Credit Card Transaction Fraud Analysis Report

Chenyuan Liu

Jiaozhe Xu

Kaicheng Yang

Xiaoxian Zhang

Yunsi Chen

June 8th, 2022

Table of Content

Executive Summary

Fraud has impacted nearly every corner of business – from data breaches that affect end customers' privacy rights and payment security to ransom attacks that demand vast sums of money from organizations. When not dealt with proactively, fraud can affect companies' bottom lines by pulling resources from the core business and priorities, damaging brand reputation, and squandering profits.

The objective of our project is to identify and capture fraudulent transactions by training and developing machine learning models. Although we might not be able to capture all the potential fraud transactions, we believe our model could greatly reduce the fraud events and take proactive action by leveraging the supervised model that our team built.

The purpose of this report is to provide detailed information on the entire modeling process, including the data preparation, development process, and result discussion. Our team has generated 804 variables and filtered the top 30 variables in our final modeling procedure. Our final model is the LightBGM model which gets 55.55 for the OOT FDR @3%. The following are the specific steps we took during the modeling process: data preparation & data cleaning, feature creation, feature engineering, feature selection, modeling, and result discussion.

Data Description

The data is actual credit card purchases from a US government organization. The time frame of this dataset is from 2006-01-01 to 2006-12-31, 12 months in total. The dataset includes card number, date, merchandise description, merchandise state, merchandise state, merchandise zip, transaction type, amount, fraud identifier, etc.

- Number of fields: There are 10 columns/fields.
- Number of records: There are 96,753 records.

Summary Tables:
Among the 10 fields, there are two numeric(Datetime) fields and eight categorical fields.

Table for Numeric or Date Time：

Field Name	% Populated	Min	Max	Mean	Std dev	% Zero
Date	100%	2006-01-01	2006-12-31	NA	NA	0%
Amount	100%	0.01	3102045.53	427.89	10006.14	0%

Figure 1

Table for Categorical Fields：

Field Name	% Populated	# Unique Values	Most Common Value
Recnum	100.00%	96,753	NA
Cardnum	100.00%	1,645	5142148452
Merchnum	96.51%	13,092	930090121224
Merch description	100.00%	13,126	GSA-FSS-ADV
Merch state	98.76%	228	TN
Merch zip	95.19%	4,568	38118.0
Transtype	100.00%	4	P
Fraud	100.00%	2	0

Figure 2

Important Fields used in the following analyses:

1. Fraud Label

There are 2 unique values in this field: 0 and 1, indicating whether each of the records is a fraudulent transaction (1) or a non-fraudulent transaction (0). As the histogram shows below, the dataset is obviously unbalanced, as there are 1,059 fraudulent transactions and 95,684 non-fraudulent transactions.

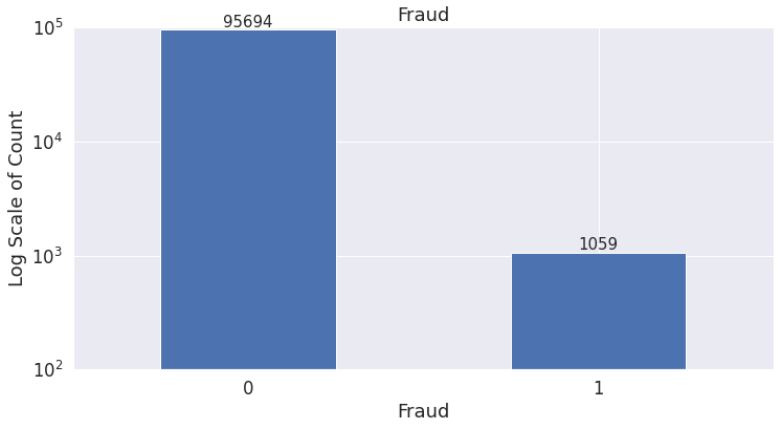


Figure 3

2. Date

The field is the date of each transaction. The time frame is from 2006-01-01 to 2006-12-31. There are 365 unique entries. The most common date in this dataset is 2006-02-28.

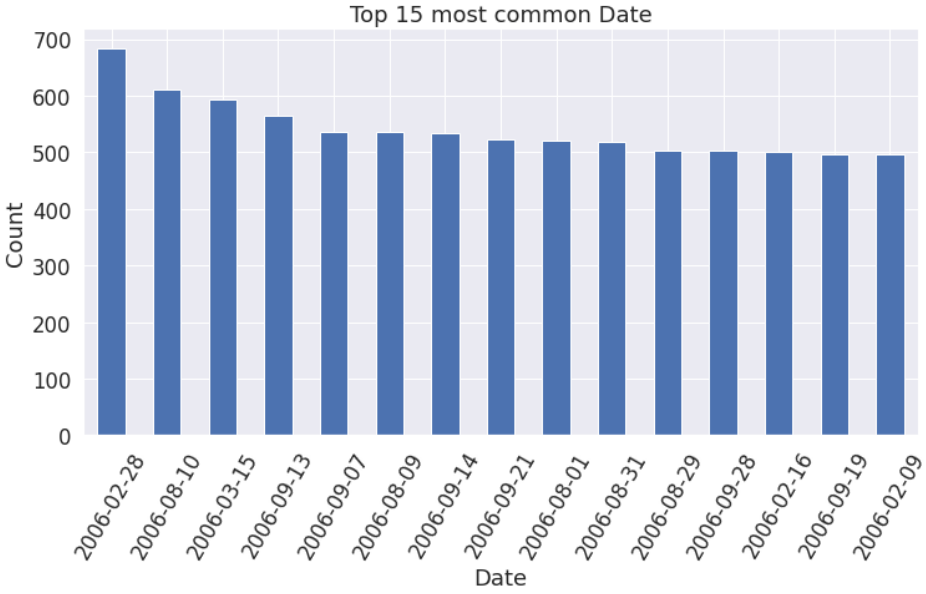


Figure 4

To better identify the pattern of transactions, we plotted the count of transactions across the day, week, and month. We found the dataset has spikes on weekends because of the nature of this dataset that it's the transaction data of a US government organization. And the monthly volume drops significantly in September because, for the Federal government, the fiscal year runs from October 1st to September 30th.

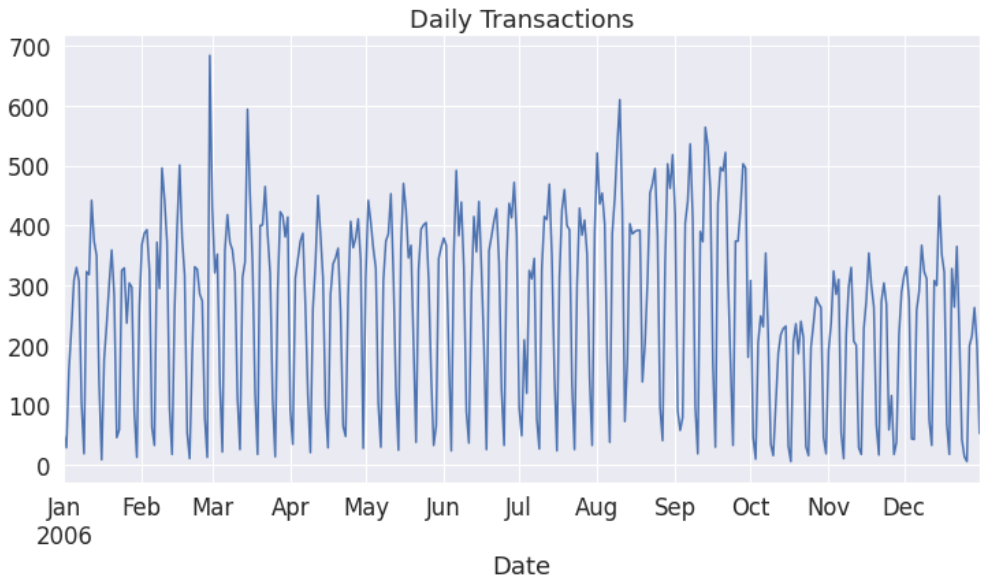


Figure 5

Figure 6

3. Merchnum

The field records the merchandise number of each transaction. There are 13,092 unique entries. The most common value is 930090121224 which appears 9,310 times in the dataset. The second most common Merchnum is blank which represents the missing value so that the %populated of this field is 96.51%. So we filled in the missing values later when preparing the dataset for feature engineering.

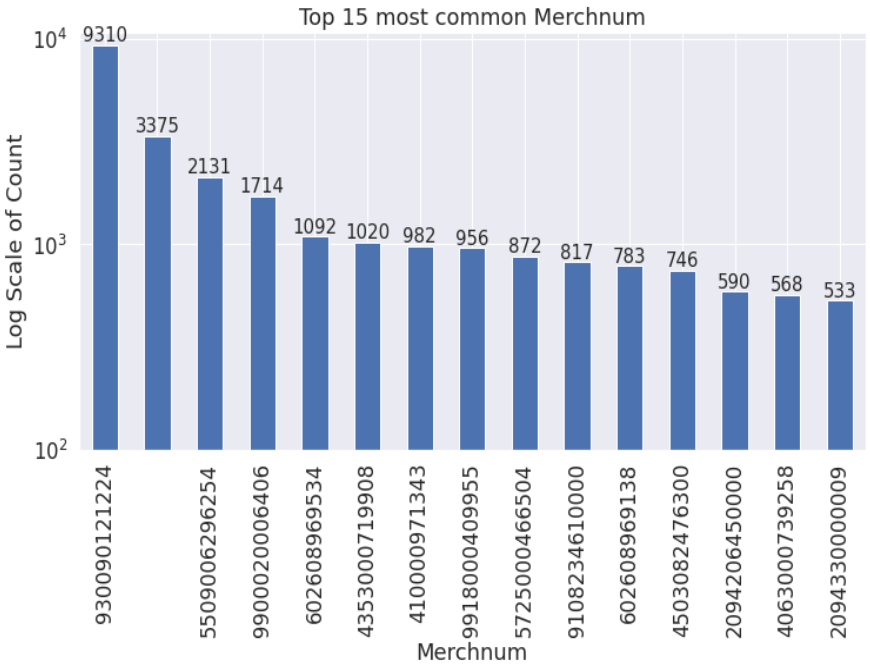


Figure 7

4. Merch description

The field contains descriptions of each transaction. There are 13,126 unique entries in this field. The most common value is GSA-FSS-ADV which appears 1,688 times in the dataset.

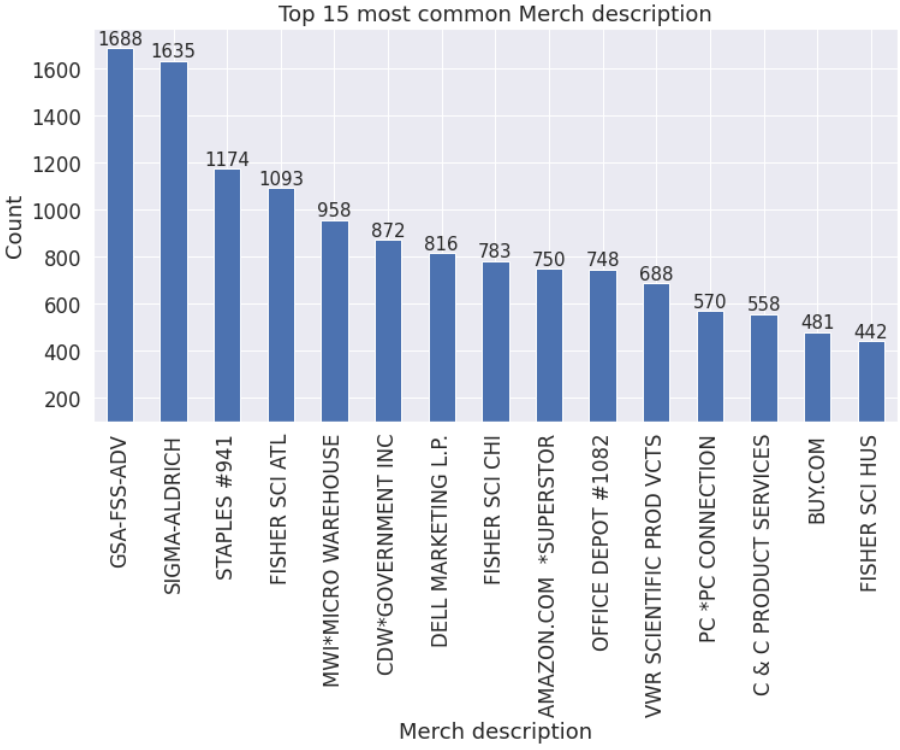


Figure 8

5. Merch state

The field records the state information of each transaction, numbers representing international locations, or bad data. There are 228 unique values in this field. The most common value is TN which appears 12,035 times in the dataset.

