Credit Card Fraud Detection Using Machine Learning Algorithms

Anshika Gupta

Computer Science Department, The NorthCap University, Gurugram, India

anshikagupta399@gmail.com

Abstract:- Credit card fraud is a significant concern for financial institutions and consumers. While traditional fraud detection methods have been used in the past, machine learning models have gained popularity due to their ability to identify fraudulent activities with high accuracy. This paper aims to provide an overview of the current research on the use of machine learning algorithms for credit card fraud identification. It examines the different types of machine learning algorithms like logistic regression, naïve bayes, k-nearest neighbours and random forest classifier, the challenges associated with using these algorithms, and potential future research directions. Dataset of credit card transactions is sourced from European cardholders containing 284,807 transactions. The dataset is highly imbalanced. The work is implemented using Python. The paper concludes that machine learning techniques offer a promising approach for identifying and preventing credit card fraud, but there is still much work to be done to improve the accuracy and reliability of these methods.

---

1. Introduction

Credit card fraud is a major issue that impacts both consumers and financial organizations. Financial losses, credit score damage, and a lack of trust in the financial system can all arise from fraudulent activity. Financial institutions have utilised a range of fraud identification strategies to fight this issue, including rule-based systems, anomaly detection, and machine learning.

Because of their ability to learn from previous data and discover trends and anomalies that may suggest fraudulent conduct, machine learning (ML) techniques have risen in favour in recent years. Because these tools can quickly analyse vast amounts of data, financial institutions can detect fraud in real time.

The use of machine learning to detect credit card fraud is not a new concept. Researchers have been exploring the efficiency of various ML algorithms for this aim for several years. Decision trees, logistic regression, neural networks, and support vector machines are some of the most commonly used fraud identification methods. These algorithms use a number of methodologies based on previous data to classify transactions as fraudulent or non-fraudulent.

While ML has demonstrated encouraging results in detecting fraud, its application is not without hurdles. One of the most serious issues is the issue of skewed data. Fraudulent transactions are generally rare in the identification of credit card fraud, and the data is

frequently skewed, which might damage the algorithm's accuracy. Another issue is the requirement for continuous training of the algorithm to adjust to new types of fraud.
Despite these challenges, ML techniques for credit card fraud identification have significant potential for improving fraud identification and prevention. These techniques can detect fraud in real time, reducing losses for financial institutions and increasing customer trust. Furthermore, ML algorithms can continuously learn and adapt to new types of fraud, making them more effective than traditional rule-based systems. Finally, the use of ML techniques for credit card fraud identification is a promising area of research. This research paper aims to provide a comprehensive understanding of the current state of ML for credit card fraud identification, including the technique's challenges and limitations, as well as the potential for future research in this area.

2.Related Works

Sr. No.	Title and Author	Learning Method	Algorithms	Results
1.	**Title:-** Credit Card Fraud Detection -Machine Learning methods **Author:-**Dejan Varmedja, Mirjana Karanovic, Srdjan Sladojevic, Marko Arsenovic, Andras Anderla	Supervised	Logistic Regression (LR)	Accuracy:- 97.46% Precision:- 58.82% Recall:- 91.84%
			Naïve Bayes (NB)	Accuracy:- 99.23% Precision:- 16.17% Recall:- 82.65%
			Random Forest	Accuracy:- 99.96% Precision:- 96.38% Recall:- 81.63%
2.	**Title:-** Credit Card Fraud Detection using Machine Learning Algorithms **Author:-** Vaishnavi Nath Dornadulaa* , Geetha Sa	Supervised	Support Vector Machine (SVM)	Accuracy:- 99.87% Precision:- 76.81% MCC:- 0.5257
			LR	Accuracy:- 99.90% Precision:- 87.5% MCC:- 0.67766
			Decision Tree	Accuracy:- 99.94% Precision:- 88.54% MCC:- 83.56%
			Random Forest	Accuracy:- 99.94% Precision:- 93.10% MCC:- 82.68%
3.	**Title:-** Comparative Evaluation of Credit Card Fraud Detection Using Machine Learning Techniques **Author:-** Olawale Adepoju, Julius Wosowei, Shiwani lawte, Hemaint Jaiman	Supervised	SVM	Accuracy:- 97.53% Sensitivity:- 97.56% Specificity:- 97.53% Precision:-85.1%
			KNN	Accuracy:- 96.91% Sensitivity:- 89.36% Specificity:- 98.19% Precision:- 89.36%
			NB	Accuracy:- 95.99% Sensitivity:- 0 Specificity:- 1 Precision:- 1

