# Credit Card Fraud Detection

Report from group 2

LI SHIYU: Framework/Visualization/Feature selection/Preprocessing/Unsupervised methods

MARCUS CHAN HUA RUI​: XGBoost

XU SHANGQING: Random Forest/Conclusion

YANG YINGTONG: Logistics regression/SVM

ZHANG YANG: Decision tree

(Equal contribution)

# 1 Problem Analysis

In general, credit card fraud is an unbalanced classification problem. Researchers get some information about the customer's personal information and the data of a specific transaction and build a model to judge whether this transaction is a fraud transaction. The problem of credit card fraud detection plays a very role in bank risk management.

# 2 Objective

Our objective of this project is to build a model based on given information, which can judge whether a transaction is a fraud transaction with a satisfactory precision and get some insights from the results.

# 3 Dataset Description and Visualization

The dataset contains transactions made by credit cards in September 2013 by european cardholders, presenting transactions that occurred in two days, where we have 492 frauds out of 284,807 transactions. The dataset is highly unbalanced, the positive class (frauds) account for 0.172% of all transactions.

It contains only numerical input variables which are the result of a PCA transformation. Due to confidentiality issues, we cannot get the original features and more background information about the data.

表格

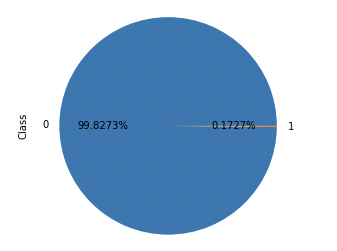
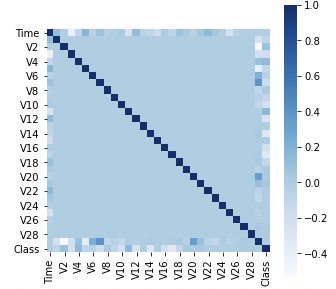
描述已自动生成

**Figure 1:** Description of the dataset

Features V1, V2, … V28 are the principal components obtained with PCA, the only features which have not been transformed with PCA are 'Time' and 'Amount'. Feature 'Time' contains the seconds elapsed between each transaction and the first transaction in the dataset. The feature 'Amount' is the transaction Amount. Feature 'Class' is the response variable and it takes value 1 in case of fraud and 0 otherwise.

By doing data description and visualization, we got the main characteristics of the data:

1. The dataset dimension is 284807\*31 (284807 transactions with 31 variables provided), no missing data.
2. Heat map shows that V1-V28 uncorrelated. Since V1-V28 are the principle component after applying PCA, they are definitely uncorrelated with each other. The uncorrelation of most of the features make it convenient for us to find the relationship between features and response variable.
3. The response variable Class is extremely unbalanced.



**Figure 2**: Heat map of the features **Figure 3:** Distribution of feature Class

Data description and visualization gives the insights:

1. Resampling methods are used here to deal with unbalanced problem.
2. From the correlation matrix, V2, V3, V4, V7, V10, V11, V12, V14, V16, V17, V18 are highly correlated with the response variable Class. These features should play significant roles in classification.

# 4 Feature Engineering

Since there are 31 principles variables, we consider to first select some important features and drop the useless one.

As the statement above, correlation matrix gives the insight that V2, V3, V4, V7, V10, V11, V12, V14, V16, V17, V18 should be important. From other perspective of view, drawing distributions of variables in different class 0 and 1 can help judge which variables distributes differently in different class, which gives the insights that which features should be important in classification (figure 4).

The result of heat map and distribution figure give the highly consistent insights that V2,V3,V4,V7,V10,V11,V12,V14,V16,V17,V18 are important and V8, V13, V15, V20, V21 , V22, V23, V24, V25, V26, V27, V28 are useless.

