Data Challenge: Credit Card Fraud Transaction Prediction

Name: Xiaoyi Wang

Question 1: Load

1. Using 'read_json' in Pandas to read the raw zip file from Github; Convert the file to Pandas DataFrame

2. The structure of the data:
   - 786363 records
   - 29 fields

3. Basic summary statistics
   - The number of null value in each column

```
accountNumber                    0
customerId                       0
creditLimit                      0
availableMoney                   0
transactionDateTime              0
transactionAmount                0
merchantName                     0
acqCountry                    4562
merchantCountryCode            724
posEntryMode                  4054
posConditionCode               409
merchantCategoryCode             0
currentExpDate                   0
accountOpenDate                  0
dateOfLastAddressChange          0
cardCVV                          0
enteredCVV                       0
cardLast4Digits                  0
transactionType                698
echoBuffer                  786363
currentBalance                   0
merchantCity                786363
merchantState               786363
merchantZip                 786363
cardPresent                      0
posOnPremises               786363
recurringAuthInd            786363
expirationDateKeyInMatch         0
isFraud                          0
```

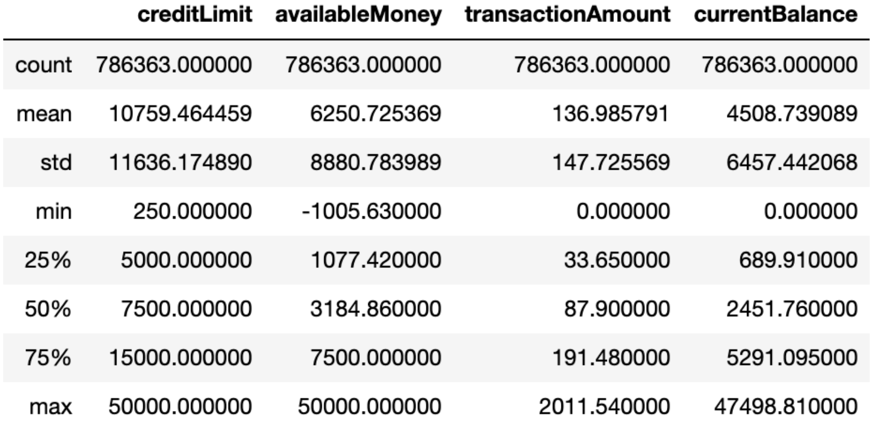
   We can find that the columns 'echoBuffer', 'merchantCity', 'merchantState', 'merchantZip', 'posOnPremises', 'recurringAuthInd' are all null

   - Maximum and minimum values in appropriate columns

```
                                       Maximum                 Minimum
creditLimit                              50000                     250
availableMoney                           50000                -1005.63
transactionDateTime        2016-12-30 23:59:45     2016-01-01 00:01:02
transactionAmount                      2011.54                       0
currentExpDate                         2033-08                 2019-12
accountOpenDate                     2015-12-31              1989-08-22
dateOfLastAddressChange             2016-12-30              1989-08-22
currentBalance                         47498.8                       0
```

*For the datetime type variables, the maximum value is the newest one, the minimum value is the oldest one

- Statistic description for numerical variables



	creditLimit	availableMoney	transactionAmount	currentBalance
count	786363.000000	786363.000000	786363.000000	786363.000000
mean	10759.464459	6250.725369	136.985791	4508.739089
std	11636.174890	8880.783989	147.725569	6457.442068
min	250.000000	-1005.630000	0.000000	0.000000
25%	5000.000000	1077.420000	33.650000	689.910000
50%	7500.000000	3184.860000	87.900000	2451.760000
75%	15000.000000	7500.000000	191.480000	5291.095000
max	50000.000000	50000.000000	2011.540000	47498.810000

All these four variables are left skewed. Most transaction has small credit limit, available money, transaction amount and current balance.

