CARD FRAUD DETECTION

Abishek Anbarasan
Purdue University
aanbara@purdue.edu

Mayur Deo
Purdue University
mayurdeo@purdue.edu

ABSTRACT

In this project, we use data mining approach to solve the problem of fraudulent transaction by detecting whether it is fraud or not. The motivation behind selecting this project is because card fraud is a serious and elevating problem. Although, many predictive models are in actively in use, they do not perform well because of change in patterns over time and that is where data mining and machine learning models comes to the rescue. This paper evaluates five different data mining approaches namely logistic regression, random trees, support vector machines, and two other boosting techniques to provide a better detection. This is a Kaggle competition carried out by IEEE-CIS incorporated with Vesta corporation.

Keywords

Data mining; Kaggle; Machine learning

1. INTRODUCTION

A staggering $24.26 Billion was lost in 2018 due to payment card fraud worldwide and United States still leads the world as the most card fraud prone country with 38.6 % of reported card fraud losses in 2018. This is a huge problem because it costs consumers and financial company billions of dollars annually, and fraudsters continuously try to find new rules and tactics to commit illegal actions. What makes it even more interesting is that one may not be even aware of fraud even if they are victimized. Plenty of steps have taken to address the issue but with less efficacy because of the lack of predictive model's tendency to perform well with an unseen pattern. It is still a major issue in datamining techniques, but it performs better than the former but is less widely used.

This essentially needs to be reduced with constant innovations and in this project, we evaluate in detail random forests, support vector machines, logistic regression, light gradient boosting and extreme gradient boosting algorithms as an attempt to detect the fraudulent transactions better. Though the data mining techniques are in profuse use in many fields, little have been their usage in card detection. But, many papers in early 1990's have used neural networks and to some extent Markov models. Their study was focused on impact of aggregating transaction level data of fraudulent prediction. The reason we have not tried neural networks is because they are prone to get overfit and may get struck in a local minima or saddle point. Also, the data mining models gives flexible optimization approach. We have preferred ensemble methods because they have very good generalization performance. Many statistical methods like logistic regression, nearest neighbor, Bayes classifier have been used to develop models and with evolution of AI and machine learning, the other advanced techniques were also used and compared with the earlier mentioned models. We have trained, tested, parameterized and compared all those stated models based on the metrics, which is discussed in detail later and lastly we have evaluated the results.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full cita- tion on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or re- publish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

© 2019 ACM. ISBN 978-1-4503-2138-9.
DOI: 10.1145/1235

2. Literature Review

[1] Artificial Neural Networks: The type of neural network in this case is the Feed Forward Multi-Layer Perceptron. It consists of different layers of perceptron's that are inter-connected by a set of weighted connections. The three types of distinguished layers are: Input layer, Hidden layer and Output layer. The type of learning used in this case is supervised learning and the algorithm used for it is called Backpropagation of Error signals or short Back prop, every iteration of this algorithm consists of the Forward pass and the Backward pass. To get a clear idea of the result the authors of the paper introduced a representation of the output called the Receiver Operating Curve (ROC). After training the data it will be applied to features it has never seen before, for the purpose of testing. Finally, they will be tracing the true positive rates and the false positive rates to get a good understanding of the outputs received.

[2] Bayesian Networks: A Bayesian network is a directed acyclic graph that consists of a set of random variables, each variable has a finite set of mutually exclusive states. A set of directed links or arrows connect the pairs of nodes. A Bayesian network represents the dependence between variables and gives a compact specification of the joint probability distribution. Bayesian belief networks are the scheme for knowledge representation in this case, it mainly has two steps: 1) Identifying the topology of the network 2) Learning the parameters. The authors trained and then tested the model on an unseen set of data. A very similar approach as the previous paper was used to develop and see the results (Receiver Operating Curve (ROC))authors of the paper introduced a representation of the output called the Receiver Operating Curve (ROC). After training the data it will be applied to features it has never seen before, for the purpose of testing. Finally, they will be tracing the true positive rates and the false positive rates to get a good understanding of the outputs received.

