Credit Card Fraud Detection With Machine Learning

Author: Songling Han
Supervisor: Dr Yuhua Li

A dissertation submitted in partial fulfilment of
the requirements for the degree of:
Master of Advance Computer Science

School of Computer Science and Informatics
Cardiff University
2020

Abstract

As the number of credit card transactions keep growing and represent 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 behaviour 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-neighbour(KNN), Support vector machine(SVM), Decision Tree(DT) and Catboost on credit card fraud data. The dataset of credit card transactions obtained from European cardholders containing 284,807 transactions. A mixture of under-sampling and oversampling techniques applied to the unbalanced data. The five strategies used to the raw and pre-processed data, respectively. This work 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 SVM, k nearest neighbour, logistic regression and Catboost 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 analysed 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

List of Figures

List of Tables

1. Introduction

1.1 Research motivation

Over the past year 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 and 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 want 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 used a lot. We think we have ambitions to complete the fraud detection project. Besides, we hope we achieve an opportunity to realise 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 to buy 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, unauthorised 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 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 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 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 focus 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 with 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

According to Seeja and Zareapoor, there are two main phases for credit card fraud detection.

- 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.

2.4 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 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 acne 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.

- Detecting the use of anonymous mailboxes and proxies

Many legitimate customers use free email addresses because they are convenient and economical.

Indeed, most fraudsters use free email addresses to remain anonymous. Detecting new domain registrations for email addresses is one of the most important ways to do a better job of fighting fraud(Bhatla et al. 2003). Secondly, anonymous proxy servers allow Internet users to hide their actual IP addresses. The primary purpose of using a proxy server is to remain anonymous or to avoid detection so people need to save the list of proxies as a web service to prevent credit card fraud.

- Using Neural Networks to Detect Credit Card Payment Fraud

Most of the existing techniques 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.

2.5 Credit 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 unauthorised third party pathways. The recent credit fraud, as well as becomes more challenging to identify. Because if there has anyone who knows them credit card number, as well as expiration date, he can make a transaction on the website without them 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.

2.6 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.

2.7 Fraud countermeasures

2.7.1 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)

2.7.2 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 using 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.

2.7.3 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).

2.8 Machine learning classifiers

In this project, we used a total of five classifications methods （**Logistic regression, KNN, Support vector machine (SVM), Decision tree(DT), Category&boosting(Cat boots)**. These classification algorithm methods are widely used for problems such as differential training dataset. Also, it 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 final results.

2.8.1 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 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

$$g(z) = \frac{1}{1+e^{-z}}$$

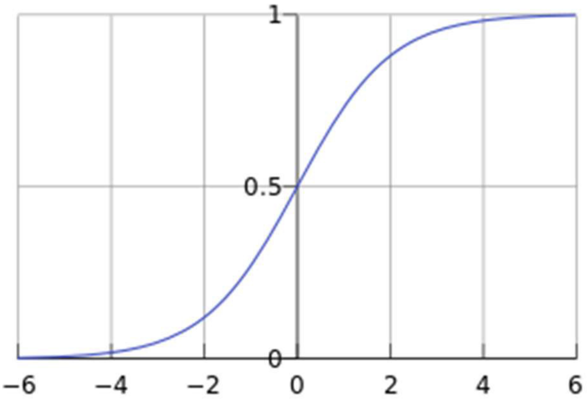


Figure 1: Logical function expressions

- 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).

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

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

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 summarised as follows

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

Figure 3: The process function of updating θ

2.8.2 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 analyse 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.

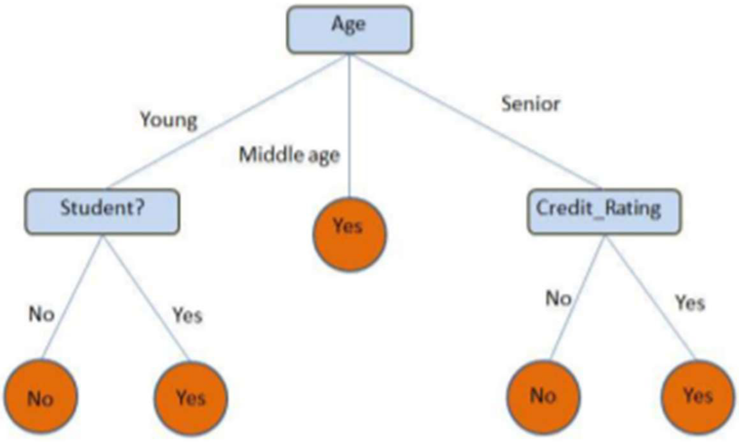


Figure 4: A simple decision tree

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 the logic makes decisions. Plain and simple.

Then, what is a decision tree now? A decision tree is kind like a tree 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.

2.8.3 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 mean 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).

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

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).

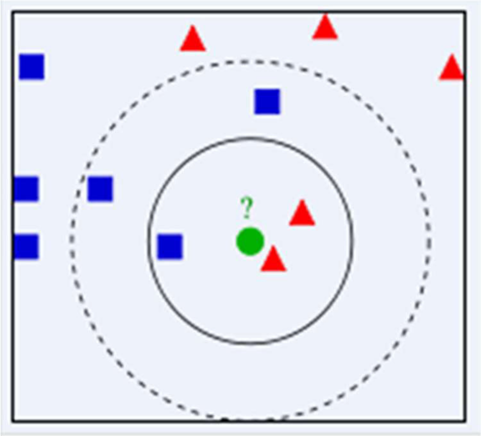


Figure 6: The k-nearest neighbour sample

2.8.4 Support vector machine(SVM)

Support Vector Machine (often abbreviated as SVM) is a supervised learning method, most widely used in statistical classification and regression analysis. It is also the focus of this project. Support vector machines belong to a family of generalised linear classifiers which are characterised by their ability to both minimise empirical errors and maximise geometric edge regions. Hence support vector machines are also known as maximum edge region classifiers.

The core principle of the support vector machine is: mapping the vectors into a higher dimensional space where a maximum spacing hyperplane is established. Two parallel hyperplanes are built on either side of the hyperplane that separates the data. Also, the separated hyperplanes maximise the distance between the two parallel hyperplanes(Singh et al. 2012). It is assumed that the greater the space or gap between the parallel hyperplanes, the smaller the total error of the classifier. In this project, SVM is the supervised learning algorithm used to solve the multi-class classification(Bhattacharyya et al. 2011).

Distance from example to the separator is $r = y\frac{\mathbf{w}^T\mathbf{x} + b}{\|\mathbf{w}\|}$

Examples closest to the hyperplane are support vectors. Margin ρ of the separator is the width of separation between support vectors of classes.

