



Credit Card Fraud Detection with Machine Learning

Author: Sivadeep kotha

Venkat Sai sheelam

Vineela Pedaapudi

A dissertation submitted in partial fulfilment of
the requirements for the degree of:

Master of Computer Science – Data science

Oklahoma City university

Abstract

As the number of credit card transactions keeps growing and represents an increasing share of the European payment system. Leading to several stolen account numbers and subsequent losses to banks, also people believed that credit card transaction fraud is a growing threat with severe implications for the financial industry. Data mining plays a crucial role in detecting credit card fraud in both online and offline transactions. Credit card fraud detection which is a data mining problem becomes challenging for two main reasons. First, the characteristics of normal and fraudulent behavior are continually changing, and second, the credit card fraud dataset is highly asymmetric. The performance of fraud detection in credit card transactions is greatly affected by the sampling method of the dataset and the choice of variables and the detection techniques used. This paper investigates the performance of logistic regression (LR), k-nearest-neighbor (KNN), naive bayes, Random Forest, Decision Tree (DT) and on credit card fraud data. The dataset of credit card transactions obtained from European cardholders contains 284,807 transactions. A mixture of under-sampling and over-sampling techniques applied to the unbalanced data. The five strategies were used to the raw and pre-processed data, respectively. This work is implemented in Python. The performance of the methods was evaluated based on accuracy, sensitivity, specificity, precision and recall rate. The results are shown in comparison. After cross-validation, the accuracy of the best classification algorithm model in, k nearest neighbor, naïve bayes, Random Forest, logistic regression and is 97.92% which using the over-sampling model.

Acknowledgements

We would like sincerely to thank the author from the Kaggle platform which offers the dataset. Furthermore, this dataset collected and analyzed during a research collaboration of Worldline and the Machine Learning Group (<http://mlg.ulb.ac.be>) of ULB (Université Libre de Bruxelles) on big data mining and fraud detection who collects credit card transaction data for providing with sample reference content as well as advice.

Thank you to Dr Yuhua Li. The supervisor and the tractor for his unlimited support guidance and the external device. Thre is no doubt that with the unique idea throughout all phases of this dissertation, the outhouse of this dissertation became more successful.

Contents

1. Introduction	5
1.1 Research motivation	5
1.2 Research Problem	6
1.3 Research aims and objectives	7
1.4 Target audience	7
1.5 Dissertation Outline	8
2 Background and literature review	9
2.1 Credit Card Fraud definition	9
2.2 Credit card fraud anatomy	9
2.3Credit Card Fraud detection	10
2.4 Credit Card Fraud identification	11
2.5 Consequences of credit card fraud	12
2.6 Fraud countermeasures	13
2.6.1 General countermeasures	13
2.6.2 User training and education	13
2.6.3 Government legislation	13
2.7 Machine learning classifiers	14
2.7.1 Logistic regression algorithm	14
2.7.2 Decision tree(DT)	16
2.7.3 k-nearest neighbour (KNN)	17
2.7.4 Random Forest	18
2.7.5 Naives bayes	19
2.8 Synthetic Minority Over-sampling Technique(SMOTE)	21
2.9 Python libraries for data science and machine learning	23
2.10 Related works	24
3 Dataset pre-processing and features extraction	25
3.1 Dataset preparation and pre-processing	25
3.2 General components of the credit card dataset	25
3.3 Summary of Factors Influencing Choice of Project	31
4 Implementation	32
4.1 Segmentation/reservation of the original training set	32

4.2 Sampling process	32
4.2.1 Undersampling	32
4.2.2 Oversampling	33
4.3 Individual classifiers evaluation	35
4.4 Result and Conclusion	42
5 Future work	47
6 Conclusion	48
7 Reflection on learning	
Reference	50

List of Figures

Figure 1: Logical function expressions	15
Figure 2: The Cost and J functions	15
Figure 3: The process function of updating θ	16
Figure 4: A simple decision tree	17
Figure 5: The formula for calculating the distance between two points	18
Figure 6: The k-nearest neighbor sample.....	18
Figure 7: Geometric Margin	19
Figure 8: The example of SMOTE formation sample	22
Figure 9: check data empty result	27
Figure 10: Number of categories of inspection samples	27
Figure 11: A scatter plot of time versus amount	29
Figure 12: The part of the similar shape of the distribution.....	30
Figure 13: Sample classification	33
Figure 14: Parameter setting code for each model.....	36
Figure 15: The learning curve of the models	40
Figure 16: Logistic Regression with oversample & undersample	40

List of Tables

Table 1: Original dataset picture	27
Table 2: Attributes of the European dataset	27
Table 3: First training accuracy	30

1. Introduction

1.1 Research motivation

Over the past months of study, we have been following the news on information security and financial fraud as it is essential to all online and offline financial transaction systems. Although fraudulent transactions account for a relatively small percentage of most medium credit card transactions, as soon as a customer is unfortunate enough to have a credit card transaction, the loss of money to the business and a crisis of trust for the customer can ensue. Some reports show that Credit card fraud can easily accomplish their purpose. Large amounts of money can transact in a short period without any indication of risk or the owner's permission. Every fraudulent transaction can be legitimized by a fraudster's operation which makes fraud very challenging and difficult to detect (Dornadula and Geetha 2019). As a result, we are sufficiently motivated to improve credit card fraud detection by training a pass-through machine learning classification method. The final purpose is to help this project to select a better model. The banks want to detect credit card transactions and quickly predict whether the trade is risky, regulators need to delay or hold the transaction, and the marketing needs to be blocked the next time the card is used a lot. We think we have ambitions to complete the fraud detection project. Besides, we hope we achieve an opportunity to realize the need for improved customer detection capabilities.

1.2 Research Problem

The popularity of credit cards has greatly facilitated transactions for both merchants and users but it has also led to many cases of fraud. Credit card fraud is a general term for fraud committed using a payment card such as a credit or debit card. The purpose may be to obtain goods or services or to make a payment to another account controlled by the offender.

There are two types of fraud on the market today. Card-present fraud is now less common than buying the other kind of deception, and absent card fraud is currently widespread. They may execute in many ways, usually occurring without the cardholder's knowledge. The maintenance of the security of the Internet database has always been a big problem. A slight leak will cause the threat of stolen card information on the user's account.

Billions of euros are lost each year globally due to credit card fraud. In 2018, unauthorized financial fraud losses from payment cards and remote banking in the UK totalled £844.8 million. Given that banks and card companies prevented £1.66 billion of unauthorised fraud in 2018. It equates to £2 out of every £3 prevented from £3 of fraud attempts (Worobec 2019).

However, traditional credit card fraud-detection models such as manual detection, expert rules, cost analysis models. For example, they might have shortcomings such as low detection accuracy, long detection time, and high maintenance costs. Therefore, financial institutions urgently need a well-designed fraud detection system to detect credit card fraud.

1.3 Research aims and objectives

The objective of this research is to evaluate the performance of the user's fraud detection model using different supervised machine algorithms to obtain a higher detection accuracy by comparing other methods, also to achieve this goal of the detection capability improvement, several objectives considered.