[3] Genetic Algorithm: Genetic algorithm is an optimization technique and evolutionary search based on the principles of genetic and natural selection, heuristic used to solve high complexity computational problems. The authors in this paper strive to find the detection of credit card fraud mechanism and examine the result based on the principles of this algorithm. The methodology used for genetic algorithm is shown below

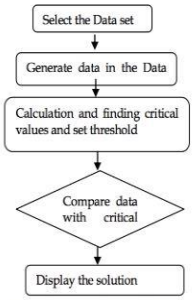
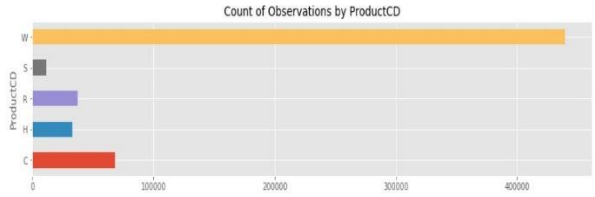


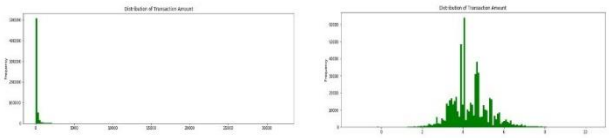
Fig.1 Procedure for Genetic Algorithm

3. DATASET

The dataset is obtained from Vesta corporation via Kaggle competition. The training dataset is of the shape 590540*434 after merging the train identity and train transaction datasets. The two datasets have Transaction Id's as common. The Train transaction has many important features like Time delta which is the transaction timeline from a given reference. The data feature isFraud is the response variable which is a binary feature. It also has Product codes which haven't been revealed.

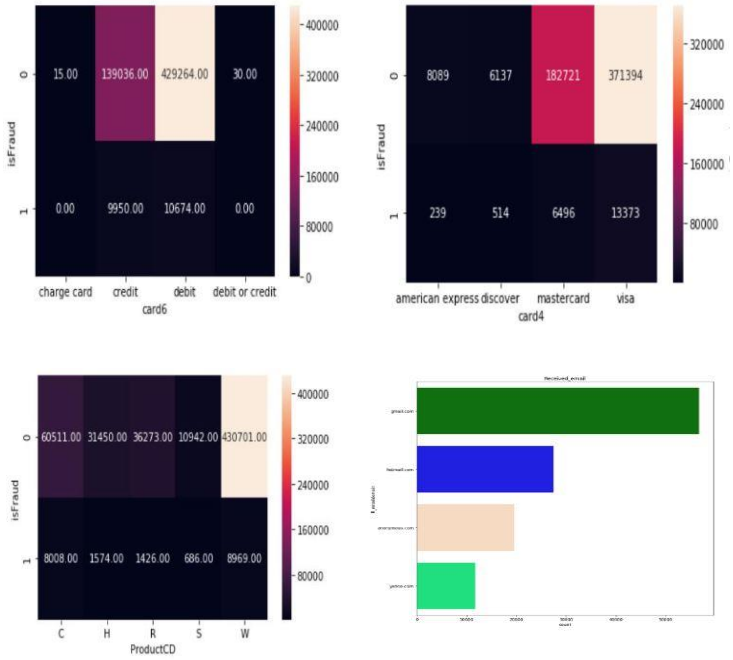


The Transaction amount is given in USD and it has a fair distribution only after transforming it which can be seen in the plot below.

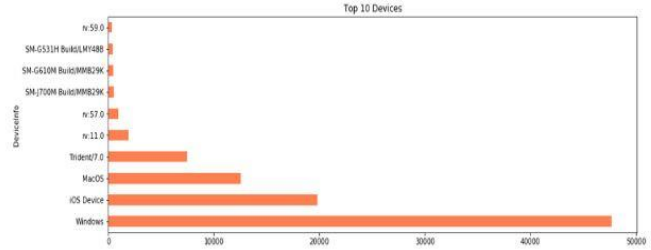


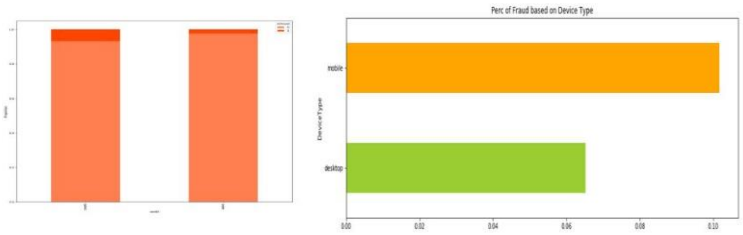
We have created new features hour, day and month when the transaction is been done. These details are extracted from Time delta which is in the seconds format and spans over four months in time. This is also been offset to get correct day of the week. These new features were very important in determining which time of day of week were the fraudulent transactions at peak, etc.

