**REPORT**

**BUILDING SUPERVISED MODEL FOR CARD TRANSACTIONS**

**Prepared for:**

Professor Stephen Coggeshall

**Prepared by:**

CHEN, Siqi

GUO, Qianfei

HUNG, Kai-Ling

JIA, Xuerong

LIN, Xiaoyan

TSAI, Pei Yu

March 28, 2019

**Table of Content**

[1 Executive Summary 3](#_Toc4626427)

[1.1 Project Goal 3](#_Toc4626428)

[1.2 Work Performed 3](#_Toc4626429)

[1.3 Conclusions 3](#_Toc4626430)

[2 Description of Data 3](#_Toc4626431)

[2.1 Data Source 3](#_Toc4626432)

[2.2 Summary of Numerical Fields 3](#_Toc4626433)

[2.3 Summary of Categorical Fields 4](#_Toc4626434)

[2.4 Response Field 4](#_Toc4626435)

[2.5 Overview of Card Transactions Data 4](#_Toc4626436)

[3 Data Cleaning 6](#_Toc4626437)

[3.1 Removing Outlier & Filter Transtype 6](#_Toc4626438)

[3.2 Filling Missing Fields 7](#_Toc4626439)

[4 Candidate Variables 7](#_Toc4626440)

[4.1 Amount Variables 7](#_Toc4626441)

[4.2 Frequency Variables 8](#_Toc4626442)

[4.3 Days Variables 8](#_Toc4626443)

[4.4 Velocity Change Variables 8](#_Toc4626444)

[5 Feature Selection Process 9](#_Toc4626445)

[5.1 Filter method by KS and FDR 9](#_Toc4626446)

[5.2 Wrapper method by RFECV 9](#_Toc4626447)

[6 Model Algorithms 10](#_Toc4626448)

[6.1 Logistic Regression 10](#_Toc4626449)

[6.2 Neural Nets 11](#_Toc4626450)

[6.3 Random Forests 11](#_Toc4626451)

[6.4 Boosting Trees 12](#_Toc4626452)

[7 Results 12](#_Toc4626453)

[8 Conclusions 14](#_Toc4626454)

[8.1 Steps Performed 14](#_Toc4626455)

[8.2 Recommendations 15](#_Toc4626456)

[9 Appendix 16](#_Toc4626457)

[9.1 Appendix 1 - Candidate Variables 16](#_Toc4626458)

[9.2 Appendix 2 – Data Quality Report 19](#_Toc4626459)

# Executive Summary

## 1.1 Project Goal

The goal of the project is to build and test different machine learning algorithms based on card transaction data between January 1, 2010 and December 31, 2010 and find out the optimal model to predict future fraud.

## 1.2 Steps Performed

To build and select an optimal supervised fraud model based on the card transaction data, we first cleaned the data, and then built 269 variables incorporating amount information, card information and transaction location information. We used feature selection methods to reduce the number of variables to 20. We split the card transaction data and used the data in November and December 2010 as out out-of-time data. 75% of the transactions from January to October 2010 were randomly selected to be used as training data and the remaining 25% used as testing data. After that, we built four supervised fraud models using four different types of machine learning algorithms: Logistic Regression, Neural Nets, Random Forests, and Boosting Trees, and found that Random Forests outperforms the rest three.

## 

## 1.3 Conclusions

We concluded that Random Forest is superior to the rest three models we built. At 3% population cut-off, the result is FDR@3% of 99.0% for the training dataset and FDR@3% of 85.3% for the testing dataset. For the OOT dataset, we got FDR@3% of 39.1%.

Due to limited time and resources, we were only able to build four different machine learning models. Given more time, we would try other machine learning techniques such as Decision Trees, Bagging, Support Vector Machine, etc. We might find a model that is a better fit than Random Forest to predict future fraud.

# Description of Data

## 2.1 Data Source

The Card Transactions Data was provided by Professor Stephen Coggeshall on February 21, 2019. This dataset contains 10 fields and 96,753 records. Each record in this dataset represents a card payment between January 1, 2010 and December 31, 2010. The dataset includes information such as card numbers, dates of transactions, merchant numbers, merchant descriptions, merchant state, merchant zip, transaction type, payment amount in US dollars and fraud scores.

## 2.2 Summary of Numerical Fields

In this dataset there is only one numerical field. Refer to Table 1 below for basic statistics for this field.

**Table 1**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| field | count | %  Populated | mean | std | min | 25% | 50% | 75% | max |
| Amount | 96,753 | 100.00% | 427.9 | 10,006 | 0.01 | 33.48 | 137.98 | 428.2 | 3,102,046 |

## 2.3 Summary of Categorical Fields

There are eight categorical fields in this data set. Refer to Table 2 below for basic statistics for this field.

**Table 2**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Field | Count | % Populated | Unique Value | Most Common |
| Recnum | 96,753 | 100.00% | 96,753 | Uniform Distributed |
| Cardnum | 96,753 | 100.00% | 1,645 | 5142148452 |
| Date | 96,753 | 100.00% | 365 | 2/28/10 |
| Merchnum | 93,378 | 96.51% | 13,092 | 9.3009E+11 |
| Merch description | 96,753 | 100.00% | 13,126 | GSA-FSS-ADV |
| Merch state | 95,558 | 98.76% | 228 | TN |
| Merch zip | 92,097 | 95.19% | 4,568 | 38118 |
| Transtype | 96,753 | 100.00% | 4 | P |

## 2.4 Response Field

The column labelled “Fraud” represents a response field in this dataset. There are two types, “0” and “1”. Type “0” categorized as non-fraud transaction and Type “1” represents fraud transaction. Summary information is shown as Table 3.

**Table 3**

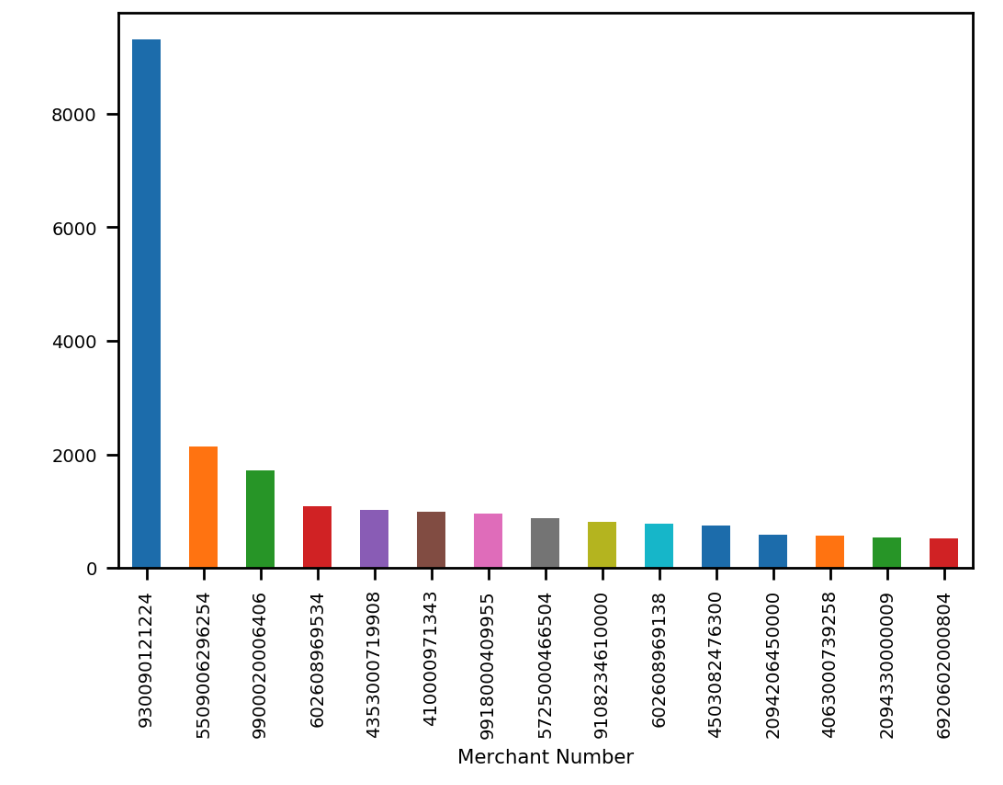
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Field | records | % Populated | Non-Fraud | Fraud |
| Fraud | 96,753 | 100.00% | 95694 (98.9%) | 1059 (1.1%) |

## 2.5 Overview of Card Transactions Data

The line charts and bar charts below give a high-level understanding of some important features of Card Transaction Data. Please refer to the Appendix 2 for a complete Data Quality Review Report.

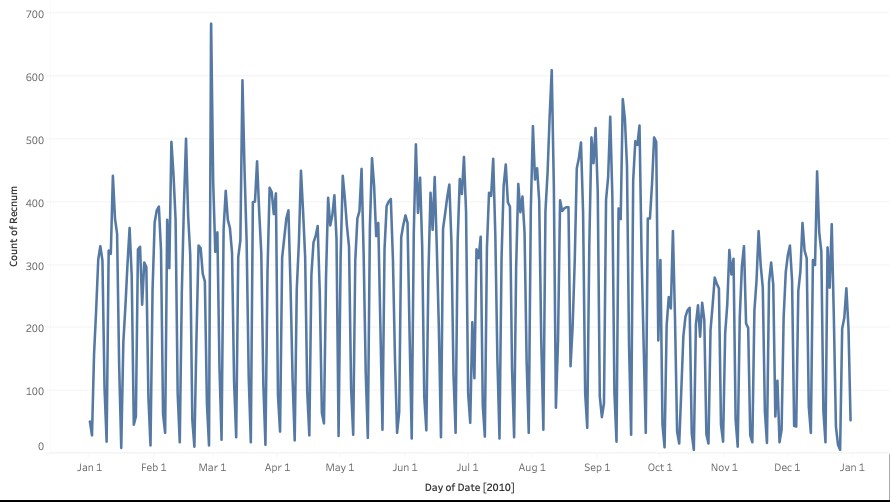
* + - 1. Merchnum

Merchnum represents merchant number. The graph below shows the count of the top 15 most common merchant numbers.

