

Team Name

**The Three-Body Problem**

Team Members

**Bei Zhang**

**Ruochen Geng**

**Yao Lu**

December 12, 2018

# Table of Contents

[Executive Summary 3](#_Toc532403369)

[Description of Data 4](#_Toc532403370)

[Data Cleaning 6](#_Toc532403371)

[Candidate Variables Creation 8](#_Toc532403372)

[Feature Selection Process 12](#_Toc532403373)

[Model Algorithms 13](#_Toc532403374)

[Oversampling 13](#_Toc532403375)

[Logistic Regression 14](#_Toc532403376)

[Neural Network 17](#_Toc532403377)

[Support Vector Machine 18](#_Toc532403378)

[Random forest 18](#_Toc532403379)

[Results 19](#_Toc532403380)

[Conclusions 22](#_Toc532403381)

[Appendix 23](#_Toc532403382)

# Executive Summary

This report provides an overview of the process that a credit card company uses to determine fraudulent transactions that can be used to minimize fraud related expenses. Methods of analysis include feature engineer using KS and FDR score, and model building using logistic regression, neural networks, support vector machines and random forests. The most important features that have been considered by the models to determine whether a transaction is fraudulent is also listed below in the order of importance. These models are later tested against a future dataset that has not been used in the training or testing process also known as the out of time dataset to evaluate their performances. In the end, the random forest gave the best performance in terms of FDR, and final financial estimates have been derived from the predictions of that algorithm. The final part concludes that with an approximate 20% cutoff using a random forest algorithm, the maximum financial saving is around $94,000 during the out of time period. The report also discusses the potential overfit problem of the models used and the final cutoff.

# Description of Data

The card transaction data contains detailed information of card transaction information during 2010, including card number, transaction data, merchant number, transaction description, transaction location (state and zip), transaction type, transaction amount and fraud information. There are 96753 records in the dataset and each record has 9 fields with it.

**Summary Information of All Fields**

Numeric fields

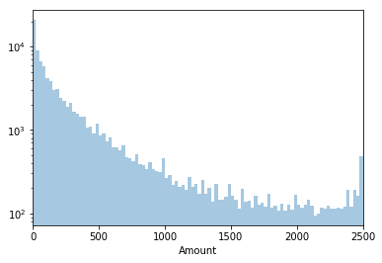
|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Field Name | Field Type | # Records with Value | % Populated | # Unique Values | Mean | Standard Deviation | Min | Max |
| Recnum | index | 96753 | N/A | 96753 | N/A | N/A | N/A | N/A |
| Amount | numeric | 96753 | 100 | 34909 | 427.89 | 10006.14 | 0.01 | 3102045.53 |

Categorical and other fields

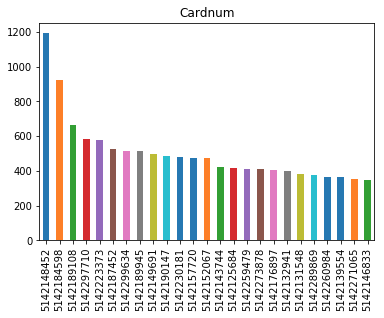
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Field Name | Field Type | # Records with Value | % Populated | # Unique Values | Most Common Value |
| Cardnum | categorical | 96753 | 100 | 1645 | 5142148452 |
| Date | date | 96753 | 100 | 365 | 2010-02-28 |
| Merchnum | categorical | 93378 | 96.51 | 13092 | 930090121224 |
| Merch\_description | text | 96753 | 100 | 13126 | GSA-FSS-ADV |
| Merch\_state | categorical | 95558 | 98.76 | 228 | TN |
| Merch\_zip | categorical | 96753 | 100 | 4568 | 38118 |
| Transtype | categorical | 96753 | 100 | 4 | P |
| Fraud | categorical | 96753 | 100 | 2 | 0 |

The most important distributions of fields are shown in the below. More details of each field are shown in the Appendix (Data Quality Report).

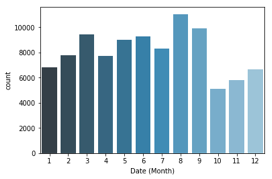
**Amount:** transaction amount.



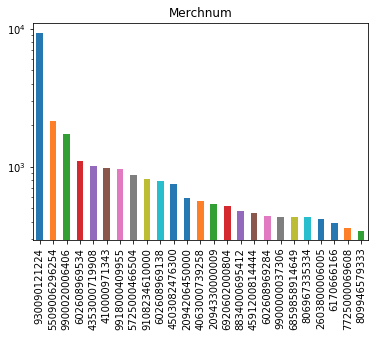
**Cardnum:** card number of transactions.



**Date:** payment date of the transaction.



**Merchnum:** merchandise number of a transaction.



# Data Cleaning

**Remove outliers**



As can be seen from the amount distribution plot, there is an apparent outlier. We check it in the raw data and found it’s not a fraud. This amount is far larger than other normal ones, so we decide to remove it. According to the Data Quality Report (Appendix), all the frauds appear in the transaction type P. Thus, we only include data of type P in our supervised model.

**Fill in missing values**

|  |  |  |  |
| --- | --- | --- | --- |
| **variable** | **# NaN** | **% populated** | **# unique** |
| Recnum | 0 | 100 | 96753 |
| Cardnum | 0 | 100 | 1645 |
| Date | 0 | 100 | 365 |
| Merchnum | 3375 | 96.51174 | 13091 |
| Merch description | 0 | 100 | 13126 |
| Merch state | 1195 | 98.7649 | 227 |
| Merch zip | 4656 | 95.18775 | 4567 |
| Transtype | 0 | 100 | 4 |
| Amount | 0 | 100 | 34909 |
| Fraud | 0 | 100 | 2 |

Notes: # NaN stands for number of missing values. % populated means the percentage that non-missing values take up for each variable. # unique means the number of unique values/categories in this variable.

The above table shows the summary of each variable. Merchnum, Merch state and Merch zip need to be filled in reasonable values.