			LR	Accuracy:- 99.074% Sensitivity:- 1 Specificity:- 98.92% Precision:- 93.61%
4.	**Title:-** Credit card fraud detection using Machine Learning Techniques: **Author:-** John O. Awoyemi, Adebayo O. Adetunmbi, Samuel A. Oluwadare	Supervised	NB	Accuracy:- 0.9752 Sensitivity:- 0.8210 Specificity:- 0.9754 Precision:- 0.0546 MCC:- +0.2080 Balanced Classification on Rate:- 0.8975
			KNN	Accuracy:- 0.9715 Sensitivity:- 0.8285 Specificity:- 1.0000 Precision:- 1.0000 MCC:- +0.8950 Balanced Classification on Rate:- 0.9143
			LR	Accuracy:- 0.3639 Sensitivity:- 0.7155 Specificity:- 0.2939 Precision:- 0.1678 MCC:- +0.0077 Balanced Classification on Rate:- 0.5047

Table 1

3.Experimental Set Up and Methodology

This section provides details about the dataset and the algorithms used for the classification of transactions into legit or fraud. Logistic regression(LR), NaïveBayes(NB), K-nearest neighbour(KNN), Random Forest(RF), and Support vector Machine(SVM) were the algorithms used to compile the results of this study. Data Pre-processing converts data into a usable format and categorises transactions as legitimate or fraudulent. In terms of analysis, the dataset has already been reduced using PCA (Principal Component Analysis). The dataset will be divided into two parts: training and testing data. The classifier algorithms will be fed processed data during the training phase. On the basis of the results, the accuracy of the algorithms used in the analysis will be assessed based upon True Positives(TP), True Negatives(TN) , False Positives(FP) , False Negatives(FN) , AUC of the ROC Curve.

3.1 Dataset

The dataset is sourced from Kaggle. The dataset contains credit card transactions with amount of transactions made by European cardholders in September 2013.The dataset is highly imbalanced towards normal transactions. It contains only numerical input values. The background information of these features cannot be disclosed due to privacy of the users.

Total Transactions	Total number of features	Fraudulent transactions	Percentage of fraudulent transaction in the dataset	Null values	Period over which data is collected	Year of the collected Dataset
284,807	30	492	0.172%	0	2 days	2013

Table2:- Dataset Description

Sr. No.	Feature	Description
1.	V1, V2, V3,……..V28	PCA applied features whose background information cannot be revealed due to privacy reasons
2.	Time	Time specified in seconds that elapses between the current transaction and the first transaction
3.	Amount	Amount of that particular transaction
4.	Class	Binary class 0 – not fraud 1 - fraud