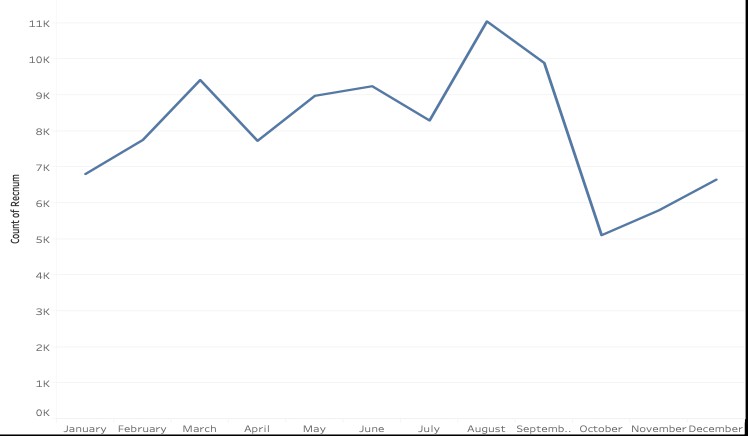


* + - 1. Date

The Date field contains 365 unique values, which are 365 different dates from January 1, 2010 to December 31, 2010. The line chart below represents number of payment transactions by date. We can see the weekly seasonality from this chart.

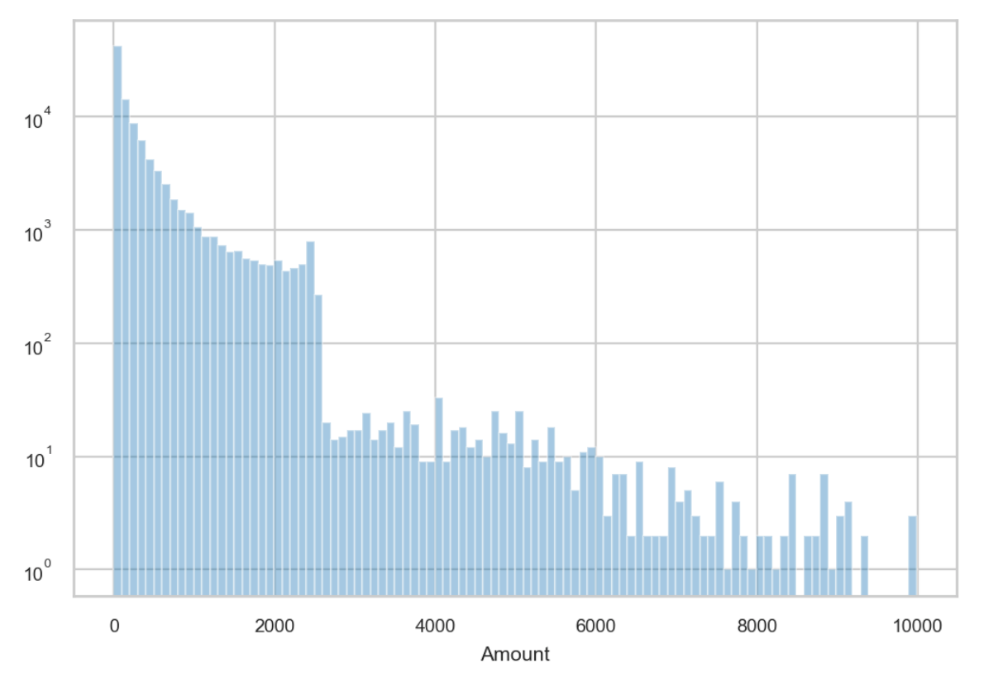


The line chart below represents number of transactions by month. A sharp decrease is noted in October.



* + - 1. Amount

The graph below shows the log scale distribution of transaction amount with amount limited to 10,000.



# Data Cleaning

## 3.1 Removing Outlier & Filter Transtype

As a first step of data cleaning, we excluded one extremely large value in the Amount Field: 3,102,045.53. We further removed the column named Transtype because there is only one value “P” in this column representing purchase.

## 3.2 Filling Missing Fields

Next, we filled missing values in the three fields, Merch state, Merch zip and Merchnum, with innocuous values according to following rules.

* + - 1. Merch state
* Grouped by Merch zip, filled the missing values with most frequent Merch state of the group; and
* Filled the remaining missing values with TN, the most frequent Merch state in the dataset.

1. Merch zip

* Grouped by Merchnum, filled the missing values with most frequent Merch zip in the group;
* Grouped by Merch description, filled the missing values with the most frequent Merch zip in the group;
* Grouped by Merch state, filled the missing values with the most frequent Merch zip in the group; and
* Filled the remaining missing values with the most frequent Merch zip in the whole dataset.

1. Merchnum

* Grouped by Merch description, filled the missing values with the most frequent Merchnum in the group;
* Grouped by Merch zip, filled the missing values with most frequent Merchnum of the group; and
* Grouped Merch state, filled the remaining missing values with the most frequent Merchnum in the group.

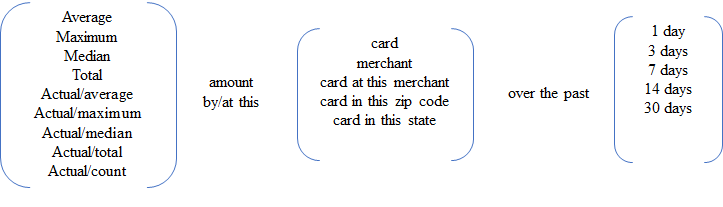
# Candidate Variables

The goal is to build a supervised fraud model on the card transaction data. To that end, we built 269 variables and used feature selection methods illustrated in the Section 5 to reduce the number of variables.

We built variables based on transaction amount, transaction frequency, days since last transaction and velocity change of transaction.

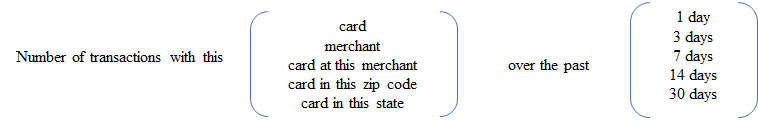
## 4.1 Amount Variables

One indication of credit card fraud is unusual transaction amount. For example, unusually large payment amounts for the same or different merchants. To build quality amount variables, we wanted to build variables incorporating amount information, card information and transaction location information over a certain period. The following chart illustrates how we created the amount variables. For example, one of the variables is created by grouping all records by card number and calculating the average transaction amount during the past 1 day. Another example would be grouping all records by card number and merchant number and calculating the total transaction amount of this group over the past month. As shown below, we created 9 x 5 x 5 = 225 variables.



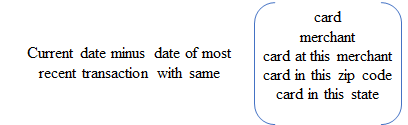
## 4.2 Frequency Variables

Another indication of credit card fraud is unusual transaction frequency. For example, bursts of activities at different merchants or transactions at merchants where the card had not been used before. To build quality frequency variables, we want to build variables that incorporate frequency information, card information and transaction location information over a certain period. The following chart illustrates how we created the frequency variables. For example, one of the variables is created by grouping all records by card number, counting the number of transactions during the past 1 day. Another example would be grouping all records by card number and merchant number, counting the number of transactions over the past month. As shown below, we created 5 x 5 = 25 variables.



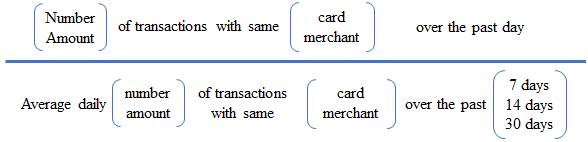
## 4.3 Days Variables

Unusually long or short duration of time since the last transaction could also be a signal of credit card fraud. To build quality Days variables, we want to build variables that incorporate the number of days since the last transaction, card information and transaction location information. The following chart illustrates how we created the Days variables. We created the variable by subtracting the date of most recent transaction from the current date for the same card number. As shown below, we created 5 variables.



## 4.4 Velocity Change Variables

We also wanted to compare the frequency and amount of transactions with the same card or at the same merchant over the past day to the frequency and amount over the past seven, 14 and 30 days, respectively. The following chart illustrates how we created the Velocity Change variables. For example, we grouped all records by card number and calculated the total transaction amount during the past day. Then we divided the result by the daily average amount for the same card number over the past seven days. As shown below, we created 2 x 2 x 3 = 12 variables.



A list of all candidate variables is included in Appendix 1 – Candidate Variables.

# Feature Selection Process

After creating 269 variables, we conducted filter method through KS and FDR, and wrapper method for feature selection to reduce dimensionality, correlation. We reduced the number of variables to 20.

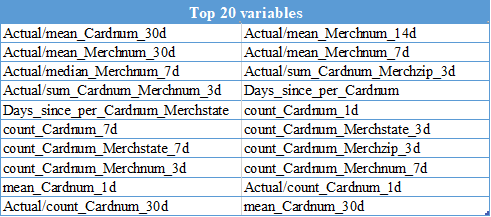
## 5.1 Filter method by KS and FDR

1. KS is a robust measure of how well good and bad distribution are separated. We used KS as a guide to evaluate how well a certain variable can separate Fraud and non-Fraud records. The higher the KS, the better the variable serve as an important feature.
2. Fraud Detection Rate (FDR) represents what percent of frauds are caught at a certain examination cutoff location. Here we used 3% FDR, meaning that after rank order records based on a particular variable, what percent of Fraud we can detect by looking at the top 3% of the population. The higher the FDR, the better the variable serve as an important feature.
3. After we got KS and 3% FDR for each variable, we first ranked all variables according to their KS ranking among 269 variables and got KS ranking number. Then we ranked all variables again according to their FDR and got FDR ranking number. For each record, we calculated the average of KS ranking number and FDR ranking number to have a combined score.
4. We sorted all variables based on the combined score and select only top 50% of the variables (135 variables).