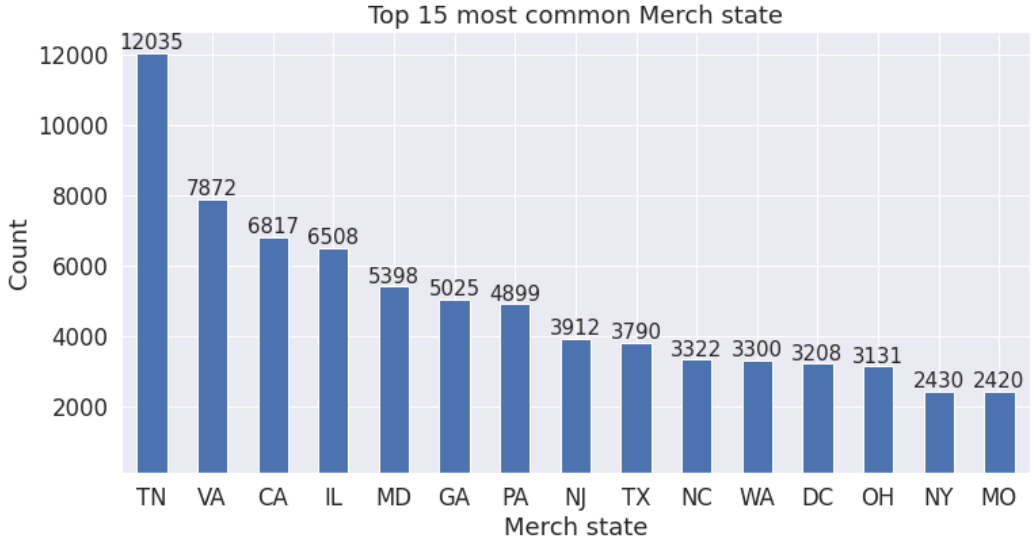


Figure 9

6. Merch zip

The field records the zip code of each transaction. There are 4,568 unique values in this field. The most common value is 38118 which appears 11,868 times in the dataset.

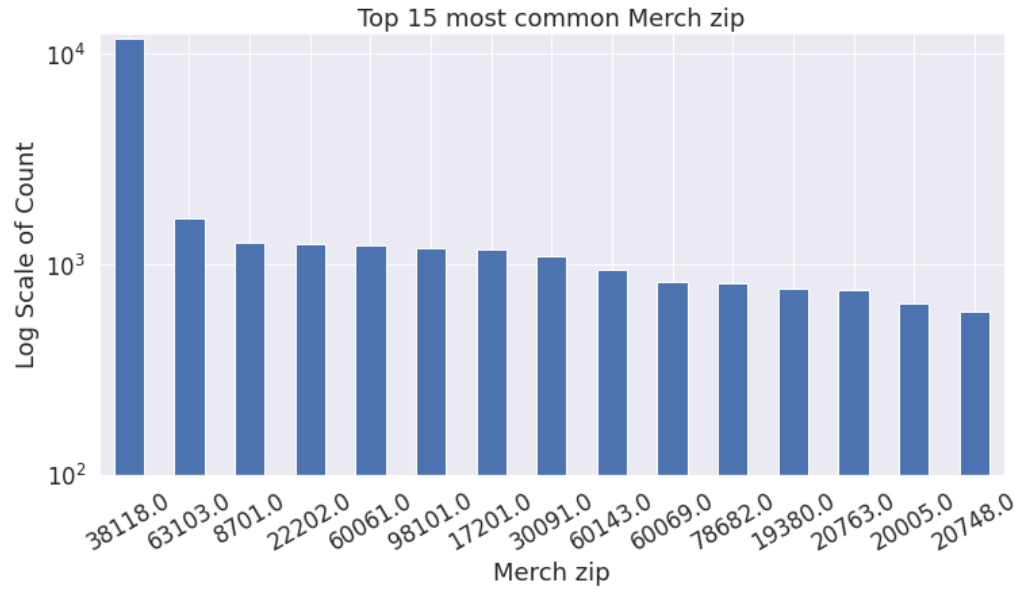


Figure 10

7. Amount

The field records the amount of each transaction. There is an obvious drop point at the amount of $2,500. We suggest this because the credit limit for some cards is $2,500.

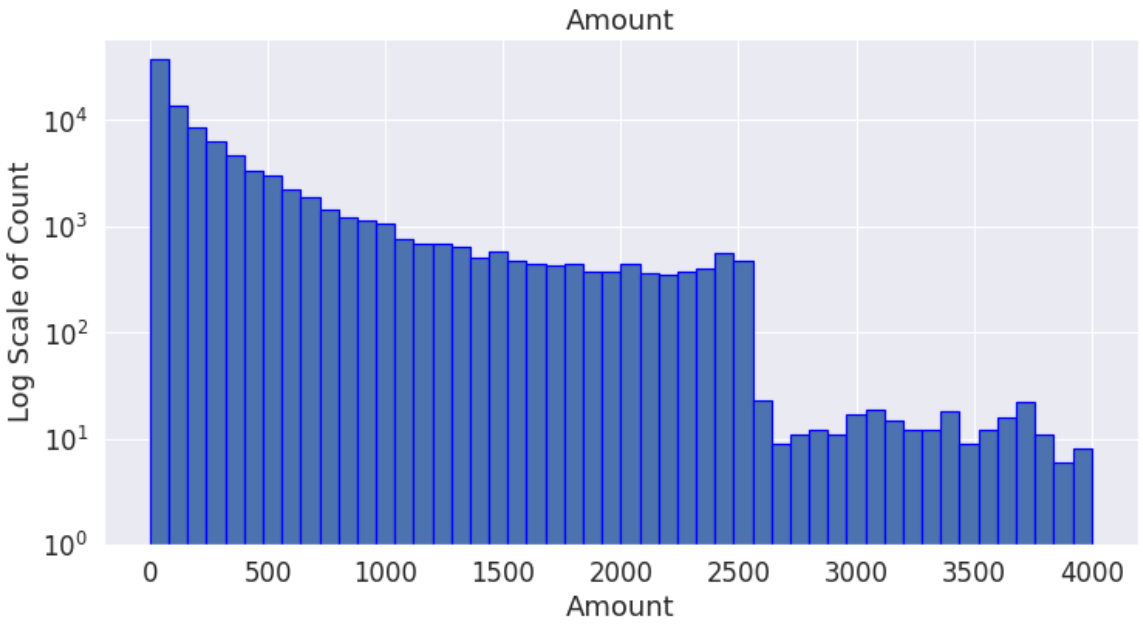


Figure 11

Data Cleaning

After taking a look at all the records and fields from the dataset, we found some records with unreasonable values and missing values. We then started to perform data cleaning by removing outliers and filling missing values.

Missing Values

There are 1,195 missing values for Merch state, and 4,656 missing values for Merch zip. For Merchnum, it has no missing values but has lots of values as zero. We need to fill missing values for Merch zip and Merch state and replace the zero values for Merchnum.

Filling Missing Values

- Merch state:
  1. For records that did not have a “Merch state” value but had a “Merch zip” value, we created a dictionary of states relative to their United States zip codes to fill these instances.
  2. The records that had a Puerto Rican zip code were filled with “PR”.
  3. We used the mode of the Merchnum or Merch description to fill some records.
  4. There were lots of unknown fields, we filled them with “Unknown”.

- Merchnum:
  1. There were many fields that were “0”, so we replaced them with “NaN”.
  2. We filled part of the null values with the mode of Merch description.
  3. We filled the remaining missing values with “Unknown”.

- Merch zip
  1. We filled in missing values with the mode of “Merchnum” and then “Merch description”.
  2. We replaced the remaining missing values with “Unknown”.

Removing Outliers

There was an outlier in which the purchase amount was $3,102,045.53 and amounts of all other records were well under $50,000, so we removed it. We also only kept records with value types of “P” and there were 96,397 records left.

Feature Engineering

Variable types	Number of variables
Amount variables	560
Frequency variables	70
Days-since variables	10
Velocity change variables	160
Risk variables	2
Benford's law variables	2
Total	804

Chart 1

A total of 804 expert variables were created based on the 10 entities, which include velocity variables, relative velocity variables, amount variables, days since last seen variables, risk variables, and Benford's law variables. Entities are fields we used to group and link the records as shown below:

1. Cardnum - card number (Card)
2. Merchnum - merchant number (Merchant)
3. card_merch - card number + merchant number (Card at this merchant number)
4. card_zip - card number + zip (Card in this zip code)
5. merch_zip - merchant number + zip (Merchant in this zip code)
6. zip3 - first 3 digits of zip
7. card_zip3 - card number + zip3 (Card in this first 3 digits of zip code)
8. Card_merchdesc - card number + merchant description (card at this merchant description)
9. Merchnum_desc - merchant number + merchant description (merchant number at this description)
10. Card_Merchnum_desc - Card number + merchant number + merchant description (card at this merchant number and description)

Amount Variables

Amount variables employ eight measures of the amount by each of the entities over the past number of days:

1. Average amount
2. Maximum amount
3. Median amount
4. Total amount

5. Actual amount divided by average amount
6. Actual amount divided by maximum amount
7. Actual amount divided by median amount
8. Actual amount divided by total amount

The amount variables are useful in identifying transactions of abnormal amounts. As an example, in the case where a particularly large amount appeared, it would skew the average and total amount upward. A total of 560 amount variables are created across the 10 entities.

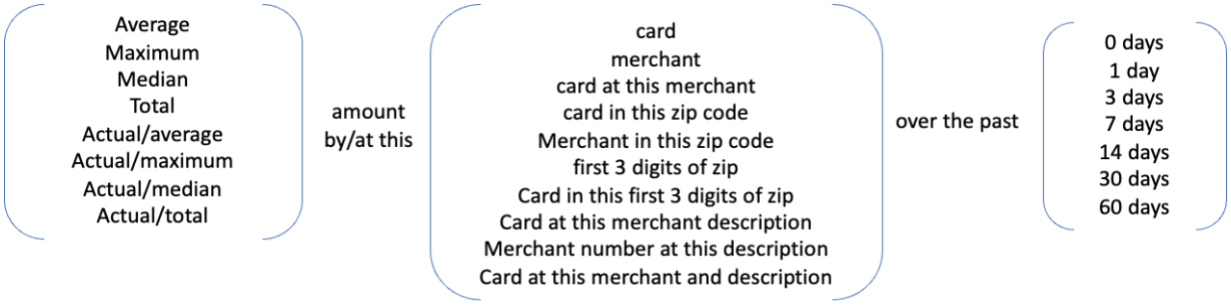


Figure 12

Frequency Variables

Frequency variables indicate the number of occurrences for each value seen over the past certain number of days. A total of 70 velocity variables were created based on different combinations of entities and the number of day counts. All 10 entities were considered and the number of occurrences for each value across the 10 entities in the previous 0, 1, 3, 7, 14, 30, and 60 days were calculated.

