MACHINE LEARNING ALGORITHMS FOR DETECTION OF CREDIT CARD FRAUDULENT TRANSACTIONS

Finalization Phase

Academic Year: 2022
Course: PROJECT: COMPUTER SCIENCE PROJECT (DLMCSPCSP01) - Repetition
Student: Fares, Joseph
Matriculation Number: 32103910

Table of Contents

1. Abstract

With the advancement of the global communication highway and modern technology, credit cards fraudulent activity is on the rise. Credit card fraud activities charge financial companies millions of dollars yearly. As a result, fraud detection systems have become critical for financial institutions and banks seeking to reduce their losses and the impact of fraudulent activities. Fraudulent transactions detection is difficult due to an unbalanced set of records.

An earlier Project that was submitted in Partial Fulfillment of the Requirements for the Degree of Master of Engineering at the University of Victoria titled "Supervised Machine Learning Algorithms for Credit Card Fraudulent Transaction Detection: A Comparative survey" in which several machine learning algorithms as Logistic Regression, KNN, Random Forest, Decision tree and Naïve Bayes were used to tackle and detect the credit card fraudulent issues.

Our target in this project is to build up on the previously mentioned project by modifying the applied algorithms as follows:

- **K-Fold Cross validation:** is a popular technique used to validate the performance of our model. The model evaluation is done using the data set's different parts as a valid set. Then taking these iterations average results. This technique will be applied in all the algorithms to study its impact. This technique will be adopted in all algorithms in this project in comparison with the mentioned project that used the Stratified Shuffle Split function.
- **In the KNN-algorithm**: In this project, it will be tried in order to find the optimal K to build the model through enumerating from (K = 1 to K = 25) rather than just using only K = 3 which was used previously in the mentioned project.
- **Decision trees Pruning:** this method will be used to reduce the final tree complexity and therefore overfitting is reduced. And compare the results with those gained in the previously mentioned project which did not use the pruning strategy.
- **In the Random Forest Algorithm:** In each fold iteration, Random Forest will be created by more than one time by varying the following variables: maximum features, number of estimators and the maximum depth, then finding the corresponding model results to find the best combination of Random Tree variables. Instead of just train the data with the random forest classifier without changing any variable of the Random Forest as was done in the previously mentioned project.
- **Support Vector Machine Algorithm**: will be implemented in addition to the above algorithms the on the credit card transactions dataset and study its impact on the dataset. (This algorithm was not used in the previously mentioned project).

Using PyCharm IDE Python with programming language and a data set from Kaggle, which is data set of European credit card transactions and is considered highly imbalanced. A Comparative evaluation will be carried out to discover which algorithm has the highest overall performance through matrices like accuracy, precision, recall, f1 score and compare the gained results with the previous mentioned project results. for the data set balancing, Under Sampling, Over Sampling and SMOTE techniques are used with each algorithm and compare the results to recognize which performs better.

2. Introduction

Credit cards are cards issued to customers (cardholders) and enable them to purchase services and goods within their available credit or withdraw money ahead of time. Credit cards give the cardholders a time advantage, by allowing their customers to pay back later in a specified time by postponing it to the following billing cycle. (Vaishnavi Nath Dornadulaa, 2019)

Credit card scams are easy to commit. In a short period of time, a substantial sum of money can be withdrawn without the owner's knowledge with no risk. Fraudsters always attempt to make every fraudulent transaction appear legitimate, making fraud detection a difficult task. According to FTC statistics (https://www.ftc.gov/news-events/press-releases/2019/02/imposter-scams-top-complaints-made-ftc-2018, n.d.), 1,579 data thefts were detected in 2017 and approximately 179 million records, with credit card frauds being the most common type with 133,015 reports, accompanied by tax evasion or employment with 82,051 reports of phone frauds (Vaishnavi Nath Dornadulaa, 2019).

According to the US Payments Forum report from 2017 (https://www.ftc.gov/news-events/press-releases/2019/02/imposter-scams-top-complaints-made-ftc-2018, n.d.), criminals have shifted their focus to activities related to CNP transactions as chip card security has improved. Figure 2 depicts the number of CNP fraud cases reported in each year (Vaishnavi Nath Dornadulaa, 2019).

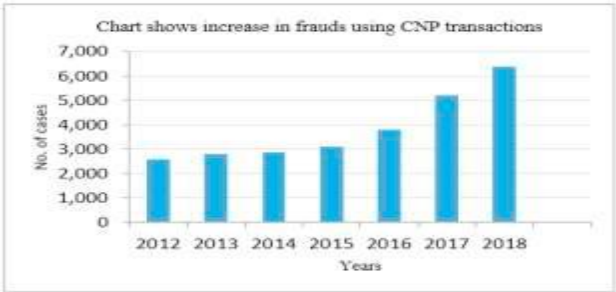

Figure 1: Frauds Using Card Not Present Transaction

Fraud can be prevented in two ways: detection and prevention. Prevention serves like a layer of defense against fraudulent attacks. After prevention has failed, detection occurs. As a result, detection aids in identifying and alerting when a frauds transaction is initiated. CNP transactions in credit card processes have recently gained popularity between online payment gateways (Anuruddha Thennakoon, July 2019).

As per the Nilson Report published October 2016, online payment systems generated more than $31 trillion globally by in 2015, with an increase 7.3% than 2014. Losses from around the world as a result of credit cards fraud have rose in 2015 to $21 billion, and has reached to $31 billion in 2020. (Robertson, Investments &amp; Acquisitions, September 2016) However, there has been a significant increase in fraudulent transactions, which has had a significant impact on the economy. (Anuruddha Thennakoon, July 2019)

Card-not-present and Card-present frauds are the two types of frauds that are recognized in a series of transactions These two categories are further subdivided into theft/counterfeit fraud, application fraud, bankruptcy fraud, and behavioral fraud. (Anuruddha Thennakoon, July 2019)

Although the information mining technique is from the most effective and well-known techniques for detecting fraud, the true motive and legitimacy of any transaction cannot be guaranteed. In fact, the best effective option is to search for possible evidence of fraud from available data using statistical algorithms. Card fraud identification is the conceptual model for classifying fraudulent activity into sub classes of authentic class and not-genuine class (Maes, 2002; Adepoju et al.,2019).

Fraud of credit card is detected by analyzing a card's spending behavior. Support Vector Machine (Singh, 2012), Genetic algorithm (Kalyani, 2012), Decision tree (Patil, 2015), Artificial Neural Network (Ogwueleka, 2011), and Naive Bayes have all been linked to card fraud recognition (Bahnsen, 2014; Adepoju et al.,2019).

Credit card companies are attempting to predict the authenticity of a purchase by assessing inconsistencies in different areas such as purchasing site, payment amount, and user past purchases. Even though, with the dramatic rise in credit card fraud scenarios, optimizing algorithm solutions is critical for credit card companies. (Gonzalez; Adepoju et al.,2019)

Credit card detection is associated with a number of challenges, for example, dynamic fraudulent behavior patterns, which means that falsified operations will, in general, look like authentic ones; additionally, card transactions informational collections are occasionally accessible and extremely imbalanced; and thirdly, the efficiency of the identification of credit card fraud is incredibly influenced by the type of testing approach, variable selection, and identification technology. (Adepoju et al.,2019)

Data constantly changes and alters over time, causing the status of regular and fraud behaviors that were genuine activities in the past to be always different and possibly the current fraud or vice versa. (Adepoju et al.,2019)

<u>Issues and challenges for credit fraud detection</u>

Issue	Description
No standard credit card dataset (Jha et al., 2012; Zareapoor et al., 2012; A. Singh, 2019)	It is one of the most serious issues in the fraud detection domain. There is no available standard, real-world, or benchmark dataset to evaluate proposed fraud detection methodologies. In the majority of cases, researchers conducted research using their dataset (Synthetic dataset).
Skewed class distribution (Zareapoor et al., 2012; Abdallah et al., 2016; Seeja & Zareapoor, 2014; Sorournejad, Zojaji et al., 2016; A. Singh, 2019)	From most serious issues is skewed distribution (or imbalanced class). In comparison to normal card transitions, only a small proportion of all card transitions are fraudulent. When percentage of the minority class is low in a supervised learning technique, an imbalance problem occurs.
Cost-sensitive classification problem (Sorournejad et al., 2016; Sahin et al., 2013; A. Singh, 2019)	Due to the financial impact ranging from a few to thousands of dollars, credit card transactions are misclassified (fraudulent transaction as a legitimate transaction and legitimate transaction as a fraudulent transaction).
Presently no suitable evaluation criterion (Sorournejad et al., 2016; Zareapoor et al., 2012; A. Singh, 2019)	No standard evaluation tool is available for evaluating and compare results of a fraud detection system. Because the data set is imbalanced, accuracy is not a suitable CCFD metric.
Fraudsters behavior dynamic (Zareapoor et al., 2012; A. Singh, 2019)	Fraudsters change their behavior from time to time in order to obtain card details and avoid detection systems, or to modify their fraud styles.
Cardholder Behavior (Concept Drift) (Abdallah et al., 2016; A. Singh, 2019).	The cardholder's behavior is constantly changing, and because of specific situations/occasions (e.g., New Year), users' purchasing power will be increased. If the CCFD system does not consider these changes to be normal, they will be considered fraudulent behavior.
Pattern Recognition Algorithm (Abdallah et al., 2016; A. Singh, 2019).	To reduce fraud cases, pattern recognition algorithms are used to recognize fraudster and customer patterns.