There are also other important features like, card1, card2, card3, card4, card5 and card6 which has details of card type, card category, issued bank, country, etc which were very useful. Also, addresses of the receiver and purchaser were given including their email domain which can be visualized in the below plot.



There are also other features which are mentioned as vesta engineered features, and the actual meaning of it were masked. The train Identity also has Transaction Identity which is to be merged with train transaction. It has categorical features like id which has only True, False or nan values and their meanings are also masked. They contain the type of device like mobile, desktop, etc from which the transactions were performed and Device Info such as Samsung, RedMI, Ios, windows, etc. The below plots visualizes the top devices from which transactions are done, the device type counts and most importantly the percentage of fraud which were done on device type which has quite interesting result.



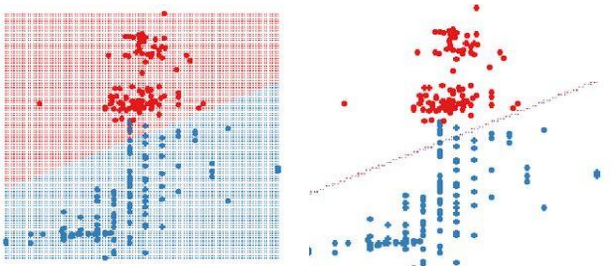


The testing set is of shape 506691*433 and it doesn't have the ground truth labels which made it impossible to find test roc.

There were a lot of missing values, and there were 232 columns that has missing values more than 30 percent and few columns had more than 90 percent missing values, which were dropped because, they add no importance because of the lack of information it could provide. There are about 90 columns with missing values in between 10 and 30 percentage, with majority of them being Vesta engineered rich features. We imputed values to the columns that has less than 10 percent missing values and dropped the rows of columns that seemed to offer no importance. For the Categorical values with less than ten percent values, imputation with mode is considered.

Because the datasets were taking up huge memory, we employed a memory reduction technique which affected the accuracy by an inconsiderable amount (very less).

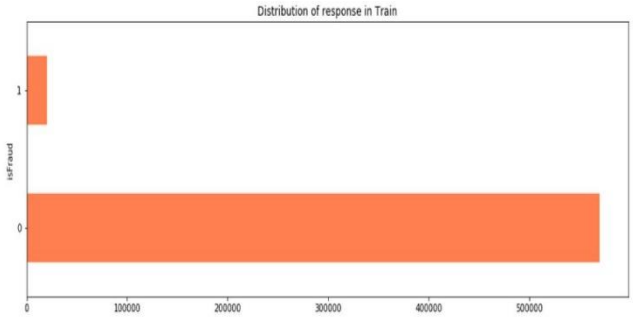
Also, because of large number of features we employed feature reduction techniques like Principle component Analysis and Linear Discriminant Analysis. The steps involved in LDA are computing the d dimensional mean vectors for the two classes from the dataset followed by computing scatter matrices and corresponding eigen vectors. Then, we sort the eigenvectors by decreasing eigenvalues and choose *k* eigenvectors with the largest eigenvalues to form a *d*×*k* dimensional matrix and then to matrix to transform the samples onto the new subspace.



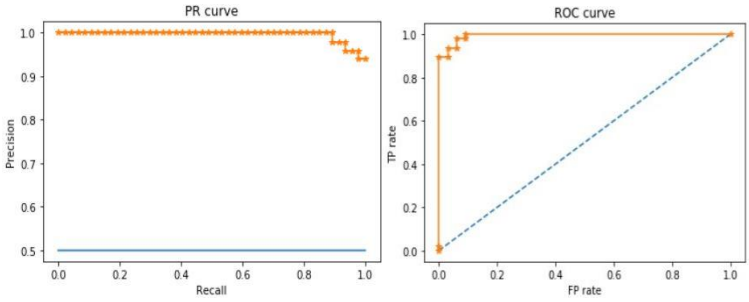
In our project we performed both LDA and PCA and we choose principle component Analysis, because it explained the total variance in the data better than LDA and we worked with only 100 features for modelling which yielded impressive results.

One of the major problems that we had to tackle was dealing with the class imbalance. Almost 96 percent of the values in the training set were found to be not associated with any type of fraud activity. We only have 4 percent of the data to be identified as fraudulent. This when not dealt with correctly will lead to prediction results which will be high, but they will be misleading because the number of False positives might be very dangerous in card fraud detection. So, we have followed two methods to tackle this problem.