## 5.2 Wrapper method by RFECV

After conducting Filter Method to select 135 candidate variables, we performed Feature Elimination Cross-Validation (RFECV) to select the best 20 features. RFECV is a feature selection method that removes the weakest features using cross-validation recursively to find the best features. Through the selection process, it also eliminates correlation that might exist between variables. We conducted the following steps:

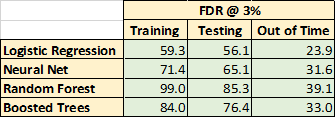
1. Performed RFECV to find important features using 5-fold cross-validation. Since the number of records for credit card transaction data is relatively small, instead of directly using the best 14 features generated by this algorithm;
2. Ranked all variables, and kept the top 45 variables;
3. Performed RFECV again on these 45 variables. Once again, instead of directly using the best 12 features generated by the algorithm; and
4. Ranked the 45 variables, kept the top 20 as our final 20 features.



# Model Algorithms

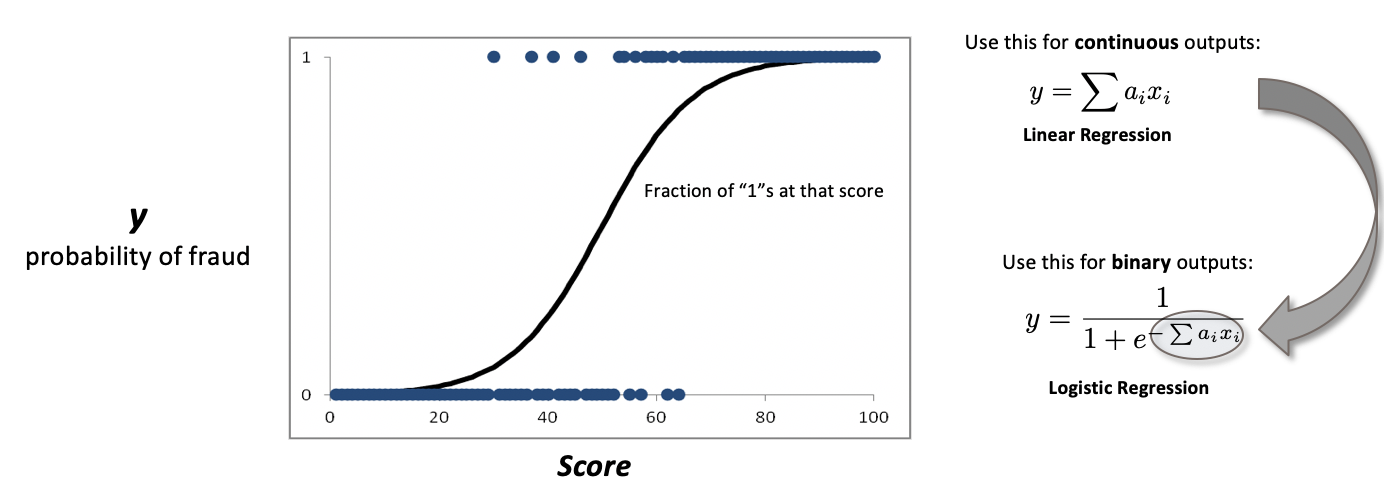
We split the dataset into three parts before training the models: training data, testing data, and out-of-time data. In our case, we took the last two months, between November 1, 2010 and December 31, 2010, as our out-of-time data. For the transactions between January 1, 2010 and October 31, 2010, we randomly selected 75% of them to be our training data and the remaining 25% was used as testing data. In our models, we used training set to fit the model, and then tune the model according to the testing set. Finally, we tested our model with out-of-time data.

We tried four different machine learning algorithms to build supervised models. In each model, we used the probability for data belonging to class 1 as fraud score, and then we sorted all the records by scores in descending order. After getting the top 3% transactions for each model, we summarized the fraud detection rate shown as below:



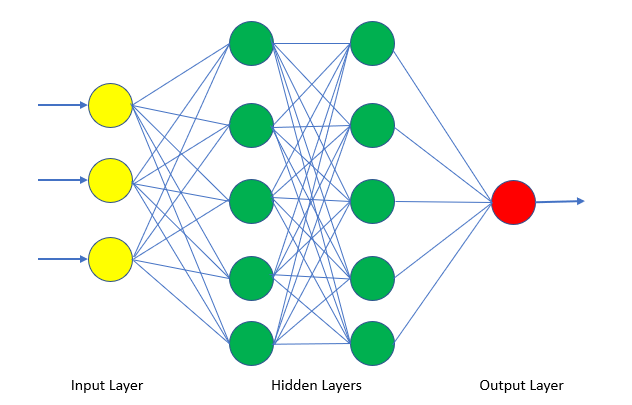
## 6.1 Logistic Regression

Logistic regression is an algorithm suitable to apply for data with binary output. Based on the input information, it transforms model output using the logistic sigmoid function to return a probability value which can then be mapped to the binary classes. For this card transaction project, we ranked order all the records based on the probability for that record to be fraudulent and evaluate FDR for top 3% of the listed records. We ran logistic regression ten times and take the average FDR@3%. The result is FDR@3% of 59.3% for the training dataset and FDR@3% of 56.1% for the testing dataset. For the OOT dataset, we got FDR@3% of 23.9%.



## 6.2 Neural Nets

A neural network has an input layer, one or more hidden layers, and an output layer. For this model, each node in the input layer consists the information of all selected variables for each record. The nodes in the output layer are either 0 or 1; 0 represents non-fraudulent transaction, and 1 represents fraud. For the hidden layer, we chose 1 hidden layer with 5 nodes at the beginning. Then we slightly increased the size of hidden layers and hidden nodes to 2 layers with 5 nodes in each layer.



We ran ten times and took the average FDR@3%. The result is FDR@3% of 71.4% for the training dataset and FDR@3% of 65.1% for the testing dataset. For the OOT dataset, we got FDR@3% of 31.6%.

## 6.3 Random Forests

Random forests provide an improvement over bagged trees by way of a random small tweak that decorrelates the trees. As in bagging, we build a number forest of decision trees on bootstrapped training samples. After that, different from bagging, each time a split in a tree is considered, a random sample of predictors is chosen as split candidates from the full set of predictors. Finally, we combine all the results by averaging or voting.

We ran ten times using different parameters and achieved the best performance on FDR@3% using the following parameters:

* n.trees = 20

n.trees is the total number of trees to fit. This is equivalent to the number of iterations and the number of basic functions in the additive expansion.

* Min.sample.leaf= 5

It is the minimum number of samples in newly created leaves. A split is discarded if after the split, one of the leaves would contain less then min.samples.leaf samples.

* interaction.depth = 15

Interaction. depth parameter as a number of splits it has to perform on a tree (starting from a single node).

We got FDR@3% of 99.0% for the training dataset and FDR@3% of 85.3% for the testing dataset. For the OOT dataset, we got FDR@3% of 39.1%.

## 6.4 Boosting Trees

Gradient boosting is a machine learning technique for regression and classification problems, which produces a prediction model in the form of an ensemble of weak prediction models, typically decision trees. It builds the model in a stage-wise fashion like other boosting methods do, and it generalizes them by allowing optimization of an arbitrary differentiable loss function.

We ran ten times using different parameters and achieved the best performance on FDR@3% using the following parameters:

* n.trees = 190.

n.trees is the total number of trees to fit. This is equivalent to the number of iterations and the number of basic functions in the additive expansion.

* shrinkage = 0.001.

It is a shrinkage parameter applied to each tree in the expansion. Also known as the learning rate or step-size reduction.

