FRAUD ANALYSIS

ON CREDIT CARD TRANSACTIONS

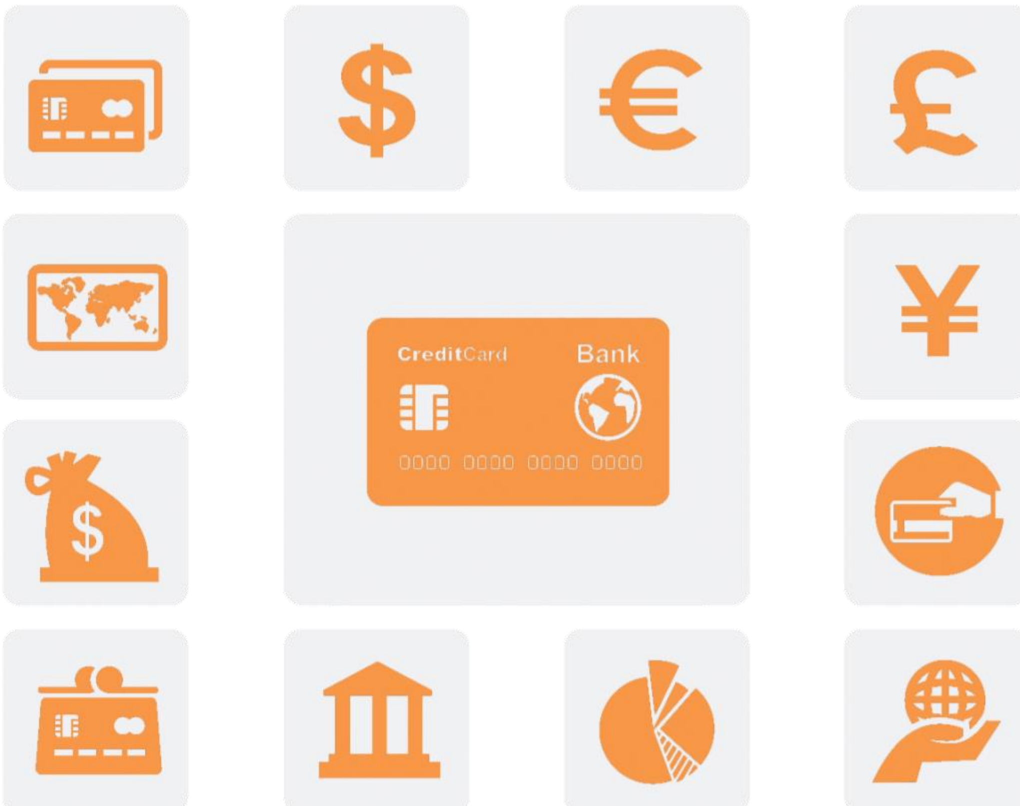


TABLE OF CONTENTS

1. Executive Summary

This report provides an analysis and evaluation of the *Credit Card Transaction Data* for detecting fraud using supervised machine learning methods. The general flow of our process is from data description, data preparation of expert variables, feature selection, application of the fraud algorithm, result evaluation and business interpretation. R and Python were the tools that we used to derive our results.

The data set contains 96,708 records of approved card transactions with 9 features of transaction details. We first handled missing values and exclude all non-purchasing type records. Next, we focused our attention on four important variables including card number, merchant number, merchant zip code and transaction amount for further analysis. We built 75 expert variables by link analysis and profile method.

Dataset was split into three parts - training set, testing set, and out-of-time (OOT) set. Then, feature selection was conducted on training and testing set by Kolmogorov-Smirnov (KS) test to reduce dimension before building models. Twenty variables with high KS values were selected and used in building five fraud detecting models including Logistic Regression, Random Forest, XGboost, Neural Network, and Support Vector Machine. Models were tuned by testing set, which is based on the performance measured by fraud detection rate (FDR) at 2% population bin. Finally, models were validated by out-of-time set.

As a result, Random Forest model achieved the best outcome, which is 64.54% of Fraud Detection Rate (FDR) at 2% location for OOT set. Taking assumptions to monetize the prediction, this model leads to a net return of $172,840 when binning at 7% location (assumptions indicated below). For future improvement of the results, we consider two aspects regarding dependent and independent variables. Together with enriching independent variables by more data collection and exploration, we also advise to enlarge the number of classes (very risky/risky/good, etc.) to allow for a wider range of risk tolerance and actions.

2. Data Description

2.1 Summary of Dataset

File Name: card transactions.csv
Data Source: This dataset is partly simulated based on real card transaction records
Data Size: 96,708 records
Number of Fields: 9 (excluding the first **Recordnum** field as it can be considered as index column)
Time: Jan 1st, 2010 - Dec 31st, 2010

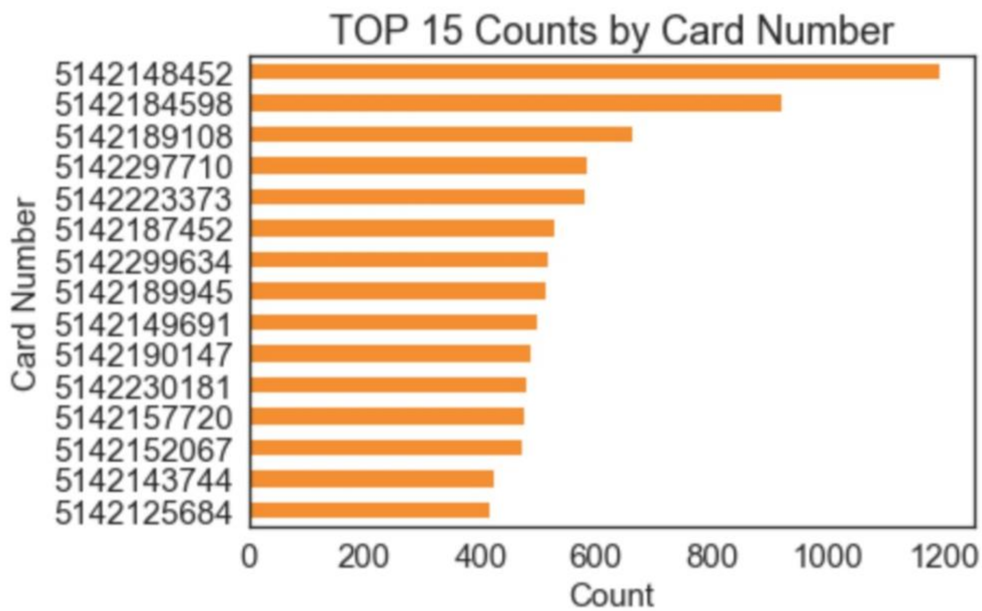
The Credit Card Transaction data contains 96,708 records of transaction information. It includes information such as the card number, the date of the transaction, the merchant number, the merchant description, the merchant state, the merchant zip code, transaction types, and transaction amount. The last field is used to indicate whether the record is fraudulent or not. In total, there are 1014 labeled fraudulent records. The information that we believe to be important comprising card number, merchant number, merchant zip code and transaction amount are also recorded in this data.

2.2 Important Variables

There are four variables in the dataset that we deem important in our analysis of potential fraud in *Credit Card Transaction Data*. Following is the description of those variables. The complete Data Quality Report can be found in appendix.

1. Cardnum

Cardnum is the categorical variable indicating credit card number used for the transaction. The numbers in this field all have 10 digits. The field is 100% populated with 1,644 unique entities, meaning that this data involves transaction information from 1,644 credit cards. **Cardnum** is important as it is a unique number of each credit card, and it allows us to build expert variables that count the number of transactions each card has in the past 1, 3, 7, 14, 30 days. The top 15 frequently reported **Cardnum** records are shown by the bar chart below followed by their specific counts.



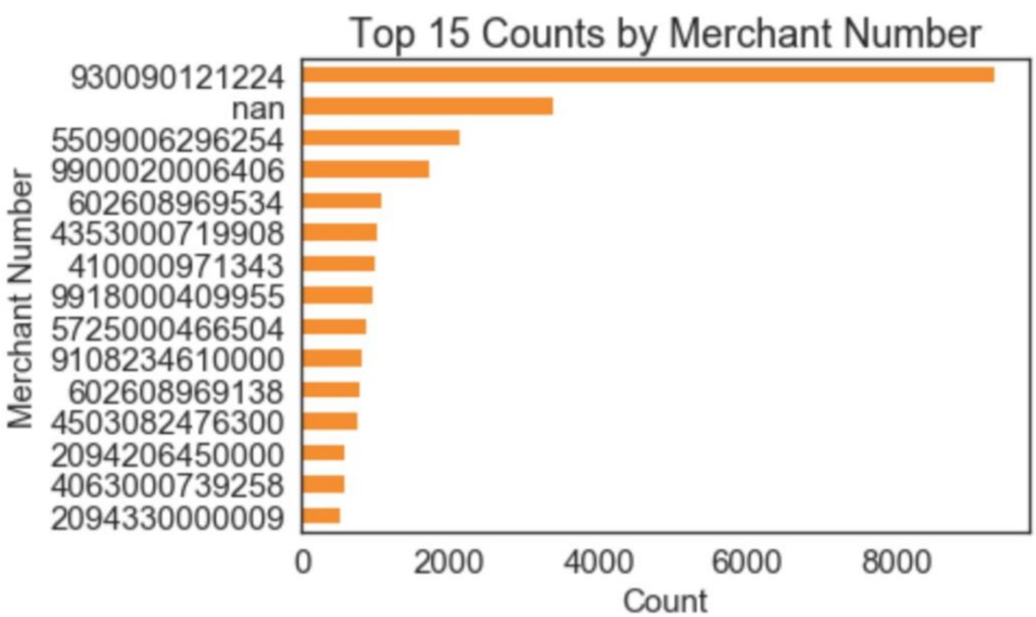
The **Cardnum** record of "5142148452" appeared extremely often (1192 times).

