Credit Card Fraud Detection

Chenbo Yang, 1041895189

Yi Liu, 7717540026

Sen Zhang, 6612445912

Yi Liu, 8666552780

Kai Sun, 7191075100

Fandi Ma, 3652892113

Date: 5/4/2022

Table of Contents

Executive Summary

Transaction fraud, or sometimes called payment fraud occurs when a stolen payment card or data is used to generate an unauthorized transaction. In 2018, 24.26 billion dollars were lost due to transaction fraud worldwide. The US reports the largest share of credit card fraud losses in 2018, which is 38.6%. While consumers are shielded from the cost of transaction fraud, financial institutions are left to pay for most of the losses, which is about 72%, while merchants and ATM acquirers assume the rest 28%. Hence, there is a strong motivation for financial institutions to develop reliable models to detect transaction frauds.

In this project, our team used a data file which contains 96,753 credit card transaction records from a US government organization to build a supervised machine learning algorithm that detects transaction fraud. We have six people in the team. We began working on the project on March 24th, presented our findings on April 28th, and completed the project on May 5th.

We took five steps to complete the project. In the first step, we explored the data ('card transactions.csv') and built Data Quality Reports (DQR). In the second step, we constructed as many candidate variables as we could. We ended up with 560 variables. In the third step, we performed feature selections using a filter and a wrapper, discarding most of the variables created in the previous step and keeping only the 20 most important candidate variables. In the last step before preparing for the presentation and the report, we built dozens of supervised machine learning models using different algorithms and hyperparameters. We measured model performances using average Fraud Detection Rate (FDR) at 3% of Training, Testing, and Out of Time (OOT) data. In the last step, we chose a champion algorithm with a fixed tuning of hyperparameters and used all available data to train a final model. Our champion model is Multi-layer Perceptron classifier (MLPClassifier) with default hyperparameters except hidden_layer_sizes is set at (50,). The Fraud Detection Rates (FDR) at the 3% threshold for Training, Testing, and Out-of-Time (OOT) data are 0.752, 0.731, and 0.575 respectively.

This report will follow the steps we took in the project. First, we will give a description of data (the full DQR will be included in the appendix). Then, we will introduce our data cleaning process. After that, we will describe the process of feature creation (the full list of variables created will be included in the appendix). Next, we will discuss our feature selection process, the algorithms we tried, and the results of each algorithm. Lastly, we will present a conclusion summarizing the entire project with suggestions for future projects.

Description of Data

Overall Description

This dataset contains actual credit card purchase records in 2016, from a US government organization. It also contains fraud identification, enabling us to train supervised machine learning models to distinguish fraud records. Only 1,059 frauds are in the table. There are 10 attributes and 96,753 records are available in the dataset.

Field Name	%Populated	Min	Max	Mean	Stdev	%Zero
Date	100	2016-01-01	2016-12-31	N/A	N/A	0
Amount	100	0.01	3,102,045.53	427.89	10,006.14	0

Table 1 Summary of Numeric Fields

Field Name	% Populated	# Unique Values	Most Common Value
Merchnum	96.51	13,091	930,090,121,224
Merch description	100	13,126	GSA-FSS-ADV
Merch state	98.76	227	TN
Transtype	100	4	P
Cardnum	100	1,645	5,142,148,452
Merch zip	95.19	4,567	38,118
Fraud	100	2	0

Table 2 Summary of Categorical Fields

Descriptions of Fields

Cardnum

‘Cardnum’ denotes the card numbers related to some purchasing histories. There are 1,645 unique credit card numbers for these fields, without missing values. Card number ‘5,142,148,452’ has the highest frequency. Following are the distributions of the top 15 categories.

	Cardnum
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

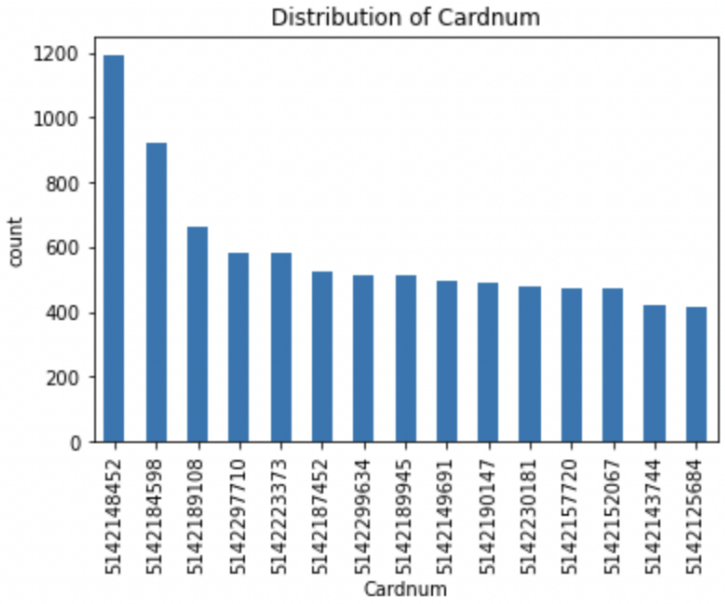

Figure 1 Distribution of ‘Cardnum’ variable

Merchnum

Merchnum represents the number of certain products. There are 13,091 unique numbers for this field with 3,375 missing values (3.5%). The most frequently bought is the product of ‘930,090,121,224’. Following is the distribution of the top 15 categories.

	Merchnum
930090121224	9310
5509006296254	2131
9900020006406	1714
602608969534	1092
4353000719908	1020
410000971343	982
9918000409955	956
5725000466504	872
9108234610000	817
602608969138	783
4503082476300	746
2094206450000	590
4063000739258	568
2094330000009	533
6920602000804	523

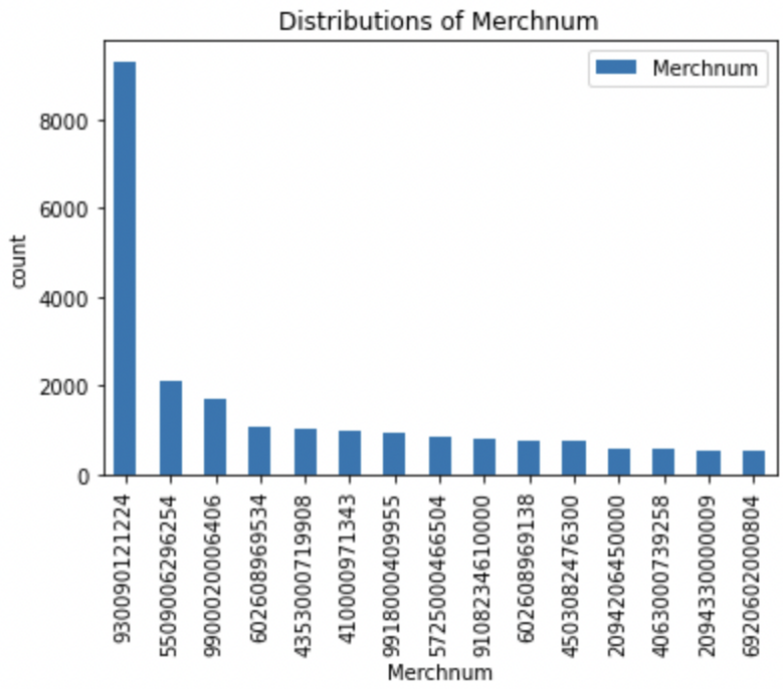

Figure 2 Distribution of 'Merchnum' variable

Merch description

Merch description denotes the name of the product. There are 13,126 unique descriptions for this field, without missing/null values. The most frequent category is 'GSA-FSS-ADV', appearing 1,688 times.

	Merch description
GSA-FSS-ADV	1688
SIGMA-ALDRICH	1635
STAPLES #941	1174
FISHER SCI ATL	1093
MWI*MICRO WAREHOUSE	958
CDW*GOVERNMENT INC	872
DELL MARKETING L.P.	816
FISHER SCI CHI	783
AMAZON.COM *SUPERSTOR	750
OFFICE DEPOT #1082	748
VWR SCIENTIFIC PROD VCTS	688
PC *PC CONNECTION	570
C & C PRODUCT SERVICES	558
BUY.COM	481
FISHER SCI HUS	442

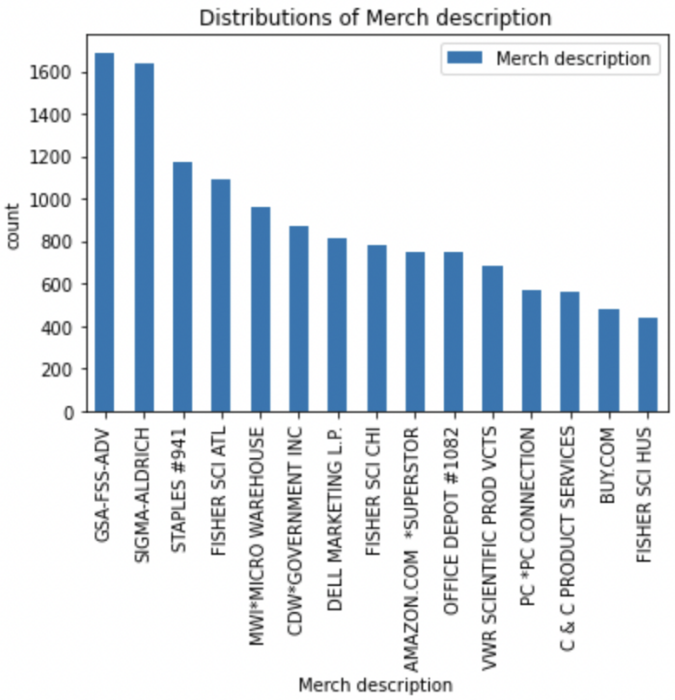

Figure 3 Distribution of 'Merch description' variable

Merch zip

This categorical variable denotes the zip code of the merchant. There are 4,567 unique values for this field with 4,656 missing values (4.8%). Following is the distribution of the top 15 categories.

	Merch zip
38118.0	11868
63103.0	1650
8701.0	1267
22202.0	1250
60061.0	1221
98101.0	1197
17201.0	1180
30091.0	1092
60143.0	942
60069.0	826
78682.0	817
19380.0	769
20763.0	749
20005.0	648
20748.0	592

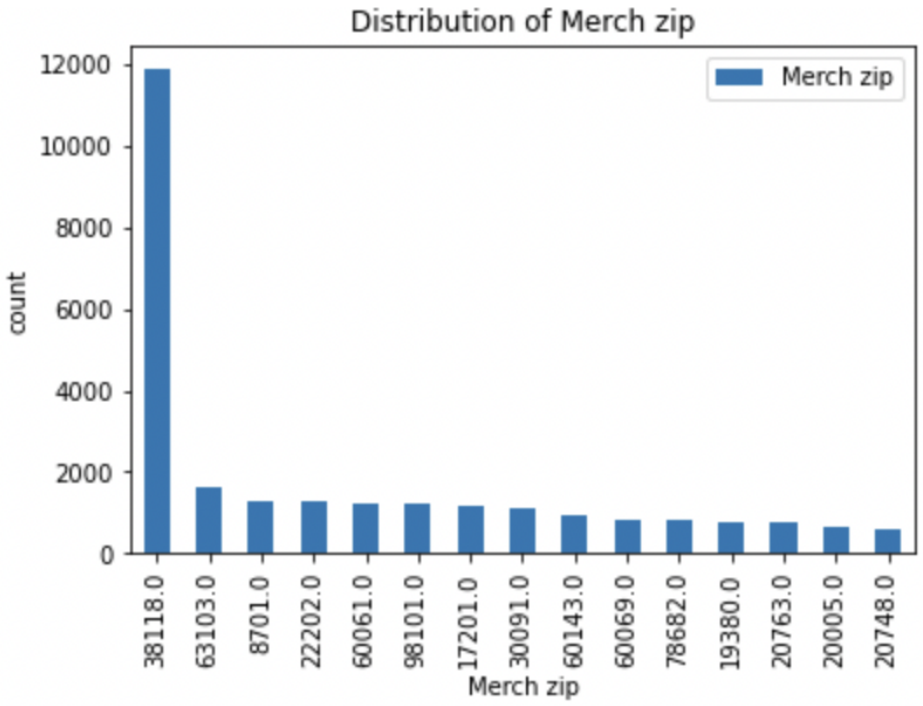

Figure 4 Distribution of 'Merch zip' variable

Amount

This numerical variable represents the amount of each transaction. There are no missing/null values. Following graph shows the distribution of the ‘Amount’ variable.

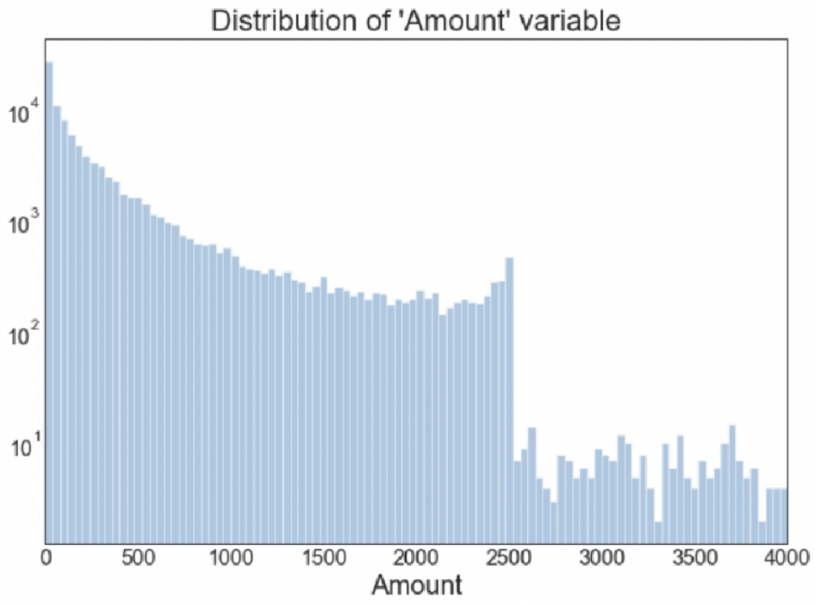

Figure 5 Distribution of ‘Amount’ variable

Fraud

This variable is the target feature in this dataset, indicating if the transaction is fraud or not. 1 denotes that the transaction is fraudulent; 0 indicates that the transaction is not fraudulent. There are two unique values for this field without missing values. Following is the distribution of the two categories.