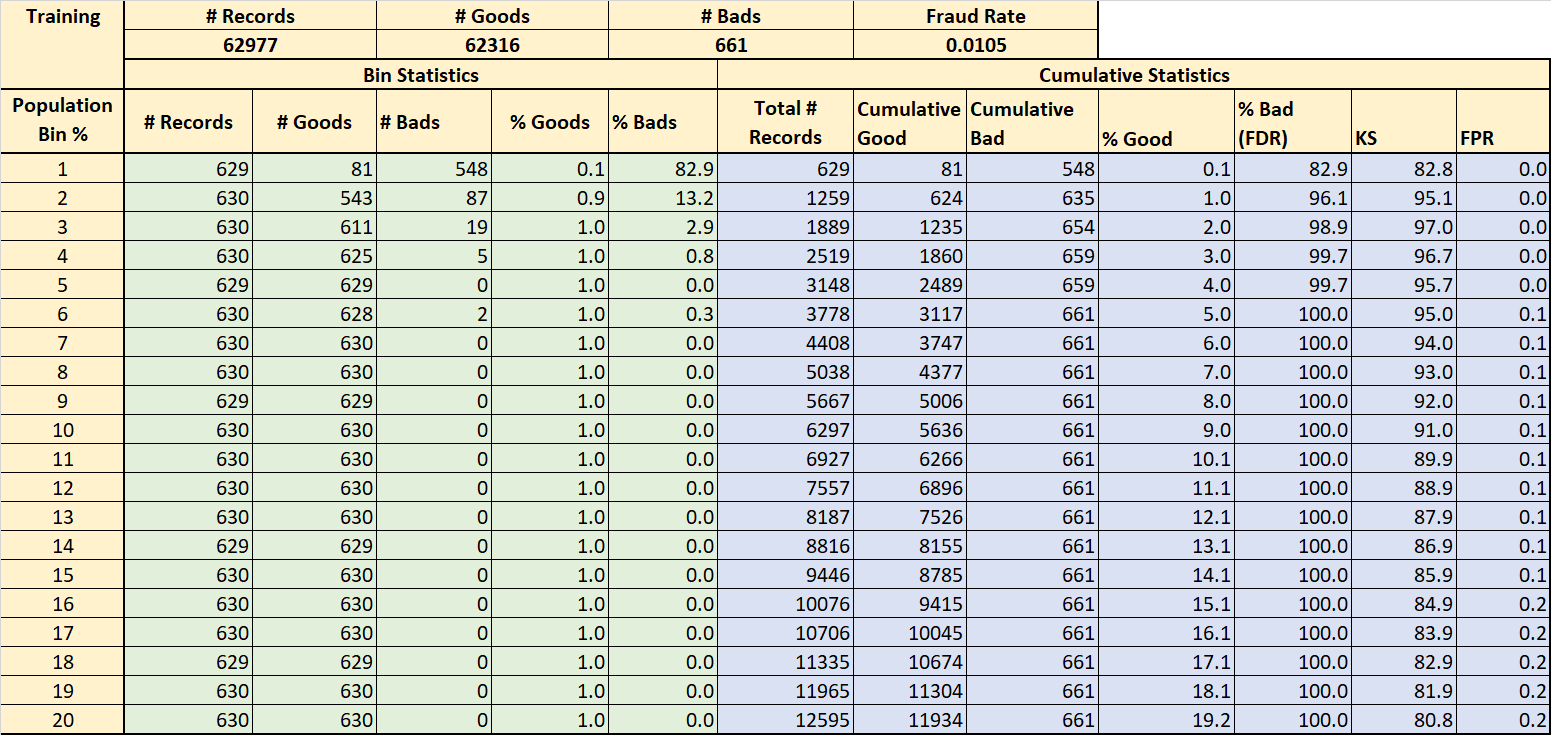
* interaction.depth = 5

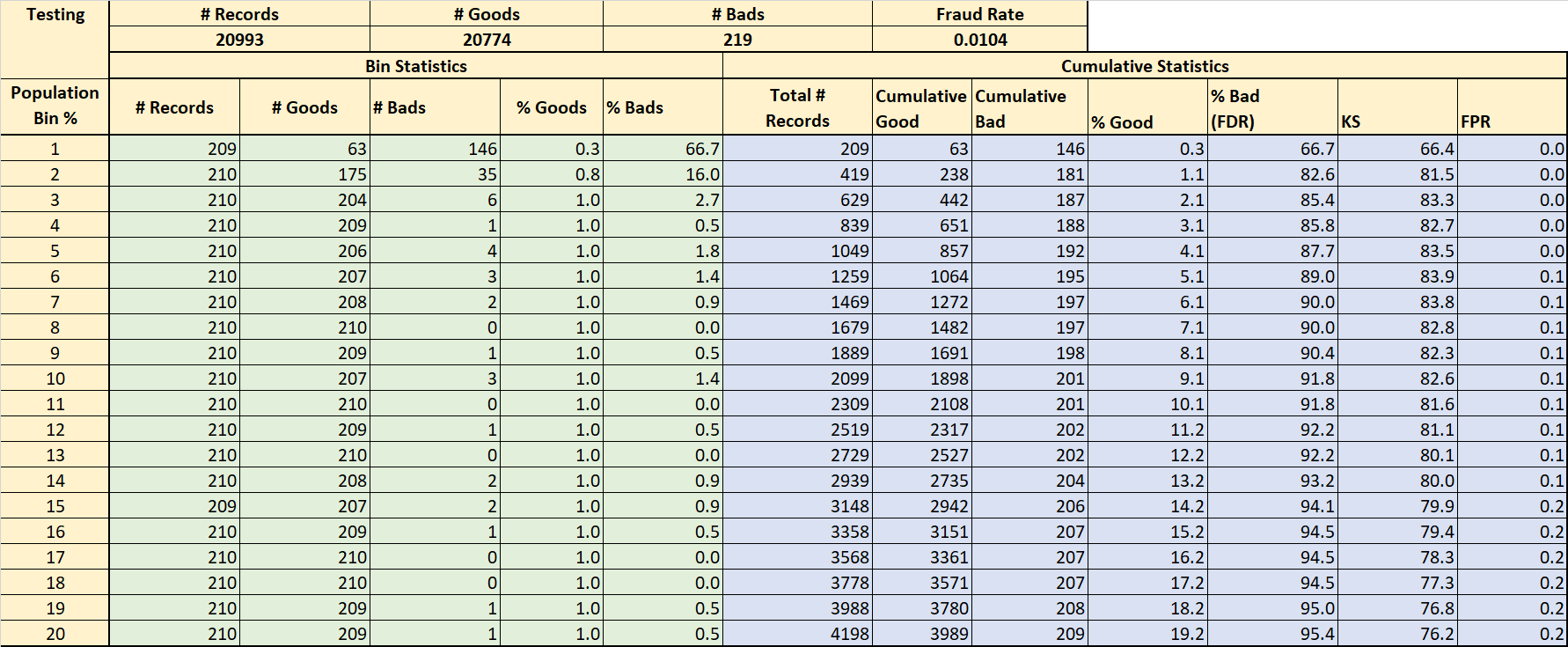
Interaction. depth parameter as a number of splits it has to perform on a tree (starting from a single node).

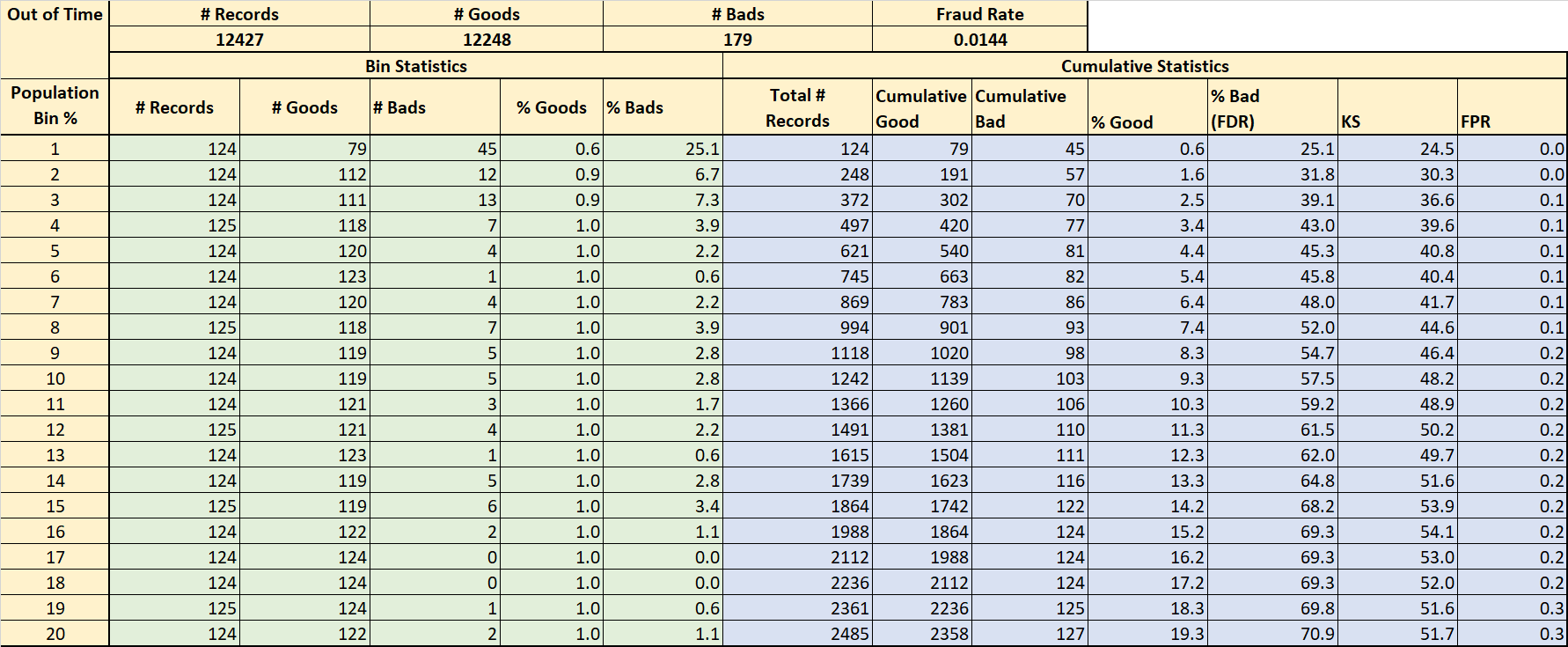
Using Gradient boosting, we got FDR@3% of 84.0% for the training dataset and FDR@3% of 76.4% for the testing dataset. For the OOT dataset, we got FDR@3% of 33.0%.

# Results

According to the analysis above, we concluded that Random Forest is superior to the rest three models we built. The following tables summarize both bin statistics and cumulative statistics when random forest is applied to detect fraud from 1% to 20% population in training, testing and Out-of-time datasets. Each table contains the information about the number of goods, bads, percentage of goods, percentage of bads, cumulative goods, cumulative bads, FDR, KS and false positive ratio. At 3% population cut-off, the result is FDR@3% of 99.0% for the training dataset and FDR@3% of 85.3% for the testing dataset. For the OOT dataset, we got FDR@3% of 39.1%.







Fraud Savings Calculation

* Loss Sales = $50 \* number of false positive

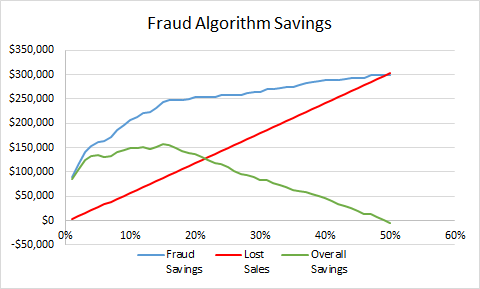
When we flag a good transaction as a fraud, we anger the customer, and some of them leave. On average, we assume a $50 loss for each false positives.

* Fraud Saving = $2000 \* number of true positive.

We save $2000 for every fraud we catch with our algorithm.

* Overall savings = Fraud Saving - Loss Sales

Based on our out-of-time data, we could calculate our model’s business value. The following graph shows the fraud algorithm savings.



The blue line shows our business value on how much we saved by catching the right fraudulent transactions. The lost sales red line represents the wrongly caught transaction values. Combining these two lines together, we could get the net saving green line. At 15% cut-off, the loss sale is $87,100, and the fraud saving is $244,000. Therefore, the net savings of applying the random forest model in out-of-time dataset achieves the maximum ($156,900) at 15% population cutoff.