CARDNUM	FREQUENCY
5142148452	1192
5142184598	921
5142189108	663
5142297710	583
5142223373	579
5142187452	526
5142299634	515
5142189945	512
5142149691	497
5142190147	488
5142230181	479
5142157720	475
5142152067	473

5142143744	422
5142125684	415

2. Merchantnum

Merchantnum is the categorical variable that signals the merchant's number recorded in each transaction. This field is 96.51% populated with 13,091 unique values. **Merchantnum** is important as we use it to build expert variables that count the number of transactions related to each merchant in the past 1, 3, 7, 14, 30 days. We also explored the links between **Cardnum** and **Merchantnum** by creating variables to see the number of merchants related to each card and the number of cards related to each merchant in the past 1, 3, 7, 14, 30 days. The top 15 frequently reported **Merchantnum** records are shown by the bar chart below followed by their specific counts.



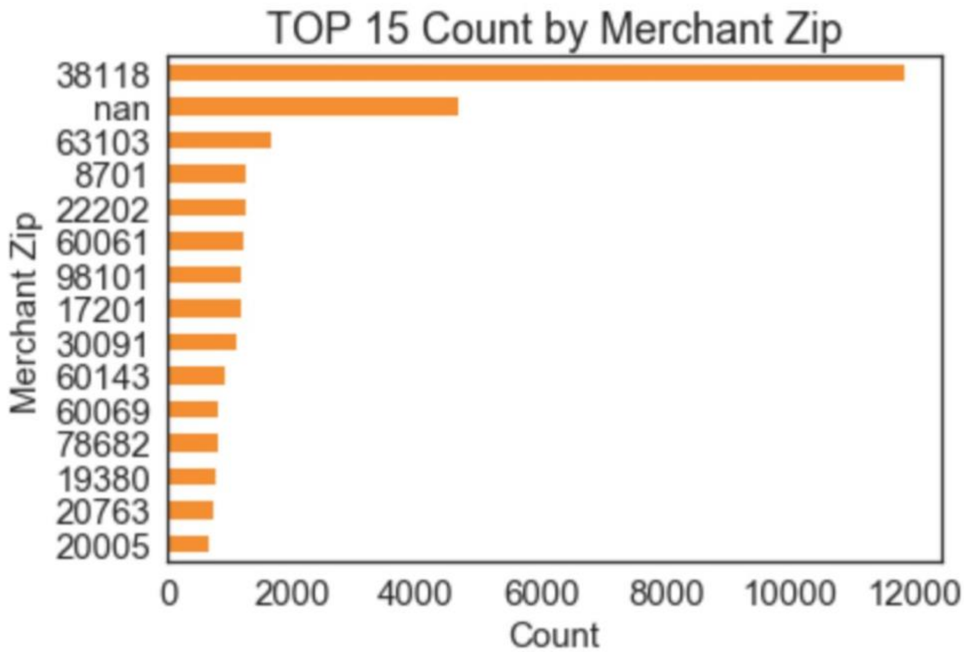
The top record "930090121224", which appeared 9310 times, is linked to FEDEX. The result indicates that "930090121224" is a frivolous record while FEDEX in fact is considered as a legitimate entity. We will need to look more into this issue to understand the result. The top record's appearance is more than three times that of the second most frequent record, which suggests further analysis.

MERCHANTNUM	FREQUENCY
930090000000	9310
602609000000	2758
5509010000000	2131

9900020000000	1881
4503080000000	1736
9900000000000	1727
9108230000000	1403
4353000000000	1022
410001000000	982
9918000000000	958
5725000000000	872
6859860000000	858
900009000000	819
2376700000000	777
2094210000000	696

3. Merchant Zip

Merchant Zip is a categorical variable indicating the zip code of the merchant. This field is 98.76% populated with 4,568 unique zip codes. Zip code is important as it indicates the location of the merchant, which would be useful in identifying fraud. We create expert variables to understand how zip codes related to each card and each merchant in the past 1, 3, 7, 14, 30 days. The top 15 frequently reported **Merchant Zip** records are shown below, including the “nan” records (4301 counts).



The counts showed that “38118” appeared the most with 11823 records.

MERCHANT ZIP	FREQUENCY
38118	11823
nan	4301
63103	1650
08701	1267
22202	1250
60061	1221
98101	1197
17201	1180
30091	1092
60143	942
60069	826
78682	817
19380	769
20763	749

20005	648

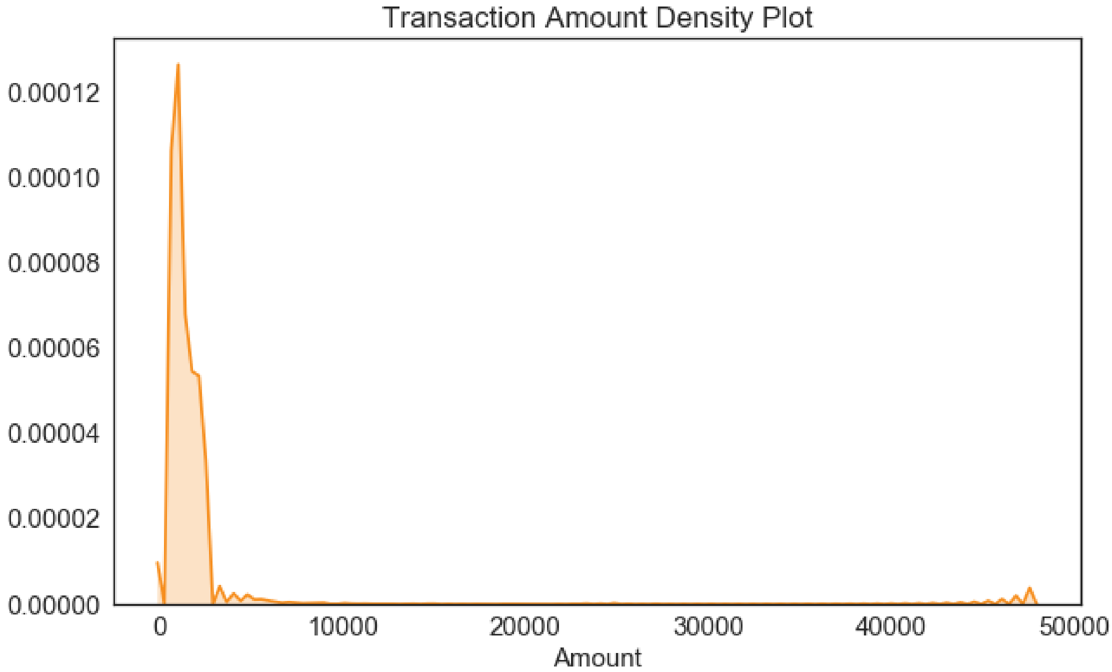
4. Amount

Amount is a numerical variable indicating the amount of each transaction. This field is 100% populated with 34,876 unique values. **Amount** is important for us to create variables that record the average/total/maximum/median amount spent by each card in the past 1, 3, 7, 14, 30 days; and the average/total/maximum/median amount received by each merchant in the past 1, 3, 7, 14, 30 days.

Mathematical statistics such as mean, median (which is the 50th percentile), standard deviation, minimum, maximum, the first and third percentile for **Amount** are shown below.

STATISTICS	RESULT
Count	96708
Mean	427.865
Std	10008.47
Min	0.01
25%	33.45
Median (50%)	137.9
75%	427.715
Max	3102046

Following is the distribution for **Amount**.



The most extreme amount is over $3 Million, which may indicate a currency error. We excluded this outlier amount.

3. Data Preparation

For purposes of fraudulent transaction detection, we only focused on approved transactions, so we selected records that have **Transtype** as P. Then, we removed the dollar sign of each **Amount** variable.

3.1 Handling Missing Value

Based on the Data Quality Report (Appendix), **Merchantnum**, **Merchant State** and **Merchant Zip** are affected by missing values.

- 3.49% of all records do not have **Merchantnum**
- 1.24% of all records do not have **Merchant State**
- 4.81% of all records do not have **Merchant Zip**.

For the records that have **Merchant Zip** value but missing **Merchant State** value, we used another zip code and state table found online [1] to impute the missing **Merchant State** value. The rest of missing values in **Merchant State** are replaced with “UNKNOWN”. Missing values in **Merchant Zip** are replaced with 0.

For the missing values in **Merchantnum**, we filled them with the number of the merchant that shares the same description. The rest of missing values are replaced with 0.

3.2 Expert Variables

Among 9 categorical variables, we choose 4 variables including **Cardnum, Merchantnum, Merchant Zip and Amount** to build expert variables. Since the dataset related to time series, we built expert variables regarding different time windows, meaning 1 day, 3 days, 7 days, 14 days and 30 days, for further analysis.

1. **Type I variables** capture unusual records based on transaction frequency for each specific card or merchant in a particular time window.

- **Card_frequency_x** describes transaction frequency of a specific card in past x days.
- **Merchant _frequency_x** describes transaction frequency of a specific merchant in past x days.

2. **Type II variables** capture unusual records based the count of one entity associated with another entity in a particular time window.

- **Card_merchant_x** describes the number of unique merchants related to a specific card in past x days.
- **Merchant_card _x** describes the number of unique cards related to a specific merchant in past x days.

3. **Type III variables** capture unusual transactions based on the ratio of transaction amount to its related statistical data in a particular time window.