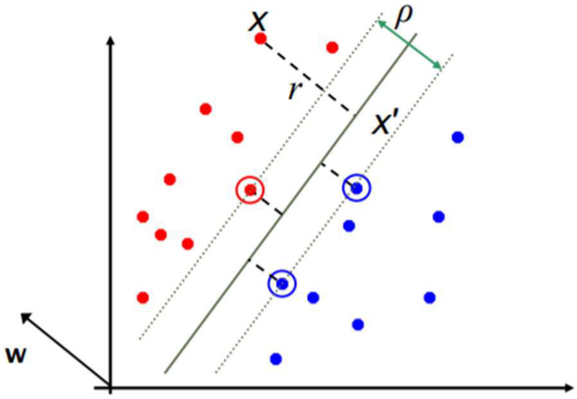


Figure 7: Geometric Margin

2.8.5 Category&boosting(Catboost)

CatBoost, a machine learning library open-sourced by Russian search giant Yandex in 2017, is Gradient Boosting + Categorical Features which is a machine learning framework that based on gradient boosting decision trees. The name CatBoost comes from two words "Category" and

"Boosting". As mentioned earlier, the library is a universal library of gradient boosting algorithms which contains many tree type algorithms(Hancock and Khoshgoftaar 2020). For example, it can handle a variety of category-type data well and is a library of gradient boosting algorithms that can handle category-type features well. That is the reason why we finally chose this algorithm. We wanted to compare the performance of the comprehensive library with the first four individual classification algorithms, including DT and catboost, also possesses some of these features(Dorogush et al. 2018).

Boost algorithm is another even beyond "lintbm" and "xgboost" by the author from a developer's point of view. The catboost has some of the following advantages:

- The catboost has a unique way of dealing with categorical features. First, it does some statistics on the categories and calculates the frequency of a type, such as the fraudulent transaction class in this question and then adds hyper-parameters to generate new numerical features(Al Daoud and Engineering 2019). Catboost is also robust: it reduces the need for tuning many hyperparameters and the chance of overfitting which makes the model more versatile and is the biggest motivation of the algorithm. catboost is also practical.

- The catboost is also a more practical method. It can handle both Category and numerical features and uses combined category features that can take advantage of the links between elements which significantly enriches the feature dimension.

- The base model of catboost uses symmetric trees, and the way to calculate the leaf-value is different from the traditional booster algorithm which calculates the average. However, catboost has been optimized to use other algorithms to prevent overfitting of the model. That is why the catboost algorithm can rival any advanced machine learning algorithm in terms of performance.

- Catboost is easy to use: catboost provides a Python interface for integration with scikit, as well as R and command-line interfaces which facilitate quick calls and reduce the number of calls. Also holds a custom loss function which also reflects his extensibility.

2.9 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 sample(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 xi use the k-nearest neighbour method (k values need to be specified in advance) to find the k nearest minority samples to xi(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 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 xi and x^i.

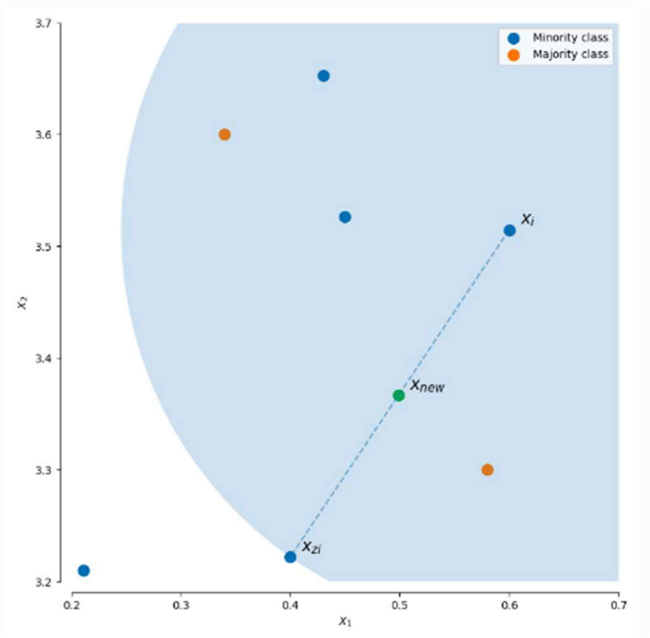


Figure 8: The example of SMOTE formation sample

2.10 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.

From this project, the library of catboost should be improved. It is an all-purpose algorithm library, and as we have mentioned before, to use this library, you need to download and install the catboost package first. This project is about the use of sklearn and catboost open source library.

2.11 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.

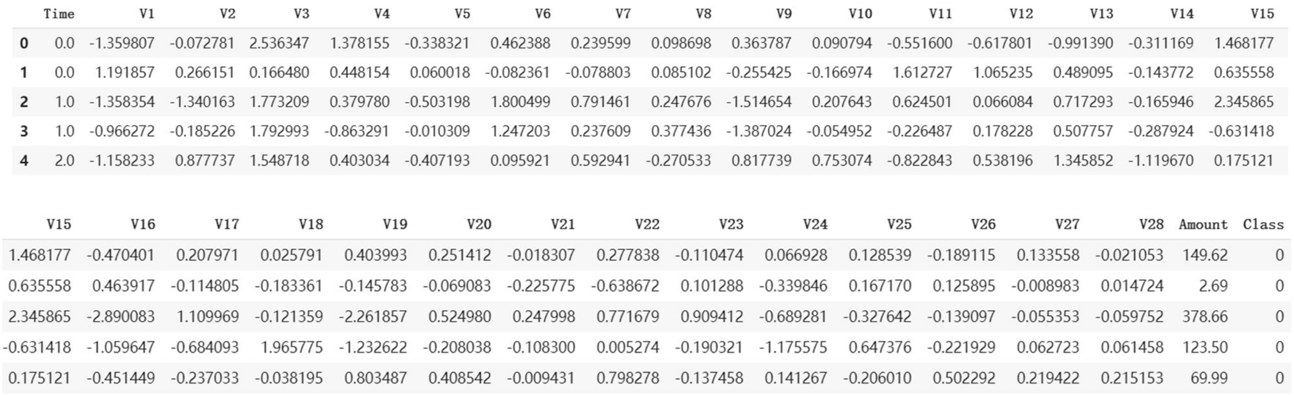
3 Dataset pre-processing and features extraction

3.1 Dataset preparation and pre-processing

There is one dataset of credit card fraud detection from the kaggle.com.The dataset that contains data from European cardholders using credit cards for transactions in September 2013. Also, this dataset is complete 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.cvs. 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...V28 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.



	Time	V1	V2	V3	V4	V5	V6	V7	V8	V9	V10	V11	V12	V13	V14	V15
0	0.0	-1.359807	-0.072781	2.536347	1.378155	-0.338321	0.462388	0.239599	0.098698	0.363787	0.090794	-0.551600	-0.617801	-0.991390	-0.311169	1.468177
1	0.0	1.191857	0.266151	0.166480	0.448154	0.060018	-0.082361	-0.078803	0.085102	-0.255425	-0.166974	1.612727	1.065235	0.489095	-0.143772	0.635558
2	1.0	-1.358354	-1.340163	1.773209	0.379780	-0.503198	1.800499	0.791461	0.247676	-1.514654	0.207643	0.624501	0.066084	0.717293	-0.165946	2.345865
3	1.0	-0.966272	-0.185226	1.792993	-0.863291	-0.010309	1.247203	0.237609	0.377436	-1.387024	-0.054952	-0.226487	0.178228	0.507757	-0.287924	-0.631418
4	2.0	-1.158233	0.877737	1.548718	0.403034	-0.407193	0.095921	0.592941	-0.270533	0.817739	0.753074	-0.822843	0.538196	1.345852	-1.119670	0.175121