# Conclusions

## 8.1 Steps Performed

To build and select an optimal supervised fraud model based on the card transaction data, we first cleaned the data, and then built 269 variables incorporating amount information, card information and transaction location information. We used feature selection methods to reduce the number of variables to 20. We split the card transaction data and used the data in November and December 2010 as out out-of-time data. 75% of the transactions from January to October 2010 were randomly selected to be used as training data and the remaining 25% used as testing data. After that, we built four supervised fraud models using four different types of machine learning algorithms: Logistic Regression, Neural Nets, Random Forests, and Boosting Trees, and found that Random Forests outperforms the rest three.

## 8.2 Recommendations

Due to limited time and resources, we were only able to perform the work described in previous sections of this report. Given more time, we could use other machine learning techniques such as Decision Trees, Bagging, Support Vector Machine, etc. We might find a model that is a better fit than Random Forest to predict future fraud.

# Appendix

## 9.1 Appendix 1 - Candidate Variables

|  |  |  |  |
| --- | --- | --- | --- |
| 1 | Fraud | 135 | Actual/sum\_Cardnum\_Merchnum\_3d |
| 2 | mean\_Cardnum\_1d | 136 | sum\_Cardnum\_Merchnum\_7d |
| 3 | Actual/mean\_Cardnum\_1d | 137 | Actual/sum\_Cardnum\_Merchnum\_7d |
| 4 | mean\_Cardnum\_3d | 138 | sum\_Cardnum\_Merchnum\_14d |
| 5 | Actual/mean\_Cardnum\_3d | 139 | Actual/sum\_Cardnum\_Merchnum\_14d |
| 6 | mean\_Cardnum\_7d | 140 | sum\_Cardnum\_Merchnum\_30d |
| 7 | Actual/mean\_Cardnum\_7d | 141 | Actual/sum\_Cardnum\_Merchnum\_30d |
| 8 | mean\_Cardnum\_14d | 142 | count\_Cardnum\_Merchnum\_1d |
| 9 | Actual/mean\_Cardnum\_14d | 143 | Actual/count\_Cardnum\_Merchnum\_1d |
| 10 | mean\_Cardnum\_30d | 144 | count\_Cardnum\_Merchnum\_3d |
| 11 | Actual/mean\_Cardnum\_30d | 145 | Actual/count\_Cardnum\_Merchnum\_3d |
| 12 | max\_Cardnum\_1d | 146 | count\_Cardnum\_Merchnum\_7d |
| 13 | Actual/max\_Cardnum\_1d | 147 | Actual/count\_Cardnum\_Merchnum\_7d |
| 14 | max\_Cardnum\_3d | 148 | count\_Cardnum\_Merchnum\_14d |
| 15 | Actual/max\_Cardnum\_3d | 149 | Actual/count\_Cardnum\_Merchnum\_14d |
| 16 | max\_Cardnum\_7d | 150 | count\_Cardnum\_Merchnum\_30d |
| 17 | Actual/max\_Cardnum\_7d | 151 | Actual/count\_Cardnum\_Merchnum\_30d |
| 18 | max\_Cardnum\_14d | 152 | mean\_Cardnum\_Merch zip\_1d |
| 19 | Actual/max\_Cardnum\_14d | 153 | Actual/mean\_Cardnum\_Merch zip\_1d |
| 20 | max\_Cardnum\_30d | 154 | mean\_Cardnum\_Merch zip\_3d |
| 21 | Actual/max\_Cardnum\_30d | 155 | Actual/mean\_Cardnum\_Merch zip\_3d |
| 22 | median\_Cardnum\_1d | 156 | mean\_Cardnum\_Merch zip\_7d |
| 23 | Actual/median\_Cardnum\_1d | 157 | Actual/mean\_Cardnum\_Merch zip\_7d |
| 24 | median\_Cardnum\_3d | 158 | mean\_Cardnum\_Merch zip\_14d |
| 25 | Actual/median\_Cardnum\_3d | 159 | Actual/mean\_Cardnum\_Merch zip\_14d |
| 26 | median\_Cardnum\_7d | 160 | mean\_Cardnum\_Merch zip\_30d |
| 27 | Actual/median\_Cardnum\_7d | 161 | Actual/mean\_Cardnum\_Merch zip\_30d |
| 28 | median\_Cardnum\_14d | 162 | max\_Cardnum\_Merch zip\_1d |
| 29 | Actual/median\_Cardnum\_14d | 163 | Actual/max\_Cardnum\_Merch zip\_1d |
| 30 | median\_Cardnum\_30d | 164 | max\_Cardnum\_Merch zip\_3d |
| 31 | Actual/median\_Cardnum\_30d | 165 | Actual/max\_Cardnum\_Merch zip\_3d |
| 32 | sum\_Cardnum\_1d | 166 | max\_Cardnum\_Merch zip\_7d |
| 33 | Actual/sum\_Cardnum\_1d | 167 | Actual/max\_Cardnum\_Merch zip\_7d |
| 34 | sum\_Cardnum\_3d | 168 | max\_Cardnum\_Merch zip\_14d |
| 35 | Actual/sum\_Cardnum\_3d | 169 | Actual/max\_Cardnum\_Merch zip\_14d |
| 36 | sum\_Cardnum\_7d | 170 | max\_Cardnum\_Merch zip\_30d |
| 37 | Actual/sum\_Cardnum\_7d | 171 | Actual/max\_Cardnum\_Merch zip\_30d |
| 38 | sum\_Cardnum\_14d | 172 | median\_Cardnum\_Merch zip\_1d |
| 39 | Actual/sum\_Cardnum\_14d | 173 | Actual/median\_Cardnum\_Merch zip\_1d |
| 40 | sum\_Cardnum\_30d | 174 | median\_Cardnum\_Merch zip\_3d |
| 41 | Actual/sum\_Cardnum\_30d | 175 | Actual/median\_Cardnum\_Merch zip\_3d |
| 42 | count\_Cardnum\_1d | 176 | median\_Cardnum\_Merch zip\_7d |
| 43 | Actual/count\_Cardnum\_1d | 177 | Actual/median\_Cardnum\_Merch zip\_7d |
| 44 | count\_Cardnum\_3d | 178 | median\_Cardnum\_Merch zip\_14d |
| 45 | Actual/count\_Cardnum\_3d | 179 | Actual/median\_Cardnum\_Merch zip\_14d |
| 46 | count\_Cardnum\_7d | 180 | median\_Cardnum\_Merch zip\_30d |
| 47 | Actual/count\_Cardnum\_7d | 181 | Actual/median\_Cardnum\_Merch zip\_30d |
| 48 | count\_Cardnum\_14d | 182 | sum\_Cardnum\_Merch zip\_1d |
| 49 | Actual/count\_Cardnum\_14d | 183 | Actual/sum\_Cardnum\_Merch zip\_1d |
| 50 | count\_Cardnum\_30d | 184 | sum\_Cardnum\_Merch zip\_3d |
| 51 | Actual/count\_Cardnum\_30d | 185 | Actual/sum\_Cardnum\_Merch zip\_3d |
| 52 | mean\_Merchnum\_1d | 186 | sum\_Cardnum\_Merch zip\_7d |
| 53 | Actual/mean\_Merchnum\_1d | 187 | Actual/sum\_Cardnum\_Merch zip\_7d |
| 54 | mean\_Merchnum\_3d | 188 | sum\_Cardnum\_Merch zip\_14d |
| 55 | Actual/mean\_Merchnum\_3d | 189 | Actual/sum\_Cardnum\_Merch zip\_14d |
| 56 | mean\_Merchnum\_7d | 190 | sum\_Cardnum\_Merch zip\_30d |
| 57 | Actual/mean\_Merchnum\_7d | 191 | Actual/sum\_Cardnum\_Merch zip\_30d |
| 58 | mean\_Merchnum\_14d | 192 | count\_Cardnum\_Merch zip\_1d |
| 59 | Actual/mean\_Merchnum\_14d | 193 | Actual/count\_Cardnum\_Merch zip\_1d |
| 60 | mean\_Merchnum\_30d | 194 | count\_Cardnum\_Merch zip\_3d |
| 61 | Actual/mean\_Merchnum\_30d | 195 | Actual/count\_Cardnum\_Merch zip\_3d |
| 62 | max\_Merchnum\_1d | 196 | count\_Cardnum\_Merch zip\_7d |
| 63 | Actual/max\_Merchnum\_1d | 197 | Actual/count\_Cardnum\_Merch zip\_7d |
| 64 | max\_Merchnum\_3d | 198 | count\_Cardnum\_Merch zip\_14d |
| 65 | Actual/max\_Merchnum\_3d | 199 | Actual/count\_Cardnum\_Merch zip\_14d |
| 66 | max\_Merchnum\_7d | 200 | count\_Cardnum\_Merch zip\_30d |
| 67 | Actual/max\_Merchnum\_7d | 201 | Actual/count\_Cardnum\_Merch zip\_30d |
| 68 | max\_Merchnum\_14d | 202 | mean\_Cardnum\_Merch state\_1d |
| 69 | Actual/max\_Merchnum\_14d | 203 | Actual/mean\_Cardnum\_Merch state\_1d |
| 70 | max\_Merchnum\_30d | 204 | mean\_Cardnum\_Merch state\_3d |
| 71 | Actual/max\_Merchnum\_30d | 205 | Actual/mean\_Cardnum\_Merch state\_3d |
| 72 | median\_Merchnum\_1d | 206 | mean\_Cardnum\_Merch state\_7d |
| 73 | Actual/median\_Merchnum\_1d | 207 | Actual/mean\_Cardnum\_Merch state\_7d |
| 74 | median\_Merchnum\_3d | 208 | mean\_Cardnum\_Merch state\_14d |
| 75 | Actual/median\_Merchnum\_3d | 209 | Actual/mean\_Cardnum\_Merch state\_14d |
| 76 | median\_Merchnum\_7d | 210 | mean\_Cardnum\_Merch state\_30d |
| 77 | Actual/median\_Merchnum\_7d | 211 | Actual/mean\_Cardnum\_Merch state\_30d |
| 78 | median\_Merchnum\_14d | 212 | max\_Cardnum\_Merch state\_1d |
| 79 | Actual/median\_Merchnum\_14d | 213 | Actual/max\_Cardnum\_Merch state\_1d |
| 80 | median\_Merchnum\_30d | 214 | max\_Cardnum\_Merch state\_3d |
| 81 | Actual/median\_Merchnum\_30d | 215 | Actual/max\_Cardnum\_Merch state\_3d |
| 82 | sum\_Merchnum\_1d | 216 | max\_Cardnum\_Merch state\_7d |
| 83 | Actual/sum\_Merchnum\_1d | 217 | Actual/max\_Cardnum\_Merch state\_7d |
| 84 | sum\_Merchnum\_3d | 218 | max\_Cardnum\_Merch state\_14d |
| 85 | Actual/sum\_Merchnum\_3d | 219 | Actual/max\_Cardnum\_Merch state\_14d |
| 86 | sum\_Merchnum\_7d | 220 | max\_Cardnum\_Merch state\_30d |
| 87 | Actual/sum\_Merchnum\_7d | 221 | Actual/max\_Cardnum\_Merch state\_30d |
| 88 | sum\_Merchnum\_14d | 222 | median\_Cardnum\_Merch state\_1d |
| 89 | Actual/sum\_Merchnum\_14d | 223 | Actual/median\_Cardnum\_Merch state\_1d |
| 90 | sum\_Merchnum\_30d | 224 | median\_Cardnum\_Merch state\_3d |
| 91 | Actual/sum\_Merchnum\_30d | 225 | Actual/median\_Cardnum\_Merch state\_3d |
| 92 | count\_Merchnum\_1d | 226 | median\_Cardnum\_Merch state\_7d |
| 93 | Actual/count\_Merchnum\_1d | 227 | Actual/median\_Cardnum\_Merch state\_7d |
| 94 | count\_Merchnum\_3d | 228 | median\_Cardnum\_Merch state\_14d |
| 95 | Actual/count\_Merchnum\_3d | 229 | Actual/median\_Cardnum\_Merch state\_14d |
| 96 | count\_Merchnum\_7d | 230 | median\_Cardnum\_Merch state\_30d |
| 97 | Actual/count\_Merchnum\_7d | 231 | Actual/median\_Cardnum\_Merch state\_30d |
| 98 | count\_Merchnum\_14d | 232 | sum\_Cardnum\_Merch state\_1d |
| 99 | Actual/count\_Merchnum\_14d | 233 | Actual/sum\_Cardnum\_Merch state\_1d |
| 100 | count\_Merchnum\_30d | 234 | sum\_Cardnum\_Merch state\_3d |
| 101 | Actual/count\_Merchnum\_30d | 235 | Actual/sum\_Cardnum\_Merch state\_3d |
| 102 | mean\_Cardnum\_Merchnum\_1d | 236 | sum\_Cardnum\_Merch state\_7d |
| 103 | Actual/mean\_Cardnum\_Merchnum\_1d | 237 | Actual/sum\_Cardnum\_Merch state\_7d |
| 104 | mean\_Cardnum\_Merchnum\_3d | 238 | sum\_Cardnum\_Merch state\_14d |
| 105 | Actual/mean\_Cardnum\_Merchnum\_3d | 239 | Actual/sum\_Cardnum\_Merch state\_14d |
| 106 | mean\_Cardnum\_Merchnum\_7d | 240 | sum\_Cardnum\_Merch state\_30d |
| 107 | Actual/mean\_Cardnum\_Merchnum\_7d | 241 | Actual/sum\_Cardnum\_Merch state\_30d |
| 108 | mean\_Cardnum\_Merchnum\_14d | 242 | count\_Cardnum\_Merch state\_1d |
| 109 | Actual/mean\_Cardnum\_Merchnum\_14d | 243 | Actual/count\_Cardnum\_Merch state\_1d |
| 110 | mean\_Cardnum\_Merchnum\_30d | 244 | count\_Cardnum\_Merch state\_3d |
| 111 | Actual/mean\_Cardnum\_Merchnum\_30d | 245 | Actual/count\_Cardnum\_Merch state\_3d |
| 112 | max\_Cardnum\_Merchnum\_1d | 246 | count\_Cardnum\_Merch state\_7d |
| 113 | Actual/max\_Cardnum\_Merchnum\_1d | 247 | Actual/count\_Cardnum\_Merch state\_7d |
| 114 | max\_Cardnum\_Merchnum\_3d | 248 | count\_Cardnum\_Merch state\_14d |
| 115 | Actual/max\_Cardnum\_Merchnum\_3d | 249 | Actual/count\_Cardnum\_Merch state\_14d |
| 116 | max\_Cardnum\_Merchnum\_7d | 250 | count\_Cardnum\_Merch state\_30d |
| 117 | Actual/max\_Cardnum\_Merchnum\_7d | 251 | Actual/count\_Cardnum\_Merch state\_30d |
| 118 | max\_Cardnum\_Merchnum\_14d | 252 | Days\_since\_per\_Cardnum |
| 119 | Actual/max\_Cardnum\_Merchnum\_14d | 253 | Days\_since\_per\_Merchnum |
| 120 | max\_Cardnum\_Merchnum\_30d | 254 | Days\_since\_per\_Cardnum\_Merchnum |
| 121 | Actual/max\_Cardnum\_Merchnum\_30d | 255 | Days\_since\_per\_Cardnum\_Merch zip |
| 122 | median\_Cardnum\_Merchnum\_1d | 256 | Days\_since\_per\_Cardnum\_Merch state |
| 123 | Actual/median\_Cardnum\_Merchnum\_1d | 257 | ['Cardnum']\_7dvelo\_sum |
| 124 | median\_Cardnum\_Merchnum\_3d | 258 | ['Merchnum']\_7dvelo\_sum |
| 125 | Actual/median\_Cardnum\_Merchnum\_3d | 259 | ['Cardnum']\_14dvelo\_sum |
| 126 | median\_Cardnum\_Merchnum\_7d | 260 | ['Merchnum']\_14dvelo\_sum |
| 127 | Actual/median\_Cardnum\_Merchnum\_7d | 261 | ['Cardnum']\_30dvelo\_sum |
| 128 | median\_Cardnum\_Merchnum\_14d | 262 | ['Merchnum']\_30dvelo\_sum |
| 129 | Actual/median\_Cardnum\_Merchnum\_14d | 263 | ['Cardnum']\_7dvelo\_count |
| 130 | median\_Cardnum\_Merchnum\_30d | 264 | ['Merchnum']\_7dvelo\_count |
| 131 | Actual/median\_Cardnum\_Merchnum\_30d | 265 | ['Cardnum']\_14dvelo\_count |
| 132 | sum\_Cardnum\_Merchnum\_1d | 266 | ['Merchnum']\_14dvelo\_count |
| 133 | Actual/sum\_Cardnum\_Merchnum\_1d | 267 | ['Cardnum']\_30dvelo\_count |
| 134 | sum\_Cardnum\_Merchnum\_3d | 268 | ['Merchnum']\_30dvelo\_count |
|  |  | 269 | RANDOM |

