# Machine Learning Methods for Credit Card Fraud Detection

Yihong He*

Dept. of Engineering, University of California, Davis, 1 Shield Avenue, Davis, CA, USA 95616-0345

*Corresponding author: yihhe@ucdavis.edu

**Abstract.** Machine learning is an innovative and efficient tool to prevent credit card fraud, however, given the variety of machine learning models, which model is the most suitable for fraudulent transaction predictions becomes a tough question to answer. In this research, a comprehensive evaluation method is borrowed to compare performances between different machine learning models. More precisely, this research uses the Area under the ROC Curve (AUC) metric to evaluate and compare performances between four different machine learning models with the same transaction information dataset. The four models are K Nearest Neighbor, Logistic Regression, Random Forest, and Support Vector Machine. In this research, a dataset that contains over one million credit card transaction data is processed and divided into training data and testing data. After preprocessing, the same training data are fitted into four different models and being test against the same testing data. After a series of hyperparameter tuning, the AUC score of each model is obtained and compared. The comparison result indicates that Random Forest makes the most accurate and consistent predictions on fraudulent transactions in this dataset, and thus can be recommended as the primary machine learning algorithm to prevent credit card fraudulent transactions.

**Keywords:** Machine learning, K-nearest neighbor, random forest, Support vector machine.

## 1. Introduction

Credit cards, as one of the most prevalent paying methods, bring convenience to our society by significantly simplifying the payment process. Statistics show that 70% of U.S citizens have credit cards, with 34% of them having 3 or more credit cards and 14% of them holding 10 or more credit cards [1]. Thousands of credit card transactions happen every day. Unfortunately, the benefits credit card brings also come with fearsome potential risk. Each credit card contains special private information: credit card number, holder's name, expiration date, etc. With this information, individuals can make purchases even if he/she is not the owner of the card. Credit card fraud emerges in scenarios like this. Credit card fraud is a fraud from identity theft that steals credit card information and creates fraudulent credit card transactions.

To avoid credit card fraud, a traditional approach is to have the banking company go over all credit card transactions and stop any suspicious transactions [2]. This approach is costly, as it takes noteworthy time and labor force to accomplish. To solve the limitation of this method to a certain extent, machine learning methods can be considered in this case. Machine learning algorithms can make predictions on fraudulent transactions efficiently and prevent fraudulent transactions in a fully automated process, which saves time and labor. There are a variety of well-developed machine learning programs that detect fraudulent transactions, however, each of them uses various techniques with different machine learning models. There are many different machine learning algorithms available, which becomes troublesome for banking companies to decide which algorithm they should pick to prevent credit card fraud. In one article, it goes over how using the random forest algorithm is able to achieve great accuracy with a 94% AUC score when predicting fraudulent transactions [3]. In another research article, KNN and Naive Bayes methods are used to predict credit card fraud, with 85% accuracy [4]. However, it is hard to use the result in these articles to compare the efficiency of the models, since they are trained and tested on very different datasets with unconnected evaluation metrics. Hence, there are a couple of questions left to be answered. Which model has the best accuracy? Which model is the most efficient? This paper will figure out which machine learning algorithms are able to make the most accurate predictions on credit card fraudulent transactions.

To effectively compare efficiency between different models, the general category of machine learning algorithms this paper will discuss is the classification models in supervised learning. Within this category, four algorithms are picked to analyze efficiency, which are K-Nearest Neighbor (KNN) [5, 6], logistic regression [7, 8], Support vector machine (SVM), and random forest algorithms. To analyze which model has good performance in detecting fraudulent transactions, a dataset from Kaggle is borrowed which contains a million credit card transaction data, each with 27 different features. Each of the four models will be fitted in the same training data and compare their performance with the same testing data. As a result, the random forest model and the KNN show excellent performance, with the area under the ROC curve being 90.26% and 93.32% respectively. SVM and random forest, while not performing as well as the other two, make acceptable predictions within the dataset, with the area under the ROC curve being 89.77% and 85.47% respectively. This paper also analyzes the runtime of each model. Notably, SVM requires significantly more CPU time to go through the data compared to the other three models.

In the following sections, this paper will first go over the research method that is used to preprocess data and build up four different machine learning models in the first section. The second section will go over the underlying concepts of each algorithm and also how it tunes hyperparameters to overcome overfitting or underfitting problems in each of the models. In the third section, this paper will compare the performance of all four algorithms with different scoring metrics, including classification reports and ROC curve scoring. In the last section, a conclusion will be derived that discusses the potential revenue that the machine learning model will bring to the banking company, and also the benefits it will have to our society.

## 2. Method

### 2.1. Dataset

In this project, the selected dataset that is used for model training and testing is a credit card transaction information dataset from Kaggle [9]. It is generated from 2019 to 2020 by transactions between over 1, 000 customers and over 800 merchants. The dataset contains 24 different features with over one million different transaction data. Table 1 presents a part of the sample data in the dataset.