* Merchnum

Firstly, a fair amount of 0 values are found in Merchum, which makes no sense, so we regard those as missing values. Secondly, after observations we find Merch description corresponds well with Merchnum, so we fill the missing values with the most frequent merchnum in this Merch description. With only 100 more records are filled, we turn to use Merch zip as an indicator and fill in with the most frequent Merch zip.

* Merch zip

We also find that Merch description corresponds very well with Merch zip, so we fill missing zip with the zip number in the same Merch description. It is not uncommon that the same Merchant number and Card number will appear in one zip at most of time. Thus, we fill missing zip with the most frequent zip value in the same Merchnum and Cardnum respectively.

* Merch state

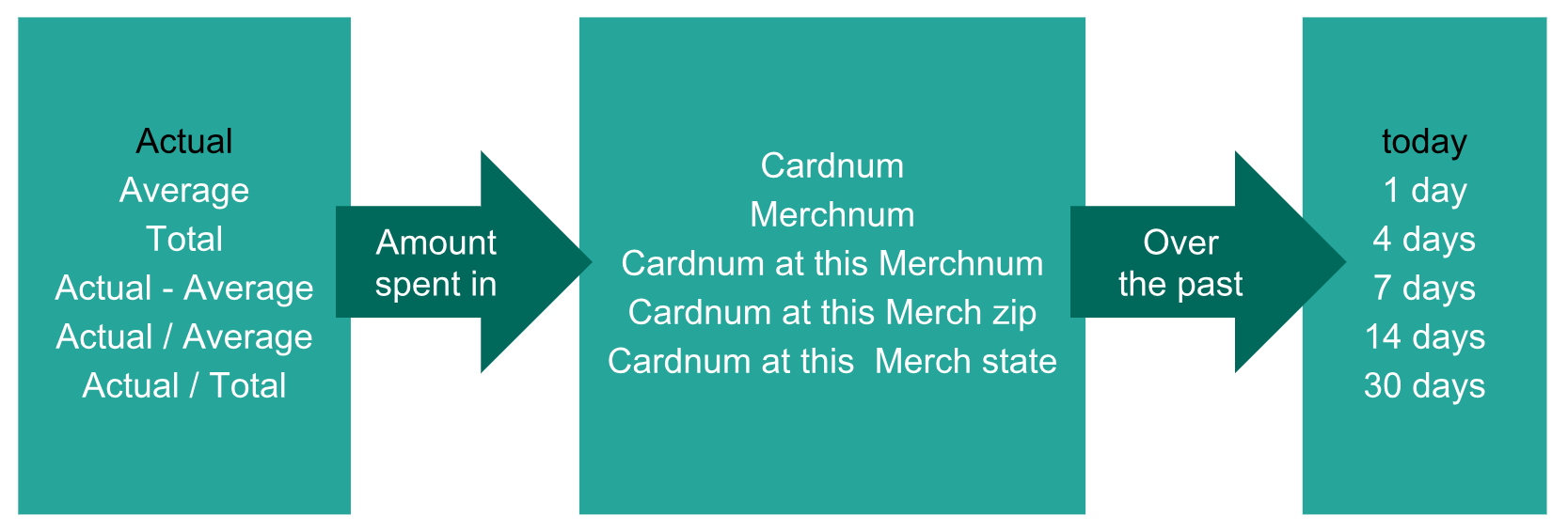
Apparently, a Merch state could be well inferred by using Merch zip, hence, that’s the first method to fill in NA. While most zip and state are missing at the same time, other methods should be used. The same to Merch zip, we use Merch description to help fill missing states in the second step. Lastly, we also find the most frequent state in its Merchnum group and Cardnum group and fill in.

With steps mentioned above to fill in missing values, there are still a small number of records remained missing. But the missing values only represent for less than 2% of all values in each variable and none of those missing values are labeled as frauds, so we decide to drop them.

# Candidate Variables Creation

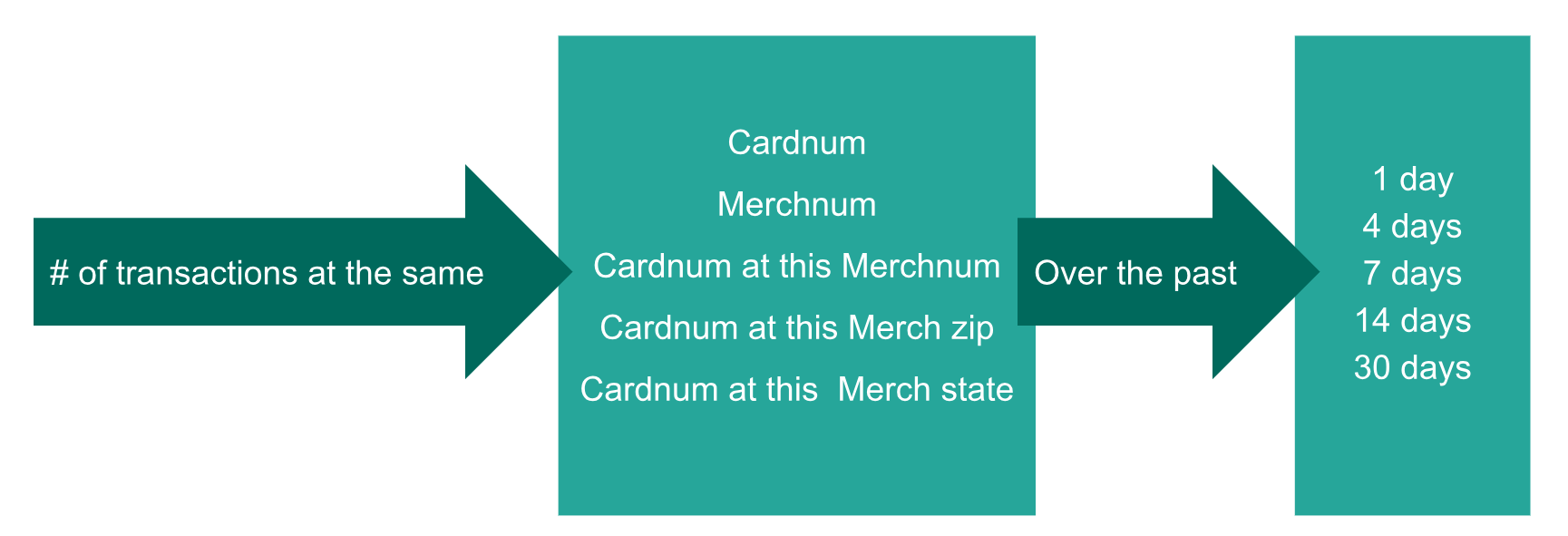
**Amount expert variables**

Based on our domain knowledge, those amounts larger than normal purchase would be a good signal in detecting frauds. Card number, merchant number and card number at this merchant, in this zip code or in this state, those 5 entity combinations could serve as important groups. Then, for each group, we calculate the average, total amount and how far the actual amount is away from the average and total spent over the past 1, 4, 7, 14 and 30 days.