- The number of unique values in all columns

```
accountNumber                 5000
customerId                    5000
creditLimit                     10
availableMoney              521915
transactionDateTime         776637
transactionAmount            66038
merchantName                  2490
acqCountry                       5
merchantCountryCode              5
posEntryMode                     6
posConditionCode                 4
merchantCategoryCode            19
currentExpDate                 165
accountOpenDate               1820
dateOfLastAddressChange       2184
cardCVV                        889
enteredCVV                     976
cardLast4Digits               5245
transactionType                  4
echoBuffer                       1
currentBalance              487318
merchantCity                     1
merchantState                    1
merchantZip                      1
cardPresent                      2
posOnPremises                    1
recurringAuthInd                 1
expirationDateKeyInMatch         2
isFraud                          2
```

-- The columns which only have 1 unique the value is the columns of all null value

-- The columns which have 2 unique the value is the columns of Boolean values

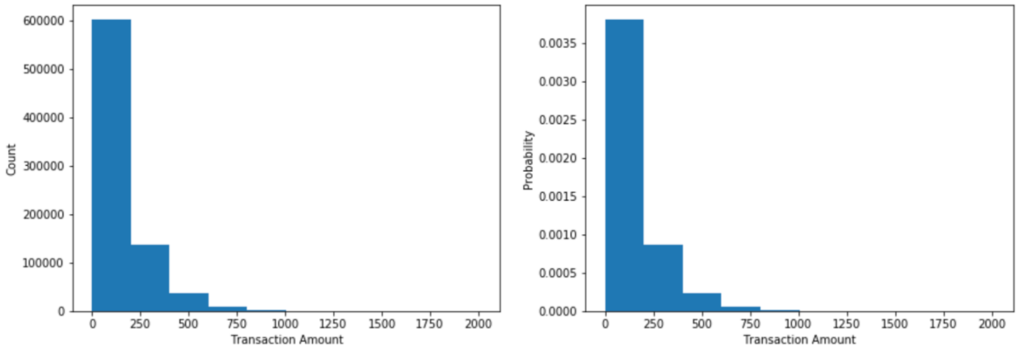
-- The following is some examples of unique value in appropriate columns

```
creditLimit [5000 2500 50000 15000 10000 250 500 1000 7500 20000]
acqCountry ['US' nan 'CAN' 'MEX' 'PR']
merchantCountryCode ['US' 'CAN' nan 'PR' 'MEX']
posEntryMode ['02' '09' '05' '80' '90' nan]
posConditionCode ['01' '08' '99' nan]
merchantCategoryCode ['rideshare' 'entertainment' 'mobileapps' 'fastfo
od' 'food_delivery' 'auto' 'online_retail' 'gym' 'health' 'personal ca
re' 'food' 'fuel' 'online_subscriptions' 'online_gifts' 'hotels' 'airl
ine' 'furniture' 'subscriptions' 'cable/phone']
transactionType ['PURCHASE' 'ADDRESS_VERIFICATION' 'REVERSAL' nan]
```

4. Basic cleaning
   - Drop the columns where values are all null
   - Drop one 'accountNumber' and 'customerId' because these two columns are just the same

Question 2: Plot

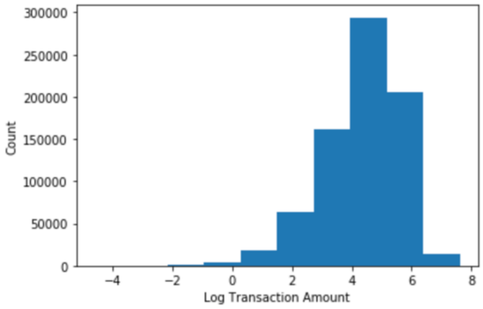
1. The histogram of transaction amount



The count and probability histogram of transaction amount

2. The feature of the structure
   -- The histogram skews to the left and has a long tail on the right
   -- Most of the transaction amount are very small, but there are a few amount very large

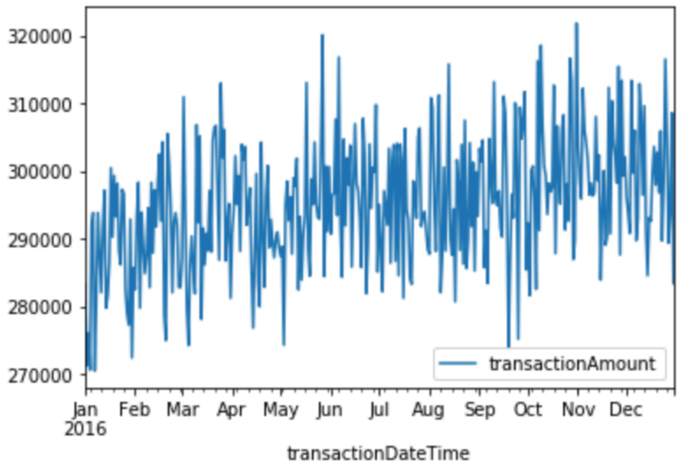
3. Hypothesis: the distribution of transaction amount follows log-normal distribution
   -- Conduct log transformation on the transaction amount: there are 22225 zero transaction amount, the number is very small compare to the 786363 in total
   -- Plot the histogram of log non-zero transaction amount