3. Related Work

Machine learning algorithms (Mohammed, 2018) as Naive Bayes Classification, Decision Trees, Least Squares Regression, Logistic Regression, and SVM are used to detect fraudulent transactions in real-time datasets. To train the characteristics of abnormal and normal transactions, two random forests methods (Xuan, 2018) are used. They are based on random forests and random trees. Even though random forest produces good results on small data sets, it has some limitations when dealing with imbalanced data. The previously mentioned issue will be the focus of future work. The algorithm of the random forest should be enhanced. (Vaishnavi Nath Dornadulaa, 2019)

The performance of Logistic Regression, Nave Bayes and K-Nearest Neighbor is examined on highly skewed fraud data of credit cards. In addition, studies are being conducted to investigate meta-learning approaches and meta-classifiers to deal with credit card fraud data which is highly imbalanced. Although supervised learning methods can be used, they may fail to detect fraud in some cases. “Restricted Boltzmann machine (RBM) model” is capable of constructing normal transactions in order to detect anomalies in normal patterns. (Pumsirirat, 2018). Furthermore, a hybrid method is developed by combining Ad boost and Majority Voting methods (Randhawa, 2018). (Vaishnavi Nath Dornadulaa, 2019)

Tyler et al., for example, extended a framework proposed in (M. F. Zeager, 2017), implemented the model, and applied the model to a real-world transaction log. Logistic Regression (LR) was used to solve the classification problem. The occurrences of fraudulent transactions were discretized into strategies using “Gaussian Mixture Models (GMMs)”. The SMOTE technique is utilized to tackle the class imbalance. Sensitivity analysis was used to highlight the significance of estimates in economic value. The results demonstrated that a practical method that uses fewer steps to retrain a model can perform as well as a classifier that typically retrains every round (T.Cody, 2018). (Thennakoon, 2019)

Chee et al. used twelve hybrid methods and standard models which use majority voting methods and AdaBoost to achieve better accuracy rates in credit card fraud detection (K. Randhawa, 2018). They were evaluated using both real-world data and benchmark. The methods' strengths and weaknesses were summarized. “The Matthews Correlation Coefficient (MCC)” metric has been chosen as the performance metric. To test the algorithms' robustness, noise was introduced into the data. They also demonstrated that no effect on the majority voting method by the added noise. (Thennakoon, 2019)

Except for accuracy, the analysis on highly imbalanced data in the paper (J.O. Awoyemi, 2017) shows that KNN performs admirably for sensitivity, specificity, and MCC. (Gianey, 2017) discussed commonly used supervised techniques and provided a thorough evaluation of supervised learning techniques in his paper. They also demonstrated that all algorithms change depending on the problem area. (Thennakoon, 2019)

The fraud detection system presented in the paper (G. E. Melo-Acosta, 2017) is designed to deal with class imbalance, the formation of labelled and unlabeled data, and the processing of large datasets. the proposed system has successfully tackled all of the challenges. (Thennakoon, 2019)

In the paper (A. Mishra, 2018), LR, GB, SVM, RD and a mix of classifiers were used, resulting in a high recall of more than 91 percent on a European dataset. Only after balancing the data by under sampling, high recall and precision were achieved. (Karanovic et al., 2019)

In the paper (S. V. S. S. Lakshmi), a European dataset was also used, and models based on LR, DT, and RF were compared. RF proved to be the most accurate of the three models, with a 95.5 percent accuracy, followed by DT with a 94.3 percent accuracy and LR with a 90 percent accuracy. (Karanovic et al., 2019)

“K-Nearest neighbors (KNN)” and outlier detection techniques, according to (N. Malini, 2017) and (Mrs. C. Navamani), can also be effective in fraud detection. They are shown to reduce false alarm rates while increasing the rates of fraud detection. The KNN algorithm also performed well in an experiment for paper (J. O. Awoyemi, 2017), where the authors tested and compared it to other traditional algorithms. (Karanovic et al., 2019)

Deep learning techniques and some classical algorithms were compared in a paper (Z. Kazemi, 2017). The tested techniques achieved around 80% accuracy. The authors of the paper (S. Dhankhad, 2018) used a European dataset to compare the following algorithms: Decision Tree, KNN, Naïve Bayes, GB, RF, Support Vector Machine, Logistic Regression, XGBoost (XGB), MLP, and stacking classifier (a combination of multiple machine learning classifiers). All of the algorithms achieved high accuracy of more than 90% as a result of thorough data preprocessing. With an accuracy of 95% and a recall value of 95%, the stacking classifier was the most successful. (Karanovic et al., 2019)

The European dataset was tested by a neural network in a paper (C. Wang, 2018). The experiment included a “back propagation neural network” optimized with the Whale algorithm. Exceptional results were achieved on 500 test samples thanks to an optimization algorithm: accuracy of 96.40% and recall of 97.83%. The authors of the papers (N. Kalaiselvi, 2018) and (F. Ghobadi, 2016) used neural networks to demonstrate how ensemble techniques improve results. (Karanovic et al., 2019)

Three datasets were used in the paper (A. Pumsirirat, 2018) to compare “Restricted Boltzmann Machine and Auto-encoder algorithms”, which reached the conclusion that algorithms such as MLP can be used to detect credit card fraud. (Karanovic et al., 2019)

Several papers have been published on the using of deep neural networks for detection of fraudulent transactions. These models, on the other hand, require more computation and better performing on larger datasets. (Learning - Towards Data Science. [online]). This approach may produce excellent results, as evidenced by some papers, but what if the same or even better results can be obtained with fewer resources? Our main goal is to demonstrate that with proper preprocessing, various machine learning algorithms can produce acceptable results. (Karanovic et al., 2019)

The authors of most of the papers mentioned used the under-sampling technique, which motivated them to use a different approach - the oversampling technique.

4. Technical Background

4.1. Machine Learning

Is an Artificial Intelligence branch in which computers are trained to recognize patterns in large informational indexes or datasets and naturally improve those examples without the need for human intervention. The training procedure begins with a basic machine-learning algorithm that generates training data in order to dissect the relationship between various components and an objective esteem. During the training stage, the algorithm is unambiguously given the target value. When trained, the model is used to forecast unknown target values for other data instances. (Adepoju et al., 2019)

Depending the type of training data weather it is labeled or not , the machine learning algorithms are classified as supervised algorithms in case of labeled data or unsupervised algorithms in case of unlabeled data. Supervised learning is concerned with discovering the relation between the input value and the output value in order to predict more output values when providing more input. (Adepoju et al., 2019)

Supervised learning problems can be classified as regression or classification. (Adepoju et al., 2019)

Classification algorithms classifies output (as legit or fraud) while regression output is a specific value (as length). Machine learning algorithms have no output, but perform analysis on the relationship between the input and output and known as unsupervised because the training data have no labels or classification (What is Machine Learning? A definition, Expert System, 05-Oct-2017). (Adepoju et al., 2019)