	Fraud
0	95694
1	1059

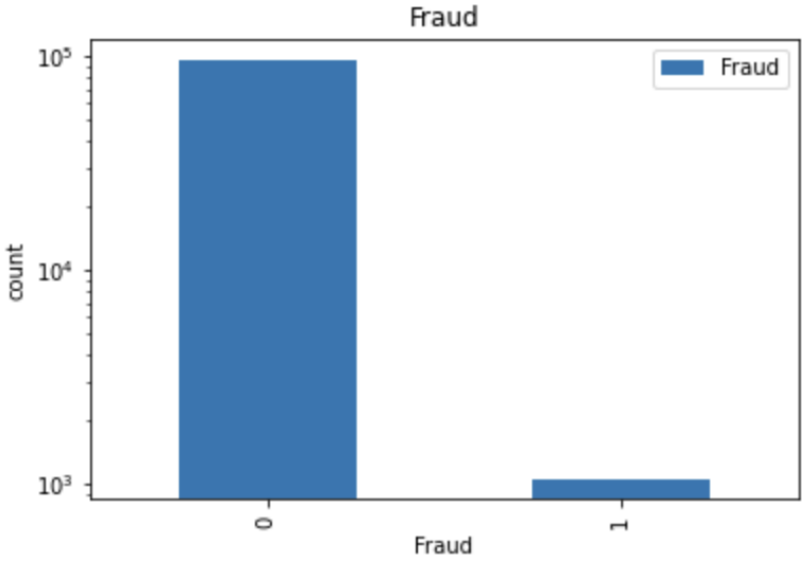

Figure 6 Distribution of ‘Fraud’ variable

Data Cleaning

Removing outliers and bad records

We will only focus on the transaction type records, which is indicated by “P”, so we kept records with transaction type “P”. Also, we found one outlier with an unusual amount of 3,102,045.53, and we deleted the record. Therefore , we ended up with 96,397 records.

Filling in missing values

Merchnum

- Replaced “0” with NaN
- Filled in with mode of Merch description
- Filled remained missing values in with “Unk” (unknown)

Merch zip

- Matched missing values with records having the same mode of merchant number.
- Filled remained missing values in with “Unk” (unknown)

Merch state

- Matched missing values with records having the same zip number.
- For records with merch zip in the range of 00600-00799 and 00900-00999 , we set ‘PR’ (Puerto Rico) as the merch state.
- Manually filled in some missing state based on the zip code.
- Then, we matched again with records having the same Merchnum and Merch description.
- The remaining missing values records were recorded as “Unk”(unknown)

Combining Related Variables

After filling the missing values, we combined related variables to construct expert attributes. For instance, we combined ‘Cardnum’ and ‘Merchnum’ into one ‘card_merch' variable. We now have 19 variables in total, which can be seen below.

No.	variable	combination	description

1	Recnum	-	index of record
2	Cardnum	-	card number
3	Date	-	date of transaction
4	Merchnum	-	number of merchant
5	Merch description	-	-
6	Merch state	-	state where merchants locate
7	Merch zip	-	zip code of merchant
8	Transtype	-	type of transaction
9	Amount	-	transaction amount
10	Fraud	-	-
11	card_merch	Cardnum + Merchnum	card at that merchant
12	card_des	Cardnum + Merch description	card with that merchant description
13	card_state	Cardnum + Merch state	card in this state
14	card_zip	Cardnum + Merch zip	card in this zip code
15	merch_des	Merchnum + Merch description	merchant with this description
16	merch_state	Merchnum + Merch state	merchant in this state
17	merch_zip	Merchnum + Merch zip	merchant in this zip code
18	des_state	Merch description + Merch state	merchant description in this state

19	des_zip	Merch description + Merch zip	merchant description in this zip code

Table 3 Variables including the combined variables

Further variable creation

We need to understand how fraud could happen so that we can explore the relationship between fraud(the dependent variable) and the responsive variables. As is known, fraud transactions always come with signals, for example:

- Burst of transactions of a card at different merchants
- Larger than normal purchase amounts
- Using the card at merchants not used before, at a very different geography, or at a high-risk merchant
- Infrequent recurring charges with the same amount or at the same merchant
- Increased usage in the card that no longer presents
- Fictitious merchant or transactions invented by employee or merchants

We can see that fraud signals could possibly be detected from the amount, time interval, frequency, velocity, corresponding likelihood, etc, and we want to capture such signals for fraud prediction. However, the 19 variables that we already have are not enough, so we created 540 more variables including 1 target encoded variable, 2 Benford's Law variables, as well as several days since, amount, frequency, and velocity variables based on above signals.

Target-encoded Variables - 1

We replaced a categorical field with a target encoded variable, the value of which is the average of the dependent variable for all the records in that category. This variable measures the likelihood of fraud on certain days of the week (from Monday to Sunday).

However, target-encoding may cause the loss of some interaction information and has the risk of overfitting. Therefore, we only used training data when creating the variables and we also applied a smoothing formula.

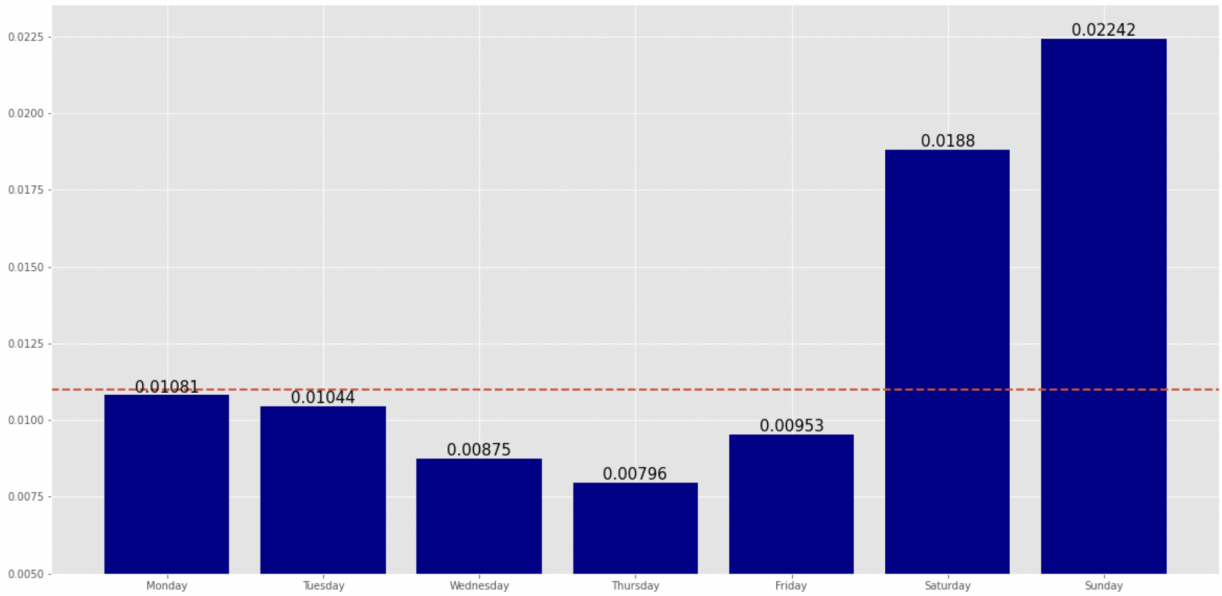

Figure 7 Risk table of 'day of the week'

Amount Variables - 432

We created 432 variables based on the transaction amount. At first, we aggregated by card number with merchant description, merchant number, zip code and state, merchant number with description and state, and merchant description with state, zip code and nothing, to respectively to calculate the mean, maximum, median and sum of amount over six-time windows ([0,1,3,7,14,30] days) using roll functions. This results in 216 variables. Further, we also created variables by using the actual amount of each entity or combination group divided by the outcome we calculated for the entity or group resulting in the remaining 216 variables. Hence, we ended up with 432 variables, as pictured below.

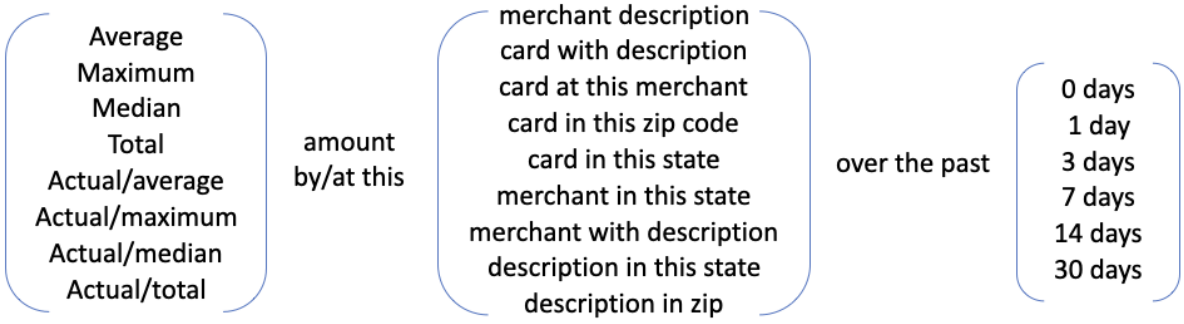

Figure 8 Formulation of amount variables

Frequency Variables - 54

We then focused on the calculation of how many of the 9 entities occur over six-time windows ([0,1,3,7,14,30] days). This results in 54 variables, as pictured below.

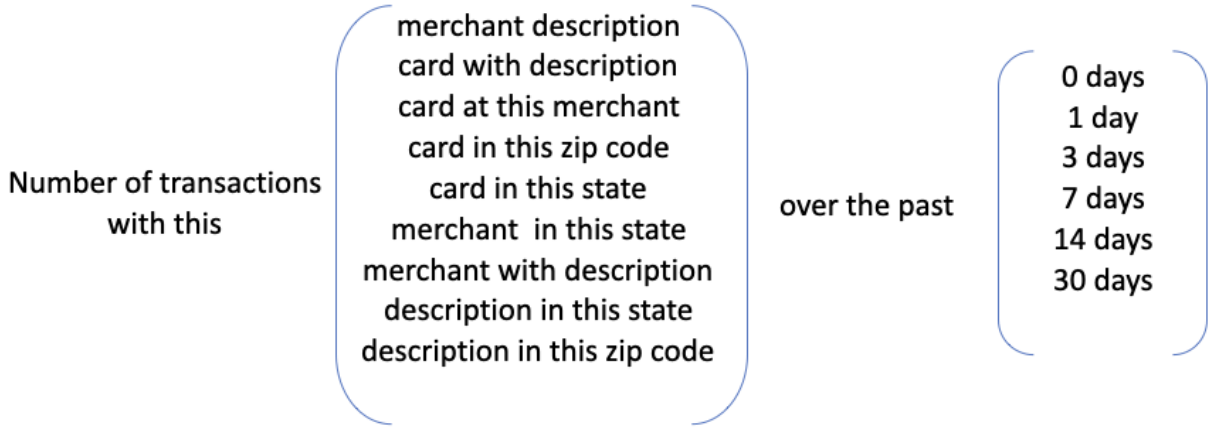

Figure 9 Formulation of frequency variables

Days Since Variables - 9

We calculated the number of days since we last saw a specific combination group or entity. For each attribute, we created one 'Day since' variable. Overall, we created **9** 'Days Since' variables. For instance, 'merchant_description_ day_since' indicates how many days since a transaction is related to a unique merchant description.

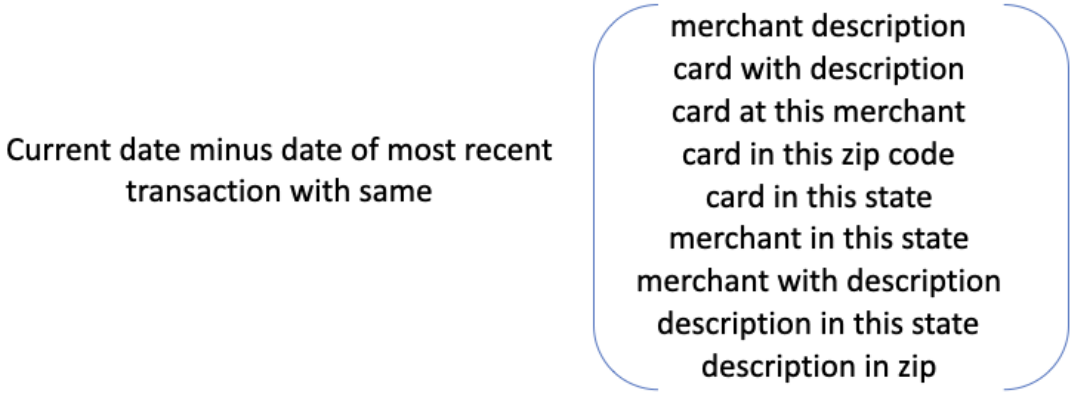

Figure 10 Formulation of day-since variables

Velocity Variables - 42

Velocity is basically the frequency velocity. It is the number of transactions with the same card or merchant over the past 0 and 1 day, divided by the average number of the transactions of the same card or merchant over the past 3, 7, 14 and 30 days. We totally created **42** velocity variables.

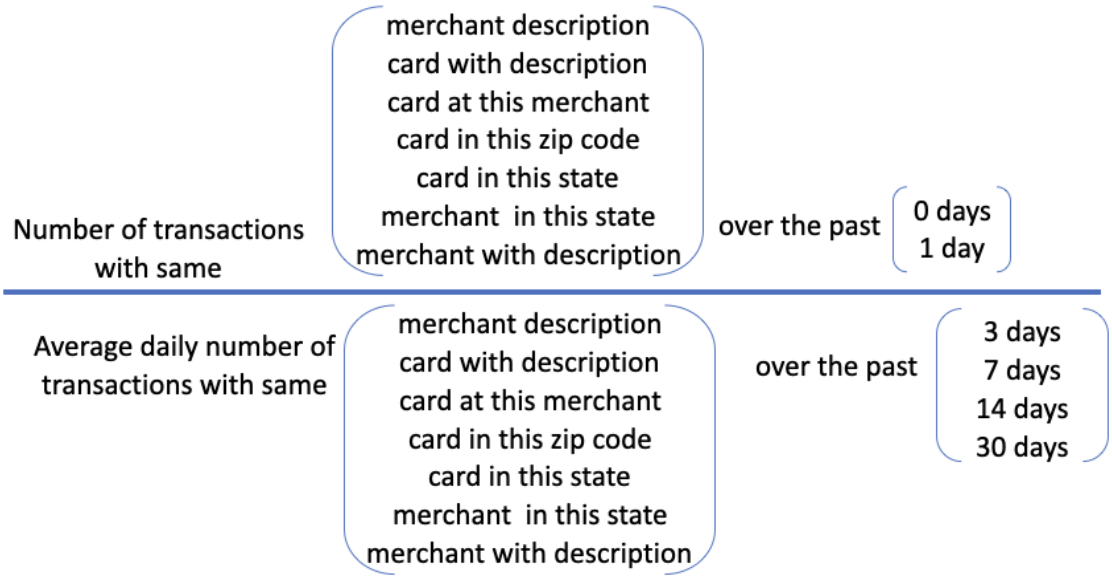

Figure 11 Formulation of velocity variables

Benford's Law Variables - 2

According to Benford's law, the first digit of many measurements is **not** uniformly distributed, and those that begin with "1" or "2" always appear around 47.7% of the time. We checked how the actual amount distributions of each card and merchant are different from the Benford's law distribution, and marked records with substantial differences as unusual records. Thus, we created two benford's law variables, one for cardnum and one for merchant number.

