ISSN 2278-3091
Volume 10, No.3, May - June 2021
International Journal of Advanced Trends in Computer Science and Engineering
Available Online at http://www.warse.org/IJATCSE/static/pdf/file/ijatcse801032021.pdf
https://doi.org/10.30534/ijatcse/2021/811032021

Credit Card Fraud Detection using Imbalance Resampling Method with Feature Selection

Kajal[1], Dr. Kamaljit Kaur[2]
[1]MTech, Student, Department of Computer Engineering and Technology,
Guru Nanak Dev University Amritsar, India, kajal.kaler96@gmail.com
[2]Assistant Professor, Department of Computer Engineering and Technology,
Guru Nanak Dev University Amritsar, India, kamaljit.dcse@gndu.ac.in

ABSTRACT

Many fraud transactions exist in the online world that affects various financial institutions but Credit Card Fraud transaction is the most occurring problem in the world. Credit Card fraud is the situation in which fraudsters misuse credit cards for illegal purposes. Hence, detection of fraudulent transactions is essential. Several researchers have worked on detecting fraud transactions and also provide solutions whose surveys are given in this paper. This study makes a major contribution to research on the detection of Credit Card fraud transactions through Machine Learning Algorithms such as Decision Tree and Naive Bayes. The data have been selected from Kaggle and categorize into training (80%) and testing (20%) data. The whole experiment was performed on the Jupyter Notebook tool for which the Anaconda Navigator has been installed. The Heatmap is used for visualization and colorfully represents the data. The main aim of this work is to balance the dataset with Near-Miss Under-sampling Method. The information gain method is applied for feature selection. The best algorithm founded in this paper is Decision Tree with 97% accuracy as compared to Naïve Bayes with 90%. The results are achieved based on Accuracy, Recall, Precision, and F1-score. We have also shown the ROC Curve and Precision-Recall Curve of the algorithm in this paper.

Key words: Credit Card Fraud Detection, Near - Miss, Information Gain, Classification Methods.

1. INTRODUCTION

The dependency on fast-growing technology lead to the online payment and credit card is hugely needed for it [13]. The extreme usage of credit cards leads to credit card fraud detection. Credit card fraud detection is recognized as a vital issue that affects various organizations and the economy of a Country. Every year several new cases came across the world. Due to the lack of high-security systems, credit cards became the most common fraud issue globally [14]. Hence, the detection of credit cards is quite challenging.

Credit card fraud is the misuse of credit cards when banks and cardholders did not know that credit card is being misused by another person [4]. Fraudsters use those cards which are canceled, lost, or stolen and obtain them for their purpose.

The main aim behind credit card fraud is to buy goods and services without knowing, or to do something illegal. Fraudsters steal the PIN code or account number rather than stealing physical Credit cards from the individual and do illegal transactions.

Hence, by detection, we can check whether the transactions made by them are real or fraudulent. The figure 1 shows the process of credit card fraud detection.

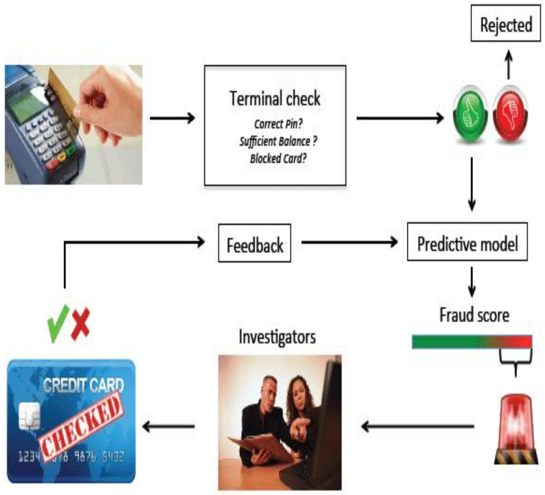


Figure 1: Credit Card Fraud Detection

Generally, Credit Card fraud is separated into the below categories:
1. Application Fraud: Application Fraud is a type in which a cardholder opens an account with stolen or fake papers in the name of another person [6].
2. CNP (Card Not present) fraud: Nowadays, it is the most common fraud. CNP (Card Not Present) fraud takes place when fraudsters did not get a Credit Card physically. This fraud occurs through online hacking.
3. Account Takeover: It is a fraud in which the fraudster completely takes over the control of the cardholder's account which included Credit Card, Bank, Email, etc.
4. Offline Fraud: It is a fraud that happened when the Credit Card is lost or stolen and can be used to pay for goods and services. This type of fraud is usually done by pick-pockets in busy crowds. [6].

2. LITERATURE SURVEY

Various research studies have already been done on credit card fraud detection. The literature survey for some of the researcher papers is given below:

Sara Makki et al [4] have proposed the model that faces the imbalanced dataset problem. To tackle this problem, they compared the performance of different eight machine algorithms out of which Logistic Regression (LR), C5.0 decision tree algorithm, Support Vector Machine (SVM), and ANN with 96% accuracy are the best techniques based on Accuracy, Sensitivity, and AUPRC.

Kuldeep Randhawa et al [3] used standard models first which include Naïve Bayes (NB) and Support Vector Machine (SVM). Then, they applied a Hybrid approach that utilizes Adaboost and Majority Voting. Real-World Dataset has been used in this paper. Majority voting is considered as the best technique to predict Credit Card Fraud Detection with an MCC score of 0.942 when 30% noise is an increase in the dataset.

Altyeb Altaher Taha et al [2] have proposed an approach using Light Gradient Boosting Machine (OLightGBM). The Two Real-World Datasets have been used and the comparison of machine learning algorithms such as Random Forest (RF), Logistic Regression, K-NN, Decision Tree and Naïve Bayes obtained better performance based on Accuracy (98.40%), AUC (92.88%), Precision (97.34%) and F1 score (56.95%).

Alex G.C. de Sa et al [8] used a customized Bayesian Network Classifier (BNC) algorithm for fraud detections. This algorithm was outperformed by Hyper-Heurist Evolutionary Algorithm (HHEA). The Dataset was obtained from PagSeguro. Results indicate that BNC is considered as best algorithm in terms of F1 score and Economic efficiency with 72.64%.