- **Avg_Amount_Card_x** is the ratio of transaction amount of a particular card to the historical averages amount of the same card in past x days.
- **Total_Amount_Card_x** is the ratio of transaction amount of a particular card to the total transaction amount of the same card in past x days.
- **Median_Amount_Card_x** is the ratio of transaction amount of a particular card to the historical median amount of same card in past x days.
- **Max_Amount_Card_x** is the ratio of transaction amount of a particular card to the max amount of same card in past x days.
- **Avg_Amount_Merch _x** is the ratio of transaction amount of a particular merchant to the historical averages amount of the same card in past x days.
- **Total_Amount_ Merch _x** is the ratio of transaction amount of a particular merchant to the total transaction amount of the same card in past x days.
- **Median_Amount_ Merch _x** is the ratio of transaction amount of a particular merchant to the historical median amount of same card in past x days.
- **Max_Amount_ Merch _x** is the ratio of transaction amount of a particular merchant to the max amount of same card in past x days.

4. **Type IV variables** capture unusual transactions based on location.

- **zip_card_x** is the number of different zip codes related to a specific card in past x days.
- **zip_merchant _x** is the number of different zip codes related to a specific merchant in past x days.

5. **Type V variables** capture unusual transactions based on fraud history of the same card.

- **fraud_times_x is the count of fraud records related to a specific card in past x days.**

3.3 Table of Expert Variables

Variable	Description
card_freq_1	The transaction frequency of a specific card in the past 1 day
card_freq_3	The transaction frequency of a specific card in the past 3 days
card_freq_7	The transaction frequency of a specific card in the past 7 days
card_freq_14	The transaction frequency of a specific card in the past 14 days
card_freq_30	The transaction frequency of a specific card in the past 30 days
merchant_freq_1	The transaction frequency of a specific merchant in the past 1 day

merchant_freq_3	The transaction frequency of a specific merchant in the past 3 days
merchant_freq_7	The transaction frequency of a specific merchant in the past 7 days
merchant_freq_14	The transaction frequency of a specific merchant in the past 14 days
merchant_freq_30	The transaction frequency of a specific merchant in the past 30 days
merchant_card_1	The number of merchants related with a certain card the in past 1 day
merchant_card_3	The number of merchants related with a certain card in the past 3 days
merchant_card_7	The number of merchants related with a certain card in the past 7 days
merchant_card_14	The number of merchants related with a certain card in the past 14 days
merchant_card_30	The number of merchants related with a certain card in the past 30 days
card_merchant_1	The number of cards related with a certain merchant in the past 1 day
card_merchant_3	The number of cards related with a certain merchant in the past 3 days
card_merchant_7	The number of cards related with a certain merchant in the past 7 days
card_merchant_14	The number of cards related with a certain merchant in the past 14 days
card_merchant_30	The number of cards related with a certain merchant in the past 30 days
Avg_Amount_Card_1	The ratio of transaction amount to the historical averages amount of the same card in the past 1 day
Avg_Amount_Card_3	The ratio of transaction amount to the historical averages amount of the same card in the past 3 days
Avg_Amount_Card_7	The ratio of transaction amount to the historical averages amount of the same card in the past 7 days
Avg_Amount_Card_14	The ratio of transaction amount to the historical averages amount of the same card in past the 14 days
Avg_Amount_Card_30	The ratio of transaction amount to the historical averages amount of the same card in the past 30 days
Total_Amount_Card_1	The ratio of transaction amount to the total amount of the same card in past the 1 day
Total_Amount_Card_3	The ratio of transaction amount to the total amount of the same card in the past 3 days
Total_Amount_Card_7	The ratio of transaction amount to the total amount of the same card in the past 7 days

Total_Amount_Card _14	The ratio of transaction amount to the total amount of the same card in the past 14 days
Total_Amount_Card _30	The ratio of transaction amount to the total amount of the same card in the past 30 days
Median_Amount_Card_1	The ratio of transaction amount to the historical median amount of the same card in the past 1 day
Median_Amount_Card_3	The ratio of transaction amount to the historical median amount of the same card in the past 3 days
Median_Amount_Card_7	The ratio of transaction amount to the historical median amount of the same card in the past 7 days
Median_Amount_Card_14	The ratio of transaction amount to the historical median amount of the same card in the past 14 days
Median_Amount_Card_30	The ratio of transaction amount to the historical median amount of the same card in the past 30 days
Max_Amount_Card_1	The ratio of transaction amount to the max amount of the same card in the past 1 day
Max_Amount_Card_3	The ratio of transaction amount to the max amount of the same card in the past 3 days
Max_Amount_Card_7	The ratio of transaction amount to the max amount of the same card in the past 7 days
Max_Amount_Card_14	The ratio of transaction amount to the max amount of the same card in the past 14 days
Max_Amount_Card_30	The ratio of transaction amount to the max amount of the same card in the past 30 days
Avg_Amount_Merch_1	The ratio of transaction amount to the historical averages amount of the same merchant in the past 1 day
Avg_Amount_Merch_3	The ratio of transaction amount to the historical averages amount of the same merchant in the past 3 days
Avg_Amount_Merch_7	The ratio of transaction amount to the historical averages amount of the same merchant in the past 7 days
Avg_Amount_Merch_14	The ratio of transaction amount to the historical averages amount of the same merchant in the past 14 days
Avg_Amount_Merch_30	The ratio of transaction amount to the historical averages amount of the same merchant in the past 30 days

Total_Amount_Merch_1	The ratio of transaction amount to the total amount of the same merchant in the past 1 day
Total_Amount_Merch_3	The ratio of transaction amount to the total amount of the same merchant in the past 3 days
Total_Amount_Merch_7	The ratio of transaction amount to the total amount of the same merchant in the past 7 days
Total_Amount_Merch_14	The ratio of transaction amount to the total amount of the same merchant in the past 14 days
Total_Amount_Merch_30	The ratio of transaction amount to the total amount of the same merchant in the past 30 days
Median_Amount_Merch_1	The ratio of transaction amount to the historical median amount of the same merchant in the past 1 day
Median_Amount_Merch_3	The ratio of transaction amount to the historical median amount of the same merchant in the past 3 days
Median_Amount_Merch_7	The ratio of transaction amount to the historical median amount of the same merchant in the past 7 days
Median_Amount_Merch_14	The ratio of transaction amount to the historical median amount of the same merchant in the past 14 days
Median_Amount_Merch_30	The ratio of transaction amount to the historical median amount of the same merchant in the past 30 days
Max_Amount_Merch_1	The ratio of transaction amount to the max amount of the same merchant in the past 1 day
Max_Amount_Merch_3	The ratio of transaction amount to the max amount of the same merchant in the past 3 days
Max_Amount_Merch_7	The ratio of transaction amount to the max amount of the same merchant in the past 7 days
Max_Amount_Merch_14	The ratio of transaction amount to the max amount of the same merchant in the past 14 days
Max_Amount_Merch_30	The ratio of transaction amount to the max amount of the same merchant in the past 30 days
zip_card_1	The number of different zip codes related to a particular card in the past 1 day
zip_card_3	The number of different zip codes related to a particular card in the past 3 days

zip_card_7	The number of different zip codes related to a particular card in the past 7 days
zip_card_14	The number of different zip codes related to a particular card in the past 14 days
zip_card_30	The number of different zip codes related to a particular card in the past 30 days
zip_merchant_1	The number of different zip codes related to a particular merchant in the past 1 day
zip_merchant_3	The number of different zip codes related to a particular merchant in the past 3 days
zip_merchant_7	The number of different zip codes related to a particular merchant in the past 7 days
zip_merchant_14	The number of different zip codes related to a particular merchant in the past 14 days
zip_merchant_30	The number of different zip codes related to a particular merchant in the past 30 days
fraud_times_1	The count of fraudulent transaction related to the same card in the past 1 day
fraud_times_3	The count of fraudulent transaction related to the same card in the past 3 days
fraud_times_7	The count of fraudulent transaction related to the same card in the past 7 days
fraud_times_14	The count of fraudulent transaction related to the same card in the past 14 days
fraud_times_30	The count of fraudulent transaction related to the same card in the past 30 days

3.4 Feature Selection

We implemented feature selection to reduce dimension before building models. However, before feature selection, we split dataset into three parts - training set, testing set and out of time set. We selected variables on training set and estimated model properties on testing set, and finally validated model on out-of-time set to calculate Fraud Detect Rate. We reserved the most recent two-month data as out of time set (12586 rows). For the rest of data, we used function **train_test_split()** with **test_size = 0.3** to randomly split data into testing set (25131 rows) and training set (58636 rows) on Python. Furthermore, we changed the training set's fraud labels ('0' and '1') into integer for predictor selection.

In the process of feature selection, we implemented an approach named Kolmogorov-Smirnov (KS) test. KS is used for a variable that is continuous or has a metric or ordering. For each variable, KS can make separate distributions for the two populations (good/bad). The amount of separation (D value in KS) between the distributions is the importance of the variable. KS value represents the maximum of the difference of the cumulative.

Then we calculated KS value for each variable. We found out that there are several variables with high KS value, meaning that they are important variables. Then we sorted KS values from largest to smallest and filtered out the following 20 variables with KS values larger than or equal to 0.34.