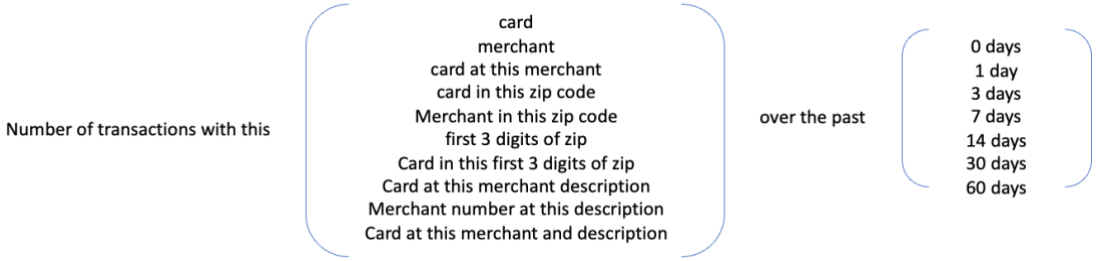


Figure 13

Days-since Variables

The days-since variables measure the time since a transaction has been made by an entity. They were calculated by subtracting the most recent transaction with the same entity from the date of the current transaction. 365 days were used for days-since if it's the first time seen. A total of 10 days-since variables are created.

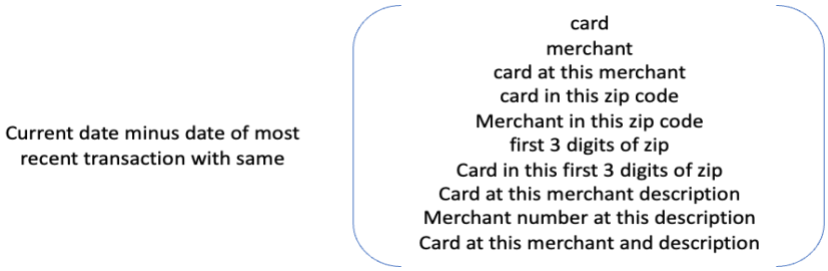


Figure 14

Velocity change Variables

These velocity change variables seek to identify burst buying behavior associated with stolen credit cards. They were calculated by the number of transactions with the same entity over a time period (0, 1 day), divided by the average number of transactions with the same entity over a longer time period (7, 14, 30, 60 days). The relative velocity variables are good indicators of fraudulent credit cards where fraudsters make a burst of purchases in a short time frame. A total of 160 relative velocity variables are created.

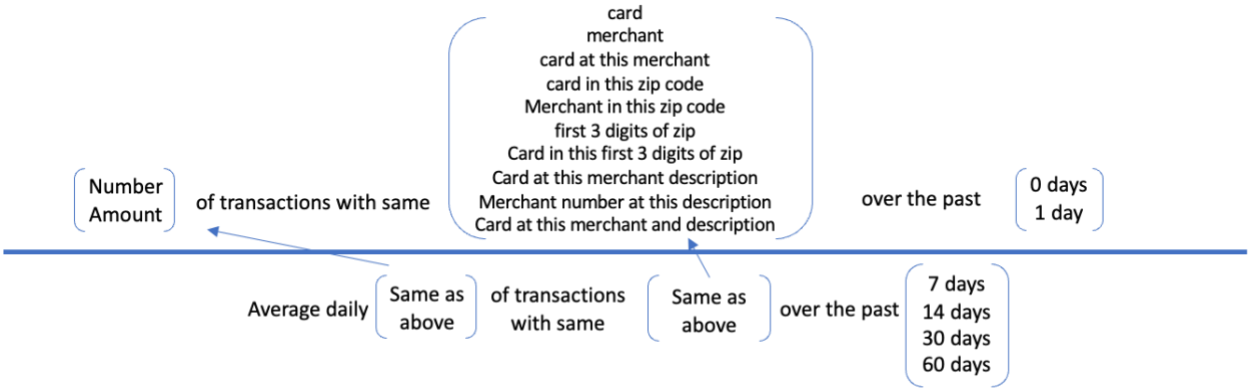


Figure 15

Risk variable

Target encoding variables, in general, entail assigning a particular value (in our case the average) relating the dependent variable to each level of a categorical variable. We used training data to construct the risk tables, which are records before November 1, 2010. In addition, a smoothing formula is applied such that the value is less variant and less affected by values calculated based on statistically insufficient observations in each category.

$$\text{Value} = Y_{\text{low}} + \frac{Y_{\text{high}} - Y_{\text{low}}}{1 + e^{-(n - n_{mid})/c}}$$

Equation 1

1) Day of Week

For each day of the week, the average fraud rate is computed for all records using training data.

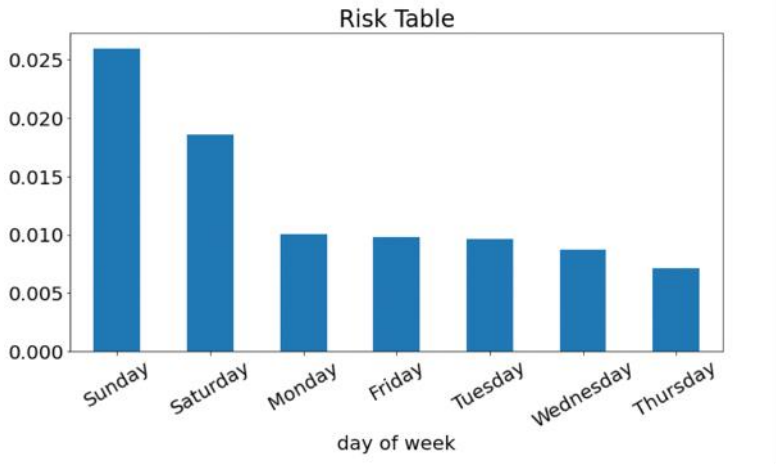


Figure 16

2) State

For each state, the average fraud rate is computed for all records using training data. States with fewer than 500 records are combined together as 'others'.

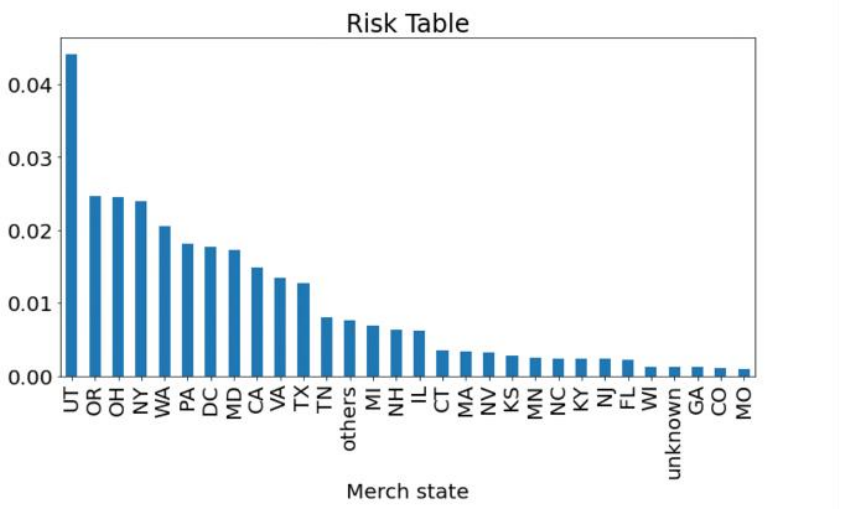


Figure 17

Benford's law variables

We then used Benford's Law – the distribution of the first digit of many entities is not uniform. For instance, the first digit "1" appears in about 30% of all the records. If somebody just makes transactions, they usually don't know about Benford's law, so the transaction amounts are uniformly distributed in random numbers. Hence, we can look at the amount distributions for each cardholder and merchant in the case that a merchant makes up many fraudulent purchase amounts or a fraudulent financial person makes up many amounts. If the amount distributions substantially violate Benford's Law, we label them as fraud.

Feature Selection

After creating features, we got 804 variables and one possible problem we are facing is high dimensionality. High dimensionality makes data become sparse very quickly. Data points not intuitively turn to outliers so that we need exponentially more data to see true nonlinearities rather than noise. According to the reasons above, the higher the dimensions, the more likely a nonlinear model will accidentally fit noise rather than true shape. Therefore, it is super important to decrease dimensions because a lower dimension will enable the nonlinear model to run faster to optimize model architecture and hyperparameters. One way to decrease dimension is feature selection.

Feature selection methods are able to reduce the dimensions without losing much information. In other words, feature selection helps us only keep variables with substantial information, and minimize the number of variables at the same time.

For this project, we first utilized the filter to sort the importance of all variables and then used the wrapper on the remaining top variables and selected the final 30 variables.

Filter

Filter feature selection is one way to conduct feature selection. It sorts all variables by their importance for predicting the dependent variable. The filter is independent of any modeling method. As our project is a binary classification problem, we used Kolmogorov-Smirnov(KS) score to calculate the rank of all variables.

Kolmogorov-Smirnov(KS)

Generally, the Kolmogorov–Smirnov statistic quantifies a distance between the empirical distribution function of the sample and the cumulative distribution function of the reference distribution, or between the empirical distribution functions of two samples. In this particular case, For each candidate variable, KS plots the goods and bads separately, the distance between the two curves is the KS statistic. The more different the curves the better the variable for separating, the bigger the KS score, and thus the more significant the variable. The KS score is calculated based on the following equation.

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

Equation 2

Wrapper

Unlike filter feature selection, which does not consider subsequent learners, the wrapper directly uses the performance of the final model to be used as the evaluation criterion for selecting features, that is, the purpose of the wrapper is to select the final list of features that are most beneficial to its performance for a given model. And this is why we use wrapper after the filter.
We used the forward selection for the wrapper. The forward selection first builds a lot of separate 1-d models, keeps the best variable, and then continues to add one more variable, until there is no performance when adding a new variable.
As for the model performance metric, we used the FDR as the evaluation metric. FDR(Fraud detection rate) describes how many frauds we can catch within a certain population. In a generic FDR, a machine learning model gives ranking to all records based on the probability of being a fraud, and the records in the given population bin, usually top 3%-5%, is considered as predicted fraud. The FDR is calculated to be the number of true frauds in the bin, which are correctly recognized by the model, divided by the total number of true frauds in the entire dataset.
After trying the wrapper several times, a total of 30 variables were chosen to be used for modeling. The final features are listed below.

List of variable and variable definition

- Card_zip3_total_7

Total transaction amount for each card number and top 3 digits of zip combinations over the past 7 days.

- Card_Merchnum_desc_max_60

Maximum transaction amount for each card number, merchant number, and merchant description combination over the past 60 days.

- Merchnum_total_1

Total amount of transactions for each merchant number over the past 1 day.

- Card_Merchnum_desc_total_60

Total amount of transactions for each card number, merchant number, and merchant description combination over the past 60 days.

- Cardnum_total_7

Total amount of transactions for each card number over the past 7 days.

- Card_Merchdesc_max_60

Maximum amount of transactions for each card, merchant description over the past 60 days.

- Cardnum_avg_0

Average amount of transactions for each card in one day.

- Card_Merchnum_desc_avg_60

Average amount of transactions for each card, merchant number, and merchant description combination over the 60 days.

- Card_zip_total_3

Total amount of transactions for each card over the past 3 days.

- Merch_zip_total_1

Total amount of transactions for each merchant and merchant zip combination over the past 1 day.

- Card_zip_total_7

Total amount of transactions for each card and card zip number combination over the past 7 days.

- Merchnum_avg_0

Average transaction amount for each merchant number over the past 0 days.

- Card_Merchdesc_total_7

Total amount of transactions for each card and card merchant description combination over the past 7 days.

- Card_merch_total_60

Total amount of transactions for each card and merchant number combination over the past 60 days.

- Card_Merchnum_desc_total_7

Total amount of transactions for each card and merchant and merchant description combination over the past 7 days.

- Merchnum_desc_avg_0

Average amount of transactions for each merchant and merchant description combination over the past 0 days.

- Cardnum_med_0

Median amount of transactions for each card over the past 0 days.

- Merch_zip_avg_0

Average amount of transactions for each merchant and merchant number combination over the past 0 days.

- Card_zip3_max_1

Maximum amount of transactions for each card and card top3 digits zip number combination over the past 1 day.

- Card_zip3_max_0

Maximum amount of transactions for each card and card top3 digits zip number combination over the past 0 days.