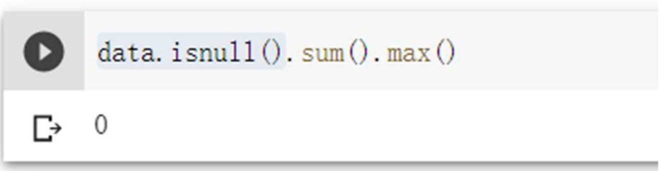
V15	V16	V17	V18	V19	V20	V21	V22	V23	V24	V25	V26	V27	V28	Amount	Class
1.468177	-0.470401	0.207971	0.025791	0.403993	0.251412	-0.018307	0.277838	-0.110474	0.066928	0.128539	-0.189115	0.133558	-0.021053	149.62	0
0.635558	0.463917	-0.114805	-0.183361	-0.145783	-0.069083	-0.225775	-0.638672	0.101288	-0.339846	0.167170	0.125895	-0.008983	0.014724	2.69	0
2.345865	-2.890083	1.109969	-0.121359	-2.261857	0.524980	0.247998	0.771679	0.909412	-0.689281	-0.327642	-0.139097	-0.055353	-0.059752	378.66	0
-0.631418	-1.059647	-0.684093	1.965775	-1.232622	-0.208038	-0.108300	0.005274	-0.190321	-1.175575	0.647376	-0.221929	0.062723	0.061458	123.50	0
0.175121	-0.451449	-0.237033	-0.038195	0.803487	0.408542	-0.009431	0.798278	-0.137458	0.141267	-0.206010	0.502292	0.219422	0.215153	69.99	0

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

By observing the statistical information of the data: it was found that the mean, maximum, minimum, median, etc. of Time and Amount are very different from V1-V28, and the mean values of V1-V28 and Class are concentrated around 0. The variance of the data is in the range of 0-1. It means that the information is evenly distributed, and the data of the amount dimension is very unevenly distributed, and the data of the Amount column is too big floating. At the same time, the scaling is different from

V1-V28, so in the process of machine learning, we need to ensure that the difference between the eigenvalues cannot be too large, so we need to carry out feature scaling standardisation on the amount. Also, we observed that the Time class which is a counting function, is not very useful for this research project, and that after all the Amounts are standardised, the delete operation can also be performed.

Then we counted the number of categories of regular and fraudulent consumption, as shown in the figure below. 0 indicates typical consumption, 1 tells fraudulent consumption, and the histogram shows that the amount of fraudulent data is minimal, while regular consumption is enormous. You can see that the distribution of positive and negative samples in the information is very unbalanced. It is important to note that if we build the model directly with this unbalanced data, the model will be inferior at predicting the small number of samples, so we will later balance the examples using the sampling Up/Throughout Technique (Han et al. 2005).

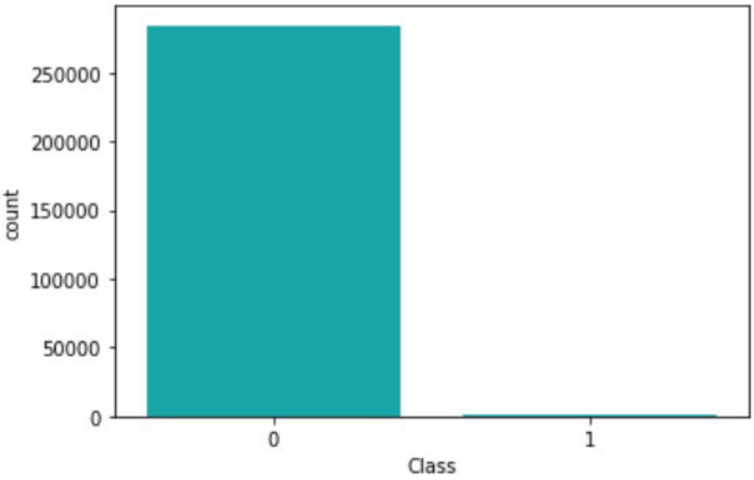


Figure 10: Number of categories of inspection samples

3.3 Feature engineering

In this chapter, we will complete the feature engineering by constructing a diagram to get a comprehensive view of the overall distribution of the data. Also, it might need to extract as many features as possible from the raw data for use by the algorithm and the model, and to integrate, select, and scale the elements for better performance. In this project, we hope to improve the accuracy and precision of the subsequent model training by conducting feature engineering on the dataset.

Figure 11 shows that a comparison of the time dimensions of the fraud and standard classes shows that the time distribution of regular transactions varies with some regularity. At the same time, there is no obvious time pattern for fraud transactions.

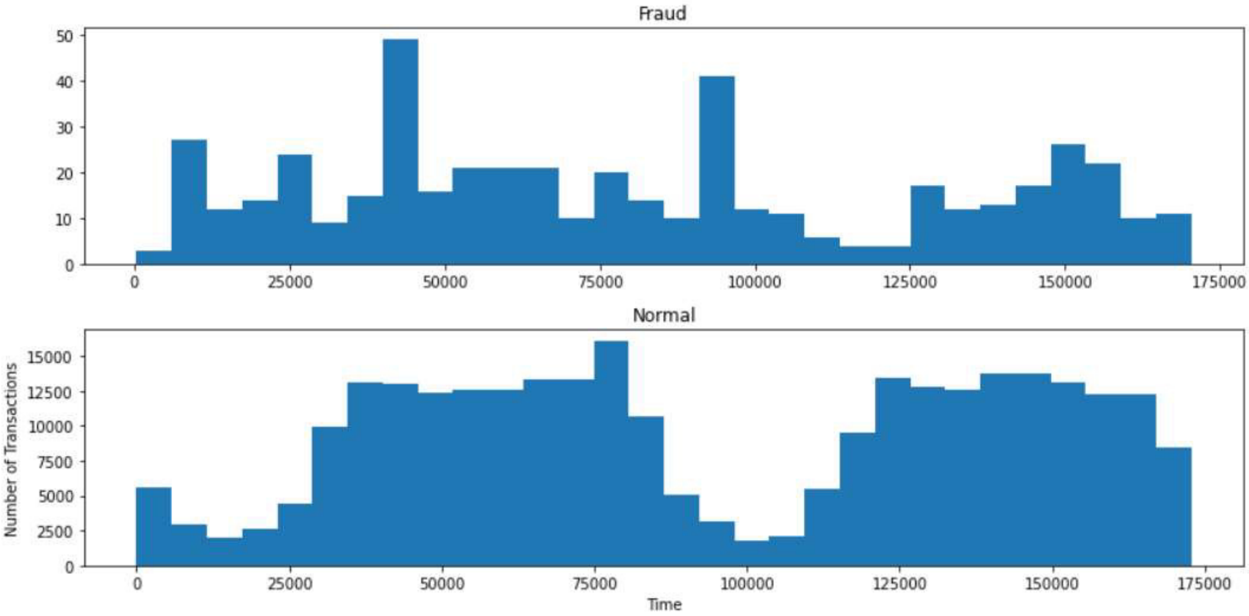


Figure 11: A comparison of the time dimensions of the fraud and regular classes

Then there is a comparison between order amounts, where both types of transactions exist in the form of long-tail distributions, but in terms of the number of fraudulent orders are mostly small orders, generally less than 1000, while the positive transactions range from 0-15000.