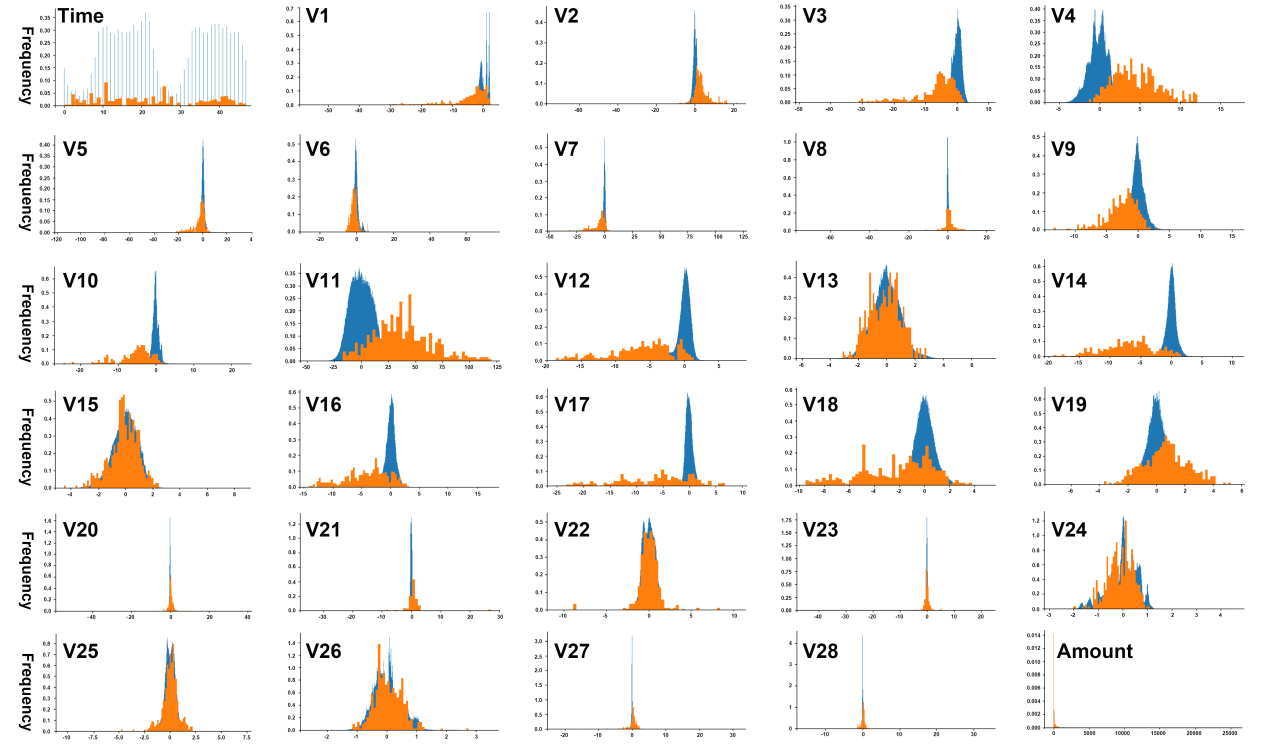
Based on the results of feature analysis, V8, V13, V15, V20, V21, V22, V23, V24, V25, V26, V27, V28 are dropped.

# 5 Data Preprocessing

5.1 Split the datasetUse stratified split to split the train (80%) and test (20%) dataset. Using stratified is based on the assumption that the ratio of fraud won’t change with time duration and avoid randomness of making the ratio in train and test differs a lot (make sure the ratio of fraud is the same in training and testing dataset.

5.2 ScalingV1-V28 are already been scaled. For some methods, scaling variable  
“Time”, “Amount” to make sure the performance or improve the convergent speed.

5.3 ResamplingSince the response variable Class is extremely unbalanced, consider using resampling to deal with the dataset. Five resampling methods are prepared to deal with the dataset: SMOTE, ADASYN, SMOTETomek, SMOTEENN, RENN. When fitting the model, use the one that is suitable for specific classification method or make comparison of all of them.



**Figure 4:** Distribution of features in Class 0 and Class 1

# 6 Unsupervised Methods

The unsupervised method used here is the methods used in outlier detection. Since fraud samples are rare in the dataset, they can be viewed as outliers and been detected by outlier detection methods.

These outlier detection methods are based on the characteristics space of the features and actually don’t need to have a training process here.

