Credit Card Fraud Detection Using Different Machine Learning Classifier

Project Report

Submitted By

1. **Nibedita Roy, Roll- 1220CMSH-0040, Reg. No. 121-1212-0731-20**
2. **Shivam Das, Roll- 1220CMSH-0051, Reg. No. 121-1111-0742-20**
3. **Gourab Haldar, Roll- 1220CMSH-0035, Reg. No. 121-1112-0726-20**

Department of Computer Science
Gour Mahavidyalaya
Malda
August, 2023

BONAFIDE CERTIFICATE

This is to certify that this project report entitled **"Credit Card Fraud Detection Using Different Machine Learning Classifier"** submitted to Gour Mahavidyalaya, Malda, is a bonafide record of work done by "Nibedita Roy, Shivam Das and Gourab Haldar" under my supervision for the session of 2022-23.

Arijit Bhattacharya
Head, Assistant Professor
Department of Computer Science
Gour Mahavidyalaya
Malda

Declaration by Author(s)

This is to declare that this report has been written by us. No part of the report is plagiarized from other sources. All information included from other sources has been duly acknowledged. We aver that if any part of the report is found to be plagiarized, we are shall take full responsibility for it**.**

Nibedita Roy

Shivam Das

Gourab Haldar

Table of Contents

Credit Card Fraud Detection Using Different Machine Learning Classifier

1. Introduction

At this present time, worldwide we all are using credit cards for shopping or other online payments. Credit card transactions are electronic e-payments where cash payments are converted to e-payments. As credit card transactions increase, the scammers are trying new technology to steal data from the cardholders. They send deceptive websites, email to the victims. Through this website frauder collect victims' personal information such as card numbers, username, and password.

As per the report, By 2027 financial service providers are expected to take a $40 billion hit globally in credit card losses, a significant increase compared to $27.85 bn in 2018[16]. Our objective of the project is to minimise these financial losses, for which we have developed an expert system to overcome the problem. Although Credit Card fraud detection is an extremely difficult process due to scarcity of fraudulent transactions, which limits the power of recognizing the pattern for learning. To overcome this problem, different techniques were available in literature. Under-sampling and over-sampling are the most common technique. Among them SMOTE and Adasyn were utilized in our study to balance the dataset. Among the various Machine Learning(ML) techniques, we have utilized five standalone classifiers- to build the model Logistic Regression(LR), Support Vector Machine (SVM), K-Nearest Neighbors (KNN), Gaussian Naïve Bayes (GNB), and Decision Tree (DT). Apart from this, five ensemble classifiers also utilized in this study. These were Random Forest (RF), AdaBoost, XGBoost, LGBM, Extra Tree classifier.

1.1 What is Credit Card Fraud?

Credit card fraud is the unauthorized use of a payment tool, such as a credit or debit card to fraudulently obtain money. Detect and preventing fraudulent transactions using machine learning is becoming easier and more efficient. ML-based fraud detection solutions can track patterns and prevent abnormal transactions. Unsupervised machine learning methods use unlabeled data to find patterns and dependencies in the credit card fraud detection dataset, making it possible to group data samples by similarities without manual labeling. The challenge is to recognize fraudulent credit card transactions so that customers of credit card companies are not charged for items that they did not purchase. Financial institutions and businesses like e-commerce are taking firm steps to flag the fraudsters entering the system using various advanced machine learning technologies.

1.2 Impact of Credit Card fraud

Credit card fraud can negatively impact a person's credit if the deceitful activity appears on their credit reports. Once the fake transaction is detected and reported, the counterfeit transactions or accounts can be removed. In 2021, there were 1,862 data breaches, a 68% increase from 2020 and an all-time high. The number of people affected by data breaches decreased from 2020, with 293,927,708 people impacted in 2021.[18] Experts predict that credit card fraud will increase due to the Covid pandemic, and businesses often spend millions to protect themselves from fraud. While the Fair Credit Billing Act, Electronic Fund Transfer Act, and Truth in Lending Act are designed to protect consumers from card fraud, some experts say they are not enough to protect smaller businesses from chargebacks caused by fraudulent transactions.

2. Literature Review

Pozzolo et al.[1] provided some answers from the practitioner's perspective by focusing on three crucial issues: unbalancedness, nonstationarity and assessment. In their experiment, Random Forests clearly outperforms among other classifiers. As per the result, SMOTE was the best balancing technique for the datasets.

Chaudhary et al. [2] used a neural network based fraud detection system to train on a large sample of credit card account transactions which come from a credit card issuer. It shows much great accuracy and high processing speed in fraud detection but it is limited to one-network per customer.

Srivastava et al.[3] used classification techniques i.e neural networks, data mining, clustering methods etc. The good thing here was that HMM model which can detect the Fraud efficiently and provide accurate security.

Shen et al.[4] built a fraud detecting model and used three classification methods. The data mining techniques including neural networks, logistic regression and decision tree to the credit card fraud detection. Credit card issuers utilize this models to compare transaction information with previous transaction patterns.