Variable	KS Value
fraud_times_1	0.63
fraud_times_14	0.58
fraud_times_7	0.57
fraud_times_3	0.54
fraud_times_1	0.44
Total_Amount_Card_7	0.40
Total_Amount_Card_14	0.39
Avg_Amount_Merch_3	0.38
Max_Amount_Card_30	0.37
Max_Amount_Merch_3	0.36
Total_Amount_Card_3	0.36
Total_Amount_Merch_7	0.36
Avg_Amount_Card_3	0.35
Median_Amount_Card_3	0.35
Total_Amount_Merch_3	0.35
Max_Amount_Card_14	0.35
Max_Amount_Card_3	0.35
Total_Amount_Card_30	0.35
Max_Amount_Card_7	0.34
Avg_Amount_Card_30	0.34

4. Building Fraud Algorithm

4.1 Logistic Regression

Logistic Regression model is a regression model where dependent variables are categorical. It uses sigmoid function to estimate the probability of fraud or non-fraud for each record. Compared with the following candidate models, this one is easy-to-built and fast-to-train. If taking the fraud label of '0' and '1' numerically, linear regression could be similarly applied to this project as we are taking the continuous regression outputs as the fraud score.

We used **glm()** function by setting **family = "binomial"** to run logistic regression in R**.**

4.2 Random Forest

Decision Tree partitions the feature spaces into multiple high-dimensional boxes and give predictions according to the majority vote in each box. Random Forest is a modified version of decision tree, by using an ensemble of trees and averaging the result across all the trees. Each tree is built from a random subset of features from the entire feature set.

We imported **RandomForestClassifier()** from **sklearn.ensemble** package in Python. We built 150 trees in the forest (**n_estimator = 150**) and also remained other parameters as default.

4.3 XGBoost

XGboost stands for Extreme Gradient Boosting Trees. Tree boosting is an ensemble method that seeks to create a strong classifier based on "weak" classifiers. In this context, weak and strong refer to a measure of the correlation between the learners and the actual target variable. By adding models on top of each other iteratively, the errors of the previous model are corrected by the next predictor, until the training data is accurately predicted or reproduced by the model. Gradient Boosting also comprises an ensemble method that sequentially adds predictors and corrects previous models. However, instead of assigning different weights to the classifiers after every iteration, this method fits the new model to new residuals of the previous prediction and then minimizes the loss when adding the latest prediction. In the end, the model is updated using gradient descent. XGBoost implements this algorithm with an additional custom regularization term in the objective function to control overfitting.

We imported **XGBClassifier()** from **xgboost** package in Python with following parameters:

- The fraction of columns to be randomly samples for each tree: colsample_bytree = 0.9.
- Learning rate: eta = 0.1.
- The maximum depth of a tree: max_depth = 7.
- The min sum of weights of all observations required in a child: min_child_weight = 1.
- Number of trees: n_estimators = 500.
- Objective: objective = 'binary:logistic'.
- The fraction of observations to be randomly samples for each tree: subsample = 0.8.

4.4 Neural Network

Neural Network neural network consists of units (neurons), arranged in layers, which convert an input vector into some output. Each unit takes an input, applies a (often nonlinear) function to it and then passes the output on to the next layer. Generally, the networks are defined to be feed-forward: a unit feeds its output to all the units on the next layer, but there is no feedback to the previous layer. Weightings are applied to the signals passing from one unit to another, and it is these weightings which are tuned in the training phase to adapt a neural network to the particular problem at hand. This is the learning phase.

We imported **neural_network.MLPClassifier()** from sklearn package in Python with **hidden_layer_size = 3** and nodes on each layer being 15, 8, 3 respectively.

4.5 Support Vector Machine

Support Vector Machine (SVM) tries to project observations to higher dimension to find a split boundary. A “kernel trick” is used to construct a distance measure in an abstract higher dimension. The concept of “margin” is used to find a more robust linear separator location. The separator is completely defined by the location of the data points on the boundary, which are called the “support vectors.”

We imported **svm** from sklearn package in Python to run SVM.

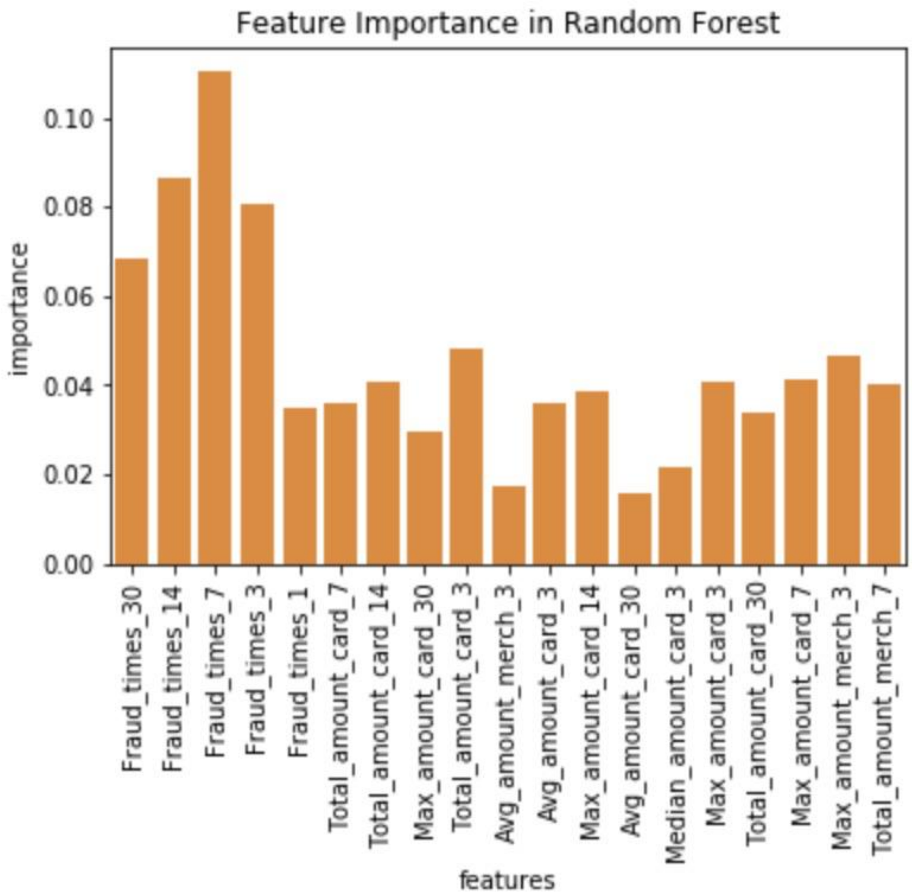
5. Results

The five candidate models we used in this project cover the linear and non-linear models with top popularity and accuracy. And the results in detecting top 2% fraud scores are listed in the below table.

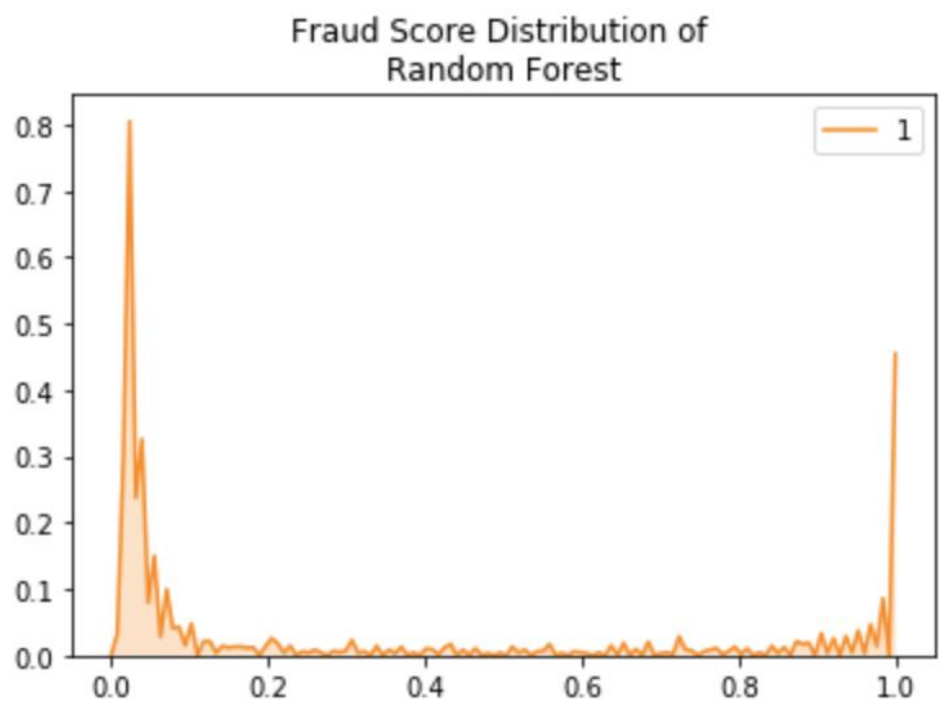
	FDR @ 2%		
Model	Train	Test	OOT
Logit	68.93	68.42	43.60
Random Forest	100.00	95.30	64.50
XGBoost	100.00	89.42	58.58
Neural Network	75.50	73.10	62.70
Support Vector Machine	63.56	62.69	50.00

Random Forest gives the most outstanding general performance and produces the best accuracy on test set, so we choose Random Forest as our best model.

Important Features selected by Random Forest are the following. In scikit-learn, it implements the feature importance calculation with consideration of "gini impurity" (or "mean decrease impurity"), which calculates each feature importance as the sum over the number of splits (across all trees) that include the feature, proportionally to the number of samples it splits. Clearly in this model, Type V and Type III variables contributes significantly.