- Card_zip3_med_0

Median amount of transactions for each card and card top3 digits zip number combination over the past 1 day.

- Card_zip_max_1

Maximum amount of transactions for each card and card zip number combination over the past 1 day.

- Card_Merchnum_desc_max_0

Maximum amount of transactions for each card and merchant and merchant description combination over the past 0 days.

- Card_zip3_max_3

Maximum amount of transactions for each card and card top3 digits zip number combination over the past 3 days.

- Card_Merchdesc_max_0

Maximum amount of transactions for each card and merchant description combination over the past 0 days.

- Card_Merchnum_desc_max_1

Maximum amount of transactions for each card and merchant and merchant description combination over the past 1 day.

- Card_merch_avg_60

Average amount of transactions for each card and merchant combination over the past 60 days.

- Card_zip3_total_0

Total amount of transactions for each card and card top3 digits zip number combination over the past 0 days.

- Card_zip_max_0

Maximum amount of transactions for each card and card zip number combination over the past 0 days.

- Card_merch_max_0

Maximum amount of transactions for each card and merchant combination over the past 0 day.

The chart below shows the filtered 30 variables and their KS score.

wrapper order	variable	filter score
1	card_zip3_total_7	0.696
2	Card_Merchnum_desc_max_60	0.639
3	Merchnum_total_1	0.607
4	Card_Merchnum_desc_total_60	0.647
5	Cardnum_total_7	0.600
6	Card_Merchdesc_max_60	0.641
7	Cardnum_avg_0	0.570
8	Card_Merchnum_desc_avg_60	0.588
9	card_zip_total_3	0.678
10	merch_zip_total_1	0.605
11	card_zip_total_7	0.685
12	Merchnum_avg_0	0.583
13	Card_Merchdesc_total_7	0.671
14	card_merch_total_60	0.643
15	Card_Merchnum_desc_total_7	0.666
16	Merchnum_desc_avg_0	0.583
17	Cardnum_med_0	0.557
18	merch_zip_avg_0	0.580
19	card_zip3_max_1	0.629

20	card_zip3_max_0	0.605
21	card_zip3_med_0	0.563
22	card_zip_max_1	0.626
23	Card_Merchnum_desc_max_0	0.601
24	card_zip3_max_3	0.657
25	Card_Merchdesc_max_0	0.604
26	Card_Merchnum_desc_max_1	0.621
27	card_merch_avg_60	0.587
28	card_zip3_total_0	0.615
29	card_zip_max_0	0.604
30	card_merch_max_0	0.601

Chart 2

Modeling

We used the top 30 variables from our feature selection process as independent variables and a fraud label field as our target or dependent variable, which indicates if the transaction is fraudulent (1) or non-fraudulent (0). Then, we used records from November 1st to December 31st as OOT (out-of-time) validation data set. We trained the model on the rest of the data records by ten-fold manual cross-validation to address overfitting problems.

Logistic regression

We started our model exploration with logistic regression as a baseline. Logistic regression is a linear and simple method for binary classification and it is also easy to modify and interpret. Logistic regression uses maximum likelihood to fit the logistic function and gives outputs of the probability of fraud.

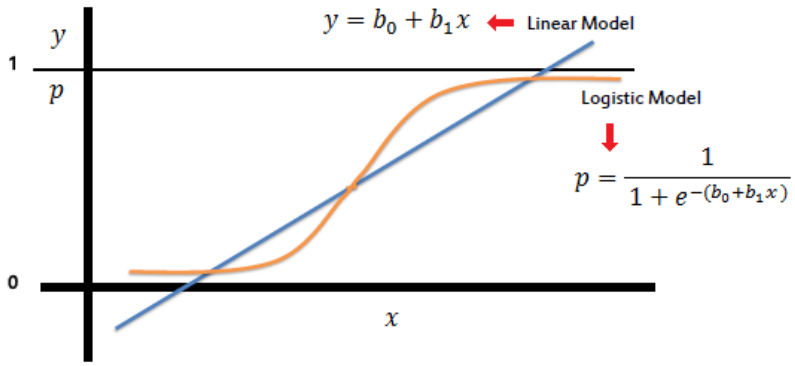


Figure 18

Hyperparameter tuned:

- C: inverse of regularization strength
- penalty: regularization type: l1, l2, elastic net or none
- solver: optimization algorithm

Performance:

	Parameters					Avg FDR at 3%		
	iteration	#variables	penalty	C	solver	train	test	oot
Logistics Regression	1	15	l2	1	lbfgs	0.679	0.678	0.346
	2	15	l2	0.1	lbfgs	0.675	0.668	0.341
	3	15	l2	10	lbfgs	0.688	0.667	0.348
	4	30	l2	1	lbfgs	0.69	0.679	0.355
	5	30	l2	0.1	lbfgs	0.679	0.674	0.351
	6	30	l2	10	lbfgs	0.692	0.682	0.362
	7	30	l1	1	liblinear	0.691	0.681	0.357
	8	30	l2	1	liblinear	0.687	0.684	0.354
	9	30	l2	10	liblinear	0.692	0.683	0.362
	10	30	l2	100	liblinear	0.694	0.678	0.364

Figure 19

Decision Tree

A decision tree is a tree-like structure whereby an internal node represents an attribute, a branch represents a decision rule, and the leaf nodes represent an outcome. This works by splitting the data into separate partitions according to an attribute selection measure, which in this case is the Gini index or information gain.

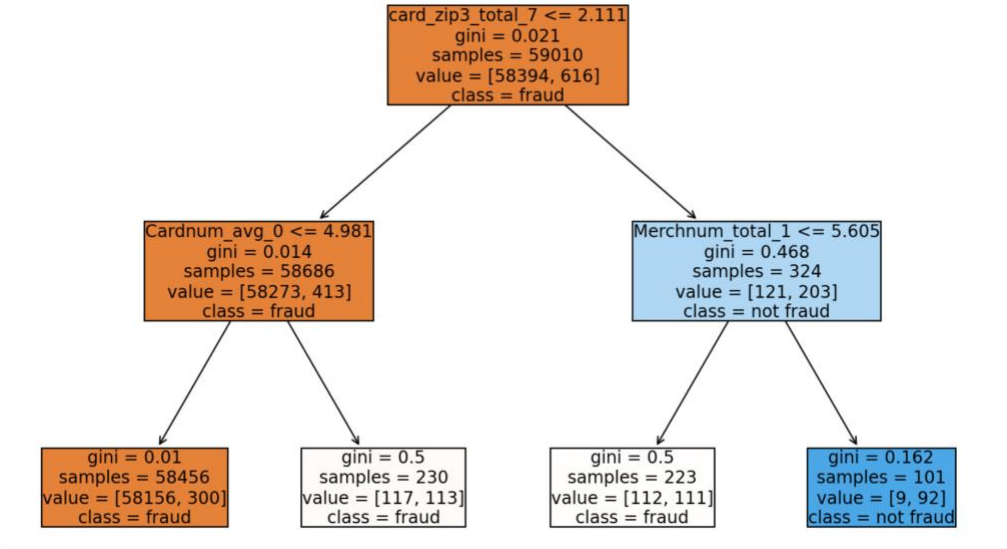


Figure 20

Hyperparameter tuned:

- criterion: The function to measure the quality of a split.
- splitter: The strategy used to choose the split at each node.
- max_depth: The maximum depth of the tree.
- min_samples_split: The minimum number of samples required to split an internal node.
- min_samples_leaf: The minimum number of samples required to be at a leaf node.

Performance:

	Parameters							Avg FDR at 3%		
	iteration	#variables	criterion	max_features	max_depth	min_samples_split	min_samples_lea	train	test	oot
Decision Tree	1	15	gini	None	10	100	60	0.794	0.725	0.45
	2	15	gini	None	8	100	60	0.754	0.717	0.474
	3	15	gini	8	8	100	60	0.749	0.691	0.446
	4	15	entropy	None	8	100	60	0.808	0.746	0.439
	5	15	gini	None	10	1000	500	0.659	0.657	0.403
	6	30	gini	None	10	100	60	0.806	0.747	0.465
	7	30	gini	8	10	100	60	0.795	0.728	0.454
	8	30	entropy	None	10	100	60	0.849	0.758	0.432
	9	30	gini	None	10	100	60	0.805	0.734	0.399
	10	30	gini	None	10	1000	500	0.676	0.644	0.368

Figure 21

Random Forest

Random forests is an ensemble learning method for classification that operates by constructing a multitude of decision trees at training time. For classification tasks, the output of the random forest is the class selected by most trees.

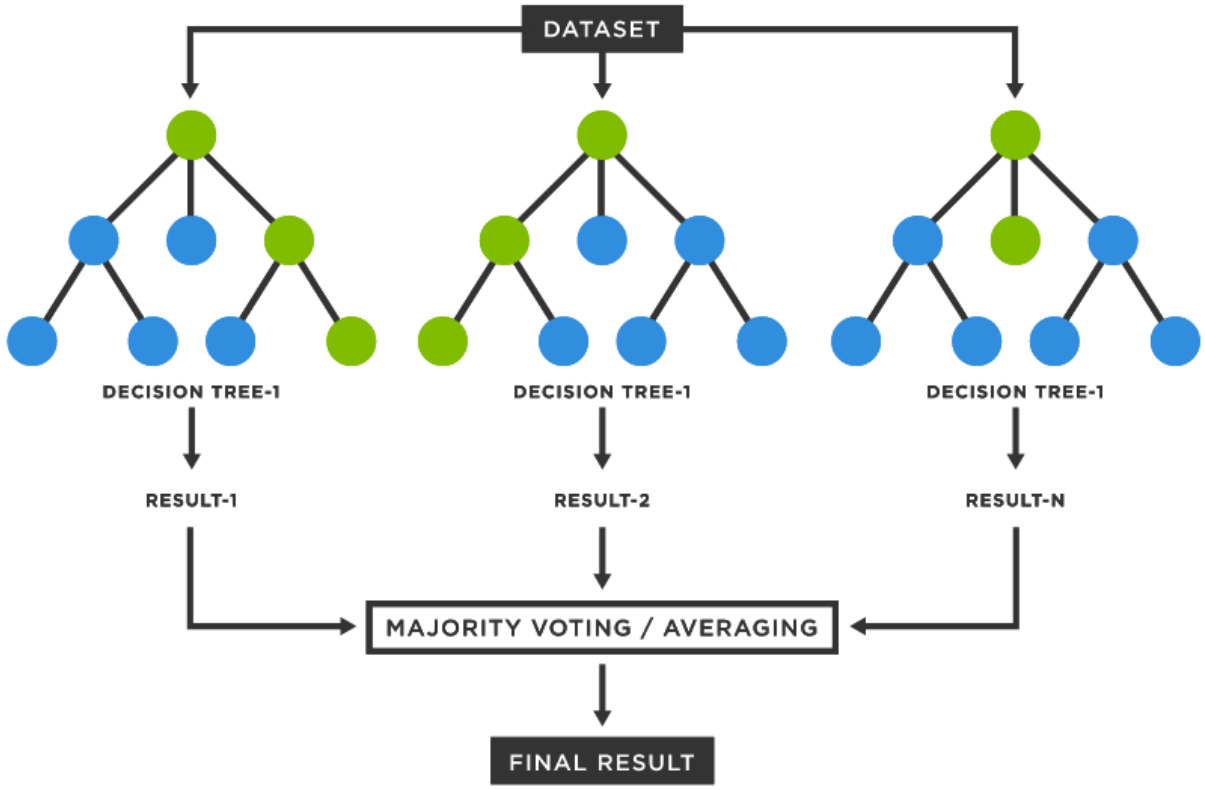


Figure 22

Hyperparameters tuned:

- n_estimators: The number of trees in the forest. (default = 100)
- max_features: The number of features to consider when looking for the best split. (default = ”sqrt”)
- max_depth: The maximum depth of the tree. If None, then nodes are expanded until all leaves are pure or until all leaves contain less than min_samples_split samples. (default = None)
- min_samples_split: The minimum number of samples required to split an internal node (default = 2)
- min_samples_leaf: The minimum number of samples required to be at a leaf node. (default = 1)

Performance:

		Parameters						Average FDR at 3%		
	iteration	#variables	n_estimators	max_features	max_depth	min_samples_split	min_samples_leaf	train	test	oot
Random Forest	1	15	40	10	6	120	100	0.740	0.721	0.468
	2	15	50	10	10	120	100	0.802	0.775	0.509
	3	15	100	15	10	120	100	0.796	0.767	0.501
	4	15	100	15	10	150	150	0.790	0.756	0.503
	5	15	500	15	20	120	120	0.821	0.794	0.523
	6	10	500	10	20	120	120	0.818	0.786	0.404
	7	30	50	6	10	120	120	0.799	0.773	0.472
	8	30	100	6	10	120	120	0.804	0.776	0.494
	9	30	50	10	15	120	120	0.817	0.782	0.501
	10	30	100	10	20	120	120	0.816	0.783	0.513

Figure 23

LightGBM

LightGBM is a gradient boosting framework that is based on tree-based learning algorithms. Unlike other boosting tree models, the LightGBM model splits the tree leaf-wise instead of level-wise and grows the leaf that has the highest delta loss. The figure below shows how the leaf-wise tree growth works. The LightGBM model will cause overfitting very easily with small datasets, but for datasets that have over 10,000 records like our data, it will perform well.

