Machine Learning In Fraud Detection

07/31/2022

Baoshu Feng

ILLINOIS TECH

Project Overview

Fraud behavior generate huge influence to all customer experience since fraud behavior is hidden as a small partition in massive transaction. Our objective is to create a predictive model for expected fraud in the Vesta Corporation dataset, in order to identify the fraud before it happen. This fraud prevention system is actually saving consumers millions of dollars per year.

Problem Statement

- Data mining for key features set from all feature
- Generate different statistical models for fraud detection (For each TransactionID in the test set, predict a probability for the isFraud variable). (Binary problem)
- Benchmark machine learning models on a challenging large-scale dataset.

Roadmap of Project

1. Join tables
2. EDA
3. Feature Selection
4. Feature Engineering
5. Modeling
    a. LightGBM
    b. RandomForest
    c. MLP
6. Evaluation

Overview of Data: IEEE-CIS Fraud Detection

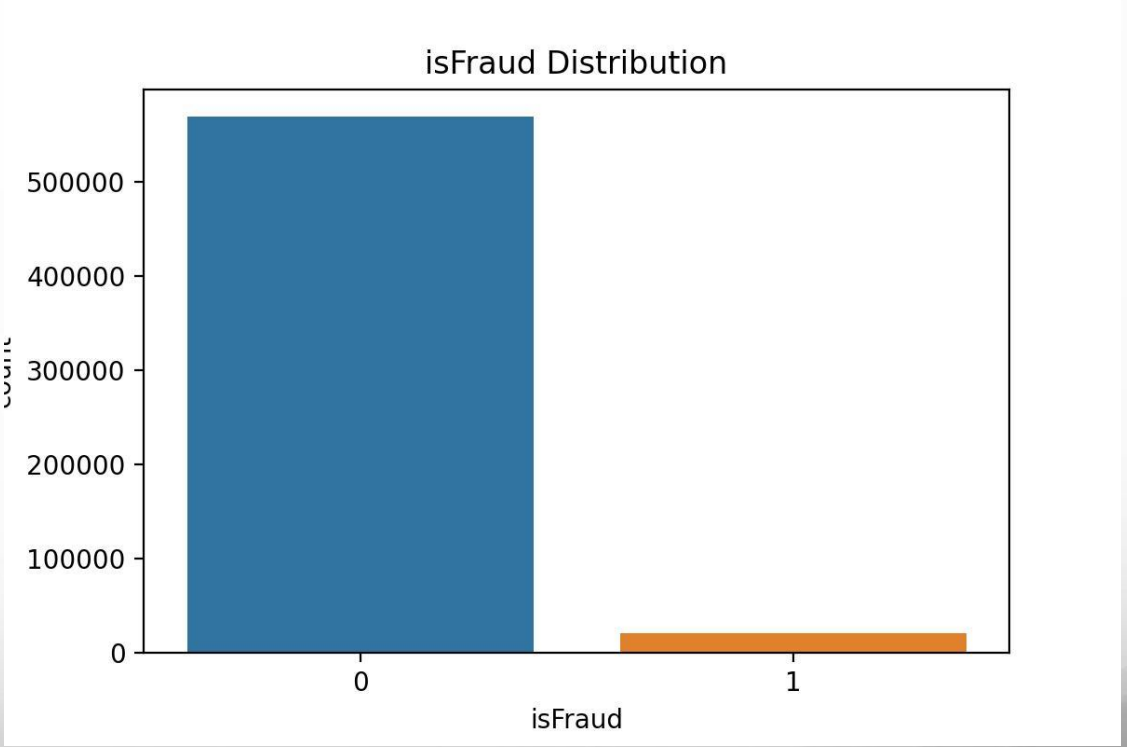
- **Transaction Table**
  - **TransactionDT**: timedelta from a given reference datetime (not an actual timestamp)
  - **TransactionAMT**: transaction payment amount in USD
  - **ProductCD**: product code, the product for each transaction
  - **card1 - card6**: payment card information, such as card type, card category, issue bank, country, etc.
  - **addr**: address
  - **dist**: distance
  - **P_ and (R__) emaildomain**: purchaser and recipient email domain
  - **C1 - C14**: counting, such as how many addresses are found to be associated with the payment card, etc. The actual meaning is masked.
  - **D1 - D15**: timedelta, such as days between previous transaction, etc.
  - **M1 - M9**: match, such as names on card and address, etc.
  - **Vxxx**: Vesta engineered rich features, including ranking, counting, and other entity relations.