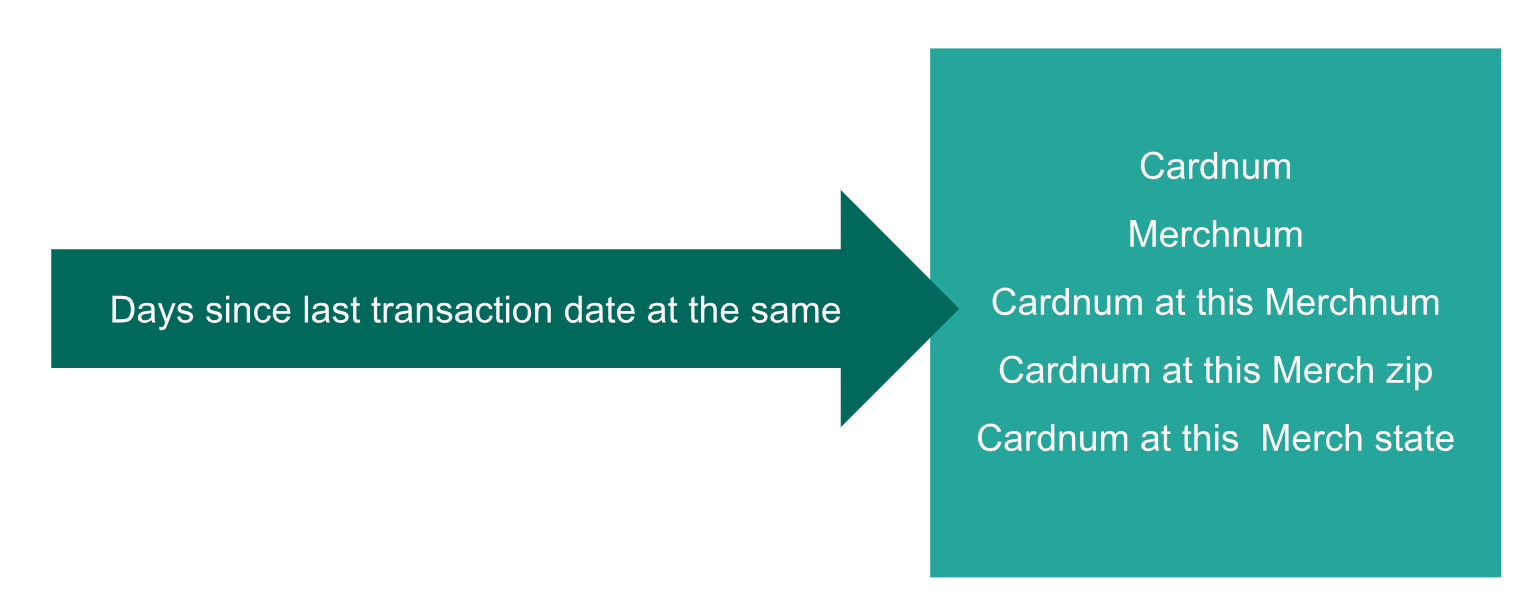
**Frequency expert variables**

Burst of activity at different merchants or card owners is usually a signal of fraud. Thus, we create frequency expert variables to detect whether the activity is abnormal.