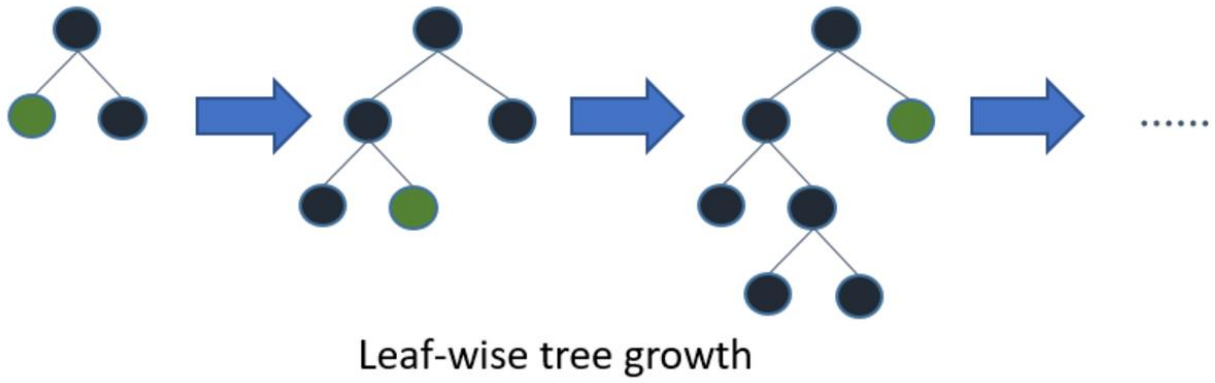


Figure 24

Hyperparameters tuned:

- n_estimators: Number of boosted trees to fit (with default=100).
- num_leaves: Maximum tree leaves for base learners (with default=31).
- max_depth: Maximum tree depth for base learners, <=0 means no limit (with default=-1).
- learning_rate: Boosting learning rate (with default=0.1).
- boosting_type: Has different options which are ‘gbdt’ which stands for traditional Gradient Boosting Decision Tree, ‘dart’ stands for Dropouts meet Multiple Additive Regression Trees, ‘goss’ which stands for Gradient-based One-Side Sampling, and ‘rf’ which stands for Random Forest (‘gbdt’ is the default).

Performance:

		Parameters					Average FDR at 3%		
	Iteration	#variables	n_estimators	num_leaves	max_depth	learning_rate	train	test	oot
LightGBM	1	10	50	3	3	0.01	0.668	0.654	0.336
	2	10	100	10	3	0.01	0.757	0.730	0.401
	3	10	300	40	10	0.01	0.977	0.837	0.516
	4	10	500	40	10	0.1	1.000	0.831	0.417
	5	10	1000	30	15	0.1	1.000	0.843	0.385
	6	15	100	50	6	0.01	0.904	0.815	0.555
	7	15	300	100	6	0.01	0.949	0.856	0.523
	8	15	500	100	10	0.1	1.000	0.852	0.409
	9	15	1000	100	15	0.1	1.000	0.858	0.432
	10	15	1000	300	20	0.1	1.000	0.848	0.473

Figure 25

K-Nearest Neighbor (KNN)

K-Nearest Neighbor is a supervised machine learning algorithm that is mainly used to solve classification problems. Based on how close a data point is to each group, the algorithm will make the prediction on the likelihood of the data point becoming part of each group and then assign it to the appropriate group. KNN calculates the distance between the target data point and the nearest data point based on the choice of K. The most commonly used distance calculated function is shown in the figure below which is called Euclidean distance.

$$d(x,y) = \sqrt{\sum_{i=1}^{n} (y_i - x_i)^2}$$

Equation 3

The figure below shows that when K=3, the target data point is more likely to be assigned to Class B since there are two out of three nearest data points are in Class B, but if K=7, then the target data point is more likely to be assigned to Class A since there are four out of seven nearest data points are in Class A.

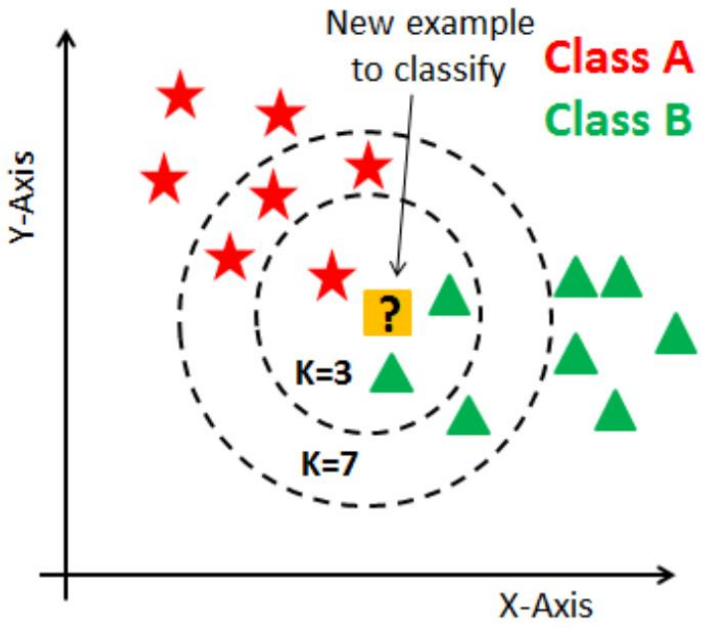


Figure 26

Hyperparameters tuned:

- n_neighbors: Number of neighbors to use for k-neighbors queries (with default=5).
- leaf_size: Leaf size passed to BallTree or KDTree (with default=30).
- weights: Weight function used in prediction (with default='uniform', can also be 'distance' or [callable]).
- algorithm: Algorithm used to compute the nearest neighbors (with default='auto', can also be 'ball_tree', 'kd_tree' or 'brute').

Performance:

		Parameters					Average FDR at 3%		
	Iteration	#variables	n_neighbors	leaf_size	weights	algorithm	train	test	oot
K-Nearest Neighbour	1	10	50	30	uniform	auto	0.842	0.804	0.445
	2	10	100	30	uniform	auto	0.806	0.780	0.444
	3	10	300	30	uniform	auto	0.772	0.740	0.424
	4	10	500	40	distance	auto	1.000	0.783	0.454
	5	10	1000	30	uniform	auto	0.715	0.714	0.418
	6	15	100	50	uniform	auto	0.810	0.781	0.441
	7	15	300	50	uniform	auto	0.758	0.740	0.412
	8	15	500	100	uniform	auto	0.744	0.738	0.399
	9	15	1000	200	distance	auto	1.000	0.769	0.441
	10	15	1000	300	uniform	auto	0.719	0.708	0.422

Figure 27

Neural network

Neural networks, also known as artificial neural networks (ANNs) or simulated neural networks (SNNs), are a subset of machine learning and are at the heart of deep learning algorithms. Their name and structure are inspired by the human brain, mimicking the way that biological neurons signal to one another. Artificial neural networks (ANNs) are composed of a node layer, containing an input layer, one or more hidden layers, and an output layer. Each node, or artificial neuron, connects to another and has an associated weight and threshold. If the output of any individual node is above the specified threshold value, that node is activated, sending data to the next layer of the network. Otherwise, no data is passed along to the next layer of the network.

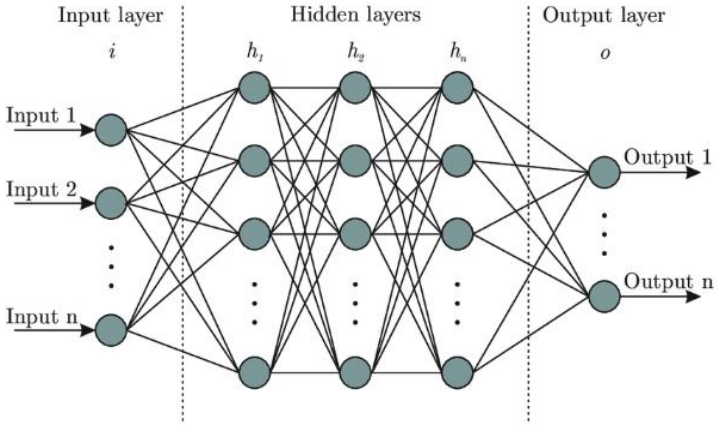


Figure 28

Hyperparameters tuned:

- Hidden_layer_sizes: Refers to the ith element that represents the number of neurons in the hidden layer
- Activation: it is the function through which we pass our weighted sum, in order to have a significant output, namely as a vector of probability or a 0–1 output.
- Solver: Refers to the solver for weight optimization. (default='adam')
- Learning_rate: this hyperparameter refers to the step of backpropagation when parameters are updated according to an optimization function.

Performance:

	Parameters								Avg FDR at 3%		
	iteration	#variables	max_iter	hidden_layer_sizes	activation	solver	learning_rate	learning_rate_init	train	test	oot
Neural Network	1	10	1000	12	tanh	adam	adaptive	0.03	0.789	0.765	0.463
	2	10	1000	8	tanh	adam	adaptive	0.03	0.776	0.764	0.463
	3	10	1000	5	tanh	adam	adaptive	0.03	0.75	0.768	0.523
	4	10	100	5	relu	adam	adaptive	0.03	0.727	0.716	0.44
	5	10	100	5	relu	adam	adaptive	0.01	0.762	0.736	0.534
	6	10	200	8	relu	adam	adaptive	0.001	0.745	0.749	0.518
	7	10	5000	8	relu	adam	adaptive	0.001	0.753	0.732	0.523
	8	10	5000	15	tanh	adam	adaptive	0.001	0.756	0.76	0.525
	9	10	5000	15	tanh	lbfgs	adaptive	0.001	0.841	0.782	0.47
	10	10	5000	12	tanh	lbfgs	adaptive	0.01	0.826	0.789	0.487

Figure 29

Catboost

CatBoost is a kind of Boosting family of algorithms. Similar to XGBoost and LightGBM, it is an improved implementation under the framework of the GBDT algorithm. It is a symmetric decision tree (oblivious trees) algorithm with fewer parameters, support for categorical variables, and a high Accurate GBDT framework. The pain point to solve is to efficiently and reasonably process categorical features. CatBoost is composed of categorical and boost and has high algorithm accuracy and generalization ability.

