Machine Learning In Fraud Detection

Thesis submitted for CSP571 Course Project
by

Baoshu Feng
(A20343813)

July 2022

Abstract

With the popularity of the internet, online payments are now becoming more and more popular. People cannot live without online payments every day, but this has also led to online fraud. Many online frauds occur every day and this has led to many people losing a lot of money as a result. In this project we worked on solving this problem and we got the IEEE open source fraud dataset from kaggle. We used the most popular machine learning methods to build a fraud model that predicts whether a user will commit a fraud based on their transaction history. Here we used random forest, Deep Neural network, Lightgbm and achieved a maximum accuracy of 98.62% on the test set

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Background

Payment fraud[1] has a long history and is the most common form of online fraud in the United States and around the world. However, in recent times, digital fraud has increased significantly and many people are unable to distinguish well

The use of up-to-date systems with the ability to learn has become indispensable as companies strive to stay ahead of the game in the face of the relentless threat from criminals. This reflects how organised crime and state-sponsored fraudsters are stepping up their fraud efforts.

The most common approaches to combating online fraud include rules and predictive models that are no longer adapted to the sophistication of today's increasingly sophisticated online threats. The vast majority of emerging attacks in the digital fraud space rely on machine learning and other automated techniques to perpetrate fraud. Integrating AI-based platforms into high-stakes games to detect online fraud is a key part of AI, and today, it enables us to extend online fraud prevention[2]. Digital businesses with specific business models and their fraud analysts can derive results from fraud analysis based on surveillance and unattended machine learning to provide business models with the information they need to detect and stop threats at an early stage. The results of unsupervised and supervised machine learning are characterised by the detection of anomalies in emerging data, and integrating them into risk assessment is an important step towards artificial intelligence to detect online crime and enhance online prevention.

Chapter 2

Data Source

2.1 Data Description

We used a dataset from a Kaggle competition[3], run by the IEEE, which provides a large-scale feature-rich dataset for fraud Detection.For this competition, the organisers provided a very challenging and large-scale dataset on which we could benchmark. This data is derived from real transaction records from Vesta and contains many additional features that can be used to model well

Vesta, which provided the dataset for the competition, pioneered the process of securing e-commerce payment solutions. Founded in 1995, Vesta pioneered the process of cardless payments and since then Vesta has expanded and consolidated its data science and machine learning capabilities globally to secure its e-commerce payments

In the data provided by this competition, the data is divided into two files which are linked by the TransactionID. The task is to predict the probability of an online transaction being fraudulent

2.2 Data Visualisation

After obtaining the data set we perform a simple visual analysis of the data set, which allows us to have a better understanding of the data. As the Fig. 2.1 Shown, we visualise the labels of the data set,we can see that fraud accounts for a very small proportion of this dataset

Since there are many features in the dataset, here we classify them into categorical features and numerical features with respect to the type of values they take. We visualised the correlation between the categorical features and the numerical features separately.As the Fig. 2.2 and Fig. 2.4 shown,We can see that for numerical features the correlation is low, but for categorical features the correlation between many features is high, e.g. between C4 and C1 the correlation is 0.97