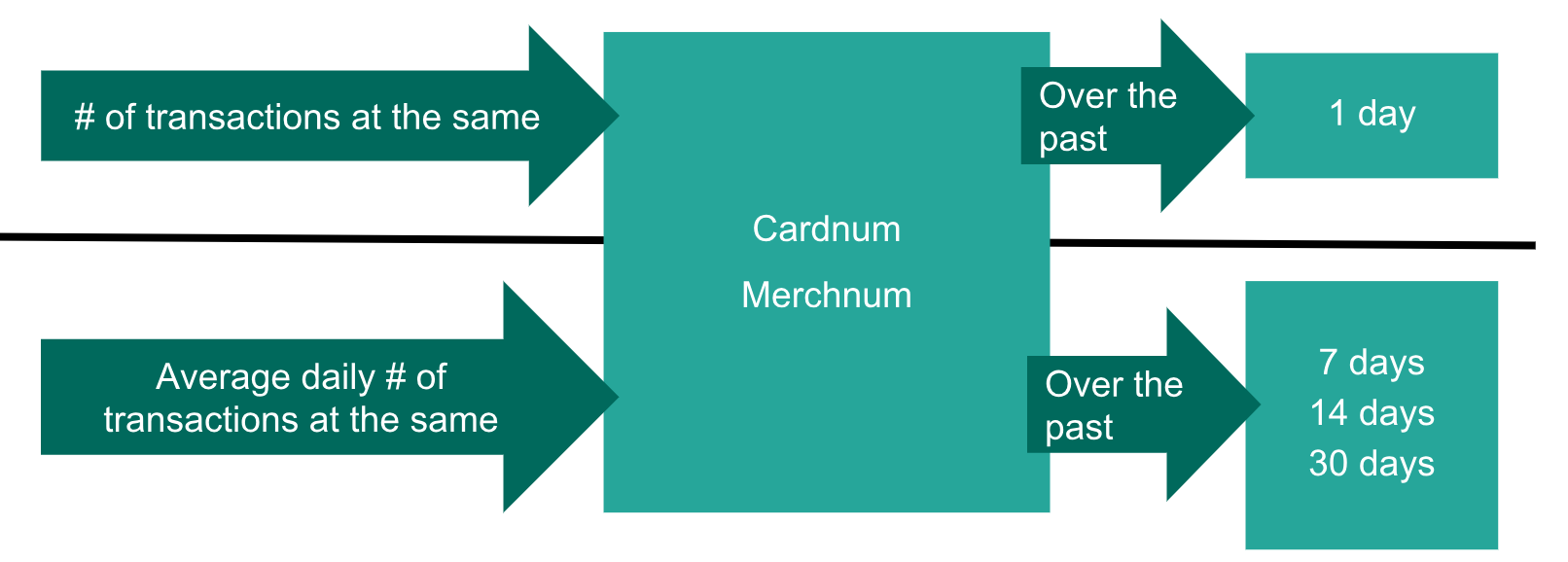
**Days since last purchase variables**

It is not uncommon that if a card is left behind for a long time, the chance to be reused is smaller. We could infer that day intervals since last purchase would be a good variable in detecting the fraud, so for each important group mentioned above, we calculate the time difference between current transaction date and last most recent transaction date.



**Velocity** **deviation expert variables**

These variables aim to compare the number of transactions over the past 1 day and average daily number of transactions over 7, 14 and 30 days in card number and merchant number groups.



**Variable summary**

The table below takes 1 day as an example and describes how each variable is calculated. The same calculation is applied to 7, 14 and 30 days except for 5 actual amounts. In all, we have 166 candidate variables, including 130 amount expert variables, 25 frequency expert variables, 5 days since last purchase variables, 6 velocity deviation expert variables.

|  |  |  |
| --- | --- | --- |
| **Variable** | **Description** | **Category** |
| Amount\_cn\_actual | The actual total amount spent on each day in this card number | Amount Expert Variables |
| Amount\_mn\_actual | The actual total amount spent on each day in this merchant number |
| Amount\_cn\_mn\_actual | The actual total amount spent on each day in this card number at this merchant number |
| Amount\_cn\_zip\_actual | The actual total amount spent on each day in this card number at this zip |
| Amount\_cn\_st\_actual | The actual total amount spent on each day in this card number at this state |
| Amount\_cn\_1d\_avg | The average of total amount spent over the past 1 day in this card number group |
| Amount\_cn\_1d\_sum | The sum of total amount spent over the past 1 day in this card number group |
| Amount\_cn\_1d\_act-avg | The actual number minus average number of total amount spent over the past 1 day in this card number group |
| Amount\_cn\_1d\_act/avg | The actual number divided by average number of total amount spent over the past 1 day in this card number group |
| Amount\_cn\_1d\_act/sum | The actual number divided by total number of total amount spent over the past 1 day in this card number group |
| Amount\_mn\_1d\_avg | The average of total amount spent over the past 1 day in this merchant number group |
| Amount\_mn\_1d\_sum | The sum of total amount spent over the past 1 day in this merchant number group |
| Amount\_mn\_1d\_act-avg | The actual number minus average number of total amount spent over the past 1 day in this merchant number group |
| Amount\_mn\_1d\_act/avg | The actual number divided by average number of total amount spent over the past 1 day in this merchant number group |
| Amount\_mn\_1d\_act/sum | The actual number divided by total number of total amount spent over the past 1 day in this merchant number group |
| Amount\_cn\_mn\_1d\_avg | The average of total amount spent over the past 1 day in this card number at different merchant numbers number group |
| Amount\_cn\_mn\_1d\_sum | The sum of total amount spent over the past 1 day in this card number at different merchant numbers number group |
| Amount\_cn\_mn\_1d\_act-avg | The actual number minus average number of total amount spent over the past 1 day in this card number at different merchant numbers number group |
| Amount\_cn\_mn\_1d\_act/avg | The actual number divided by average number of total amount spent over the past 1 day in this card number at different merchant numbers number group |
| Amount\_cn\_mn\_1d\_act/sum | The actual number divided by total number of total amount spent over the past 1 day in this card number at different merchant numbers number group |
| Amount\_cn\_zip\_1d\_avg | The average of total amount spent over the past 1 day in this card number at different zip number group |
| Amount\_cn\_zip\_1d\_sum | The sum of total amount spent over the past 1 day in this card number at different zip number group |
| Amount\_cn\_zip\_1d\_act-avg | The actual number minus average number of total amount spent over the past 1 day in this card number at different zip number group |
| Amount\_cn\_zip\_1d\_act/avg | The actual number divided by average number of total amount spent over the past 1 day in this card number at different zip number group |
| Amount\_cn\_zip\_1d\_act/sum | The actual number divided by total number of total amount spent over the past 1 day in this card number at different zip number group |
| Amount\_cn\_st\_1d\_avg | The average of total amount spent over the past 1 day in this card number at different states number group |
| Amount\_cn\_st\_1d\_sum | The sum of total amount spent over the past 1 day in this card number at different zip number group |
| Amount\_cn\_st\_1d\_act-avg | The actual number minus average number of total amount spent over the past 1 day in this card number at different zip number group |
| Amount\_cn\_st\_1d\_act/avg | The actual number divided by average number of total amount spent over the past 1 day in this card number at different zip number group |
| Amount\_cn\_st\_1d\_act/sum | The actual number divided by total number of total amount spent over the past 1 day in this card number at different zip number group |
| Freq\_cn\_1d | Total number of transactions over the past 1 day in this card number | Frequency expert variables |
| Freq\_mn\_1d | Total number of transactions over the past 1 day in this merchant number |
| Freq\_cn\_mn\_1d | Total number of transactions over the past 1 day in this card number at different merchant numbers |
| Freq\_cn\_zip\_1d | Total number of transactions over the past 1 day in this card number at different zips |
| Freq\_cn\_st\_1d | Total number of transactions over the past 1 day in this card number at different states |
| last\_cn | Days since last transaction in this card number | Days since last purchase variables |
| last\_mn | Days since last transaction in this merchant number |
| last\_cn\_mn | Days since last transaction in this card number at this merchant number |
| last\_cn\_zip | Days since last transaction in this card number at this merchant zip |
| last\_cn\_st | Days since last transaction in this card number at this merchant state |
| velo\_cn\_avg7d | the number of transactions over the past 1 day divided by average daily number of transactions over 7 days at this card number | Velocity deviation expert variables |
| velo\_mn\_avg7d | the number of transactions over the past 1 day divided by average daily number of transactions over 7 days at this merchant number |

# Feature Selection Process

Starting out from 166 variables created above, these variables have been narrowed down to 19 final features through the following procedures.