- Under Sampling
- Over Sampling

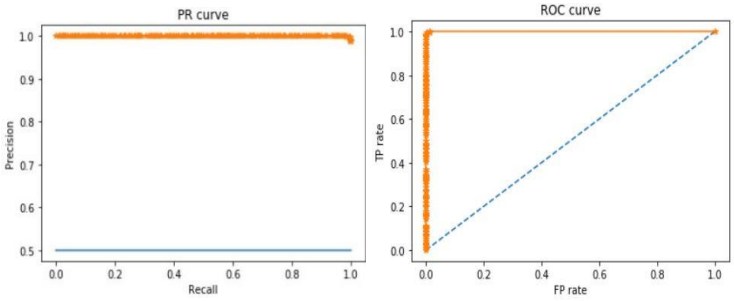


3.1 Under Sampling



Under sampling is a method in which the majority class is reduced to match the specified number or the minority class. Then the results of it are evaluated by fitting a logistic regression classifier to check the area under the ROC curve.

3.2 Over Sampling



Over sampling is a method in which the minority class is increased to match the specified number or the majority class. Then the results of it are evaluated by fitting a logistic regression classifier to check the area under the ROC curve.

From both the plots we could see that the results are pretty similar and we choose under sampling of the majority classes with less than half of the ratio for our modeling. We did not consider using synthetic sampling because of large number of features to be generated.

4. METRICS

We have considered the following metrics for evaluation of each of our classification algorithm and compared them.

4.1 AUROC

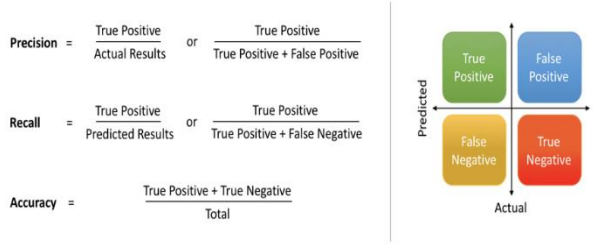
ROC curve (**receiver operating characteristic curve**) is a graph showing the performance of a classification model at all classification thresholds. This curve plots two parameters:

- True Positive Rate
- False Positive Rate

ROC curve plots TPR vs. FPR at different classification thresholds. Lowering the classification threshold classifies more items as positive, thus increasing both False Positives and True Positives. **AUC** stands for "Area under the ROC Curve." That is, AUC measures the entire two-dimensional area underneath the entire ROC curve (think integral calculus) from (0,0) to (1,1). AUC, or Area Under Curve, is a metric for binary classification. It's probably the second most popular one, after accuracy. AUC provides an aggregate measure of performance across all possible classification thresholds. One way of interpreting AUC is as the probability that the model ranks a random positive example more highly than a random negative example. As a measure of classification performance AUC has many advantages compared to other "single number" performance measures:

- Independence of the decision threshold.
- Invariance to prior class probabilities or class prevalence in the data.
- Can choose/change a decision threshold based on cost-benefit analysis after model training

4.2 PRECISION, RECALL, ACCURACY



Accuracy is the ratio of the correctly labeled subjects to the whole pool of subjects. It is the most intuitive one.

Accuracy = (TP+TN)/(TP+FP+FN+TN)

numerator: +ve labeled diabetic people ; denominator: all +ve labeled by our program (whether they're diabetic or not in reality).

Recall is defined as the number of true positives divided by the total number of elements that actually belong to the positive class (i.e. the sum of true positives and false negatives, which are items which were not labeled as belonging to the positive class but should have been). Recall is the ratio of the correctly +ve labeled by our program to all who are diabetic.

numerator: +ve labeled diabetic people; denominator: all people who are diabetic (whether detected by our program or not)

In statistical analysis of binary classification, the F1 score (also F-score or F-measure) is a measure of a test's accuracy. It considers both the precision p and the recall r of the test to compute the score. Precision, p is the number of correct positive results divided by the number of all positive results returned by the classifier, Recall, r is the number of correct positive results divided by the number of all relevant samples (all samples that should have been identified as positive). The F1 score is the harmonic mean of the precision and recall, where an F1 score reaches its best value at 1 (perfect precision and recall) and worst at 0.

$$\text{F1 Score} = 2 * \frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}}$$

5. VALIDATION

We have followed two validation methods for validating our model and they are as follows,

5.1 Adversarial Validation

In any dataset when the data is split between training and testing, if done randomly leads to a violation of their distributions being identical, it is then difficult to make a representative validation set. The method of adversarial validation is used for selecting training examples most like test examples and using them as a validation set. The core of this idea is training a probabilistic classifier to distinguish train/test examples i.e. getting identical distributions for the two, thus helping us in achieving a better AUC score.Involved Method - Check the degree of similarity between training and tests in terms of feature distribution: if they are difficult to distinguish, the distribution is probably similar and the usual validation techniques should work. It does not seem to be the case, so we can suspect they are quite different. This intuition can be quantified by combining train and test sets, assigning 0/1 labels (0 - train, 1-test) and evaluating a binary classification task.