4.2. Fraud on Credit Card Detection

With credit card payments becoming the most popular method of payment both online and in person, credit card fraud is on the rise. Ability to distinguishing fraudulent transaction by utilizing conventional methods for manual identification are inaccurate and tedious. As a result, the proliferation of large amounts of data has rendered traditional method strategies increasingly unrealistic. Corporate organizations have switched to smart methods, however with smart methods based on artificial learning. (Adepoju et al., 2019)

Methods for detecting predictive fraud were classified into: Unsupervised and Supervised (Bolton). Design is projected based on features of deceptive and legit operations in supervised fraud detection methods (Bhattacharyya, 2011) to classify new transactions as fraudulent or legitimate while outliers' transactions are detected as potential instances of fraudulent transactions in unsupervised fraud detection. A point-by-point dialog of supervised technique and unsupervised machine technique can be discovered in (Kou, 2004). Variety of research have been conducted on various method to solve the problem of detection card scam. These methods include: NB, ANN, LR, DT, SVM, KNN etc. (Adepoju et al., 2019)

4.3. Decision Tree

A Supervised Learning algorithm, A decision tree has a tree structure, with a root which is split into child nodes using a splitting algorithm, until no more splitting is required because there will be no change in the model results. As the decision tree grows more, overfitting of the training data in branches is possible to occur. So, the tree performance is improved by the tree pruning method by removing certain nodes. What make decision trees quite popular is ease in the use, and the flexibility it provides to handle different data types of attributes (Hauska, 1977).(Tiwari et al., 2021)

Decision Tree Pruning

Pruning procedures for a complicated tree is developed to facilitate interpretation. Pruning, as defined by (S. Drazin, and M. Montag), is a tool for reducing the tree size by getting rid of tree parts that are not meaningful in order to avoid over-fitting and unnecessary complexity. Pruning decision trees is a critical step in optimizing a model's computational efficiency as well as classification accuracy. The pruning method used in clearly controls the tree's complexity (L. Rokach et al., 2008). Pruning methods are classified into two categories: “Pre-Pruning and Post-Pruning” According to Mahmood (A. M. Mahmood, 2010), pre-pruning works by stopping the tree's growth early based on a stopping criterion, before it can correctly classify the training set. The advantage of pre-pruning is: it does not generate a full tree, but the disadvantage is the horizon effect phenomenon (Quinlan, 1993). Post-pruning is divided into two stages: growing and pruning. It first allows over-fitting of the data before post-pruning the grown tree. It was proven that, post-pruning methods performs better than pre-pruning methods (A. M. Mahmood, 2010). Reduced Error Pruning (REP) is a decision tree used after pruning (F. Esposito et al., 1999). In relation to the pruning set, REP finds the most accurate subtree's smallest version. (J. Chen et al., 2009). (Omar, 2012)

4.4. Random Forests

The instability in single trees and sensitivity to some training data led to development of another model that is random forests. As every single tree is being built independently of one another, computational efficiency of random forest is comparatively better (Ho, Random decision forests, 1995). It is basically an ensemble of regression and/or classification trees with it obtaining variance amongst its trees and hence are easy to use because of use of only two randomness sources or parameters that is building trees using trained data separate bootstrapped along samples with considering only a random data attribute subset to build each tree as specified (S. Bhattacharyya et al., 2011). (Tiwari et al., 2021)

The main variables of the Random Forest are:

- ***Maximum Feature:***

Python provides several options for assigning maximum features. Here are some examples:

- **Auto/None**: This will take all the features which can be used in every tree without any restrictions on the individual tree.

- **sqrt**: This option will only consider the square root of the maximum features in one run. Another type that can be used is” log2″.

- ***Number of Estimators:***

This is the trees needed to be created before taking the most votes or prediction averages. The higher the performance as the number of trees increases, but the code becomes slower. it's better to decide on as high value because the processor can handle because this makes the predictions stronger and more stable, in our case we used the worth of 100.

- ***Maximum Depth:***

The tree's maximum depth. If None, nodes are grown till all leaves become pure or contain fewer than min samples split samples. (https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html, n.d.)

4.5. Logistic Regression

When the dependent variable is dyadic or binary, this technique is appropriate for predictive analysis (Ferrier, 2000). Because the classification of transactions as fraudulent is a double-edged variable, this technique can be used. The logistic curve is used by this statistical classification model based on probabilities to detect fraud. Because the logistic curve's value ranges from 0 to 1, it can be used to interpret class membership probabilities. The dataset fed into the model for training and testing is being classified. Following model training, it is tested for a minimum cut-off value for prediction. The model is then tuned based on the selection of the most significant variables. The prediction accuracy was 70 percent.

As a result, the outlier points are not effectively handled (P. Suraj et al., 2018). It employs the natural logarithmic function to compute probability and demonstrate that the results fall into a specific category (S. Bhattacharyya et al., 2011). (Tiwari et al., 2021)

4.6. Support Vector Machines

Support vector machines or SVMs are linear classifiers as stated in (S. Bhattacharyya et al., 2011) that work in high dimensions as in high-dimensions, a non-linear task becomes linear and hence this makes SVMs highly useful for detecting frauds. Due to its two most important features that is a “kernel function” to represent classification function in the dot product of input data point projection, and the fact that it tries finding a hyperplane to maximize separation between classes while minimizing overfitting of training data, it provides a very high generalization capability (Burges, 1998). (Tiwari et al., 2021)

4.7. K-Nearest Neighbors

This is a supervised learning technique that achieves consistently high performance in comparison to other fraud detection techniques of supervised statistical pattern recognition (M. L. Zhang and Z. H. Zhou, 2007). Three factors majorly affect its performance: distance to identify the least distant neighbors, some rule to deduce a categorization from k-nearest neighbor & the count of neighbors to label the new sample. This algorithm classifies any transactions that occurred by computing the least distant point to this particular transaction and if this l east distant neighbor is classified as fraudulent then the new transaction is also labeled as a fraudulent one. Euclidean distance is a good choice to calculate the distances in this scenario. This is a fast technique and results in fault alerts. Its performance can be improved by distance metric optimization (Sudha, 2017). (Tiwari et al., 2021)

4.8. Naïve Bayes

Based on prior knowledge, Naive Bayes classifiers calculate the likelihood of a sample belonging to a specific category. They employ the Nave Bayes Theorem, which is based on the assumption that the effect of a feature sample is not dependent of the rest of the features. That is, each character in a sample contributes independently to determining the likelihood of the sample's classification, generating the category with the highest likelihood of the sample. Predictors in Bernoulli Nave Bayes are Boolean variables. The class variable is predicted by the parameters which can only have yes or no values. (K.Ratna Sree Valli et al., 2020)

4.9. Evaluation (K.Ratna Sree Valli et al., 2020):

There are different measurements for various algorithms that have been evolved to assess very different things. As a result, it should serve as a criterion for evaluating the various proposed methods. TP, TN, FP, FN and the relationship between them are used by researchers to compare the results accuracies of different approaches.

True Positive (TP): indicates the fraction of fraudulent transactions that are classified correctly as fraudulent transactions.

$$TP = \frac{TP}{TP + FN}$$

- **True Negative (TN):** indicates the fraction of the normal transactions that are classified correctly as normal transactions.

$$TN = \frac{TN}{TN + FP}$$

- **False Positive (FP):** indicates the fraction of the non-fraudulent transactions that are classified wrongly as fraudulent transactions.

$$FP = \frac{FP}{FP + TN}$$

- **False Negative (FN):** indicates the fraction of the non-fraudulent transactions that are classified wrongly as normal transactions.

$$FN = \frac{FN}{FN + TP}$$

- **Confusion matrix:** The confusion matrix provides additional insight into the performance of a predictive model and also which classes are being correctly predicted, incorrectly, and what type of errors made. A two-class classification problem with negative and positive classes yields the simplest confusion matrix.

