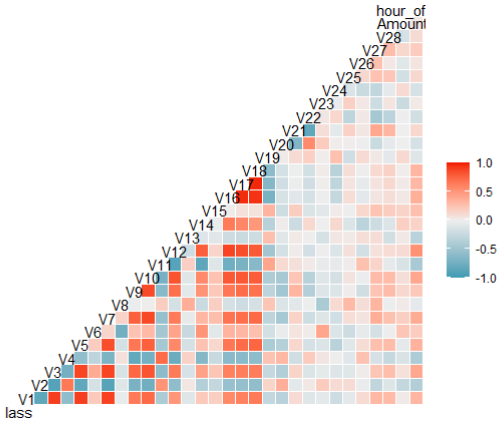
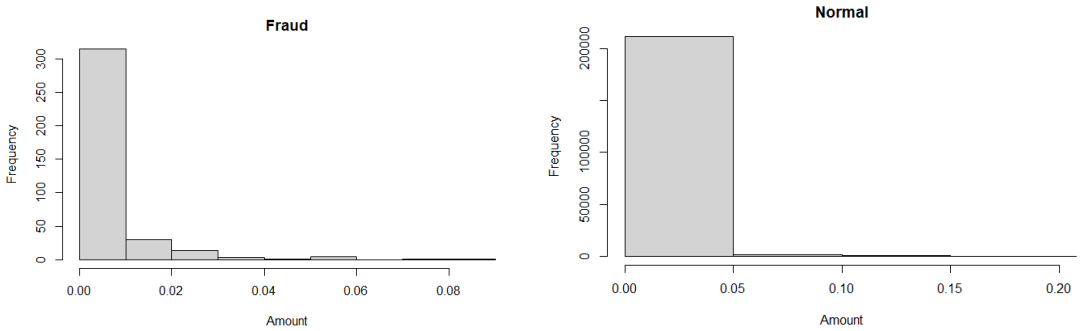
Credit Card Fraud Detection

1. Data Exploration

From the credit_train dataset, we can observe that among the 213,606 transactions in the data set, 369 are credit card fraud transactions, which account for 0.17% of the total data.



The heat map above shows the correlation among variables when *Class* is one. We can see that in the incident of credit card fraud, the correlation between some variables is more obvious. Among them, variables *V1*, *V2*, *V3*, *V4*, *V5*, *V6*, *V7*, *V9*, *V10*, *V12*, *V16*, *V17* and *V18* have strong correlation with other variables in the samples of credit card fraud.



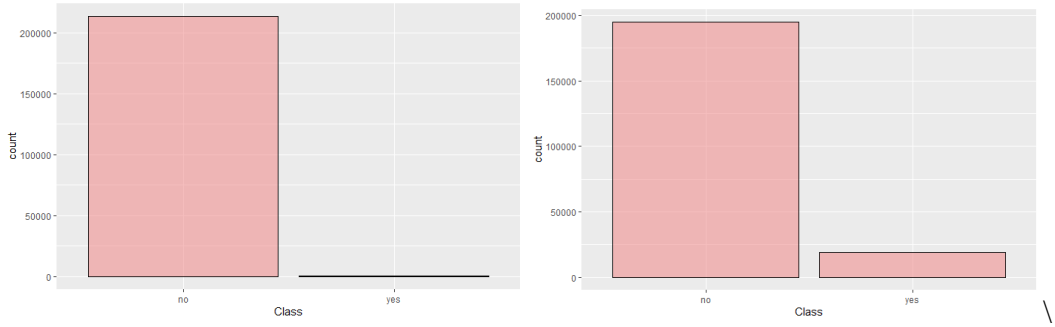
The amount of credit card fraud transactions is scattered and small compared with the amount of normal use of credit card, which shows that credit card thieves prefer to consume small amount in order not to attract the attention of credit card owners.

2. Model Training

2.1 Data Adjustment

Since the *Class* distribution of the original data is highly unbalanced, our group

first uses the SMOTE function in R programming language to generate a new dataset to address the problem. We also adopt the SMOTE plus undersampling method to our data, to increase the ratio of Fraud in our dataset.