Note: Since records of Fedex are not in compliance with Benford's law, we eliminated records of Fedex when generating Benford's law variables, and then took the average to impute the value of Benford's law variables for Fedex related records.

We created **540** variables in total. The list of variables is in the appendix.

Feature Selection

After variable creation, our dataset has a size of 96,397* 540, which is high in dimensionality. Training machine learning models using such high dimensional data may lead to a high number of outliers, increased training time, and higher possibilities of overfitting. Therefore, we need to reduce the dimensionality of features by applying feature selection methods. After filtering out the inconsequential features, the models can better fit the underlying patterns and have a better performance.

In the feature selection process , the first two weeks of records are dropped to avoid bias since they are not fully formed. The last 2 months' records are treated as out of time data (OOT), We used the remaining as modeling data, which is the input for feature selection to avoid overfitting.

Filter

Filter methods measure the relevance of features using their correlations to the dependent variable. Compared to wrapper methods, the filter methods are much faster since they do not involve training processes. There are some common filter methods for binary classification problems: Pearson Correlation, Kolmogorov-Smirnov (KS), Fraud Detection Rate (FDR), and Information Value.

Among the methods above, Univariate KS and the FDR@3% are chosen as our filter methods to rank the variables. In the end, top 80 variables were selected according to the average ranking of these two scores.

Univariate Kolmogorov-Smirnov (KS)

The Univariate KS is a statistical measure of how well the ‘bad’ records distribution and ‘good’ records distribution are separated. As the figure 12 shows, the more different the curves, the better suited the variable is for separating, and the variable is more important.

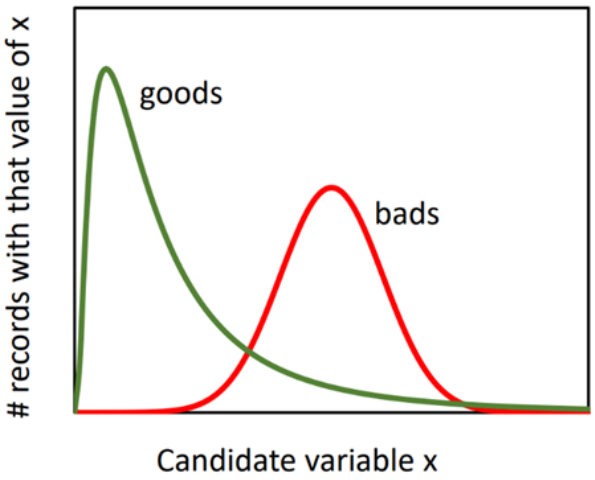

Figure 12

The KS score is the maximum of the difference cumulative and could be calculated using the formula below:

$$KS = \max_{x} \int_{x_{min}}^{x} [P_{\text{goods}} - P_{\text{bads}}]\, dx$$

The table 4 shows the top 30 variables after applying a KS filter.

	Variable	Score
1	card_zip_total_7	0.68474
2	card_merch_total_7	0.68108
3	card_zip_total_3	0.67768
4	card_merch_total_3	0.67509
5	card_merch_total_14	0.67497
6	card_state_total_3	0.67420
7	card_zip_total_14	0.67229
8	card_des_total_7	0.67127
9	card_state_total_7	0.66970
10	card_state_total_14	0.66894
11	card_des_total_14	0.66554
12	card_des_total_3	0.66133
13	card_zip_total_1	0.66058
14	card_des_max_14	0.65917

15	card_state_total_1	0.65914
16	card_merch_total_30	0.65836
17	card_merch_total_1	0.65822
18	card_zip_max_14	0.65790
19	card_zip_max_7	0.65752
20	card_des_total_30	0.65684
21	card_des_max_7	0.65675
22	card_zip_total_30	0.65671
23	card_des_max_30	0.65450
24	card_merch_max_14	0.65419
25	card_zip_max_30	0.65139
26	card_merch_max_7	0.65062
27	card_des_max_3	0.65017
28	card_zip_max_3	0.64980
29	card_merch_max_30	0.64974
30	card_state_max_3	0.64800

Table 4

Univariate Fraud Detection Rate (FDR) @ 3%

FDR is another common filter metric used in fraud detections. It is calculated as the number of true frauds caught by the model divided by the total number of true frauds in the entire dataset. FDR measures how many frauds we can catch within a certain population. For instance, FDR 50% at 3% indicates the model catches 50% of the frauds in the top 3% of the population.

Since FDR requires a classification model to give probability prediction, it cannot be used as a filter directly. However, a univariate FDR can be used by taking the values of each record as its probability. The univariate FDR measures whether frauds tend to cluster at one end of the distribution for one feature. It generally performs well in selecting useful features in fraud detections. We applied FDR at a 3% cutoff as a filter metrics. The final rank of each variable is the average rank of KS and FDR@3%, which is calculated using the formula below:

$$Average\ Rank = (KS\ Rank + FDR\ Rank)\ /2$$

We kept the top 80 variables and moved to wrapper.

Wrapper

Compared to filter methods, wrapper always runs slower but it can take the usefulness of a subset of the features into consideration. Basically, wrapper methods can find out the best subset of features by training a model on them. A wrapper method has a model "wrapped" around the process, and the model could be anything. Generally, there are three types of wrapper methods: Forward Selection, Backward Selection, and General Stepwise Selection. In Forward Selection, we start with zero features and add one or several best features each time; whereas in Backward Selection, one or several worst features are removed from the feature set. In this project, we used the Forward Selection method combined with a random forest classifier to select the top 20 variables from 80 candidates.

Forward Selection

Sequential Forward Selection (SFS) tries to find the "best" feature subset by iteratively selecting features based on classifier performance. This method can remove correlations but does not guarantee to find the global optimum. We started with zero feature and added one feature at a time in each round. Each feature is selected from the pool of all features that is not in our feature subset. It is the feature that – when added – results in the best classifier performance. Then, the greedy search looks for the next best position given the current location and repeats this step until there is no significant improvement.

Random Forest Classifier

A random forest model with five trees is used as the wrapper in this project. Combining SFS and Random Forest classifier, we could figure out the importance of variables in sequence. After wrapper, 20 variables were selected, ready to be used for modeling.

Figure 13 shows how the performance changes as the number of variables increases during forward selection:

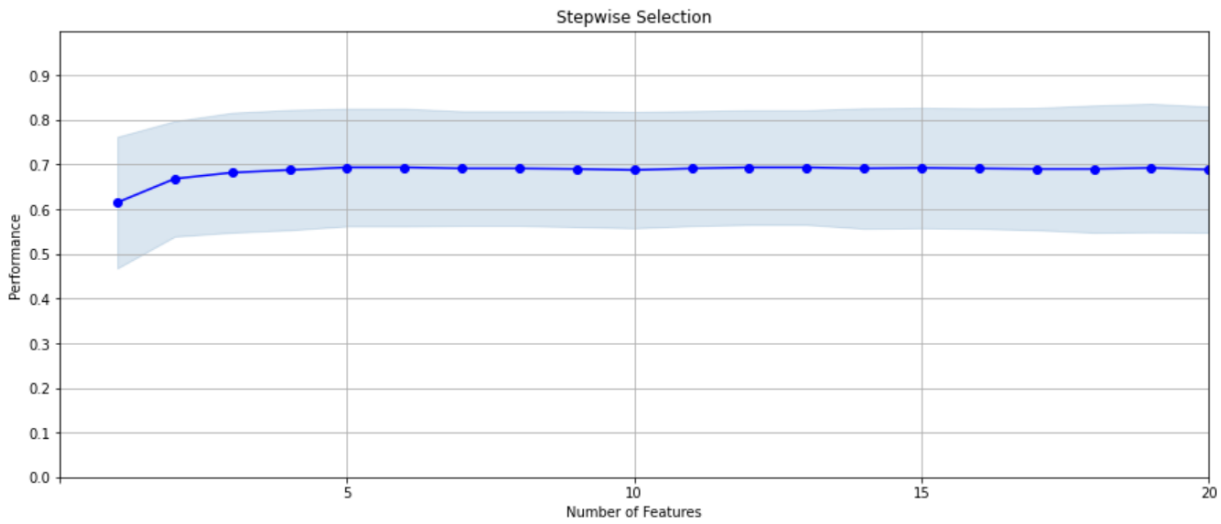

Figure 13

Table 5 provides information on our final variables.

Rank	Variable name	Univariate KS	Definition
1	card_zip_total_3	0.6777	Total amount for this cardnum+zip over the past 3 days
2	card_state_max_30	0.5979	Max amount for this cardnum+state over the past 30 days
3	card_merch_total_14	0.6750	Total amount for this cardnum+merchnum over the past 14 days
4	card_merch_total_0	0.6112	Total amount for this cardnum+merchnum over the past 0 days
5	card_zip_avg_14	0.5948	Average Amount for this cardmum+zip over the past 14 days
6	card_merch_avg_30	0.5932	Average Amount for this cardmum+merchnum over the past 14 days
7	card_state_total_0	0.6107	Total amount for this cardnum+state over the past 0 days
8	card_des_total_0	0.6100	Total amount for this cardnum+merch description over the past 0 days
9	card_zip_total_0	0.6100	Total amount for this cardnum+zip over the past 0 days
10	card_merch_avg_3	0.5898	Average Amount for this cardmum+merchnum over the past 3days
11	card_zip_avg_7	0.5927	Average Amount for this cardmum+zip over the past 7 days
12	card_merch_avg_14	0.5889	Average Amount for this cardmum+merchnum over the past 14 days
13	card_merch_total_30	0.6584	Total amount for this cardnum+merchnum over the past 30 days
14	card_zip_total_30	0.6567	Total amount for this cardnum+zip over the past 30 days
15	card_des_avg_14	0.5913	Average Amount for this cardmum+merch description over the past 14 days
16	card_des_avg_3	0.5896	Average Amount for this cardmum+merch description over the past 3 days
17	card_zip_avg_3	0.5904	Average Amount for this cardmum+zip over the past 3 days
18	card_des_max_30	0.6545	Max amount for this cardmum+merch description over the past 30 days
19	card_state_total_7	0.6697	Total amount for this cardnum+state over the past 7 days
20	card_des_total_14	0.6655	Total amount for this cardmum+merch description over the past 14 days

Table 5

Model Algorithms

Logistic Regression

Logistic Regression is a basic statistical model which is used to model the probability of an event taking place. A binary logistic model has a dependent variable with two possible values, which are labeled as "0" and "1". For the independent variables, they can each be a binary variable or a continuous variable. Logistic Regression can also be regarded as a variation of linear regression, which uses the activation function after Linear Regression to classify the dependent variable in binary class rather than the continuous variable. The common activation function is the Sigmoid function.

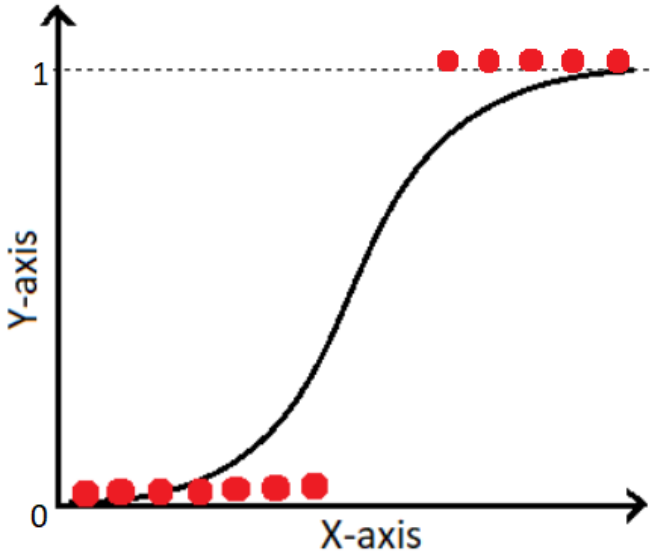

Figure 14

We ran a logistic regression using different combinations of C value, penalty, and solver. C is the inverse of regularization strength, it must be a positive float, and the smaller values specify stronger regularization. Both C and penalty are hyperparameters for regularization, and it is used to reduce the risk of overfitting. solver is the different optimization method we choose, lbfgs is a default value and we also tried liblinear, which is good for small datasets max iteration is the maximum number of iterations taken for the solvers to converge. We can see from the table below, if we set C as 0.1, the regularization maybe too strong leading to underfitting, setting C as 1 and used l1 penalty and the max iteration is 1000 is a reasonable combination and the performance of the model on oot is the best, which is 0.28.This model would serve as our baseline model to improve upon with more advanced algorithms.

Model	Parameters					Avg FDR at 3%		
	# of Variables	C	penalty	solver	max_iter	Train	Test	OOT
Logistics Regression	20	1	l2	lbfgs	1000	0.615	0.608	0.268
	20	1	l1	liblinear	1000	0.618	0.619	0.281
	20	0.1	l1	liblinear	2000	0.607	0.593	0.264
	20	1	l2	liblinear	1000	0.611	0.606	0.256
	20	0.1	l1	liblinear	1000	0.606	0.605	0.260

Table 6

Decision Tree

Decision tree is a tree structure predicting model, A tree is built by splitting the dataset, from which the root node of the tree splits into subsets, which constitute the successor children. The splitting is based on a set of splitting rules which is based on classification features. And this process is repeated on each derived subset in a recursive manner. The recursion is completed when the subset at a node has all the same values of the target variable, or when splitting no longer adds value to the predictions.