## 6.1 Isolated Forest

The purpose of isolated forest algorithm is to separate outliers from other samples. Different from other model-based anomaly detection methods, the isolated forest does not construct the profile of normal instances, that is, it does not analyze the normal cases, but explicitly isolates the outliers. Isolated forest is a classification method that uses a random hyperplane to recursively segment the dataset until all sample points are isolated. Because the outliers are less and the characteristics are different from the normal values, they are easier and more isolated than the normal values. Therefore, outliers usually appear at the root of the isolation tree, that is, outliers usually have a shorter average path length on the isolation tree. Intuitively speaking, clusters with higher density in space need to be cut more times to isolate each point, while outliers can be isolated if they are far away from each other and the number of outliers is less. Because the cutting process is completely random, the method of quantifying the "number of cuts" is to cut repeatedly from the beginning, and then calculate the average value of each segmentation result. The set with smaller average value is considered as abnormal value.

Since the isolated forest is tree-based and unsupervised, here we don’t apply data preprocessing (dropping features or scaling) to the dataset. Use isolated forest to detect the fraud samples, the parameters are as follows.

clf = IsolationForest(contamination = 0.03,n\_estimators = 100,max\_samples = 0.6,max\_features = 0.6,random\_state = 2)

The sensitive parameter is contamination, can be considered as the proportion of the outliers. Setting contamination=0.03, means that isolated forest will classify 0.03\*284807=8545 samples into fraud. Usually the contamination will be set as 5-10 times of the true proportion. If we change the contamination, then the result will change a lot. The confusion matrix is in table 8.

## 6.2 One-class SVM

One class problem refers to that there is only one class of training samples for the model, which is positive class in anomaly detection. One class support vector machine (OCSVM) can judge whether a sample does not belong to class A. as for whether it belongs to class B or class C, OCSVM does not care. OCSVM mainly has two solutions. One is to construct an optimal hyperplane in the feature space based on the dichotomy idea, and make the distance between the origin and the optimal hyperplane as large as possible under the premise of ensuring the separation of the sample and the origin; the other is to find a hypersphere as small as possible in the feature space, so that the hypersphere contains the vast majority of samples.

Here we scaled the variable Time and Amount, for SVM method is sensitive for distance. Use OCSVM to detect the fraud samples, the parameters are as follows.

OCSVM = svm.OneClassSVM(nu = 0.03,kernel = "rbf")

The parameter nu is like the parameter contamination in isolated forest, can be seen as the proportion of outlies. The confusion matrix is in table 1.

## 6.3 Autoencoder

Automatic encoder is a kind of artificial neural network, which can learn efficient data value coding in an unsupervised way. The purpose of automatic encoder is to learn the expression (encoding) of a set of data, which is usually used to reduce the dimension. Both the decoder and the encoder learn, so that the automatic encoder attempts to generate a representation as close to its original input as possible from the reduced dimension encoding.

Autoencoder can be understood as a process of compressing features and playing back features. Normal data features can be well recovered, but outliers cannot. The comparison between recovered results and original features give the results that which one is outlier (big difference between recovered and original).

Here we scale the variable Time and amount to build the neural network. The setting of the autoencoder is as follows (figure 5):

图形用户界面, 文本, 应用程序

描述已自动生成 图表, 折线图

描述已自动生成

**Figure 5:** Setting of the autoencoder **Figure 6**: ROC curve of the autoencoder

By changing the threshold, we can find a model more suitable for practical situation.

图表, 折线图

描述已自动生成

**Figure 7:** TPR and TNR in different threshold

Finally choosing threshold=3. The confusion matrix is in table 1.

In general, unsupervised methods don’t give the precision as good as supervised method, it is hard to give some insights about the dataset and how to improve the methods. The advantage is that they are easy and convenient to use and don’t need researcher to have lots of insights about the dataset.

**Table 1:** Result for unsupervised method

表格

描述已自动生成

# 7 Supervised Learning

## 7.1 Logistic Regression

Logistic Regression is a statistical method used to model the probability of binary dependent variable. In this case, with an objective to detect whether a transaction is fraud or not, logistic regression is one of the most straight-forward methods suitable for this scenario. In regression analysis, logistic regression is estimating the parameters of the fitted logistic model. Here, the fraud cases are marked as 1 in the binary dependent variable, while marked as 0 otherwise.