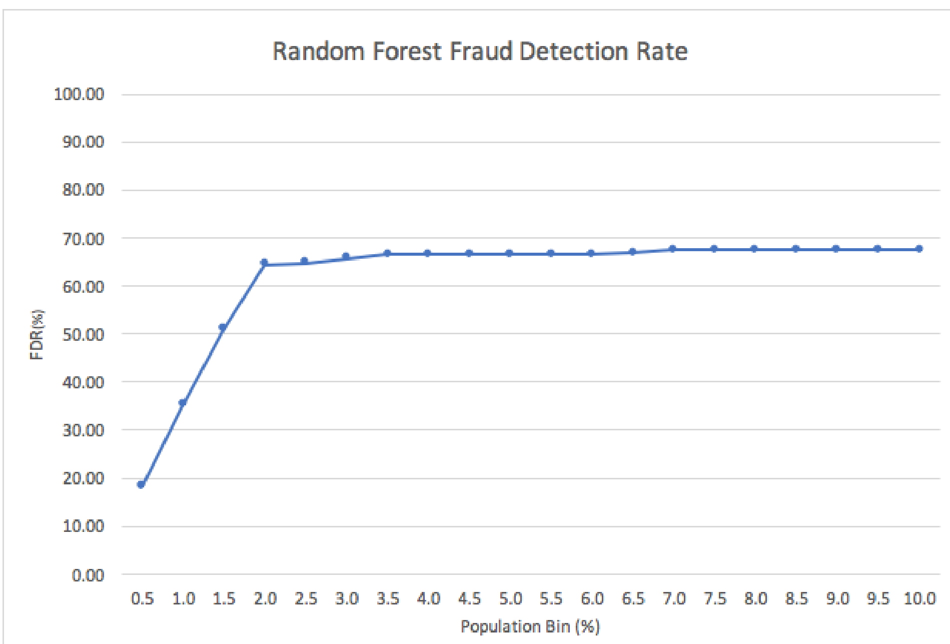
Fraud score distribution of Random Forest is the following:



Then we use **ntile()** function in R to divide the out-of-time set into 200 bins. Below are bin and cumulative statistics for both goods and bads on the out-of-time set on Random Forest model. The column of KS is the difference between the detection rate of bads and goods, indicating how well the scores of these two groups are differentiated. False positive ratio is the number of goods caught divided by the number of bads caught.

Overall bad rate: 2.686%	Bin Statistics					Cumulative Statistics					
Population bin %	# Records	# Bad	# Good	% Bad	% Good	Cumulative bad	Cumulative good	% Bad (FDR)	% Good	KS	False Pos Ratio
0.5	62	62	1	100.00	1.61	62	1	18.34	0.01	18.34	0.02
1.0	63	57	6	90.48	9.52	119	7	35.21	0.06	35.15	0.06
1.5	63	54	9	85.71	14.29	173	16	51.18	0.13	51.05	0.09
2.0	63	45	18	71.43	28.57	218	34	64.50	0.28	64.22	0.16
2.5	63	1	62	1.59	98.41	219	96	64.79	0.78	64.01	0.44
3.0	63	3	60	4.76	95.24	222	156	65.68	1.27	64.41	0.70
3.5	63	3	60	4.76	95.24	225	216	66.57	1.76	64.80	0.96
4.0	63	0	63	0.00	100.00	225	279	66.57	2.28	64.29	1.24
4.5	63	0	63	0.00	100.00	225	342	66.57	2.79	63.78	1.52
5.0	63	0	63	0.00	100.00	225	405	66.57	3.31	63.26	1.80
5.5	63	0	63	0.00	100.00	225	468	66.57	3.82	62.75	2.08
6.0	63	0	63	0.00	100.00	225	531	66.57	4.34	62.23	2.36
6.5	63	1	62	1.59	98.41	226	593	66.86	4.84	62.02	2.62
7.0	63	2	61	3.17	96.83	228	654	67.46	5.34	62.12	2.87
7.5	62	0	62	0.00	100.00	228	716	67.46	5.85	61.61	3.14
8.0	63	0	63	0.00	100.00	228	779	67.46	6.36	61.10	3.42
8.5	63	0	63	0.00	100.00	228	842	67.46	6.87	60.58	3.69
9.0	63	0	63	0.00	100.00	228	905	67.46	7.39	60.07	3.97
9.5	63	0	63	0.00	100.00	228	968	67.46	7.90	59.55	4.25
10.0	63	0	63	0.00	100.00	228	1031	67.46	8.42	59.04	4.52

The following graph represents the Fraud Detection Rate for the first 10% population bin.

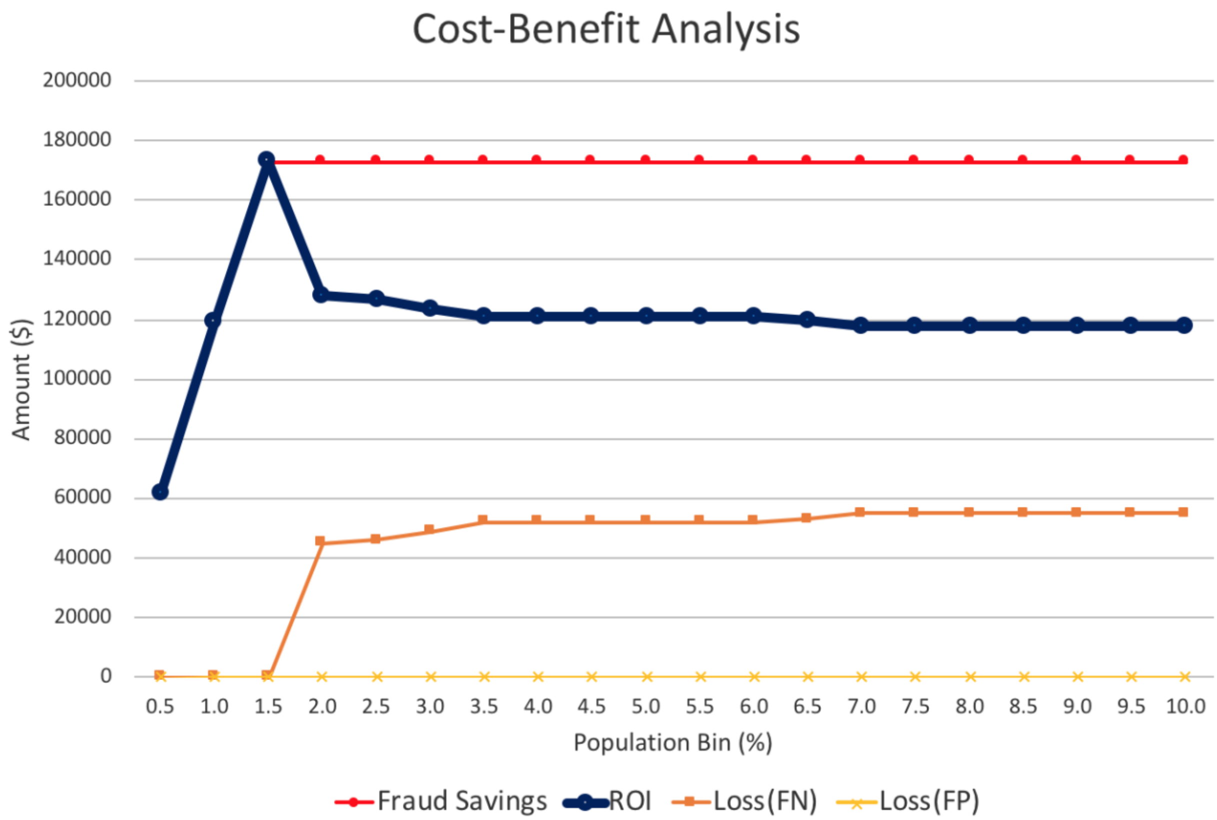


6. Cost-Benefit Analysis

It is important to incorporate economic value into model in business world. We made the following assumptions to relate return on investment to our model.

1. Assume $1000 loss for every fraud that's not caught by our model
2. Assume $10 loss for every false positive (not fraud that's flagged as fraud by our model)
3. Assume $1000 profit for every fraud that is detected by our model

Based on those assumptions, we have the following graph to show the relationship among fraud savings from True Positives(TP), loss from False Negatives(FN) and False Positives(FP), and net return on investment(ROI) in a cumulative manner.



From the above graph, we can easily tell that the ROI curve reaches the peak in the first three bins and drops dramatically in the fourth. Then the curve remains steady at around $ 120,000.

The increase in ROI is consistent with fraud savings from TP, indicating that our model predicts frauds correctly and saves money for the company. The decrease in ROI after 1.5% location is the result of cumulative FN and FP, among which missing catching actual frauds contributes the prevailing proportion. And the decrease happens dramatically from 1.5% to 2% and remains gradually after 3.5%. Eventually, ROI stays at $172,840 from 7% on.

7. Conclusion and Recommendation

In this project, we picked Random Forest model as our best model, achieving 64.54% of Fraud Detection Rate (FDR) at 2% location. In the cost-benefit analysis, this model leads to a net return of $172,840 when binning at 7%.

To further improve credit card detection, we mainly consider from the following two aspects.

1. Dependent variable:

As indicated by feature importance plot, those top expert variables we created are linked by Fraud times/ Tot amount/ Avg amount. In the future data collection part, we may pay more attention to collect these super predictive features for classification. In addition, variables capturing interactive effects could also be considered as Fraud_times_n. For example, future researchers can consider count how many times of the card transaction in a 30 min period.

2. Independent variable:

It is also a good idea to transform the two-class classification into a more labeled(very risky / risky / good) on based on different risk tolerances. It functions practically that allows decision makers a range of different actions.

8. Appendix: Data Quality Report

8.1 Data Overview

The credit card transaction data contains 96,708 records of transaction information. It includes information such as the card number, the date of the transaction, the merchant number, the merchant description, the merchant state, the merchant zip code, transaction types, and transaction amount. The last field is used to indicate whether the record is fraudulent or not.