Then, Figure 12 shows that Plotting a scatter plot of time versus the amount. And it shows that regular transactions are evenly distributed across points in time, and outliers for transaction amounts are less frequent. In contrast, fraudulent transactions are scattered across time, and outliers occur more frequently.

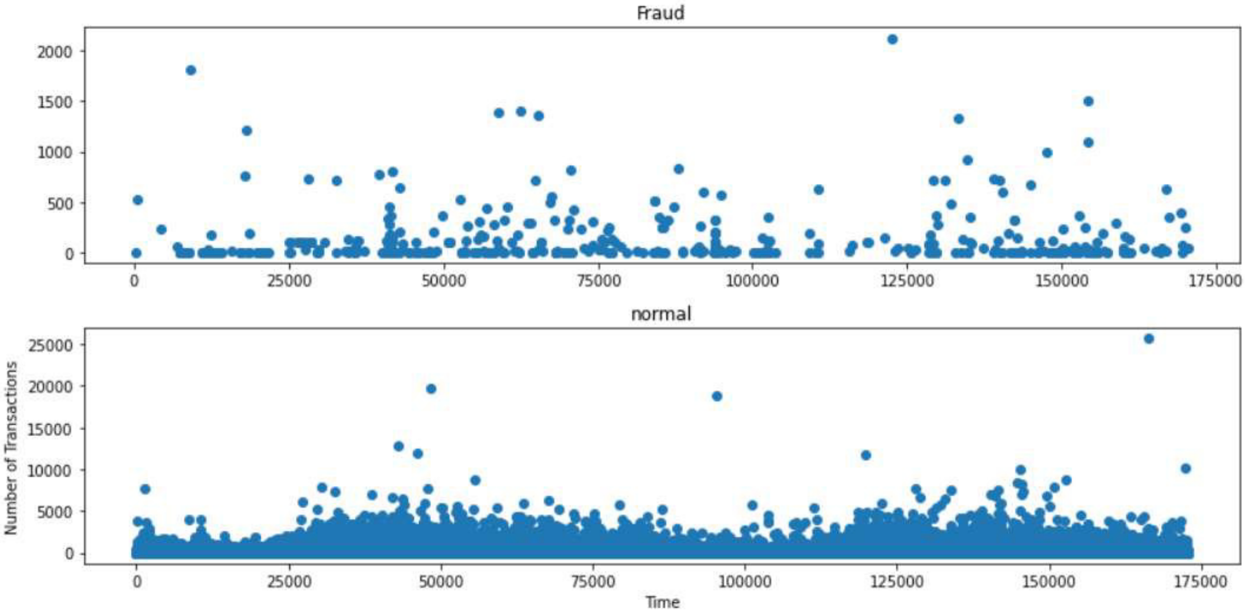
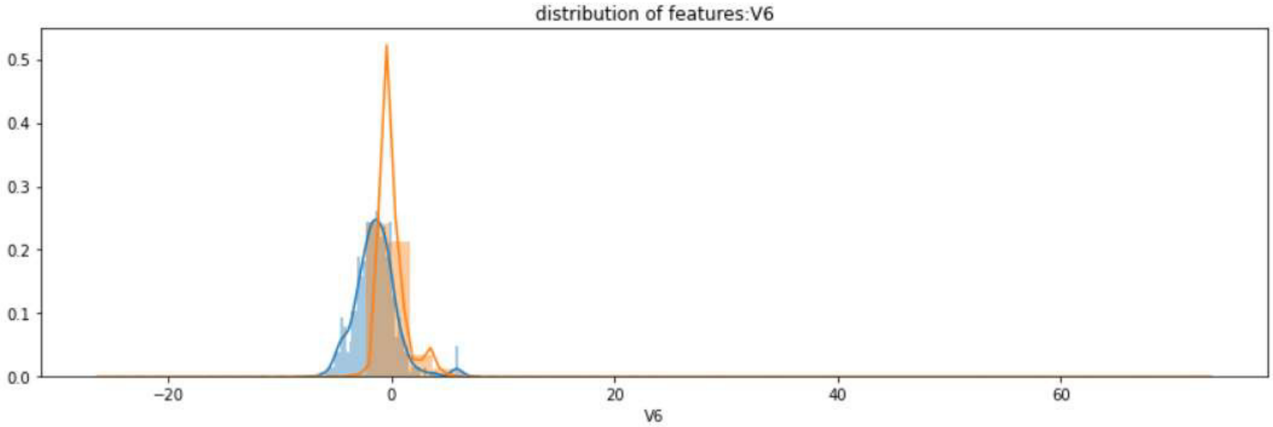


Figure 12: A scatter plot of time versus amount

Next, the distribution of each of the remaining PCA-processed features will be exported, and the distribution of each element within the standard and fraud classes will be observed.

We observed that the distribution of V6, V8, V13, V20, V22, V23, V24, V25, V26 is very similar in both categories, and the similar shape of the distribution means that the feature has little impact on the final prediction results, so it is deleted