Pseudo Code :

```
train['target'] = 0
test['target'] = 1
train_test = pd.concat([train, test], axis =0)
target = train_test['target'].values
del train, test
train, test = model_selection.train_test_split(train_test, test_size=0
.33, random_state=42, shuffle=True)
```

5.2 Cross Validation

We have considered 3fold cross validation for all of our models and have estimated their performance based on the average estimates obtained. We have split the data into 70 percent training set and 30 percent validation set.

6. MODEL IMPLEMENTATIONS

We have used five Machine Learning algorithms in this project. Firstly, data preprocessing was conducted to reduce the data size, remove missing and unknown values, and generating train/test split using Adversarial validation. The mentioned machine learning algorithms were then used to build the models which learns to detect if a credit card transaction is fraudulent or not. We have explained all the five algorithms in detail in the upcoming subsections.

1. We performed 3-fold cross validation for Random Forest Model. The hyperparameters that we tuned for Random Forest model is Number of Trees set at 100, 200 and 300 and the minimum number of samples at the leaf node 10, 30, 50. And max Depth at 8, 12. We fit the train set to the Random Forest algorithm with the optimal tuned parameters. We also plotted the cross validated results to find the optimal number of trees and its corresponding minimum number of samples at the leaf node to perform modelling fitting on combined train set and predict test set

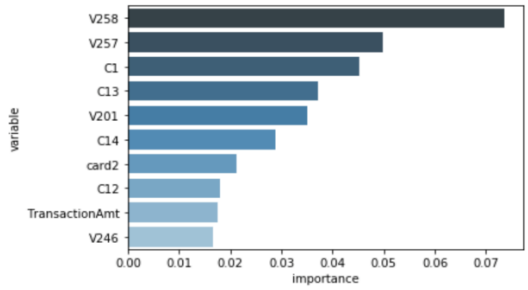
2. We performed 3-fold cross validation on the train and validation set for Logistic Regression Model. It has only dummy significance parameter tuning. This does not change the model significantly. Best way to tune is to cross validate the data set. Fit the train set to the Logistic algorithm. Performed the prediction to validation set and plotted the cross validated results to find the optimal performance to perform modelling fitting on combined train set and predict test set.

3. We performed 3-fold cross for Support Vector Machine Model. The hyperparameters that we tuned for support Vector Machine model varies from kernel to kernel. They include, gamma tuning and regularization parameter. We considered various kernels like linear, sigmoid. Then, we Fit the train set to the SVM algorithm. Plotted the results to find the optimal kernel and its corresponding optimal hyperparameter We perform modelling fitting on combined train set and predict test set

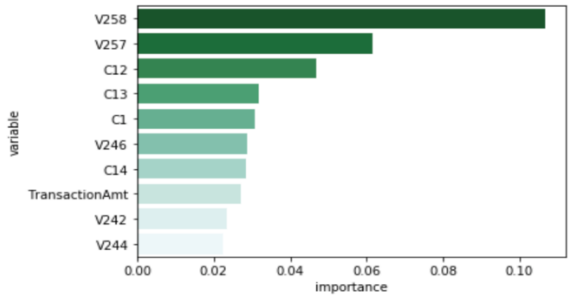
4. We performed 3-fold cross validation for Gradient Boosting Model. The hyperparameters that we tuned for Gradient Boosting model is number of trees (1000, 1500), interaction depth (1,3), learning rate (0.001, 0.005, 0.008), maxdepth(9,12). The tuning algorithm runs to find the optimal **number of trees, interaction depth,** perform modelling fitting on combined train set and predict test set. We get predictive metrics for test set. The prediction probability is valued at 1 for >0.5 and 0 for <0.5.