File Name: card transactions.csv
Data Source: This dataset is partly simulated based on real card transaction records
Data Size: 96,708 records
Number of Fields: 9 (excluding the first "Recordnum" field as it can be considered as index column)
Time: Jan 1st, 2010 - Dec 31st, 2010

8.2 Descriptive Statistics of the Data

For the 9 variables in the dataset, statistics are summarized including count, number of unique values, percentage of populated records, percentage of unique values and percentage of repetitions for that variable. A summary table is provided below to demonstrate those characteristics.

Features	Count	Populated%	Unique	Unique %	Repetitive %
Cardnum	96708	100.00%	1644	1.70%	98.30%
Date	96708	100.00%	365	0.38%	99.62%
Merchant Number	93333	96.51%	13091	13.54%	86.46%
Merch Description	96708	100.00%	13125	13.57%	86.43%
Merchant State	95513	98.76%	227	0.23%	99.77%
Merchant Zip	92052	95.19%	4568	4.72%	95.28%
Transaction Type	96708	100.00%	4	Na	Na
Amount	96708	100.00%	34876	36.06%	63.94%
Fraud	96708	100.00%	2	Na	Na

As “amount” is a numerical variable, we also provide mathematical statistics such as mean, median (which is the 50th percentile), standard deviation, minimum, maximum, the first and third percentile for that field.

	Amount
Count	96708
Mean	427.865
Std	10008.47
Min	0.01
25%	33.45
50%	137.9
75%	427.715
Max	3102046

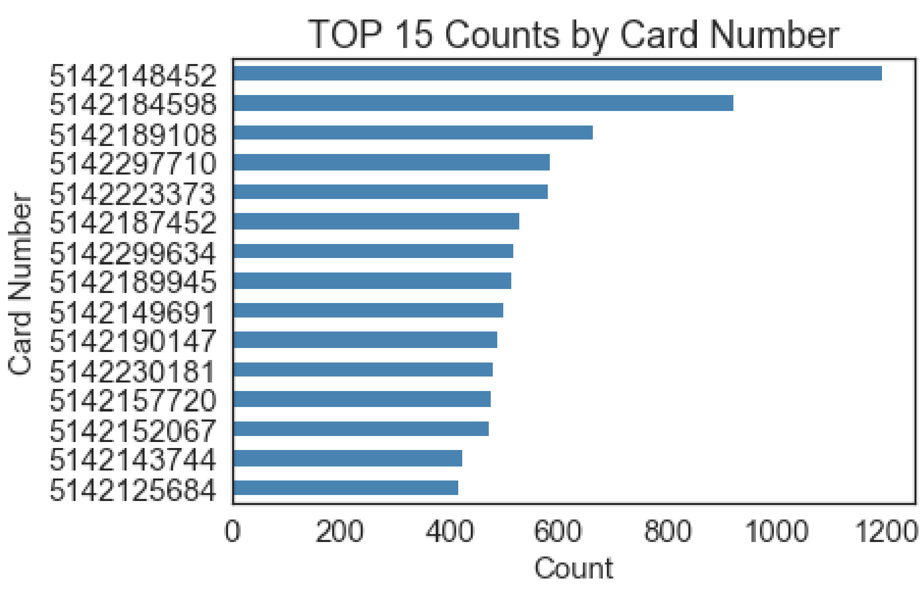
8.3 Information for Each Field

Below is the general information for the 9 fields of the dataset. Each field is demonstrated by "Field Name", "Type", "Descriptions" and "Populated %". The record of top frequency is also provided along with each field.

Field Name	**Type**	**Description**	**Populated %**
Cardnum	Categorical	The credit card number used for the transaction	100.00%

Record of Top Frequency: 5142148452 (1192 records)

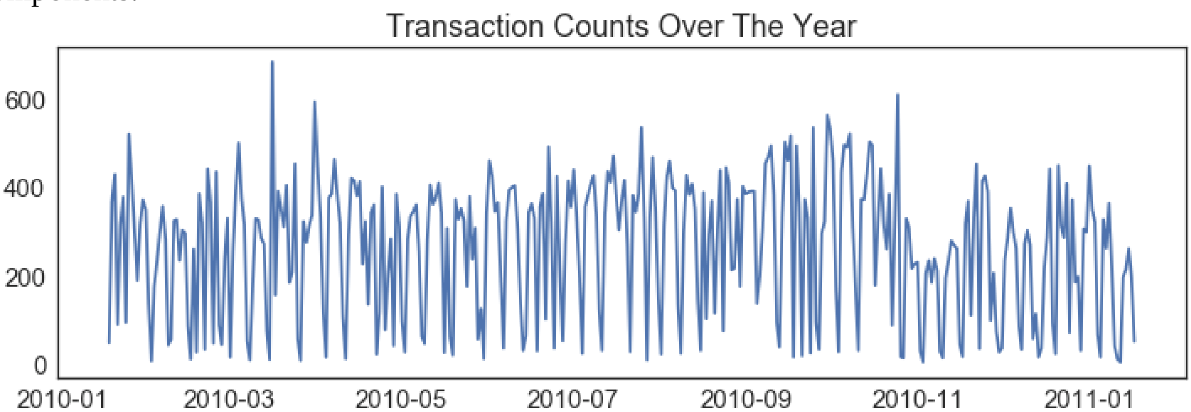
The field is 100% populated with 1,644 unique entities, indicating that this data involves transaction information from 1,644 credit cards. The top 15 frequently reported card number records are shown below.



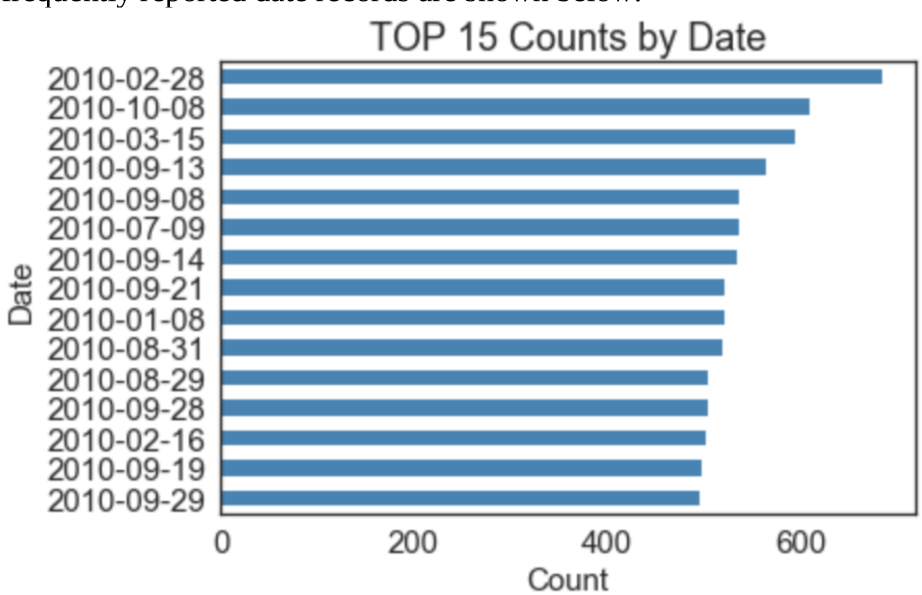
Field Name	Type	Description	Populated %
Date	Categorical	The date that the transaction take places	100.00%

Record of Top Frequency: 28/02/2010 (1478 records)

This field is 100% populated with 365 unique values, i.e. every day in year 2010. Below is the distribution of transactions over the whole year, demonstrating clear weekly cyclical components.



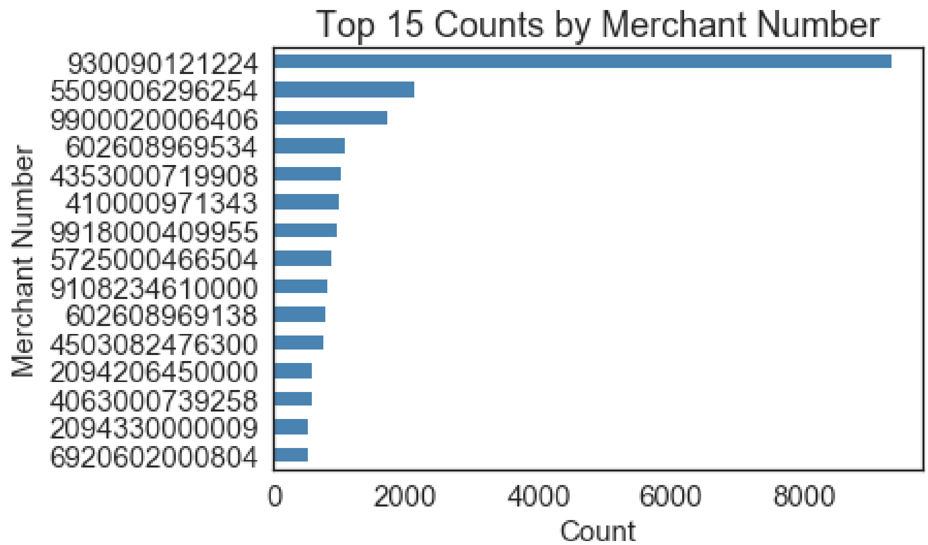
The top 15 frequently reported date records are shown below.