Table 3:- Feature Description of the Dataset

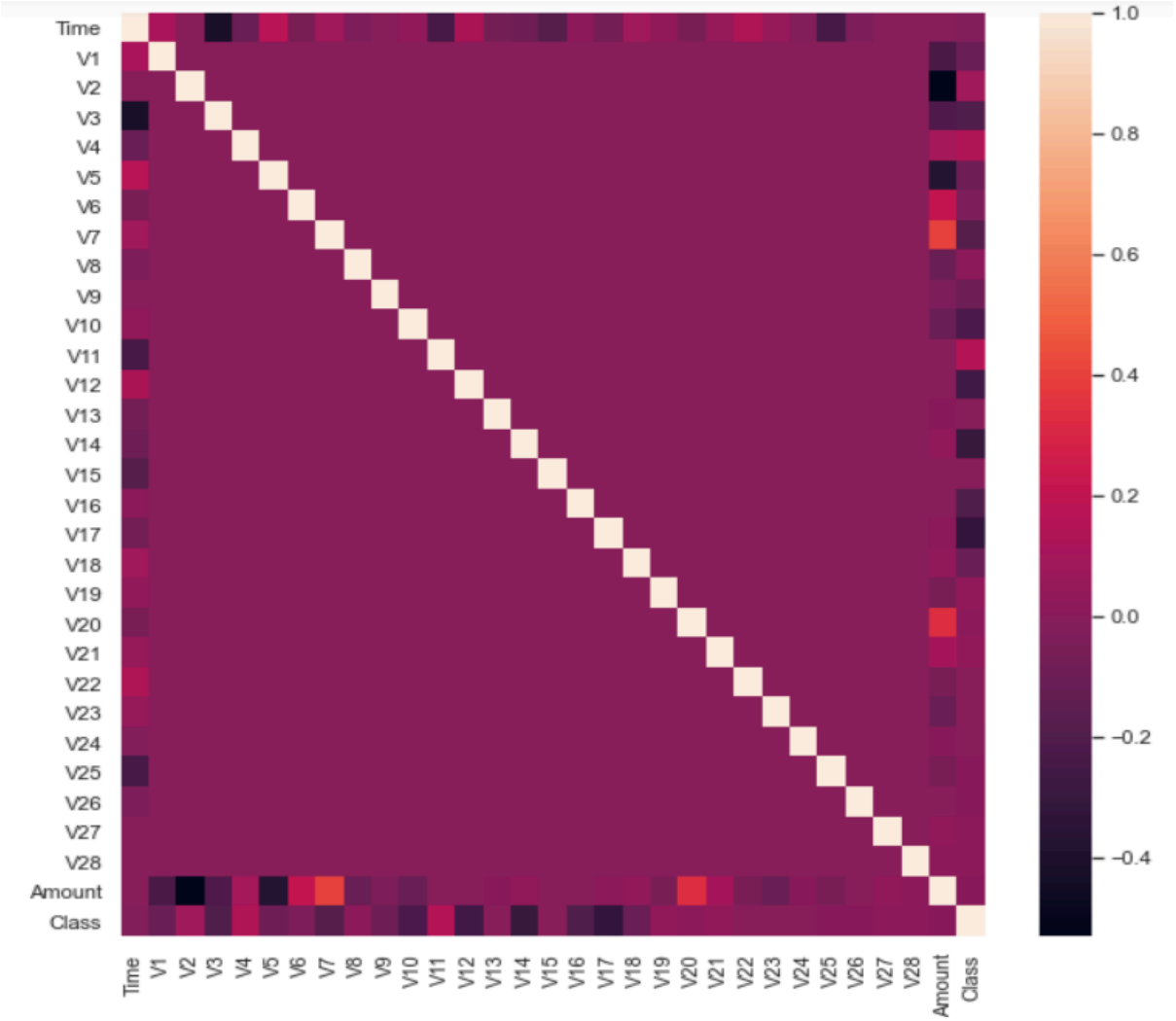

Figure 1:- Correlation Matrix for the attributes in the dataset

(both X and Y axis represent the attributes in the dataset)

3.1.1 Data Pre-processing

The dataset needs to be balanced to ensure that the model has enough exposure to both classes. This ensures that the model won't be biased towards the non-fraudulent values, as currently there are more than 280,000 entries with a class of 0, while there are only 492 fraudulent values.

To change the number of values in class 1, there are two possible methods: random under sampling (reduces the number of entries from the majority class) and synthetic oversampling (synthetically generating new entries for the minority class). For providing the model with as much data as possible, we will use a technique called Borderline-SMOTE (Borderline Synthetic Minority Oversampling Technique). The reason that Borderline-SMOTE is used as opposed to SMOTE is that it takes synthetically generates data in between different elements in the minority class, while ignoring outliers. Outliers can be defined as features which neighbour more majority points, than minority ones.

Borderline SMOTE (Synthetic Minority Over-sampling Technique) is a version of the SMOTE method, which is used to tackle class imbalance problems in ML.

SMOTE interpolates between minority class samples that already exist to produce synthetic minority class examples for binary classification. This enhances the functionality of ML models and balances the distribution of the classes.