Predicted	**Positive**	**Negative**
Positive	TP	FN
Negative	FP	TN

- **Accuracy:** is defined as:

$$Acccuracy = \frac{Number\ of\ Correct\ Predicitons}{Total\ Number\ of\ Predicitions}$$

- **Precision:** is defined as:

$$Precision = \frac{TP}{TP + FP}$$

- **Recall:** is defined as:

$$Recall = \frac{TP}{TP + FN}$$

- **F1 score:** is calculated by:

$$F1\,Score = 2\,x\,\frac{Recall\ x\ Precision}{Recall + Precision}$$

5. Methodology

5.1. Dataset Description

The dataset used contains transactions of a cardholder during two days, i.e., two days in September 2013. The data set contains 284,807 transactions, 492 of which are fraudulent, resulting in a 0.172 percent fraud rate. This dataset is significantly unbalanced. Because providing a customer's transaction details is considered a confidentiality issue, The Principal Component Analysis (PCA) was used to transform most of the dataset features. V1, V2, V3..., V28 are features produced from PCA transformations, while the rest, namely 'time,' 'amount,' and 'class,' are non-PCA-applied features, as shown in the table below. (Vaishnavi Nath Dornadulaa, 2019)

#	**Feature**	**Description**
1	Time	Time in seconds to specify the elapses between the current transaction and first transaction.
2	Amount	Transaction amount
3	Class	0 - not fraud 1 – fraud

5.2. Project Implementation

An earlier Project that was submitted in Partial Fulfillment of the Requirements for the Degree of Master of Engineering at the University of Victoria titled “Supervised Machine Learning Algorithms for Credit Card Fraudulent Transaction Detection: A Comparative survey” (Rahman, 2021) in which several machine learning algorithms as Logistic Regression, KNN, Random Forest, Decision tree and Naïve Bayes were used to tackle and detect the credit card fraudulent issues.

Our target in this project is to build up on the previously mentioned project by modifying the applied algorithms as follows:

- **K-Fold Cross validation:** is a popular technique used to validate the performance of our model. The model evaluation is done using the data set's different parts as a valid set. Then taking these iterations average results. This technique will be applied in all the algorithms to study its impact. This technique will be adopted in all algorithms in this project in comparison with the mentioned project that used the Stratified Shuffle Split function.
- **In the KNN-algorithm**: In this project, it will be tried in order to find the optimal K to build the model through enumerating from (K = 1 to K = 25) rather than just using only K = 3 which was used previously in the mentioned project.
- **Decision trees Pruning:** this method will be used to reduce the final tree complexity and therefore overfitting is reduced. And compare the results with those gained in the previously mentioned project which did not use the pruning strategy.
- **In the Random Forest Algorithm:** In each fold iteration, Random Forest will be created by more than one time by varying the following variables: maximum features, number of estimators and the maximum depth, then finding the corresponding model results to find the best combination of Random Tree variables. Instead of just train the data with the random forest classifier without changing any variable of the Random Forest as was done in the previously mentioned project.
- **Support Vector Machine Algorithm**: will be implemented in addition to the above algorithms the on the credit card transactions dataset and study its impact on the dataset. (This algorithm was not used in the previously mentioned project).

5.3. Implementation steps for Logistic Regression

- Importing the modules as Panda, NumPy, Matplotlib, LogisticRegression classifier, cross_val_score and confusion matrix.
- Importing the credit card transactions data, this dataset will allow us predict if it is a legit or a fraudulent transaction.
- Identifying if there is missing data to deal with, and in our case, there is no missing values.
- Split the data in to independent variables (the columns of the data that are used to make classification, the time, the amount, and V1 to V28), and the dependent variable that we want to predict (the class variable which indicates the type of the transaction, 0: Legit, 1 = Fraudulent).
- Stratified K Fold Cross Validation is used through the following steps:
- The dataset is split into K equal partitions or folds (10 folds in our case).
- In each iteration, one-fold is used for the testing set and the rest of folds for the training set.

- In each fold iteration, it is required to find an optimal K to get the best out of it, so we fit the model with all set of K values from (1 to 5) to identify which is the optimal K for each fold, then finding the corresponding model Accuracy, Precision, F1-score, recall.
- Use the average testing accuracy, average precision, average F1-Score and average recall as the estimate of the overall model evaluation metrics.
- Cross validation technique is used over the traditional train split function for the following reasons:

  a. More accurate.

  b. More "efficient" use of data, this is because each observation is used for training and testing.
- The above process is repeated for three techniques:
  - ➢ Random under sampling technique: where the number of legit transactions is reduced to 492 to match the number of fraudulent transactions.
  - ➢ SMOTE (Synthetic Minority Oversampling Technique): works by picking neighboring examples in the object plane, drawing a line between them, and drawing a new sample along that line.
  - ➢ Random Over Sampling technique: which involves randomly duplicating examples from the training dataset’s minority class.
- The average testing accuracy, average precision, average F1-Score and average recall of the three techniques are evaluated to know which technique works better with the Logistic Regression Classifier.
- All the results of the model metrics are discussed in the Results section.
- The code of the above Logistic Regression algorithm is available on the GitHub repository (https://github.com/JosephFares1991/CreditCardFraudDetectionProject/), in (Logistic_Regression.py)

5.4. Implementation steps for K-Neighbors Algorithm

- Importing the modules as Panda, NumPy, Matplotlib, KNeighbors classifier, cross_val_score and confusion matrix.
- Importing the credit card transactions data, this dataset will allow us predict if it is a legit or a fraudulent transaction.
- Identifying if there is missing data to deal with, and in our case, there is no missing values.
- Split the data in to independent variables (the columns of the data that are used to make classification, the time, the amount, and V1 to V28), and the dependent variable that we want to predict (the class variable which indicates the type of the transaction, 0: Legit, 1 = Fraudulent).

- Stratified K Fold Cross Validation is used through the following steps:
- The dataset is split into K equal partitions or folds (10 folds in our case).
- In each iteration, one-fold is used for the testing set and the rest of folds for the training set.
- In each fold iteration, it is required to find an optimal K to get the best out of it, so we fit the model with all set of K values from (1 to 5) to identify which is the optimal K for each fold, then finding the corresponding model Accuracy, Precision, F1-score, recall.
- Use the average testing accuracy, average precision, average F1-Score and average recall as the estimate of the overall model evaluation metrics.
- Cross validation technique is used over the traditional train split function for the following reasons:

  a. More accurate.

  b. More "efficient" use of data, this is because each observation is used for training and testing.

- The above process is repeated for three techniques:
  - ➢ Random under sampling technique: where the number of legit transactions is reduced to 492 to match the number of fraudulent transactions.
  - ➢ SMOTE (Synthetic Minority Oversampling Technique): works by picking neighboring examples in the object plane, drawing a line between them, and drawing a new sample along that line.
  - ➢ Random Over Sampling technique: which involves randomly duplicating examples from the training dataset's minority class.
- The average testing accuracy, average precision, average F1-Score and average recall of the three techniques are evaluated to know which technique works better with the KNN Classifier.
- The code of the above K-Neighbor algorithm is available on the GitHub repository (https://github.com/JosephFares1991/CreditCardFraudDetectionProject/), in (KNN.py).

5.5. Implementation steps for Decision Tree Algorithm

- Importing the modules as Panda, NumPy, Matplotlib, Decision Tree Classifier, Plot_tree, cross_val_score and confusion matrix.
- Importing the credit card transactions data, this dataset will allow us predict if it is a legit or a fraudulent transaction.
- Identifying if there is missing data to deal with, and in our case, there is no missing values.
- The data is split in to independent variables (the columns of the data that are used to make classification, the time, the amount, and V1 to V28), and the dependent variable that we want

to predict (the class variable which indicates the type of the transaction, 0: Legit, 1 = Fraudulent).

- Create a preliminary decision tree model and fit it to the training data.
- Plot the preliminary tree but it is very huge and the confusion matrix.
- Due to overfit of the training dataset, pruning the tree with cost complexity pruning which finds a small tree to improve the accuracy of the training dataset.
    - To find the optimal value of alpha, is to plot the accuracy of the tree as a function of different values of alpha and this will be done on the training and testing dataset.
    - To extract the different values of alpha that are available for the tree and build a pruned tree for each value of alpha using the (complexity_pruning_path (X_train, y_train)), and we omit the maximum value of alpha because it would prune all leaves and leaves only the root.
    - Then graph the accuracy of the training dataset and the testing dataset as a function of alpha. And from this graph we get the accuracy for the testing dataset that hits the maximum value.
- Then to know the best training and datasets, K-fold cross validation for finding the best value of alpha is used using the cross_val_score() function.
- And plotting the mean accuracies of each fold against the alpha values, and take the ideal alpha value which achieves the maximum accuracy.
- Use the ideal alpha from the last step to create the new tree and fit it to the corresponding training and datasets.
- The above process is repeated for three techniques:
    - Random under sampling technique: where the number of legit transactions is reduced to 492 to match the number of fraudulent transactions.
    - SMOTE (Synthetic Minority Oversampling Technique): works by picking neighboring examples in the object plane, drawing a line between them, and drawing a new sample along that line.
    - Random Over Sampling technique: which involves randomly duplicating examples from the training dataset's minority class.
- Use the above models to predict the average testing accuracy, average precision, average F1-Score and average recall of the three techniques are evaluated to know which technique works better with the Decision Tree Classifier.
- The code of the above Decision Tree algorithm is available on the GitHub repository (https://github.com/JosephFares1991/CreditCardFraudDetectionProject/), in (Decision_Tree.py).

