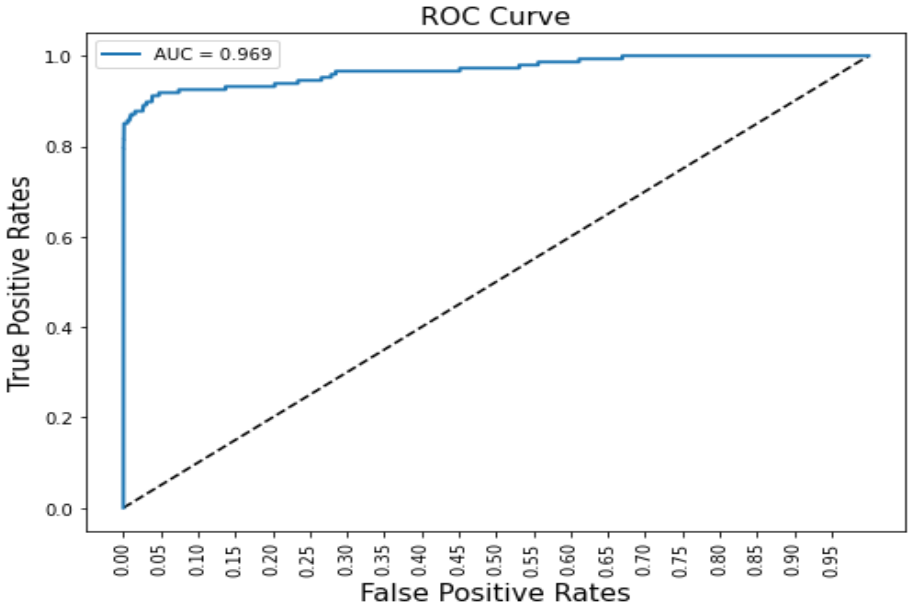
Algorithms	TP rate(%)	FP rate(%)
Logistic Regression	99.98	38.09
K-Nearest Neighbours	99.99	27.89
Support Vector Machine	99.99	34.69
Decision Trees	99.97	23.80
Naive Bayes	97.86	16.32
Random Forest	99.99	25.17

F1 score, Recall, Precision

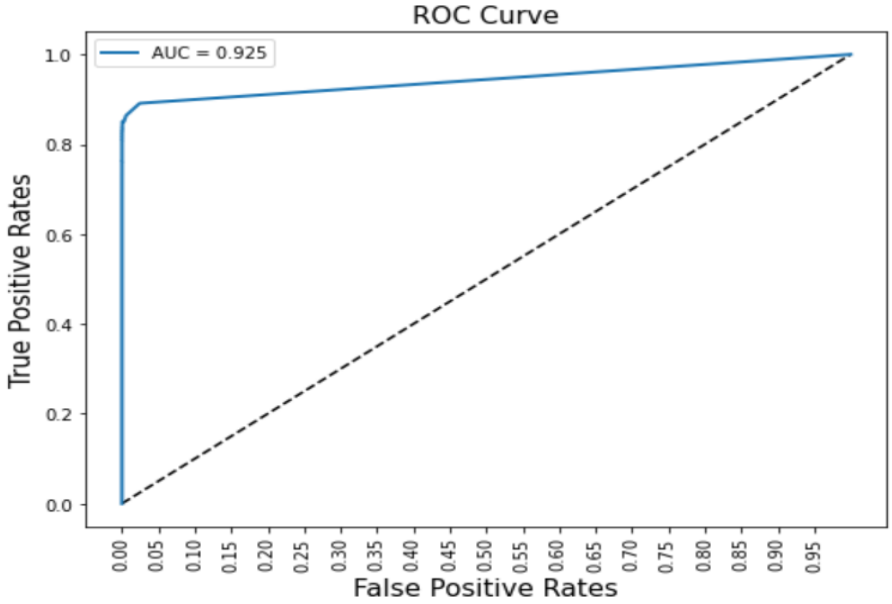
Algorithms	F1 Score	Recall	Precision
Logistic Regression	0.728	0.619	0.883
K-Nearest Neighbours	0.815	0.721	0.938
Support Vector Machine	0.774	0.653	0.950
Decision Trees	0.797	0.761	0.835
Naive Bayes	0.117	0.836	0.063
Random Forest	0.833	0.748	0.940