- Conduct regressions by collecting relevant research to identify problems with the current system in place, learn from the good experiences of other research, and also examine shortcomings
- Establish a pre-processed implementation scheme to extract useful information and standardise the data by analysing—the existing dataset.
- Use Sampling Method (SMOTE) which contains over-sampling and under-sampling to solve the problem of highly unbalanced credit card data sets faced by the current fraud detection system.
- Evaluate the effectiveness of fraud detection by calculating the detection accuracy of different machine learning classification algorithms (SVM, logistic regression, decision tree, KNN and catboost) and compare the results of this study in various aspects.
- After finding the best detection suitable for credit card spoofing detection, we can experimentally prove it by cross-validation and other methods.
- Multiple experiments using real and simulated datasets were conducted to compare the proposed four solutions with other detection schemes, and the experiments prove the superiority of our proposed solution.

1.4 Target audience

The target audience of these research groups as well as individuals interested in research in the field of financial security are based on machine learning classification algorithms. In particular, the detection and prevention of transaction fraud detection.

1.5 Dissertation Outline

This dissertation consists of the 8th chapters as follows:

Chapter 2 background and literature review: According to the description of the fraudulent credit card transaction provided. We describe credit card fraud in terms of the type and manner and even include an explanation of how credit card detection and identification are related to accounts.

At the same time, the queried data and references investigate whether this task can accomplish by using different types of machine learning Python libraries and classification models.

Chapter 3 Dataset pre-processing and features extraction: This chapter describes how the data set was prepared for this study and how to pre-process the data set. Includes the ability to select categorical features and explain how to track those features.

Chapter 4 Implementation: Calculate the accuracy of different machine learning-based classification algorithms, recall and check comparison using other datasets. Then obtain the model of the optimal detection algorithm.

Chapter 5 Comparative analysis: This chapter focuses on comparing the result for truth in this study with the other tasks.

Chapter 6 future work: Suggest how the project in the current study should be improved upon by researchers in future.

Chapter 7 Conclusion: Conclude the study in a summarised finding.

Chapter 8 Reflections on learning: Focus on Presenting reflection on what has learnt by conducting this study.

2 Background and literature review

2.1 Credit Card Fraud definition

Initially, we need to understand is: *why is it that modern detection systems, anti-fraud detection, are so complicated?*

The modern detection system, also we call it Anti-fraud programs, for most customers or owners, they probably do not have a clear definition of the credit card fraud. In other words, the purpose of fraud is vague. On a small scale, anti-fraud seems to be a dichotomous problem. However, after repeated deliberation, we found that it is a multi-classification problem because each type of fraud can be treated as a different type (Ogwueleka and Technology 2011). Besides, the single kind of fraud does not exist, and the means of the second phase fraud is always changing. Even now, most of our customers, banks and insurance companies are perennial victims of fraud. They must continually try to update their prediction system. Rather than betting on the same model, so the fraud detection is also facing this challenge right now.

2.2 Credit card fraud anatomy

- The dataset which we use is labelled so that we can use the more mature supervised learning, but there is a disadvantage that it will be slower to update over time.

- There is a significant risk of supervised learning with labels, the model learned from such historical data can only detect frauds that are similar to historical fraud.

To accomplish the task of improving the accuracy of credit card detection, we may need some research to deal with the tags and characteristics of the information we collect, and we may need to do data mining to find information that is beneficial to us.

2.3 Credit Card Fraud detection

Credit card fraud detection is designed to prevent any unauthorized credit card transactions from fraudsters and to recover losses and credibility for customers and businesses. Although there are better financial mechanisms, the fraudster is continually updating his techniques. Also, it makes the anti-credit card fraud techniques very challenging; the standard anti-credit card fraud methods available in the market today are listed below.

- Validation method through merchant trade

The merchants often require a complete list of receipts to identify the user and have added tokenisation techniques to protect credit card information by using the referenced card number instead of the current card number. It can make sure that they offer additional information like a PIN, zip code or card security code. Also, they may be requested to show them during the merchant transaction, and they are currently used by merchants to combat fraud (Contributors 2020).

- Geolocation of transactions by IP address

Geolocation technology provides an absolute geographic location through the IP address of the computer where the order is placed in a real-time e-commerce transaction which can identify areas with a high potential for fraud. It might allow merchants to attach authentication to transaction applications that vary widely in realistic examples to protect them from credit card fraud (FTC.gov 2012).

- Detect IP address countries and whether they are high-risk areas.

Detection system makes sure that the IP address country is the same as the billing address country. By using a fraud prevention service, the service can detect the IP address country for the customer placing the order. If the customer's billing and shipping address are in the UK (Duman et al. 2013) but the person placing the order logged in from a Russian IP address, a more rigorous review is required, and anti-fraud precautions are often triggered. It is also always needed that orders shipped to international addresses scrutinised if the card or shipping address is in an area prone to credit card fraud.

- Using Neural Networks to Detect Credit Card Payment Fraud

Most of the existing techniques are based on deep learning and oversampling algorithms for credit card fraud detection. The Long Short Term Memory Networks (LSTM) fraud detection model for serial classification of transaction data and integration of synthetic minority class oversampling. The Smote and the k-Nearest Neighbor (kNN) classification algorithm design and build a kNN-Smote-LSTM based fraud detection network model which can Improve fraud detection performance by continuously filtering out security-generating samples through kNN discriminant classifiers (Maes et al. 2002).

- Machine learning detection

They are using Machine Learning Classification Algorithms to Detect Credit Card Fraud. Machine learning is a very effective way to detect fraudulent transactions if his performance is good enough because he determined by choice of features, the training of the data drink testing, and the classification methods of machine learning. All of these factors contribute to different generation rates. Many studies have shown that using machine learning classification algorithms to detect credit card fraud has resulted in better accuracy. They have also compared the results of different algorithms and other studies and agreed that machine learning detection is the right choice.

2.4Credit Card Fraud identification

The identification of credit card fraud detection is currently facing challenging because of most people not familiar with credit card fraud. After all, most of the scam comes out through the valid pathway following the banks as well as financial companies, and the only difference is that they are unauthorized third-party pathways. The recent credit fraud, as well as becoming more challenging to identify. Because if there is anyone who knows their credit card number, as well as expiration date,he can make a transaction on the website without their permission. Fraudsters will get more.

information about people's finances, and they will also have more opportunities to make fraudulent transactions by swiping credit cards, rather than just the ones we see.

2.5 Consequences of credit card fraud

Credit card fraud and process directly concern the user and the financial company; it is a reason we keep focus credit card fraud this year. The following are examples of fraud transaction outcomes.

- Economic losses to users and businesses
- Customer Personal Information Breach and Corporate Disclosure
Enterprise trust crisis in information security

While there are many existing channels and methods to prevent credit card fraud, these continue to impact our financial transaction marketplace and our customers' experience. From July 2005 to mid-January 2007, there are more than 45.6 million credit cards disclosure because the TJX's systems exposed data. Also, Albert Gonzalez was accused of being the leader of the organization responsible for the theft. In August 2009, Gonzalez was also indicted for the most massive known credit card theft to date-information from more than 130 million credit and debit cards stolen from Heartland Payment Systems, retailers 7-11 and Hannaford Brothers, and two unidentified companies (Tan 2020).