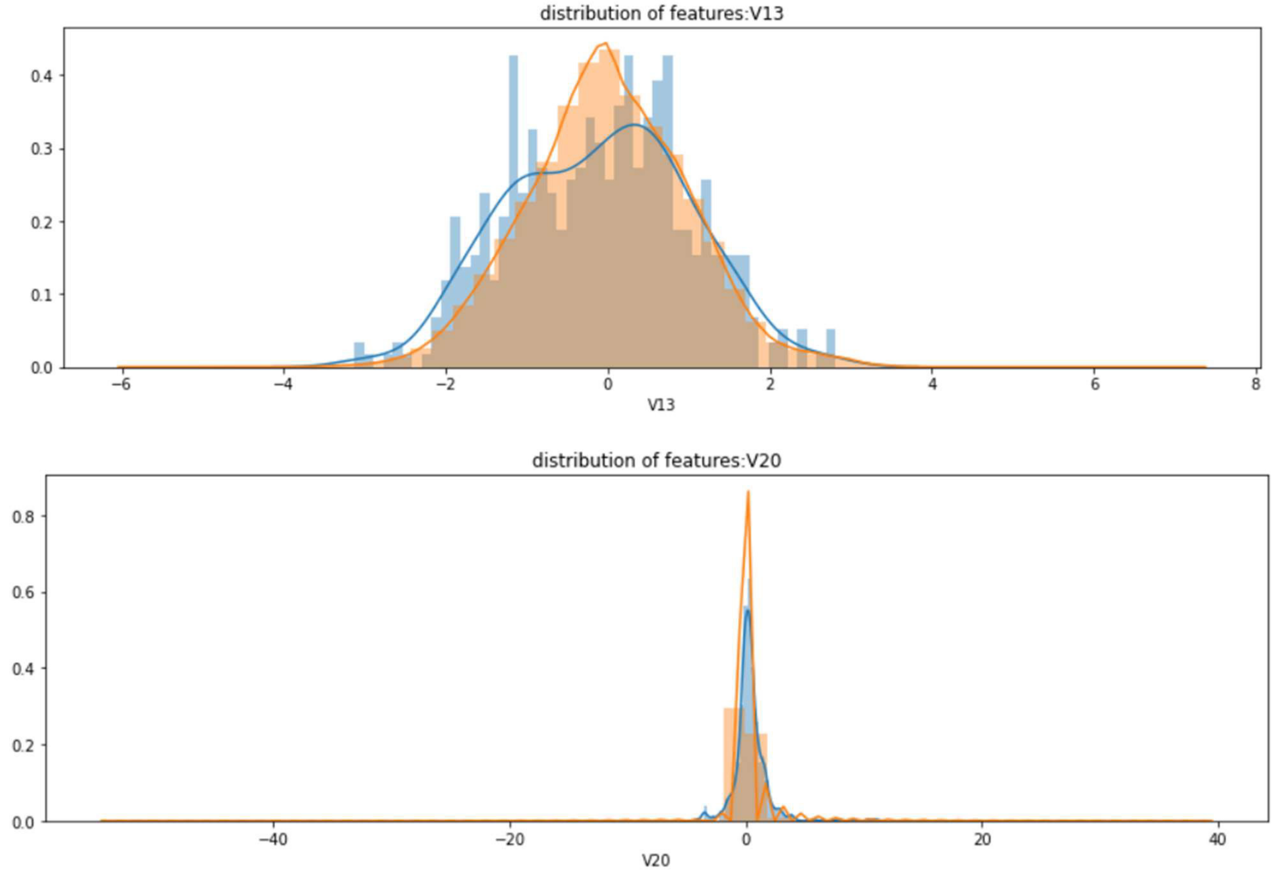
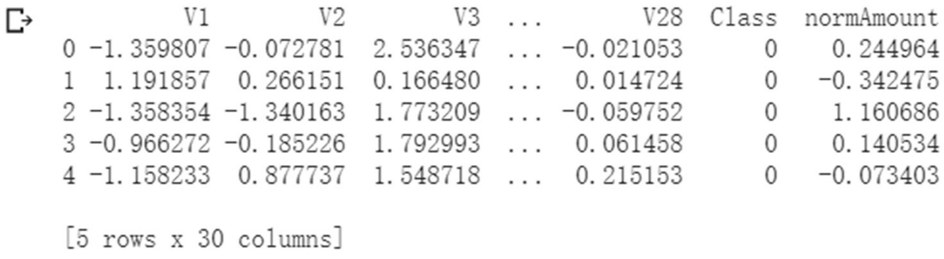


Figure 13: The part of the similar shape of the distribution

The last step, because we mentioned that the data in the Amount column floats too much, in the process of doing machine learning. We need to ensure that the eigenvalue difference cannot be too large, so we need to pre-process the amount and standardise the data. We normalised the two dimensions amount and hour by using the mean-standard deviation method.



```
         V1        V2        V3  ...       V28  Class  normAmount
0 -1.359807 -0.072781  2.536347  ... -0.021053      0    0.244964
1  1.191857  0.266151  0.166480  ...  0.014724      0   -0.342475
2 -1.358354 -1.340163  1.773209  ... -0.059752      0    1.160686
3 -0.966272 -0.185226  1.792993  ...  0.061458      0    0.140534
4 -1.158233  0.877737  1.548718  ...  0.215153      0   -0.073403

[5 rows x 30 columns]
```

Table 3: Data after standardisation

3.4 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.

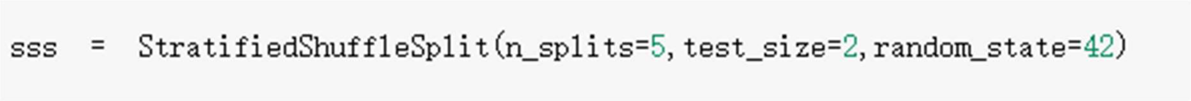
- PCA-processed features have a distribution and its similarity in the two categories:

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 StratifiedShuffleSplit method here. It is the combined form of StratifiedKFold and ShuffleSplit which ensures that each fold has the same proportion of samples for each Category while messing up the models randomly and dividing up the train/test pairs based on parameters. 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).



```
sss  =  StratifiedShuffleSplit(n_splits=5,test_size=2,random_state=42)
```

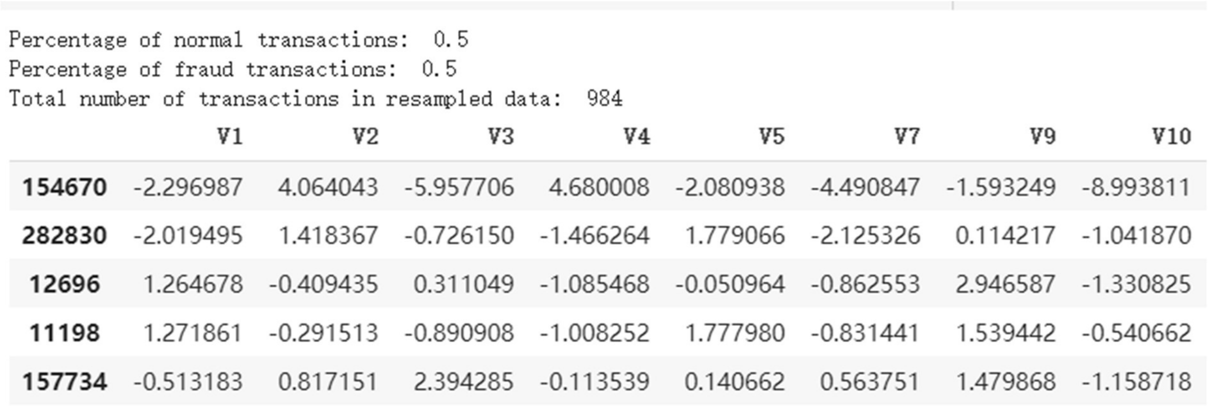
4.2 Sampling process

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



```
Percentage of normal transactions:  0.5
Percentage of fraud transactions:  0.5
Total number of transactions in resampled data:  984
```

	V1	V2	V3	V4	V5	V7	V9	V10
154670	-2.296987	4.064043	-5.957706	4.680008	-2.080938	-4.490847	-1.593249	-8.993811
282830	-2.019495	1.418367	-0.726150	-1.466264	1.779066	-2.125326	0.114217	-1.041870
12696	1.264678	-0.409435	0.311049	-1.085468	-0.050964	-0.862553	2.946587	-1.330825
11198	1.271861	-0.291513	-0.890908	-1.008252	1.777980	-0.831441	1.539442	-0.540662
157734	-0.513183	0.817151	2.394285	-0.113539	0.140662	0.563751	1.479868	-1.158718

Table 4: The dataset after undersampling

Let's use the sns.countplot function to see how the sample categories are classified after sampling.