## 9.2 Appendix 2 – Data Quality Report

1. **Dataset Summary**

This credit card transaction dataset has **96,753 records**, **9 independent fields**, and **one dependent field,** which represent whether the record is fraud or not. This dataset is gathered from state of Tennessee. The time period is from 01/01/2010 to 12/31/2010.

1. **Field Summary**

**Categorical Field**

There are 8 categorical fields in this data set. Summary information for each categorical fields is shown as Table 1.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Field | Count | % Populated | Unique Value | Most Common |
| Recnum | 96,753 | 100.00% | 96,753 | Uniform Distributed |
| Cardnum | 96,753 | 100.00% | 1,645 | 5142148452 |
| Date | 96,753 | 100.00% | 365 | 2/28/10 |
| Merchnum | 93,378 | 96.51% | 13,092 | 9.3009E+11 |
| Merch description | 96,753 | 100.00% | 13,126 | GSA-FSS-ADV |
| Merch state | 95,558 | 98.76% | 228 | TN |
| Merch zip | 92,097 | 95.19% | 4,568 | 38118 |
| Transtype | 96,753 | 100.00% | 4 | P |

**Table 1.**

**Numerical Field**

In this dataset there is only 1 numerical field. Table 2 represents some basic statistic summary for this field.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| field | count | %Populated | mean | std | min | 25% | 50% | 75% | max |
| Amount | 96,753 | 100.00% | 427.9 | 10,006 | 0.01 | 33.48 | 137.98 | 428.2 | 3102046 |

**Table 2.**

**Response Field**

“Fraud” is a response field in this dataset. There are two types, “0” and “1”. Type “0” categorized as non-fraud transaction and Type “1” represents fraud transaction. Summary information is shown as Table 3.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Field | records | % Populated | Non-Fraud | Fraud |
| Fraud | 96,753 | 100.00% | 95694 (98.9%) | 1059 (1.1%) |

**Table 3.**

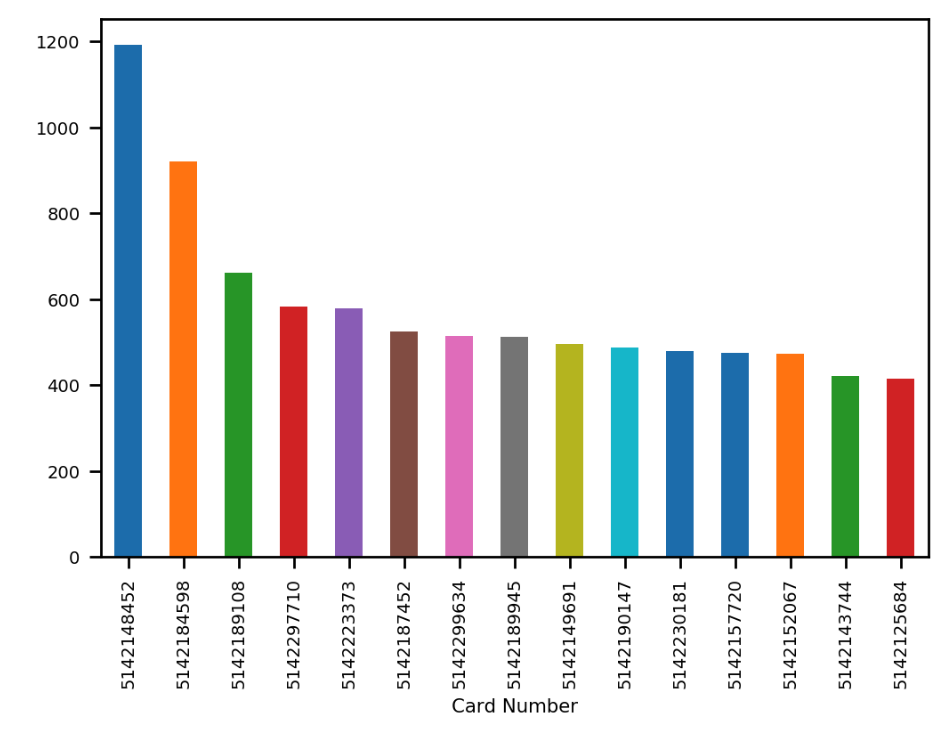
**III. Fields Description**

1. **Recnum**

**Recnum** represents the numbering for each record. There are 96753 unique values in the dataset which means each record has one corresponding **Recnum.**