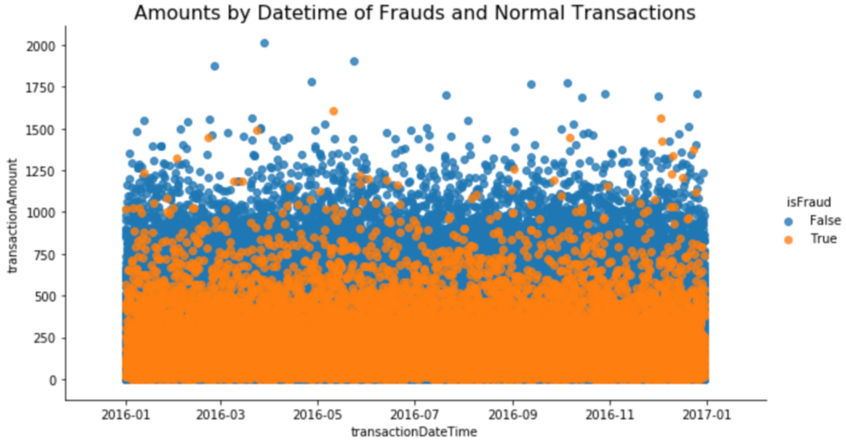
   -- The plot is more like normal distribution. Therefore, the hypothesis is reasonable

4. Some other plot about the data

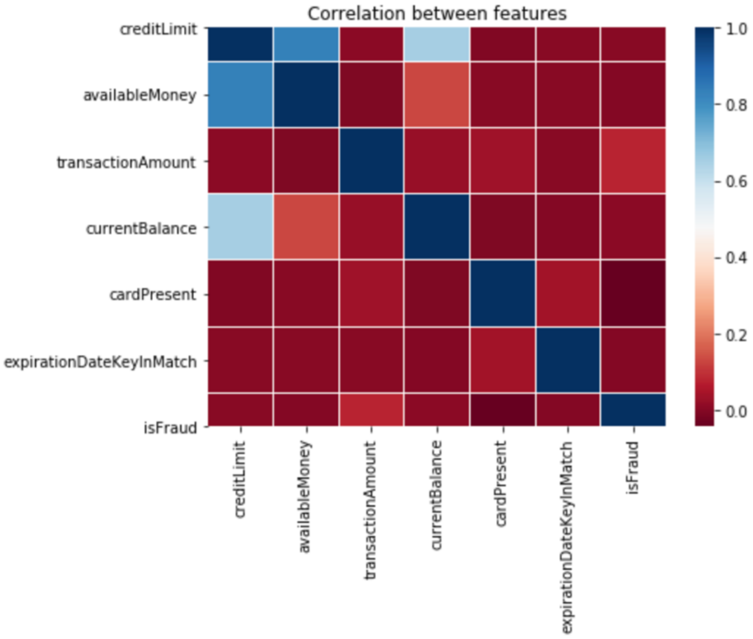
   - The sum of transaction amount everyday



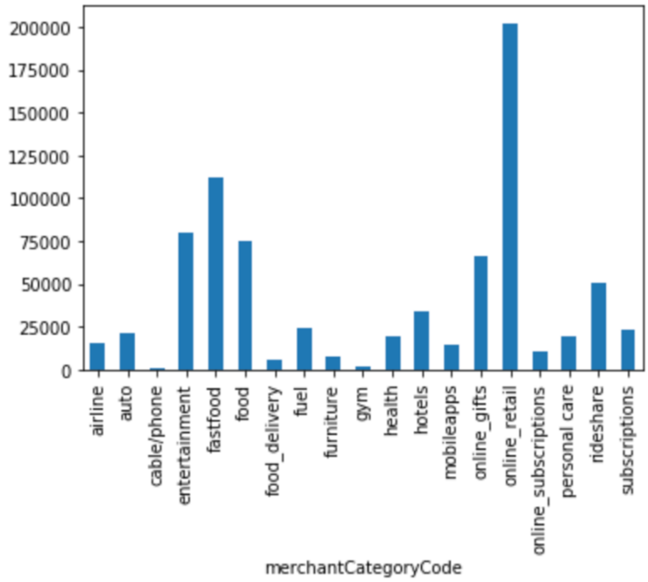
- The amount and time distribution of fraud transactions