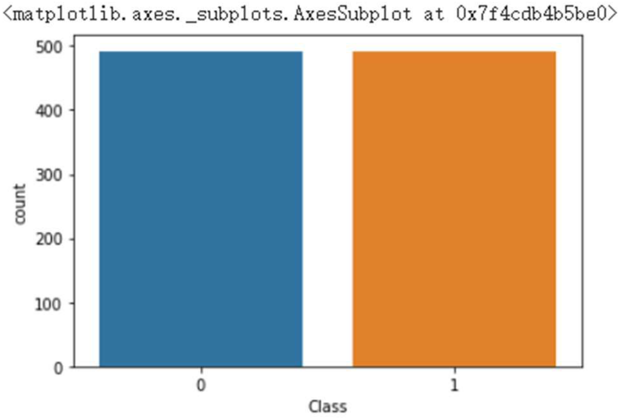


Figure 14: Sample classification

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).

After oversampling the data set, the number of samples of '1' is 227454. It means that the number of samples of '0' has also increased to 227454. 50% of each, for a total sample size of 454908. This method will also be used in Section 4.3.3, using the same training model for oversampled data.

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

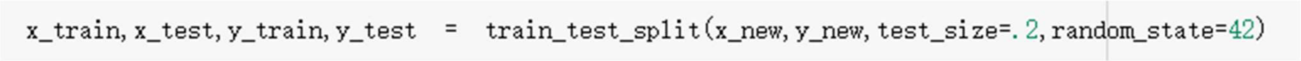
SMOTE is already well encapsulated in the same imblearn package.

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, RandonizedSearchCV 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_tain 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, Vector Machine (SVM) and Decision Tree, Catboost

Classifier Name	Accuracy
LogisticRegression(LR)	94.0%
K-Nearest(KNN)	93.0%
Support Vector Classifier(SVM)	93.0%
DesicionTreeClassifier(DT)	90.0%
CatBoostClassifier(CBT)	93.0%

Table 5: First training accuracy

We found that the logistic regression model and catboost work better, and it is a surprising bonus that the simple model of logistic regression is no worse than the complex model in terms of test performance.

The performance of the catboost model is as good as ever; after all, it is a combination of various optimised algorithms.

But here we're using models with default parameters, and the next thing we need to do is adjust the parameters for each model and then verify their accuracy.

And then we're building the model from this using the accuracy rate, not the recall rate.

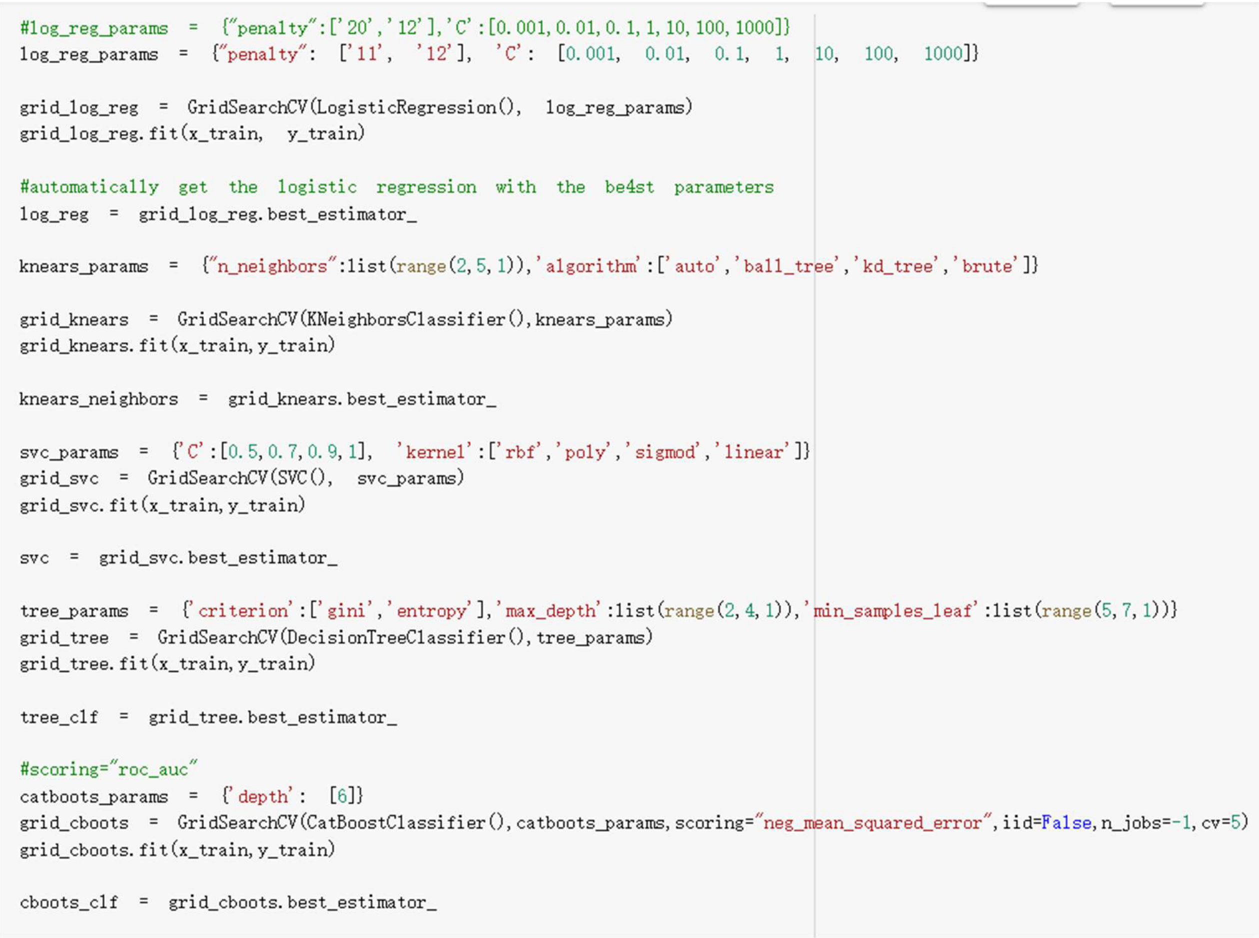
The reason for this is that data from post-sampling category balancing is now used, and accuracy has a better assessment and is more convincing

4.3.1 classifier evaluation with cross-validation

The shortcoming of our model training is that our model training and testing are conducted on the same data set which leads to overfitting of the model.

So we divide the sample. The cross-validation method partitions the data set. We let the model learn in the training set, tune the parameters in the validation set, and finally use the test set data to evaluate the performance of the model. Here we use cv cross-validation to partition the training set and the test set, and use grid search to select the optimal parameters.