Ajeet Singh et al [9] developed a model using J48, Decision Tree, Adaboost, Random Forest, Naïve Bayes, and PART algorithms which are evaluated on German Credit Card Dataset. J48 and PART accuracy have been improved with the help of Filter and Wrapper Methods. Random Forest has achieved the highest accuracy of 76.4% with the filter method and without the feature selection method. Naïve Bayes has attained an accuracy of 75% with the wrapper method.

Pallavi Kulkarni et al [15] presented a model that faces the problem of incremental learning for the detection of Credit Card fraud. They balanced the data by using the bagging method. Data is selected from the non-stationary environment which leads to the various drifts.

Priyanka Kumari et al [12] have compared the ensemble classifier such as Random Forest, Voting with few single classifiers such as MLP, SVM, and K-NN. The three different datasets have been taken. They are German, Australian, and Bank-data CSV datasets. The imbalanced dataset is balanced using SMOTE method.

John O. Awoyemi et al [16] reported Machine Learning algorithms such as Naïve Bayes, K-NN, and Logistic Regression. Dataset is taken from European Cardholder. The Hybrid approach of Undersampling and Oversampling is performed on unbalanced data. Results indicate that K-NN has achieved better accuracy than Logistic Regression and Naïve Bayes.

P.Ragha Vardhani et al [14] applied Condensed Nearest Neighbor (CNN) data mining method to predict credit card frauds. They have used the concept of Nearest Neighbor classification in their proposed work.

Dejan Varmedja et al [13] used Logistic Regression, Random Forest, Naïve Bayes, and Multilayer Perceptron on Credit Card Fraud dataset in their proposed model. SMOTE Resampling method is applied to the imbalanced dataset. The feature selection technique is done on the dataset. Random Forest obtained better accuracy and hence it is used for detecting credit card frauds.

Table 1: Comparison Table with previous Researchers

References	Year	Dataset	Methodology	Merits	Demerits	Accuracy Achieved
Pallavi Kulkarni et al [1]	2016	German Credit Card dataset	Logistic Regression	Maintained efficiency	Model is complex	_
Kang Fu et al [2]	2016	Real-world Dataset	CNN	Recognize complicated fraud patterns	Dataset was highly imbalanced	_
Kuldeep Randhawa et al [3]	2017	European Cardholder	Majority Voting+ Adaboost	Obtaining best result using MCC metric	MV is unstable in the absence of noise	EC=95%
John O. Awoyemi et al [4]	2017	European Cardholder	Undersampling + oversampling	Efficiently classify the best attributes	It did not work on a small dataset	K-NN= 97.9%
Alex G.C. de Sa et al [5]	2018	PagSeguro transactions data	Bayesian network classifier (BNC)	Increase the accuracy and economic efficiency	Results are obtained only in terms of F1 metric	_
Sara Makki et al [6]	2019	Credit Card Fraud labeled data	Random Over Sampling	This approach is used to resolve the imbalance of data	Increase the quantity of false alarm	SVM=96%
P.Ragha Vardhani et al [7]	2019	European Cardholder	Condensed Nearest Neighbor(CNN)	Minimized the no. of attributes, Improve the Processing time	Computational Complexity is not minimized.	_
Dejan Varmedja et al [8]	2019	European Cardholder	Smote	Selects the best features and reduce the overfitting	High computational time	RF=99.96%
Priyanka Kumari et al [9]	2019	German Credit dataset	Naïve Bayes	Enhance the performance of the model	Reduce the accuracy with categorical attributes	NB=90.61%
Ajeet Singh et al [10]	2019	German Credit dataset	Filter Method	Increase the accuracy, reduce the classification time.	Specificity and precision rate are very low.	RF=76.4%
Yvan Lucas et al [11]	2019	Real-world dataset	Random Forest	Handles the missing values	HMM-based attributes could not be calculated in few transactions	_
Fabrizio Carcillo et al [12]	2019	European Cardholder	Supervised and Unsupervised techniques	A hybrid approach is efficient	Not improvements in terms of Precision	_
Fayaz- Itoo et al [13]	2020	European Cardholder	Random Over Sampling	Improve the accuracy and classification time	The same data is missing in Random under Sampling	LR=95.9%
Altyeb Altaher Taha et al [14]	2020	European Cardholder	Light gradient Boosting (LGB)	LGB provides faster training speed	Recall and F1 score is very low	LGB=98.40%
Naoufal Rtayli et al [15]	2020	European Cardholder	SVM recursive feature elimination, Hyperparameter optimization	Reduce the imbalance of data	It did not work on a small dataset	Hybrid Approach=99%
Fatima Zohra et al [16]	2020	European Cardholder	multilayer Perceptron and Extreme Learning Machine	Handle the imbalanced data and improve the accuracy.	MLP and ELM have difficulty to learn	MLP=97.84%

3. PROBLEM DEFINITION

In previous Literature reviews, different challenges are described but Class imbalance was the biggest problem of the dataset. Class imbalance is a problem in which the proportion of genuine transactions is greater than the fraud transactions. Several researchers have already worked on the imbalanced issue and many solutions have also been provided by using Machine Learning Algorithm.

Our purpose is to find an efficient solution that tackles class imbalance problems based on various parameters such as Accuracy, Precision, Recall, and F1- score.

3.1. Objectives

The main objectives of this research work are:

1. The current study aims to balance the dataset then extract the features from a balanced dataset and comparing the two different models such as Naïve Bayes and Decision Tree.
2. The near-Miss Under-sampling method aims to handle the imbalanced information.
3. There is no loss of important information by this method.
4. Information Gain Feature Selection method extracts useful features for the model.
5. Improves the following parameters:
   - Accuracy
   - Precision
   - Recall
   - F1-Score

4. PROPOSED METHODOLOGY

This section examines the research methodology of the procedures involved during the experiment. This proposed methodology include the obtaining the dataset from Kaggle, data preprocessing, splitting of the data into training and testing, classification methods such as Naïve Bayes and Decision Tree and feature selection using information gain method.

The Near-Miss Undersampling method is used to distribute the data into equal classes. The performance evaluation of the algorithms is done on the basis of Accuracy, precision, recall and F1-score.

The steps included for detecting credit card fraud is represented as flowchart below in Figure 2.

