Final Report

Group 7: Stephen Ling, Chenghan Wu, Jesse Ying, Zeyang Yu, Renee Li

Introduction (Abstract)

A fraud transaction is the unauthorized use (steal) of an individual's accounts or payment information, which can lead to huge losses for both business and personal. The data we use is provided by *Vesta Cooperation* on *Kaggle*. In this project, we select explainable variables and unexplainable variables that have a significant effect on the correctness of prediction. We perform the data imputation, binning, and one-hot encoding on selected variables and train data through Decision Tree, kNN, Logistic Regression, and One-class Classification. Finally, due to its relatively higher accuracy and shorter training time, we select the *kNN* with $k = 14$ and *metric = Euclidean*. Based on results, we can conclude that fraud transactions usually happen on mobile devices. Also, credit cards are more vulnerable to fraud transactions than debit cards, and *Discovery* credit cards are more vulnerable than other credit card brands like *Visa* and *Master Card*. Usually, fraud transaction tends to have a small transaction amount (below 20 dollars).

Data

The data is retrieved from *Kaggle* uploaded by *Vesta Cooperation* (https://www.kaggle.com/competitions/ieee-fraud-detection/data). The data contains 590,540 observations and 434 variables.

Questions

In this project, we are interested in detecting fraud transactions from transaction records of customers for shopping websites. In specific, we would like to explore two statistical questions:

- *What factors can be used to identify a Fraud Transaction?*
- *Which statistical methods can effectively predict whether an online transaction is a fraud transaction?*

Figure 1

Figure 2

Based on our question and interest, we visualize the ***response variable*** isFraud. However, the fraud transaction (isFraud = 1) only occupies 3.5% of the whole data (*Figure 1*), which suggests our data is **imbalanced**. As a result, we **undersample** the majority class which is isFraud = 0, so that we have balanced data where fraud and not fraud transaction records each take 50% of the data (*Figure 2*).

Variable Selection

By comparing the **card type** proportion in the whole data, the number of Debits vs. Credit is about 3:1. However, for fraud transactions, the number of Debit vs. Credit is about 1:1. This suggests **credit cards** are more vulnerable to fraud transactions. (*Figure 3*)

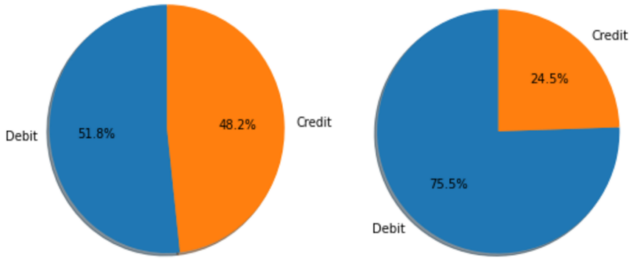

Figure 3

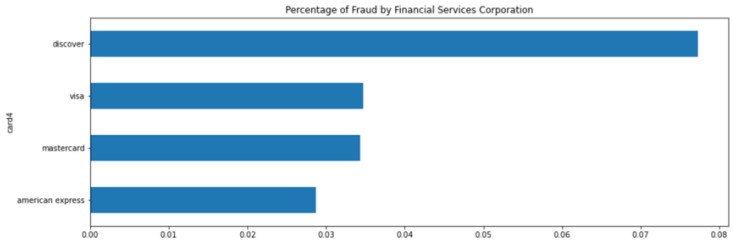

Figure 4

By observing the **credit card brand**, we find that **Discover**'s credit cards have a higher proportion of conducting fraud transactions than other variables. (*Figure 4*)

	Amount.group	isFraud
0	1	0.084450
1	2	0.035868
2	3	0.028729
3	4	0.030721
4	5	0.052512
5	6	0.020588

We also manually divide the Transaction Amount into 6 groups (0, 20, 50, 100, 300, 2000, above 2000). It is obvious that small amount transactions (< 20 dollars) are more likely to be fraud transactions. Also, the large transactions (> 2000 dollars) are the least likely to be fraud transactions.

We finally select following variables: isDiscovery, isMobile, isDesktop, isCredit, isProductC, if_amount < 20, Amount.group, V288, V282, V284, V285, V286, V287

Since there are 434 variables, we would like to select two types of variables: 1. Variable that we can interpret (e.g. Card Type, Transaction Amount, etc.); 2. The variable that has fewer missing data contributes a lot to the accuracy of the model. Since most selected variables are categorical, scaling is not necessary. However, since there are a lot of missing data, we perform the data imputation. For categorical variables, we perform the **mode imputation**. For numerical variables, we perform the **feature mean imputation**.