ROC Curves

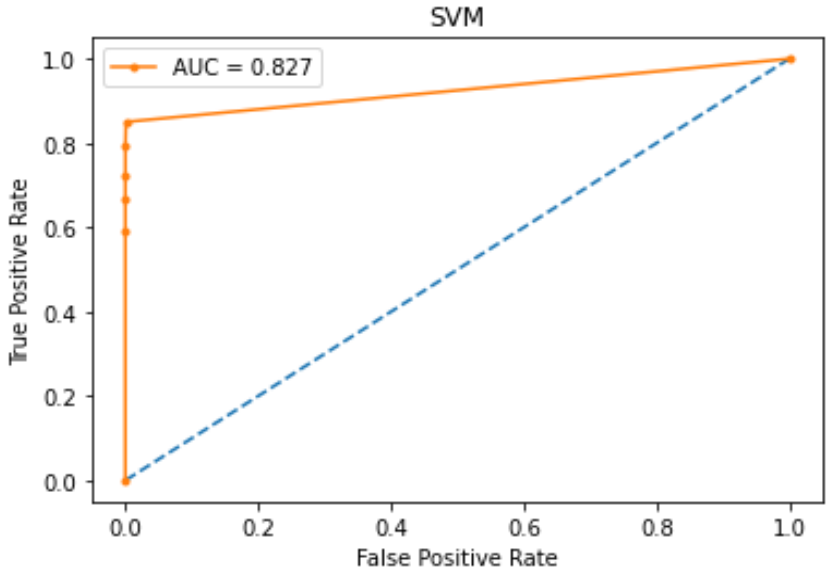
- Logistic Regression



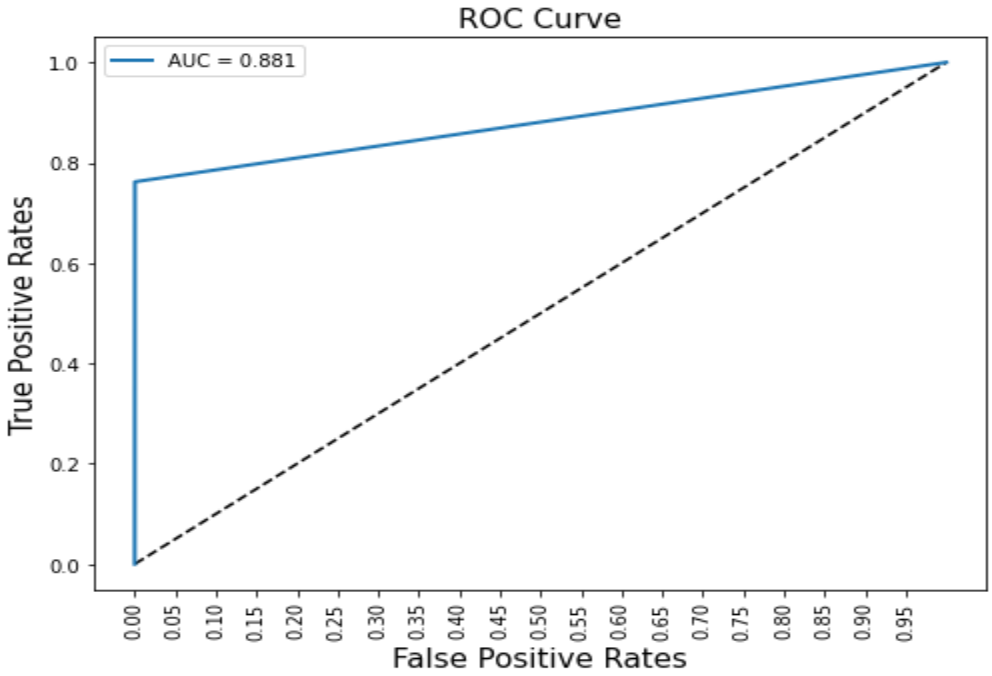
- K-Nearest Neighbours



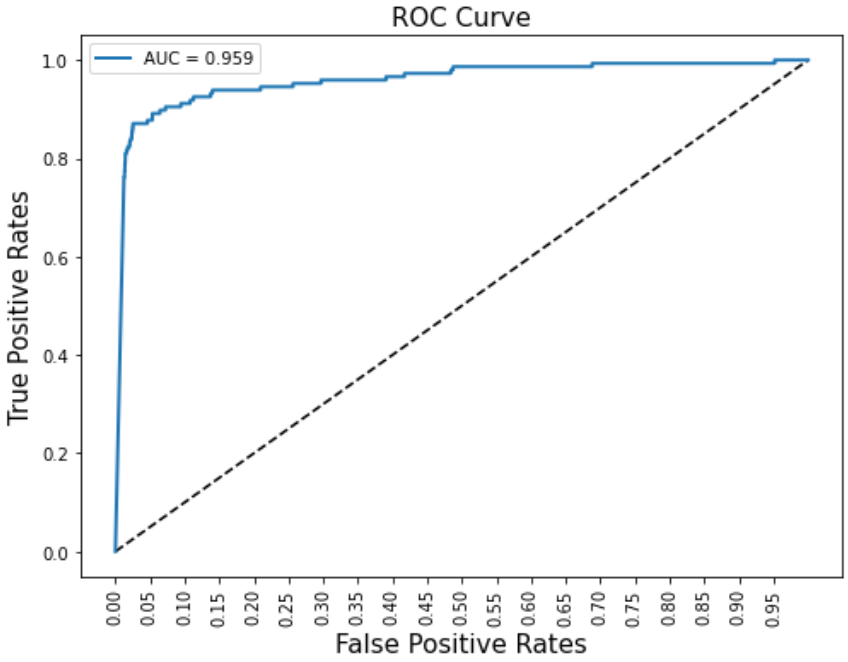
- Support Vector Machine