Maes et al.[5] used two machine learning techniques : Artificial Neural Networks(ANN) and Bayesian Belief Networks(BNN).Fraud detection process was faster with ANN techniques but better results and their training period is shorter with BNN techniques.

Varmedja et al.[7] compared certain machine learning algorithms and the comparison was made and it was established that the Random Forest algorithm gives the best results. Feature selection and balancing of the dataset have shown to be extremely important in achieving significant results.

Tiwari et al.[9] used various fraud detection techniques that exists today like HMM, Decision Tree, Random Forests, Logistic Regression, K-Nearest Neighbors, Neural Networks etc. But none of them were competent enough to detect the fraud because all the techniques discussed so far give accurate results only when performed on a particular dataset and sometimes with some special features only. Among all the techniques discussed, it is found that Neural Networks detects frauds with high precision and performs best.

Maniraj et al.[15] used a data to fit into a model and Local Outlier Factor, Isolation Forest Algorithm outlier detection modules are applied on it. While the algorithm does reach over 99.6% accuracy, its precision remains only at 28% when a tenth of the data set is taken into consideration.

3. Methodology

This part discusses the methodology adopted in this study to classify the non-fraudulent transactions from the fraudulent transactions. Figure 1 shows the steps used in this work. However, before we discuss the different steps of the methodology used in this work, we first discussed the Raw Dataset.

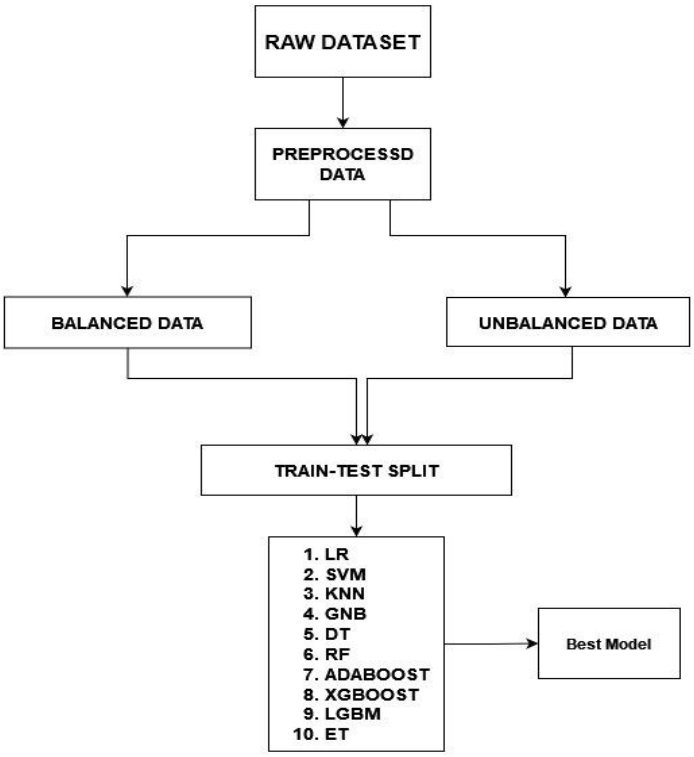
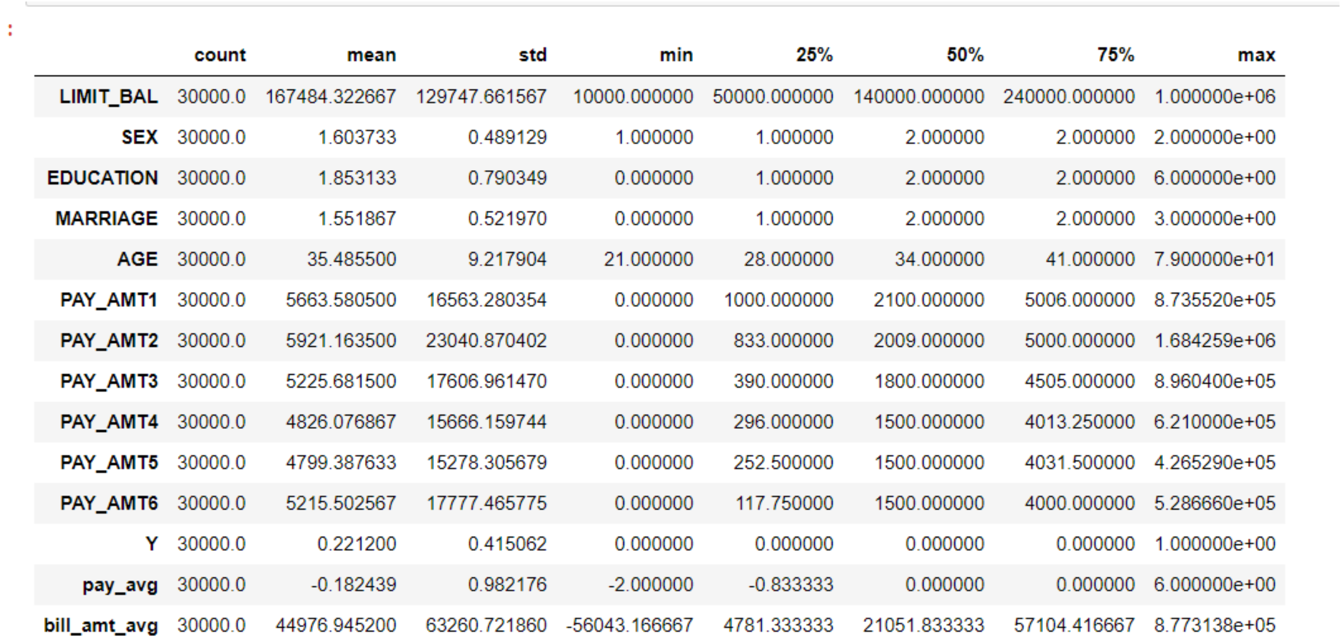