Before fitting the training data into the model, some preprocessing procedures mentioned previously are implemented. Scaling is used to avoid putting too much emphasis on time and amount variables other than variables after conducting principal components analysis.

After conducting the five alternative resampling methods grid search method is used to tune the parameters based on each resampled training data.

By using the grid search method to tune the parameters of different resampling methods based on the AUC score, the best model’s hyper-parameter is listed in table below.

**Table 2**: Parameters for Logistic Regression

|  |  |  |
| --- | --- | --- |
|  | C | Penalty |
| RENN | 0.001 | l2 |
| ENN (sot) | 0.001 | l2 |
| SMOTE (sos) | 0.001 | l2 |
| SMOTETomek (kos) | 0.001 | l2 |
| ADASYN (aos) | 0.001 | l1 |

By comparing the results of the five models after tuning the parameters, the results are evaluated by five indices as follows. Here, the precision rate, recall rate, f1-score and AUC score are displayed by using macro average results.

**Table 3:** Logistic Results

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Accuracy Rate | Precision | Recall | F1-score | AUC |
| RENN | 1 | 0.93 | 0.78 | 0.84 | 0.972 |
| ENN (sot) | 0.98 | 0.53 | 0.93 | 0.55 | 0.9755 |
| SMOTE (sos) | 0.98 | 0.53 | 0.93 | 0.55 | 0.9751 |
| SMOTETomek (kos) | 0.98 | 0.53 | 0.93 | 0.55 | 0.9751 |
| ADASYN (aos) | 0.9 | 0.51 | 0.92 | 0.49 | 0.9751 |

The Logistic Regression performed quite well all the five resampling methods when comparing the AUC scores. However, when comparing the accuracy rate, precision rate and f1-score, RENN methods shows a better result than the other four. And RENN performed worst when taking recall rate as the comparison criteria. In conclusion, to ensure the consistency of the whole project, AUC is used as the main criterion for choosing models. Therefore, Logistic regression performed on data combining over- and under-sampling using SMOTE and Edited Nearest Neighbors is chosen as the best model for logistic regression.

## 7.2 XGBoost

XGBoost classifier is an implementation of the gradient boosted classifier, which was essentially a collection of weak learners that could, together, quite well make predictions for the classification of outcomes for various datasets. Gradient boosting draws on the idea of iterating through which examples had not done well previously and improving the next subsequent tree to predict for those misclassifications. However, as this could easily lead to overfitting, these trees are meant to be weak learners, hence the tree depths and number of nodes may be limited. One drawback of this is that it is extremely slow, especially for large datasets, as it has to go through many features and possible branches. To combat this, xgboost has several hyperparameters, such as eta (learning rate, max\_depth, max\_leaf\_nodes, gamma, and more). However, for this project, I focused on only some of the more significant parameters, such as max depth, gamma, learning rate, early stopping, reg\_lambda, and scale\_pos\_weight due to the data being imbalanced.

After cross validation, and fitting the data, the model performed moderately well, with the following hyperparameters:

* 'gamma': 0.25,
* 'learning\_rate': 0.1,
* 'max\_depth': 5,
* 'reg\_lambda': 0,
* 'scale\_pos\_weight': 350

Furthermore, the model was able to achieve the following results:

**Table 4:** XGBoost Results

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Accuracy Rate | Precision | Recall | F1-score | AUC |
| 0.9982 | 0.75 | 0.92 | 0.81 | 0.9227 |

Xgboost typically performs well for large and complicated datasets, which is why they are pretty good results here. However, its performance is limited by the fact that it is susceptible to overfitting. It is essentially gradient boosting with regularisation parameters. In the case of very noisy data, xgboost is likely to learn from the trees based on the noisy data as well. Due to its black-box like nature, xgboost is hard to use as a model to decipher and interpret the data, since visualization will be difficult with that many trees. It is good to use as a starting model to benchmark what kind of accuracies or AUC results we should be achieving.