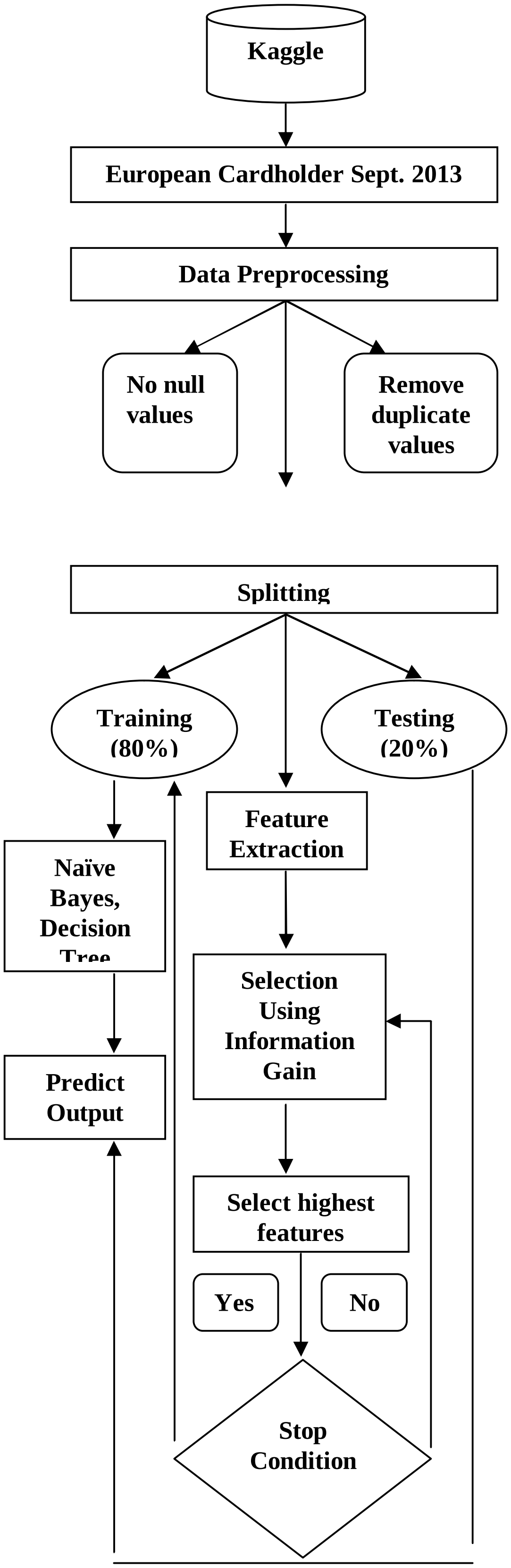


Figure 2: Flowchart of proposed work

4.1 DATA DESCRIPTION

The dataset is taken from Kaggle and described in [17]. The Dataset consists of 284,807 transactions in 2 days by European Cardholders in September 2013 with 492 fraud transactions [14]. This shows that the dataset is very imbalanced. Dataset is divided into Training and Testing Data with 80% and 20%. In this dataset entire 30 input attributes have been used for credit card fraud detection. [4]. In the Dataset Attributes from V1 to V28 are numerical values taken from PCA (Principal Component Analysis) because of high-security reasons, but Time and Amount are the only Attributes that are not changed by PCA [14]. Attribute class is considered as target class and value 1 is a total count of 473 which is for fraud detection and value 0 is 283253 count which is for non-fraud detection [4]. The dataset before applying any algorithm is shown in Figure 3.

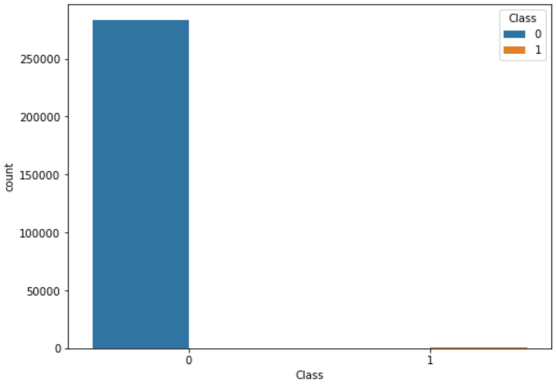


Figure 3: Unbalanced data

4.2 DATA PREPROCESSING

Data Preprocessing is an important step while making a model. Data Preprocessing is a technique in which raw data is processed in such a way that system can understand it efficiently. The Real-World Data is generally incomplete and includes missing values that did not use in the model. Hence, Preprocessing is a crucial step to solve these issues and for cleaning the data. Data Preprocessing is used to increase the accuracy and quality of the data. The Data preprocessing stages include data cleaning, data integration, data transformation, and data reduction which are shown in Figure 4.

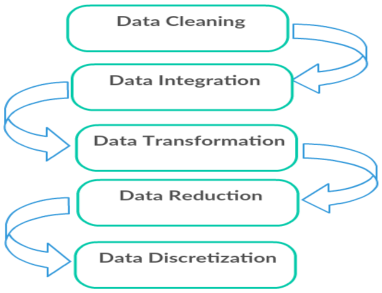
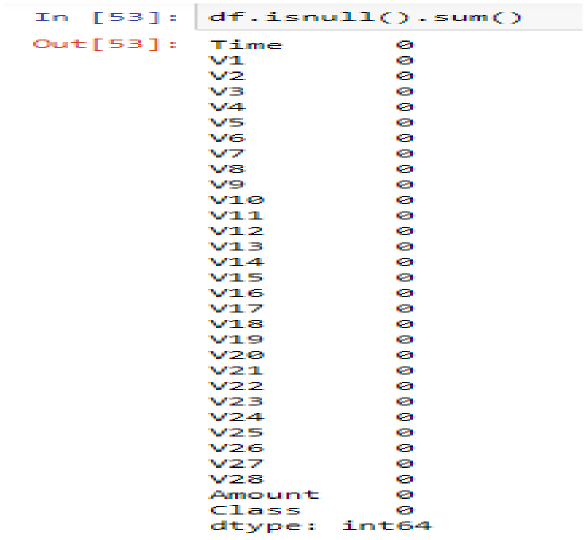