Standardize all numeric variables:

All numeric variables were standardized to the z-scale using the following equation .

Calculate KS and FDR:

The kolmogorov-smirnov (KS) score and fraud detection rate for the top 3% of data (FDR) were calculated for each feature and ranked according to decreasing KS and FDR score.

Create a weighted Rank

The features were ranked by a weighted rank where the weight for the KS score and FDR were 60% and 40% respectively. We decided to give the KS a heavier score, since it includes a better penalization for false negatives.

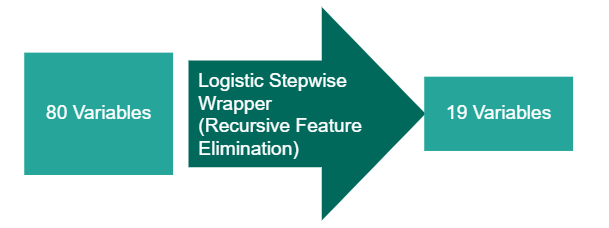
Sort and select top 80 variables

In order to make the next step faster, the bottom half of the variables were eliminated from the weighted rank.

****

Recursive feature elimination

Finally, a backwards logistic stepwise function was used to narrow it down to 19 final features.

****

# Model Algorithms

## Oversampling

**Why do we use oversampling?**

The proportion of fraud data in the total data set is only 1%, which will bias the prediction models towards the more common class. Most classification algorithms are sensitive to unbalance in the predictor classes. A machine learning model that has been trained and tested on such a dataset could now predict “non-fraud” for all samples and still gain a very high accuracy.

**When to use oversampling?**

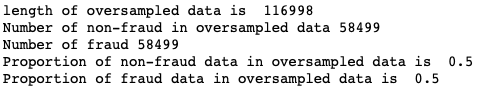
we oversample the minor class **after** splitting training and testing data sets, and we only do oversampling on training set. Because by oversampling **before** split, we may “bleed” information from the validation set into the training of the model. Then it may result in overfitting problem.

**How to use oversampling?**

With our training data created, we up-sample the non-fraud using the SMOTE algorithm (Synthetic Minority Oversampling Technique). At a high level, SMOTE:

1. Works by creating synthetic samples from the minor class (non-fraud) instead of creating copies.
2. Randomly choosing one of the k-nearest-neighbors and using it to create a similar, but randomly tweaked, new observations.

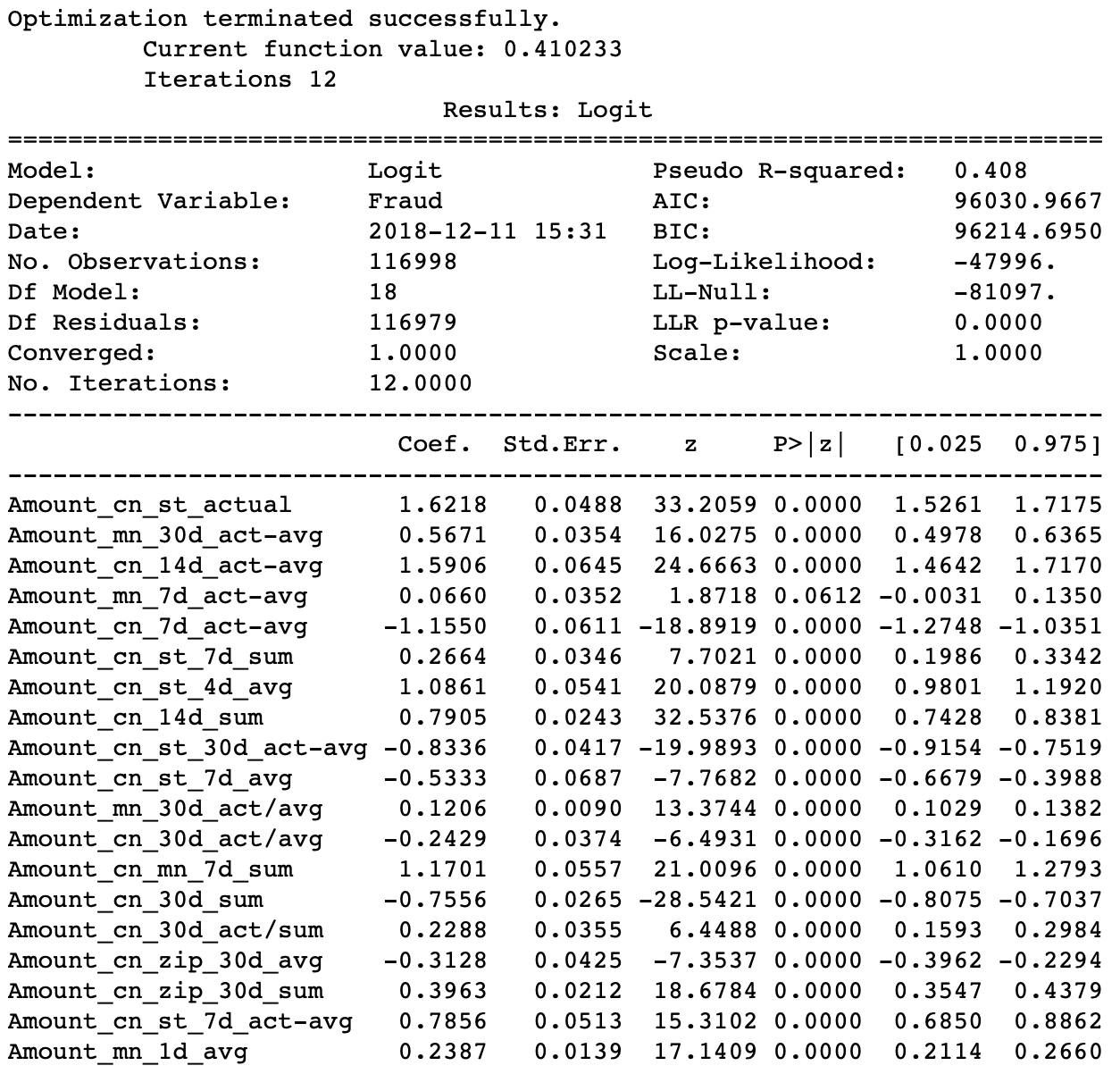
The description of oversampled data:



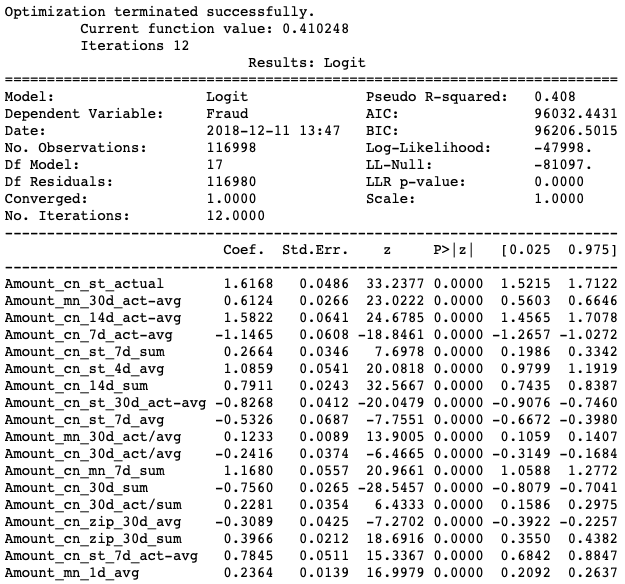
## Logistic Regression

The first algorithm we use is logistic regression. Since it is the most straightforward algorithm for classification problem and the variables and coefficients are easier to interpret.

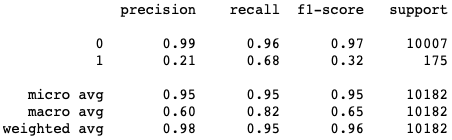
We first put in all variables after feature selection in the model.



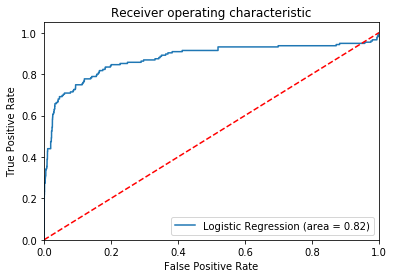
Next, we remove ‘Amount\_mn\_7d\_act-avg’ from the model since it is not significant (p-value > 0.05). The removal does not significantly influence the coefficients of other variables. Then we train the logistic regression again and get the following results:



The classification report of the logistic regression on oot data:



The ROC of the logistic regression on oot data (auc = 0.82):

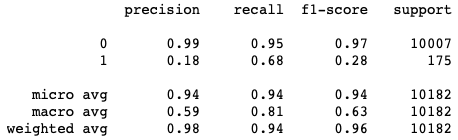


From the result of logistic regression, we notice that the total amount a card number spent in this state on that day is critical to detect fraud. If a transaction amount is higher than the average of the past 2 weeks, it is a good indicator of fraud. It is abnormal if a card spent large amount in a merchant in the previous week.

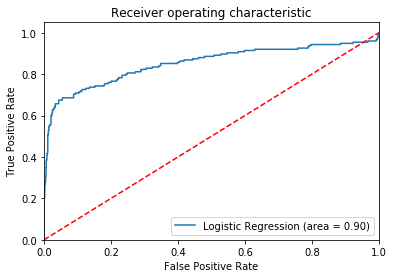
## Neural Network

Next, we try to build a neural network model to learn non-linear relationship. We select ‘relu’ as activation function, ‘auto’ batch size, (3, 3, 3) as hidden\_layer\_sizes, constant learning rate, and ‘adam’ as optimizer after tuning the model.

The classification report of neural network on oot data:



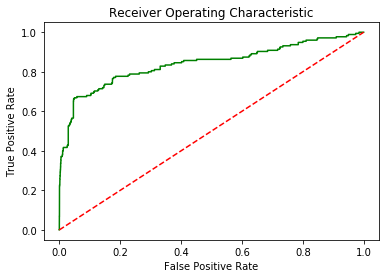
We also plot the ROC of neural network on oot data:



We notice that the AUC of neural network is higher than that of logistic regression.

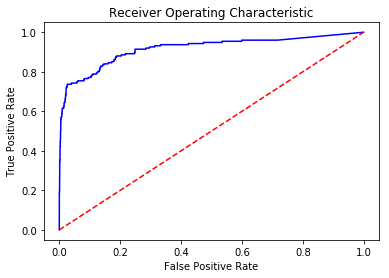
## Support Vector Machine

We used a support vector machine with a kernel of rbf to get the following results. This model was selected because it is slightly faster than a lot of other models. The hyperparameters were selected using a random search method from a given grid.



## Random forest

The random forest model has given the best performance in terms of OOT performance, and has also been selected as the final model. The following hyperparameters have been used from a random grid search. n\_estimators=1500, with min\_samples\_split= 2, min\_samples\_leaf=2, max\_depth=20.

****

# Results

After building four prediction models, we compare the performance of the models on oot data set. The metrics we use to compare the algorithms are fraud detection rate (at 3%), area under ROC curve and accuracy. We also create the fraud saving plot for our best model and suggest our client a score cut-off.

**Fraud detection rate**

|  |  |  |  |
| --- | --- | --- | --- |
| **FDR @ 3%** | | | |
|  | Training | Testing | OOT |
| Logistic Regression | 77.72% | 79.78% | 49.14% |
| Neural Network | 98.37% | 82.40% | 57.14% |
| SVM | 89.82% | 85.37% | 41.71% |
| Random Forest | 100.00% | 89.33% | 65.14% |

According to fraud detection rate (at 3%), random forest algorithm performs the best.