Overview of Data: IEEE-CIS Fraud Detection

- **Identity Table**
  - Variables in this table are identity information network connection information (IP, ISP, Proxy, etc) and digital signature (UA/browser/os/version, etc) associated with transactions. They're collected by Vesta’s fraud protection system and digital security partners.
- Except **TransactionDT** and **TransactionAMT**, the rest of the variables are categorical features in transaction table. In identity table, DeviceType, DeviceInfo and id_12 - id_38 are categorical features.
- test_transaction: 506691 line
- test_identity: 141907 line
- train_transaction: 590540 line
- train_identity: 144233 line

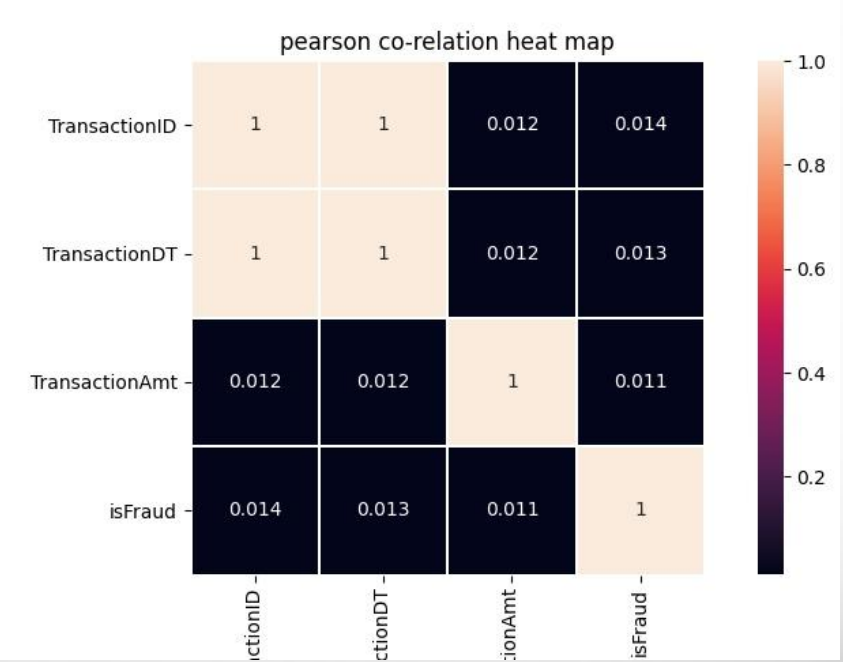
Overview of Data: Description

- Visualise the labels of the data set.
- We can see that fraud accounts for a very small proportion of this dataset.



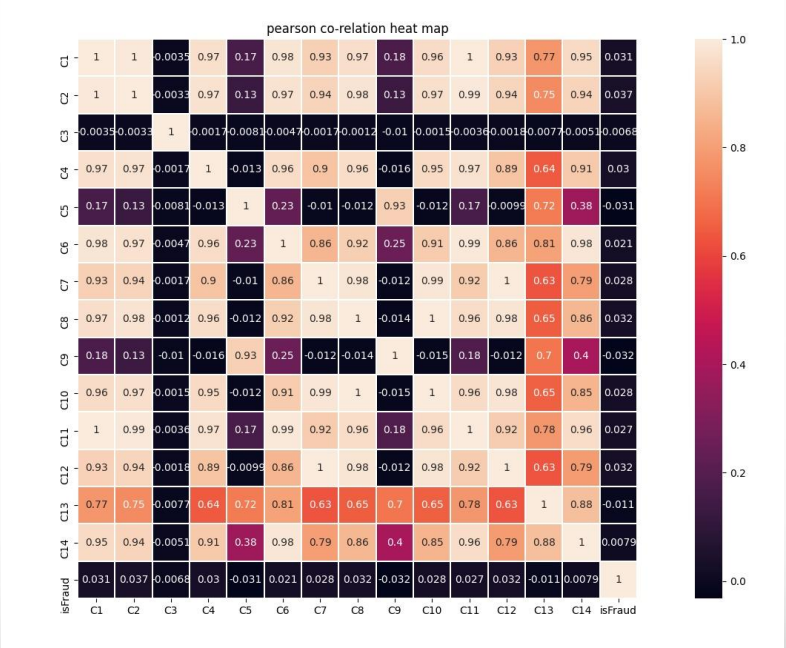
Overview of Data: Description

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.