Figure 4: Stages of data preprocessing

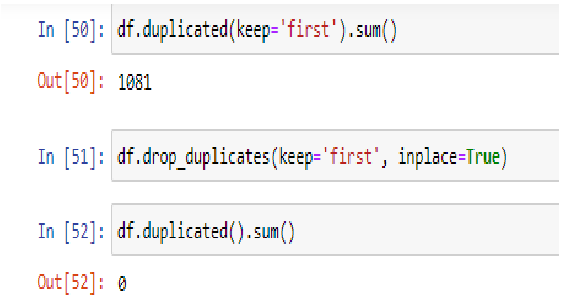
1). Data Cleaning: Real-world data influence to be not completed, noisy and uncertain. Data Cleaning helps in filling the missing values smooth out noise and identifies outsider and exact unpredictability in the information.
2). Data Integration: Data Integration is the method of merging information from various sources into a single and combined view.
3). Data Transformation: Data Transformation is the procedures where you take out data evaluate the data, understand the data and then convert it into something you can examine.
4). Data Reduction: Data Reduction is the method of decreasing the size needed to store the data. Data Reduction can expand storage ability and decrease the cost.
Firstly, we have imported the credit card dataset. Then, we analyze whether the Null values exist or not in the dataset by using Data Preprocessing. By default, no null values are present in the Dataset. But the 1081duplicate values are founded in the credit card dataset. With the help of data preprocessing we have dropped those duplicate values. The existence of no null values is shown in Figure 5.



```
In [53]: df.isnull().sum()
Out[53]: Time      0
         V1        0
         V2        0
         V3        0
         V4        0
         V5        0
         V6        0
         V7        0
         V8        0
         V9        0
         V10       0
         V11       0
         V12       0
         V13       0
         V14       0
         V15       0
         V16       0
         V17       0
         V18       0
         V19       0
         V20       0
         V21       0
         V22       0
         V23       0
         V24       0
         V25       0
         V26       0
         V27       0
         V28       0
         Amount    0
         Class     0
         dtype: int64
```

Figure 5: Null Values

Figure 6 illustrates the data after removing duplicate values.



```
In [50]: df.duplicated(keep='first').sum()

Out[50]: 1081

In [51]: df.drop_duplicates(keep='first', inplace=True)

In [52]: df.duplicated().sum()

Out[52]: 0
```

Figure 6: Removed Duplicate Values

4.3 Resampling Method

A Resampling technique is a set of methods for replicating a sample from a given sample. Resample method is the most useful technique which handles the unbalanced dataset.
The two major types of Resampling are:

1. Under-sampling: In the Under-sampling technique, the majority class removed the instances until the minority class is balanced. But the drawback of using under-sampling is the loss of some important data. Figure 7 understands the working of the Under-sampling method.

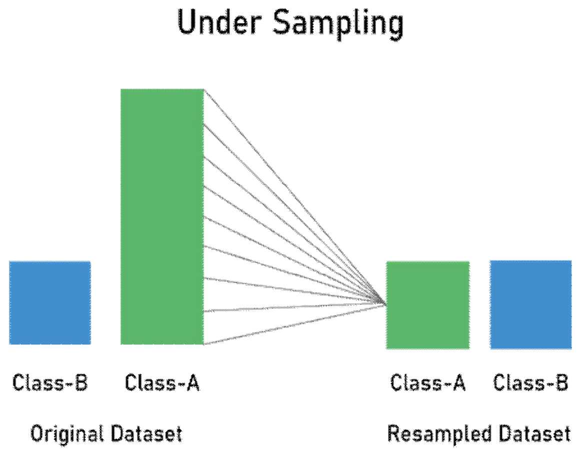


Figure 7: Undersampling Method

2. Oversampling: Oversampling duplicates instances from minority classes by replacing and supplementing the training data. In the Oversampling, some instances are duplicated several times. The illustration of Oversampling method is given in Figure 8.

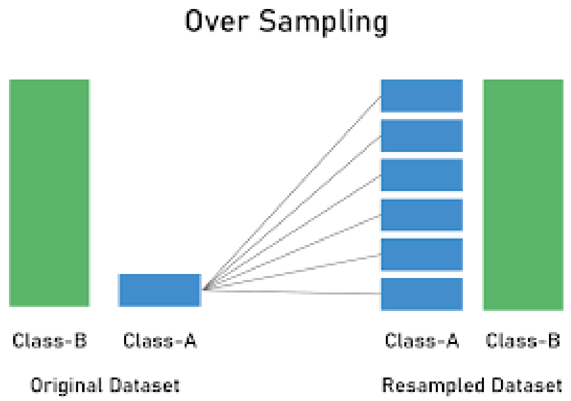


Figure 8: Oversampling Method

4.3.1 Near-Miss Under-sampling Method

Near-Miss is a type of algorithm which is used to balance the dataset. It is a kind of Under-sampling algorithm. Firstly, the algorithm calculates the distance from the maximum class to the shortest distance of the minimum class and makes the two points close to each other. Thus by removing the data points from the maximum class, the Near-Miss algorithm helps in balancing the unbalanced dataset. Near-Miss methods are used to prevent the data loss problem in the dataset. Figure 9 represents the Near-Miss algorithm for balancing the dataset.

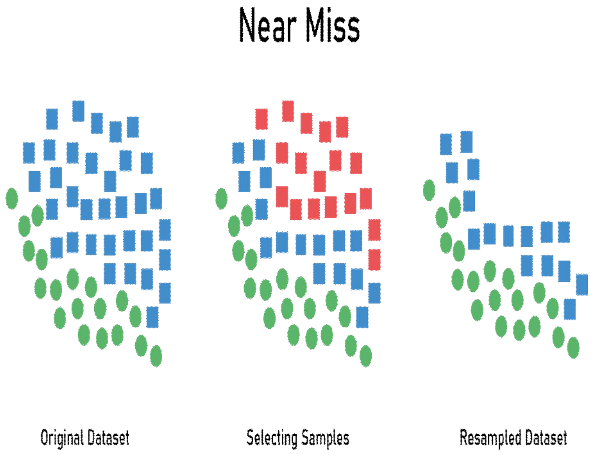


Figure 9: Near miss

The original dataset consists of value 0 is 283253 and value 1 is 473. After applying Near-Miss Undersampling Method, value 0 is 473, and value 1 is 473. The following figure represents the balanced data using the Near-Miss Under-sampling method. The following Figure 10 represents the balanced data using the Near-Miss Under-sampling method.