Unfortunately, in some circumstances, the SMOTE algorithm can generate noisy synthetic samples that are not effective for enhancing the classification accuracy. Borderline SMOTE addresses this issue by only generating synthetic examples for the minority class that are on the borderline between the minority and majority classes.

3.1.2 Classifiers

1) Logistic Regression Classifier

A classification process called logistic regression is used to forecast binary outcomes, such as whether or not a consumer will buy a product, based on a collection of input data. It simulates the likelihood of a binary response variable (the dependent variable) in the presence of one or more predictor variables (i.e., the independent variables). The logistic function, which converts every real-valued input to a probability value between 0 and 1, serves as the foundation for the LR model. The logistic function has the following formula:

p=1/(1+e-z),

where p is the probability of a positive result, z is a linear combination of the input variables, and e is the natural logarithm's base. For calculating the linear combination z, the formula is:

z=0+1x1+2x2+...+nxn,

where beta 0 is the intercept, beta 1, beta 2, ..., beta n are the coefficients associated with the independent variables x1, x2, ..., xn, respectively.

According to this model, the coefficient values are estimated so that the anticipated probability and observed probabilities are as similar as possible. A maximum likelihood estimation method is used for this. The LR model employs a threshold value (often 0.5) to categorise data as positive or negative based on the projected probability in order to make predictions. The observation is classed as positive if the expected probability is higher than or equal to the threshold, and as negative if not. By utilising a one-vs-all or softmax technique, logistic regression can be expanded to solve multi-class classification issues.

2) Naïve Bayes Classifier

The Bayes theorem is the foundation of the Naive Bayes classifier, a classification technique. It is a probabilistic algorithm that determines, based on the presence or absence of specific features, the likelihood that an input belongs to a given class.

The NB classifier makes the assumption that each feature's existence or absence stands independently of any other. This so-called "naive" assumption makes it easier to calculate probabilities.

According to the Bayes theorem, the likelihood of a hypothesis (or class) given the evidence (or input and its features) is proportional to the likelihood of the evidence given the

hypothesis, multiplied by the prior probability of the hypothesis. Mathematically, it can be expressed as:

$P(hypothesis|evidence) = P(evidence|hypothesis) * P(hypothesis) / P(evidence)$

where P(hypothesis|evidence) is the posterior probability of the hypothesis given the evidence, P(evidence|hypothesis) is the likelihood of the evidence given the hypothesis, P(hypothesis) is the prior probability of the hypothesis, and P(evidence) is the probability of the evidence.

In the NB classifier, we assume that the probability of the evidence given the hypothesis can be calculated by multiplying the probabilities of each feature given the hypothesis. This can be expressed as:

$P(evidence|hypothesis) = P(feature1|hypothesis) * P(feature2|hypothesis) * ... * P(featuren|hypothesis)$

where feature 1, feature 2, ..., feature n are the features of the input.

We calculate the posterior probabilities of each class given the data in order to make a prediction, and we then pick the class with the highest probability. In other words, given the observed qualities, we select the class that is most likely.

The NB classifier is a quick and straightforward technique that performs well in many real-world situations, particularly when there are lots of features and little data. The approach may not work effectively if the characteristics are significantly linked, and the "naive" assumption may not always be true.

3) K- nearest neighbours Classifier

The supervised ML algorithm K-nearest neighbours (KNN) is used for classification and regression tasks. It is a sort of instance-based learning, which implies that it saves the training data instead of developing a model from it and then utilises it to predict the values of incoming data points. In order to estimate the class or value of an input data point based on the most prevalent class or average value among its k-nearest neighbours, KNN first determines the k nearest neighbours to the input data point based on some distance measure. A distance metric, such as the Euclidean distance, Manhattan distance, or cosine similarity, can be used to determine the separation between two data points.

The Euclidean distance (Dij) between two input vectors (Xi, Xj) is given by:

$D_{ij} = (\sum^{n}_{k=1} (X_{ik} - X_{jk})^2)^{0.5} \quad k=1,2,\ldots,n$

Since KNN is a non-parametric algorithm, it makes no assumptions about the distribution of the data at its core. It also has the benefit of being straightforward and simple to use. However, particularly for huge datasets, it can be computationally expensive. Also, the selection of k and the distance metric has a significant impact on how well KNN performs.