Field Name	Type	Description	Populated %
Merchantnum	Categorical	The merchant's number recorded in each transaction	96.51%

Record of Top Frequency: 930090121224 (9310 records)

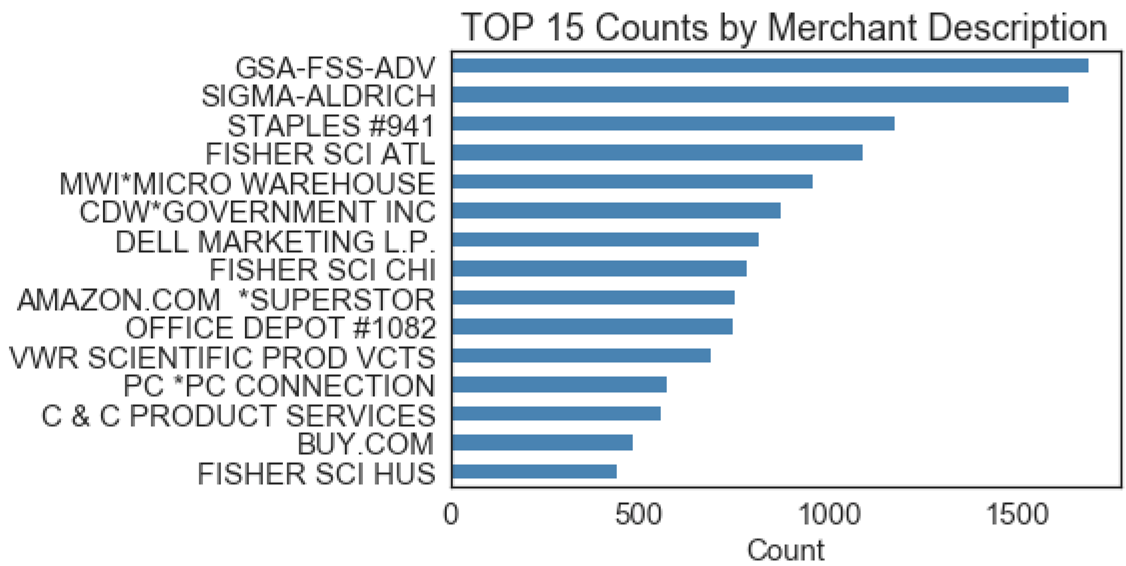
This field is 96.51% populated with 13,091 unique values. The top 15 frequently reported merchant number records are shown below. The top record is linked to FEDEX, which is unlikely to indicate a frivolous record and suggested to keep for further analysis.



Field Name	Type	Description	Populated %
Merch Description	Categorical	The information about the merchant. Some may include transaction explanation such as "FEDEX SHP 12/23/09 AB#" (name, prior date, category, etc.)	100.00%

Record of Top Frequency: GSA-FSS-ADV (1,688 records)

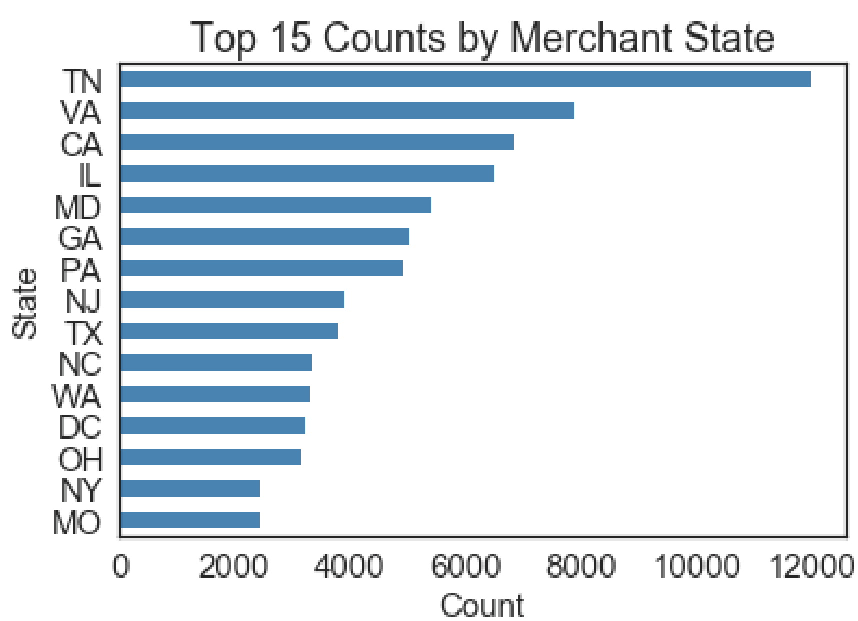
This field is 100% populated with 13,125 unique values, which slightly mismatched with the above merchant number. This is because one merchant number may link to more than one merchant description. For example, FEDEX (merchant number: 930090121224) includes plenty of dates in the description. The top 15 frequently reported merchant description records are shown below.



Field Name	Type	Description	Populated %
Merchant State	Categorical	The abbreviations of the merchant's states	98.76%

Record of Top Frequency: TN (11,990 records)

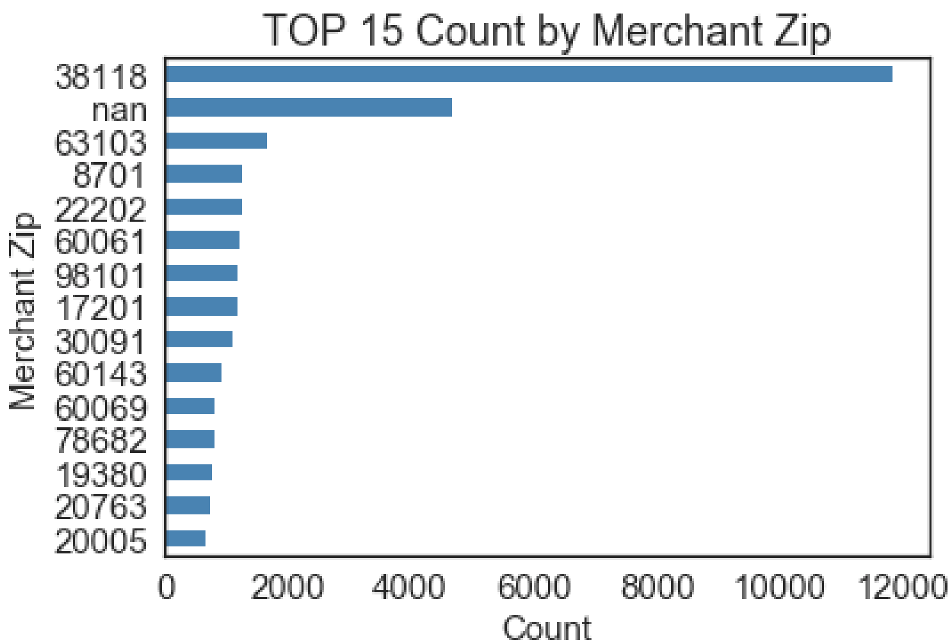
This field is 98.76% populated with 227 unique states, which are way more than the expected number of states in USA. This is probably due to the remote states out of the U.S. border or input errors. The top 15 frequently reported states records are shown below.



Field Name	Type	Description	Populated %
Merchant Zip	Categorical	The zip code of the merchant area	95.19%

Record of Top Frequency: 38118 (11,823 records)

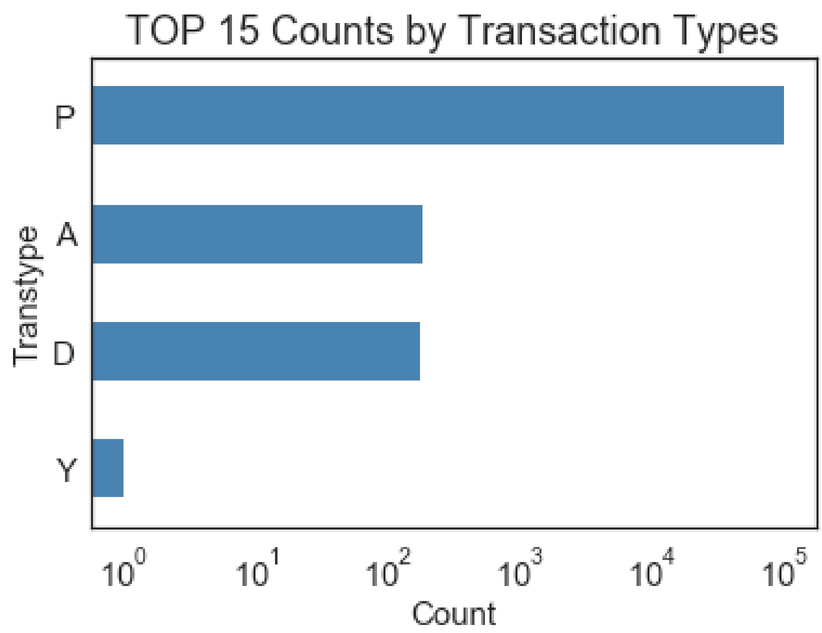
This field is 98.76% populated with 4,568 unique zip codes. The top 15 frequently reported address records are shown below, including the “nan” records (nearly 5,000 counts).