The following figure shows the parameter setting code for the five models we are tuning.



```
#log_reg_params = {"penalty":['20','12'],'C':[0.001,0.01,0.1,1,10,100,1000]}
log_reg_params = {"penalty": ['11', '12'], 'C': [0.001, 0.01, 0.1, 1, 10, 100, 1000]}

grid_log_reg = GridSearchCV(LogisticRegression(), log_reg_params)
grid_log_reg.fit(x_train, y_train)

#automatically get the logistic regression with the be4st parameters
log_reg = grid_log_reg.best_estimator_

knears_params = {"n_neighbors":list(range(2,5,1)),'algorithm':['auto','ball_tree','kd_tree','brute']}

grid_knears = GridSearchCV(KNeighborsClassifier(),knears_params)
grid_knears.fit(x_train,y_train)

knears_neighbors = grid_knears.best_estimator_

svc_params = {'C':[0.5,0.7,0.9,1], 'kernel':['rbf','poly','sigmod','linear']}
grid_svc = GridSearchCV(SVC(), svc_params)
grid_svc.fit(x_train,y_train)

svc = grid_svc.best_estimator_

tree_params = {'criterion':['gini','entropy'],'max_depth':list(range(2,4,1)),'min_samples_leaf':list(range(5,7,1))}
grid_tree = GridSearchCV(DecisionTreeClassifier(),tree_params)
grid_tree.fit(x_train,y_train)

tree_clf = grid_tree.best_estimator_

#scoring="roc_auc"
catboots_params = {'depth': [6]}
grid_cboots = GridSearchCV(CatBoostClassifier(),catboots_params,scoring="neg_mean_squared_error",iid=False,n_jobs=-1,cv=5)
grid_cboots.fit(x_train,y_train)

cboots_clf = grid_cboots.best_estimator_
```

Figure 15: Parameter setting code for each model

We use gridsearch to construct a candidate set of parameters. Then gridsearch will exhaust various combinations of parameters to find the best location of settings according to the scoring mechanism of the set evaluation.

In gridsearch, we adjust two parameters, C and kernal, where 'C' is the penalty parameter C. If the default value is 1.0, the higher C is equal to the penalty relaxation variable. We hope that the relaxation variable is close to zero, i.e., the penalty for misclassification increases. It tends to be the case that the training set is fully split into pairs which is very accurate when testing the training set but has weak generalisation ability.

The kernel arguments represent the form of the kernel function which is 'rbf' by default, but can also be 'linear', 'poly', or 'kernel'. , 'rbf', 'sigmoid', 'precomputed', conducted experiments with 5-fold cv, and model accuracy was assessed using f1 -We set the range for C to be [0.01, 0.1, 1, 10, 100], and the range for kernal to be ['linear', 'poly', 'linear', and 'poly']. The best we can get is 'kernel' = 'linear', 'C' = 0.01, and the accuracy of the parametric model in the case of 'kernel' = 'linear', 'rbf', 'sigmoid'].

Classifier Name	Accuracy after cross-validation	Accuracy before cross-validation	Change
LogisticRegression(LR)	94.78%	94.0%	0.78%
K-Nearest(KNN)	93.52%	93.0%	0.52%
Support Vector Classifier(SVM)	93.14%	93.0%	0.14%
DesicionTreeClassifier(DT)	92.25%	90.0%	2.25%
CatBoostClassifier(CBT)	93.39%	93.0%	0.39%

Table 6: Cross-validation of results after parameterization

We can see that the accuracy of each model is improved to a certain extent after the parameter adjustment. And the results show that logistic regression is still the most suitable model for this task, followed by a vector machine, KNN and catboost, and the worst is the decision tree.

The accuracy of the SVM classifier is 94.78%. The default parameter SVM classifier in the same test set on the accuracy is 94.0%, the confusion matrix, as shown in Table 6 after the tuning model accuracy improved 0.78%.

Move to the next, The k-nearest(KNN) accuracy after cross-validation is 93.52%. Also, the accuracy before that is 93.0% which improved 0.52% in after parameter setting.

The accuracy rate of SVM is 93.14% which is not much higher than that before parameters were not adjusted, only 0.14% higher than that before 93.0%. Then the accuracy rate was the second to last. The decision tree was the worst in the detection performance before. But after the adjustment, it was much improved, from 90% to 92.25%. Unfortunately, its accuracy was still at the bottom. Finally, our CatBoost algorithm didn't get much of a boost, either, at 0.39%.From 93.0% to 93.39%. But it is still the third most accurate.

However, the accuracy difference between logistic regression and vector machine is not massive; we can observe the degree of fit of each model according to their learning curve, to choose the best model. According to the research, we can see that the test set accuracy of logistic regression and the training set accuracy are always close to each other which means that the models are not falling into overfitting or underfitting. In contrast, the training set accuracy of the vector machine is higher, with some slight overfitting, to this point, we choose logistic regression as our prediction model for this task.

4.3.2 Model predicts real data

Next, we need to build a good downsampling model to predict our real data which will use the normal_data we reserved at the beginning and the StratifiedShuffleSplit with the parameters we set. An unbalanced test set is the only way the predictions will make sense.

There are also several new methods we'll use in 'imblearn', including NearMiss which is a wrapped downsampling method in 'imblearn', make_pipeline which is similar to the pipeline mechanism in sklearn, but this is set up separately for sampling, and the model which selects the optimal model that we start tuning the parameters, and then train it

We then print the individual scores of the post-prediction model, and the graphs show that the recall is good. Still, the accuracy is low which means that we trained the model to make the wrong positive sample operation to filter out all the negative samples as much as possible. It is the result we got using the oversampled data. Next, we will use the over-sampled data for smooth.

Type(undersampling)	Score
accuracy	0.9773398582369592
recall	0.8769133570886076
precision	0.0584832501705173
f1	0.12836630579923764
roc_auc	0.9367326523286519

Table 7: Indicators for classifiers (undersampling)

4.3.3 Test set/confusion matrix

We next have an oversampling method, the rest is the same as just now, the same process for training, and we end up with a logistic regression model training oversampling.

Type(oversampling)	Score
accuracy	0.9773398582369592
recall	0.8759493670886076
precision	0.0684832501705173
f1	0.11836630579923764
roc_auc	0.9267326523286519

Table 8: Indicators for classifiers (oversampling)

As you can see from the various metrics of the classifiers in Figure 16, the accuracy of the oversampling method has improved significantly compared to the previous undersampling way. Still, at the same time, there is a slight decrease in the corresponding recall rate. But even so, we still need to make a final prediction on the test set and then draw a confusion matrix of the two for comparison.

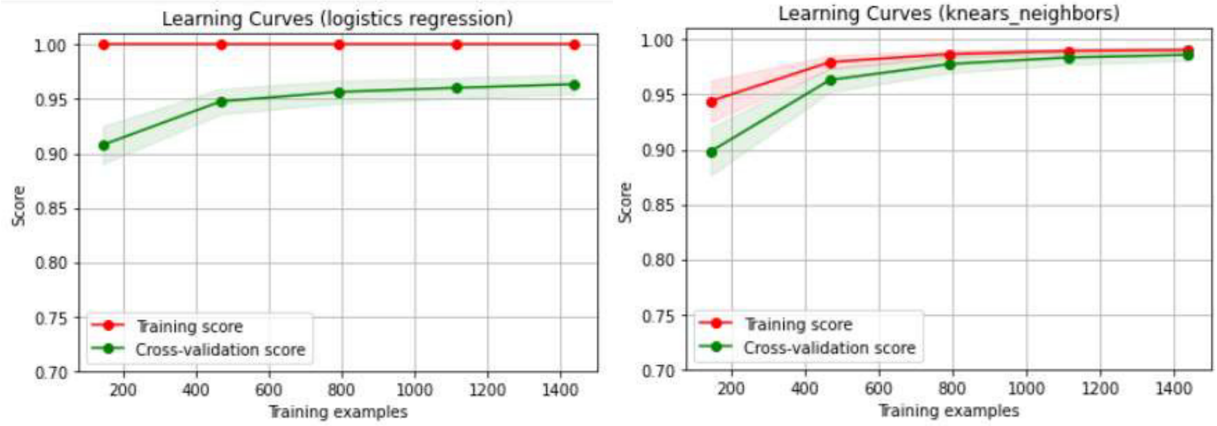


Figure 16: The learning curve of the models

Next, we performed the same operation as above using the same over-sampling method, and we came up with the results. We plotted the confusion matrix to compare the differences between the two sampling methods.