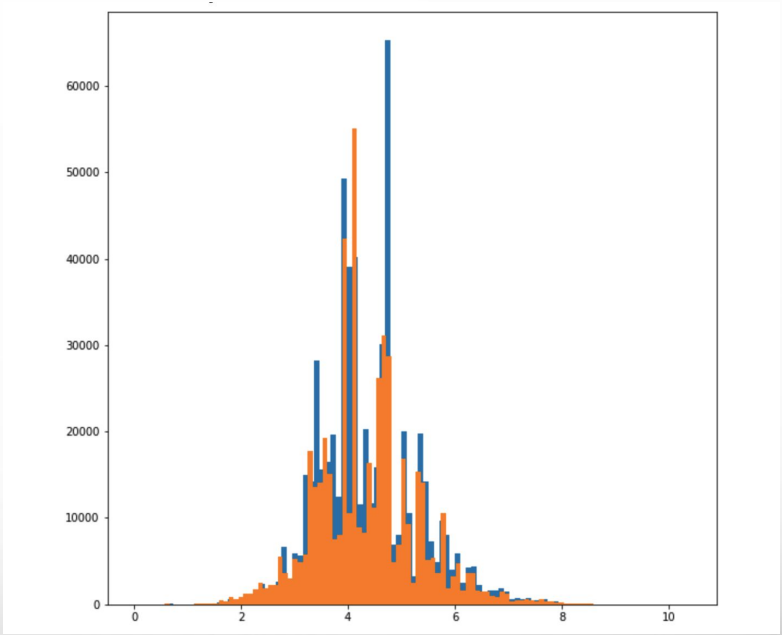
Overview of Data: Description

As the Fig. 2 and Fig. 3 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



Feature Selection & Engineering

- Transaction Amount (Log) Distribution
- 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.