7. VARIABLE IMPORTANCE

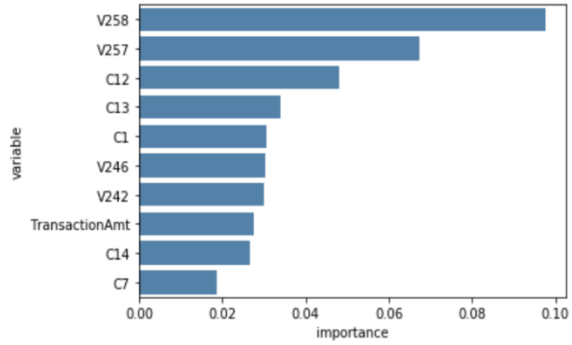
7.1 Random Forests



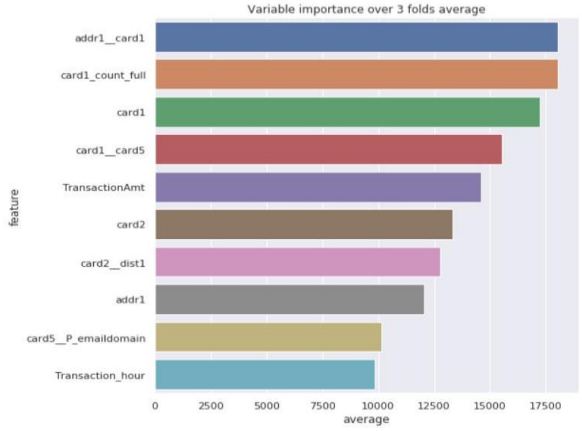
7.2 Logistic Regression



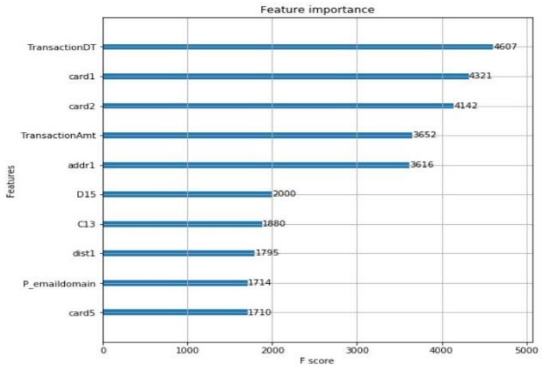
7.3 Support Vector Machines



7.4 Light Gradient Boosting

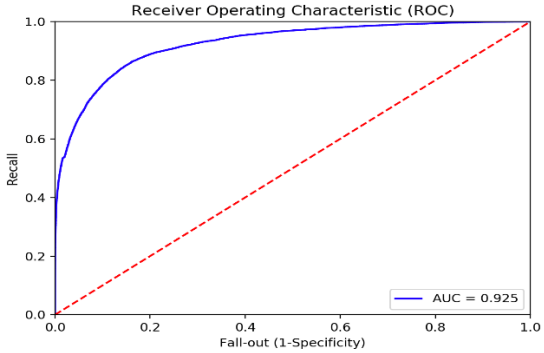


7.5 Extreme Gradient Boosting

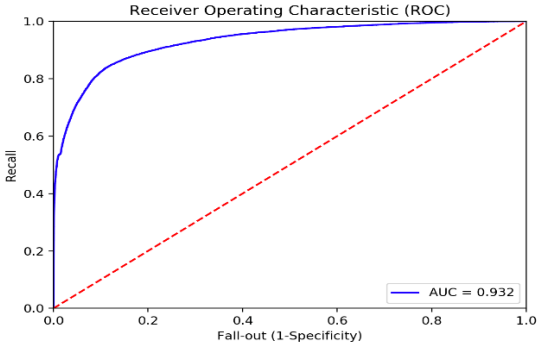


8. AUROC

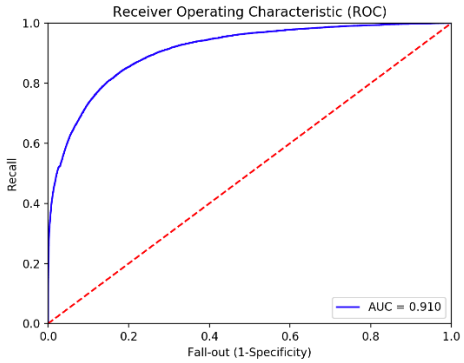
8.1 Random Forests



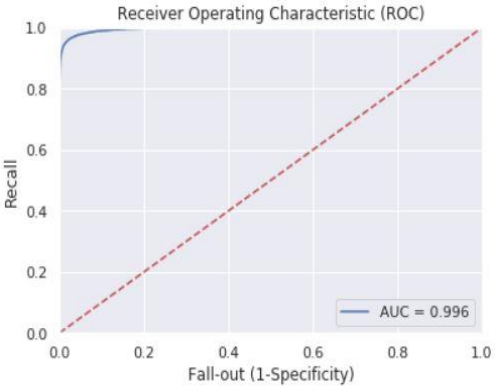
8.2 Logistic Regression