- Correlation between features



- The number of transaction in each category



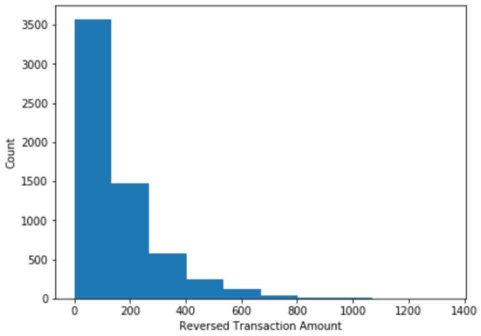
Question 3: Data Wrangling - Duplicate Transactions

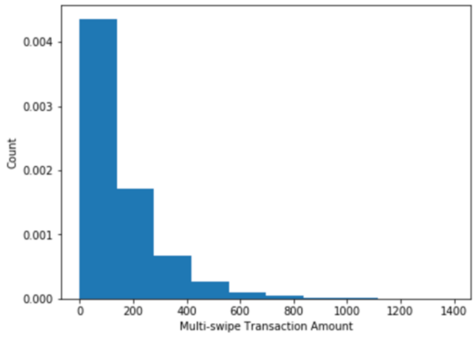
1. Identify reversed and multi-swipe transactions
   - Reversed type
     The rules for identification:
     -- There are a pair of transaction. One of the transaction type is REVERSAL and the other is PURCHASE
     -- The purchase one occurred before the reversal one
     -- They have same customer id, transaction amount, merchant name, card last 4 digit
   - Multi-swipe type
     The rules for identification:
     -- There are more than 1 transaction. The transaction type is all PURCHASE
     -- All transactions happen in 1 hour
     -- They have same customer id, merchant name, card last 4 digit

2. Number and amount of both types of duplicate transaction
   - Reversed type
     -- Total number: 6063
     -- Total amount: 9077059.35
   - Multi-swipe type
     -- Total number: 14176
     -- Total amount: 2108171.99

3. Interesting findings

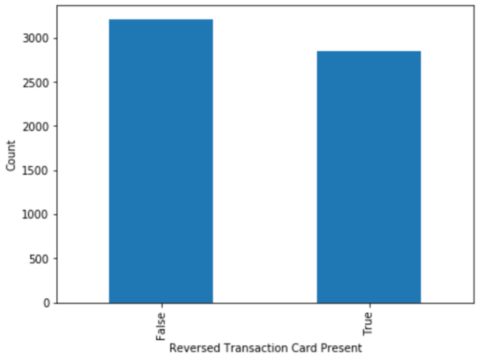
- Transaction Amount

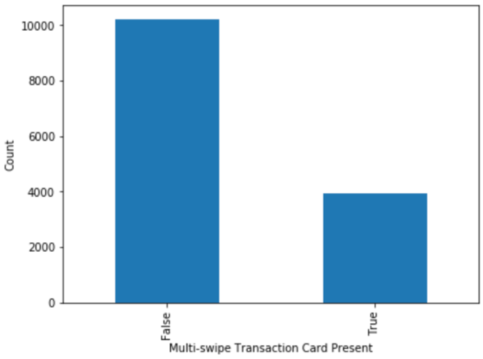




The mean, median, standard deviation, maximum, minimum of two kinds are almost same

- Card present





The percentage of not having card present is higher in multi-swipe transaction than reversed transaction

- Fraud

-- The percent of fraud transaction in all data is 1.5790417402649992

-- The percent of fraud transaction in reversed duplicate transaction is 1.764802902853373

-- The percent of fraud transaction in multi-swipe duplicate transaction is 2.0950902934537248

The fraud rate of multi-swipe duplicate transaction and reversed duplicate transaction are all higher than all transactions