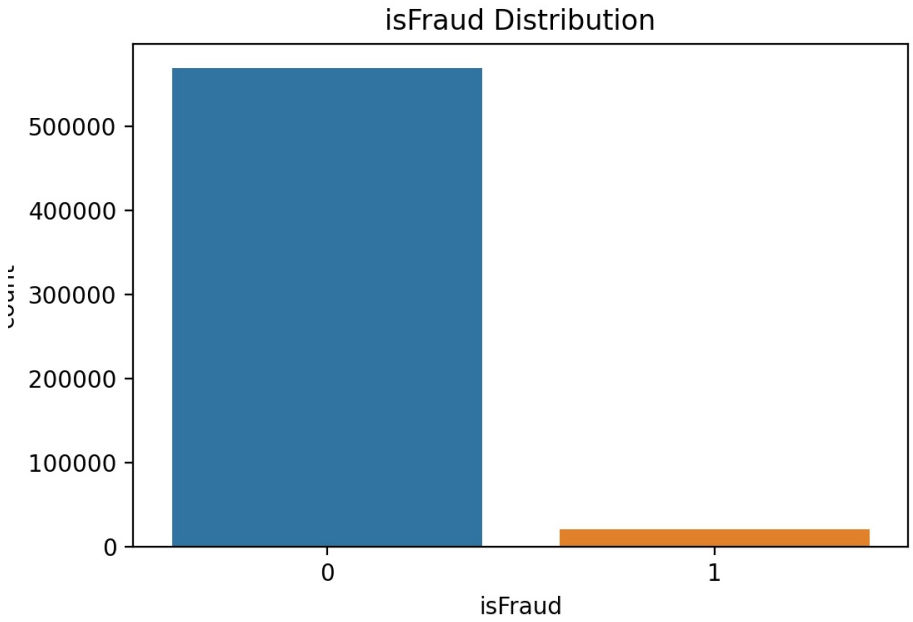

Figure 2.1: Label Distribution

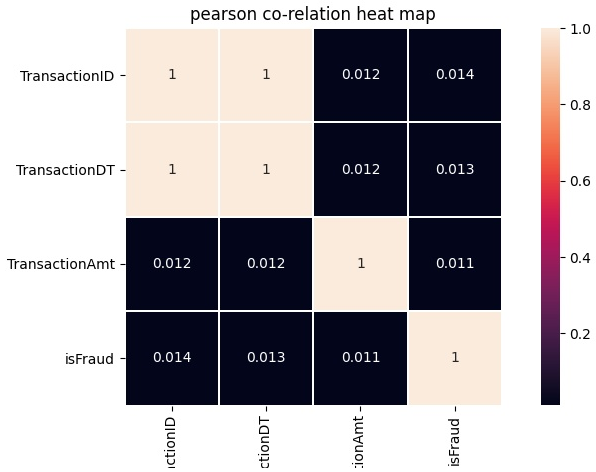

Figure 2.2: Category Feature Correlation

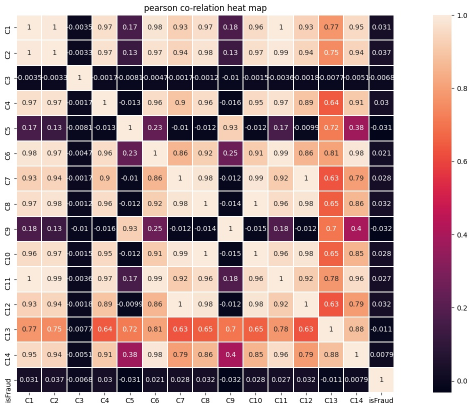

Figure 2.3: Numerical Feature Correlation

2.3 Exploratory Data Analysis

According to the various feature engineering solutions tried before, let's do a preliminary feature engineering first. Because it takes a lot of time to cross-validate each feature, all appropriate feature derivation and processing methods are carried out, and then some complex feature selection is done uniformly.

As the Fig. 2.4 shown, the overall deviation of the test set and training set for transactionAmt feature is not particularly large, except for some values. Combining the features with the label observation, from the perspective of the proportion, the transactionAmt feature, the proportion of high-value transactions of fraudsters is much higher than that of ordinary customers. Therefore, in the large transaction amount range on the right, the proportion of fraudulent users is generally relatively high.

On the whole, the feature distribution of productCD is relatively stable, and there is no obvious difference. From the perspective of positive and negative samples, the difference is relatively large, the W feature of fraudulent customers is relatively small, and the C feature accounts for a relatively large proportion. lightgbm can identify this difference and apply it to training. ProductCD has only 4 categories, so no additional processing is required.

Considering that there are too many V features, separate analysis takes a lot of time. It is not clear whether such features need to spend a lot of time to analyze. Therefore, as the Fig. 2.5 shown, we analyze and rank the importance of all features uniformly.