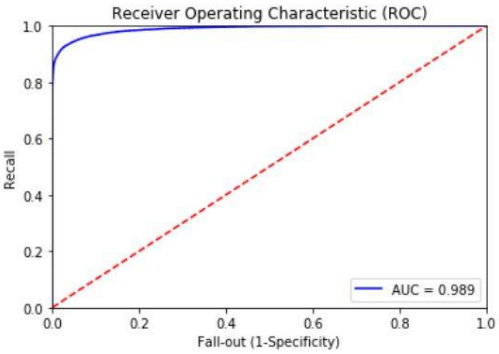
8.3 Support Vector Machines



8.4 Light Gradient Boosting

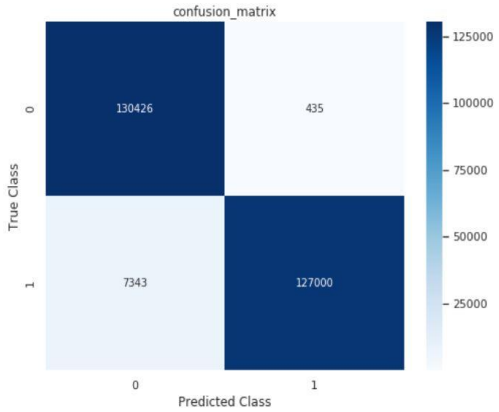


8.5 Extreme Gradient Boosting



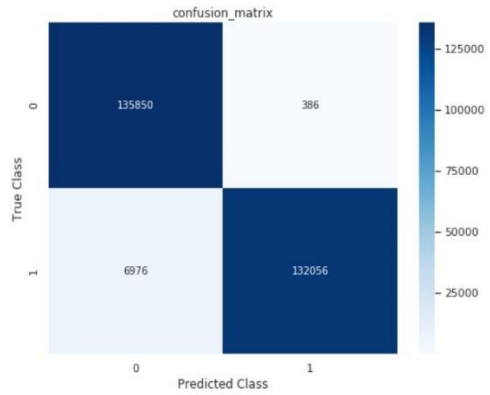
9. CONFUSSION MATRIX

9.1 Random Forests



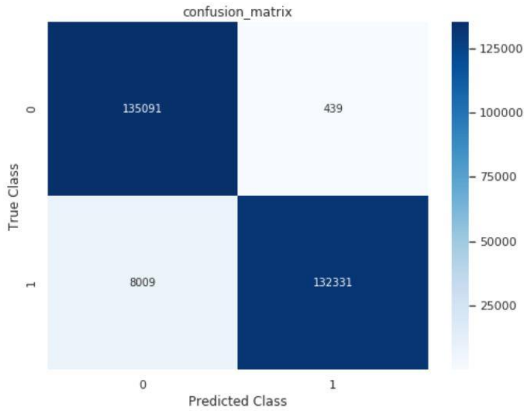
F1 SCORE=0.9710
PRECISION=0.9967
SPECIFICITY=0.9966
FP RATE=0.0034
SENSITIVITY=0.9467

9.2 Logistic Regression



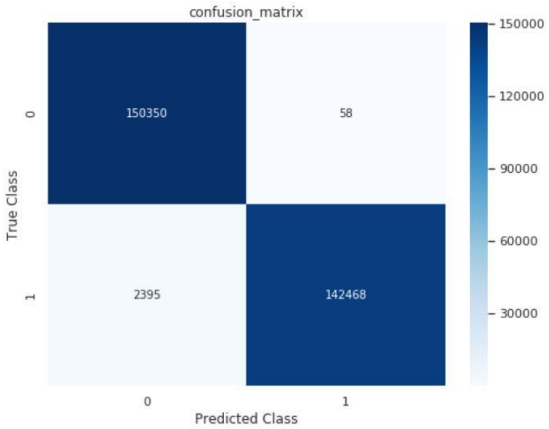
F1 SCORE=0.9737
PRECISION=0.9972
SPECIFICITY=0.9971
FP RATE=0.0029
SENSITIVITY=0.9512