5.6. Implementation steps for Random Forest Algorithm

- Importing the modules as Panda, NumPy, Matplotlib, Random Forest Classifier, cross_val_score and confusion matrix.
- Importing the credit card transactions data, this dataset will allow us predict if it is a legit or a fraudulent transaction.
- Identifying if there is missing data to deal with, and in our case, there is no missing values.
- The data is split in to independent variables (the columns of the data that are used to make classification, the time, the amount, and V1 to V28), and the dependent variable that we want to predict (the class variable which indicates the type of the transaction, 0: Legit, 1 = Fraudulent).
- Stratified K Fold Cross Validation is used through the following steps:
- The dataset is split into K equal partitions or folds (5 folds in our case).
- In each iteration, one-fold is used for the testing set and the rest of folds for the training set.
- In each fold iteration, Random Forest will be created by more than one time by varying the following variables, then finding the corresponding model Accuracy, Precision, F1-score, recall to find the best combination of Random Tree variables:
  - ***max_features:***
  - This is the highest number of features that Random Forest can try in a single tree. Python provides several options for assigning maximum features. Following are the used ones: Auto/None, Sqrt, Log2
  - ***n_estimators:***
  - This is the trees needed to be created before taking the most votes or prediction averages. The higher the performance as the number of trees increases, but the code becomes slower. it's better to decide on as high value because the processor can handle because this makes the predictions stronger and more stable, in our case we used the worth of 100.
  - ***Max_depth***
  - Represents the depth of every tree within the forest. as the tree is deeper, the greater number of splits it's then has and so it captures more info about the data. Each decision tree is fitted with depths starting from 1 to 32 then plotting both the training and test errors, we used (None, 2, 3, 4, 5).
  - Use the average testing accuracy, average precision, average F1-Score and average recall as the estimate of the overall model evaluation metrics.
  - Cross validation technique is used for the following reasons:

a. More accurate.

b. More "efficient" use of data, this is because each observation is used for training and testing.

- The above process is repeated for three techniques:
  - ➢ Random under sampling technique: where the number of legit transactions is reduced to 492 to match the number of fraudulent transactions.
  - ➢ SMOTE (Synthetic Minority Oversampling Technique): works by picking neighboring examples in the object plane, drawing a line between them, and drawing a new sample along that line.
  - ➢ Random Over Sampling technique: which involves randomly duplicating examples from the training dataset's minority class.
- The average testing accuracy, average precision, average F1-Score and average recall of the three techniques are evaluated to know which technique works better with the Random Forest Classifier.
- The code of the above Random Forest algorithm is available on the GitHub repository (https://github.com/JosephFares1991/CreditCardFraudDetectionProject/), in (Random Forest.py)

5.7. Implementation steps for Naïve Bayes Algorithm

- Importing the modules as Panda, NumPy, Matplotlib, GaussianNB classifier, cross_val_score and confusion matrix.
- Importing the credit card transactions data, this dataset will allow us predict if it is a legit or a fraudulent transaction.
- Identifying if there is missing data to deal with, and in our case, there is no missing values.
- The data is split in to independent variables (the columns of the data that will be used to make classification, the time, the amount, and V1 to V28), and the dependent variable that we want to predict (the class variable which indicates the type of the transaction, 0: Legit, 1 = Fraudulent).
- Stratified K Fold Cross Validation is used through the following steps:
- The dataset is split into K equal partitions or folds (5 folds in our case).
- In each iteration, one-fold is used for the testing set and the rest of folds for the training set.
- In each fold iteration, it is required to find an optimal K to get the best out of it, so we fit the model with all set of K values from (1 to 5) to identify which is the optimal K for each fold, then finding the corresponding model Accuracy, Precision, F1-score, recall.

- Use the average testing accuracy, average precision, average F1-Score and average recall as the estimate of the overall model evaluation metrics.
- Cross validation technique is used for the following reasons:

  a. More accurate.

  b. More "efficient" use of data, this is because each observation is used for training and testing.

- The above process is repeated for three techniques:

  - Random under sampling technique: where the number of legit transactions is reduced to 492 to match the number of fraudulent transactions.
  - SMOTE (Synthetic Minority Oversampling Technique): works by picking neighboring examples in the object plane, drawing a line between them, and drawing a new sample along that line.
  - Random Over Sampling technique: which involves randomly duplicating examples from the training dataset's minority class.

- The average testing accuracy, average precision, average F1-Score and average recall of the three techniques are evaluated to know which technique works better with the Naïve Bayes Classifier.
- The code of the above K-Neighbor algorithm is available on the GitHub repository (https://github.com/JosephFares1991/CreditCardFraudDetectionProject/), in (Naïve Bayes.py).

5.8. Implementation steps for Support Vector Machines Algorithm

- Importing the modules as Panda, NumPy, Matplotlib, svm classifier, cross_val_score and confusion matrix.
- Importing the credit card transactions data, this dataset will allow us predict if it is a legit or a fraudulent transaction.
- Identifying if there is missing data to deal with, and in our case, there is no missing values.
- The data is split in to independent variables (the columns of the data that will be used to make classification, the time, the amount, and V1 to V28), and the dependent variable that we want to predict (the class variable which indicates the type of the transaction, 0: Legit, 1 = Fraudulent).
- Stratified K Fold Cross Validation is used through the following steps:
- The dataset is split into K equal partitions or folds (5 folds in our case).
- In each iteration, one-fold is used for the testing set and the rest of folds for the training set.

- In each fold iteration, it is required to find an optimal K to get the best out of it, so we fit the model with all set of K values from (1 to 5) to identify which is the optimal K for each fold, then finding the corresponding model Accuracy, Precision, F1-score, recall.
- Use the average testing accuracy, average precision, average F1-Score and average recall as the estimate of the overall model evaluation metrics.
- The above process is repeated for three techniques:
    - ➢ Random under sampling technique: where the number of legit transactions is reduced to 492 to match the number of fraudulent transactions.
    - ➢ SMOTE (Synthetic Minority Oversampling Technique): works by picking neighboring examples in the object plane, drawing a line between them, and drawing a new sample along that line.
    - ➢ Random Over Sampling technique: which involves randomly duplicating examples from the training dataset's minority class.
- The average testing accuracy, average precision, average F1-Score and average recall of the three techniques are evaluated to know which technique works better with the Support Vector Machine Classifier.
- The code of the above algorithm is available on the GitHub repository (https://github.com/JosephFares1991/CreditCardFraudDetectionProject/), in (SVM.py)

6. Results:

Also, the results presentation idea resembles the method used in the project mentioned earlier from Victoria University, but we will present the actual results obtained from the algorithms by the methodology discussed above.