Question 4: Model

1. Data cleaning

- Split x and y

  Y is the column 'isFraud'

- Deal with the columns have missing value

  -- These columns are 'acqCountry', 'merchantCountryCode', 'posEntryMode', 'posConditionCode', 'transactionType'

  -- These columns are all categorical variable and the unique values are small

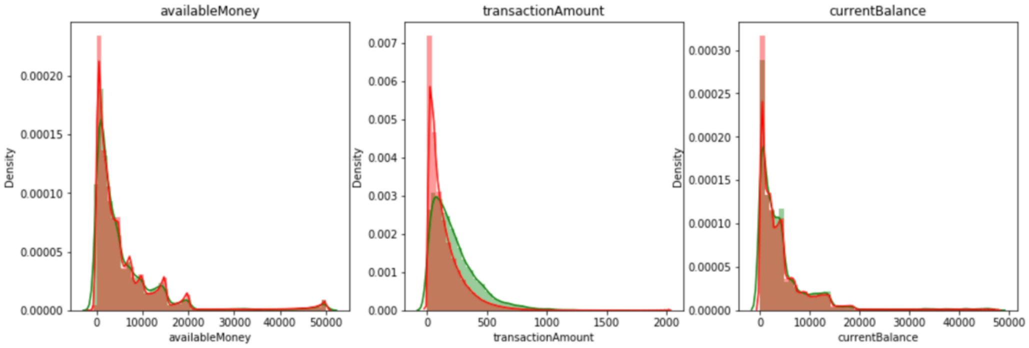
  -- We can convert the missing value to 'other' and convert these columns to dummy variables

- The columns of datetime

-- transactionDateTime: covert into 4 columns, 'transactionMonth', 'transactionDay', 'transactionHour', 'transactionMinute'
-- currentExpDate: covert to column ‘monthTillExp’ which stands for the number of months from 2016-12 till expire
-- accountOpenDate: covert to column ‘accountExistMonth’ which stands for the number of month since account open till 2016-12

- merchantName
  -- Add ‘purchasedMerchant’: covert merchantName to a Boolean value of whether the merchant is the first time appear on one's card
  -- Add ‘unpopularMerchant’: covert merchantName to a Boolean value of whether the merchant has been appear in the all the transactions less than 10 times
  -- Drop ‘merchantName’ column
- Country
  -- Convert merchantCountryCode to dummy variable
  -- Add a boolean column 'inPRorMEX' to see the merchant is in PR or MEX
  -- Add a boolean column 'sameCountry' to see the transaction and card acquired country is same or not
  -- Drop ‘acqCountry’ column
- posEntryMode, posConditionCode, merchantCategoryCode and transactionType
  -- Convert into dummy variables
- CVV
  -- Convert 'cardCVV' and 'enteredCVV' to a Boolean value of whether the 2 number is matched
- Active
  -- Add a column to show if the card is active
  -- Active = Balance / credit limit
  -- The number is closer to 0 then the card is more active; the number is closer to 1 then the card is more inactive
- Duplicate Transactions
  -- Using the result of Q3
  -- Add column 'reversedDuplicate' to see whether the transaction is reversed duplicate
  -- Add column 'multiswipeDuplicate' to see whether the transaction is multi-swipe duplicate
  -- Add column 'eitherDuplicate' to see whether the transaction is either one of the two kinds of duplicate
- Drop some irelevent or duplicated features