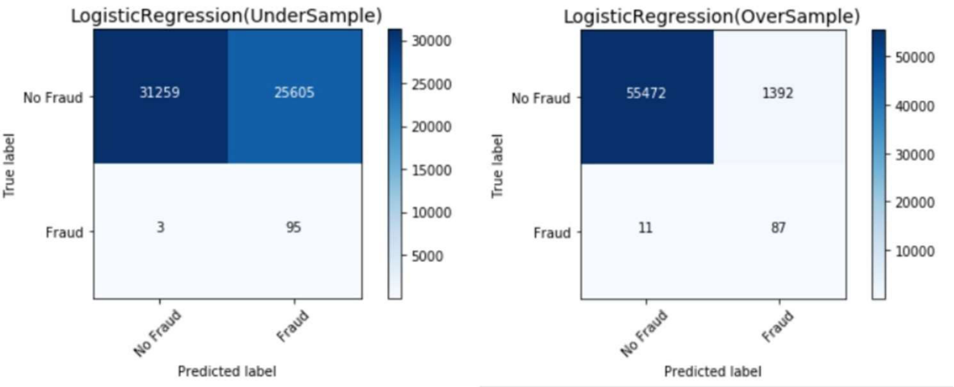


Figure 17: Logistic Regression with oversample & undersample

As we see in the figure above, we have performed up-sampled and undersampled confusion matrix calculations using our logistic regression model.

The first is the confusion matrix plot for under-sampling. As you can see, our logistic regression model is reasonably accurate in identifying fraudulent samples. Still, nearly half of the standard transactions

are predicted to be fraudulent which makes the classification of standard samples irregular. While such a model is good enough to predict the fraudulent sample we need, whether it is commercially viable from the standpoint of our project is open to question.
Since more than half of the ordinary transactions identified as fraudulent lead to failed transactions, this loss is significant, suggesting that we may need a better model.

The up-sampling confusion matrix was then plotted, and the up-sampling predictions were similar to what we expected, even though there were a few unidentified frauds. Still, the number of false positives for regular trades was much lower which is probably more in line with our expectations.

Conclusion, this anti-fraud training project, we used 'sklearn', 'imblearn', 'Keras' and other scientific computing packages. And it applied techniques such as cross-validation, grid/random search, up/down Sampling, learning curve, confusion matrix, etc., and built logistic regression, neural network, decision tree, KNN, vector machine Catboost and other learning models.

Finally, the down-sampled logistic regression model was chosen as the final model which has a recall rate of 87% and an accuracy of 97%.

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 build an anti-fraud prediction model based on the five algorithms: **Logistic regression, KNN, Support vector machine (SVM), Decision tree(DT), Category&boosting(Catboost)**. 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 SVM and universal algorithm catboost, to make training comparison together. I also believe meaningful results emerged. catboost did not perform poorly, and also we dealt with the sample imbalance problem to get significant marks.

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

5 Comparative analysis

In the present study, though, we conducted a comparative analysis. However, it is only limited to the study of these classifier algorithms and a single data set. Although the feature selection in credit card detection is similar, the collection method and method selection are different, so different research subjects will have very different results. For example, according to the study(Sahin and Duman 2011), KNN is the most accurate classifier algorithm, but logistic regression is the worst in comparison. It may have something to do with the process and purpose of the experiment.

5.1 Comparative analysis without cross-validation

We compared the performance results of this experiment with those of previous studies. The aim is to look for products on credit card fraud, although different techniques are used. But our goals are the same. While improving the accuracy of detection, many factors need to be considered such as data set size, using a classifier and final evaluation method. Are essential factors in determining accuracy.

Author	Classifier	Sample Size	Accuracy
Sahin and Duman 2011	Ann and Logistics Regression	Training:2723 Testing: 1168	94.51%
Songling Han 2020	Logistic regression, KNN, Support vector machine (SVM), Decision tree(DT), Category boosting(Catboost)	Training: 688 Testing: 295	94.0%
Awoyemi, J. O 2017	Naïve Bayes, k-nearest neighbour(KNN) and Logistic Regression(LG)	Training: 159,238 Testing: 68,236	97.37%

Shen 2007	decision tree, neural networks and logistic regression	Training: 159,238 Testing: 68,236	95.84%

Table 9: Comparison of the result of different studies without cross-validation

Table 9 is a comparison of the parameters of two different studies; We were able to see that the highest accuracy was 97.37%, and also using the highest number of samples. This research chooses the Naïve Bayes, k-nearest neighbour(KNN) and Logistic Regression(LG) classifier(Awoyemi, J. O 2017). The next largest number of datasets, again using a neural network(Shen 2007), also used the original maximum number of datasets, with an accuracy of 95.84%, meaning that the high number of samples, KNN and Bayesian algorithms were optimal before the tuning cross-validation was performed.

5.2 Comparative analysis with cross-validation

When we have used cross-validation and some other manipulations in different studies, it means that our data set and accuracy will change in some way.

Author	Classifier	Sample Size	Accuracy
Sahin and Duman 2011	Ann and Logistics Regression	Training: 6130 Testing: 2626	94.69%
Songling Han 2020	Logistic regression, KNN, Support vector machine (SVM), Decision tree(DT), category boosting(Catboost)	Training: 159,238 Testing: 68,236	94.79%
Awoyemi, J. O 2017	Naïve Bayes, k-nearest neighbour(KNN) and Logistic Regression(LG)	Training: 159,238 Testing: 68,236	97.69%
Shen 2007	decision tree, neural networks and logistic regression	Training: 159,238 Testing: 68,236	95.88%

Table 10: Comparison of the result of different studies with cross-validation

Although cross-validation is not the most significant factor in determining classifier technology; however, as can be seen from the table, it still has an impact on the different classifiers. We see that. The highest accuracy rate is 97.69% achieved by Awoyemi, J. O 2017.

Then the other studies have increased their accuracy rate accordingly, from which it is also evident that there is another point where the number of datasets is also an essential factor in the accuracy rate. When the training dataset is increased in the first(Sahin and Duman 2011) and the second(Songling Han 2020) study, there is an increase in the accuracy rate (the most massive increase from 94.0% to

94.79%), and in the subsequent third(Awoyemi, J. O 2017) and fourth(Shen 2007) research, the accuracy rate has almost stabilized and remains around 97% and 96%.
It can be seen that the size of the training set has some influence on the performance of the classification system. With the increase in the training size, the classification performance gradually improved. However, classification performance changes little after a specific size and remains at a more stable value.

However, it should be reminded that in the actual experiment, when the training scale increases, although the classification performance is improved, the training time also will be doubled, the corresponding feature potential growth, the classification time will also increase. Therefore, in future experiments, classification performance and time requirements should be considered together.

6 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.

7 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.

8 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.

Juszczak, 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 naiso 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.