**Table 1.** Sample data with first five features

| | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| Trans_date | 1/1/19 0:00 | 1/1/19 0:00 | 1/1/19 0:00 | 1/1/19 0:03 | 1/1/19 0:04 |
| cc_num | 2703186189652090 | 630423337322 | 38859492057661 | 3534093764340240 | 375534208663984 |
| merchant | Kub and Mann | Gutmann and Zieme | Buckridge | Crist | Hermiston and Farrell |
| category | misc_net | grocery_pos | entertainment | gas_transport | misc_pos |
| amt | 4.97 | 107.23 | 220.11 | 45 | 41.96 |

### 2.2. Data preprocessing

Since the number of features in this dataset is huge, decreasing dimension becomes necessary in the data preprocessing section. By skimming through the dataset, a couple of the features contain independent data that does not reveal any relationship with other features, and hence are not statistically significant. For example, the feature cc_num, which stands for the transaction number, contains a long list of random digits that does not correlate with any number in the data set. Dropping these kinds of features from the data set will significantly improve the efficiency of model training. In this regard, three features are dropped from the data set, including cc_num, trans_num, and trans_date. Moreover, there are features that are closely related to each other, and thus can be combined. For instance, features such as lat_diff, long_diff, merch_lat, merch_long, city, street, and state all represent the distance between customers and merchants. By calculating the distance with long_diff and lat_diff, the distance information of the transaction can be obtained. In this case, all the distance-related features can be dropped from the data set.

Besides dimension reduction, encoding and scaling are also necessary in order to process non-integer features and analyze the unbalanced data set. In this project, the ordinal encoding method is implemented to encode features such as 'category', 'merchant', 'job', 'location', 'city_pop_segment', and 'gender'. The unbalanced data set is also a crucial problem for credit card fraud detection, since compared to the number of normal transactions, the fraudulent transaction only takes up to 1% of the total transaction. To fix the problem, the standard scaler method from scikit-learn is used to scale our data.

### 2.3. Models' introduction

In this project, four different machine learning models will be closely analyzed, which are KNN, SVM, random forest, and logistic regression.

**KNN.** The underlying concept behind the KNN algorithm is simple. The algorithm searches for the nearest K data points given the parameter K and a data point. KNN finds the nearest data points by calculating the distance between data points with specific distance methods, such as the Euclidean distance, Minkowski distance, or Manhattan distance. With the K nearest data points, KNN can then classify the data point to the class of the majority among the K nearest data points [10].

**SVM.** The main goal of SVM is to classify data by finding an optimal hyperplane in N-dimensional space, in which N represents the number of features in the data set. In this project, the hyperplane will separate the data points into two categories: fraud and non-fraud. In order to find the optimal hyperplane, the SVM algorithm maximizes the margin between the data points. The SVM primal function can be represented as

$$min_{w,\{\xi_n\}} \frac{1}{2} \|\omega\|^2 + C \sum_n \xi_n \tag{1}$$

However, for less separable datasets, some kernel functions are needed to help transform datasets to be more separable. This research project will be focusing on tuning two hyperparameters, the cost parameter C, which determines the misclassification penalty, and the kernel parameter $\gamma$. In the end, the most optimal value for C and $\gamma$ are found to be 10 and 0.01 respectively.

**Logistic regression.** Binary logistic regression is used in this project, which is used to calculate the possibilities of fraud and non-fraud transactions. The logistic regression uses a sigmoid function to map values to its predicted outcome. The sigmoid function can be represented as [11]:

$$\mathrm{f(x)} = \frac{1}{1+e^{-(x)}} \tag{2}$$

The cost function is used to minimize errors in the model which can be represented as the following:

$$Cost(h_\theta(x), y) = -log(h_\theta(x)) \text{ if y = 1} \tag{3}$$

$$Cost(h_\theta(x), y) = -log(h_\theta(x)) \text{ if y = 0} \tag{4}$$

Besides cost functions, regularization is another essential technique for minimizing error and preventing overfitting for logistic regression. This research will be focusing on three different hyperparameters for logistic regression: C, and max_iter. The hyperparameter C determines the extent of regularization. Max_iter sets the maximum limit of the iterations for logistic regression's solver. In the end, the optimal hyperparameters found are C being 100 and max_iter being 10.

**Random forest.** The random forest algorithm makes classification in the following steps. It first draws multiple sample sets from the data set, and then one decision tree is created for each sample, each decision tree will then derive one output. With multiple outputs obtained from the decision trees, the final result will be obtained by analyzing the previous outputs with methods such as Majority Voting or Averaging. For Random forest, the hyperparameters that will be focused on in this research are N_estimators,Max_depth, Random_state, and Class_weight. N_estimator determines the maximum trees to be used in the forest, Max_depth determines the path length from root to the leaf, random_state determines the randomness in the model. Class_weight determines the weight of classes in the trees. The optimal hyperparameters found are N_estimators being 32, Max_depth being 7, random_state being 10, and class_weight being balanced.