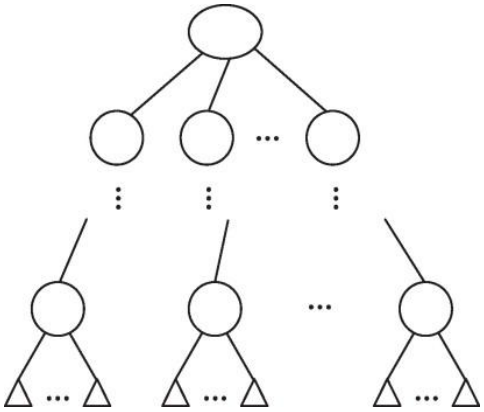

Figure 15

We used sklearn's decision tree classifier to run the decision tree model. To tune the model we tried gini and entropy as the rule to split data. Entropy is a measure of the disorder of a system. Gini is a measure of how often a randomly chosen instance from the dataset would be incorrectly labeled. We mainly tuned two important hyperparameters, min sample split and max depth. if the number of max depth is too big for example none which without a restriction, then there will be no pruning and the model will likely to be overfitting,we can see from the table, when the max depth is none, although the performance in training set is 1 but the performance in testing data and oot is bad, which means the model is severe overfitting. Min sample split is the number of records after splitting, if the number is 2, then it should have 2 records after splitting into two nodes. we turned the two hyperparameters and increased them slowly, and when the min sample split is 5 and max depth is 5, the performance of the model is the best.The best model's fraud

detection rate at the 3% threshold was 71% for training, 72% for testing, and 54% for the holdout samples.

	Iteration	# of Variables	min_samples_leaf	min_samples_split	max_features	max_depth	criterion	Train	Test	OOT
Decision Tree	1	20	1	2	None	None	gini	1	0.587	0.294
	2	20	1	2	None	None	entropy	1	0.594	0.237
	3	20	1	2	None	3	gini	0.644	0.639	0.412
	4	20	1	2	None	4	gini	0.682	0.674	0.485
	5	20	1	2	None	5	gini	0.692	0.652	0.49
	6	20	1	2	log2	5	gini	0.669	0.648	0.365
	7	20	2	2	None	5	gini	0.702	0.669	0.48
	8	20	1	3	None	5	gini	0.705	0.682	0.513
	9	20	1	4	None	10	gini	0.707	0.689	0.52
	10	20	1	4	None	10	gini	0.724	0.697	0.477
	11	20	1	5	None	5	gini	0.712	0.716	0.54
	12	20	1	5	None	10	gini	0.698	0.676	0.501
	13	20	30	30	None	5	gini	0.713	0.708	0.506
	14	20	30	50	None	5	gini	0.725	0.715	0.527
	15	20	30	80	None	10	gini	0.71	0.7	0.521
	16	20	40	50	None	10	gini	0.734	0.701	0.492

Table 7

Random Forest

Random Forest model is an ensemble learning method that constructs a multitude of decision trees. It works by sampling random records of the dataset with replacement, and constructing the optimal tree for the chosen records. Then, the model combines all the results of the trees to get to the final outcome. The main principle behind the ensemble model is that a group of weak learners come together to form a strong learner. For a regression problem, the model will return the average value. For classification problems, the model votes for the majority result. The figure below shows how Random Forest works.

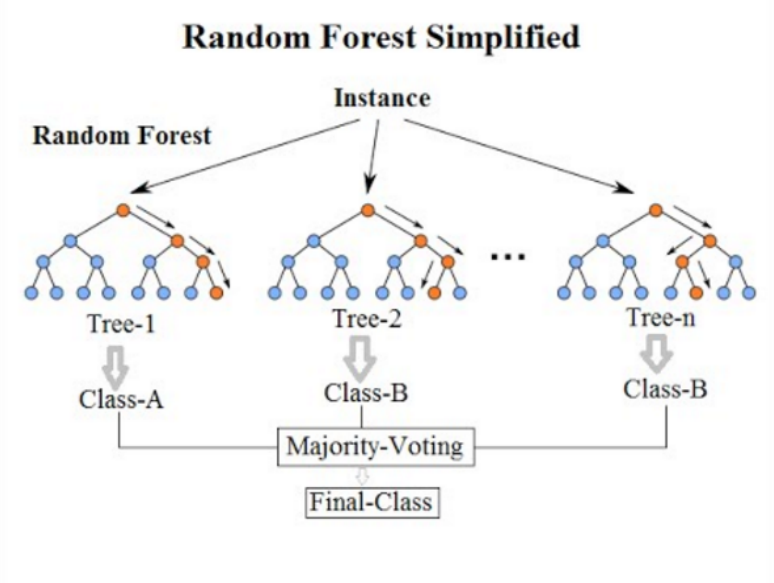

Figure 16

For this classification task, We used the RandomForestClassifier package from the library sklearn to make the Random Forest model on our variables. We varied the number of trees, the way to split, the max number of features, the minimum number of samples required to split an internal node, and the maximum depth of the tree. Then we trained our model on training data. After that, we predicted the probability of fraud over training, test and OOT. Our top performing model occurred with the number of trees as 250, maximum depth as 10, min_sample_split as 100, min_sample_leaf as 50 using the gini method to split. The model's fraud detection rate at the 3% threshold was 80% for training, 76% for testing and 57% for the holdout samples.

	Iteration	# of Variables	n_estimators	max_depth	min_samples_leaf	min_samples_split	criterion	Train	Test	OOT
Random Forest	1	20	100	None	1	2	gini	1	0.775	0.536
	2	20	100	None	1	3	gini	1	0.782	0.531
	3	20	100	None	1	3	entropy	1	0.769	0.503
	4	20	100	None	1	4	gini	1	0.785	0.528
	5	20	100	None	1	5	gini	1	0.786	0.534
	6	20	100	None	1	6	gini	1	0.79	0.53
	7	20	50	None	1	5	gini	1	0.776	0.53
	8	20	200	None	1	5	gini	1	0.777	0.54
	9	20	200	None	2	5	gini	1	0.785	0.533
	10	20	150	None	1	5	gini	1	0.775	0.54
	11	20	150	10	1	5	gini	0.908	0.781	0.56
	12	20	150	5	1	5	gini	0.757	0.721	0.529
	13	20	150	20	1	5	gini	1	0.788	0.548
	14	20	200	10	1	5	gini	0.914	0.778	0.561
	15	20	250	10	1	5	gini	0.904	0.803	0.563
	16	20	300	10	1	5	gini	0.916	0.773	0.562
	17	20	250	10	20	60	gini	0.817	0.769	0.563
	18	20	250	10	20	100	gini	0.8	0.765	0.566
	19	20	250	10	50	100	gini	0.8	0.76	0.57
	20	20	250	10	80	100	gini	0.78	0.74	0.56

Table 8

Light GBM

Light GBM model works by building a series of many weak trees and adding them up. Gradient Boosting optimizes the loss function by constructing a lot of decision trees sequentially, each adding a little more correction, so that the whole tree model can perform better than any single tree.

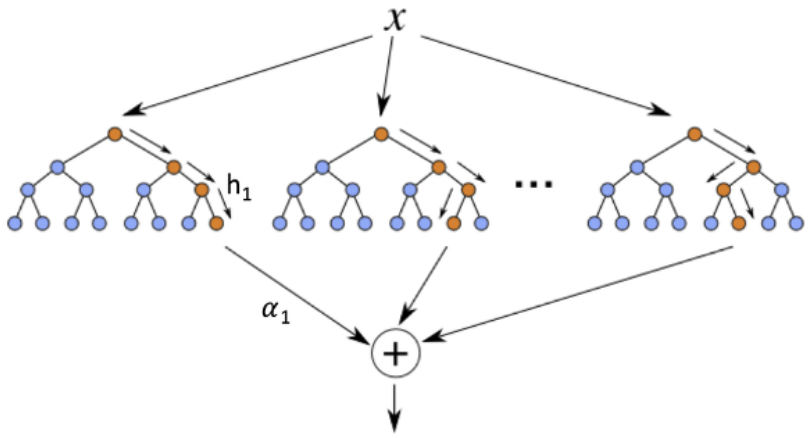

Figure 17

We applied the LGBMClassifier package and tried different settings of 6 hyperparameters: boosting_type, num_leaves, max_depth, learning rate, n_estimators, min_child_samples. The top model's fraud detection rate at the 3% threshold was 83% for training, 78% for testing and 56% for the holdout sample.

	Iteration	# of Variables	boosting_type	num_leaves	max_depth	learning_rate	n_estimators	min_child_samples	Train	Test	OOT
LGBM	1	20	gbdt	31	-1 (no limit)	0.1	100	20	0.972	0.785	0.405
	2	20	gbdt	31	5	0.1	100	20	0.929	0.796	0.446
	3	20	gbdt	31	10	0.1	100	20	0.969	0.784	0.393
	4	20	gbdt	10	5	0.1	100	20	0.889	0.8	0.507
	5	20	gbdt	10	5	0.1	100	100	0.87	0.811	0.537
	6	20	dart	10	5	0.1	100	100	0.833	0.778	0.556
	7	20	goss	10	5	0.1	100	100	0.869	0.804	0.555
	8	20	goss	10	5	0.1	50	100	0.829	0.781	0.509
	9	20	goss	10	4	0.1	100	100	0.82	0.77	0.555
	10	20	goss	10	3	0.1	100	100	0.8	0.76	0.53
	11	20	goss	10	2	0.1	100	100	0.76	0.74	0.55
	12	20	goss	10	5	0.01	100	100	0.805	0.77	0.535
	13	20	goss	10	2	0.01	100	100	0.718	0.703	0.527

Table 9

Multilayer Perceptron

A multilayer perceptron (MLP) is a feedforward artificial neural network that generates a set of outputs from a set of inputs. An MLP consists of at least three layers of nodes: an input layer, a hidden layer, and an output layer. Except for the input nodes, each node is a neuron that uses a

nonlinear activation function. MLP utilizes a supervised learning technique called backpropagation for training. It connects multiple layers in a directed graph, which means that the signal path through the nodes only goes one way. Each node, apart from the input nodes, has a nonlinear activation function. Its multiple layers and non-linear activation distinguish MLP from a linear perceptron. It can distinguish data that is not linearly separable. The figure illustrates the process of the multilayer perceptron model.

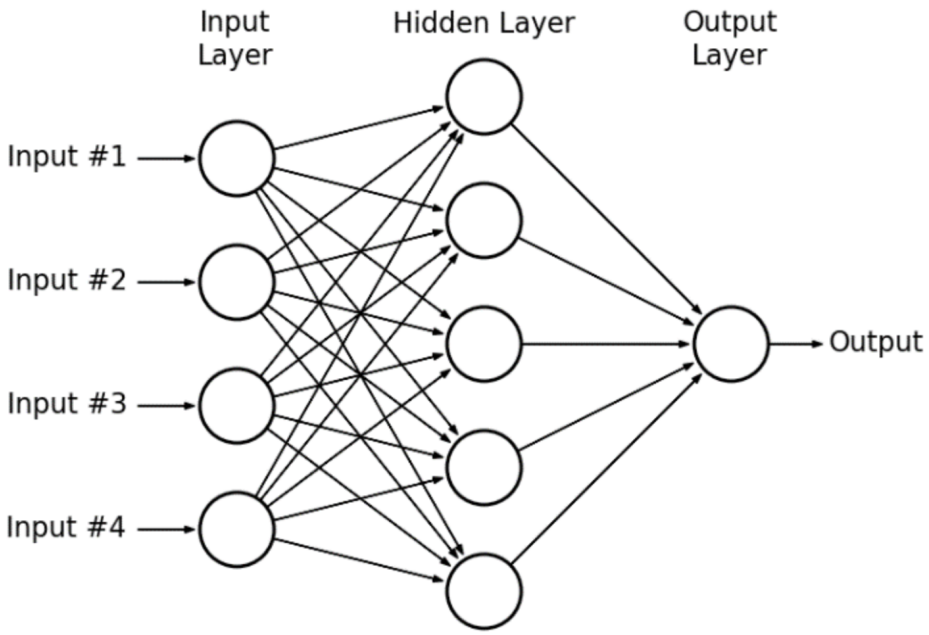

Figure 18

We used the MLPClassifier package from the library sklearn to train the neural network model on our variables. We tuned several hyperparameters, alpha, activation, hidden layer sizes and learning rate. For activation, ‘tanh’ returns f(x) = tanh(x), ‘relu’ returns f(x) = max(0, x). The bigger the hidden size is, the more time the model training would use, and they are more likely to overfit. We can see from the table, when the number of nodes is 100, the model is overfitting. When the number of nodes is 5, 10 or 20 and there is only one hidden layer, the model is underfitting. Finally we set it as 50. After we tuned the model, the performance of the best model in oot is 0.575.

	Iteration	# of Variables	solver	alpha	activation	hidden_layer_sizes	learning_rate	max_iter	Train	Test	OOT
	1	20	adam	0.0001	relu	(100,)	constant	200	0.774	0.742	0.53
	2	20	adam	0.0001	tanh	(100,)	constant	200	0.754	0.734	0.565
	3	20	adam	0.0001	tanh	(50,50,50)	adaptive	200	0.841	0.774	0.363
	4	20	adam	0.0001	tanh	(50,100,50)	constant	200	0.854	0.778	0.38
	5	20	adam	0.0001	tanh	(10,10,10)	constant	200	0.748	0.723	0.546
	6	20	adam	0.0001	tanh	(10,)	constant	200	0.724	0.711	0.57
	7	20	adam	0.0001	tanh	(5,)	constant	200	0.706	0.706	0.545
	8	20	adam	0.0001	tanh	(20,)	constant	200	0.738	0.73	0.57
	9	20	adam	0.0001	relu	(20,)	constant	200	0.722	0.702	0.54
	10	20	adam	0.0001	tanh	(50,)	constant	200	0.752	0.731	0.575
MLP Classifier	11	20	adam	0.0001	tanh	(50,)	adaptive	200	0.762	0.732	0.57
	12	20	adam	0.0001	tanh	(50,)	invscaling	200	0.752	0.738	0.573
	13	20	adam	0.0001	tanh	(80,)	constant	200	0.756	0.748	0.558
	14	20	adam	0.0001	tanh	(60,)	constant	200	0.752	0.735	0.561
	15	20	adam	0.00001	tanh	(50,)	constant	200	0.75	0.73	0.57
	16	20	sgd	0.0001	logistic	(50,25)	adaptive	200	0.632	0.638	0.292
	17	20	sgd	0.001	logistic	(30,20)	adaptive	200	0.636	0.649	0.318
	18	20	adam	0.01	relu	(25,20)	adaptive	200	0.767	0.738	0.572
	19	20	sgd	0.0005	relu	(30,20)	adaptive	500	0.788	0.763	0.484
	20	20	sgd	0.001	relu	(30,20)	constant	500	0.784	0.768	0.509
	21	20	sgd	0.001	relu	(50,)	constant	500	0.768	0.738	0.572
	22	20	adam	0.01	relu	(50,)	constant	500	0.748	0.728	0.573

Table 10

Results

Fraud Detection tables for training, testing and oot data

Our best performing algorithm is the multilayers perceptron model (neural network) with parameters solver = adam, alpha = 0.0001, activation = tanh, hidden_layer_sizes = (50,), learning_rate = constant, and max_iter = 200. Its average Fraud Detection Rate in Out-Of-Time data at 3% is 0.575.