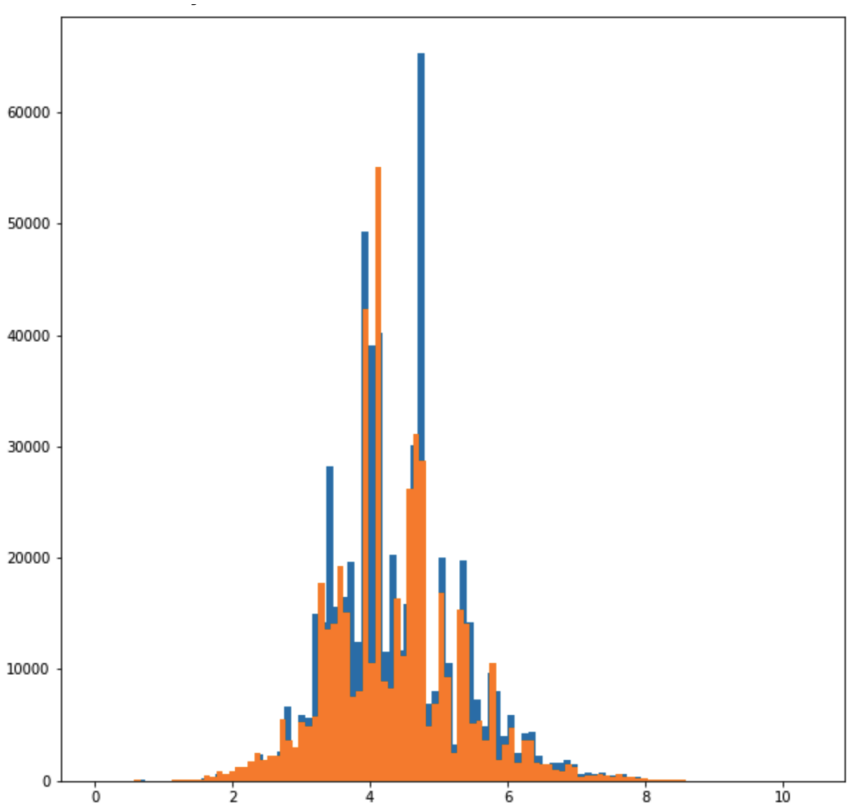

Figure 2.4: transactionAmt Feature

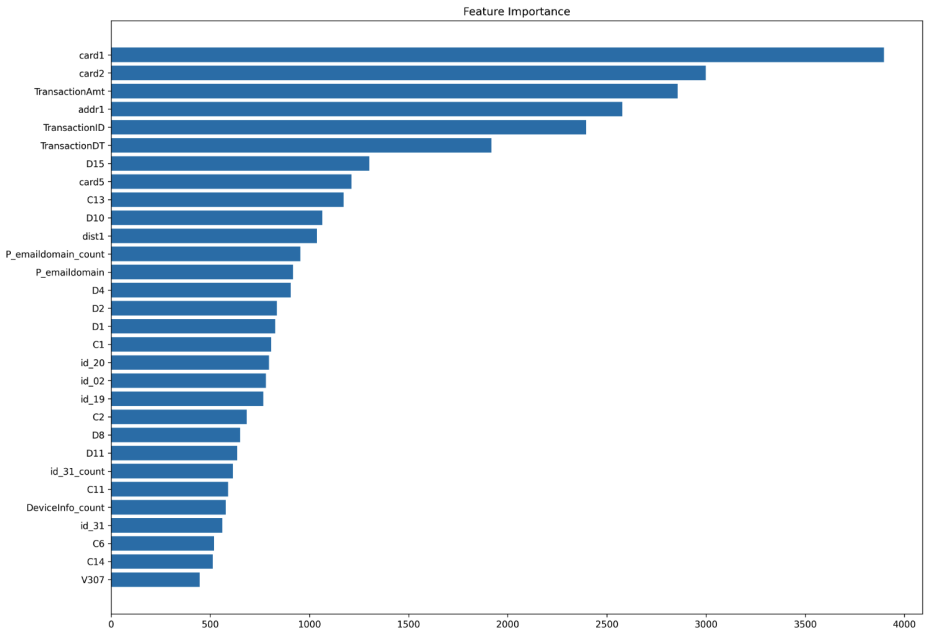

Figure 2.5: Feature Importance

Chapter 3

Method

Here we have used four very popular machine learning methods to validate this dataset, we have chosen Lightgbm, Xgboost, RandomForest, Multilayer Perceptron. below we give a brief introduction of each method.

3.1 Lightgbm

GBDT (Gradient Boosting Decision Tree) is a long-standing model in machine learning, whose main idea is to use weak classifiers (decision trees[4]) to iteratively train to get the optimal model, which has the advantages of good training effect and not easy to over-fit, etc. GBDT is not only widely used in industry, usually used for multi-classification, click rate prediction, search ranking and other tasks; it is also a deadly weapon in various data mining competitions. LightGBM[5] (Light Gradient Boosting Machine) is a framework for implementing the GBDT algorithm, which supports efficient parallel training and has faster LightGBM (Light Gradient Boosting Machine) is a framework for implementing GBDT algorithms that supports efficient parallel training and has the advantages of faster training speed, lower memory consumption, better accuracy, and distributed support for fast processing of large amounts of data.