After testing accuracy of test part several times, dup_size = 50 is used for our classification. The left graph above shows that in the original credit_train dataset, 213237 of all data are classified as no and 369 are classified as yes. After running these two methods, 194787 of all data are classified as no and 18819 are classified as yes, the ratio of credit card fraud becomes 0.0966.

2.2 XGBoost Algorithm

The XGBoost algorithm is a relatively new tree-based classification algorithm, and it is effective. There are many examples in practical application that its superiority even exceeds that of the random forest algorithm. Therefore, in this project, our group chooses XGBoost algorithm to train our model.

Using the original credit_train dataset, we set 67% of all data to be train set and the rest to be the test set, given that the predicted data set accounts for 71201/213606 = 0.33 approximately.

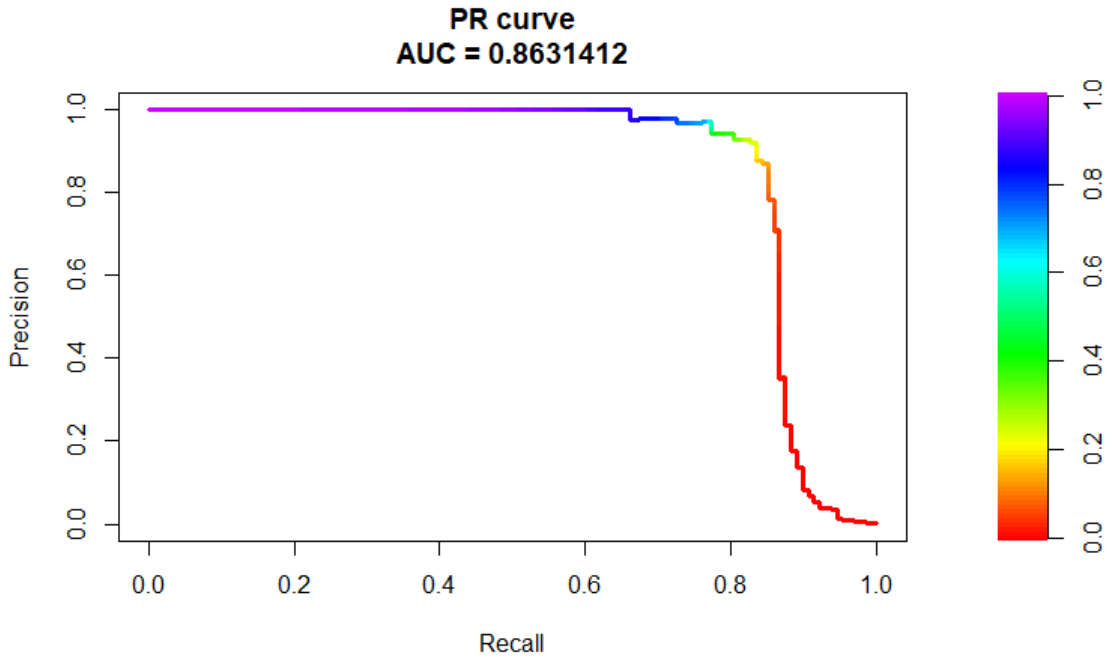
2.3 Tuning Parameters and Threshold Selection

We tune parameters by choosing the best parameters according to the area under precision-recall curve of subsample and also the actual values of recall since we want to minimize the cost caused by false negative.

Parameters are chosen as

$$nrounds = 200, eta = 0.3, gamma = 0.2, max\ \_depth = 5, min\ \_child\_weight = 5, subsample = 1, colsample\_bytree = 1$$

The following PR curve for subsample is shown below.