On 15 May 2016, a group of about 100 people used data from 1,600 South African credit cards to steal \$12.7 million from 1,400 convenience stores in Tokyo over three hours (McCurry 2016). It believed that by acting on Sundays and in countries/territories other than the issuing banks, they gained enough time to leave Japan before winning the robbery in a coordinated attack.

2.6 Fraud countermeasures

2.6.1 General countermeasures

The general countermeasures are increasing protection of customer transactions. For instance, they are adding signs to direct cardholders to designated areas. Every cardholder in the self-service program should be protected accordingly, protecting ATMs and bank assets from unauthorised use. Protected areas for any transaction can be monitored through the bank's CCTV system. Also, Cards use CHIP identification to reduce the likelihood of card theft. (Little 2009)

2.6.2 User training and education

Customers and cardholders should be educated about credit cards and how to use the reporting system to report card loss or theft to avoid fraud. Customers need to check their billing charges regularly and report unauthorised transactions immediately to their banks and financial centres if they find any irregularities. It recommended that customers install virus protection software on their computers and only record account numbers, expiration dates and the telephone number and address of each company in a secure location (NFCC.gov 2019). It should be careful when the users use credit cards for online purchases, especially on untrusted websites such as phishing sites. Customers are advised not to send credit card information via unencrypted emails and to keep the habit of not retaining PINs when using credit cards.

2.6.3 Government legislation

The identification of credit card fraud also requires the help of our government regulators. For example, the enactment laws of consumer protection related to card fraud transaction. It will help to optimise the market environment and ensure the safety of the credit card transaction market. Also based on the EU GDPR principles, any company and card issuers should publish standards, guidelines and codes to protect cardholder information and monitor fraudulent activities, or be fined (Foulsham 2019).

2.7 Machine learning classifiers

In this project, we used a total of five classifications methods (**Logistic regression, KNN, Decision tree (DT), Naïve Bayes , Random Forest**). These classification algorithm methods are widely used for problems such as differential training datasets. Also, it is commonly used in classification learning. That is the reason I compare them in the same training dataset. Also, it can be a cross-sectional comparison with other current studies in the results.

2.7.1 Logistic regression algorithm

Use logistic regression to detect credit card fraud. Logistic regression is the classical and the best bicategorical algorithm, which is preferred when dealing with classification problems, especially bicategorical ones. The choice of algorithm is based on the principle of simplicity before complexity. Logistic regression is also an excellent choice because it is a recognised statistical method used to predict the outcome of a binomial or polynomial. A multinomial logistic regression algorithm can regenerate the model. It will be a better classification algorithm when the target field or data is a set field with two or more possible values.

The advantage of logistic regression is that he is faster to process and is suitable for bicategorical problems. It is also more straightforward for any beginner to understand and directly see the weights of each feature. Then it is easier to update the model and incorporate new data for different problems (Aihua et al. 2007). Furthermore, it has a disadvantage. There is a limit to the data and the adaptability of the scene. Not as adaptable as the decision tree algorithm. But this is an issue that we can also determine in this project based on the actual situation whether the logistic regression has a better ability to adapt to an extensive data set of credit card transactions (Ng and Jordan 2002).

The main methods of logistic regression method:

Objective: It is to look for some risk factor, then in this project, they want to find a particular transaction factor or reasons that are suspected of being fraudulent.

Prediction: Predicting the probability of fraud under other independent variables, based on different algorithmic models.

Judgment: It is somewhat similar to prediction. It is also based on different models to see how likely it is that a transaction is a risk factor in a situation where fraud falls into a specific category.

Regression General Steps

- Finding the h-function (i.e., the prediction function)

Constructing the predictive function $h(x)$, the logistic function, or also known as the sigmoid function, we are generally the first step is to build the predictive process, where the training data for the vector, as well as the best parameters. The basic form of the function shown in figure 1

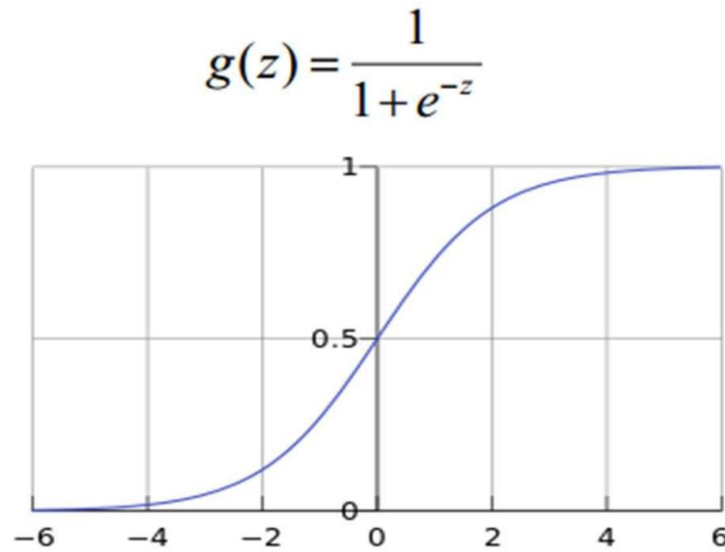


Figure 1: Logical function expressions

- Constructing the J-function (loss function)

The second step is that we need to construct the loss function-j. In general, there will be m samples, each with n characteristics. The Cost and J functions are as follows, and they are derived based on maximum likelihood estimation (Sahin and Duman 2011).

$$Cost(h_{\theta}(x), y) = \begin{cases} -\log(h_{\theta}(x)) & \text{if } y = 1 \\ -\log(1 - h_{\theta}(x)) & \text{if } y = 0 \end{cases}$$

$$J(\theta) = \frac{1}{m} \sum_{i=1}^m Cost(h_{\theta}(x_i), y_i) = -\frac{1}{m} \left[\sum_{i=1}^m (y_i \log h_{\theta}(x_i) + (1 - y_i) \log(1 - h_{\theta}(x_i))) \right]$$

Figure 2: The Cost and J functions

- Figure out how to make the J-function minimal and find the regression parameter (θ)

The final step is that we, using gradient descent, solve for the minimum value of θ . The process of updating θ can then be summarized as follows.

$$\theta_j := \theta_j - \alpha \frac{1}{m} \sum_{i=1}^m (h_{\theta}(x_i) - y_i) x_i^j$$

Figure 3: The process function of updating θ .

2.7.2 Decision tree (DT)

The use of decision tree is usually based on the known probability of various scenarios, and the decision tree is formed to find the possibility that the expected net present value is greater than or equal to zero to evaluate the risk of the training project (Shen et al. 2007). Also, it judges the feasibility of the decision analysis method. Then we know that because this decision branch is drawn as a graph much like the trunk of a tree, we name it a decision tree.

Decision trees are a primary classification and regression method, and learning typically involves three steps: feature selection, decision tree generation, and decision tree pruning.