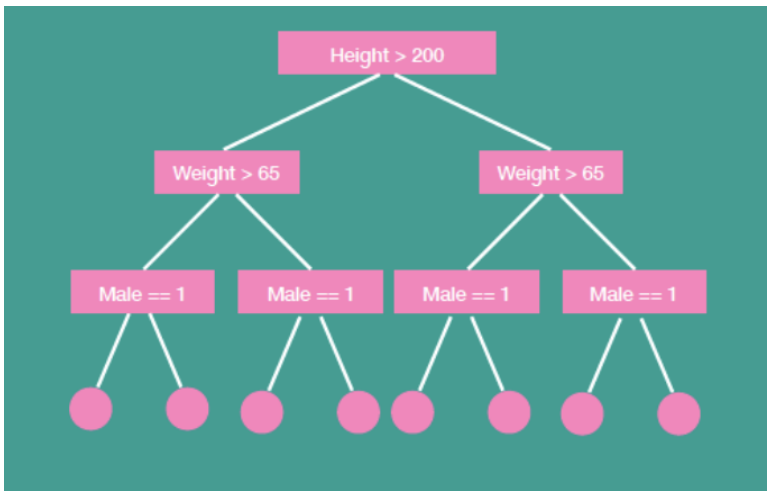


Figure 30

Hyperparameters tuned:

- n_estimators: Maximum number of trees to solve the problem
- learning_rate: The learning rate. Used for reducing the gradient step.(default = 0.03)
- depth: Depth of the tree. Too large may cause an overfitting problem. A range between 6 to 10 usually gets a good result.
- L2_leaf_reg: Coefficient at the L2 regularization term of the cost function.
- eval_metric/custom_metric: The metric used for overfitting detection (if enabled) and best model selection (if enabled).
- loss_function: The metric to use in training. The specified value also determines the machine learning problem to solve.

Performance:

	Parameters								Avg FDR at 3%		
	iteration	#variables	n_estimators	learning_rate	depth	l2_leaf_reg	eval_metric/custom_metric	loss_function	train	test	oot
Catboost	1	10	80	0.05	5	4	AUC	Multiclass	0.745	0.715	0.416
	2	10	150	0.01	5	4	AUC	logloss	0.689	0.677	0.325
	3	10	100	0.01	10	3	AUC	Multiclass	0.736	0.705	0.319
	4	20	80	0.05	7	4	AUC	Multiclass	0.756	0.742	0.454
	5	20	80	0.05	10	4	AUC	Multiclass	0.782	0.752	0.483
	6	20	80	0.05	8	4	Precision	Multiclass	0.764	0.741	0.465
	7	20	60	0.03	7	4	F1	Multiclass	0.726	0.717	0.401
	8	30	80	0.05	7	4	AUC	Multiclass	0.756	0.748	0.441
	9	30	80	0.05	8	4	Recall	Multiclass	0.759	0.742	0.455
	10	30	100	0.07	6	4	Recall	Multiclass	0.799	0.769	0.496

Figure 31

GBC (Gradient-Boosted Decision Trees)

Gradient-boosted decision trees are a machine learning technique for optimizing the predictive value of a model through successive steps in the learning process. Each iteration of the decision tree involves adjusting the values of the coefficients, weights, or biases applied to each of the input variables being used to predict the target value, with the goal of minimizing the loss function (the measure of the difference between the predicted and actual target values). The gradient is the incremental adjustment made in each step of the process; boosting is a method of accelerating the improvement in predictive accuracy to a sufficiently optimum value.

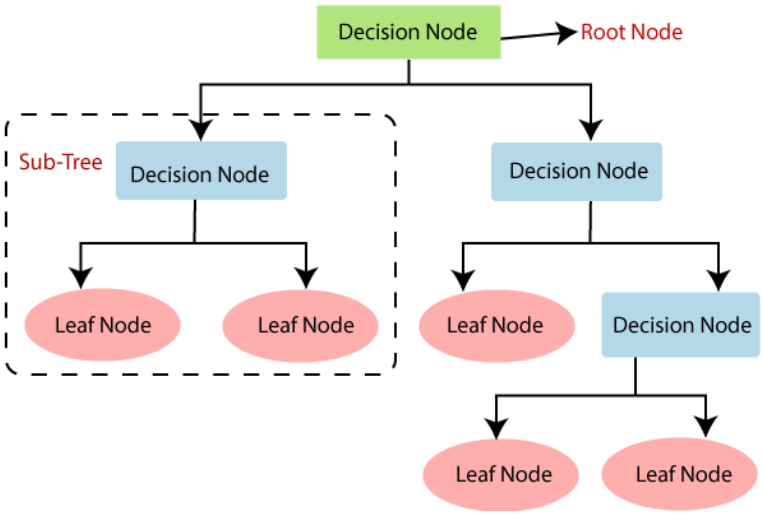


Figure 32

Hyperparameters tuned:

- Learning_rate: This parameter determines the impact of each tree on the final outcome. Lower values are generally preferred as they make the model robust to the specific characteristics of the tree and thus allow it to generalize well.
- Max_depth: The maximum depth of the individual regression estimators. The maximum depth limits the number of nodes in the tree.
- n_estimators: The number of boosting stages to perform. Gradient boosting is fairly robust to overfitting so a large number usually results in better performance.
- Max_leaf_nodes: The maximum number of terminal nodes or leaves in a tree.
- Subsample: The fraction of observations to be selected for each tree. Selection is done by random sampling. Typical values ~0.8 generally work fine but can be fine-tuned further.

Performance:

	Parameters							Avg FDR at 3%		
	iteration	#variables	learning_rate	max_depth	n_estimators	max_leaf_nodes	subsample	train	test	oot
GBC	1	30	0.01	10	100	20	0.8	0.735	0.688	0.392
	2	30	0.01	3	20	20	0.8	0.664	0.653	0.320
	3	30	0.01	3	300	20	0.8	0.734	0.718	0.385
	4	30	0.05	3	100	20	0.8	0.768	0.711	0.470
	5	20	0.01	10	100	20	0.9	0.728	0.707	0.383
	6	20	0.01	5	100	20	0.9	0.731	0.688	0.369
	7	20	0.005	3	100	15	0.9	0.692	0.677	0.331
	8	10	0.01	3	30	40	0.9	0.676	0.661	0.310
	9	10	0.01	3	100	30	0.7	0.693	0.686	0.314
	10	30	0.01	3	100	30	0.7	0.707	0.692	0.343

Figure 33

SVM

SVM is a supervised learning method used for classification. SVM models are effective in high dimensional spaces and it is effective in cases where the number of dimensions is greater than the number of samples. sklearn.svm.SVC() is a C-Support vector classification.

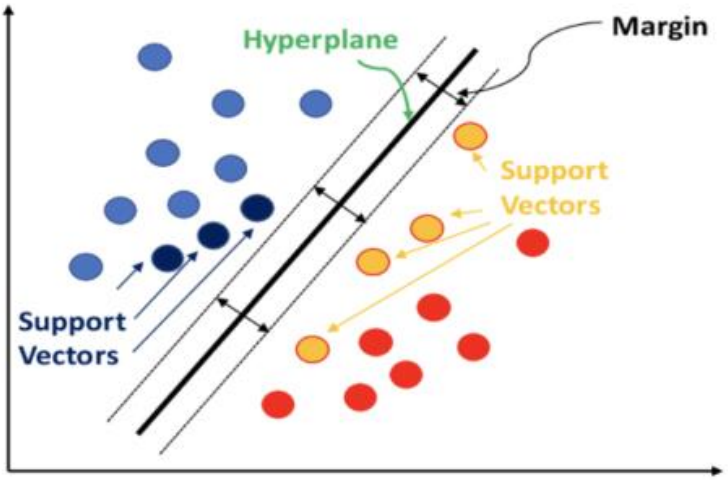


Figure 34

Hyperparameters tuned:

- C: Regularization parameter. The strength of the regularization is inversely proportional to C. Must be strictly positive. The penalty is a squared l2 penalty. (default = 1.0)
- gamma: Kernel coefficient for ‘rbf’, ‘poly’, and ‘sigmoid’. If gamma='scale' (default) is passed then it uses 1 / (n_features * X.var()) as value of gamma, if ‘auto’, uses 1 / n_features. ({‘scale’, ‘auto’} or float, default=’scale’)
- kernel: Specifies the kernel type to be used in the algorithm. If none is given, ‘rbf’ will be used. If a callable is given it is used to pre-compute the kernel matrix from data matrices; that matrix should be an array of shape (n_samples, n_samples). ({‘linear’, ‘poly’, ‘rbf’, ‘sigmoid’, ‘precomputed’} or callable, default =’rbf’)

Performance:

		Parameters						Average FDR at 3%		
	Iteration	#variables	C	n_neighbors	leaf_size	weights		train	test	oot
SVM	1	15	1	auto	poly	TRUE		0.695	0.672	0.341
	2	15	1	auto	linear	TRUE		0.673	0.673	0.277
	3	15	1	auto	rbf	TRUE		0.723	0.677	0.322
	4	15	1	scale	rbf	TRUE		0.787	0.725	0.337
	5	15	0.1	auto	poly	TRUE		0.687	0.681	0.341
	6	10	0.1	auto	poly	TRUE		0.664	0.685	0.451
	7	10	1	scale	poly	TRUE		0.699	0.659	0.458
	8	10	1	scale	rbf	TRUE		0.762	0.700	0.310
	9	10	1	scale	linear	TRUE		0.676	0.661	0.360
	10	30	1	scale	rbf	TRUE		0.786	0.743	0.381

Figure 35

Result

The final model we choose is a LightGBM model with the following parameters: LGBMClassifier(n_estimators = 100, max_depth = 6, num_leaves = 50, boosting_type = 'gbdt', learning_rate = 0.01)

We chose to use the top 15 variables that we got from the feature selection process to train the model. It performs the best compared to other parameters and other models. Since the model also only keeps the most informative variables, we avoided the curse of dimensionality.

The train test split we used is 0.7/0.3 and the validation dataset we chose is the records from the last two months. The final performance summary of our model on the training, testing, and validation datasets are as shown below:

Training	# Records	# Goods	# Bads	Fradu Rate								
	59,010	58,370	640	1.0846%								
	Bin Statistics					Cumulative Statistics						
Population Bin %	# Records	# Goods	# Bads	% Goods	% Bads	Total # Reocrds	Cumulative Goods	Cumulative Bads	% Cumulative Goods	% Bad(FDR)	KS	FPR
1	590	116	474	19.66	80.34	590	116	474	0.20	74.06	73.86	0.24
2	590	507	83	85.93	14.07	1,180	623	557	1.07	87.03	85.96	1.12
3	590	564	26	95.59	4.41	1,770	1,187	583	2.03	91.09	89.06	2.04
4	590	580	10	98.31	1.69	2,360	1,767	593	3.03	92.66	89.63	2.98
5	590	582	8	98.64	1.36	2,950	2,349	601	4.02	93.91	89.88	3.91
6	591	588	3	99.49	0.51	3,541	2,937	604	5.03	94.38	89.34	4.86
7	590	587	3	99.49	0.51	4,131	3,524	607	6.04	94.84	88.81	5.81
8	590	587	3	99.49	0.51	4,721	4,111	610	7.04	95.31	88.27	6.74
9	590	590	0	100.00	0.00	5,311	4,701	610	8.05	95.31	87.26	7.71
10	590	588	2	99.66	0.34	5,901	5,289	612	9.06	95.63	86.56	8.64
11	590	589	1	99.83	0.17	6,491	5,878	613	10.07	95.78	85.71	9.59
12	590	589	1	99.83	0.17	7,081	6,467	614	11.08	95.94	84.86	10.53
13	590	587	3	99.49	0.51	7,671	7,054	617	12.08	96.41	84.32	11.43
14	590	587	3	99.49	0.51	8,261	7,641	620	13.09	96.88	83.78	12.32
15	591	591	0	100.00	0.00	8,852	8,232	620	14.10	96.88	82.77	13.28
16	590	589	1	99.83	0.17	9,442	8,821	621	15.11	97.03	81.92	14.20
17	590	589	1	99.83	0.17	10,032	9,410	622	16.12	97.19	81.07	15.13
18	590	590	0	100.00	0.00	10,622	10,000	622	17.13	97.19	80.06	16.08
19	590	589	1	99.83	0.17	11,212	10,589	623	18.14	97.34	79.20	17.00
20	590	588	2	99.66	0.34	11,802	11,177	625	19.15	97.66	78.51	17.88