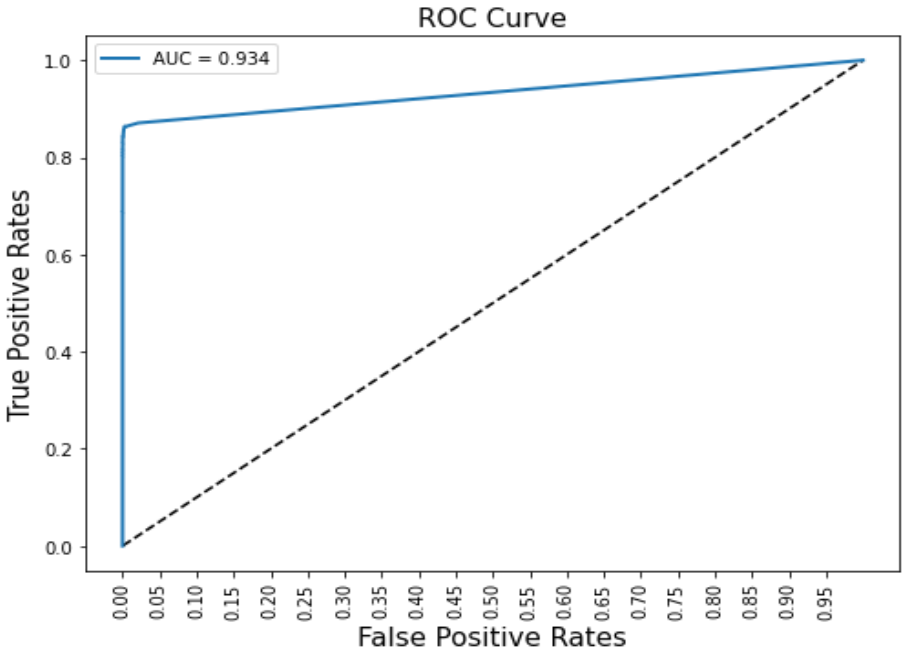
- Decision Trees



- Naive Bayes



- Random Forest



Oversampling

- There are two types of resampling methods to deal with imbalanced data,one is under sampling and another one is over sampling.