In machine learning, a decision tree is a predictive model that represents a mapping between object properties and object values. Classification tree (decision tree) is a very commonly used classification method. Similar to the dataset classification problem mentioned in this paper, the decision tree is a technique that is often used to analyze data and can also be used to make predictions. That is why we chose it for the training of the fraud detection system (Şahin and Duman 2011).

That is a simple decision tree classification model: the red boxes are features.

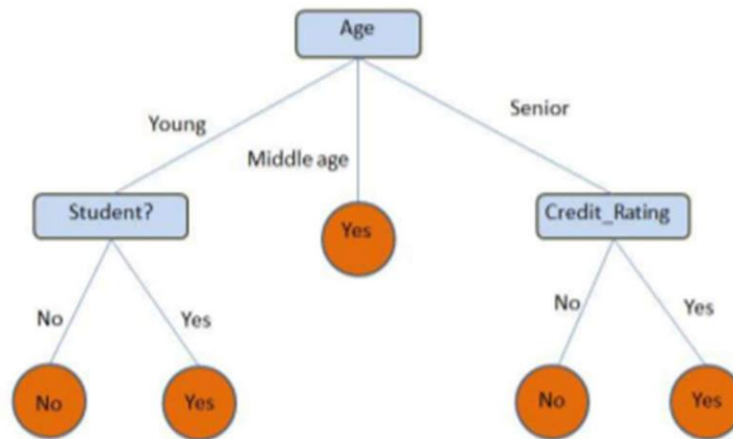


Figure 4: A simple decision tree

People may be wondering why we chose a decision tree. There are two universal reasons:

Decision trees usually mimic human horizontal thinking, so it is easy to understand the data we provide and make some excellent interpretations. Decision trees allow you to see the logic of how the data is interpreted, unlike SVM, NN. and other similar black-box algorithms where you do not see any internal information (Gaikwad et al. 2014). For example, as the figure above, we can see how logic makes decisions. Plain and simple.

Then, what is a decision tree now? A decision tree is kind like a tree in which each node represents an element (attribute), each link (branch) means a decision (rule), and each leaf represents a result (categorical or continuous value). The core of the entire decision tree is to create a tree-like this for the whole of the data. And the decision tree process individual results (or minimise errors in each leaf) on each plate.

2.7.3 k-nearest neighbour (KNN)

Initially proposed by Cover and Hart in 1968, Knn is a theoretically mature method that is one of the simplest of the data mining classification techniques. The term K nearest neighbours means K nearest neighbours which says that its closest K neighbouring values can represent each sample. The nearest neighbour algorithm is a method of classifying every record in a data set.

The implementation principle of KNN nearest neighbour classification algorithm is to determine the Category of unknown samples by taking all the examples of known types as a reference and at the

same time calculate the distance between the new models and all the available pieces, from which the nearest K has known examples are selected, according to the rule of majority-majority-voting, the unknown samples (Bahnsen et al. 2014) and the K nearest models belong to a category with more categories (Duman et al. 2013).

$$d((x_1, \dots, x_n), (y_1, \dots, y_n)) = \left(\sum_{i=1}^n |x_i - y_i|^p \right)^{1/p}$$

Figure 5: The formula for calculating the distance between two points.

The K value of the KNN algorithm in 'scikit-learn' is adjusted by the n neighbors parameter, and the default value is 5.

As shown in the figure below, *how do people determine which Category a green circle should belong to, whether it is a red triangle or a blue square?* If K=3, the green process will be judged to belong to the red triangle class because the proportion of red triangles is 2/3, and if K =5, the green circle will be considered to belong to the blue square class because the ratio of blue squares is 3/5 (Gaikwad et al. 2014).

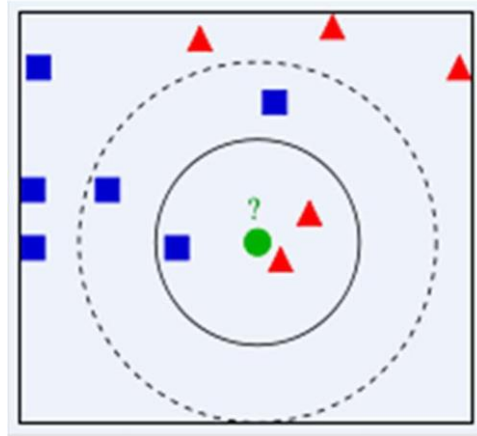


Figure 6: The k-nearest neighbor sample

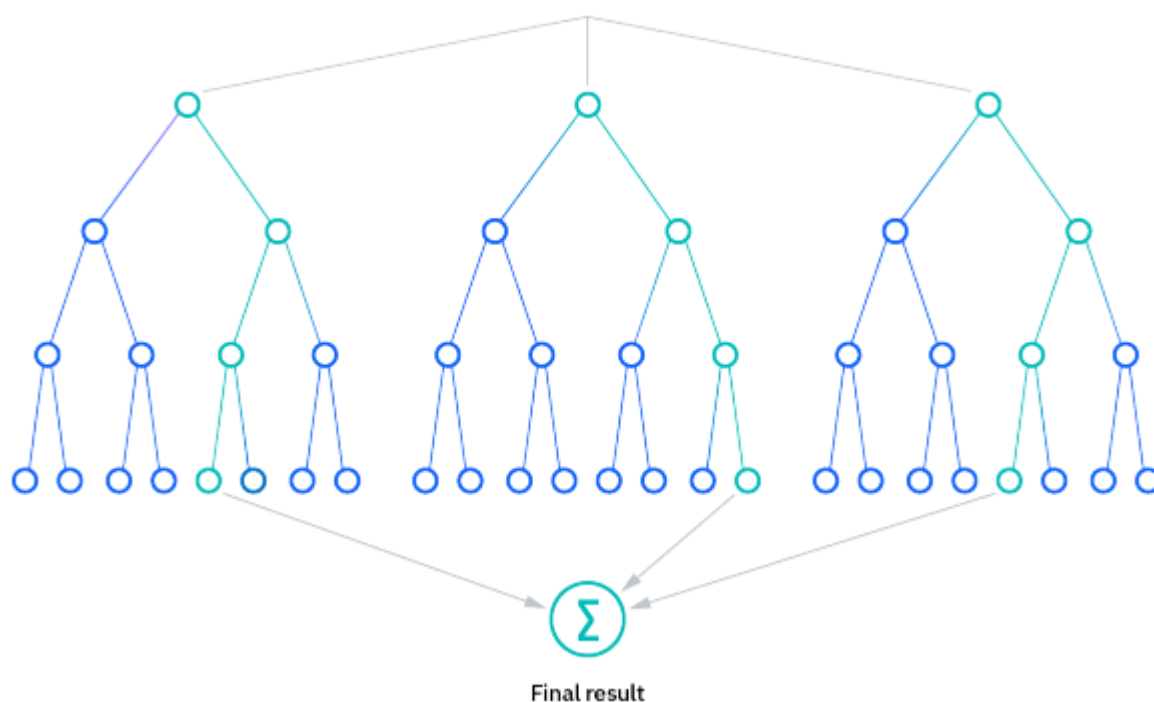
2.7.4 Random forest

The random forest algorithm is an extension of the bagging method as it utilizes both bagging and feature randomness to create an uncorrelated forest of decision trees. Feature randomness, also known as feature bagging or “[the random subspace method](#)”(link resides outside ibm.com), generates a random subset of features, which ensures low correlation among decision trees. This is a key difference between decision trees and random forests. While decision trees consider all the possible feature splits, random forests only select a subset of those features.