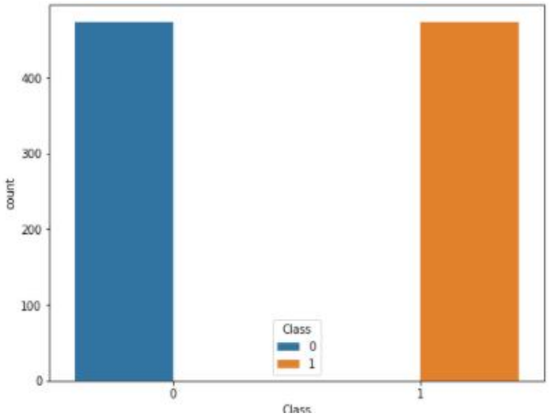


Figure 10: After Near-Miss balanced data

4.4 Feature Selection

Feature Selection also called feature extraction, is a useful method that helps in enhancing the good results of the model. The irrelevant, inconsistent, and redundant features are removed by the feature selection method as they are not more useful for the model. The feature selection method reduces the complexity and computational time of the proposed methodology. This method aims to eliminate those attribute which does not affect the accuracy of the model. There are three types of Feature Selection.

1. Filter method

Filter methods are used to check the dependency of the features. It recognizes the irrelevant features and helps in filtering them out from the model. Statistical methods are used by the Filter technique for evaluation.

2. Wrapper method

The wrapper method calculates the subset of attributes by training the model with a specific machine learning algorithm. Due to cross-validation wrapper methods are computationally expensive. These methods provide better accuracy.

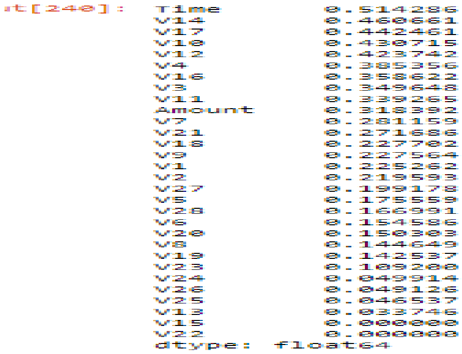
3. Embedded method

The composition of the Filter and wrapper process is known as the embedded method. The embedded method is used to decrease overfitting. These methods are based on time complexity.

4.4.1 Information Gain

In the present work, Information Gain is used for feature selection. Information gain also called Mutual Information, is used in feature selection methods to remove unusable features. This method compares all independent and dependent attributes themselves. The information Gain Method aims to eliminate the input attributes from a dataset. Eliminating the number of input features set leads to the decrease in complexity and computational cost saving of the model, which provides better accuracy of the classifier. It calculates the number of random variables for classification problems. Variables are independent only if the information gain is zero.

In the current study, when K is set to 18 and Information Gain extracts the best 18 features which are useful for the model. Figure 6 shows that the Time attribute is the most important feature which has the highest score of 0.514286 and the lowest score of attribute V22 is 0.000000. The following Figure 11 shows the score for each attribute:



```
t[240]:   Time        0.514286
          V14         0.460661
          V17         0.442461
          V10         0.430715
          V12         0.423742
          V4          0.385356
          V16         0.358622
          V3          0.349648
          V11         0.339265
          Amount      0.318392
          V7          0.281159
          V21         0.271686
          V18         0.227702
          V9          0.227564
          V1          0.225262
          V2          0.219593
          V27         0.199178
          V5          0.175559
          V28         0.166991
          V6          0.154586
          V20         0.150303
          V8          0.144649
          V19         0.142537
          V23         0.109200
          V24         0.049914
          V26         0.049126
          V25         0.046537
          V13         0.033746
          V15         0.000000
          V22         0.000000
          dtype: float64
```

Figure 11: Score for each attribute

The scoring graph of each attribute is used in the preparation of the model out of which useful features are extracted by the Information Gain method for Feature selection. Figure 12 indicates the scoring graph of each attribute.

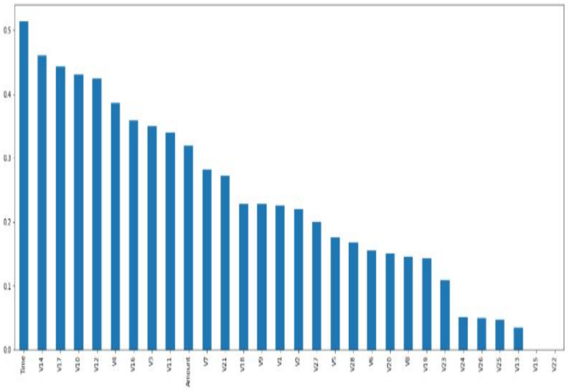
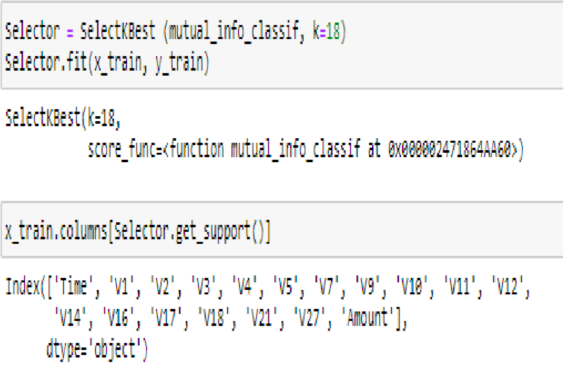


Figure 12: Score Graph of each attribute

The selected features which are chosen with the highest scores are Time, V1, V2, V3, V4, V5, V7, V9, V11, V12, V14, V16, V17, V18, V21, V27, and Amount. The following Figure 13 represents the 18 selected features:



```
Selector = SelectKBest (mutual_info_classif, k=18)
Selector.fit(x_train, y_train)
```

```
SelectKBest(k=18,
            score_func=<function mutual_info_classif at 0x000002471864AA60>)
```

```
x_train.columns[Selector.get_support()]
```

```
Index(['Time', 'V1', 'V2', 'V3', 'V4', 'V5', 'V7', 'V9', 'V10', 'V11', 'V12',
       'V14', 'V16', 'V17', 'V18', 'V21', 'V27', 'Amount'],
      dtype='object')
```

Figure 13: Selected Attributes

4.5 CLASSIFICATION Method

4.5.1 Naive Bayes

Naïve Bayes is a kind of Supervised Learning method which is used to show probabilistic knowledge. Naïve Bayes algorithm is a statistical method that uses the Bayes Theorem for classification. Naïve Bayes selects a decision that has maximum probability. Naive Bayes classifier depends upon two predictions. The first assumption is that all features which had to be labeled should evolve in the decision. Secondly, features will not provide any kind of information about another feature. The formula for Naïve Bayes is represented in Figure 14.