Figure 36

Testing	# Records	# Goods	# Bads	Fradu Rate								
	25,290	25,050	240	0.94899%								
	Bin Statistics					Cumulative Statistics						
Population Bin %	# Records	# Goods	# Bads	% Goods	% Bads	Total # Reocrds	Cumulative Goods	Cumulative Bads	% Cumulative Goods	% Bad(FDR)	KS	FPR
1	253	100	153	39.53	60.47	253	100	153	0.40	63.75	63.35	0.65
2	253	228	25	90.12	9.88	506	328	178	1.31	74.17	72.86	1.84
3	253	241	12	95.26	4.74	759	569	190	2.27	79.17	76.90	2.99
4	253	242	11	95.65	4.35	1,012	811	201	3.24	83.75	80.51	4.03
5	252	251	1	99.60	0.40	1,264	1,062	202	4.24	84.17	79.93	5.26
6	253	249	4	98.42	1.58	1,517	1,311	206	5.23	85.83	80.60	6.36
7	253	250	3	98.81	1.19	1,770	1,561	209	6.23	87.08	80.85	7.47
8	253	251	2	99.21	0.79	2,023	1,812	211	7.23	87.92	80.68	8.59
9	253	252	1	99.60	0.40	2,276	2,064	212	8.24	88.33	80.09	9.74
10	253	251	2	99.21	0.79	2,529	2,315	214	9.24	89.17	79.93	10.82
11	253	251	2	99.21	0.79	2,782	2,566	216	10.24	90.00	79.76	11.88
12	253	253	0	100.00	0.00	3,035	2,819	216	11.25	90.00	78.75	13.05
13	253	251	2	99.21	0.79	3,288	3,070	218	12.26	90.83	78.58	14.08
14	253	252	1	99.60	0.40	3,541	3,322	219	13.26	91.25	77.99	15.17
15	253	253	0	100.00	0.00	3,794	3,575	219	14.27	91.25	76.98	16.32
16	252	252	0	100.00	0.00	4,046	3,827	219	15.28	91.25	75.97	17.47
17	253	253	0	100.00	0.00	4,299	4,080	219	16.29	91.25	74.96	18.63
18	253	253	0	100.00	0.00	4,552	4,333	219	17.30	91.25	73.95	19.79
19	253	253	0	100.00	0.00	4,805	4,586	219	18.31	91.25	72.94	20.94
20	253	253	0	100.00	0.00	5,058	4,839	219	19.32	91.25	71.93	22.10

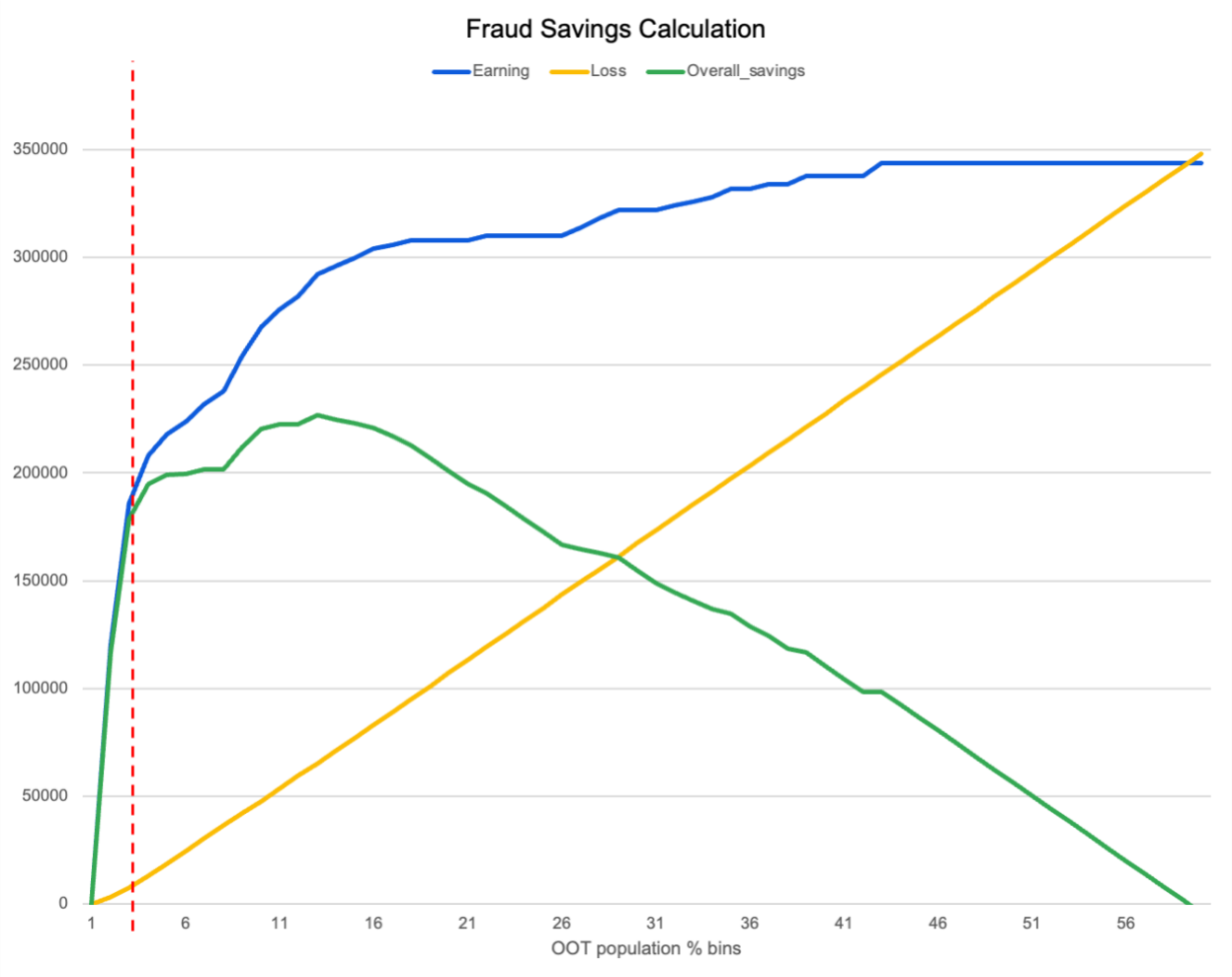
Figure 37

Validation	# Records	# Goods	# Bads	Fradu Rate								
	12,097	11,918	179	1.4797%								
	Bin Statistics					Cumulative Statistics						
Population Bin %	# Records	# Goods	# Bads	% Goods	% Bads	Total # Reocrds	Cumulative Goods	Cumulative Bads	% Cumulative Goods	% Bad(FDR)	KS	FPR
1	121	61	60	50.41	49.59	121	61	60	0.51	33.52	33.01	1.02
2	121	88	33	72.73	27.27	242	149	93	1.25	51.96	50.71	1.60
3	121	110	11	90.91	9.09	363	259	104	2.17	58.10	55.93	2.49
4	121	116	5	95.87	4.13	484	375	109	3.15	60.89	57.75	3.44
5	121	118	3	97.52	2.48	605	493	112	4.14	62.57	58.43	4.40
6	121	117	4	96.69	3.31	726	610	116	5.12	64.80	59.69	5.26
7	121	118	3	97.52	2.48	847	728	119	6.11	66.48	60.37	6.12
8	121	113	8	93.39	6.61	968	841	127	7.06	70.95	63.89	6.62
9	121	114	7	94.21	5.79	1,089	955	134	8.01	74.86	66.85	7.13
10	121	117	4	96.69	3.31	1,210	1,072	138	8.99	77.09	68.10	7.77
11	121	118	3	97.52	2.48	1,331	1,190	141	9.98	78.77	68.79	8.44
12	121	116	5	95.87	4.13	1,452	1,306	146	10.96	81.56	70.61	8.95
13	121	119	2	98.35	1.65	1,573	1,425	148	11.96	82.68	70.72	9.63
14	121	119	2	98.35	1.65	1,694	1,544	150	12.96	83.80	70.84	10.29
15	121	119	2	98.35	1.65	1,815	1,663	152	13.95	84.92	70.96	10.94
16	121	120	1	99.17	0.83	1,936	1,783	153	14.96	85.47	70.51	11.65
17	120	119	1	99.17	0.83	2,056	1,902	154	15.96	86.03	70.07	12.35
18	121	121	0	100.00	0.00	2,177	2,023	154	16.97	86.03	69.06	13.14
19	121	121	0	100.00	0.00	2,298	2,144	154	17.99	86.03	68.04	13.92
20	121	121	0	100.00	0.00	2,419	2,265	154	19.00	86.03	67.03	14.71

Figure 38

Conclusions

In conclusion, using our best performance LightBGM model, we are able to achieve a Fraud Detection Rate of 90.4% for the training set, 81.5% for the testing set, and 55.5% for the out-of-time validation set at 3% of the population. As shown in the figure below, we choose to capture fraudulent transactions at 3% because the overall saving is relatively high, and if choosing a higher cutoff rate than 3%, the overall incremental savings are low. Using the model to capture potential fraudulent transactions will help the company save $ 1,170,300 annually under the assumption that $2,000 gain for every fraud that's caught and $50 loss for every false positive.



Appendix 1: Data Quality Report

Table of Contents:

Description：

The data is actual credit card purchases from a US government organization. The time frame of this dataset is from 2006-01-01 to 2006-12-31, 12 months in total. The dataset includes card number, date, merchandise description, merchandise state, merchandise state, merchandise zip, transaction type, amount, fraud identifier, etc.

- Number of fields: There are 10 columns/fields.
- Number of records: There are 96,753 records.

Summary Tables:

Table for Numeric or Date Time：

Field Name	% Populated	Min	Max	Mean	Std dev	% Zero
Date	100%	2006-01-01	2006-12-31	NA	NA	0%
Amount	100%	0.01	3102045.53	427.89	10006.14	0%

Table for Categorical Fields：

Field Name	% Populated	# Unique Values	Most Common Value
Recnum	100.00%	96,753	NA
Cardnum	100.00%	1,645	5142148452
Merchnum	96.51%	13,092	930090121224
Merch description	100.00%	13,126	GSA-FSS-ADV
Merch state	98.76%	228	TN
Merch zip	95.19%	4,568	38118.0
Transtype	100.00%	4	P
Fraud	100.00%	2	0

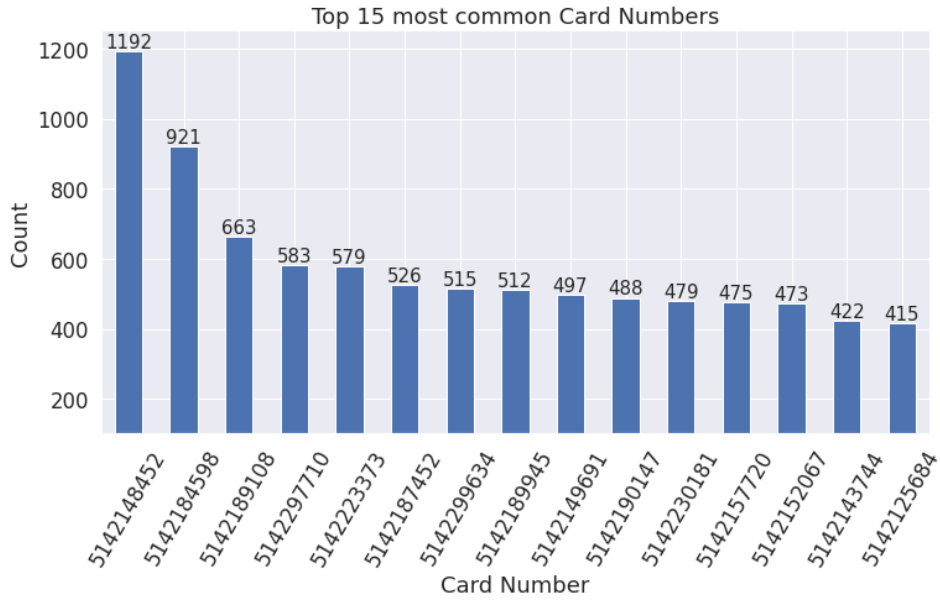
Distribution of Fields:

1. Recnum

- The field is a unique identifier for each record, essentially is an index number by time order
- Each row here has a unique non-null value, starting from 1 and ending with 96753.