## 7.3 Decision Tree

A decision tree is a flowchart-like tree structure where an internal node represents feature, a branch represents a decision rule, and each leaf node represents the outcome.

Decision tree has many advantages. It can be used for feature engineering such as predicting missing values, and it is suitable for variable selection. Decision Tree is a white box type of ML algorithm. It shares internal decision-making logic, which is not available in the black box type of algorithms such as Neural Network. However, its training time is much faster than the neural network algorithm. At the same time, decision trees can handle high dimensional data with good accuracy.

Decision tree still has some disadvantages. Its calculation can go far more complex compared to other algorithms. It is highly sensitive to noisy data and any small changes in the dataset will result in large changes in its tree structure. Single Decision tree is often a weak learner, so for further requirement, random forest is a good choice for better prediction. The preprocessing of data. Though visualization plots, it is not difficult to find that some variances are not relative to results. Hence, these fewer relative columns have been deleted. And these data have been randomly split into train data and test data. Secondly, the visualization plots show the original data are imbalance. After data progressing, accuracy, precision, recall, f1 score and AUC are calculated for each resampling method. After comparison, RENN performs best among all the resampling methods using Decision Tree. The last step is Hyperparameter tuning. Although using Grid Search CV often takes long time, benefit is that users can get the best parameter settings possible after they run it on a broad parameter space.

**Table 5:** Decision Tree Results

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Accuracy  Rate | precision | recall | F1-score | AUC |
| RENN | 0.999 | 0.999 | 0.999 | 0.772 | 0.898 |
| SMOTEENN | 0.998 | 0.998 | 0.997 | 0.523 | 0.897 |
| SMOTE | 0.997 | 0.999 | 0.999 | 0.482 | 0.881 |
| SMOTETomek | 0.997 | 0.998 | 0.997 | 0.472 | 0.871 |
| ADASYN | 0.998 | 0.999 | 0.998 | 0.557 | 0.887 |

After using tuning hyperparameters:

'n\_estimators': 80

'max\_depth': 14

'min\_samples\_split': 300

**Table 6:** Best result after tuning

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Accuracy  Rate | precision | recall | F1-score | AUC |
| RENN | 0.999 | 0.999 | 0.999 | 0.792 | 0.901 |

## 7.4 Random Forest Classifier

### Method Introduction

Random forest classifier is a machine learning method which is specifically designed for decision tree classifiers. Random forest is achieved by growing many classification trees. Random Forest Classifier is easy to use and will obtain high performance for most cases. This method is also good at avoiding overfitting issue.

### Preprocessing

Here we use stratified split to split the full data into training data and test data.Before fitting the training data to random forest Classifier,we will apply the resampling method to the training data due to the imbalance distribution of the response variable. Dealing with imbalance data, random forest classifier method will tend to the most frequently happened cases and ignore the extreme cases. Therefore, we apply five different resampling method to the training data and will compare the final prediction results in order to choose the best model. What’s more, grid search is also applied in order to tune the best hyper-parameters to fit the random forest classifier because the model is sensitive to the different hyper-parameters.

### Prediction Results

Table below shows the prediction results for random forest classifier method of five different resampled training data.

**Table 7**: Parameters for Random Forest

|  |  |  |
| --- | --- | --- |
|  | Number of Estimators | Maximum Depth |
| RENN | 60 | 11 |
| ENN (sot) | 60 | 11 |
| SMOTE (sos) | 50 | 11 |
| SMOTETomek (kos) | 50 | 11 |
| ADASYN (aos) | 50 | 11 |