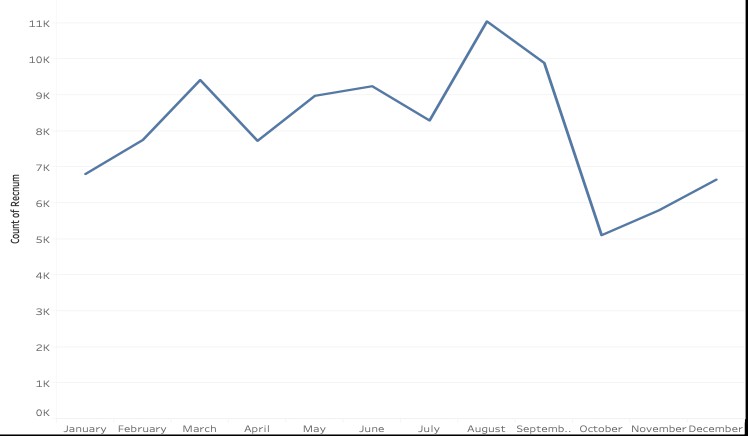
1. **Cardnum**

**Cardnum** stands for credit card number. The following graph shows the transaction count for the top 15 most common credit card numbers.

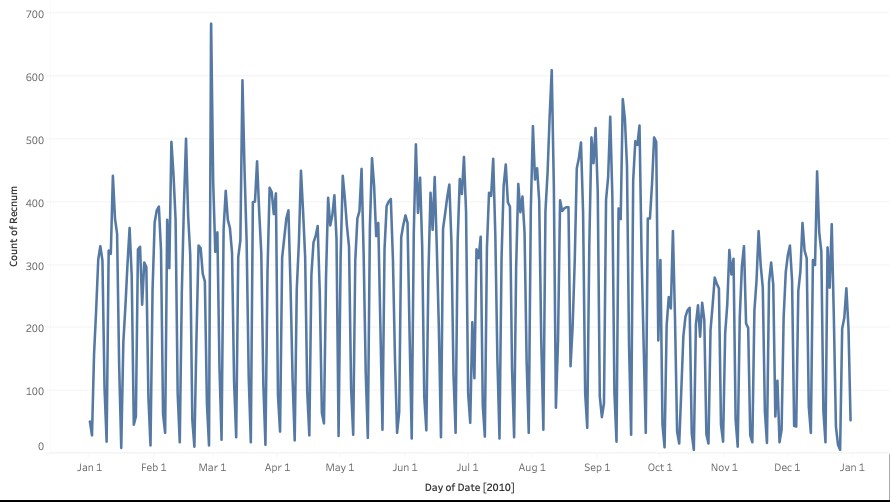


1. **Date**

The Date field contains 365 unique values, which are date from 1/1/2010 to 12/31/2010.