## 3. Result and discussion

To make a fair judgment and comparison, this research uses the Receiver Operating Characteristic Curve (ROC) to measure and compare the performance of various models. The idea behind ROC is that it calculates true positive (TP), true negative (TN), false positive (FP), and false negative (FN) to plot true positive rate (TPR) and false positive rate (FPR) in the curve, which can be represented as the following formulas.

$$\text{TPR} = \frac{TP}{TP + FN} \tag{5}$$

$$\text{FPR} = \frac{FP}{FP + TN} \tag{6}$$

With the formulated ROC curve, the Area Under the ROC curve (AUC) can be obtained to evaluate the performance of the employed models. The AUC scores for all models are shown in Table 2, it shows the AUC score when models are testing against testing data and training data.

**Table 2.** Model performance

| Model | Test AUC score(%) | Train AUC score(%) |
|---|---|---|
| KNN | 90.26 | 92.9 |
| SVM | 89.77 | 89.41 |
| Random Forest | 93.92 | 92.7 |
| Logistic Regression | 84.99 | 88.74 |

From the Table 2, it can be concluded that KNN and random forest has relatively better performance than SVM and Logistic Regression with both AUC score higher than 90%. Random Forest has better performance than KNN, and SVM has better performance than logistic regression. Hence, the fraud prediction performance for the four models can be ranked as: Random Forest, Logistic Regression, KNN, and SVM.

This result indicates that Random Forest did an excellent job when classifying data and making precise predictions on both positive cases and negative cases, which in this project stands for fraudulent transactions and non-fraudulent transactions. Given that its test AUC score and Train AUC score are close to each other with only 1.22% difference, it successfully prevents overfitting and underfitting as well. The performance of KNN is also worth noting, with the Train AUC score being

92.9%, it has the best performance when testing against training data among all models in the table, however, when testing against the test dataset, its AUC score drops to 90.26%.

## 4. Conclusion

In this research, the basic ideas and some crucial hyperparameters for four different machine learning models that may help fight against the credit card fraud problem were discussed. This paper then used the AUC score to compare the model performance with the same credit card transaction data. The result of the best performer among the four models: random forest was obtained. However, it is not mature to conclude that random forest would be the best model for predicting fraudulent transactions. More credit card transaction datasets are needed to be tested to obtain a more general evaluation of model performance such as datasets with more or fewer features, or datasets that are generated from different regions. Moreover, more models should also be evaluated to find the best one to fight credit card fraud.

## References

[1] Shift, “Credit Card Statistics.” Shift Credit Card Processing, 2021, https://shiftprocessing.com/credit-card/#:~:text=70%25%20of%20the%20United%20States,dispatched%20among%20multiple%20different%20outlets.

[2] The Motley Fool, “How to Avoid Credit Card Fraud and Scams.” 2022, https://www.fool.com/the-ascent/credit-cards/scams-fraud-how-avoid/#:~:text=How%20do%20credit%20card%20companies,to%20look%20for%20unusual%20transactions.

[3] Wandre, S., et al. “Cerdit Card Fraud Detection Using KNN & Navie Bayes Algorithm.” JETIR, JETIR(Www.jetir.org), https://www.jetir.org/view?paper=JETIR2204420.

[4] Meier, T. M., “Early Detecting Credit Card Frauds.” Medium, Towards Data Science, 5 Jan. 2022, https://towardsdatascience.com/early-detecting-credit-card-frauds-38db7c190e44.

[5] Zhang, S., et al. "Learning k for knn classification." ACM Transactions on Intelligent Systems and Technology (TIST) 8.3 (2017): 1-19.

[6] Yu, Q., et al. "Clustering Analysis for Silent Telecom Customers Based on K-means++." 2020 IEEE 4th Information Technology, Networking, Electronic and Automation Control Conference (ITNEC). Vol. 1. IEEE, 2020.

[7] LaValley, M. P. "Logistic regression." Circulation 117.18 (2008): 2395-2399.

[8] Kleinbaum, D. G., et al. "Logistic regression." New York: Springer-Verlag, 2002.

[9] Probst, P., “Hyperparameters of the Support Vector Machine.” Hyperparameters of the Support Vector Machine – Philipp Probst – Statistician, Data Scientist, Football Player, Alpinist, https://philipppro.github.io/Hyperparameters_svm_/.

[10] Kumar, A., “Cosine Similarity & Cosine Distance.” Medium, DataDrivenInvestor, 5 July 2020.

[11] Rohith, G., “Support Vector Machine — Introduction to Machine Learning Algorithms,” https://towardsdatascience.com/support-vector-machine-introduction-to-machine-learning-algorithms-934a444fca47.