We ran the selected model separately on the train, test, and OOT data set and sorted the data sets descendingly according to predicted fraud probability. Then we divided the data set into 100 bins to check the performance of our selected model.

Three tables below include numbers of Good/Bad/Total records, percentages of Good/Bad records, cumulative numbers of Good/Bad/Total records, cumulative percentages of Good/Bad records, KS, FDR (Fraud Detection Rate), and FPR (False Positive Rate) of the top 20 bins.

Training

Training	# Records	# Goods	# Bads	Fraud Rate
	59010	58387	623	1.06%

bin	Bin Statistics					Cumulative Statisitcs						
	# Records	# Goods	# Bads	% Goods	% Bads	Total # Records	Cumulative Good	Cumulative Bad	% Goods	% Bads (FDR)	KS	FPR
1	590	185	405	31.36	68.64	590	185	405	0.32	64.90	64.59	0.46
2	590	523	67	88.64	11.36	1180	708	472	1.21	75.64	74.43	1.50
3	590	557	33	94.41	5.59	1770	1265	505	2.17	80.93	78.76	2.50
4	590	578	12	97.97	2.03	2360	1843	517	3.16	82.85	79.70	3.56
5	590	578	12	97.97	2.03	2950	2421	529	4.15	84.78	80.63	4.58
6	591	581	10	98.31	1.69	3541	3002	539	5.14	86.38	81.24	5.57
7	590	583	7	98.81	1.19	4131	3585	546	6.14	87.50	81.36	6.57
8	590	585	5	99.15	0.85	4721	4170	551	7.14	88.30	81.16	7.57
9	590	587	3	99.49	0.51	5311	4757	554	8.15	88.78	80.63	8.59
10	590	584	6	98.98	1.02	5901	5341	560	9.15	89.74	80.60	9.54
11	590	580	10	98.31	1.69	6491	5921	570	10.14	91.35	81.21	10.39
12	590	585	5	99.15	0.85	7081	6506	575	11.14	92.15	81.00	11.31
13	590	589	1	99.83	0.17	7671	7095	576	12.15	92.31	80.16	12.32
14	590	585	5	99.15	0.85	8261	7680	581	13.15	93.11	79.96	13.22
15	591	587	4	99.32	0.68	8852	8267	585	14.16	93.75	79.59	14.13
16	590	588	2	99.66	0.34	9442	8855	587	15.17	94.07	78.90	15.09
17	590	586	4	99.32	0.68	10032	9441	591	16.17	94.71	78.54	15.97
18	590	589	1	99.83	0.17	10622	10030	592	17.18	94.87	77.69	16.94
19	590	587	3	99.49	0.51	11212	10617	595	18.18	95.35	77.17	17.84
20	590	586	4	99.32	0.68	11802	11203	599	19.19	95.99	76.81	18.70

Table 11

Testing

Test	# Records	# Goods	# Bads	Fraud Rate
	25290	25033	257	1.02%

	Bin Statistics					Cumulative Statisitcs						
Population Bin %	# Records	# Goods	# Bads	% Goods	% Bads	Total # Records	Cumulative Good	Cumulative Bad	% Goods	% Bads (FDR)	KS	FPR
1	253	87	166	34.39	65.61	253	87	166	0.35	64.84	64.50	0.52
2	253	224	29	88.54	11.46	506	311	195	1.24	76.17	74.93	1.59
3	253	248	5	98.02	1.98	759	559	200	2.23	78.13	75.89	2.80
4	253	249	4	98.42	1.58	1012	808	204	3.23	79.69	76.46	3.96
5	252	249	3	98.81	1.19	1264	1057	207	4.22	80.86	76.64	5.11
6	253	250	3	98.81	1.19	1517	1307	210	5.22	82.03	76.81	6.22
7	253	249	4	98.42	1.58	1770	1556	214	6.22	83.59	77.38	7.27
8	253	252	1	99.60	0.40	2023	1808	215	7.22	83.98	76.76	8.41
9	253	252	1	99.60	0.40	2276	2060	216	8.23	84.38	76.15	9.54
10	253	253	0	100.00	0.00	2529	2313	216	9.24	84.38	75.14	10.71
11	253	251	2	99.21	0.79	2782	2564	218	10.24	85.16	74.91	11.76
12	253	250	3	98.81	1.19	3035	2814	221	11.24	86.33	75.09	12.73
13	253	250	3	98.81	1.19	3288	3064	224	12.24	87.50	75.26	13.68
14	253	251	2	99.21	0.79	3541	3315	226	13.24	88.28	75.04	14.67
15	253	251	2	99.21	0.79	3794	3566	228	14.24	89.06	74.82	15.64
16	252	251	1	99.60	0.40	4046	3817	229	15.25	89.45	74.21	16.67
17	253	252	1	99.60	0.40	4299	4069	230	16.25	89.84	73.59	17.69
18	253	253	0	100.00	0.00	4552	4322	230	17.26	89.84	72.58	18.79
19	253	250	3	98.81	1.19	4805	4572	233	18.26	91.02	72.75	19.62
20	253	248	5	98.02	1.98	5058	4820	238	19.25	92.97	73.71	20.25

Table 12

OOT

OOT	# Records	# Goods	# Bads	Fraud Rate
	12097	11918	179	1.48%

	Bin Statistics					Cumulative Statisitcs						
Population Bin %	# Records	# Goods	# Bads	% Goods	% Bads	Total # Records	Cumulative Good	Cumulative Bad	% Goods	% Bads (FDR)	KS	FPR
1	121	68	53	56.20	43.80	121	68	53	0.57	29.61	29.04	1.28
2	121	92	29	76.03	23.97	242	160	82	1.34	45.81	44.47	1.95
3	121	101	20	83.47	16.53	363	261	102	2.19	56.98	54.79	2.56
4	121	117	4	96.69	3.31	484	378	106	3.17	59.22	56.05	3.57
5	121	118	3	97.52	2.48	605	496	109	4.16	60.89	56.73	4.55
6	121	118	3	97.52	2.48	726	614	112	5.15	62.57	57.42	5.48
7	121	119	2	98.35	1.65	847	733	114	6.15	63.69	57.54	6.43
8	121	120	1	99.17	0.83	968	853	115	7.16	64.25	57.09	7.42
9	121	120	1	99.17	0.83	1089	973	116	8.16	64.80	56.64	8.39
10	121	117	4	96.69	3.31	1210	1090	120	9.15	67.04	57.89	9.08
11	121	117	4	96.69	3.31	1331	1207	124	10.13	69.27	59.15	9.73
12	121	117	4	96.69	3.31	1452	1324	128	11.11	71.51	60.40	10.34
13	121	117	4	96.69	3.31	1573	1441	132	12.09	73.74	61.65	10.92
14	121	120	1	99.17	0.83	1694	1561	133	13.10	74.30	61.20	11.74
15	121	120	1	99.17	0.83	1815	1681	134	14.10	74.86	60.76	12.54
16	121	119	2	98.35	1.65	1936	1800	136	15.10	75.98	60.87	13.24
17	120	119	1	99.17	0.83	2056	1919	137	16.10	76.54	60.43	14.01
18	121	118	3	97.52	2.48	2177	2037	140	17.09	78.21	61.12	14.55
19	121	121	0	100.00	0.00	2298	2158	140	18.11	78.21	60.11	15.41
20	121	119	2	98.35	1.65	2419	2277	142	19.11	79.33	60.22	16.04

Table 13

Fraud Saving Table and Plot for the OOT (Validation Data)

In order to decide the cutoff point of rejecting suspected fraud transactions, we plotted a line graph to showcase the money we saved through the model, the loss of rejecting innocent transactions and their difference as the total gain. We assume we can gain 2K dollars for every fraud that the model catches, and we will lose 50 dollars for every false positive transaction. As we wanted to maximize the total savings, the best cutoff point is at 4% of the population bin, saying that we should reject the top 4% suspected fraud transactions with total savings of 193,100 dollars.

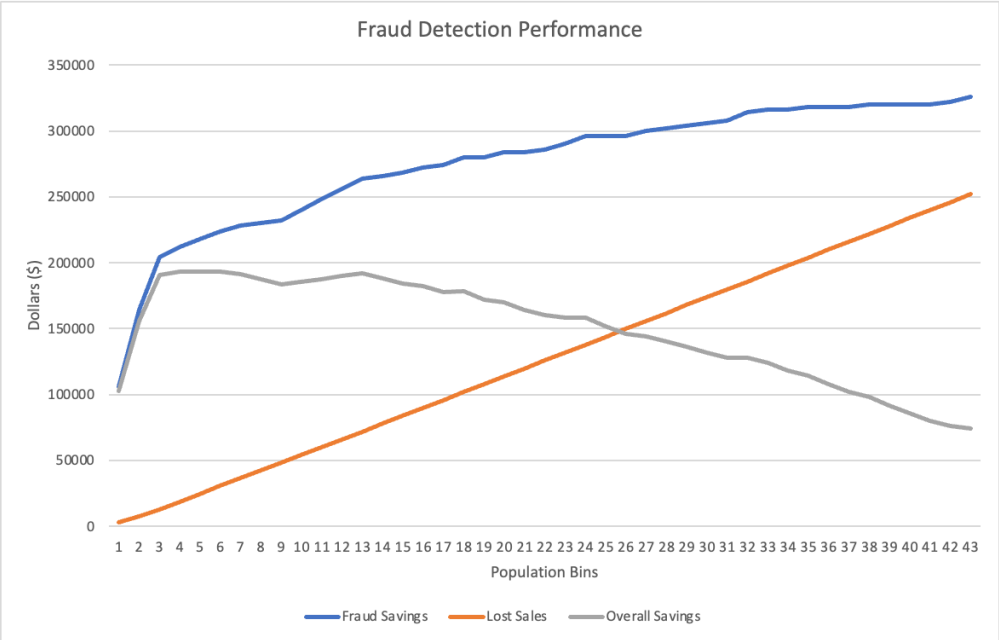

Figure 19

Bin	Records	Good	Bad	Fraud Saving	Lost Sales	Overall Saving
1	121	68	53	106000	3400	102600
2	121	92	29	164000	8000	156000
3	121	101	20	204000	13050	190950
4	121	117	4	212000	18900	193100
5	121	118	3	218000	24800	193200
6	121	118	3	224000	30700	193300
7	121	119	2	228000	36650	191350
8	121	120	1	230000	42650	187350
9	121	120	1	232000	48650	183350
10	121	117	4	240000	54500	185500
11	121	117	4	248000	60350	187650
12	121	117	4	256000	66200	189800
13	121	117	4	264000	72050	191950
14	121	120	1	266000	78050	187950
15	121	120	1	268000	84050	183950
16	121	119	2	272000	90000	182000
17	120	119	1	274000	95950	178050
18	121	118	3	280000	101850	178150
19	121	121	0	280000	107900	172100
20	121	119	2	284000	113850	170150

Table 14

Dynamic Plot for Entities

We further observed how the fraud score changed along with the number of transactions and time period for credit card number 5142223373 and merchant number 9108234610000. Usually the fraud score looks normal in the first few transactions and starts to look unusual as the number of transactions increases. In December, there were totally 63 transactions of the card, and about 25.4% of which happened within 4 days from 12.16 to 12.20. From figure 20, we can observe a sharp increase in fraud score when there are more transactions within a short period of time. Similarly, from figure 21, the fraud score starts to increase fast when the accumulated transactions in this merchant exceed 40% of the total transactions in May.

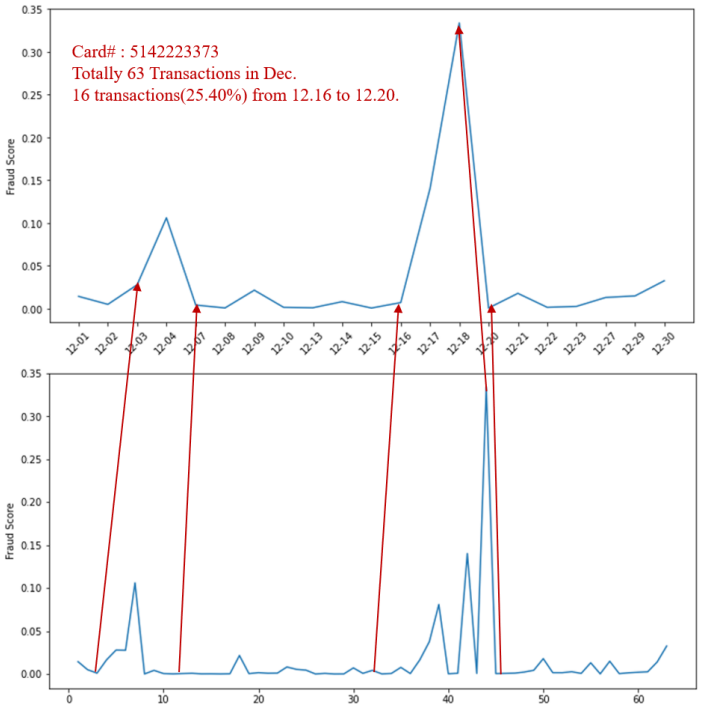

Figure 20

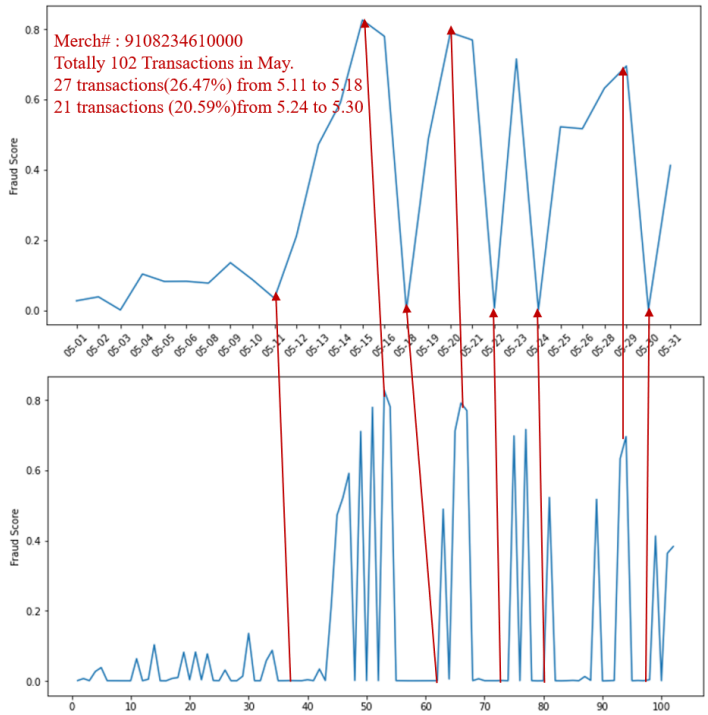

Figure 21

Feature Importance

Since our final model is the MLP classifier which does not have an intrinsic feature importance, we used a tuned random forest classifier that achieved a similar FDR of 56.10% in OOT data with the same 20 features to search the top 5 most significant features. From the plot we can see that the top 5 features are: card_des_total_3, card_merch_total_3, card_zip_total_3, card_des_total_0, card_merch_total_14.