Fig. 1: Classification Methodology

3.1 Raw Dataset

The dataset for this project work was obtained from [17]. Originally it contains 30000 instances with 25 attributes. Among these attributes, few were very closely co-related. For better and easier insight, attributes with high correlation and also unnecessary attributes were deleted from the dataset. Following table shows the description of the dataset.



	count	mean	std	min	25%	50%	75%	max
LIMIT_BAL	30000.0	167484.322667	129747.661567	10000.000000	50000.000000	140000.000000	240000.000000	1.000000e+06
SEX	30000.0	1.603733	0.489129	1.000000	1.000000	2.000000	2.000000	2.000000e+00
EDUCATION	30000.0	1.853133	0.790349	0.000000	1.000000	2.000000	2.000000	6.000000e+00
MARRIAGE	30000.0	1.551867	0.521970	0.000000	1.000000	2.000000	2.000000	3.000000e+00
AGE	30000.0	35.485500	9.217904	21.000000	28.000000	34.000000	41.000000	7.900000e+01
PAY_AMT1	30000.0	5663.580500	16563.280354	0.000000	1000.000000	2100.000000	5006.000000	8.735520e+05
PAY_AMT2	30000.0	5921.163500	23040.870402	0.000000	833.000000	2009.000000	5000.000000	1.684259e+06
PAY_AMT3	30000.0	5225.681500	17606.961470	0.000000	390.000000	1800.000000	4505.000000	8.960400e+05
PAY_AMT4	30000.0	4826.076867	15666.159744	0.000000	296.000000	1500.000000	4013.250000	6.210000e+05
PAY_AMT5	30000.0	4799.387633	15278.305679	0.000000	252.500000	1500.000000	4031.500000	4.265290e+05
PAY_AMT6	30000.0	5215.502567	17777.465775	0.000000	117.750000	1500.000000	4000.000000	5.286660e+05
Y	30000.0	0.221200	0.415062	0.000000	0.000000	0.000000	0.000000	1.000000e+00
pay_avg	30000.0	-0.182439	0.982176	-2.000000	-0.833333	0.000000	0.000000	6.000000e+00
bill_amt_avg	30000.0	44976.945200	63260.721860	-56043.166667	4781.333333	21051.833333	57104.416667	8.773138e+05

3.2 Pre-Processed Data

Data preprocessing is a step in the data mining and data analysis process that takes raw data and transforms it into a format that can be understood and analyzed by computers and machine learning. Data preprocessing purposes are to clean and prepare the data to a spot that comprises more concise prejudice, checking for missing values, and more variation. Data contains both numerical and categorical, which means encoding the categorical data is necessary before using them for modeling. Outlier detection and removal was performed.

- **Feature Selection**: Feature selection is the process of deciding which variables (features, characteristics, categories, etc.) are most important to your analysis. These features will be used to build the ML model.

The techniques for feature selection in machine learning can be broadly classified into the following categories:-

Supervised Techniques: These techniques can be used for labeled data and to identify the relevant features for increasing the efficiency of supervised models like classification and regression. For Example- linear regression, decision tree, SVM, etc.

Unsupervised Techniques: These techniques can be used for unlabeled data. For Example- K-Means Clustering, Principal Component Analysis, Hierarchical Clustering, etc. From a taxonomic point of view, these techniques are classified into filter, wrapper, embedded, and hybrid methods.

Filter Methods :-

Filter methods pick up the intrinsic properties of the features measured via univariate statistics instead of cross-validation performance. These methods are faster and less computationally expensive than wrapper methods.

1. **Information Gain :-** Information gain calculates the reduction in entropy from the transformation of a dataset. It can be used for feature selection by evaluating the Information gain of each variable in the context of the target variable.
2. **Correlation Coefficient :-** Correlation is a measure of the linear relationship between 2 or more variables. Through correlation, we can predict one variable from the other. The logic behind using correlation for feature selection is that good variables correlate highly with the target.

   If two variables are correlated, we can predict one from the other. Therefore, if two features are correlated, the model only needs one, as the second does not add additional information. We will use the **Pearson Correlation** here.