- **Under sampling:** We take random samples from non-fraud observa-tions to equal the amount of fraud observations. But we are randomly throwing away a lot of data and information.

- **Over sampling:** We take random samples from fraud cases and copy these observations to increase the amount of fraud samples in our data.In the end we are increasing the size of data.

- Hence these methods are generally not employed due the fact that former involves wastage of data and latter involves training the model with many duplicate samples.

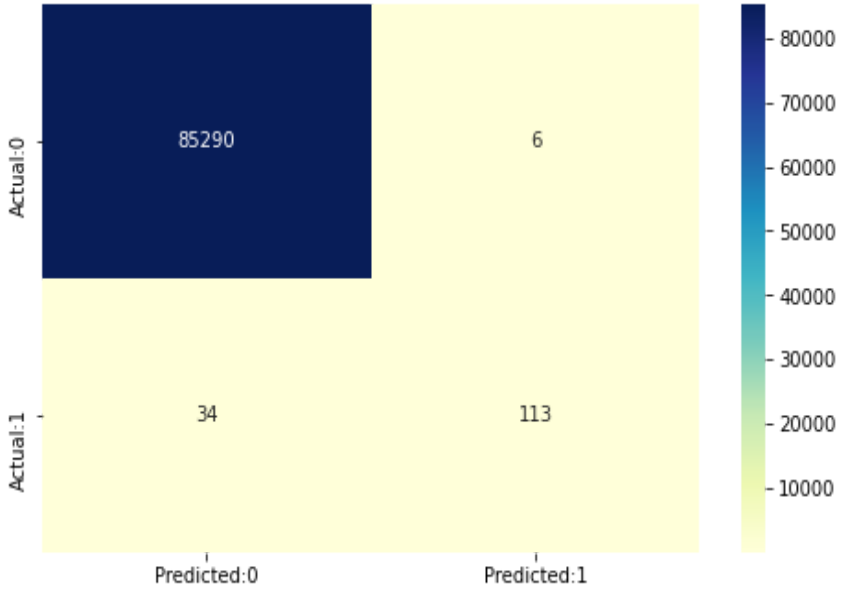
Random Forest on Oversampled data

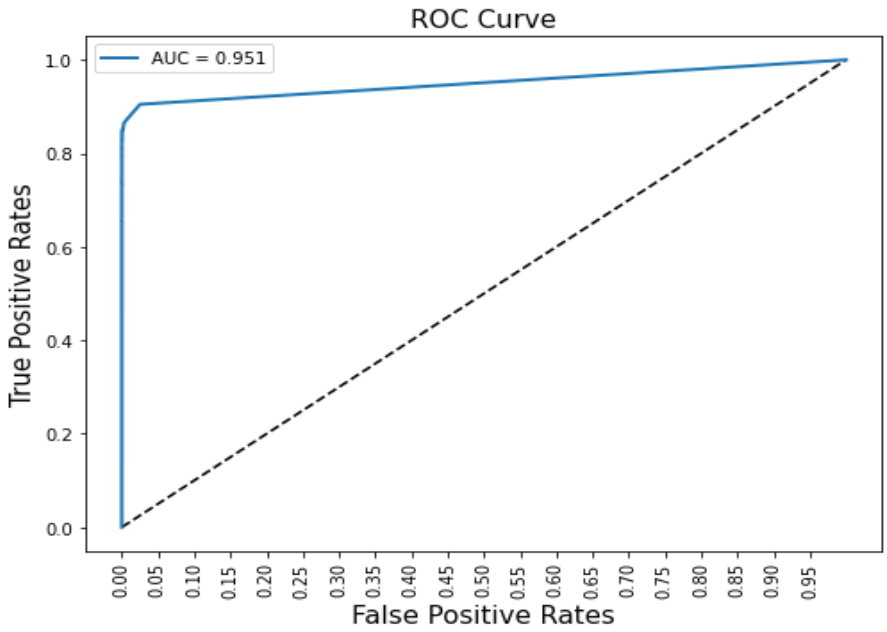
- For the sake of learning and to see how this method goes, we've tried an oversampling method and trained the model using the Random Forest algorithm. We used Random oversampler to oversample training data.