Feature Name	Description
card_des_total_3	Total amount for this card# + merchant description over the past 3 days
card_merch_total_3	Total amount for this card# + merchant #over the past 3 days
card_zip_total_3	Total amount for this card# +zip code over the past 3 days
card_des_total_0	Total amount for this card# + merchant description over the past 0 day
card_merch_total_14	Total amount for this card# + merchant # over the past 14 days

Table 15

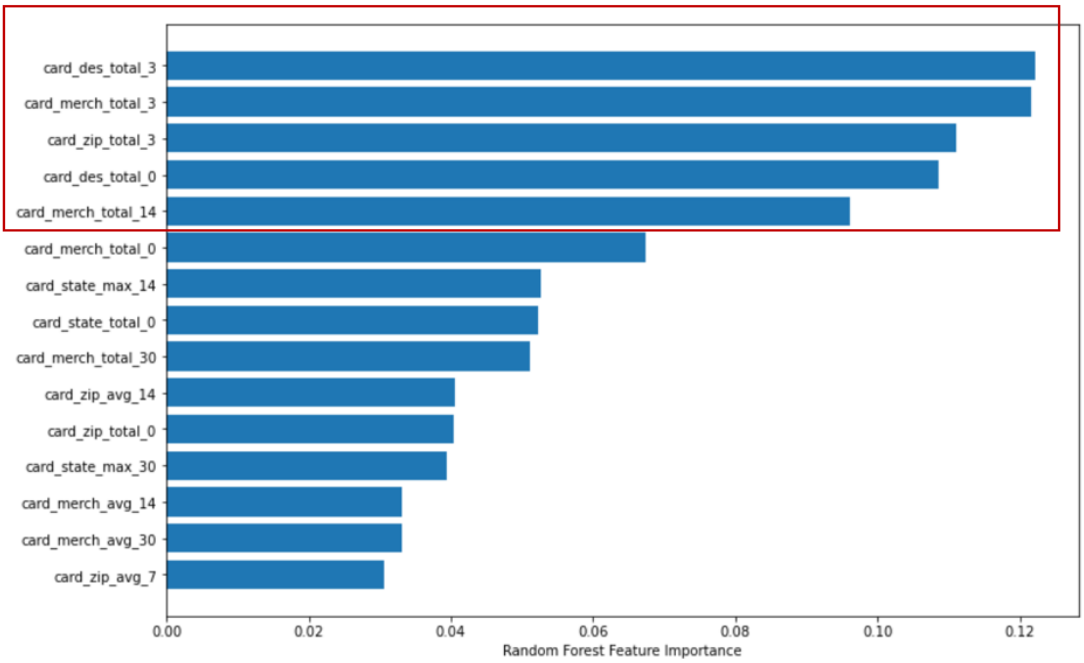

Figure 22

Conclusions

Conclusions

In this report, we examined the dataset and went through the entire process of building a fraud detection model, from data exploration, and variables construction, to variable and model selection. After testing several models and dozens of hyperparameter combinations, we have decided that the best model is Neural Network. The FDR for this model is 75.2%, 73.1%, and 57.5% on train, test, and OOT datasets over 3% of the population.

Based on our model, we would say it can eliminate 57.5% of all the frauds while declining only 3% of the applications. Also, we would recommend the best cutoff point is at 4% of the population bin to maximize the savings.

Future Improvements

We believe that our results are produced to the best of our ability and effort in a limited time. In some areas of the project, however, we can improve our operations if we have more time and more resources and draw lessons applicable to future related projects.

First, we may consult with a domain expert to create more useful variables. Also, we would like to explore more models and ensemble models and try more combinations of hyperparameters. All of these operations may improve our results.

Secondly, in our dataset, the fraud records only take up around 1% of the population. So there exists the data imbalance issue which may affect our results. We understand that fraud is an unlikely occurrence and there are more normal records than fraud records by nature, but there are remedies like Synthetic Minority Oversampling Technique (SMOTE) available for us to make the dataset more balanced in order to get better results.

Finally, we would like to try to collect more related data since this dataset is too small and we only have about 96,000 records and 10 features. If we have more data, we may have a better result.

Appendix

Full Data Quality Report

Recnum

Recnum can be considered as both the time order of the records and the index of the table. Each value presents once.

Cardnum

Cardnum denotes the card numbers related to some purchasing histories. There are 1,645 unique credit card numbers for these fields, without missing values. Card number ‘5,142,148,452’ has the highest frequency. Following are the distributions of the top 15 categories.

	Cardnum
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

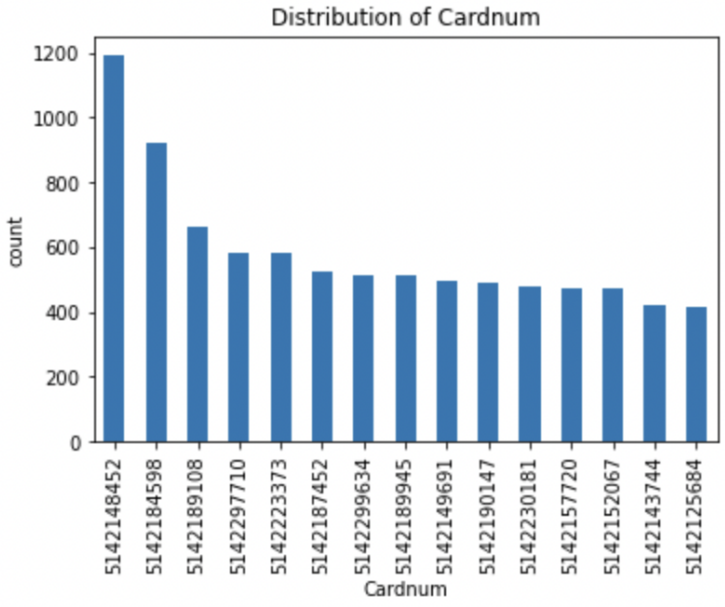

Figure 23 Distribution of ‘Cardnum’ variable

Date

Date notes the purchase date of each record, ranging from ‘2016-01-01’ to ‘2016-12-31’. There are 365 unique values, and without missing/null values. Following are the distributions of the top 15 categories.

	Date
2006-02-28	684
2006-08-10	610
2006-03-15	594
2006-09-13	564
2006-09-07	536
2006-08-09	536
2006-09-14	533
2006-09-21	522
2006-08-01	521
2006-08-31	518
2006-09-28	503
2006-08-29	503
2006-02-16	501
2006-09-19	497
2006-02-09	496

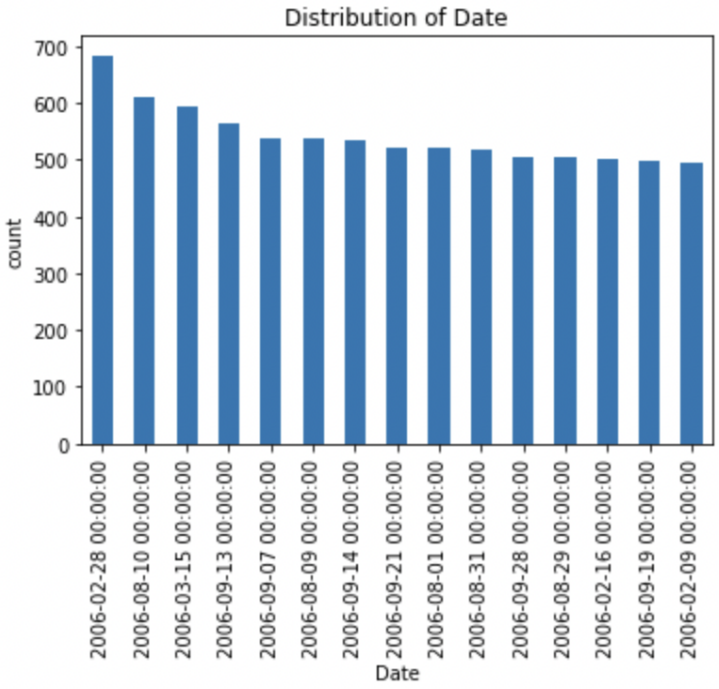

Figure 24 Distribution of ‘Date’ variable

After the summarization of application records grouped by date, followings are variations of daily transaction counts and weekly counts. Apparently, records labeled as fraudulent (red line ,Fraud= 0) have greater volatility, same as daily plotting and weekly plotting.

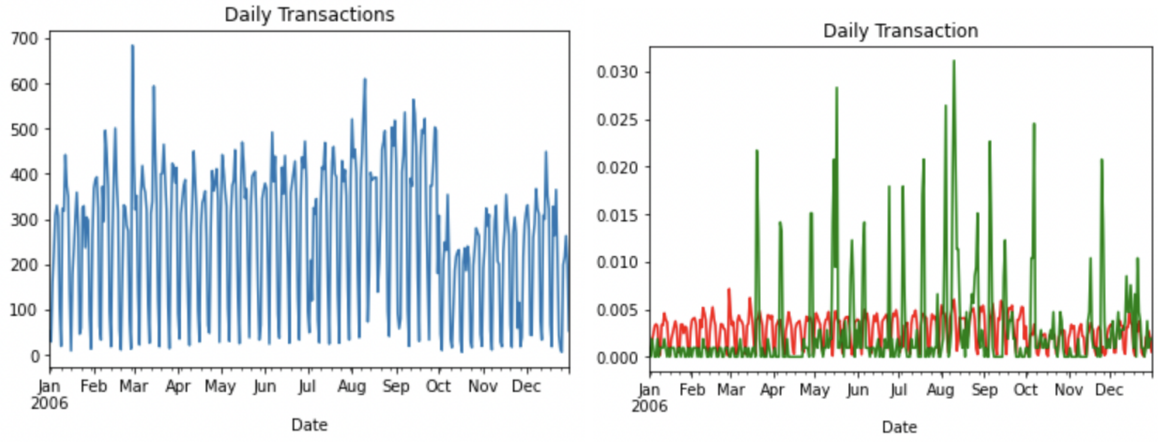

Figure 25 Variation of daily transaction

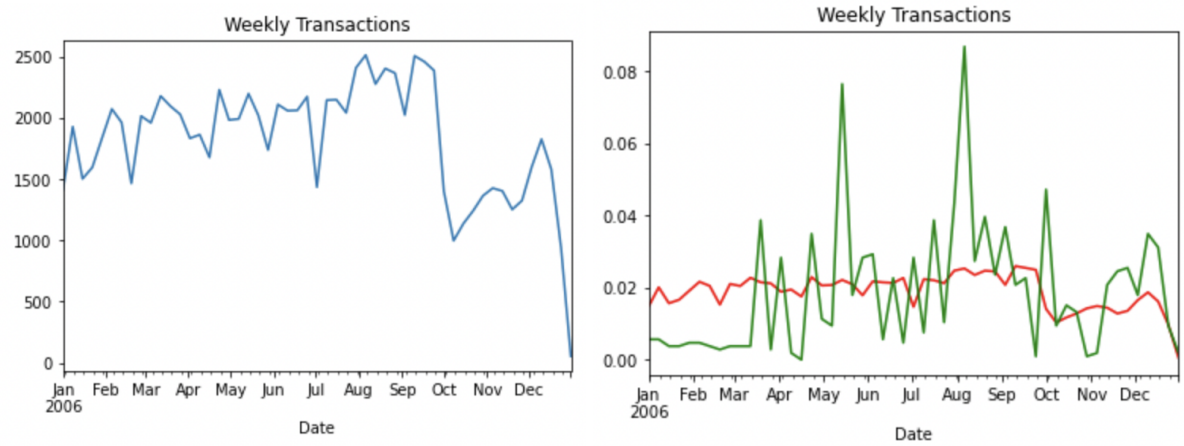

Figure 26 Variation of weekly transactions

Merchnum

Merchnum represents the number of certain products. There are 13,091 unique numbers for this field with 3,375 missing values (3.5%). The most frequently bought is the product of '930,090,121,224'. Following is the distribution of the top 15 categories.

	Merchnum
930090121224	9310
5509006296254	2131
9900020006406	1714
602608969534	1092
4353000719908	1020
410000971343	982
9918000409955	956
5725000466504	872
9108234610000	817
602608969138	783
4503082476300	746
2094206450000	590
4063000739258	568
2094330000009	533
6920602000804	523

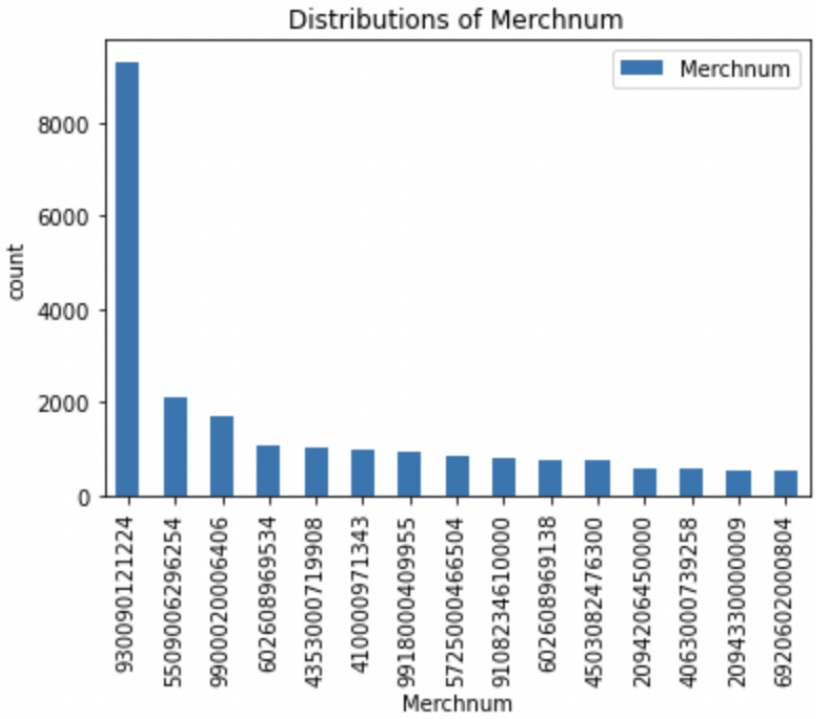

Figure 27 Distribution of 'Merchnum' variable

Merch description

Merch description denotes the name of the product. There are 13,126 unique descriptions for this field, without missing/null values. The most frequent category is ‘GSA-FSS-ADV’, appearing 1,688 times.