We perform the **one-hot encoding** to transforms categorical features into binary feature on variables: ProductCD (product type), card6 (card type), card4 (credit card brand), and DeviceType (mobile/desktop).

Besides, we create a variable isUnder20 which equals 1 when the transaction amount is below 20 dollars and equals 0 when the amount is not below 20 dollars. Then, we try **binning** on TransactionAmt which divides the transaction amount into the following groups (0, 20, 50, 100, 300, 2000, above 2000). Then, we compare the accuracy increase caused by isUnder20 and the binning variable. The result shows that the combination of numerical variables TransactionAmt + isUnder20 leads to the highest accuracy increase.

Machine Learning Models

We first split our data into the train (80%) and test (20%) data. Then, we train our data through 3 supervised learning models: Decision Tree (ID3), kNN, and Logistic Regression. Also, we filter out the fraud transaction

and remove the response variable (isFraud) and train our data through an unsupervised learning model: One-class classification.

Name	Accuracy
Decision Tree	74.1%
kNN	74%
Logistic Regression	72.3%
One-class Classification	66%

We tend to pick the learning model based on training speed and accuracy. Since kNN and Logistic Regression have much faster training speed than the other two but kNN has much higher accuracy than Logistic Regression on the same training and testing data, we finally pick **kNN** as our learning model.

Then, we conduct the **hyperparameter tuning** on kNN. We first explore the relationship between k and accuracy. The *Figure* 5 shows that when *k = 14*, the accuracy reaches its maximum value. Moreover, we compare the 5 distance calculation method, and table below shows *Euclidean* has the most accurate prediction.

Euclidean	72.9%
Manhattan	72.8%
Chebyshev	72.3%
Minkowski	72.9%
Hamming	72.4%

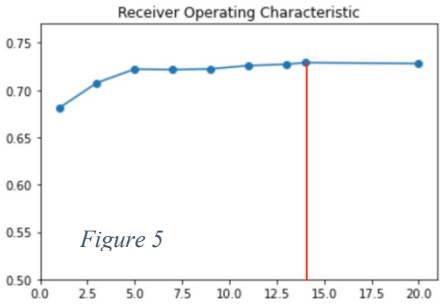

Figure 5

Overfitting Issue: based on the description above, we carefully select and test each variable combination so that the overfitting issue is not obvious to our model.

Summary & Assessment

```
The confusion matrix is:
      0     1
0  3175   958
1  1290  2843
```

The confusion matrix suggests:

Precision = 2843 / (958 + 2843) = 74.8%;

Recall = 2843 / (1280 + 2843) = 68%

There are some weaknesses of work: The Precision and Recall suggest the ability of our model to find all positive examples is not strong, even though oversampling is performed on training data. The usual way to handle categorical variables is to use the one-hot encoding. As a result, it probably produces multicollinearity among the various variables and lowers the accuracy of the model during training. Also, we should further consider the weight of different variables in the whole model, since some variables, such as types of credit cards, are significant to model performance, but some are not as important as other variables. Thus, we believe that building a scorecard based on variables could improve accuracy in predicting fraud. Finally, we can also consider using PCA to obtain a better visualization of the results (predictions).

Conclusion

In conclusion, fraud transactions usually happen on mobile devices. Also, credit cards are more vulnerable to fraud transactions than debit cards, and Discovery credit cards are more vulnerable than other credit card brands like Visa and Master Card. Usually, fraud transaction tends to have a small transaction amount (below 20 dollars). Besides, kNN with k = 14 and metric = Euclidean has the most accurate prediction of fraud transactions on provided data.

In future work, we may need some other better feature engineering methods and more precise data selection models. Also, we can explore other machine learning models like Naïve Bayes, Convolutional Neural Networks (CNNs), etc.

Contributions

1. **Proposal**: *Stephen Ling, Jesse Ying, Renee Li
2. **Exploratory Data Analysis**: *Stephen Ling, *Chenghan Wu, Jesse Ying, *Zeyang Yu
3. **Machine Learning Models**: *Chenghan Wu, *Zeyang Yu, Stephen Ling
4. **Model Assessment**: *Chenghan Wu, *Stephen Ling
5. **Presentation Slides**: Renee Li, Jesse Ying, *Stephen Ling, *Zeyang Yu
6. **Final Report**: *Stephen Ling, *Chenghan Wu

** suggests the main contributors of the project*