   - **Pearson Correlation Coefficient :-** In statistics, PCC also known as Pearson's r. **PCC** is the most common way of measuring a linear correlation. It is a number between -1 and +1 that measures the strength and direction of the relationship between two variables.

     When one variable changes, the other variable changes in the same direction it means **Positive Correlation** and r is between 0 and 1.

     When there is no relationship between the variables it means **No Correlation** and r is 0.

     When one variable changes, the other variable changes in the Opposite Direction it means **Negative Correlation** and r is between 0 and -1.

The Pearson Correlation Coefficient also tells you whether the slope of the line of best fit is negative or positive. When the slope is negative, r is negative. When the slope is positive, r is positive. When r is 1 or -1, all the points fall exactly on the line of best fit. When r is greater than .5 or less than -.5, the points are close to the line of best fit. When r is between 0 and .3 or between 0 and -3, the points are far from the line of best fit.

The pcc is a good choice when all of the following are true:

A. Both variables are quantitative.

B. The variables are normally distributed.

C. The data have no outliers.

D. The relationship is linear.

E. No missing value.

Wrapper Methods :-

Wrappers require some method to search the space of all possible subsets of features, assessing their quality by learning and evaluating a classifier with that feature subset. The wrapper methods usually result in better predictive accuracy than filter methods.

1. **Forward Feature Selection:** This is an iterative method wherein we start with the performing features against the target features. Next, we select another variable that gives the best performance in combination with the first selected variable. This process continues until the preset criterion is achieved.
2. **Backward Feature Selection:** This method works exactly opposite to the Forward Feature Selection method. Here, we start with all the features available and build a model. Next, we use the variable from the model, which gives the best evaluation measure value. This process is continued until the preset criterion is achieved.

Embedded Methods: These methods encompass the benefits of both the wrapper and filter methods by including interactions of features but also maintaining reasonable computational costs. Embedded methods are iterative in the sense that takes care of each iteration of the model training process and carefully extract those features which contribute the most to the training for a particular iteration.

1. **Random Forest Importance:** Random Forests is a kind of Bagging Algorithm that aggregates a specified number of decision trees. The tree-based strategies used by random forests naturally rank by how well they improve the purity of the node, or in other words, a decrease in the impurity over all trees. Nodes with the greatest decrease in impurity happen at the start of the trees, while notes with the least decrease in impurity occur at the end of the trees. Thus, by pruning trees below a particular node, we can create a subset of the most important features.

3.2.1 Balanced Data : Our data set 30,000 rows and 14 columns. Then we checked outliers and removed the dataset. From this data set, using MLXTEND Feature selection,

3.2.2 Unbalanced Data: Our dataset was unbalanced because the fraud value 6636, and legitimized value 23364. Here, the fraud value minimum and legitimized value maximum so there occurred unbalancedness in dataset so we used oversampling technique (SMOTE) for balanced our data set.

3.3 Train Test split : Train test split is used for evaluating the performance of a machine learning algorithm. In this method, the dataset are divided two subsets.

Train Dataset : here, used to fit the machine learning model.

Test Dataset: here, used to evaluate the fit machine learning model.

In our dataset, Train:70%,Test:30%

3.4 Classifiers :-

1) Logistic Regression(LR):

This type of statistical model also called as logit model is often used for classification and predictive analytics. Logistic regression estimates the probability of an event occurring, such as voted or didn’t vote, based on a given dataset of independent variables. Since the outcome is a probability, the dependent variable is surrounded between 0 and 1. In logistic regression, a logit transformation is applied on the ratio between probabilities —that is, the probability of success divided by the probability of failure. This is also commonly known as the log odds, or the natural logarithm of odds, and this logistic function is represented.

2) Support Vector Machine(SVM):

Support Vector Machine (SVM) is a powerful machine learning classifier used for linear or nonlinear classification, regression, and even outlier detection tasks. SVMs can be used for a variety of tasks, such as text classification, image classification, spam detection, handwriting identification, gene expression analysis, face detection, and anomaly detection. SVMs are adaptable and efficient in a variety of applications because they can manage high-dimensional data and nonlinear relationships.

3) K-Nearest Neighbors(KNN):

The k-nearest neighbors algorithm, also known as KNN or k-NN, is a non-parametric, supervised learning classifier. Which is use to make classifications or predictions about the grouping of an individual data point. While it can be used for either regression or classification problems, it is typically used as a classification algorithm, working off the assumption that similar points can be found near one another.

4) Gaussian Naive Bayes(GNB):

Gaussian Naive Bayes (GNB) is a classification technique used in Machine Learning (ML) based on the probabilistic approach and Gaussian distribution. Gaussian Naive Bayes assumes that each parameter (also called features or predictors) has an independent capacity of predicting the output variable. The combination of the prediction for all parameters is the final prediction, that returns a probability of the dependent variable to be classified in each group. The final classification is assigned to the group with the higher probability.