Random forest algorithms have three main hyperparameters, which need to be set before training. These include node size, the number of trees, and the number of features sampled. From there, the random forest classifier can be used to solve for regression or classification problems.

How it works

The random forest algorithm is made up of a collection of decision trees, and each tree in the ensemble is comprised of a data sample drawn from a training set with replacement, called the bootstrap sample. Of that training sample, one-third of it is set aside as test data, known as the out-of-bag (oob) sample, which we'll come back to later. Another instance of randomness is then injected through feature bagging, adding more diversity to the dataset and reducing the correlation among decision trees. Depending on the type of problem, the determination of the prediction will vary. For a regression task, the individual decision trees will be averaged, and for a classification task, a majority vote—i.e., the most frequent categorical variable—will yield the predicted class. Finally, the oob sample is then used for cross-validation, finalizing that prediction.



Key Benefits

- Reduced risk of overfitting: Decision trees run the risk of overfitting as they tend to tightly fit all the samples within training data. However, when there's a robust number of decision trees in a random forest, the classifier won't overfit the model since the averaging of uncorrelated trees lowers the overall variance and prediction error.
- Provides flexibility: Since random forest can handle both regression and classification tasks with a high degree of accuracy, it is a popular method among data scientists. Feature bagging also makes the random forest classifier an effective tool for estimating missing values as it maintains accuracy when a portion of the data is missing.
- Easy to determine feature importance: Random Forest makes it easy to evaluate variable importance, or contribution, to the model. There are a few ways to evaluate feature importance. Gini importance and mean decrease in impurity (MDI) are usually used to measure how much the model's accuracy decreases when a given variable is excluded. However, permutation importance, also known as mean decrease accuracy (MDA), is another important measure. MDA identifies the average decrease in accuracy by randomly permuting the feature values in oob samples.

2.7.5 Naïve bayes classifier

Naive Bayes is a simple technique for constructing classifiers: models that assign class labels to problem instances, represented as vectors of [feature](#) values, where the class labels are drawn from some finite set. There is not a single [algorithm](#) for training such classifiers, but a family of algorithms based on a common principle: all naive Bayes classifiers assume that the value of a particular feature is [independent](#) of the value of any other feature, given the class variable. For example, a fruit may be considered to be an apple if it is red, round, and about 10 cm in diameter. A naive Bayes classifier considers each of these features to contribute independently to the probability that this fruit is an apple, regardless of any possible [correlations](#) between the colour, roundness, and diameter features.

In many practical applications, parameter estimation for naive Bayes models uses the method of [maximum likelihood](#); in other words, one can work with the naive Bayes model without accepting [Bayesian probability](#) or using any Bayesian methods.

Despite their naive design and apparently oversimplified assumptions, naive Bayes classifiers have worked quite well in many complex real-world situations. In 2004, an analysis of the Bayesian classification problem showed that there are sound theoretical reasons for the apparently implausible [efficacy](#) of naive Bayes classifiers. Still, a comprehensive comparison with other classification algorithms in 2006 showed that Bayes classification is outperformed by other approaches, such as [boosted trees](#) or [random forests](#).

An advantage of naive Bayes is that it only requires a small amount of training data to estimate the parameters necessary for classification.

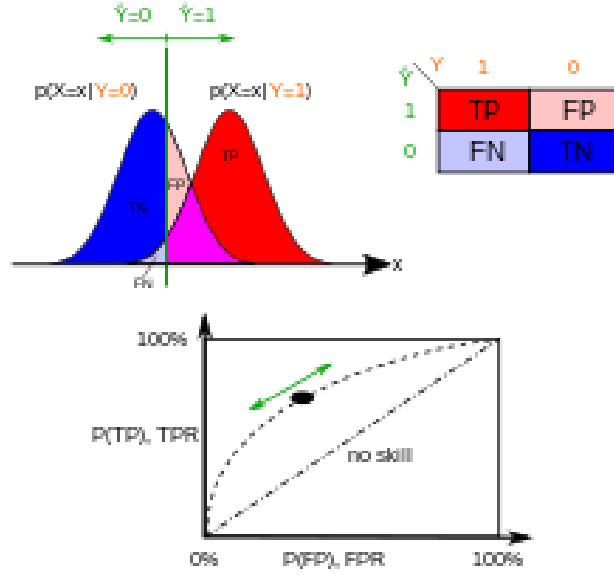


Figure 8 : Geometric Margin

2.8 Synthetic Minority Over-sampling Technique (SMOTE)

SMOTE is called Synthetic Minority Oversampling Technique which is an improvement of the random oversampling algorithm. The basic idea of the SMOTE algorithm is to analyse a small number of samples data and add new samples to the dataset based on the analysis of a small number of samples (Stolfo et al. 1997) s.

However, the class-imbalance problem that we need to solve next in this project refers to the uneven distribution of classes in the training set used in the training classifier (Pun 2011). For example, for a binary problem with 1000 training samples, ideally, the number of positive and negative models are similar; if there are 995 positive samples and only five negative samples, it means there is class-imbalance. There is also the case for the dataset in this project. We can see more details in section 3.3.

For now, there are three main approaches.

- Adjusting the value of θ

Adjust the value of θ according to the proportion of positive and negative samples in the training set. It is done based on the assumptions made about the training set, as described above. However,

whether this assumption holds in the given task is open to discussion.

- Over sampling

The classes with a small number of samples inside the training set (few types) are oversampled, and new models are synthesized to mitigate class imbalance.

- Under sampling

Under-sampling of classes with a large number of samples inside the training set (most categories), discarding some examples to mitigate class imbalance (Dal Pozzolo et al. 2015).

In this project, we use oversampling and under sampling to perform comparison operations. At the same time, we can also compare the results to analyse whether the two methods are more suitable for this project's dataset, and what are the advantages and disadvantages of each technique (Alghamdi et al. 2017).

The core idea of SMOTE (synthetic minority oversampling technique) in a nutshell is to interpolate between minority class samples to generate additional models. For example, for a minority sample x_i use the k-nearest neighbour method (k values need to be specified in advance) to find the k nearest minority samples to x_i (Sahin et al. 2013). The distance is defined as the Euclidean distance in the n-dimensional feature space between the models. One of the k nearest neighbours is then randomly selected to generate a new sample using the following formula (Han et al. 2005).

$$\mathbf{x}_{new} = \mathbf{x}_i + (\hat{\mathbf{x}}_i - \mathbf{x}_i) \times \delta$$

Where \hat{x} is the elected k-nearest neighbour point, and $\delta \in [0,1]$ is a random number. An example of a SMOTE-generated sample, using 3-nearest neighbours, is shown in the following figure which shows that the SMOTE-generated model generally lies on the line connected by x_i and \hat{x}_i .