2. Search differents distribuitions of features



3. Divide train and test set
   Train set 70%, test set 30%

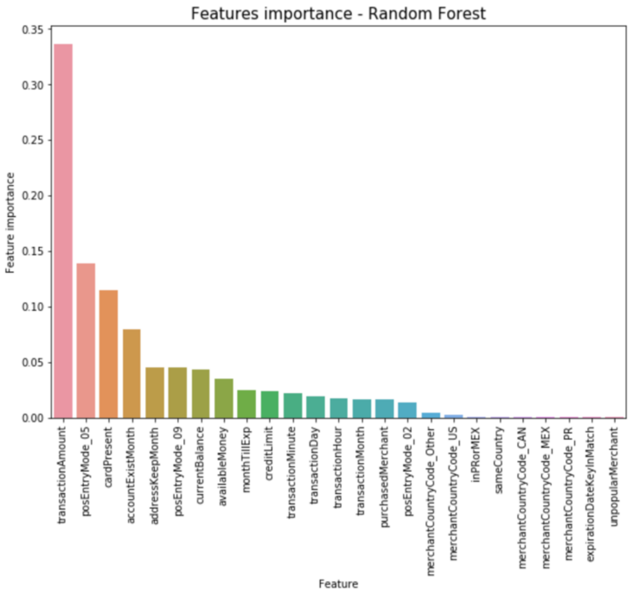
4. Sampling the dataset
   -- Because the dataset is highly imbalance (only 1.58% of the data is fraud), we need to sampling the dataset.
   -- Using undersampling will lead to only work on 1% of the original data, so we think oversampling is a better way
   -- Considering SMOTE or SMOTENC will built some 'fake' transaction record which will lead to some problems, so we choose random over sampling

5. Feature Selection based on feature importance
   -- There are 60 features in the cleaned dataset, using all the features will cause overfitting and increase the runtime significantly
   -- Before modeling, run a simple random forest model to select the most important 25 features. Only these features will be used in the following models.

6. Modeling
   - Choosing models
     The result we need to predict is a Boolean variable so I prefer classification models. Decision Tree, Random Forest and Gradient Boosting is the three model I choose to use.
   - Advantages and disadvantage of the three models

	Decision Tree	**Random Forest**	**Gradient Boosting**
Advantages	-- low runtime	-- handle overfitting -- reduce variance	-- reduce bias -- reduce variance
disadvantage	-- not robust -- can overfit	-- independent classifiers	-- can overfit -- runtime high

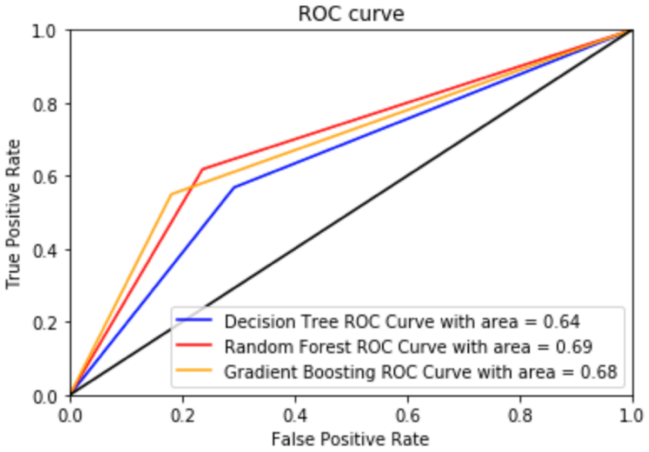
   - Feature importance in Random Forest



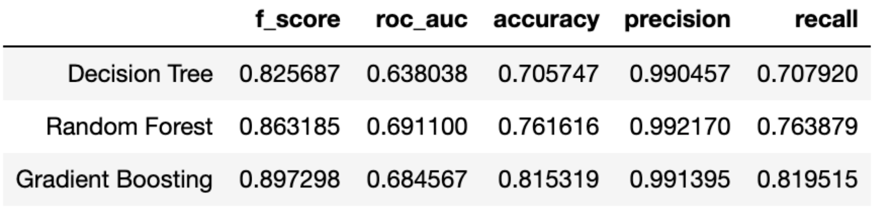
Most important features are transaction amount, pos entry mod 05, card present, account exist month

7. Model Comparison

- ROC Curve



- Evaluation Indicators



	f_score	roc_auc	accuracy	precision	recall
Decision Tree	0.825687	0.638038	0.705747	0.990457	0.707920
Random Forest	0.863185	0.691100	0.761616	0.992170	0.763879
Gradient Boosting	0.897298	0.684567	0.815319	0.991395	0.819515

- Conclusion

  Due to the highly imbalanced featured of the dataset, we think AUC is the better indicator to select the model.

  What's more, Random Forest and Gradient Boosting also are more robust.

  Therefore, Random Forest is the best model to predict fraud.

8. Future improvement
   - Consider other models, like logistic regression and neural networks
   - Using cross-validation to split the train set and test set, like split based on the cusromer id
   - Add some business rules besides the model, like whether the transaction happens the first time in a country, or whether the transaction's amount is much more higher than the historical amount in the same card. Combining the model and the business rules, we may get a better result