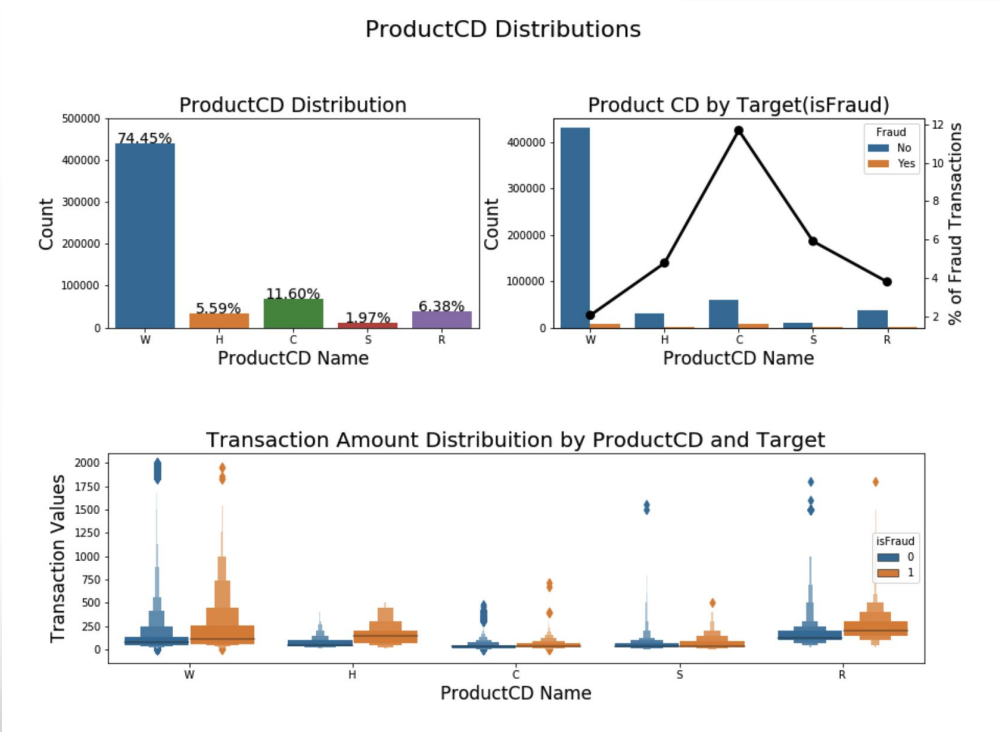
Feature Selection & Engineering

```
Class: year
Kolmogorov-Smirnov test:    KS-stat = 0.235465    p-value = 0.000e+00

Class: month
Kolmogorov-Smirnov test:    KS-stat = 0.764535    p-value = 0.000e+00

Class: day
Kolmogorov-Smirnov test:    KS-stat = 0.025674    p-value = 1.368e-156

Class: hour
Kolmogorov-Smirnov test:    KS-stat = 0.017204    p-value = 1.529e-70

Class: minute
Kolmogorov-Smirnov test:    KS-stat = 0.002595    p-value = 5.066e-02

Class: weekday
Kolmogorov-Smirnov test:    KS-stat = 0.016545    p-value = 2.826e-65
```


```
Class: year
Kolmogorov-Smirnov test:    KS-stat = 0.235465    p-value = 0.000e+00

Class: month
Kolmogorov-Smirnov test:    KS-stat = 0.764535    p-value = 0.000e+00

Class: day
Kolmogorov-Smirnov test:    KS-stat = 0.025674    p-value = 1.368e-156

Class: hour
Kolmogorov-Smirnov test:    KS-stat = 0.017204    p-value = 1.529e-70

Class: minute
Kolmogorov-Smirnov test:    KS-stat = 0.002595    p-value = 5.066e-02

Class: weekday
Kolmogorov-Smirnov test:    KS-stat = 0.016545    p-value = 2.826e-65
```

Feature Selection & Engineering

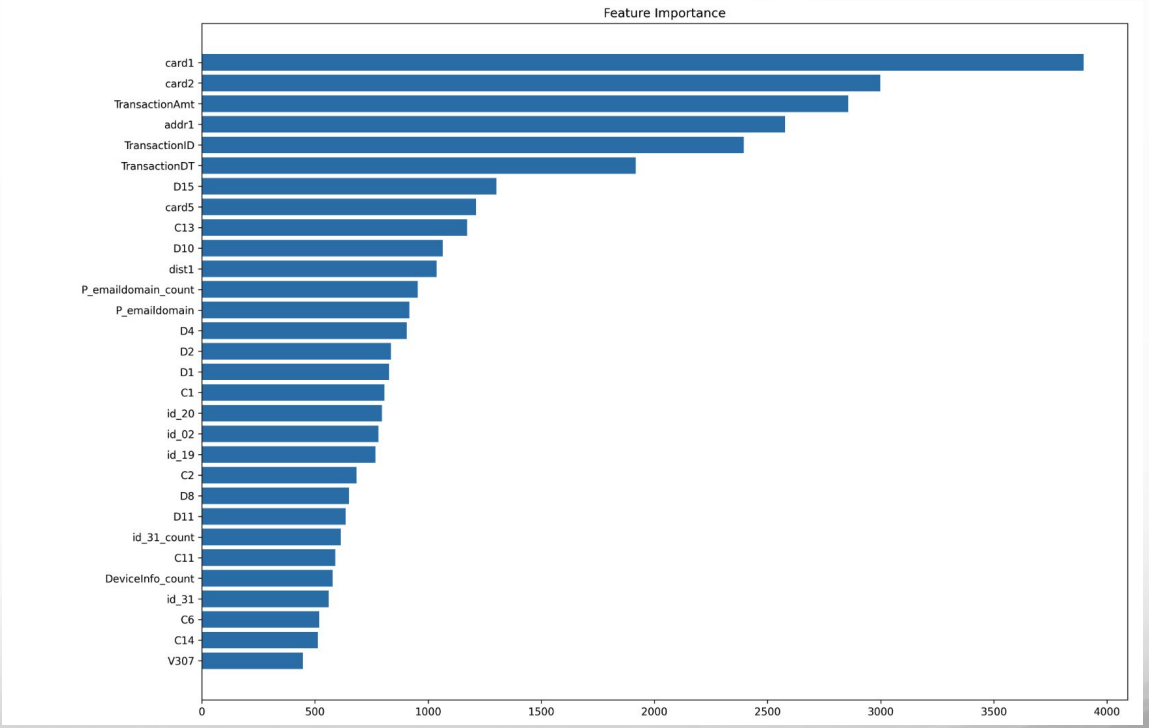
On the whole, the feature distribution of productCD is relatively stable, and there is no obvious difference.