**Table 8:** Random forest result after resampling

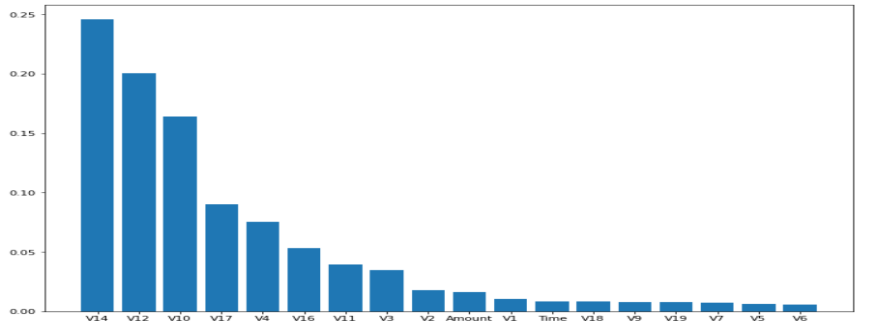
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Accuracy  Rate | precision | recall | F1-score | AUC |
| RENN | 0.9992 | 0.92 | 0.86 | 0.89 | 0.9844 |
| ENN(sot) | 0.9982 | 0.75 | 0.92 | 0.81 | 0.9873 |
| SMOTE(sos) | 0.998 | 0.75 | 0.91 | 0.81 | 0.9796 |
| SMOTETomek(kos) | 0.9984 | 0.76 | 0.92 | 0.82 | 0.9829 |
| ADASYN(aos) | 0.9932 | 0.59 | 0.90 | 0.64 | 0.9699 |

### Result Evaluation

From the result table, we can see random forest classifier method maintains high performance for all different resampled training data. Based on AUC score, the best model for random forest classifier method is with smote ENN resampling method and the AUC score is 0.9873 which is very high. And the accuracy for the prediction is 0.9982 which is also quite good.

### Feature Importance Visualization

We also plot the importance for features from the random forest model. The plot is shown below.



**Figure 8:** Feature importance of random forest

Based on the plot, we can tell the variable V14 is the most important feature and V6 is the least important feature. This result corresponds to the previous feature engineering part.

## 7.5 Support Vector Machine

Besides all the modeling methods stated above, Support Vector Machine is another widely used model to predict categorical response variable. It is a supervised learning model developed for classification and regression models, which can perform linear and non-linear classification using the kernel trick.

We’ve tried to perform SVM with three different kernel types for data after each of the five resampling methods. However, due to the time constraints and limit computing power, no results can be achieved with the whole data set. The code for this part is also attached and up to further improvement.

To improve the existing methods, other sampling methods like under-sampling can be taken into consideration. Also, checking the feature importance figure first and choosing only the important features is another way to scale down the data set to fit in the SVM models better.

In conclusion, since SVM requires extremely strong computing power and is considerably time consuming when putting in huge data set as in this project. It is not suitable for this data set in this scenario.

# Comparison and reflection

From the above prediction score results of all models. And based on AUC score, the optimal model we select is random forest classifier with ENN resampling method. The prediction result for the test data is shown below as a confusion matrix.

**Table 9:** Confusion Matrix

|  |  |  |
| --- | --- | --- |
|  | Prediction 0 | Prediction 1 |
| Actual 0 | 56779 | 85 |
| Actual 1 | 16 | 82 |

Based on the result, we can see the random forest classifier performs well at prediction the major case(0),with very high accuracy. And for the extreme case(0), the prediction accuracy is still good with more than 80%. And the AUC score for this model is 0.9873 which is high. Therefore, we recommend random forest classifier method as the best method for this project.

Future Improvement

* Consider data as time series: Since the data set for this project have a variable ‘Time’, therefore this data can be considered as a time series. Further approach such as LSTM can be applied. Since the data is about credit card fraud, converting the data into time series can better explain the situation of credit card fraud since cases might differ due to different time.
* Improve resampling methods to avoid some noise of the data set: Since resampling methods (oversampling) always produce some noise in the dataset and make the performance of classifier worse than normal case, we can try to modify the process of resampling to avoid some noise and hence make the classifiers more precise.

Reflection

* Limited computational capacity: Since the size of this data set is quite large, and after resampling , the size becomes even larger. Due to the limited computational capacity, running machine learning method for this data set becomes very difficult and it takes a long time to run the code and for grid search with more hyper-parameters and wider range, the computer will run for a long time and could not obtain the result. If better computational capacity is available, we can apply more approach and more specific improvement on the machine learning models.
* Too many tree-based methods: In this dataset, there is a time series characteristic. It may be hard for tree-based methods to present this time series. Using deep learning may help to present the time series characteristic and provide the result with a probably higher precision.