5) Decision Tree(DT):

Decision tree is a supervised learning technique. It is used for classification and regression problems, but it is mainly used to solve classification problems. It is a graphical representation to solve all problems on a given condition.

6) Random Forest(RF):

Random forest is a supervised learning technique. It is used for high dimensionality dataset. In this classifier the model accuracy and decrease the over fitting problems. Random Classifier are used lot of tree to avoids the over fitting problem. It take minimum training time as compared to others algorithm.

7) Adaptive Boosting(AdaBoost):

AdaBoost is a iterative ensemble method. It is combine multiple poor classifier to get strong classifier. In this method, It is used weight of classifier and training the data sample. It is calculated accurate prediction in the dataset.

8) Extreme Gradient Boosting(XGBoost):

XG Boost algorithm are support to recognize our dataset and make better decision. It is a supervised algorithm. It is improved execution speed and model performance. It is designed for problems where we have bunch of training data and then we are classify our dataset.

9) Light Gradient Boosting Machine(LGBM):

Light GBM (Light Gradient Boosting Machine) is a popular open-source framework for gradient boosting. It is designed to handle large-scale datasets and performs faster than other popular gradient-boosting frameworks like XGBoost and CatBoost. Light GBM uses a gradient-based one-sided sampling method to split trees, which helps to reduce memory usage and improve accuracy.

10) Extra Tree(ET):

Extra trees (short for extremely randomized trees) is an ensemble supervised machine learning method that uses decision trees and is used by the Train Using AutoML tool. See Decision trees classification and regression algorithm for information about how decision trees work. This method is similar to random forests but can be faster.

Evaluation Metrices

Confusion Matrix: The confusion matrix is a matrix used to determine the performance of the classification models for a given set of test data.

n=Total Prediction	**Actual: No**	**Actual: Yes**
Predicted: No	True Negative	False Positive
Predicted: Yes	False Negative	True Positive

True positive(TP): Actual value positive and predicted value positive this is called true positive .

False Positive(FP): Actual value negative and predicted value positive this is called false positive.

False Negative(FN): Actual value positive and predicted value negative this is called false negative.

True Negative(TN): Actual value negative and predicted value negative this is called true negative.

Accuracy: Accuracy is used to calculate the Total number of performance of the model.

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

Specificity: Specificity is used to calculate the true negative value divided by the total number of negatives. It is called negative rate.

Specificity=TN/FN+TN

Cohen Kappa score: The Cohen-Kappa score can be used to measure the degree to which two or more raters can diagnose, evaluate, and rate behavior. A credible and dependable indicator of inter-rater agreement is Cohen’s Kappa. Both raw data and the values of the confusion matrix may be used to compute Cohen’s Kappa

Matthews Correlation Coefficient(MCC): Matthews correlation coefficient is calculated of the four values of the confusion matrix.

$$\mathrm{MCC} = \frac{\mathrm{TP} \cdot \mathrm{TN} - \mathrm{FP} \cdot \mathrm{FN}}{\sqrt{(\mathrm{TP} + \mathrm{FP})(\mathrm{TP} + \mathrm{FN})(\mathrm{TN} + \mathrm{FP})(\mathrm{TN} + \mathrm{FN})}}$$

Precision : precision is calculated of the number of true positive predictions divided by the total number of positive predictions. It is called positive predictive value.

Precision=TP/TP+FP

Recall: Recall is calculated of the number of positive predictions divided by the total number of positive predictions.

Recall=TP/TP+FN

F1 score:F1 score is a weighted average of precision and recall. As we know in precision and in recall there is false positive and false negative so it also considers both of them. F1 score is usually more useful than accuracy, especially if you have an uneven class distribution.

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

4.Result:-

		Without_Oversampling	**B_Adasyn**	**F_Adasyn**	**B_SMOTE**	**F_SMOTE**	**Best Result**
LR	**Accuracy**	0.7994	0.6118	0.6048	0.6408	0.6362	0.7994
	Specificity	0.9793	0.5867	0.6286	0.6347	0.6505	0.9793
	Cohen Kappa	0.1805	0.2232	0.2095	0.2813	0.2726	0.2813
	MCC	0.2485	0.2235	0.2097	0.2813	0.2729	0.2813
	F1 Score	0.2459	0.6229	0.595	0.6463	0.6345	0.6463
	Precision	0.6691	0.61	0.6098	0.6459	0.6473	0.6691
	Recall	0.1506	0.6364	0.5808	0.6466	0.6223	0.6466
	10-fold CV	0.8017	0.613	0.6015	0.6383	0.6336	0.8017
	5-fold CV	0.8017	0.6129	0.6015	0.6384	0.6338	0.8017
	RoC AUC Score	0.7085	0.6771	0.6543	0.7046	0.6975	0.7085