3.2 RandomForest

Random Forest[6] (RF) has a wide range of applications, ranging from marketing to health care insurance. It can be used to model marketing simulations, count customer sources, retention and churn, and predict disease risk[7]. and susceptibility of patients. Initially, I was introduced to the random forest algorithm while participating in an off-campus competition. In recent years of domestic and foreign competitions, including the 2013 Baidu Campus Movie Recommender System Competition, the 2014 Alibaba Tianchi Big Data Competition and the Kaggle Data Science Competition, participants used a high percentage of random forests.

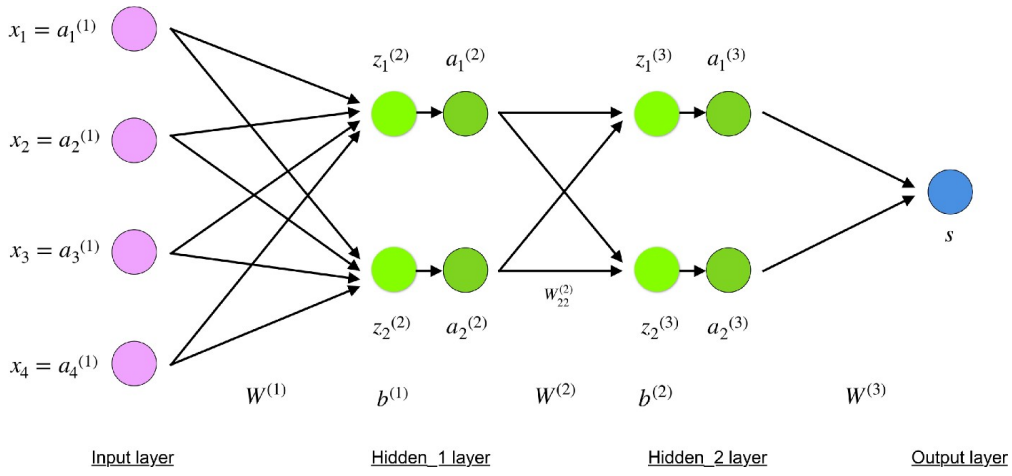

Figure 3.1: MLP Architecture

3.3 Multilayer Perceptron

MLP is also called artificial neural network[8]. A standard MLP is composed of input layer, hidden layer and output layer, of which there can be more than one hidden layer. The structure of an MLP model with two hidden layers is shown in Figure. 3.1

Chapter 4

Analysis and Results

4.1 Experiment

For the dataset, as our dataset is from a Kaggle competition, only the training set given by the competition is labelled, the test set given by the competition is unlabelled. We divided the dataset given by the competition into training set, validation set, and test set in the ratio of 8:1:1

For the model, we present the hyperparameters used one by one

For the LightGBM model, the hyperparameters we use are shown in Table. 4.1.solver represents the solver for weight optimization.max_iter represents the maximum round of model training.learning_rate represents the learning rate for training the model.hidden_layer_sizes represents the number of neurons in each layer of the model

For the RandomForest model, the hyperparameters we use are shown in Table. 4.2.n_estimators represents the number of decision trees to build.max_depth indicates the maximum depth for building a decision tree.

For the MLP model, the hyperparameters we use are shown in Table. 4.3.num_leaves represents the maximum number of leaves to build a decision tree.max_depth indicates the maximum depth for building a decision tree.lambda_l1 represents the coefficient of l1 regularization

Hyperparameter	Value
num_leaves	60
max_depth	-1
lambda_l1	0.1

Table 4.1: LightGBM Hyperparameter

Hyperparameter	Value
n_estimators	100
max_depth	6

Table 4.2: RandomForest Hyperparameter

Hyperparameter	Value
solver	adam
learning_rate	0.01
hidden_layer_sizes	(128,32)
max_iter	100

Table 4.3: MLP Hyperparameter

4.2 Metric

Our task is a binary classification task, so we use commonly used metrics for evaluating binary classification. Here we use Accuracy, Recall, Precision, F1-Score as our evaluation metrics