4) Random Forest Classifier

An ensemble learning approach called Random Forest Classifier (RFC) combines the predictions of various decision trees to increase the model's stability and accuracy. RFC's excellent accuracy, robustness, and capacity to handle highly dimensional data with intricate

interconnections have led to its widespread use in both academic research and industrial applications for classification and regression tasks.

Building a collection of decision trees utilising a set of randomly chosen features and training data is the main goal of RFC. This strategy aids in lowering overfitting and enhancing the model's generalisation capabilities. RFC aggregates all the trees' predictions at prediction time to produce a final prediction, which can either be the mean or the mode of the predicted classes (for classification problems).

5) Support Vector Machine

Support Vector Machines (SVMs) are a form of ML technique used for classification and regression analysis. They are often employed in supervised learning tasks where the goal is to predict the class of an input data point based on a set of training data.

The primary principle underlying SVMs is to determine the hyperplane that optimally separates the input data into two classes. The hyperplane is defined as a boundary that maximises the margin, which is the distance between the hyperplane and the nearest data points from each class. The closer the data points are to the hyperplane, the better the confidence of the categorization.

SVMs transform the input data into a higher dimensional space using a kernel function, making it simpler to analyse.

4.Proposed Hybrid Model for Credit Card Fraud Identification using ML

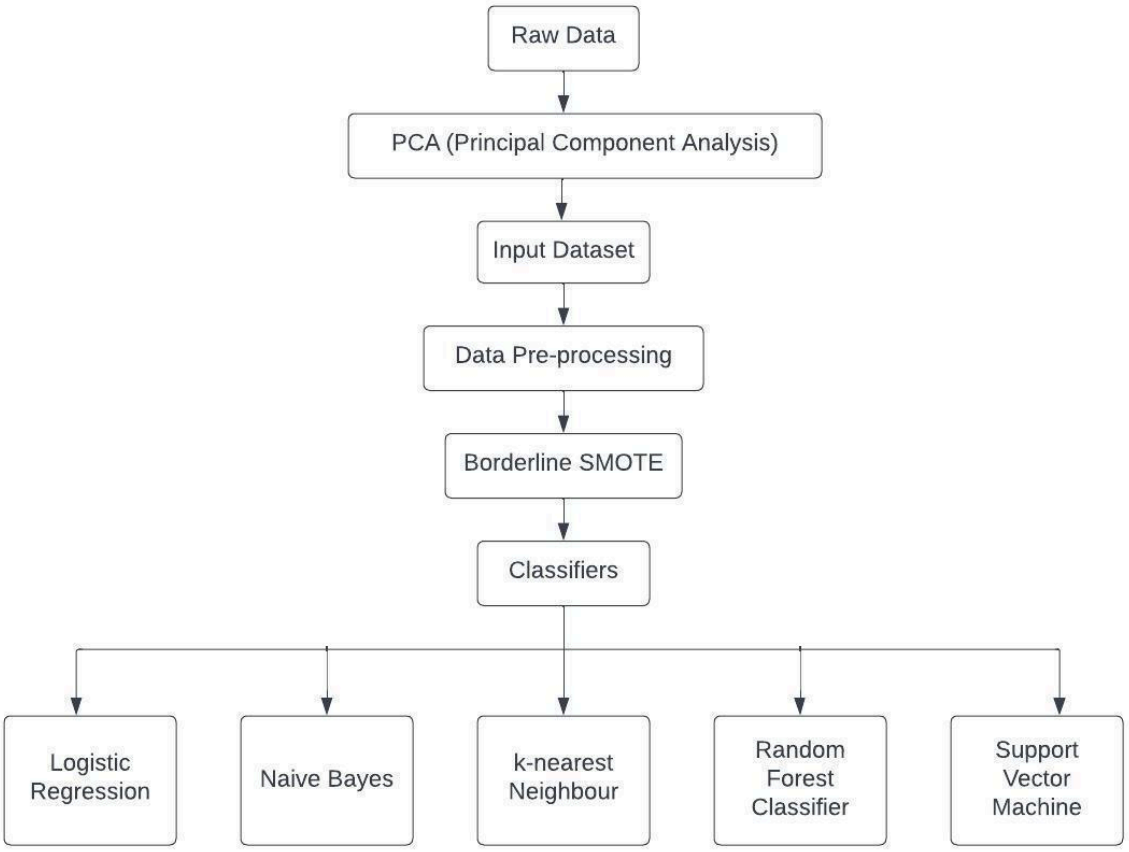

Figure-2

5.Evaluation Metrics

The evaluation criteria used for the ML algorithms used in this research are True Positive Rate (TPR), True Negative Rate (TNR), False Positive Rate (FPR), False Negative Rate (FNR), AUC Curve, ROC Curve respectively.

TPR = TP/P

TNR = TN/N

FPR = FP/N

FNR = FN/P