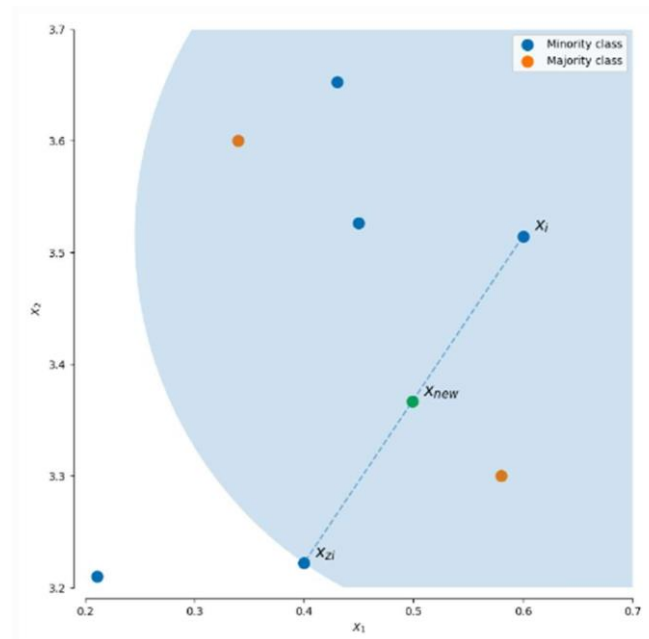


Figure 8: The example of SMOTE formation sample

2.9 Python libraries for data science and machine learning

This article uses the Python programming language for implementation. As you know the same, Python language, in addition to using basic pandas, NumPy and other open-source libraries, we project the most important, user data analysis and data mining the most essential one Python library. Then he is an efficient and straightforward open-source library. It is built on NumPy and other Python libraries on top (Pedregosa et al. 2011). And he contains classification, regression, clustering, dimensionality reduction, model selection and drinking pre-processing and other functions. Can save developers a lot of time and work.

In this project we used python libraries like NumPy, pandas , matplotlib , seaborn , and sklearn .

2.10 Related works

The study (Shen et al. 2007), Three alternative models based on logistic regression, neural networks and decision trees were developed. To evaluate these models, we used the 2005 transaction for training and the 2006 trade for validation and testing. According to the performance comparison of the predictive models, respectively, the neural network improved the accuracy by 5.88%, the logistic regression by 5.84%, and the decision tree by 3.89%.

In the study (Awoyemi et al. 2017), They used a dataset from the European trading market, containing 284807 trades. They used a hybrid technique of under-sampling drinking oversampling, implemented in Python, and used three classifiers for training. The accuracy of KNN and logistic regression was 97.69% and 54.86%. The results from his experimental study indicated that KNN performs better than all other linking techniques. It can provide us with a reference, the reason why the logistic regression is so low, and a way to adjust the KNN accuracy.

From the study (Sahin and Duman 2011), This study is in 2011 and is based on a comparison of ANN and logistic regression (LR) models. The study compares the performance of credit card fraud detection while comparing their performance on a test dataset. The results showed that the ANN model outperformed the LR model. However, as the number of models increased, an overfitting condition occurred. So, the LR model reduced the number of frauds detected in the test set despite the reduced accuracy of the test dataset. The authors argue that the number of fraudulent transactions captured is more important than the accuracy of the model because he concludes that a machine learning logistic regression model detecting a more significant number of frauds is a better choice.

3 Dataset pre-processing and features extraction

3.1 Dataset preparation and pre-processing

There is one dataset of credit card fraud detection from kaggle.com. The dataset contains data from European cardholders using credit cards for transactions in September 2013. Also, this dataset is complete and shows all transactions that took place over two days. This dataset has a memory size of 166Mb. It is a straightforward piece of data mining and self-classification done in the format of packet format.csv. We can put it directly on our local hard drive or a network drive and use it for direct access. There will be more details on the content of the relationship dataset in Section 3.2 (Dornadula and Geetha 2019).

3.2 General components of the credit card dataset

From the analysis of the dataset we obtained, the first thing we can do is open directly, and it was showing that in two days European cardholders made a total of 284,908 transactions via credit cards, of which only 492 were fraudulent. The dataset showed a highly unbalanced profile.

From Table 1, it contains only the numerical input variables because of the PCA transformation. Unfortunately, due to confidentiality issues, they are unable to provide the original characteristics of the data and additional background information. Features V1 V2...V7 are the principal components obtained by principal component analysis (PCA), the only features that are not transformed by PCA are "time" and "quantity". The element "Time" contains the number of seconds that elapsed between each transaction and the first transaction in the data set. Feature 'Amount' is the amount of the transaction, a part that can be used to rely on cost-aware learning. Feature 'Class' is the Category which takes the value 1 in case of fraud and 0 otherwise.

Figure 2.2: Dataset Summary

```
> summary(df)
      DH      DLT      RATIO      RR
Min.   : 0.005  Min.   : 0.000  Min.   : 0.0044  Min.   :0.0000
1st Qu.: 3.878  1st Qu.: 0.297  1st Qu.: 0.4757  1st Qu.:1.0000
Median : 9.968  Median : 0.999  Median : 0.9977  Median :1.0000
Mean   :26.629  Mean   : 5.037  Mean   : 1.8242  Mean   :0.8815
3rd Qu.:25.744  3rd Qu.: 3.356  3rd Qu.: 2.0964  3rd Qu.:1.0000
Max.   :10632.724  Max.   :11851.105  Max.   :267.8029  Max.   :1.0000

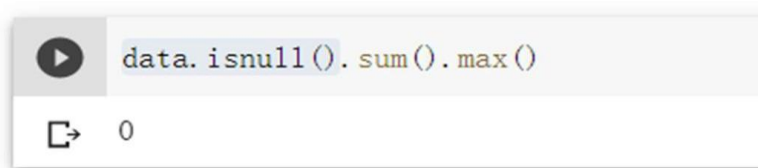
      CHIP      PIN      OO      Fraud
Min.   :0.0000  Min.   :0.0000  Min.   :0.0000  Min.   :0.0000
1st Qu.:0.0000  1st Qu.:0.0000  1st Qu.:0.0000  1st Qu.:0.0000
Median :0.0000  Median :0.0000  Median :1.0000  Median :0.0000
Mean   :0.3504  Mean   :0.1006  Mean   :0.6506  Mean   :0.0874
3rd Qu.:1.0000  3rd Qu.:0.0000  3rd Qu.:1.0000  3rd Qu.:0.0000
Max.   :1.0000  Max.   :1.0000  Max.   :1.0000  Max.   :1.0000
```

Table 1: Original dataset picture

No.	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

Table 2: Attributes of the European dataset

Then performed the data review process in the pre-processing data section of the data as the code is shown in Figure 9, `data.IsNull()` checks for missing values and the result is 0, so the data set is a good one that doesn't need to be processed for complementary values and can be used straight away.



```
data.isnull().sum().max()
```

```
0
```

Figure 9: check data empty result

3.3 Summary of Factors Influencing Choice of Project

In this section, we made a visual analysis of all the details of the sample, data review and data reduction. We found the following features and elaborated on whether it was decisive for the project.