4.3 Result

The results of our experiments are shown in Table. 4.4, Table. 4.5, Table. 4.6, in which we show the metrics of training set, validation set, test set.It can be seen from the results that whether it is a training set, a validation set or a test set, the lightgbm model have the best effect. It can be seen from the results that the accuracy of the model is generally relatively high, but the scores of Recall and F1-Score are relatively low. This is because the proportion of positive samples in the data set is very small, resulting in a very low score for Recall, which in turn leads to low F1-Score

Similarly, when we compare Table. 4.4, Table. 4.5, Table. 4.6, we can see that the gap between the indicators of the training set, validation set, and test set is not large. Especially for RandomForest and MLP, the gap is almost 0, which also shows the robustness and

Model	Accuracy	F1-Score	Recall	Precision
LightGBM	**0.9927**	**0.8841**	**0.7945**	**0.9964**
RandomForest	0.9714	0.3342	0.2048	0.9066
MLP	0.9750	0.4960	0.3506	0.8475

Table 4.4: Metrics On Train Dataset

Model	Accuracy	F1-Score	Recall	Precision
LightGBM	**0.9860**	**0.7575**	**0.6252**	**0.9606**
RandomForest	0.9708	0.3131	0.1900	0.8891
MLP	0.9736	0.4639	0.3264	0.8017

Table 4.5: Metrics On Valid Dataset

Model	Accuracy	F1-Score	Recall	Precision
LightGBM	**0.9862**	**0.7615**	**0.6332**	**0.9551**
RandomForest	0.9713	0.3239	0.1980	0.8884
MLP	0.9748	0.4878	0.3462	0.8246

Table 4.6: Metrics On Test Dataset

generalization ability of our model

Chapter 5

Conclusions

We have thoroughly researched and compared popular machine learning models for the fraud detection problem. We used a dataset from a Kaggle competition organized by IEEE. We used the three models of Lightgbm, RandomForest and MLP on the data set of this competition, and achieved 98.62% accuracy on the test set, which also shows the reliability of our model.

Optimizing code performance can reduce crashes during training on large datasets. In this data analysis, the optimization is achieved by optimizing the string type that occupies a high memory in pandas. The temporal properties of the dataset for this competition. However, through specific feature analysis, the verification results show that the time feature does not significantly improve the accuracy of the model.

For feature selection, I process more than 300 V features by comparing train set and test set. The V feature is redundant because it is derived from the C, D, and M features. Because of the particularity of this competition, the real meaning of many features is masked, and a few are not masked. In this minority, prior knowledge can be used to make some progress of the model, especially in post-feature processing. For example, P_emaildomain can generate new features based on mailbox suffixes. Categorical data can be useful in forecasting if the distribution is very different.

After the analysis of each feature, we start to check the correlation. Pearson's correlation is generally used to evaluate the correlation of numerical features. But in this dataset, missing values can affect the evaluation of the correlation between the two features. Therefore, it cannot reflect the overall situation of the sample.

From the results of the model analysis, the generalization ability of the model is limited. Removing outlier samples does not solve the problem of poor generalization performance. The root cause of poor generalization performance is data shift. For data shift of categorical features, we try to convert categorical features into continuous values (labelencoder); for numerical features, we can try discretization or log transform.

Last but not least, we obtain the hyperparameters through Bayesian optimization to guarantee the high accuracy of the model.

Bibliography

[1] Richard J Sullivan. The changing nature of us card payment fraud: Issues for industry and public policy. In *WEIS*. Citeseer, 2010.

[2] Gary W Adams, David R Campbell, Mary Campbell, and Michael P Rose. Fraud prevention. *The CPA Journal*, 76(1):56, 2006.

[3] `https://www.kaggle.com/competitions/ieee-fraud-detection/overview/description`.

[4] Sotiris B Kotsiantis. Decision trees: a recent overview. *Artificial Intelligence Review*, 39 (4):261–283, 2013.

[5] Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. Lightgbm: A highly efficient gradient boosting decision tree. *Advances in neural information processing systems*, 30, 2017.

[6] Steven J Rigatti. Random forest. *Journal of Insurance Medicine*, 47(1):31–39, 2017.

[7] Mohammed Khalilia, Sounak Chakraborty, and Mihail Popescu. Predicting disease risks from highly imbalanced data using random forest. *BMC medical informatics and decision making*, 11(1):1–13, 2011.

[8] Sun-Chong Wang. Artificial neural network. In *Interdisciplinary computing in java programming*, pages 81–100. Springer, 2003.