**Area under ROC curve**

|  |  |  |  |
| --- | --- | --- | --- |
| **AUC** | | | |
|  | Training | Testing | OOT |
| Logistic Regression | 88.76% | 90.22% | 81.78% |
| Neural Network | 98.30% | 93.05% | 81.26% |
| SVM | 93.92% | 91.65% | 69.83% |
| Random Forest | 94.74% | 84.77% | 59.40% |

According to AUC, neural network performs the best.

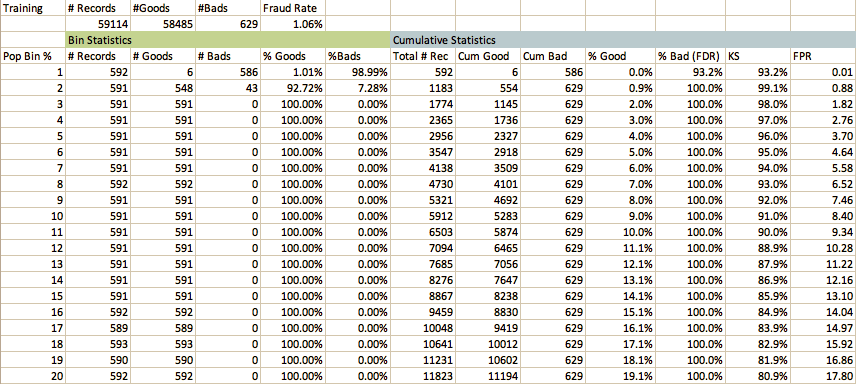
**Accuracy**

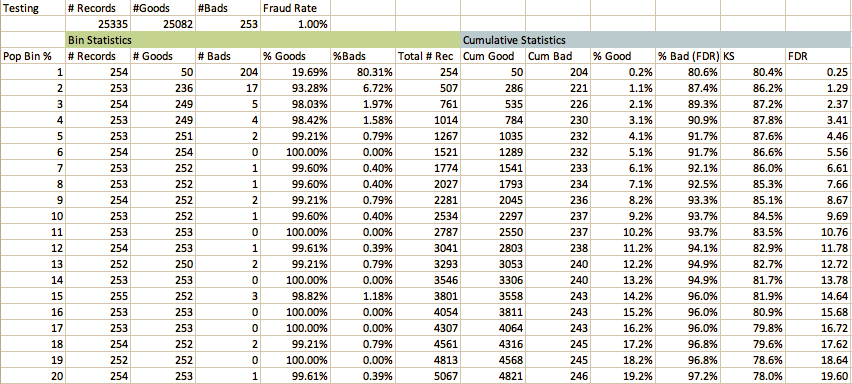
|  |  |  |  |
| --- | --- | --- | --- |
| **Accuracy** | | | |
|  | Training | Testing | OOT |
| Logistic Regression | 95.26% | 94.94% | 95.09% |
| Neural Network | 98.39% | 94.68% | 94.06% |
| SVM | 97.93% | 97.80% | 96.98% |
| Random Forest | 99.87% | 99.67% | 98.55% |

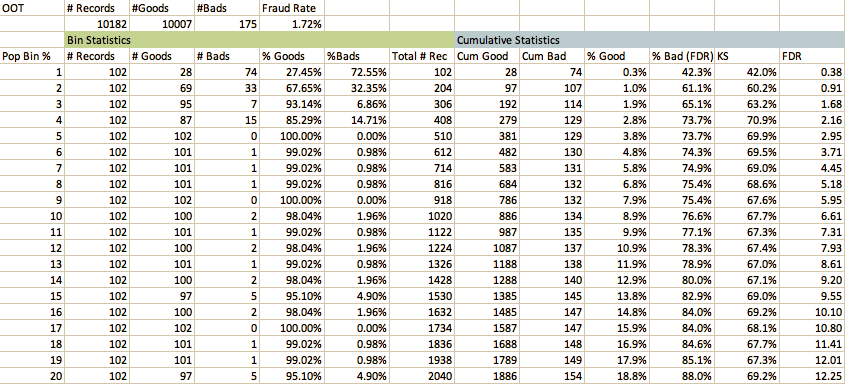
According to accuracy, random forest performs the best.

After the comparison, we choose random forest as our best and final model since FDR is the most important metrics in this case.

**Statistics**

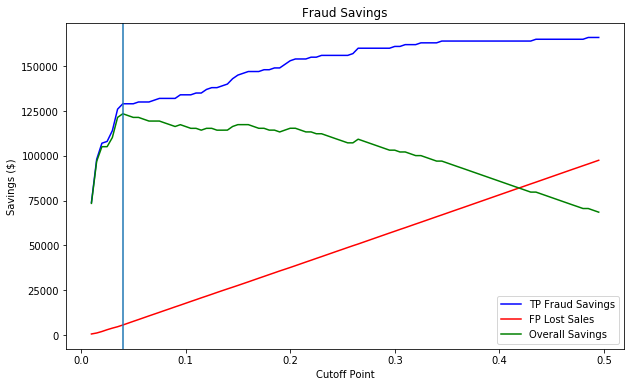
****





**Fraud saving plot and score cut-off**

The following chart shows the optimal cutoff, which is also the global maxima for the overall savings line. The cut-off point is set at approximately 3.9 % which translates to a $123,000 in overall savings.



# Conclusions

To conclude, transaction amount is a critical indicator to detect fraud. The total amount a card number spent in this state on that day is critical to detect fraud. If a transaction amount is higher than the average of the past 2 weeks, it is a good indicator of fraud. It is abnormal if a card spent large amount in a merchant in the previous week.

# Appendix

**Data Quality Report on Card Transaction Data**

**Description of Data**

The card transaction data contains detailed information of card transaction information during 2010, including card number, transaction data, merchant number, transaction description, transaction location (state and zip), transaction type, transaction amount and fraud information. There are 96753 records in the dataset and each record has 9 fields with it.