- Detected samples without missing values:

The sample is complete, which is an advantage for the project; we can use it directly without manipulation.

- The sample class is too unbalanced:

The sample sampling class, the proportion of 0 and 1 is unbalanced, and we need to perform SMOTE sampling operation. Otherwise, it may affect the accuracy of our training model afterwards.

- The transaction time of the fraud class is irregular:

It is a distinct feature of the dataset, and although it does not have much impact on the sample training, we can have an intuitive comparison in the last result prediction.

- The fraud class is mostly small-value orders:

The fraud class typically has transaction orders that do not exceed \$1,000. Then this feature is also indicative. It is possible to compare the predictions of the final metadata results and predict whether the amount of transactions in the fraud class matches the characteristics of the original data.

- The amount and time distribution of fraudulent transactions are scattered:

The frequency of anomalies in fraudulent transactions is also relatively large, which we may use later in the comparison of results.

- PCA-processed features have a distribution and its similarity in the two categories:

The distribution of nine categories in both categories is so similar that we hope that a deletion process was performed—reduced confounding factors.

4 Implementation

4.1 Segmentation/reservation of the original training set

In this chapter, before we start training, we need to split and reserve the original dataset. The purpose of dynamite is that since we need to sample the data afterwards which will change the original dataset, we need to reserve a copy of the data first, and we are using the `train_test_split` method here. Because only in this way can we maintain the original imbalance of the test set which is essential because we will use it later in the validation of the prediction results. More details will be explained in Section 5.2 (Juszczak et al. 2008).

```
X_train, X_test, y_train, y_test = train_test_split(resampled_df[features],
resampled_df.fraud, test_size=0.2, random_state=42, stratify=resampled_df.fraud)
```

4.2 Sampling process

The target column `Class` presents a massive sample imbalance which can cause problems for model learning. In this project, we use the SMOTE (Synthetic Minority Oversampling Technique) to handle the sample imbalance.

4.2.1 Undersampling

We start with undersampling which is actually very simple, just randomly draw the same number of samples from a large number of pieces as a small number of samples. Here we generate a new dataset called `data_new`, and then we train the machine.

After using undersampling the proportion of standard and fraudulent transactions was 50 per cent and 50 per cent, and after sample reduction, the final sample size of trades was 984.

4.2.2 Oversampling

From the study which includes the principle of oversampling is increasing the number of positive samples, makes the number of positive and negative models. Then they were learning to process the data, constructing oversampled data (Barandela et al. 2004).

We use the SMOTE algorithm for up-sampling, and as an up-sampling technique, the SMO

algorithm does not simply copy the original small number of samples. But it can select an interval for each of its features that fluctuates by a small margin, performs new feature generation, and combines these features to generate new samples of that class. The models developed using this technique are much more common-sense.

SMOTE is already well encapsulated in the same imblearn package.

Since our original logistic regression is built based on the downsampled dataset, the selected parameters may not be suitable for the upsampled data. So, we need to find the optimal parameters for it again, here we use 'RandomizedSearchCV' for tuning, compared to GridSearchCV, RandomizedSearchCV will not try all the parameters so that it will consume less memory and time.

4.3 Individual classifiers evaluation

Before building the model, we need to slice and dice the dataset.

We used the **train_test_split** function to slice the dataset into x_train,x_test,y_train and y_test.

```
x_train, x_test, y_train, y_test = train_test_split(x_new, y_new, test_size=.2, random_state=42)
```

I prepared four pre-selected models, observed their predictive performance, and chose the best performing model.

The five models are LogisticRegression, KNN,

Classifier Name	Accuracy
Logistic Regression (LR)	94.0%
K-Nearest (KNN)	99.1%
Naïve bayes	85%
Decision Tree Classifier (DT)	99.0%
Random Forest	99.0%

Table 3: First training accuracy

4.4 Result and Conclusion

Dataset of the bank credit card is from kaggle.com. Also, we are pre-processing, and feature engineering scales and selects features, and uses the smote algorithm (undersampling and downsampling) to deal with the unbalance of the data set. Then we built an anti-fraud prediction model based on the five algorithms: **Logistic regression, KNN, Naïve bayes, Decisiontree(DT), Random Forest**. The model can predict whether a user has made fraudulent purchases.

Then we used a confusion matrix to compare the results of the two sampling methods. The best solution is logistic regression (undersampling) which is more in line with our expectations. It also achieves an accuracy of 97.00%. Then although credit card spoofing detection, most of the current research is still using decision tree and logistic regression test. But in this project, I think two points where we added Naïve bayes and universal algorithm Random Forest, to make training comparison together. I also believe meaningful results emerged. Random Forest did not perform poorly, and also, we dealt with the sample imbalance problem to get significant marks.

Finally, while KNN and Random Forest perform well, it is also possible to get a better notation if they are trained later on for integration. Secondly, the training of Decision Tree algorithms usually takes a long time, and if we are still increasing the amount of data, we may process the results differently.

5 Future work

This research project was more successful in completing the training of the credit card fraud-detection model, but there are many areas for improvement in future work.

- After completing the training of the optimal model, we can try to combine two or more classifiers with training and evaluating the detection performance. It can provide more possibilities.
- Use deep learning similar to neural networks. Deep learning is different from machine learning in that it is unsupervised learning. It uses unstructured or unlabeled data and does not require the developer to tell it what to look for in the data. It is then possible to train credit card fraud models in a simplified way (Rushin et al. 2017).
- Although we try to use catboost which is an excellent algorithm, due to the limit time, after adjusting the parameters, the performance can be more optimized.
- In the data source, as we are using someone else's original dataset possibly. At a later stage, if we then extract more data from the network. The amount of data is gradually increasing, which may be useful for training. The final predictive performance of the model is also improved. In other words, the detection accuracy is enhanced by a large data set.
- The classifier of machine learning is tested for different types of attacks. And analyse its performance under attack. And then use this. Make appropriate measures to improve its security.
- Using the existing mature and effective classification methods, we can enhance credit card detection—fraud detection performance. Then we use the current bank's credit card system to evaluate whether this model is accurate, as a way to test the real credit card fraud detection.

6 Conclusion

This research is all about studying credit card fraud-detection models based on different machine learning classification algorithms. The goal is to be in this training and testing. To find out the best way to process the dataset and the best machine learning classification algorithm for the dataset of this credit card transaction.

So, to achieve this, we chose five different classifiers, respectively. Between them, ten different combinations of algorithms and sampling methods were used to evaluate their predicted performance as a way to get better results for credit card fraud detection. Finally, we cross-validated the technique applied to all the individual classifiers to obtain more accurate results.