9.3 Support Vector Machines



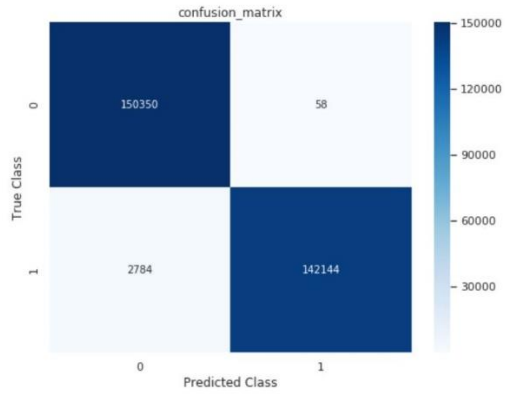
F1 SCORE=0.9697
PRECISION=0.9968
SPECIFICITY=0.9967
FP RATE=0.0033
SENSITIVITY=0.9440

9.4 Light Gradient Boosting



F1 SCORE=0.9919
PRECISION=0.9996
SPECIFICITY=0.9996
FP RATE=0.0004
SENSITIVITY=0.9843

9.5 Extreme Gradient Boosting



F1 SCORE=0.9906

PRECISION=0.9996

SPECIFICITY=0.996

FP RATE=0.0004

SENSITIVITY=0.9818

10. EVALUATION

The figures under the section AUROC plots and confusion matrix shows all the quantitative results that we obtained from our models. We can clearly see that the Ensemble models outperforms all the other models. Particularly, boosting was so effective on predicting whether the card transaction is fraudulent or not. The best model we obtained was Light Gradient Boosting Model which has a roc almost equal to 1. Although the other metrics like precision, sensitivity, specificity are similar amongst the models, the key parameter in determining the difference was roc. And light gradient boosting has the lowest False Positive rate which was our aim to reduce along with higher auroc score. This was possible only because of correcting the class imbalance which made our model more robust to new pattern. Visualization helped us to understand how data was changing over time and did not have much effect on the results. We were successful in achieving the main goal of benchmarking the discussed data mining algorithms and obtained results that were on par with the best results obtained so far.

11. CONTRIBUTION

Abishek Anbarasan – lgbm, xgbm, feature engineering, class imbalance, visualization, report writing

Mayur Deo – Logistic regression, SVM, Random Forests, plots, report writing

12. INSIGHTS

Boosting methods produced good quality results that revolved around distinct concepts. Logistic regression underperformed as it was not able to differentiate the two classes better. Usage of adversarial validation helped in making better predictions and obtained low False positive rate. Also, from the variable importance plot we can see that Transaction amount and time were one of the most important variables. Higher the transaction amount in peak time, higher the probability of fraudulent transaction. Also, few models detected vesta engineered features to be important, but we were not able to identify them because the real description was masked for privacy purposes. Also, the type of card that had most fraud transactions was visa card which is because they are more in number than the rest of the cards and another interesting fact is that the transactions done through mobile have most fraud transaction probability. It was also noted that IOS did have a less probability of fraud detection.

13. FUTURE WORK

1. We consider our next step in this project as the process of finding the location of the fraudster. I.e., “Zero in on the fraudster”. We will employ deep learning methodologies to detect the fraudulent transactions, we could build models using the information of fraudulent transaction and the transactions happened prior to the fraudulent transaction to recognize a pattern in the data and detect the closed location of the fraudster with % of confidence.
2. Our primary emphasis on this project is to detect fraudulent transactions. We can also use deeper algorithms to prevent such transactions. We can take steps to prevent fraudulent transactions by recognizing the pattern on the fraudulent transactions and take preventive measures by recognizing the pattern to automatically request change of card for the customer or request password or pin change to the customer who may be vulnerable to fraud.

14. REFERENCES

B, V. (2009). Machine learning techniques for fraud detection.

Hastie, T. (2009). *The Elements of Statistical Learning.* NY: Springer.

Kaggle. (n.d.). Retrieved from Kaggle: https://www.kaggle.com/datasets?utm_medium=paid&utm_source=google.com+search&utm_campaign=datasets&gclid=CjwKCAiA27LvBRB0EiwAPc8XWeuBLDuqpLmNg8wBv8yE_r77PTbVR-K4S3SU4FbwbffPzYh3Tlf4gxoCiJkQAvD_BwE

P. (2007). *Modeling and Simulation Design.* Natick,MA: AK Peter.

Varmedja, D., Karanovic, M., Sladojevic, S., Arsenovic, M., & Anderla, A. (20-22 March 2019). Credit card fraud detection. *IEEE*. Retrieved from https://ieeexplore.ieee.org/document/8717766