6.1. Under Sampling Technique

Classifier	**Accuracy**	**Precision**	**Recall**	**F1-Score**
Logistic Regression	93.393	96.487	90.053	93.123
KNN	63.819	64.524	61.983	63.162
Decision Tree	91.371	89.32	93.875	91.542
Random Forest	94.61	98.88	90.46	94.44
Naïve Bayes	84.954	98.048	71.354	81.954
SVM	86.767	97.961	74.576	83.115

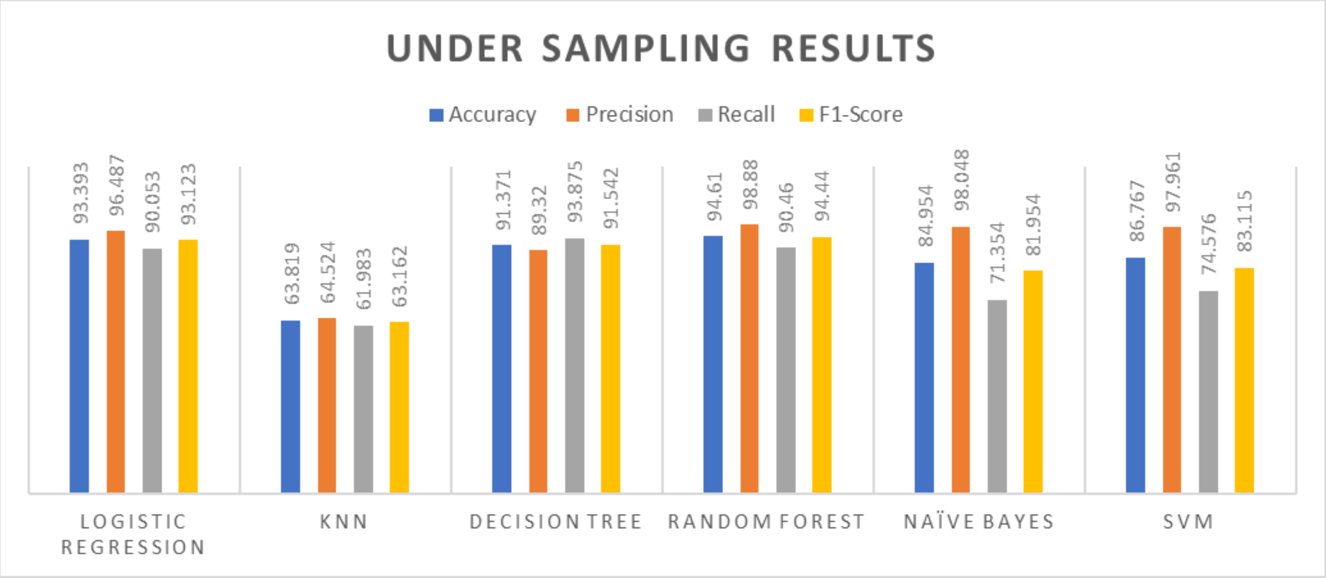

Figure 2: Under Sampling Technique Results

6.2. Over Sampling Technique

Classifier	Accuracy	Precision	Recall	F1-Score
Logistic Regression	95.269	6.553	88.608	11.361
KNN	99.819	45.167	20.735	28.341
Decision Tree	99.891	67.307	71.428	69.306
Random Forest	99.95	94.54	79.02	85.74
Naïve Bayes	98.996	11.618	72.127	19.903

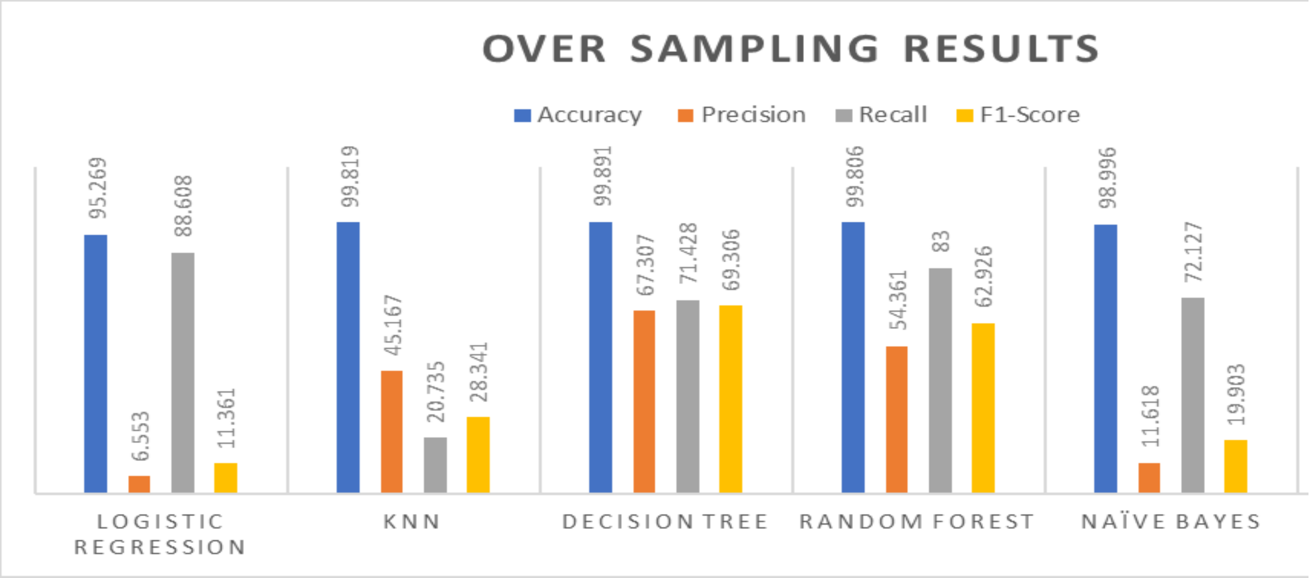

Figure 3: Over Sampling Technique results

6.3. SMOTE Technique

Classifier	Accuracy	Precision	Recall	F1-Score
Logistic Regression	96.464	14.181	87.581	19.809
KNN	97.204	3.146	50.832	59.261
Decision Tree	99.787	42.391	79.591	55.319
Random Forest	99.95	87	87.93	85.41
Naïve Bayes	99.187	14.091	72.319	23.473

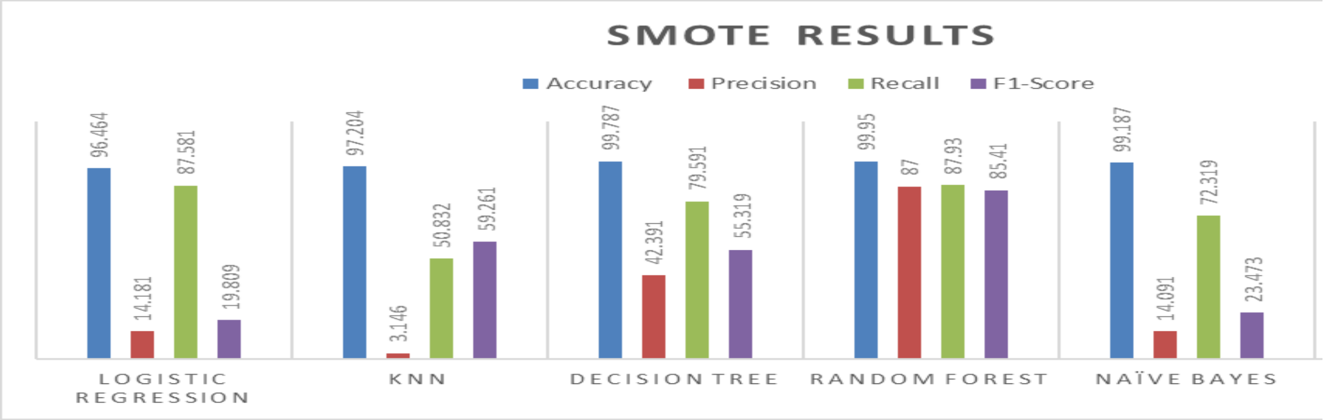

Figure 4: SMOTE Technique Results

6.4. Random Forest Accuracies, F1, Precisions, Recall with Under Sampling Technique

*Note: The data in each cell in the below table are written in format (Accuracy/F1-Score/Precession/Recall) due to table space boundaries.