	Merch description
GSA-FSS-ADV	1688
SIGMA-ALDRICH	1635
STAPLES #941	1174
FISHER SCI ATL	1093
MWI*MICRO WAREHOUSE	958
CDW*GOVERNMENT INC	872
DELL MARKETING L.P.	816
FISHER SCI CHI	783
AMAZON.COM *SUPERSTOR	750
OFFICE DEPOT #1082	748
VWR SCIENTIFIC PROD VCTS	688
PC *PC CONNECTION	570
C & C PRODUCT SERVICES	558
BUY.COM	481
FISHER SCI HUS	442

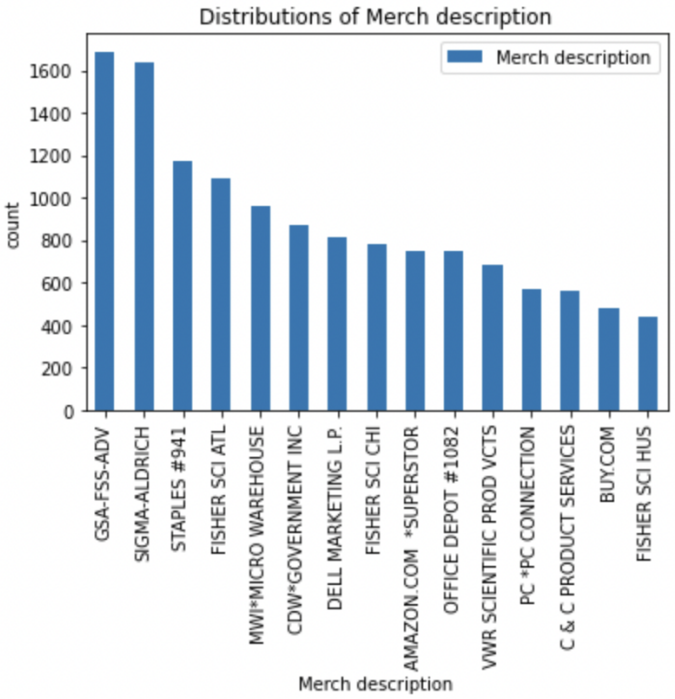

Figure 28 Distribution of ‘Merch description’ variable

Merch state

Merch state defines the state of the merchant, indicating the location of the merchant. There are 227 unique values for this field with 1,195 missing values (1.2%). Following is the distribution of the top 15 categories.

	Merch state
TN	12035
VA	7872
CA	6817
IL	6508
MD	5398
GA	5025
PA	4899
NJ	3912
TX	3790
NC	3322
WA	3300
DC	3208
OH	3131
NY	2430
MO	2420

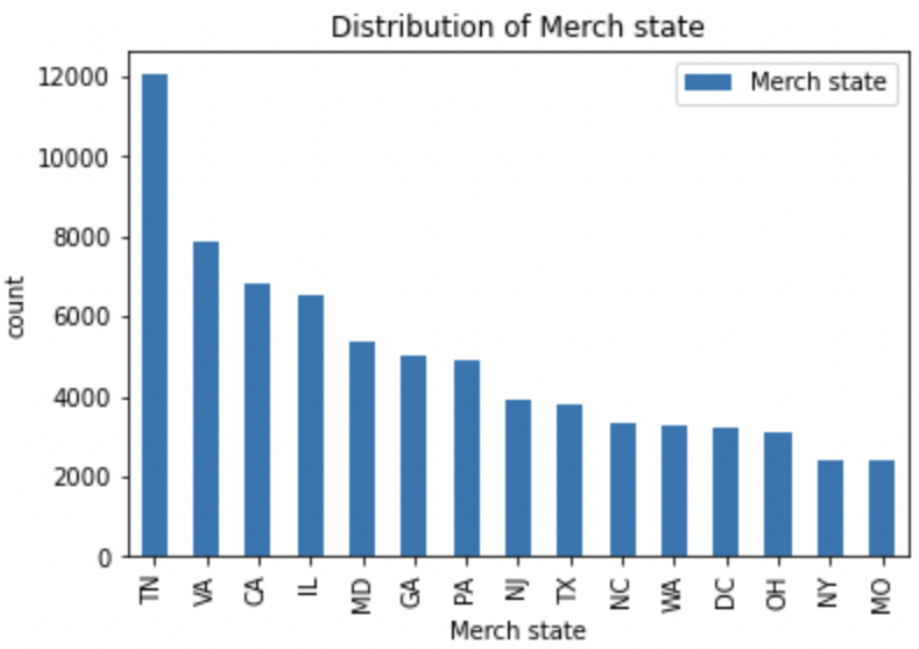

Figure 29 Distribution of 'Mech state' variable

Merch zip

This categorical variable denotes the zip code of the merchant. There are 4,567 unique values for this field with 4,656 missing values (4.8%). Following is the distribution of the top 15 categories.

	Merch zip
38118.0	11868
63103.0	1650
8701.0	1267
22202.0	1250
60061.0	1221
98101.0	1197
17201.0	1180
30091.0	1092
60143.0	942
60069.0	826
78682.0	817
19380.0	769
20763.0	749
20005.0	648
20748.0	592

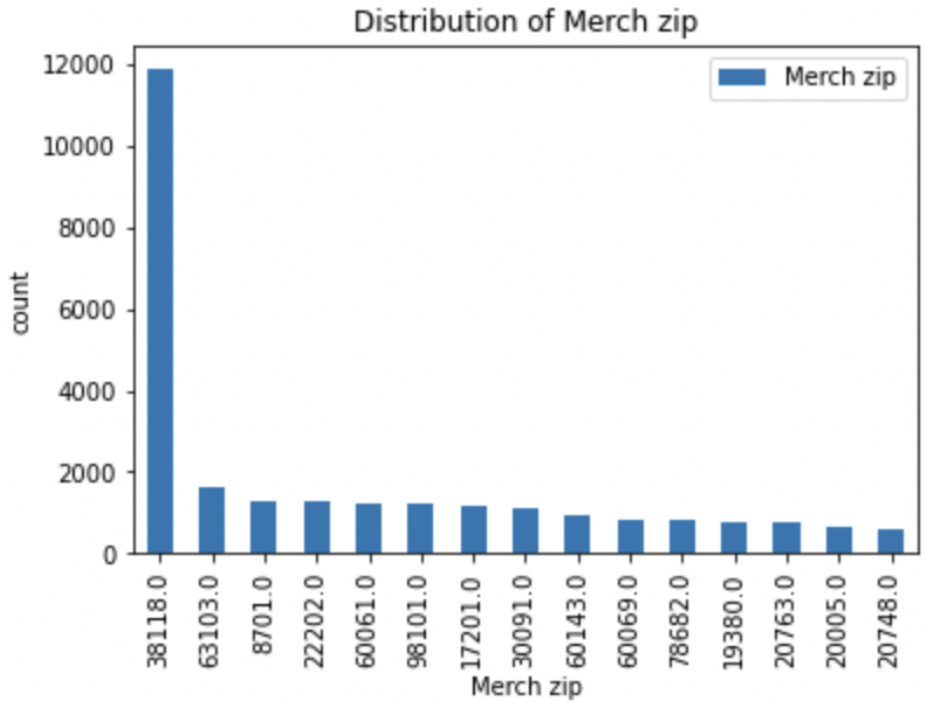

Figure 30 Distribution of 'Merch zip' variable

Transtype

Transtype defines the type of transaction of each record. There are 4 unique values for this field without missing values. Following is the distribution of all categories.

	Transtype
P	96398
A	181
D	173
Y	1

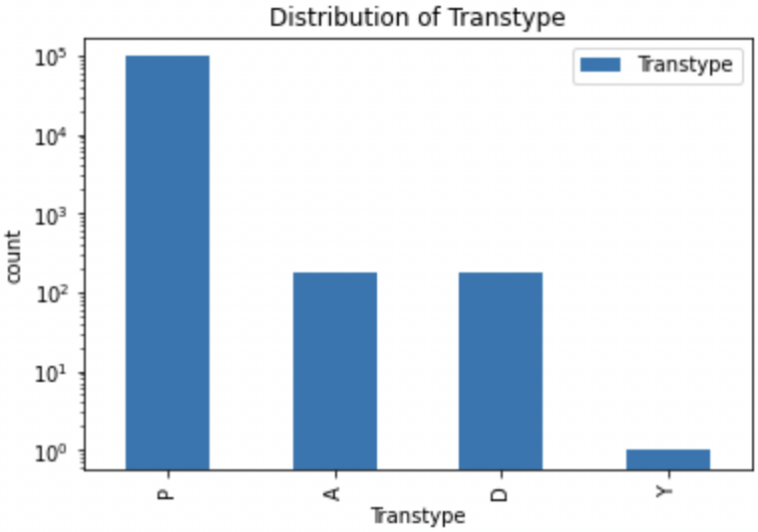

Figure 31 Distribution of ‘Transtype’ variable

Amount

This numerical variable represents the amount of each transaction. There are no missing/null values. Following graph shows the distribution of the ‘Amount’ variable.

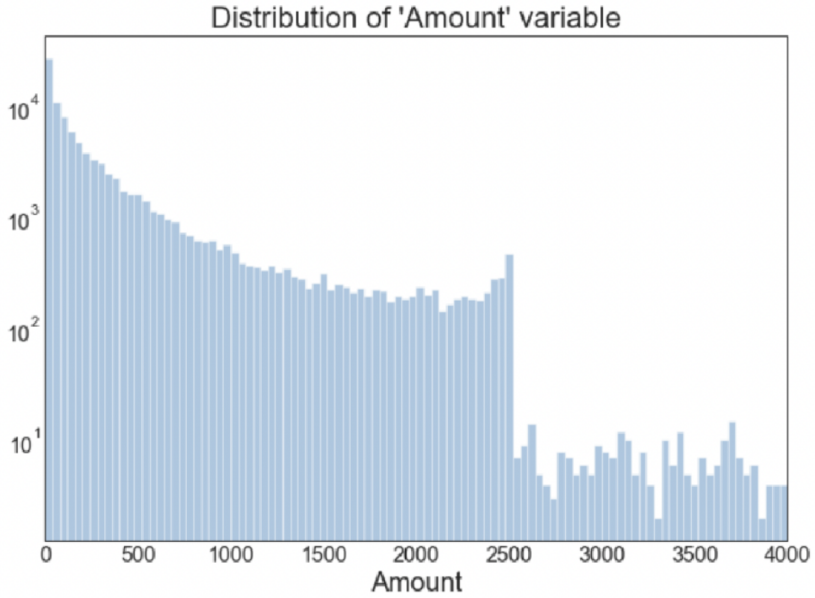

Figure 32 Distribution of ‘Amount’ variable

Fraud

This variable is the target feature in this dataset, indicating if the transaction is fraud or not. 1 denotes that the transaction is fraudulent; 0 indicates that the transaction is not fraudulent. There are two unique values for this field without missing/ null values. Following is the distribution of the two categories.

	Fraud
0	95694
1	1059

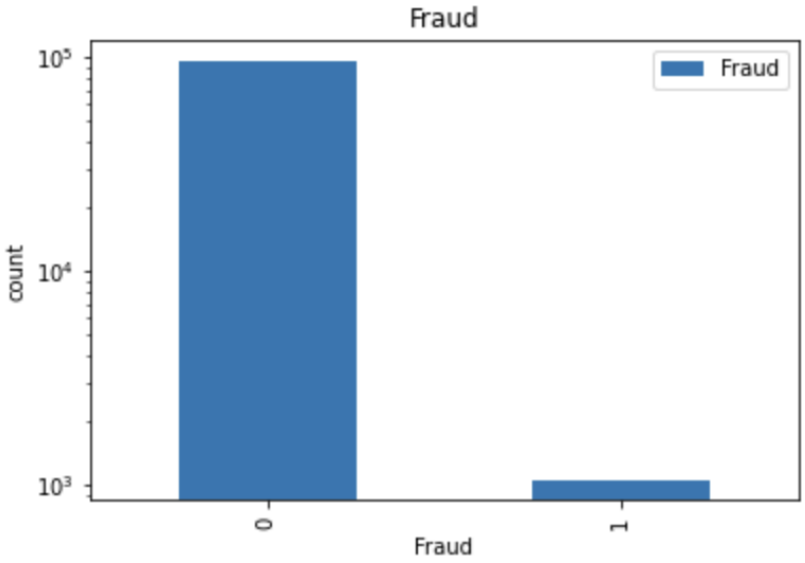

Figure 33 Distribution of 'Fraud' variable

List of All Created Variables

	Variables
1	merch_state_med_30
2	card_zip_actual/total_3
3	card_merch_actual/max_14
4	card_merch_count_1_by_7
5	merch_state_actual/total_30
6	card_merch_actual/total_30
7	des_zip_actual/total_0
8	card_des_actual/max_3
9	des_zip_avg_3
10	merch_state_count_1_by_7
11	U* x
12	des_state_actual/total_7
13	card_des_avg_0
14	card_merch_actual/avg_3
15	merch_state_actual/total_14
16	card_state_actual/avg_3
17	Merch description_avg_3
18	merch_state_actual/avg_0
19	card_zip_count_3
20	card_des_actual/med_3
21	merch_des_actual/total_7
22	card_state_avg_0
23	merch_des_actual/med_1
24	Merch description_actual/avg_14
25	des_zip_actual/med_30
26	card_state_count_14
27	card_state_actual/max_30
28	merch_state_actual/total_7
29	merch_state_med_0
30	merch_state_actual/avg_7
31	card_zip_actual/total_7
32	card_zip_count_0_by_14
33	card_state_max_0
34	card_des_actual/total_14
35	merch_des_actual/avg_14
36	card_zip_count_0_by_7
37	des_zip_actual/total_3