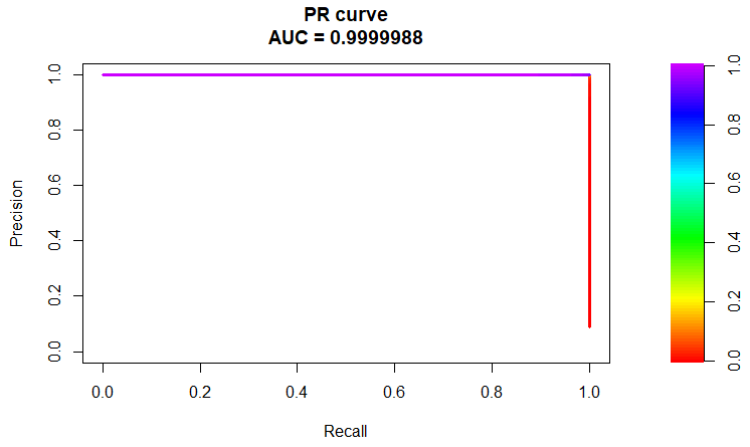
For the prediction result of the test set, we set the threshold to be 0.07 based on the distributions of Zero One probability for test set after many times in order to minimize cost.

We obtain the following confusion matrix of our separated test.

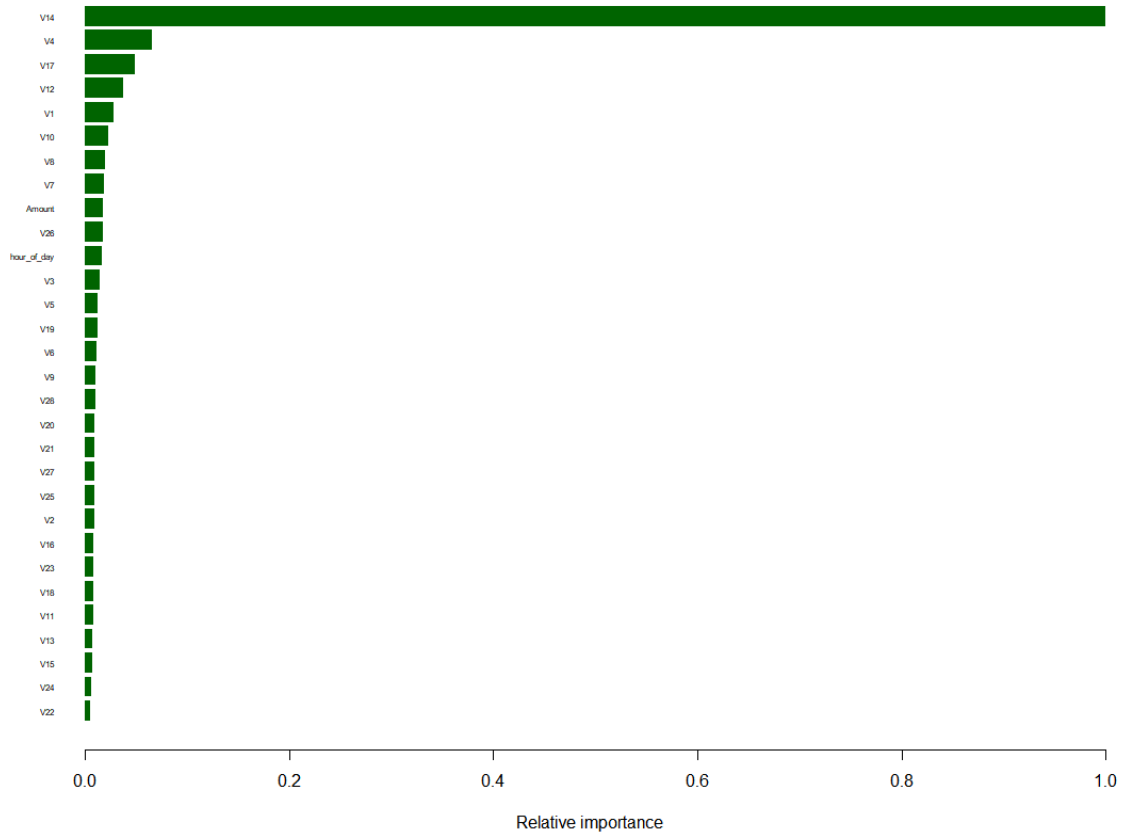
Prediction \ True	no	yes
no	70330	18
yes	32	109

3. Final Model

Using the original dataset after SMOTE and undersampling, we use the XGBoost algorithm to build our final classification model. The Area Under the Precision-Recall Curve of the final model in whole credit_train is about 0.9999988.



This figure below displays the relative importance of our model analysis



We can see that V14 accounts for a large proportion for classification.