We also have some findings for this study:

- Using oversampling to deal with a too unbalanced credit card transaction dataset in the confusion matrix ended up with the same results as we expected.
- Logistic regression, as one of the simpler few algorithms, still has their advantages in targeting differential data processing, followed by the SVM algorithm. There is also the catboost algorithm which both perform well.
- We can compare to the previously mentioned literature for the model training and testing, this study obtains an optimal machine learning algorithm for credit card fraud detection - logistic regression (oversampling) - and also achieves high accuracy results.

7 Reflection on learning

This project used the training features of machine learning. It was also a validation of the technical and academic activities of the course I was studying in school - Machine Learning Applications which also started our learning life in Artificial Intelligence Training.

In the past year, we have carried out many machine learning projects such as text classification and natural language learning, all of which contributed to it. This time, however, credit card fraud detection was an area that I had not been exposed to before and provided me with the opportunity to use machine learning in working in this field.

The second insight was. This project was a real challenge. Because we would be using five different several-area learning classification algorithms to train the model, and by tuning and comparing the optimisation, we will build a more complex model to get a better result. It is also learning new techniques for cross-validation.

Third point. Because we are trying to choose a new fraud detection system, we broke through the traditional barriers, adding a new algorithm catboost which is not mentioned in previous research, from mostly processing of differentiated data. The catboost is a perfect algorithm. I think our attempt is meaningful, but also to provide a new direction for later research.

Reference

- Aihua, S. et al. 2007. Application of Classification Models on Credit Card Fraud Detection. IEEE.
- Al Daoud, E. J. I. J. o. C. and Engineering, I. 2019. Comparison between XGBoost, LightGBM and CatBoost Using a Home Credit Dataset. 13(1), pp. 6-10.
- Alghamdi, M. et al. 2017. Predicting diabetes mellitus using SMOTE and ensemble machine learning approach: The Henry Ford Exercise Testing (FIT) project. 12(7), p. e0179805.
- Awoyemi, J. O. et al. eds. 2017. *Credit card fraud detection using machine learning techniques: A comparative analysis. 2017 International Conference on Computing Networking and Informatics (ICCNi)*. IEEE.
- Bahnsen, A. C. et al. eds. 2014. *Improving credit card fraud detection with calibrated probabilities. Proceedings of the 2014 SIAM international conference on data mining*. SIAM.
- Barandela, R. et al. eds. 2004. *The imbalanced training sample problem: Under or over sampling? Joint IAPR international workshops on statistical techniques in pattern recognition (SPR) and structural and syntactic pattern recognition (SSPR)*. Springer.
- Bhatla, T. et al. 2003. Understanding credit card frauds. Cards Business Review# 2003–1.
- Bhattacharyya, S. et al. 2011. Data mining for credit card fraud: A comparative study. 50(3), pp. 602-613.
- Contributors, W. W. 2020. *Bibliographic details for "Credit card fraud"*. Available at: https://en.wikipedia.org/w/index.php?title=Credit_card_fraud&oldid=970300096 [Accessed: 10 September 2020].
- Dal Pozzolo, A. et al. eds. 2015. *Calibrating probability with undersampling for unbalanced classification. 2015 IEEE Symposium Series on Computational Intelligence*. IEEE.
- Dornadula, V. N. and Geetha, S. J. P. C. S. 2019. Credit Card Fraud Detection using Machine Learning Algorithms. 165, pp. 631-641.
- Dorogush, A. V. et al. 2018. CatBoost: gradient boosting with categorical features support.
- Duman, E. et al. eds. 2013. *A novel and successful credit card fraud detection system implemented in a turkish bank. 2013 IEEE 13th International Conference on Data Mining Workshops*. IEEE.
- Foulsham, M. 2019. Living with the new general data protection regulation (GDPR). *Financial Compliance*. Springer, pp. 113-136.
- FTC.gov, C. 2012. *Protecting Against Credit Card Fraud (2012)*. Available at: <https://www.consumer.ftc.gov/articles/0216-protecting-against-credit-card-fraud> [Accessed: 4 September 2020].

- Gaikwad, J. R. et al. 2014. Credit Card Fraud Detection using Decision Tree Induction Algorithm. 4(6),
- Han, H. et al. eds. 2005. *Borderline-SMOTE: a new over-sampling method in imbalanced data sets learning*. International conference on intelligent computing. Springer.
- Hancock, J. and Khoshgoftaar, T. M. 2020. CatBoost for Big Data: an Interdisciplinary Review.
- Juszcak, P. et al. 2008. Off-the-peg and bespoke classifiers for fraud detection. 52(9), pp. 4521-4532.
- Little, A. 2009. *BBC NEWS / UK / Overseas credit card scam exposed*. Available at: http://news.bbc.co.uk/2/hi/uk_news/7953401.stm [Accessed: 14 September 2020].
- Maes, S. et al. eds. 2002. *Credit card fraud detection using Bayesian and neural networks. Proceedings of the 1st international naisto congress on neuro fuzzy technologies*.
- McCurry, J. 2016. 100 thieves steal \$13m in three hours from cash machines across Japan.
- NFCC.gov. 2019. NFCC Consumer Tips For Preventing or Recovering From Identity Theft.
- Ng, A. Y. and Jordan, M. I. eds. 2002. *On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes*. Advances in neural information processing systems.
- Ogwueleka, F. N. J. J. o. E. S. and Technology. 2011. Data mining application in credit card fraud detection system. 6(3), pp. 311-322.
- Pedregosa, F. et al. 2011. Scikit-learn: Machine learning in Python. 12, pp. 2825-2830.
- Pun, J. K.-F. 2011. *Improving credit card fraud detection using a meta-learning strategy*.
- Rushin, G. et al. 2017. Horse race analysis in credit card fraud-deep learning, logistic regression, and Gradient Boosted Tree. IEEE.
- Sahin, Y. et al. 2013. A cost-sensitive decision tree approach for fraud detection. 40(15), pp. 5916-5923.
- Sahin, Y. and Duman, E. 2011. Detecting credit card fraud by ANN and logistic regression. IEEE.
- Seeja, K. and Zareapoor, M. J. T. S. W. J. 2014. FraudMiner: A novel credit card fraud detection model based on frequent itemset mining. 2014,
- Shen, A. et al. eds. 2007. *Application of classification models on credit card fraud detection. 2007 International conference on service systems and service management*. IEEE.
- Singh, G. et al. 2012. A machine learning approach for detection of fraud based on svm. 1(3), pp. 192-196.
- Stolfo, S. et al. eds. 1997. *Credit card fraud detection using meta-learning: Issues and initial results*.

AAAI-97 Workshop on Fraud Detection and Risk Management.

Tan, W. 2020. *Tjx Hacker Gets 20 Years in Prison / News / Communications of the ACM*. Available at: <https://cacm.acm.org/news/80681-tjx-hacker-gets-20-years-in-prison/fulltext> [Accessed: 14 September 2020].

Worobec, K. 2019. *Fraud the Facts 2019 / UK Finance*. Available at: <https://www.ukfinance.org.uk/policy-and-guidance/reports-publications/fraud-facts-2019> [Accessed: 14 September 2020].

Şahin, Y. G. and Duman, E. 2011. Detecting credit card fraud by decision trees and support vector machines.