38	card_zip_max_14
39	des_zip_count_3
40	des_zip_med_3
41	Merch description_count_30
42	des_state_avg_30
43	card_des_count_7
44	card_zip_med_30
45	card_merch_med_7
46	des_zip_actual/max_1
47	des_state_actual/avg_1
48	card_des_max_0
49	merch_state_actual/med_14
50	card_merch_actual/med_1
51	des_state_med_1
52	card_des_med_3
53	Merch description_actual/total_30
54	merch_des_total_14
55	des_state_med_7
56	Merch description_avg_0
57	Merch description_actual/max_7
58	merch_state_max_0
59	merch_state_total_1
60	card_state_avg_1
61	card_state_count_1
62	card_des_actual/avg_3
63	card_merch_actual/max_1
64	card_zip_actual/total_14
65	merch_des_max_0
66	merch_des_actual/med_14
67	card_state_actual/avg_1
68	card_state_med_7
69	des_zip_total_7
70	des_zip_actual/total_30
71	card_state_count_1_by_14
72	des_state_actual/avg_14
73	des_zip_max_0
74	des_state_total_14
75	card_state_avg_30
76	card_merch_max_3
77	card_state_total_30
78	merch_state_med_3
79	des_state_total_3
80	merch_des_med_7

81	merch_state_actual/total_3
82	des_zip_actual/avg_30
83	merch_des_actual/avg_0
84	des_state_count_7
85	des_state_actual/total_30
86	card_des_total_1
87	card_merch_actual/avg_0
88	card_merch_count_0_by_14
89	card_des_count_1
90	des_state_total_30
91	card_zip_actual/med_1
92	card_zip_count_0_by_30
93	des_zip_avg_14
94	des_zip_med_30
95	card_merch_actual/med_0
96	merch_state_avg_3
97	merch_state_count_1
98	des_state_med_14
99	merch_des_count_0
100	des_zip_actual/avg_3
101	card_des_count_0
102	Merch description_actual/max_0
103	card_merch_med_0
104	des_zip_total_30
105	merch_des_actual/med_0
106	card_des_actual/total_0
107	Merch description_total_14
108	Merch description_actual/total_3
109	merch_des_actual/max_0
110	des_state_actual/max_0
111	des_state_max_14
112	des_zip_actual/avg_0
113	merch_des_count_0_by_14
114	des_state_count_30
115	merch_state_actual/avg_30
116	merch_state_actual/total_1
117	card_zip_med_14
118	card_des_total_0
119	card_des_actual/med_1
120	merch_state_med_14
121	card_merch_actual/max_3
122	des_state_med_30
123	card_merch_actual/max_0

124	card_state_max_14
125	merch_state_actual/avg_3
126	card_des_actual/med_14
127	card_state_actual/total_3
128	Merch description_day_since
129	Merch description_count_1_by_30
130	des_zip_actual/max_14
131	des_state_avg_0
132	merch_state_count_14
133	des_state_max_3
134	card_merch_total_7
135	merch_state_max_3
136	Merch description_avg_14
137	merch_des_actual/max_14
138	card_state_max_1
139	Merch description_med_30
140	merch_des_avg_3
141	Merch description_count_3
142	des_zip_actual/max_7
143	card_state_count_0_by_30
144	card_state_actual/avg_14
145	U*_y
146	card_des_count_14
147	merch_des_total_0
148	des_state_max_7
149	card_merch_avg_0
150	Merch description_count_1_by_7
151	card_zip_count_1_by_7
152	des_zip_day_since
153	card_merch_med_3
154	Merch description_total_30
155	card_des_max_14
156	merch_des_actual/med_7
157	card_merch_max_14
158	merch_des_max_3
159	des_zip_actual/med_7
160	card_merch_med_30
161	des_zip_actual/total_14
162	merch_state_avg_14
163	des_zip_total_14
164	merch_state_avg_0
165	card_state_actual/avg_0
166	merch_des_count_14

167	merch_des_max_7
168	des_zip_med_1
169	des_state_avg_14
170	card_merch_total_0
171	merch_state_actual/med_3
172	merch_state_max_14
173	card_des_actual/max_14
174	card_des_total_30
175	card_zip_max_7
176	Merch description_actual/max_30
177	card_des_actual/avg_7
178	card_state_actual/total_7
179	card_merch_count_1_by_30
180	card_state_actual/max_7
181	des_zip_count_14
182	card_merch_avg_3
183	card_zip_count_30
184	card_state_med_30
185	card_des_med_30
186	merch_des_total_7
187	card_merch_actual/med_7
188	card_des_actual/total_30
189	card_merch_max_0
190	Merch description_max_7
191	Merch description_med_14
192	merch_des_total_30
193	des_state_actual/max_3
194	des_state_actual/max_7
195	merch_des_avg_30
196	des_zip_avg_7
197	merch_des_actual/avg_7
198	card_zip_actual/max_0
199	Merch description_count_7
200	merch_state_actual/avg_14
201	card_zip_actual/med_0
202	card_zip_day_since
203	merch_des_count_0_by_7
204	merch_state_count_3
205	merch_state_actual/total_0
206	merch_des_actual/total_14
207	des_state_actual/total_14
208	Merch description_actual/avg_3
209	des_state_actual/med_30

210	card_zip_actual/med_30
211	card_des_med_7
212	Merch description_total_0
213	des_state_actual/med_3
214	card_des_actual/avg_30
215	merch_des_actual/total_30
216	merch_des_actual/med_3
217	card_merch_actual/avg_1
218	card_zip_total_0
219	card_zip_total_3
220	card_state_count_0
221	card_state_count_30
222	card_zip_count_1_by_14
223	Merch description_actual/total_0
224	card_state_actual/max_14
225	merch_des_actual/max_7
226	card_des_actual/total_7
227	merch_state_actual/avg_1
228	card_des_count_0_by_30
229	des_state_count_14
230	card_zip_avg_30
231	card_zip_count_14
232	des_zip_actual/avg_7
233	card_merch_count_0_by_7
234	card_merch_day_since
235	Merch description_total_7
236	merch_state_total_30
237	merch_des_avg_14
238	card_merch_count_7
239	merch_des_max_30
240	des_zip_actual/max_0
241	card_des_count_0_by_7
242	merch_des_count_1_by_14
243	merch_des_avg_7
244	card_zip_actual/avg_0
245	card_merch_total_3
246	card_zip_avg_7
247	des_state_actual/avg_7
248	des_state_max_30
249	card_merch_actual/avg_30
250	card_state_med_0
251	card_state_actual/total_0
252	card_state_count_3

253	des_state_total_0
254	card_zip_avg_3
255	des_state_max_1
256	card_zip_actual/total_30
257	Merch description_avg_7
258	card_state_actual/med_1
259	card_zip_count_1
260	card_zip_max_1
261	merch_state_count_30
262	card_merch_avg_7
263	card_zip_actual/total_0
264	card_state_actual/max_0
265	card_state_actual/med_30
266	des_zip_actual/total_7
267	card_zip_max_3
268	des_zip_total_1
269	des_state_avg_1
270	des_state_med_0
271	card_des_actual/max_1
272	Merch description_actual/med_14
273	card_des_avg_30
274	des_zip_actual/med_0
275	des_state_avg_3
276	card_zip_med_3
277	card_zip_total_1
278	merch_des_avg_0
279	Merch description_med_1
280	card_zip_actual/avg_14
281	card_zip_actual/max_3
282	des_state_count_3
283	des_zip_count_7
284	card_des_count_1_by_14
285	merch_des_actual/avg_30
286	card_state_actual/total_14
287	card_state_actual/total_1
288	des_state_actual/med_14
289	card_state_total_1
290	card_zip_actual/max_14
291	merch_des_med_1
292	card_zip_med_0
293	Merch description_count_1_by_14
294	card_state_med_1
295	des_state_actual/max_1

296	des_zip_count_0
297	card_state_actual/med_7
298	card_state_actual/total_30
299	merch_des_day_since
300	card_zip_avg_1
301	merch_state_actual/max_30
302	merch_state_med_1
303	des_state_actual/med_1
304	merch_state_actual/med_1
305	des_zip_total_0
306	card_des_count_1_by_30
307	card_state_avg_7
308	Merch description_max_1
309	card_merch_actual/total_7
310	merch_des_avg_1
311	card_zip_actual/avg_7
312	des_zip_actual/total_1
313	card_des_total_14
314	merch_des_actual/avg_3
315	des_zip_avg_0
316	card_des_med_1
317	card_merch_avg_14
318	card_state_actual/max_1
319	card_zip_actual/med_7
320	merch_state_actual/max_1
321	Merch description_actual/avg_0
322	card_state_actual/max_3
323	Merch description_actual/med_7
324	card_des_med_0
325	card_merch_actual/total_14
326	merch_state_max_7
327	des_state_actual/max_30
328	card_des_actual/max_7
329	card_state_actual/med_14
330	des_state_day_since
331	card_des_avg_14
332	merch_state_count_1_by_30
333	card_state_count_1_by_7
334	Merch description_max_0
335	card_state_count_1_by_30
336	card_zip_actual/med_14
337	des_state_actual/max_14
338	card_zip_max_30

339	card_merch_actual/total_3
340	merch_state_avg_7
341	merch_des_max_1
342	card_zip_total_7
343	Merch description_count_1
344	Merch description_avg_30
345	card_merch_max_30
346	card_des_actual/total_1
347	card_merch_count_1
348	card_merch_actual/med_14
349	Merch description_actual/total_14
350	merch_state_actual/max_7
351	merch_des_count_1_by_30
352	Merch description_count_0_by_30
353	merch_state_actual/med_0
354	card_merch_actual/max_7
355	Merch description_max_14
356	merch_des_med_30
357	card_zip_count_0
358	des_state_total_1
359	merch_des_count_0_by_30
360	Merch description_med_0
361	des_state_avg_7
362	des_zip_actual/max_3
363	card_merch_med_14
364	merch_des_actual/total_3
365	card_des_actual/total_3
366	merch_des_max_14
367	merch_state_count_0_by_7
368	des_zip_max_30
369	card_merch_med_1
370	merch_state_count_0_by_30
371	merch_state_actual/max_14
372	card_state_count_7
373	merch_des_count_3
374	card_zip_med_1
375	card_des_count_3
376	card_state_actual/avg_30
377	Merch description_actual/med_3
378	card_des_avg_3
379	des_zip_actual/avg_14
380	Merch description_actual/med_1
381	des_state_actual/avg_3

382	des_state_actual/total_3
383	merch_state_count_0_by_14
384	merch_state_actual/med_30
385	card_merch_max_7
386	Merch description_actual/total_7
387	des_state_med_3
388	merch_state_day_since
389	des_zip_med_0
390	Merch description_count_0_by_7
391	card_des_day_since
392	card_des_total_7
393	des_zip_max_1
394	card_merch_count_0
395	card_zip_max_0
396	Merch description_actual/avg_1
397	card_zip_actual/max_1
398	card_state_max_3
399	des_zip_max_3
400	merch_state_total_14
401	card_des_total_3
402	des_state_count_0
403	Merch description_med_7
404	card_des_avg_7
405	merch_state_actual/max_0
406	Merch description_max_3
407	des_state_total_7
408	Merch description_actual/max_3
409	card_merch_total_14
410	merch_des_med_14
411	Merch description_count_14
412	Merch description_max_30
413	card_merch_total_1
414	merch_des_count_30
415	merch_state_max_30
416	card_state_total_0
417	card_state_day_since
418	Merch description_count_0
419	card_merch_count_0_by_30
420	card_des_avg_1
421	card_state_actual/med_0
422	des_zip_actual/med_1
423	Merch description_actual/max_14
424	card_des_actual/avg_0

425	card_state_total_14
426	card_state_max_7
427	des_zip_avg_1
428	merch_state_total_3
429	card_merch_total_30
430	merch_des_total_3
431	merch_state_count_0
432	card_des_actual/med_7
433	card_zip_actual/avg_30
434	Merch description_actual/med_0
435	card_zip_actual/med_3
436	Merch description_actual/total_1
437	des_state_actual/total_0
438	merch_des_actual/max_3
439	card_merch_count_14
440	des_zip_med_14
441	merch_state_max_1
442	card_zip_count_1_by_30
443	card_des_actual/avg_1
444	card_state_total_7
445	des_zip_actual/med_14
446	merch_des_actual/avg_1
447	card_zip_total_14
448	card_des_max_1
449	des_state_actual/med_0
450	card_des_actual/max_30
451	card_zip_med_7
452	des_zip_med_7
453	Merch description_total_1
454	card_state_avg_14
455	Merch description_actual/avg_30
456	des_zip_actual/med_3
457	merch_state_total_7
458	card_state_max_30
459	card_merch_actual/max_30
460	card_zip_actual/avg_1
461	card_merch_count_30
462	Merch description_count_0_by_14
463	card_state_med_14
464	Merch description_med_3
465	card_state_count_0_by_7
466	merch_des_total_1
467	merch_state_avg_1

468	card_zip_avg_14
469	card_zip_actual/total_1
470	Merch description_actual/med_30
471	card_zip_avg_0
472	des_state_actual/total_1
473	card_merch_actual/total_0
474	merch_des_actual/max_1
475	des_zip_actual/avg_1
476	merch_des_count_1_by_7
477	card_merch_actual/avg_14
478	merch_state_total_0
479	card_merch_count_1_by_14
480	merch_state_count_1_by_14
481	des_zip_max_7
482	merch_des_actual/total_0
483	card_des_actual/med_0
484	merch_des_count_1
485	Merch description_total_3
486	card_state_total_3
487	card_state_med_3
488	card_des_max_30
489	des_state_actual/avg_0
490	des_state_count_1
491	card_des_med_14
492	card_merch_actual/med_30
493	card_state_actual/avg_7
494	des_state_actual/med_7
495	des_state_max_0
496	merch_state_avg_30
497	card_des_actual/max_0
498	card_merch_actual/med_3
499	card_merch_avg_30
500	card_des_count_1_by_7
501	card_merch_actual/total_1
502	card_zip_actual/max_7
503	merch_state_actual/max_3
504	Merch description_actual/max_1
505	card_merch_actual/avg_7
506	card_merch_count_3
507	merch_des_med_3
508	merch_des_med_0
509	merch_des_actual/med_30
510	des_zip_avg_30

511	des_zip_max_14
512	merch_des_count_7
513	card_des_count_30
514	merch_state_med_7
515	Merch description_avg_1
516	card_des_max_3
517	dow_risk
518	card_state_avg_3
519	des_zip_count_30
520	des_zip_actual/max_30
521	card_state_actual/med_3
522	card_merch_avg_1
523	card_zip_actual/max_30
524	card_zip_count_7
525	card_zip_total_30
526	merch_state_actual/med_7
527	des_zip_total_3
528	card_des_count_0_by_14
529	merch_des_actual/total_1
530	des_state_actual/avg_30
531	Merch description_actual/avg_7
532	card_state_count_0_by_14
533	card_des_actual/med_30
534	card_des_max_7
535	des_zip_count_1
536	merch_state_count_7
537	card_des_actual/avg_14
538	card_merch_max_1
539	card_zip_actual/avg_3
540	merch_des_actual/max_30

Table 16 All created variables