Max Features / Max depth	**None**	**2**	**3**	**4**	**5**
1	93.59/93.34/98.17/89.02	91.05/90.29/99.50/82.69	91.46/90.82/99.26/83.77	91.76/91.12/100/83.73	92.68/92.19/99.76/85.76
Square Root	94.61/94.44/98.88/90.46	92.37/91.87/99.29/85.51	93.39/93.04/99.31/87.56	94.01/93.75/99.32/88.83	94.01/93.75/98.87/89.19
Logarithmic	94/93.77/98.64/89.41	92.37/91.83/100/84.95	93.29/92.92/99.31/87.34	93.59/93.29/99.32/88.02	94.01/93.74/99.32/88.79

6.5. Random Forest Accuracies, F1, Precisions, Recall with Over Sampling Technique

Max Features / Max depth	**None**	**2**	**3**	**4**	**5**
1	99.95/82.62/94.96/73.25	99.75/54.8/41.37/82.33	99.72/49.94/36.36/82.5	99.86/66.99/58.29/83.28	99.91/75.93/69.52/83.87
Square Root	99.95/85.75/94.54/79.02	99.72/52.23/36.59/84.12	99.69/48.26/35.16/85.35	99.68/50.51/34.42/86.42	99.79/59.06/43.93/86.62
Logarithmic	99.95/86.16/94.26/78.24	99.81/60.05/46.88/84.11	99.75/55.38/40.22/84.75	99.78/57.47/43.41/85.36	99.79/59.28/45.51/85.78

6.6. Random Forest Accuracies, F1, Precisions, Recall with SMOTE Technique

Max Features / Max depth	None	2	3	4	5
1	99.94/85.24/88.27/83.07	99.82/61.46/49.11/82.32	99.8/57.45/43.96/82.7	99.87/70.37/60.89/83.08	99.88/71.63/63.13/83.47
Square Root	99.95/85.41/87/83.93	99.65/44.75/30.91/84.35	99.56/40.22/26.01/84.79	99.48/36.12/23.27/85.18	99.52/36.79/23.61/86.23
Logarithmic	99.95/85.24/86.86/83.69	99.78/57.38/42.98/83.08	99.61/43.34/29/84.35	99.47/36.42/23.02/85.19	99.53/39.1/26.52/85.82

7. Discussion, Conclusion and Future Work:

- As mentioned above, this project adopted some methods that was not used in the previously mentioned project like the K-Fold Cross Validation technique, also adopted the decision tree pruning to reduce the tree overfitting, as well as changing the variables of the Random Forest as: maximum features, number of estimators and the maximum depth. Also, the SVM algorithms was adopted in this project.
- In the Under Sampling Technique, The Logistic Regression algorithm achieved the highest results while KNN achieved the lowest results in terms of Accuracy, Precision, Recall and F1-score.
- In the Over Sampling Technique, The Random Forest with max_features settings set to None and Max depth setting set to square root algorithm achieved the highest results terms of Accuracy, Precision, Recall and F1-score, while the SVM achieved nearly the good results in terms of Precision, Recall and F1-score but the lowest accuracy.
- In the SMOTE Technique, The Random Forest with max_features settings set to None and Max depth setting set to square root algorithm achieved the highest results terms of Accuracy, Precision, Recall and F1-score.
- The used method of K-fold Cross validation has much improved the Precision, Recall and F1-Score for all used algorithms compared with mentioned paper “Supervised Machine Learning Algorithms for Credit Card Fraudulent Transaction Detection: A Comparative survey” (Rahman, 2021) in the Under Sampling technique.
- Regarding the Under Sampling Technique, the results of KNN (Accuracy and Recall) is less than the results in the mentioned paper.
- Regarding the SMOTE method, the results of the used method of setting the depth of the tree to the square root of the total depth has better results in the sensitivity and F1-Score.
- SVM works well with the Under Sampling technique, but is not suitable with the Over Sampling and SMOTE techniques, as the complexity of algorithm’s training mainly depends on the dataset size, meaning the training time grows as the dataset gets larger to a point where it becomes infeasible to train. And this was proved in the SVM algorithm in the Over Sampling and SMOTE techniques where it took more than two hours in training phase without returning any result.

- To conclude, Random Forests has the best results among all other algorithms in all techniques (Under Sampling, Over Sampling and SMOTE).
- For future work, deep learning algorithms can be adopted to tackle the problem of credit card fraud detection and examine the performance if these deep learning algorithms against the machine learning algorithms.

8. Bibliography

(n.d.). Retrieved from https://www.ftc.gov/news-events/press-releases/2019/02/imposter-scams-top-complaints-made-ftc-2018.

[Online]. (22-Sep-2016). Supervised and Unsupervised Machine Learning Algorithms. *Machine Learning Mastery*.

A. M. Mahmood, P. G. (2010). A New Pruning Approach For Better and Compact Decision Trees. Vol.02, No.08, Pages 2551-2558.

A. Mishra, C. G. (2018). Credit Card Fraud Detection on the Skewed Data Using Various Classification and Ensemble Techniques. *IEEE International Students' Conference on Electrical, Electronics and Computer Science (SCEECS)*, pp. 1-5.

A. Pumsirirat, L. Y. (2018). “Credit Card Fraud Detection using Deep Learning based on Auto-Encoder and Restricted Boltzmann Machine. *International journal of advanced computer science and applications*, 9(1), pp. 18-25.

A. Singh, A. J. (2019). An Empirical Study of AML Approach for Credit Card Fraud Detection–Financial Transactions. *INTERNATIONAL JOURNAL OF COMPUTERS COMMUNICATIONS & CONTROL*, ISSN 1841-9836, e-ISSN 1841-9844, 14(6), 670-690.

Abdallah, A., Maarof, M. A., & Zainal, A. (2016). Fraud detection system: A survey. *Journal of Network and Computer Applications*, 68, 90-113.

Anuruddha Thennakoon, S. M. (July 2019). Real-time Credit Card Fraud Detection Using Machine Learning.

Bahnsen, A. C. (2014). Improving credit card fraud detection with calibrated probabilities. *In Proceedings of the 2014 SIAM International Conference on Data Mining*, (pp. 677-685).

Bhattacharyya, S. J. (2011). Data mining for credit card fraud: A comparative study. *Decision Support Systems,*, 50(3), 602-613.

Bolton, R. J. (n.d.). Unsupervised profiling methods for fraud detection. *Conference on Credit Scoring and Credit Control.* Edinburgh.

Burges, C. J. (1998). A tutorial on support vector machines for pattern recognition. *Data mining and knowledge discovery*, 2(2), 121-167.

C. Wang, Y. W. (2018). “Credit card fraud detection based on whale algorithm optimized BP neural network. *13th International Conference on Computer Science & Education (ICCSE)*, pp. 1-4. IEEE.

F. Esposito, D. M. (1999). The Effects of Pruning Methods on the Predictive Accuracy of Induced Decision Trees. *Applied Stochastic Models in Business and Industry*, Pages 277-299.

F. Esposito, D. Malerba, G. Semeraro, and V. Tamma. (1999). The Effects of Pruning Methods on the Predictive Accuracy of Induced Decision Trees. *Applied Stochastic Models in Business and Industry*, Pages 277-299.

F. Ghobadi, M. R. (2016). Cost Sensitive Modeling of Credit Card Fraud using Neural Network strategy”, 2016 Signal Processing and Intelligent Systems (ICSPIS). *International Conference of pp. 1-5. IEEE.*

Ferrier, J. P. (2000). Evaluating the predictive performance of habitat models developed using logistic regression. *Ecological modelling*, 133(3), 225-245.

G. E. Melo-Acosta, F. D.-M.-L. (2017). Fraud detection in big data using supervised and semi-supervised learning techniques. *Commm. Comput. (COLCOM), 2017 IEEE Colomb. Conf.*, pp 1-6.

G. Liu, W. Z. (2018). A new FDS for credit card fraud detection based on behavior certificate.

Gianey, R. C. (2017). Comprehensive Review on Supervised Machine Learning Algorithms. *2017 Int. Conf. Mach. Learn. Data Sci*, pp 37-43.

Gonzalez, J. S. (n.d.). Credit card fraud and ID theft statistics. *CreditCards.com. [Online]*.

Hauska, P. H. (1977). The decision tree classifier: Design and potential. *IEEE Transactions on Geoscience*, 15(3), 142-147.

Ho, T. K. (1995, a). Random decision forests. *In Proceedings of 3rd international conference on document analysis and recognition*.

Ho, T. K. (1995, August). Random decision forests. *In Proceedings of 3rd international conference on document analysis and ecognition*, pp. Vol. 1, pp. 278-282.

https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html. (n.d.).

J. Chen, X. Wang, and J. Zhai. (2009). Pruning Decision Tree Using Genetic Algorithms. *International Conference on Artificial Intelligence and Computational Intelligence*, pp 244-248.

J. O. Awoyemi, A. O. (2017). Credit card fraud detection using Machine Learning Techniques: A Comparative Analysis. *Computing Networking and Informatics (ICCNI)*, International Conference on pp. 1-9. IEEE.

J.O. Awoyemi, A. O. (2017). Credit Card Fraud Detection using machine learning techniques: A Comparative Analysis. *2017 Int. Conf. Comput. Netw. Informatics*, pp 1-9.