		Without_Oversampling	**B_Adasyn**	**F_Adasyn**	**B_SMOTE**	**F_SMOTE**	**Best Result**
SVM	**Accuracy**	0.8007	0.6633	0.6477	0.6853	0.6785	0.8007
	Specificity	0.9708	0.645	0.7454	0.7525	0.8287	0.9708
	Cohen Kappa	0.2128	0.3264	0.2952	0.3718	0.3598	0.3718
	MCC	0.2677	0.3266	0.301	0.3756	0.3778	0.3778
	F1 Score	0.2902	0.6709	0.6094	0.6667	0.6272	0.6709
	Precision	0.6407	0.6609	0.6834	0.7208	0.7622	0.7622
	Recall	**0.1876**	0.6813	0.5498	0.6201	0.5328	0.6813
	10-fold CV	**0.801**	0.6607	0.6454	0.6865	0.6784	0.801
	5-fold CV	**0.8011**	0.6593	0.6447	0.6858	0.6785	0.8011
	RoC AUC Score						0

		Without_Oversampling	**B_Adasyn**	**F_Adasyn**	**B_SMOTE**	**F_SMOTE**	**Best Result**
KNN	**Accuracy**	**0.7826**	0.7532	0.7199	0.7656	0.7465	0.7826
	Specificity	**0.9224**	0.6165	0.635	0.675	0.6663	0.9224
	Cohen Kappa	**0.2391**	0.5052	0.4399	0.5299	0.4917	0.5299
	MCC	**0.2538**	0.5244	0.4464	0.5379	0.4973	0.5379
	F1 Score	**0.3575**	0.7837	0.7418	0.7871	0.7674	0.7871
	Precision	**0.4989**	0.7016	0.6879	0.7302	0.718	0.7302
	Recall	**0.2785**	0.8876	0.8048	0.8535	0.8241	0.8876
	10-fold CV	**0.779**	0.6849	0.6689	0.7116	0.7055	0.779
	5-fold CV	**0.7788**	0.6774	0.6624	0.705	0.7002	0.7788
	RoC AUC Score	**0.6757**	0.8237	0.7808	0.8398	0.8188	0.8398

		Without_Oversampling	**B_Adasyn**	**F_Adasyn**	**B_SMOTE**	**F_SMOTE**	**Best Result**
GNB	**Accuracy**	**0.4453**	0.5591	0.548	0.5773	0.5637	0.5773
	Specificity	**0.3329**	0.2096	0.1705	0.2359	0.1807	0.3329
	Cohen Kappa	**0.101**	0.1134	0.0962	0.1459	0.1172	0.1459
	MCC	**0.1661**	0.1567	0.1468	0.1957	0.1772	0.1957
	F1 Score	**0.3997**	0.6736	0.6718	0.6857	0.6851	0.6857
	Precision	**0.2612**	0.5371	0.5273	0.5506	0.5405	0.5506
	Recall	**0.8504**	0.9031	0.9256	0.9085	0.9352	0.9352
	10-fold CV	**0.4326**	0.5646	0.548	0.5798	0.5625	0.5798
	5-fold CV	**0.4327**	0.5645	0.5481	0.58	0.5625	0.58
	RoC AUC Score	**0.6921**	0.6622	0.6483	0.6969	0.6865	0.6969

		Without_Oversampling	**B_Adasyn**	**F_Adasyn**	**B_SMOTE**	**F_SMOTE**	**Best Result**
DT	**Accuracy**	**0.7085**	0.7287	0.6937	0.7438	0.7197	0.7438
	Specificity	**0.7989**	0.703	0.6812	0.7201	0.7019	0.7989
	Cohen Kappa	**0.1747**	0.4571	0.3873	0.4872	0.439	0.4872
	MCC	**0.1751**	0.4576	0.3874	0.4875	0.4391	0.4875
	F1 Score	**0.363**	0.7368	0.6973	0.7523	0.7274	0.7523
	Precision	**0.3454**	0.7205	0.6888	0.7385	0.7181	0.7385
	Recall	**0.3826**	0.7539	0.7061	0.7667	0.7369	0.7667
	10-fold CV	**0.7172**	0.6763	0.6536	0.6989	0.6804	0.7172
	5-fold CV	**0.7165**	0.6709	0.6496	0.6933	0.6739	0.7165
	RoC AUC Score	**0.5909**	0.7325	0.6943	0.7464	0.7214	0.7464

		Without_Oversampling	**B_Adasyn**	**F_Adasyn**	**B_SMOTE**	**F_SMOTE**	**Best Result**
RF	**Accuracy**	**0.8033**	0.8177	0.7732	0.8237	0.7876	0.8237
	Specificity	**0.95**	0.8081	0.7556	0.8303	0.7834	0.95
	Cohen Kappa	**0.2796**	0.6354	0.5464	0.6473	0.5751	0.6473
	MCC	**0.3103**	0.6355	0.5467	0.6474	0.5751	0.6474
	F1 Score	**0.3774**	0.8205	0.7771	0.8247	0.791	0.8247
	Precision	**0.6037**	0.8141	0.7638	0.8323	0.7902	0.8323
	Recall	**0.2745**	0.8271	0.7908	0.8172	0.7917	0.8271
	10-fold CV	**0.8028**	0.762	0.7241	0.7759	0.7432	0.8028
	5-fold CV	**0.8026**	0.7566	0.719	0.7707	0.7299	0.8026
	RoC AUC Score	**0.7476**	0.8934	0.842	0.9009	0.8633	0.9009