- Before oversampling non-fraudulent ('0') data is 199019 and fraudulent ('1') data is 345, after oversampling non-fraudulent ('0') data remains same as 199019 where as fraudulent ('1') data increased from 345 to 149264, which makes training dataset balanced. Test dataset remains unchanged.

Frequency of fraudulent and non fraudulent training data after oversampling

Confusion matrix and ROC curve of model after training with oversampled data





	Tp rate(%)	Fp rate(%)	F1 score	Recall	Precision
Random Forest without Oversampling	99.99	25.17	0.833	0.748	0.940
Random Forest with Oversampling	99.99	23.12	0.849	0.768	0.949

Interpretation of ROC Curve

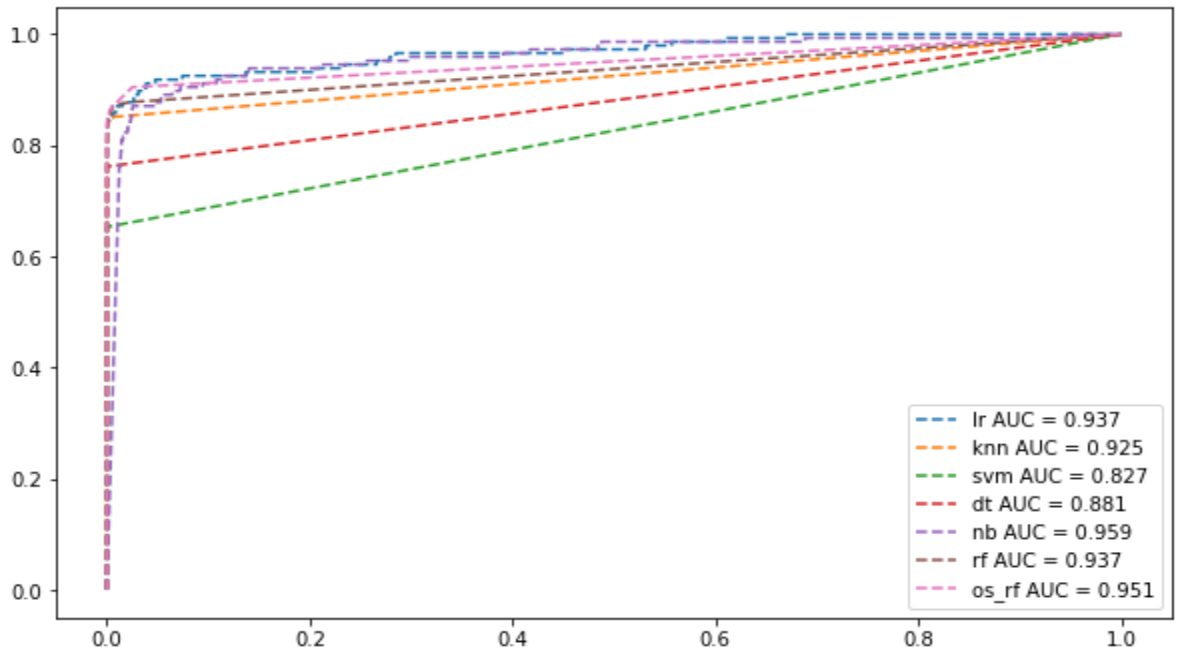
- ROC is a probability curve and AUC represents degree or measure of separability. It tells how much a model is capable of distinguishing between classes.