Jha, S., Guillen, M., & Westland, J. C. (2012). Employing transaction aggregation strategy to detect credit card fraud. *Expert systems with applications*, 39(16), 12650-12657.

K. Randhawa, C. K. (2018). Credit Card Fraud Detection using AdaBoost and majority voting. *IEEE Access*, vol. XX, pp.1-1, 2018.

K.Ratna Sree Valli , P.Jyothi , G.Varun Sai , R.Rohith Sai Subash. (2020). Credit card fraud detection using Machine learning algorithms. *Journal of Research in Humanities and Social Science*, Volume 8 ~ Issue 2 pp.: 04-11.

Kou, Y. L.-T.-P. (March 21-23, 2004). Survey of Fraud Detection Techniques. *IEEE International Conference on Networking, Sensing & Control.* Taipei, Taiwan.

L. Rokach, a. O. (2008). Data Mining With Decision Trees: Theory and Applications. *Series in Machine Perception and Artificial Intelligence, World Scientific Publishing, Singapore*.

Learning – Towards Data Science. [online] Available at:. ([Accessed 19 Jan. 2019].). *https://towardsdatascience.com/deep-learning-vs-classical-machinelearning*.

M. F. Zeager, A. S. (2017). Adversarial learning in credit card fraud detection. *Syst. Inf. Eng. Des. Symp.*, pp 112-116.

M. L. Zhang and Z. H. Zhou. (2007). A lazy learning approach to multi-label learning. Pattern recognition. 40(7), 2038-2048.

Maes, S. T. (2002). Credit card fraud detection using Bayesian and neural networks. *Proceeding International NAISO Congress on Neuro Fuzzy Technologies*.

Mirjana Karanovic, S. S. (2019). Credit Card Fraud Detection - Machine Learning methods. *ResearchGate*.

Mohammed, E. a. (1 July 2018). Supervised Machine Learning Algorithms for Credit Card Fraudulent Transaction Detection: A Comparative Study. *IEEE Annals of the History of Computing, IEEE*, doi.ieeecomputersociety.org/10.1109/IRI.2018.00025.

Mrs. C. Navamani, M. P. (n.d.). Credit Card Nearest Neighbor Based Outlier Detection Techniques.

N. Kalaiselvi, S. R. (2018). Credit card fraud detection using learning to rank approach. *International Conference on Computation of Power, Energy, Information and Communication (ICCPEIC) ional conference on computation of power, energy, Information and Communication (ICCPEIC)*, pp. 191-196. IEEE.

N. Malini, D. M. (2017). Analysis on Credit Card Fraud Identification Techniques based on KNN and Outlier Detection, Advances in Electrical, Electronics, Information, Communication and Bio-Informatics (AEEICB). *Third International Conference*, pp 255-258.

Ogwueleka, F. N. (2011). Data Mining Application in Credit Card Fraud Detection System. *Journal of Engineering Science and Technology*, Vol. 6, No. 3, pp. 311 – 322.

Olawale Adepoju, J. W. (October 2019). Comparative Evaluation of Credit Card Fraud Detection Using Machine. *ResearchGate*.

Omar, A. H. (2012). A comparative study of Reduced Error Pruning method in decision tree algorithms. *IEEE International Conference on Control System, Computing and Engineering*.

P. Suraj, N. Varsha and S. P. Kumar. (2018). Predictive modelling for credit card fraud detection using data analytics. *Procedia Computer Science*, 132, 385-395.

Patil, S. S. (2015). Credit Card Fraud Detection Using Decision Tree Induction Algorithm. *International Journal of Computer Science and Mobile Computing (IJCSMC).*, Vol.4, Issue 4, pp. 92-95, ISSN: 2320-088X.

Pooja Tiwari, S. M. (2021). *CREDIT CARD FRAUD DETECTION USING MACHINE LEARNING.*

Pumsirirat, A. a. (2018). Credit Card Fraud Detection using Deep Learning based on Auto-Encoder and Restricted Boltzmann Machine. *International Journal of Advanced Computer Science and Applications*, 9(1).

Quinlan, J. R. (1993). C4.5:Programs for machine learning. *Morgan Kaufmann Publishers Inc., California*.

Rahman, R. (2021). Supervised Machine Learning Algorithms for Credit Card Fraudulent Transaction Detection: A Comparative survey. *University of Victoria*.

RamaKalyani, K. a. (2012). Fraud Detection of Credit Card Payment System by Genetic Algorithm. *International Journal of Scientific & Engineering Research*, Vol. 3, Issue 7, pp. 1 – 6, ISSN 2229-5518.

Randhawa, K. e. (2018). Credit Card Fraud Detection Using AdaBoost and Majority Voting. *IEEE Access*, vol. 6, 2018, pp. 14277–14284., doi:10.1109/access.2018.2806420.

Robertson, D. (n.d.). Investment &amp; a.

Robertson, D. (September 2016). Investments &amp; Acquisitions. *Top Card Issues in Asia-Pacific Card Fraud Losses Reach $21.84 Bilion*, Nilson Rep., no.1096, 1090.

S. Bhattacharyya, S. Jha, K. Tharakunnel, J. Westland. (2011). Data mining for credit card fraud: A comparative study. *Decision Support Systems*, 50, 602-613.

S. Bhattacharyya, S. Jha, K. Tharakunnel, J. Westland. (2011). Data mining for credit card fraud: A comparative study, Decision Support Systems. 50, 602-613.

S. Dhankhad, B. F. (2018). Supervised Machine Learning Algorithms for Credit Card Fraudulent Transaction Detection: A Comparative Study. *IEEE International Conference on Information Reuse and Integration (IRI)*, pp. 122-125. IEEE.

S. Drazin, and M. Montag. (n.d.). Decision Tree Analysis using Weka. *Machine Learning-Project II, University of Miami.*

S. V. S. S. Lakshmi, S. D. (n.d.). Machine Learning For Credit Card Fraud Detection System",unpublished.

Sahin, Y., Bulkan, S., & Duman, E. (2013). A cost-sensitive decision tree approach for fraud detection. *Expert Systems with Applications*, 40(15), 5916-5923.

Seeja, K., & Zareapoor, M. (2014). Fraudminer: a novel credit card fraud detection model based on frequent item set mining. *The Scientific World Journal, ID 252797*.

Singh, G. G. (2012). A Machine Learning Approach for Detection of Fraud based on SVM. *International Journal of Scientific Engineering and Technology.*, Volume No.1, Issue No.3, pp. 194-198, ISSN: 2277-1581.

Sorournejad, S., Zojaji, Z., Atani, R. E., & Monadjemi, A. H. (2016). A survey of credit card fraud detection techniques:Data and technique oriented perspective. *arXiv preprint arXiv:1611.06439*.

Sudha, T. R. (2017). Credit Card Fraud Detection in Internet using K Nearest Neighbour Algorithm. *IPASJ international journal of computer science*, vol. 5, no. 1.

Supervised and Unsupervised Machine Learning Algorithms, Machine Learning Mastery. (2016, Sep 22).

T.Cody, S. a. (2018). A Utilitarian Approach to Adversarial Learning in Credit Card Fraud Detection. pp 237-242.

Vaishnavi Nath Dornadulaa, G. S. (2019). Credit Card Fraud Detection using Machine Learning Algorithms. *NTERNATIONAL CONFERENCE ON RECENT TRENDS IN ADVANCED COMPUTING*.

What is Machine Learning? A definition, Expert System. (05-Oct-2017). *[Online]*.

Xuan, S. e. (2018). Random Forest for Credit Card Fraud Detection. *IEEE 15th International Conference on Networking, Sensing and Control (ICNSC)*, doi:10.1109/icnsc.2018.8361343.

Y. Sahin, S. B. (2013). A cost-sensitive decision tree approach for fraud detection. *Expert Syst. Appl.*, 40, 5916-5923.

Z. Kazemi, H. Z. (2017). Using deep networks for fraud detection in the credit card transactions. *Knowledge-Based Engineering and Innovation (KBEI), 2017 IEEE 4th International Conference*, pp. 630-633. IEEE.

Zareapoor, M., Seeja, K., & Alam, M. A. (2012). Analysis on credit card fraud detection techniques: Based on certain design criteria. *International Journal of Computer Applications*, 52 (3), 35-42.