		Without_Oversampling	**B_Adasyn**	**F_Adasyn**	**B_SMOTE**	**F_SMOTE**	**Best Result**
ADABOOST	**Accuracy**	**0.8046**	0.6795	0.66	0.7061	0.7044	0.8046
	Specificity	**0.9526**	0.7141	0.7447	0.7547	0.7651	0.9526
	Cohen Kappa	**0.2797**	0.3593	0.3198	0.4129	0.4098	0.4129
	MCC	**0.3128**	0.3603	0.3245	0.4151	0.4132	0.4151
	F1 Score	**0.3755**	0.6699	0.6283	0.6946	0.6891	0.6946
	Precision	**0.6134**	0.6963	0.6925	0.7346	0.7391	0.7391
	Recall	**0.2706**	0.6454	0.5751	0.6587	0.6454	0.6587
	10-fold CV	**0.8013**	0.6763	0.6634	0.7028	0.702	0.8013
	5-fold CV	**0.8011**	0.6753	0.663	0.7033	0.701	0.8011
	RoC AUC Score	**0.7557**	0.7495	0.7243	0.7817	0.7735	0.7817

		Without_Oversampling	**B_Adasyn**	**F_Adasyn**	**B_SMOTE**	**F_SMOTE**	**Best Result**
XGBOOST	**Accuracy**	**0.7986**	0.7981	0.7279	0.8078	0.7476	0.8078
	Specificity	**0.9414**	0.8003	0.7332	0.8267	0.7605	0.9414
	Cohen Kappa	**0.2754**	0.5962	0.4558	0.6158	0.4953	0.6158
	MCC	**0.2998**	0.5962	0.4558	0.6163	0.4956	0.6163
	F1 Score	**0.3796**	0.7989	0.7264	0.8065	0.7472	0.8065
	Precision	**0.5735**	0.8019	0.7302	0.8244	0.7598	0.8244
	Recall	**0.2836**	0.7959	0.7225	0.7894	0.755	0.7959
	10-fold CV	**0.7968**	0.7756	0.6978	0.7866	0.7241	0.7968
	5-fold CV	**0.7948**	0.772	0.6969	0.7807	0.7231	0.7948
	RoC AUC Score	**0.7478**	0.8846	0.8045	0.8903	0.8281	0.8903

		Without_Oversampling	**B_Adasyn**	**F_Adasyn**	**B_SMOTE**	**F_SMOTE**	**Best Result**
LGBM	**Accuracy**	**0.8091**	0.7116	0.7803	0.7876	0.7301	0.8091
	Specificity	**0.956**	0.7307	0.7755	0.8123	0.763	0.956
	Cohen Kappa	**0.2957**	0.4231	0.5605	0.5753	0.7606	0.7606
	MCC	**0.3311**	0.4234	0.5605	0.5762	0.4619	0.5762
	F1 Score	**0.3889**	0.7048	0.7826	0.7848	0.7241	0.7848
	Precision	**0.6381**	0.7198	0.7803	0.8074	0.7523	0.8074
	Recall	**0.2797**	0.6923	0.7849	0.7634	0.698	0.7849
	10-fold CV	**0.8045**	0.694	0.7582	0.7658	0.722	0.8045
	5-fold CV	**0.803**	0.6928	0.7574	0.7648	0.7203	0.803
	RoC AUC Score	**0.7633**	0.7875	0.8656	0.8727	0.8123	0.8727

		Without_Oversampling	**B_Adasyn**	**F_Adasyn**	**B_SMOTE**	**F_SMOTE**	**Best Result**
ET	**Accuracy**	**0.799**	0.8457	0.7827	0.8471	0.8004	0.8471
	Specificity	**0.9526**	0.8206	0.7634	0.8421	0.7856	0.9526
	Cohen Kappa	**0.2502**	0.6913	0.5653	0.694	0.6005	0.694
	MCC	**0.2845**	0.6921	0.5657	0.694	0.6007	0.694
	F1 Score	**0.3461**	0.8504	0.7867	0.8497	0.8055	0.8504
	Precision	**0.5896**	0.8313	0.7721	0.8476	0.7966	0.8476
	Recall	**0.245**	0.8704	0.8018	0.8518	0.8147	0.8704
	10-fold CV	**0.7995**	0.7812	0.7245	0.7912	0.7634	0.7995
	5-fold CV	**0.7993**	0.7727	0.7186	0.7835	0.7573	0.7993
	RoC AUC Score	**0.7377**	0.9209	0.8518	0.9214	0.8704	0.9214

In this chapter, we present the summary result along with the limitation and future research direction.