ILLINOIS TECH

Feature Selection & Engineering

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.



ILLINOIS TECH

Machine Learning Model I - LightGBM

The full name of LightGBM is a lightweight gradient boosting machine, which is a top-level Boosting algorithm framework. Like XGBoost, LightGBM is also an engineering implementation of the GBDT algorithm framework, but it is faster and more efficient. Detail of the implementation of LightGBM include four directions, histogram algorithm, unilateral gradient sampling, exclusive feature bundling algorithm and leaf-wise growth strategy [6].

Machine Learning Model I - LightGBM

For the LightGBM model, the hyperparameters we use are shown in Table. 4.1.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

Machine Learning Model II - RandomForest

Random forest (RF) is an ensemble learning algorithm designed based on the Bagging framework. Random forest is integrated with decision tree as the base classifier, and further introduces the method of randomly selecting data features in the decision tree training process. The random forest is named because of this randomness in the process of building the model [5] [7] [8].

Machine Learning Model II - RandomForest

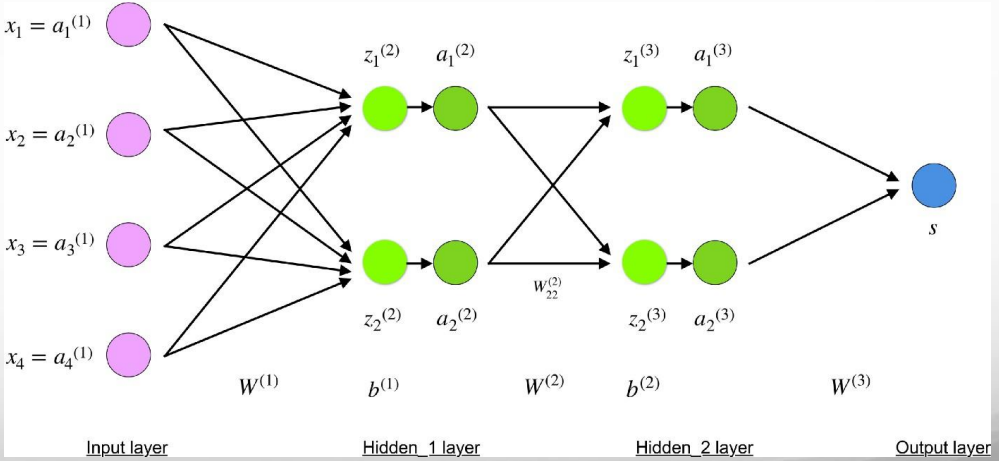
For the RandomForest model, the hyperparameters we use are shown in Table. 4.2.n estimators represents the number of decision trees to build.max depth indicates the maximum depth for building a decision tree.

Hyperparameter	Value
n_estimators	100
max_depth	6

Table 4.2: RandomForest Hyperparameter

Machine Learning Model III - Multilayer Perceptron

MLP is also called artificial neural network[9]. 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 [9].



Machine Learning Model III - Multilayer Perceptron

For the MLP model, the hyperparameters we use are shown in Table. 4.3. 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

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

Table 4.3: MLP Hyperparameter

Evaluation

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

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

Evaluation

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

Optimization

- Code Performance
  - optimizing the string type
- Feature Engineering
  - prior knowledge
- Correlation of Features
  - missing values
- Generalization Performance
  - data shift issue
- Hyperparameters

Conclusion

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.

Reference

[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.