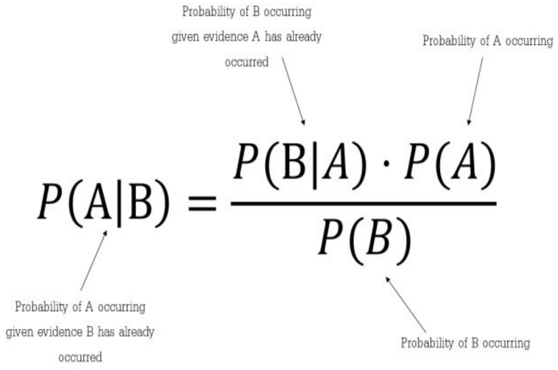

Probability of B occurring given evidence A has already occurred
Probability of A occurring
P(A|B) = P(B|A) · P(A) / P(B)
Probability of A occurring given evidence B has already occurred
Probability of B occurring

Figure 14: Naive Bayes

4.5.2 Decision Tree

The decision tree is a graphical representation for providing solutions to classification and regression issues that are based on various conditions. It is tree-structured which helps in classification and regression problems but generally, it is utilized for classification problems. It has two nodes:

Decision Node: When a major node divides into several nodes and makes decisions then that node is known as a decision node.

Leaf Node: Leaf nodes are the outcome nodes and do not include more branches is called a Leaf node. The following Figure 15 shows the Decision Tree of Credit card transactions:

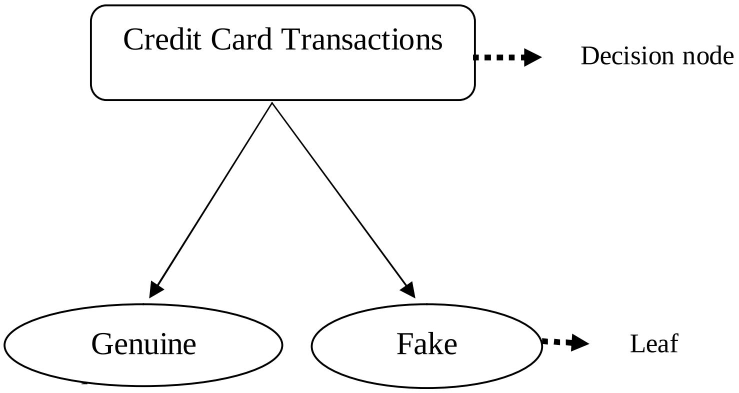


Figure 15: Basic Structure of Decision Tree

Some Important terminologies of Decision Tree:

- **Root Node:** It is the node from which the whole decision tree starts and divides into two or more sets.
- **Splitting:** Division of nodes into various sub-nodes is known as splitting.
- **Pruning:** Removal of unwanted sub-nodes is known as pruning.
- **Parent/Child Node:** The major node of the Decision tree is also known as a parent node. The nodes which succeed the parent node is called child node.
- **Branch/Sub Tree:** Separate tree formed by the process of splitting is called a subtree.

5. Performance Evaluation Metrics

5.1Confusion Matrix

Confusion Matrix is in a tabular form that shows the number of correct and wrong predictions. It is used to calculate the performance of a classifier. Confusion Matrix involves True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN).

		Predicted Value	
		Non-Fraud (NO)	Fraud (YES)
Actual Value	Non-Fraud (NO)	TN	FP
	Fraud (YES)	FN	TP

Figure 16: Confusion Matrix

a) **True Positive (TP):** True value is the same as predicted value i.e. true value is positive and hence, predicted value is also positive.

b) **True Negative (TN):** True value is negative and hence, predictive is also negative.

c) **False Positive (FP):** True value is not the same as predicted value i.e. True value is negative whereas the predicted value is positive. Also known as Type 1 error.

d) **False Negative (FN):** True value is positive while predicted value is negative. Also known as Type 2 error.

A Few performance metrics are utilized to assess the performance of the classification model. The following are performance measurements:

- **Accuracy:** Accuracy helps calculate the total performance of the classifier.
- **Precision:** Precision calculates that out of all positive predictions, how many times fraud cases are there.

- **Recall:** Recall is the opposite of the Precision metric. The recall is the proportion of how often positive classes are correctly predicted.
- **F1 Score:** F1 Score is an overall measure of Precision and Recall and calculates the balance between them.
- **ROC Curve:** ROC is Receiver Operating Characteristics Curve. ROC Curve shows the graph of True Positive and False Positive Rate.
- **Precision-Recall Curve:** Precision-Recall curve is the graph that represents recall on the x-axis and Precision on the y-axis for balanced datasets. Simply, this graph is based on high Precision and High Recall.

6. RESULTS

The results obtained from the proposed algorithm in our experiment are listed below:

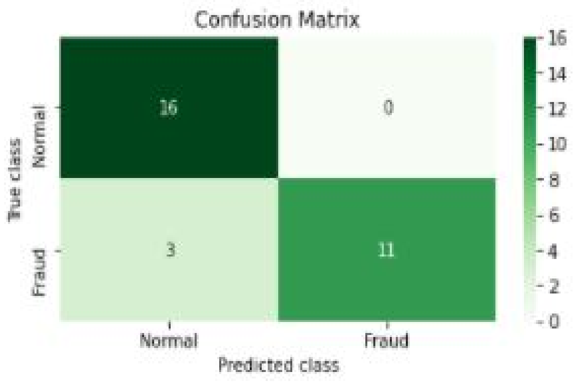


Figure 17: Confusion Matrix of Naive Bayes

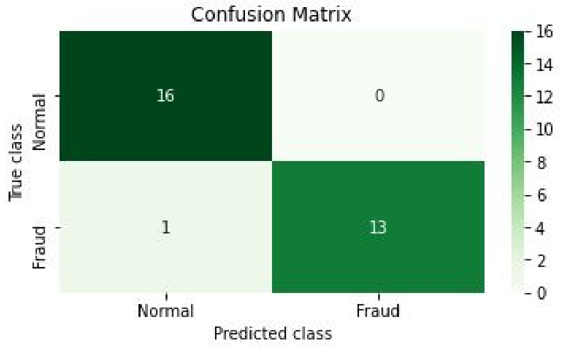
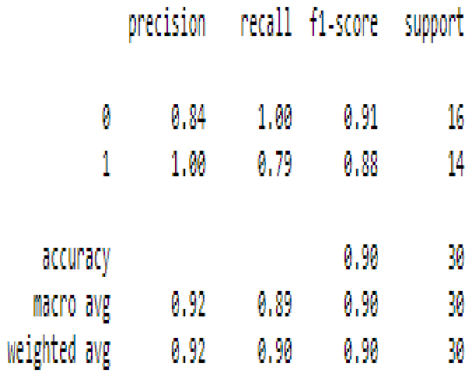


Figure 18: Confusion Matrix of Decision Tree

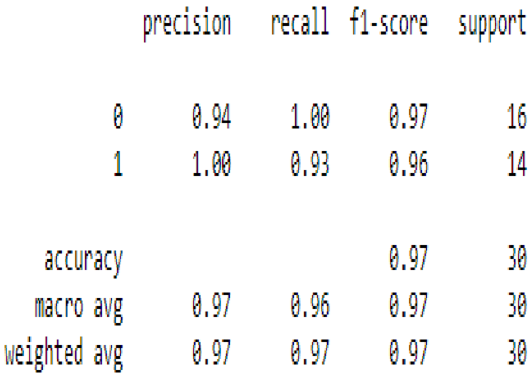


```
              precision    recall  f1-score   support

           0       0.84      1.00      0.91        16
           1       1.00      0.79      0.88        14

    accuracy                           0.90        30
   macro avg       0.92      0.89      0.90        30
weighted avg       0.92      0.90      0.90        30
```

Figure 19: Classification Report of Naive Bayes



```
              precision    recall  f1-score   support

           0       0.94      1.00      0.97        16
           1       1.00      0.93      0.96        14

    accuracy                           0.97        30
   macro avg       0.97      0.96      0.97        30
weighted avg       0.97      0.97      0.97        30
```

Figure 20: Classification Report of Decision Tree

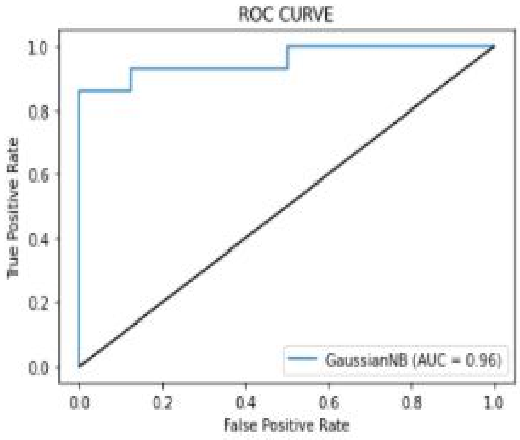


Figure 21: ROC of Naive Bayes

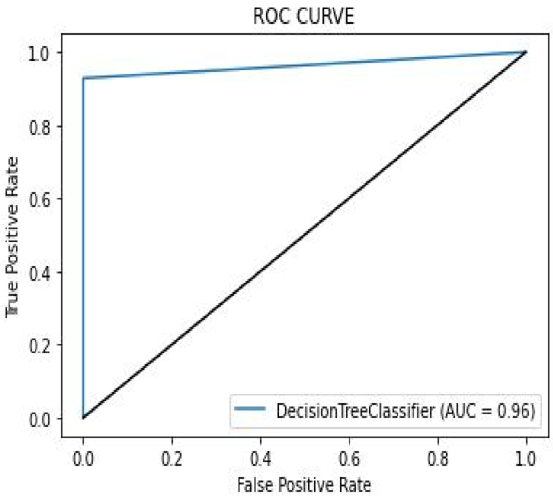


Figure 22: ROC of Decision Tree

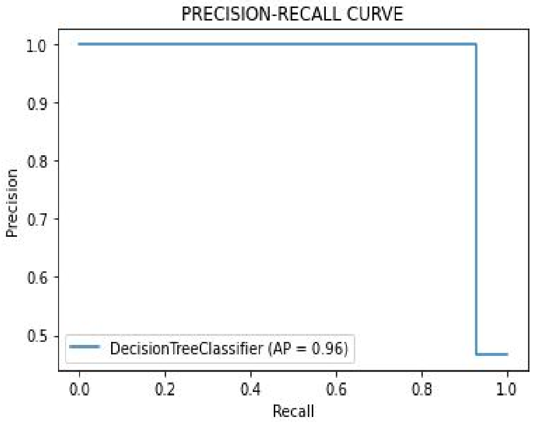


Figure 23: Precision-Recall Curve of Naïve Bayes

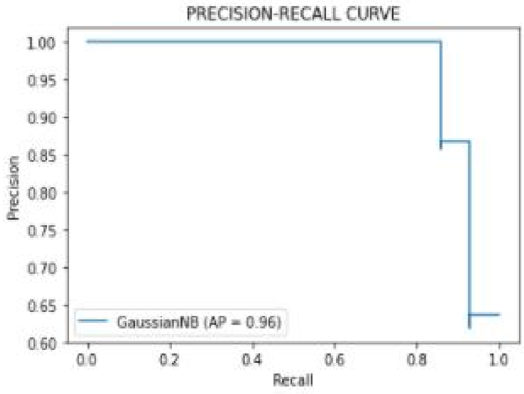


Figure 24: Precision-Recall Curve of Decision Tree

Table 2: Comparison table of proposed algorithms

Feature Selection	**Information gain feature selection when, (K=18)**	
Algorithms	**Naïve Bayes**	**Decision Tree**
Accuracy	90%	**97%**
Precision	92%	**97%**
Recall	90%	**97%**
F1-Score	90%	**97%**

The above comparison table of proposed algorithms shows that Decision Tree has attained an accuracy of 97%, Precision (97%), Recall (97%), and F1-score (97%) when K=18 in Information Gain Feature Selection. As compared to Naïve Bayes has an accuracy of 90%, Precision (92%), Recall (90%), and F1-score (90%). The Decision Tree is considered a better algorithm based on Confusion Matrices such as Accuracy, Precision, Recall, and F1-score for detecting credit card frauds in the proposed work.