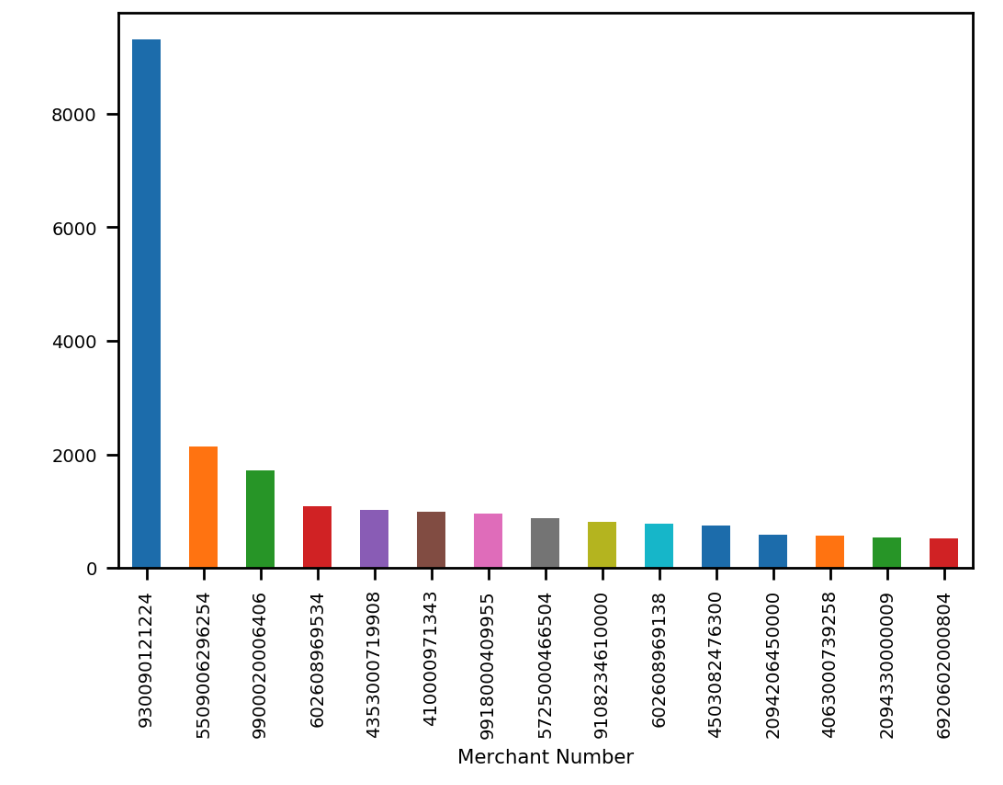
Count of Transactions by month



Count of Transactions by date

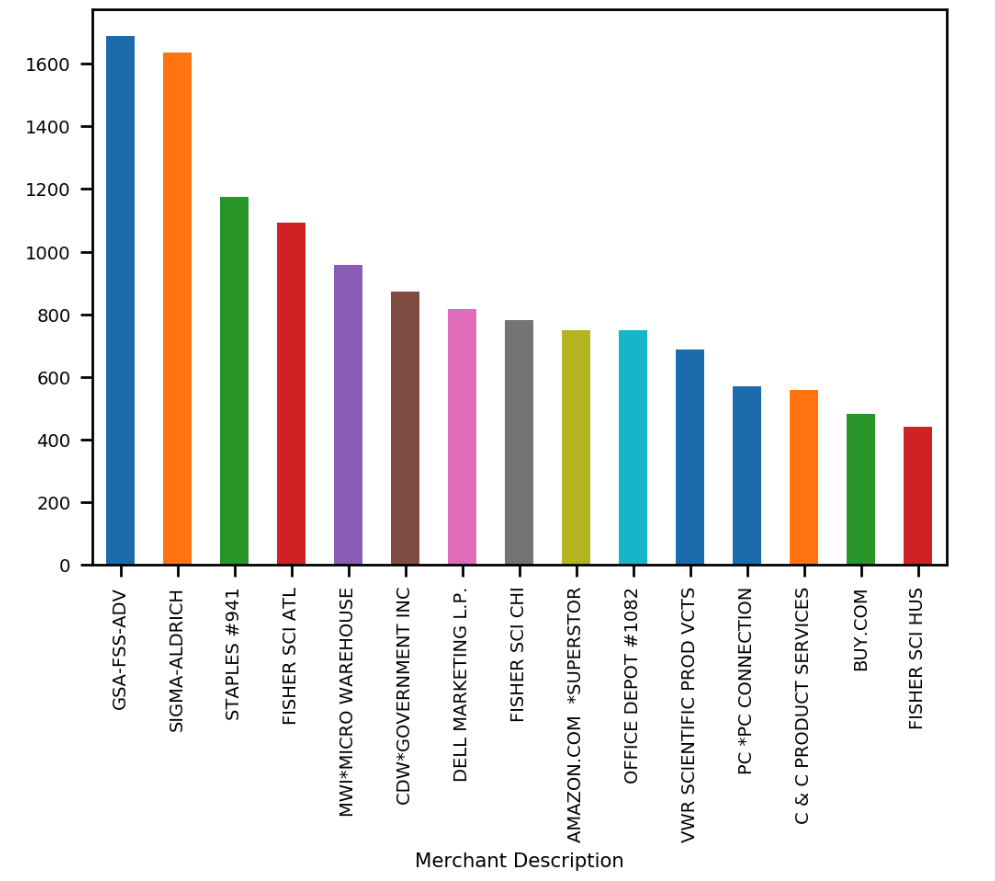
1. **Merchnum**

**Merchnum** represents merchant number. The graph below shows the count for the top 15 most common merchant numbers.



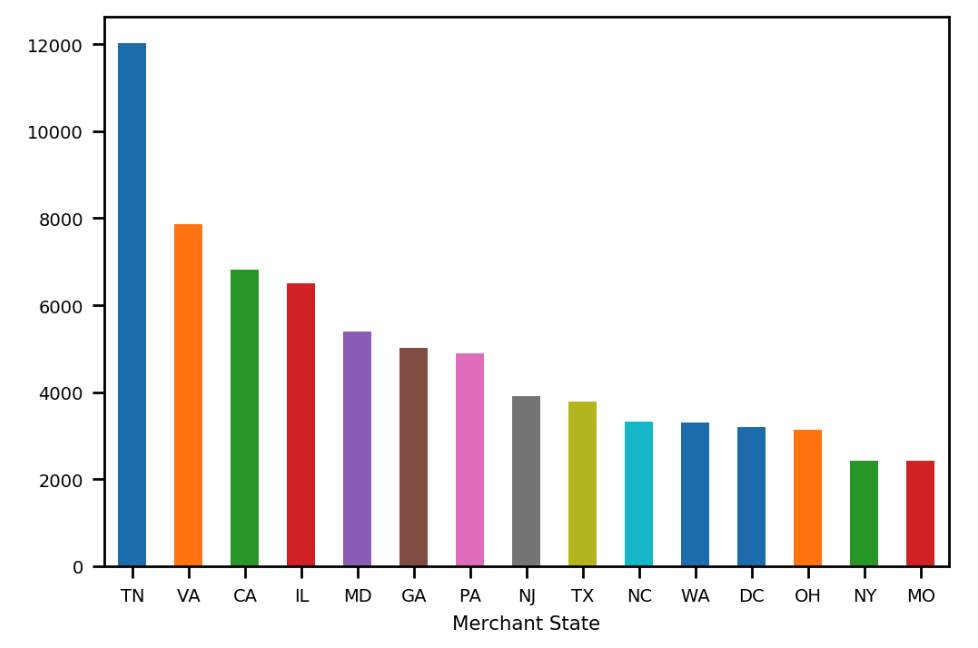
1. **Merch description**

**Merch description** is merchant description. Notice that there are 13,126 unique values for merchant description, which is more than that of merchant numbers. Below graph shows the count for the top 15 most common merchant description.



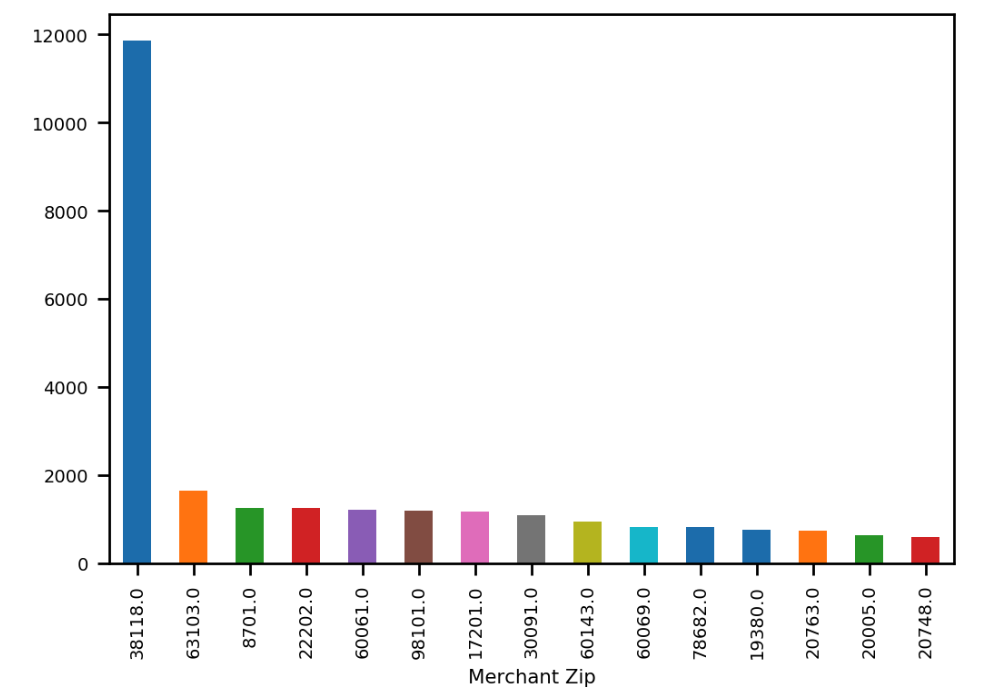
1. **Merch state**

**Merch state** represents the state where the merchant located. The following graph shows the count for the top 15 most common merchant states.



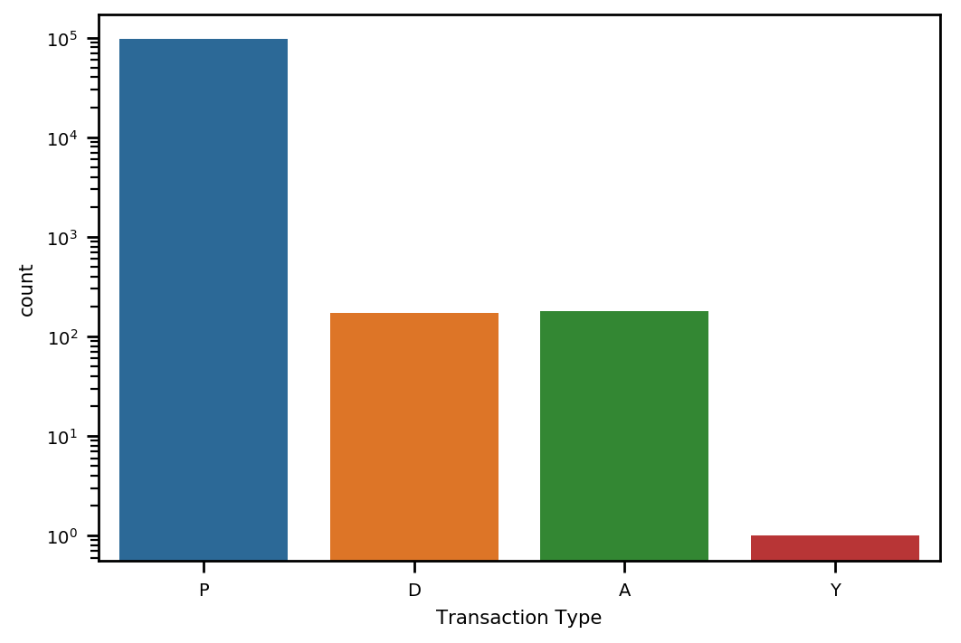
1. **Merch zip**

**Merch zip** represents the zip code where the merchant located. The following graph shows the count for the top 15 most common merchant zip code.



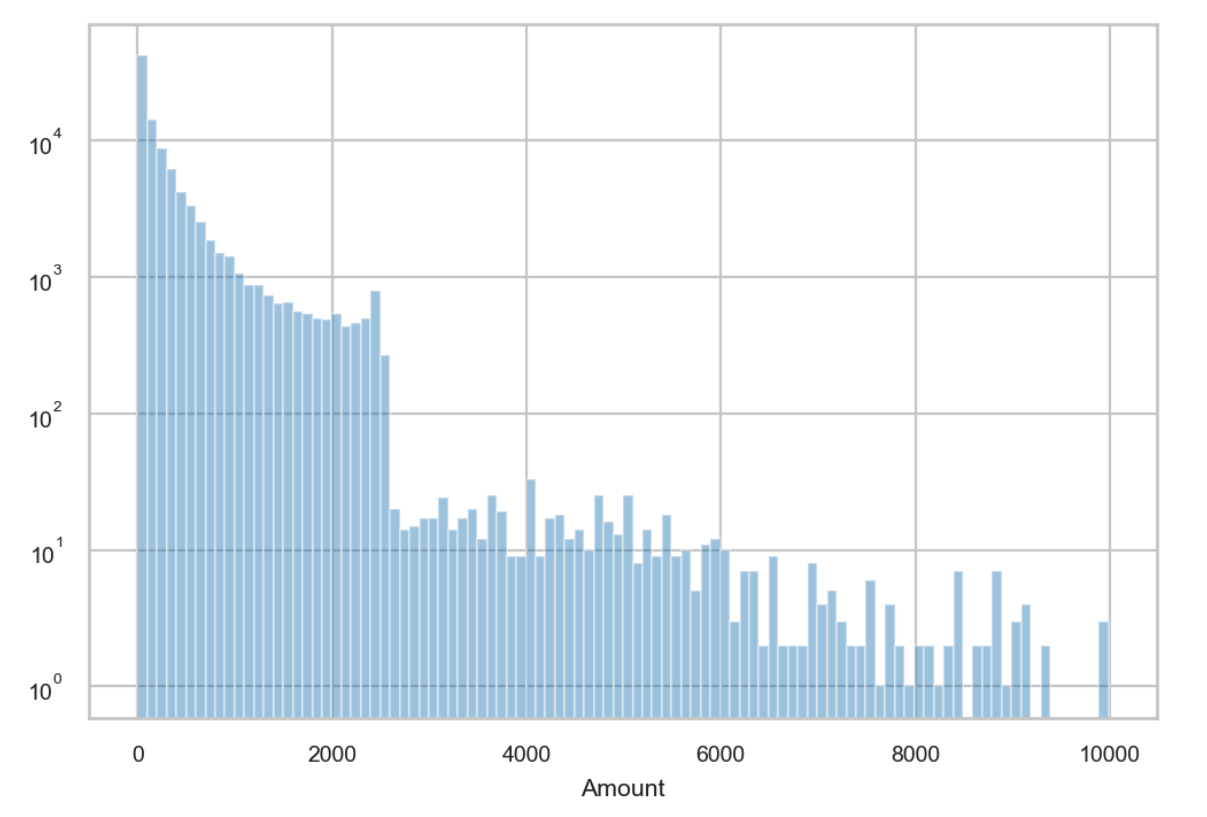
1. **Transtype**

**Transtype** stands for transaction types. There are 4 types of transaction in this dataset. “P”, which means “Purchase”, is the most common type in this filed. The following graph show the count for all types of transaction.



1. **Amount**

**Amount** is the transaction amount. Below graph shows the log scale distribution of transaction amount with amount limited to 10,000.



1. **Fraud**

**Fraud** is a response field in this dataset. There are two types, “0” and “1”. Type “0” categorized as non-fraud transaction and Type “1” represents fraud transaction. The fraud records account for about 1.1% of the whole dataset. The following is the log scale count of Fraud type.