5. Conclusion

In term of Accuracy, Extra Tree Classifier outperforms the other classifiers in Boruta and Smote based model by 84.71%. In term of Specificity, Logistic Regression Classifier outperforms the other classifiers in Without_oversampling based model by 97.93%. Cohen Kappa score was highest in Light Gardient Boosting Machine Classifier when utilized with Sequential Feature selection with Smote by 76.06%. In terms of MCC, the Extra Tree Classifier outperforms the other classifiers when Boruta based Feature selector with Smote was utilized by 69.40%. F1 Score was recorded highest for Extra Tree Classifier when employed with Boruta based feature selector with Adasyn by 85.04%. In term of Precision, Extra Tree Classifier outperforms the other classifiers in Boruta and Smote based model by 84.76%. In term of Recall, Gaussian Naive Bayes Classifier outperforms the other classifiers in Sequential Feature selection model by 93.52%. 10-fold CV was highest in Light Gradient Boosting Machine Classifier when utilized without_oversampling by 80.45%. 5-fold CV was highest in Light Gradient Boosting Machine Classifier when utilized without_oversampling by 80.30%. Roc AUC score was highest in Extra Tree Classifier when utilized with Boruta based Feature selector with Smote by 92.14%.

References :-

[1] Dal Pozzolo, A., Caelen, O., Le Borgne, Y. A., Waterschoot, S., & Bontempi, G. (2014). Learned lessons in credit card fraud detection from a practitioner perspective. *Expert systems with applications*, *41*(10), 4915-4928.

[2] Chaudhary, K., Yadav, J., & Mallick, B. (2012). A review of fraud detection techniques: Credit card. *International Journal of Computer Applications*, *45*(1), 39-44.

[3] Srivastava, A., Kundu, A., Sural, S., & Majumdar, A. (2008). Credit card fraud detection using the hidden Markov model. *IEEE Transactions on dependable and secure computing*, *5*(1), 37-48.

[4] Shen, A., Tong, R., & Deng, Y. (2007, June). Application of classification models on credit card fraud detection. In *2007 International conference on service systems and service management* (pp. 1-4). IEEE.

[5] Maes, S., Tuyls, K., Vanschoenwinkel, B., & Manderick, B. (2002, January). Credit card fraud detection using Bayesian and neural networks. In *Proceedings of the 1st international naiso congress on neuro fuzzy technologies* (Vol. 261, p. 270).

[6] Chan, P. K., Fan, W., Prodromidis, A. L., & Stolfo, S. J. (1999). Distributed data mining in credit card fraud detection. *IEEE Intelligent Systems and Their Applications*, *14*(6), 67-74.

[7] Varmedja, D., Karanovic, M., Sladojevic, S., Arsenovic, M., & Anderla, A. (2019, March). Credit card fraud detection-machine learning methods. In *2019 18th International Symposium INFOTEH-JAHORINA (INFOTEH)* (pp. 1-5). IEEE.

[8] Patidar, R., & Sharma, L. (2011). Credit card fraud detection using neural networks. *International Journal of Soft Computing and Engineering (IJSCE)*, *1*(32-38).

[9] Tiwari, P., Mehta, S., Sakhuja, N., Kumar, J., & Singh, A. K. (2021). Credit card fraud detection using machine learning: a study. *arXiv preprint arXiv:2108.10005*.

[10] Bahnsen, A. C., Aouada, D., Stojanovic, A., & Ottersten, B. (2016). Feature engineering strategies for credit card fraud detection. *Expert Systems with Applications*, *51*, 134-142.

[11] Dornadula, V. N., & Geetha, S. (2019). Credit card fraud detection using machine learning algorithms. *Procedia computer science*, *165*, 631-641.

[12] Bin Sulaiman, R., Schetinin, V., & Sant, P. (2022). Review of machine learning approach on credit card fraud detection. *Human-Centric Intelligent Systems*, *2*(1-2), 55-68.

[13] Carcillo, F., Le Borgne, Y. A., Caelen, O., Kessaci, Y., Oblé, F., & Bontempi, G. (2021). Combining unsupervised and supervised learning in credit card fraud detection. *Information sciences*, *557*, 317-331.

[14] Thennakoon, A., Bhagyani, C., Premadasa, S., Mihiranga, S., & Kuruwitaarachchi, N. (2019, January). Real-time credit card fraud detection using machine learning. In *2019 9th International Conference on Cloud Computing, Data Science & Engineering (Confluence)* (pp. 488-493). IEEE.

[15] Maniraj, S. P., Saini, A., Ahmed, S., & Sarkar, S. (2019). Credit card fraud detection using machine learning and data science. *International Journal of Engineering Research*, *8*(9), 110-115

[16] https://www.altexsoft.com/blog/credit-card-fraud-detection/

[17] Yeh,I-Cheng. (2016). default of credit card clients. UCI Machine Learning Repository. https://doi.org/10.24432/C55S3H.

[18] https://www.fool.com/the-ascent/research/identity-theft-credit-card-fraud-statistics/