7. CONCLUSION

Credit cards are becoming a trend in online shopping, Business organizations. Its fraud transactions also have been growing globally. Hence, detecting Credit Card fraud is extremely challenging. A few issues make it difficult to identify solutions for detecting credit card frauds in which class imbalance issue is one significant.

The proposed approach includes the comparison of Naive Bayes and Decision Tree algorithms Near-Miss Under-sampling method which is used to handle the imbalanced data. Also, Information gain is used on balanced data for feature selection to selects the best features. It is shown that Decision Tree achieved better accuracy than Naive Bayes. The Naïve Bayes obtained an accuracy of 90% and a Decision Tree of 97% for classifying Credit Card Fraud transactions. Decision Tree has also achieved good results in Precision, Recall, and F1 score.

8. FUTURE SCOPE

From the above results, we can conclude that there are many more Resampling methods to balance the dataset and different machine learning techniques are also applied to detect the credit card fraud results correctly.

REFERENCES

1. Itoo, F. and Singh, S. (2020) **Comparison and analysis of Logistic Regression, Naïve Bayes and KNN machine learning algorithms for credit card fraud detection**. *Springer,* pp.1-9.
2. Taha, A. A., & Malebary, S. J. (2020) **An intelligent approach to credit card fraud detection using an optimized light gradient boosting machine.** *IEEE Access*, *8*, pp. 25579-25587.
3. Randhawa, K., Loo, C. K., Seera, M., Lim, C. P., and Nandi, A. K. (2018) **Credit card fraud detection using AdaBoost and majority voting.** *IEEE Access*, *6*, pp. 14277-14284.
4. Makki, S., Assaghir, Z., Taher, Y., Haque, R., Hacid, M. S., and Zeineddine, H. (2019**) An experimental study with imbalanced classification approaches for credit card fraud detection.** *IEEE Access*, *7*, pp. 93010-93022.
5. Fu, K., Cheng, D., Tu, Y., and Zhang, L. (2016 October**) Credit card fraud detection using convolutional neural networks.** *Springer,* pp. 483-490.
6. Rtayli, N., and Enneya, N. (2020). **Enhanced credit card fraud detection based on SVM-recursive feature elimination and hyper-parameters optimization.** *Journal of Information Security and Applications*, pp. 55, 102596.
7. Carcillo, F., Le Borgne, Y. A., Caelen, O., Kessaci, Y., Oblé, F., and Bontempi, G. (2019). **Combining unsupervised and supervised learning in credit card fraud detection.** *Information Sciences*.
8. de Sá, A. G., Pereira, A. C., & Pappa, G. L. (2018**) A customized classification algorithm for credit card fraud detection.** *Engineering Applications of Artificial Intelligence*, *72*, pp. 21-29.
9. Singh, A., & Jain, A. (2019) **Adaptive credit card fraud detection techniques based on feature selection method.** In *Advances in computer communication and computational sciences*, pp. 167-178). Springer, Singapore.
10. Riffi, J., Mahraz, M. A., El Yahyaouy, A., & Tairi, H. (2020, June) **Credit Card Fraud Detection Based on Multilayer Perceptron and Extreme Learning Machine Architectures.** In *2020 International Conference on Intelligent Systems and Computer Vision (ISCV)*, pp. 1-5. IEEE.
11. Lucas, Y., Portier, P. E., Laporte, L., He-Guelton, L., Caelen, O., Granitzer, M., & Calabretto, S. (2020) **Towards automated feature engineering for credit card fraud detection using multi-perspective HMMs.** *Future Generation Computer Systems*, *102*, pp. 393-402.
12. Kumari, P., & Mishra, S. P. (2019). **Analysis of credit card fraud detection using fusion classifiers.** In *Computational Intelligence in Data Mining*, pp. 111-122. Springer, Singapore.
13. Varmedja, D., Karanovic, M., Sladojevic, S., Arsenovic, M., & Anderla, A. (2019, March) **Credit card fraud detection-machine learning methods**. *(INFOTECH)*, pp. 1-5. IEEE.
14. Vardhani, P. R., Priyadarshini, Y. I., & Narasimhulu, Y. (2019) **CNN data mining algorithm for detecting credit card fraud**. In *Soft Computing and Medical Bioinformatics*, pp. 85-9. Springer, Singapore.
15. Kulkarni, P., & Ade, R. (2016) **Logistic regression learning model for handling concept drift with unbalanced data in credit card fraud detection system.** In *Proceedings of the Second International Conference on Computer and Communication Technologies*, pp. 681-689. Springer, New Delhi.
16. 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.
17. https://www.kaggle.com/mlg-ulb/creditcardfraud
18. Dornadula, V. N., & Geetha, S. (2019) **Credit card fraud detection using machine learning algorithms.** *Procedia Computer Science*, *165*, pp. 631-641.
19. Dhankhad, S., Mohammed, E., & Far, B. (2018, July) **Supervised machine learning algorithms for credit card fraudulent transaction detection: a comparative study.** In *2018 IEEE International Conference on Information Reuse and Integration (IRI)*, pp. 122-125. IEEE.
20. Fiore, U., De Santis, A., Perla, F., Zanetti, P., & Palmieri, F. (2019**). Using generative adversarial networks for improving classification effectiveness in credit card fraud detection**. *Information Sciences*, *479*, pp. 448-455.
21. Bahnsen, A. C., Aouada, D., Stojanovic, A., & Ottersten, B. (2016). **Feature engineering strategies for credit card fraud detection.** *Expert Systems with Applications*, *51*, pp.134-142.
22. Ahmad, S. N. W., Ismail, M. A., Sutoyo, E., Kasim, S., Mohamad, M. S. (2020). **Comparative performance of Machine Learning Methods for Classification on Phishing Attack Detection.** *In world Conference on Technology, Innovation and Entrepreneurship*, pp. 349-354.