**Summary Information of All Fields**

Numeric fields

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Field Name | Field Type | # Records with Value | % Populated | # Unique Values | Mean | Standard Deviation | Min | Max |
| Recnum | index | 96753 | N/A | 96753 | N/A | N/A | N/A | N/A |
| Amount | numeric | 96753 | 100 | 34909 | 427.89 | 10006.14 | 0.01 | 3102045.53 |

Categorical and other fields

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Field Name | Field Type | # Records with Value | % Populated | # Unique Values | Most Common Value |
| Cardnum | categorical | 96753 | 100 | 1645 | 5142148452 |
| Date | date | 96753 | 100 | 365 | 2010-02-28 |
| Merchnum | categorical | 93378 | 96.51 | 13092 | 930090121224 |
| Merch\_description | text | 96753 | 100 | 13126 | GSA-FSS-ADV |
| Merch\_state | categorical | 95558 | 98.76 | 228 | TN |
| Merch\_zip | categorical | 96753 | 100 | 4568 | 38118 |
| Transtype | categorical | 96753 | 100 | 4 | P |
| Fraud | categorical | 96753 | 100 | 2 | 0 |

**Description of Each Field**

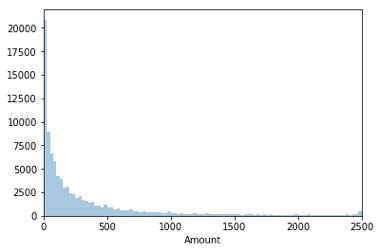
**Recnum**

This field is the index of the dataset. All values are unique.

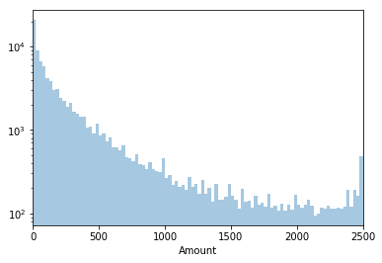
**Amount**

Amount indicates the transaction amount. It is a numeric field. I set the range of 0 to 2500 to plot the distribution since the number of properties with frontage more than 2500 feet is only 745, which is only 0.77% of the whole dataset. I also log the Y scale so that the distribution is better to observe.

Original distribution:

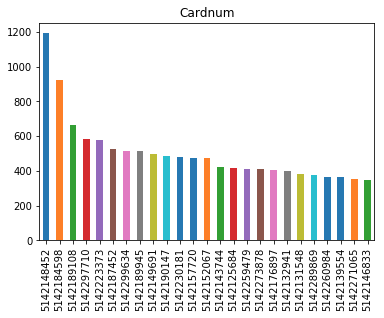


Distribution after log transformation:



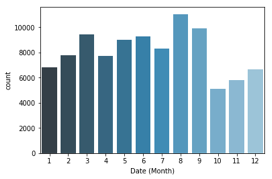
**Cardnum**

Cardnum is the card number of transactions. It is a categorical field. I select the first 25 card numbers with the greatest number of transactions to plot the bar chart.



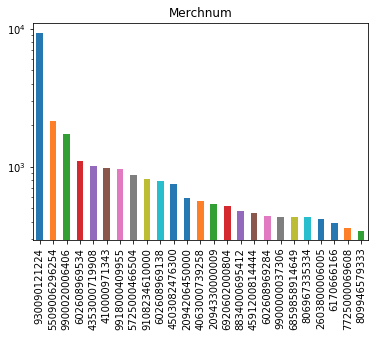
**Date**

Date indicates the payment date of the transaction. It is a date field and a categorical field. We plot the number of transactions by time using the date field. The plot shows the distribution of number of transactions each month. As can be seen, August has the most transactions, while October has the least in 2010.



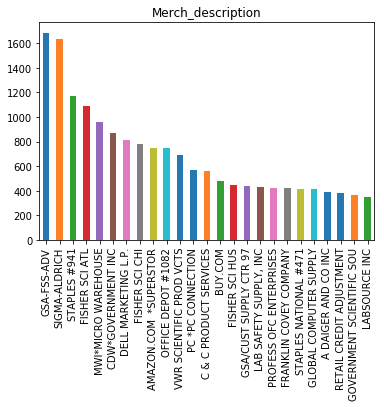
**Merchnum**

Merchnum indicates the merchandise number of a transaction. It is a categorical field. I select the first 25 Merchnum with the greatest number of transactions to plot the bar chart. I log the Y scale so that the distribution is better to observe.

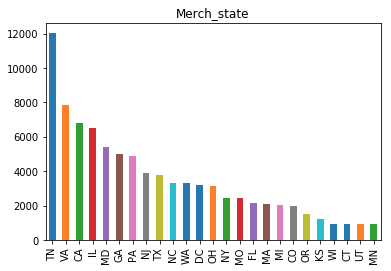


**Merch\_description**

Merch\_description is the description of a transaction. It is a text and a categorical field. I select the first 25 Merch\_description with the greatest number of transactions to plot the bar chart.

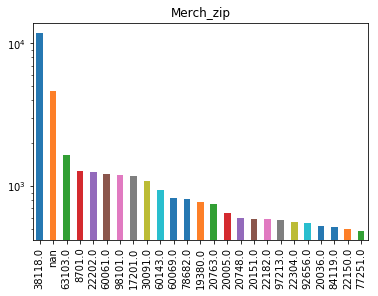


**Merch\_state**

Merch\_state is the state where a transaction occurs. It is a categorical field. I select the first 25 Merch\_state with the greatest number of transactions to plot the bar chart. 

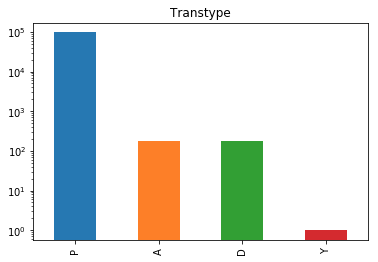
**Merch\_zip**

Merch\_zip is the zip code where a transaction occurs. It is a categorical field. I select the first 25 Merch\_zip with the greatest number of transactions to plot the bar chart. I log the Y scale so that the distribution is better to observe.



**Transtype**

Transtype indicates the transaction type. It is a categorical field. I log the Y scale so that the distribution is better to observe.



**Fraud**

Fraud indicates whether a transaction is fraudulent. It is a categorical field. I log the Y scale so that the distribution is better to observe.