Where TP, TN, FP, FN are the number of cases classified rightly as positive, cases classified rightly as negative, cases classified wrongly as positive but are truly negative and cases classified wrongly as negative but are truly positive cases respectively and P and N are the total number of positive and negative cases under test.

The performance of the ML algorithms used in the experiment are evaluated based on Pearson's correlation coefficient, accuracy, sensitivity (recall), specificity , precision, F1 score, AUC score of ROC Curve. These performance metrics are used based on their relevance in evaluating imbalanced binary classification problems.

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

Sensitivity (Recall) = TP/(TP +FN)

Specificity = TN/(FP + TN)

Precision = TP/ (TP + FP)

F1 score = 2* Precision*Recall/ (Precision + Recall)

6.Result

The dataset is split into two sets two sets which is the training and testing set. 70% of the data is used for training and 30% is used for validating and testing phase.

Metrics	Classifiers				
	Logistic Regression	Naïve Bayes	k-nearest Neighbour	Random Forest	SVM
True Positive Rate	117	90	32	108	109
False Positive Rate	949	548	7	16	22
True Negative Rate	81614	84759	28427	82547	85285
False Negative Rate	19	46	15	28	27

Table-4

Metrics (*100)	Classifiers				
	Logistic Regression	Naïve Bayes	k-nearest Neighbour	Random Forest	SVM
Accuracy	98.829490	99.304799	99.922755	99.946795	99.942651
Sensitivity	86.029412	66.176470	68.085106	79.411765	80.147058
Specificity	98.850574	99.357614	99.975381	99.980620	99.974210
Precision	10.975610	14.106583	82.051282	87.096774	83.206106
Recall	86.029412	66.176470	68.085106	79.411765	80.147058
F1 Score	19.467554	23.255813	74.418604	83.076923	81.647940
AUC Score	95.360072	82.767042	84.030244	95.711247	90.060634

Table-5

1) Comparative Study

Type-I Error or False Positives: False Positives are the ones which are actually not fraud but the prediction said that they are fraud.

Type-II Error or False Negatives: False Negatives are the ones which are actually fraud but the system said that they are not.

Type II error is significantly more important than type I error as the aim is to reduce the number of false negative. Hence , recall is an important metric.