Field Name	Type	Description	Populated %
Transtype	Categorical	The type of transaction	100.00%

Record of Top Frequency: P (96,353 records)

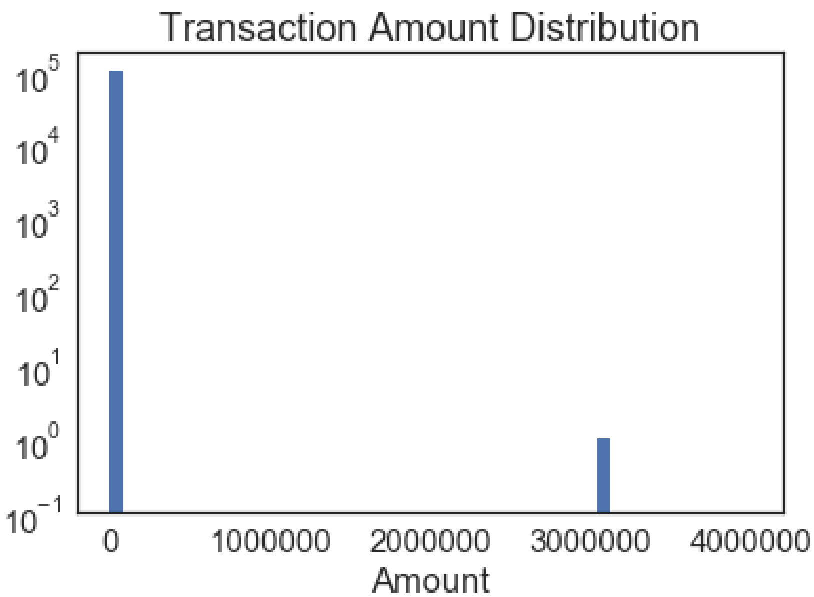
This field is 100% populated with four different types. P (approved) is the most reported class with 96,353 records (99.63%), representing approved transactions. The distributions of the four types are shown below.



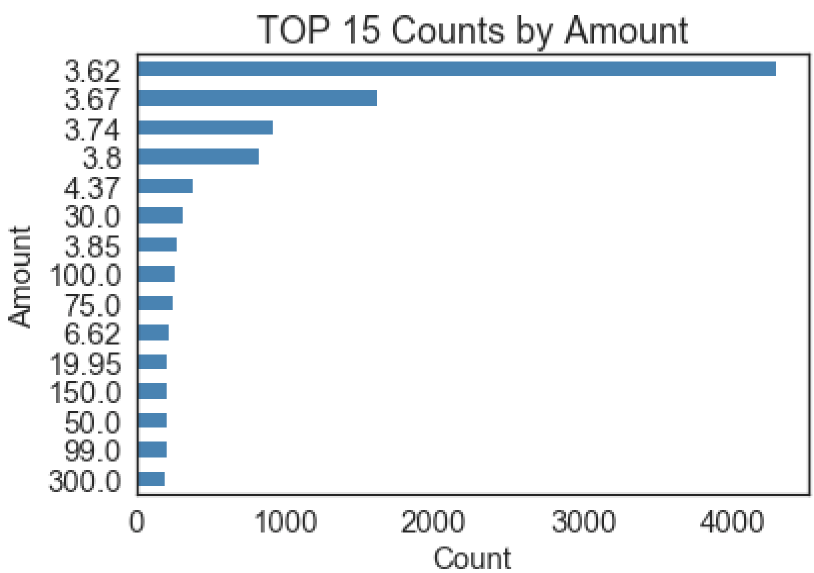
Field Name	**Type**	**Description**	**Populated %**
Amount	Numerical	The amount of each transaction	100.00%

Record of Top Frequency: $3.62 (4283 records)

This field is 100% populated with 34,876 unique values. The most extreme amount is more than $3,000,000, which may indicate a currency error. This outlier amount need to be paid special attention in the next step.



The top 15 frequently reported amount of birth records are shown below.



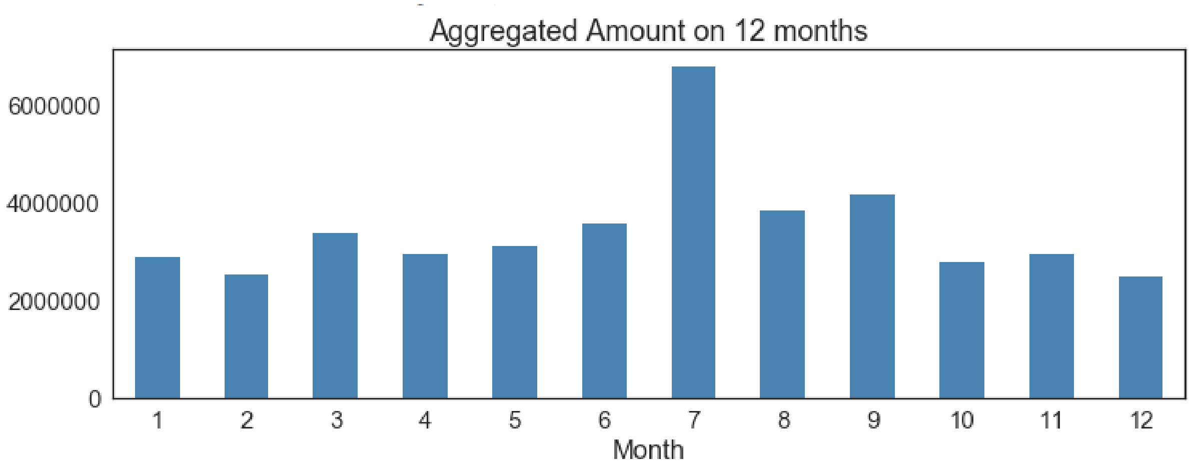
Field Name	Type	Description	Populated %	Non-Fraud	Fraud
Fraud	Categorical	If this transaction is fraud	100.00%	98.95%	1.05%

<u>Counts of Two Class</u>: 0 (Non-Fraud, 95694 records), 1 (Fraud, 1014 records)

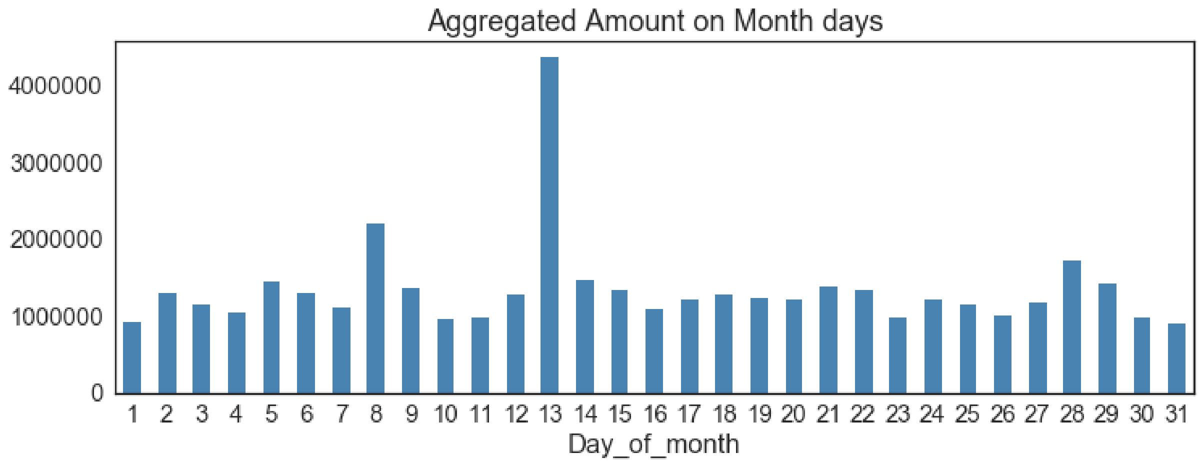
This label field is 100% populated with two unique categories. 0 and 1 mean non-fraud and fraud transactions, respectively. The low percentage (1.05%) of fraud records indicates a largely imbalanced data.

Below are three bar charts that describe the amount distributions on different seasonal levels (12 month, 31 days in month and 7 days in week)

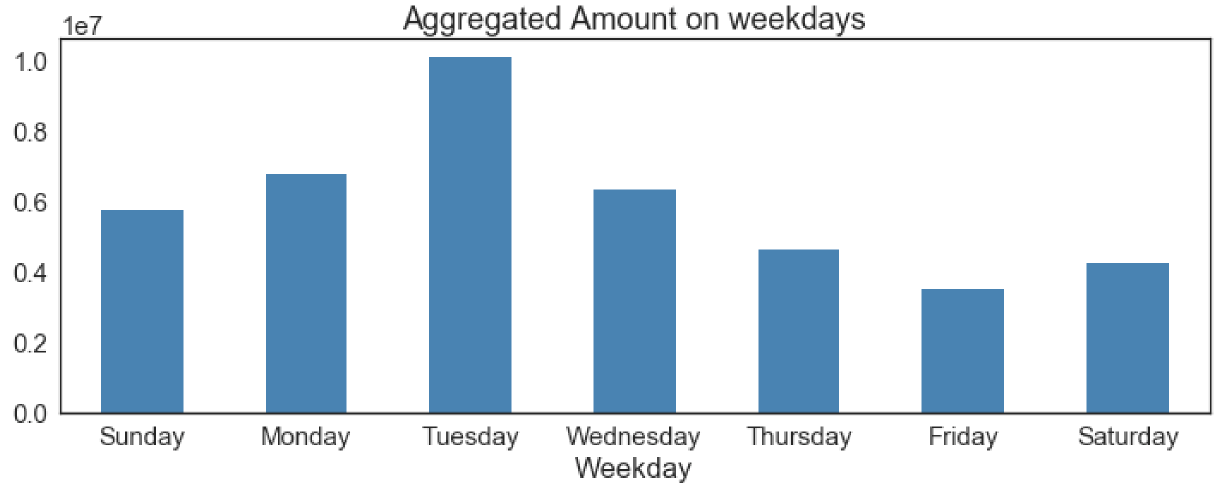
- *Amount Distribution in 12 Months*



- *Amount Distribution for Each Day over 12 Months*



- *Amount Distribution during Weekday*



9. Reference

[1] https://www.gaslampmedia.com/download-zip-code-latitude-longitude-city-state-county-csv/