- AUC is used only when we want to compare amongst various classifiers.It doesn't make any sense in using AUC for a single classifier.

- Higher the AUC, better the model is at predicting 0s as 0s and 1s as 1s. By analogy, Higher the AUC, better the model is at distinguishing between fraudulent and non fraudulent cases.

- Each point on ROC represents (fp rate, tp rate) at particular threshold. ROC is plotted for different thresholds of the classifier.

- The ideal classifier is the one whose tp rate= 1 and fp rate= 0, i.e thepoint (0,1) on the ROC curve.



Data transformation

Feature selection by using filter method

- First we removed class feature from the data set. Then we make use of the corr() function to return a correlation matrix between all features.

- Then we run a loop through the correlation matrix and if the correlation between any two features is greater than the correlation criteria (here we considered criteria as 0.8) then we remove any one of those two features from our dataset and consider the remaining subset as our final dataset.

- But in our case we get an empty correlation matrix, which means no two features in our dataset have correlation greater than 0.8. **So, we are not dropping any feature from our original dataset.**

Discussion and Conclusion

- We want tp rate to be 100% and fp rate to be 0% in ideal scenario

- We can see that the accuracy of all the models is close to 100% which implies misclassification error to be close to 0%. But the costs involved in misclassifying both the classes is not equal and hence misclassification error cannot be our metric. So, we chose confusion matrix and ROC along with F1 score, precision and Recall as our metrics

- From the tp rate and fp rate values for two models we can see that the model which was trained using Random Forest with oversampling has better results.

- From AUC of ROC values for two models we can see that the model which was trained using Naive Bayes and Random Forest with oversampling has better results.

- From the F1 score values for two models we can see that the model which was trained using Random Forest with oversampling has better results.

- From the recall values for two models we can see that the model which was trained using Naive Bayes and Random Forest with oversampling has better results.

- From the above points we can see that Random Forest with Oversampling performs better for most of the metrics. So we can conclude that **Random Forest with Oversampling** is the best model.

Comparison with other blogs and research papers which are using same data

- We have gone through some of the research papers and blogs on credit card fraud detection which are using the same data as us, following are some points which I noticed are different from ours.

➢ Many of the papers and blogs used the AUC of precision recall curve as metric. There are two types of classification errors - one is classifying non fraudulent as fraud and the other is classifying fraudulent as non fraud.

➢ Second type of error is more dangerous than the first one because calling a fraud transaction as non fraud is a huge loss. So we gave more priority to recall other than precision and not considered AUC of Precision recall curve.

➢ Many papers and blogs used SMOTE technique to handle imbalanced data. But we used oversampling and undersampling to handle imbalanced data. Undersampling gave very poor results so we continued only with oversampling.

➢ Some papers and blogs used Neural networks and XGboost algorithms which we haven't included in our project.

Following are links of research papers and blogs which we referred

A Comparative Analysis of Various Credit Card Fraud Detection Techniques

Credit Card Fraud Detection. Staying Vigilant in the Virtual World | by Randy Macaraeg

Detecting Credit Card Fraud Using Machine Learning | by Lukas Frei

http://www.jcreview.com/fulltext/197-1594986782.pdf

Application of Machine Learning Techniques in Credit Card Fraud Detection

Acknowledgement

- Throughout this project, we had to take the help and guidance of some respected persons, who deserve our deepest gratitude. As the completion of this project gave us much pleasure, we would like to show our gratitude to **Dr.M Gopal sir** for giving us good guidelines for the project throughout. Wewould also like to extend our gratitude to **K.Venugopal sir**, who helped us by clarifying the doubts, and building concepts which are very useful in the project.