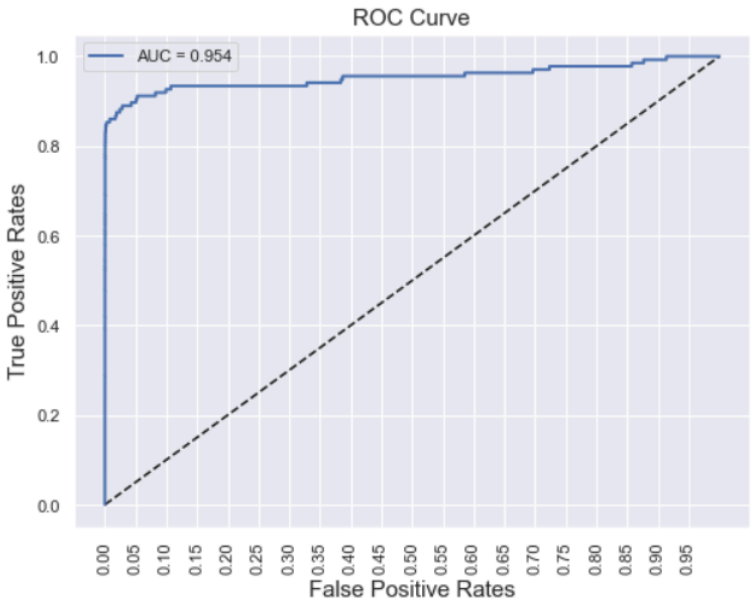

Figure 3:- ROC Curve of Logistic Regression

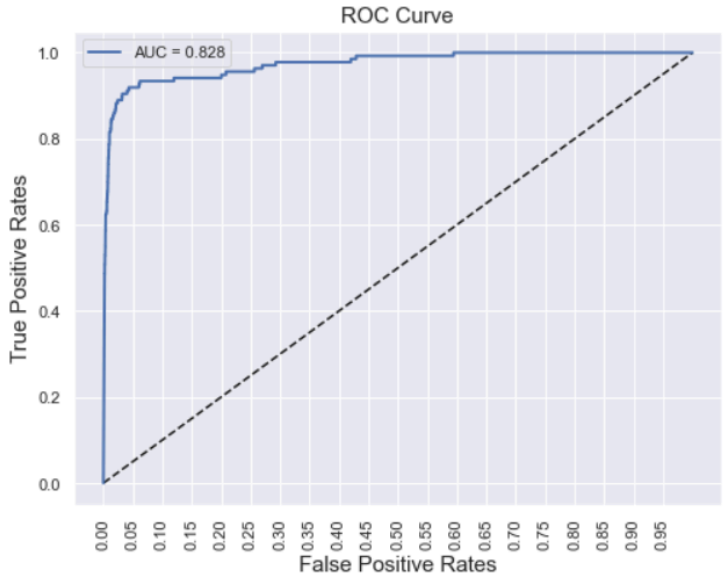

Figure 4:- ROC Curve of Naïve Bayes

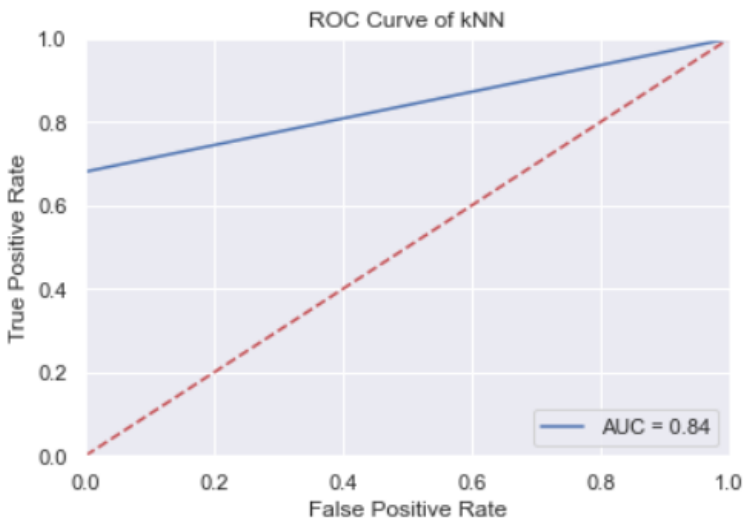

Figure 5:- ROC Curve of k- nearest neighbour

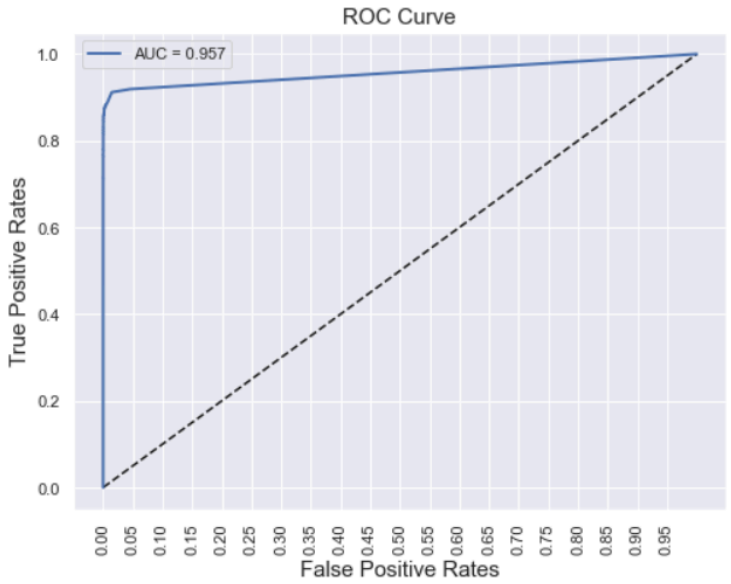

Figure 6:- ROC Curve of Random Forest

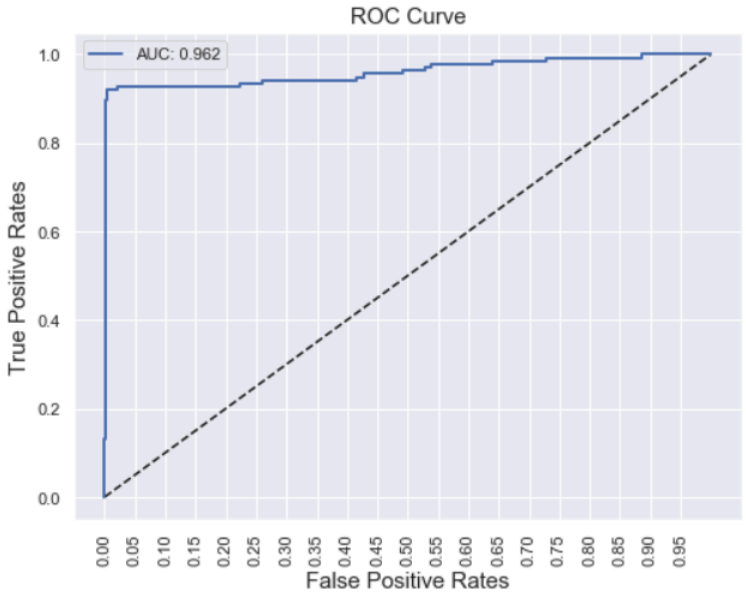

Figure 7:- ROC Curve of SVM

7.Conclusion

Five classifer models are used in this study which are Logistic regression, Naïve Bayes, k-nearest Neighbour, Random Forest, Support Vector Machine. 70% of the dataset is used for training and the other 30% is used for validation and testing. Precision, Sensitivity, Specificity, Accuracy, AUC, Recall are used to assess the performance of these classifiers. The dataset is highly imbalanced which is tackled by using Borderline SMOTE. It is observed that Logistic Regression was the most best in detecting credit card fraud. Based on this exploration, a credit card organization ought to consider executing a Logistic Regression algorithm that investigates the buy time to distinguish whether a credit card transaction is fraud.

8.References

Varmedja, Dejan, Mirjana Karanovic, Srdjan Sladojevic, Marko Arsenovic, and Andras Anderla. "Credit card fraud detection-machine learning methods." In *2019 18th International Symposium INFOTEH-JAHORINA (INFOTEH)*, pp. 1-5. IEEE, 2019.

Dornadula, Vaishnavi Nath, and Sa Geetha. "Credit card fraud detection using machine learning algorithms." *Procedia computer science* 165 (2019): 631-641.

Adepoju, Olawale, Julius Wosowei, and Hemaint Jaiman. "Comparative evaluation of credit card fraud detection using machine learning techniques." In *2019 Global Conference for Advancement in Technology (GCAT)*, pp. 1-6. IEEE, 2019.

Awoyemi, J. O., Adetunmbi, A. O., & Oluwadare, S. A. (2017, October). Credit card fraud detection using machine learning techniques: A comparative analysis. In *2017 international conference on computing networking and informatics (ICCNI)* (pp. 1-9). IEEE.