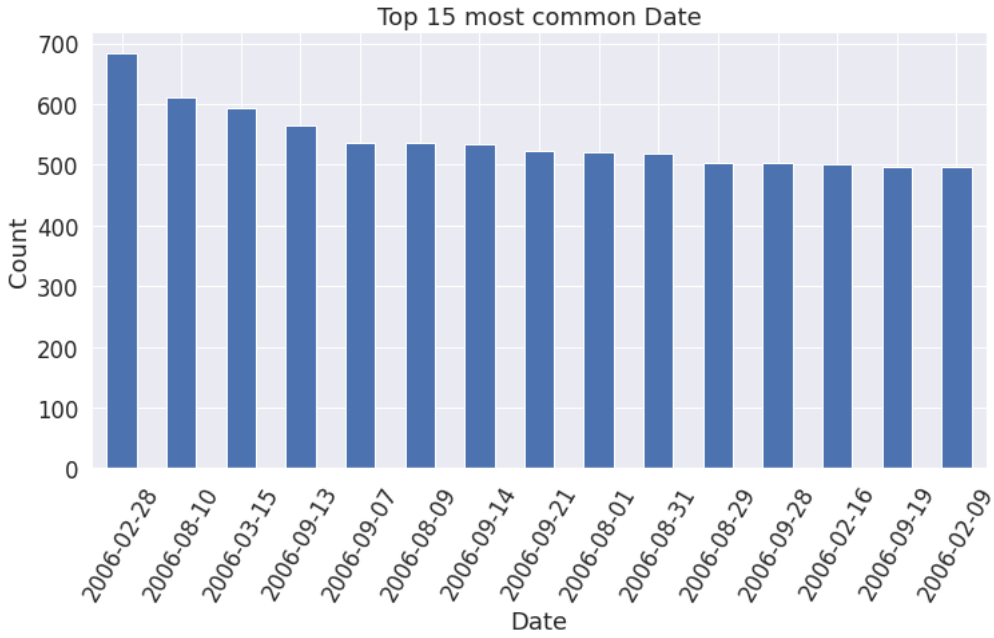
2. Cardnum

- The field is the card number of each transaction.
- The entries have 10 digits. There are 1,645 unique card numbers in this field. The most common value is 5142148452.



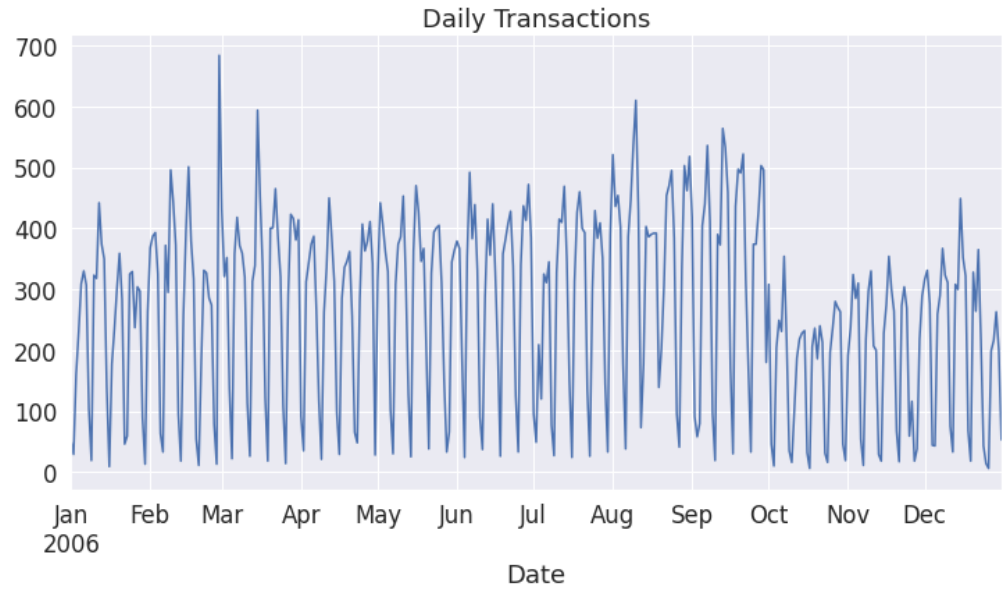
3. Date

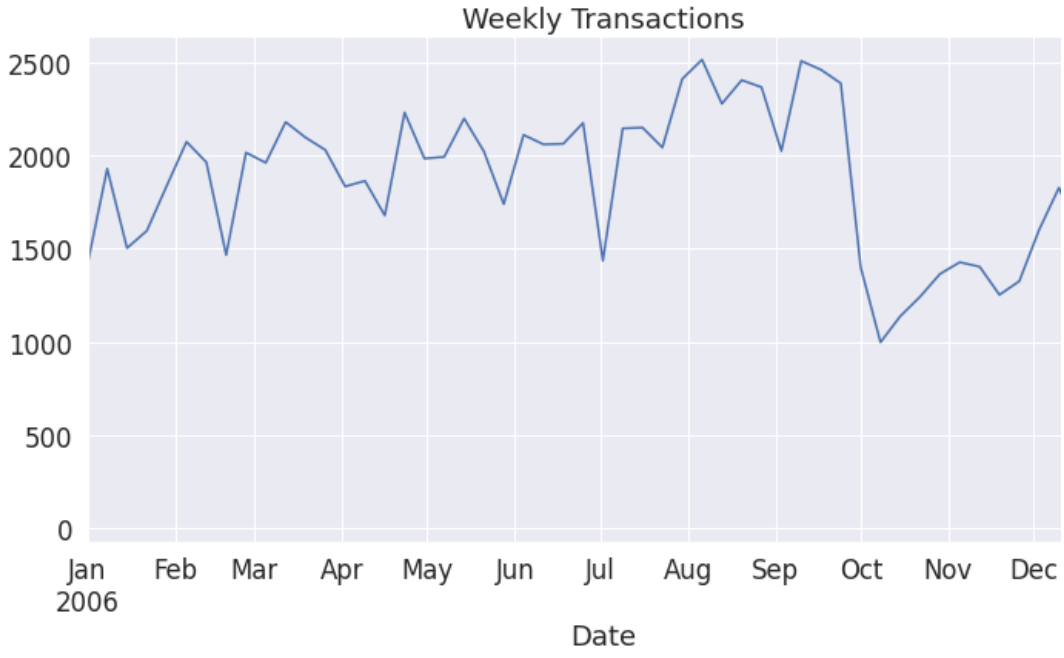
- The field is the date of each transaction. The time frame is from 2006-01-01 to 2006-12-31
- There are 365 unique entries. The most common date in this dataset is 2006-02-28

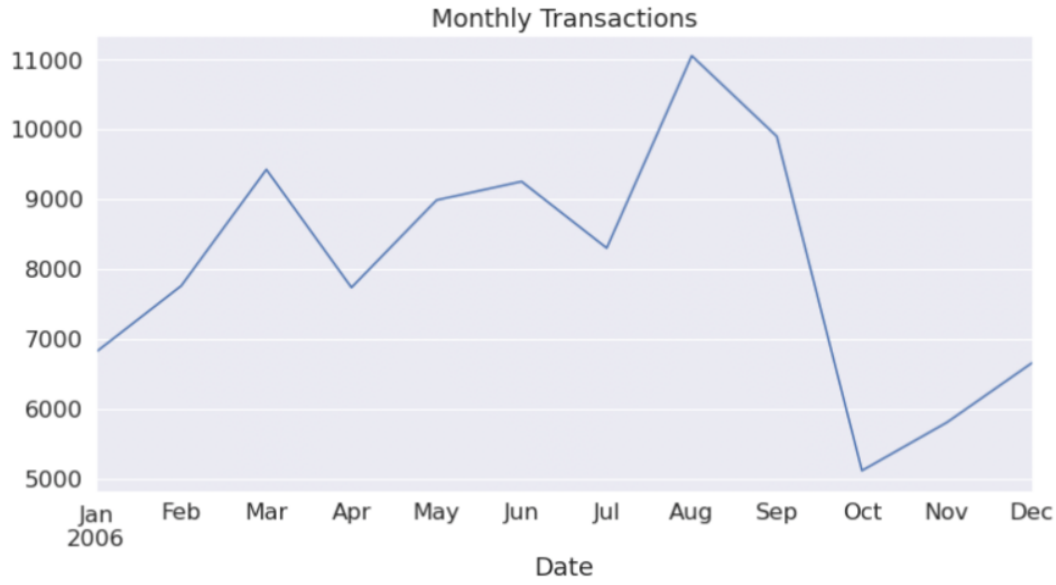


- The dataset has spikes on weekends because of the nature of this dataset that it's the transaction data of an US government organization.

- The monthly volume drops significantly in September because, for the Federal government, the fiscal year runs from October 1 to September 30.

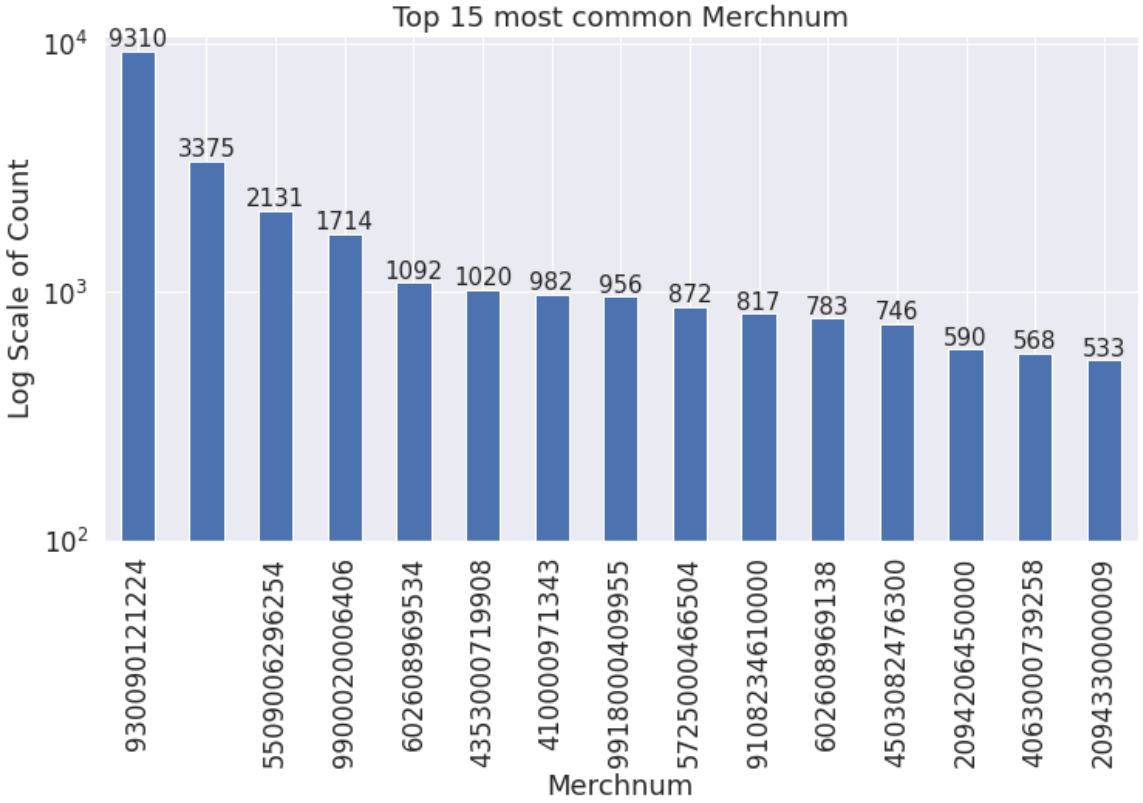






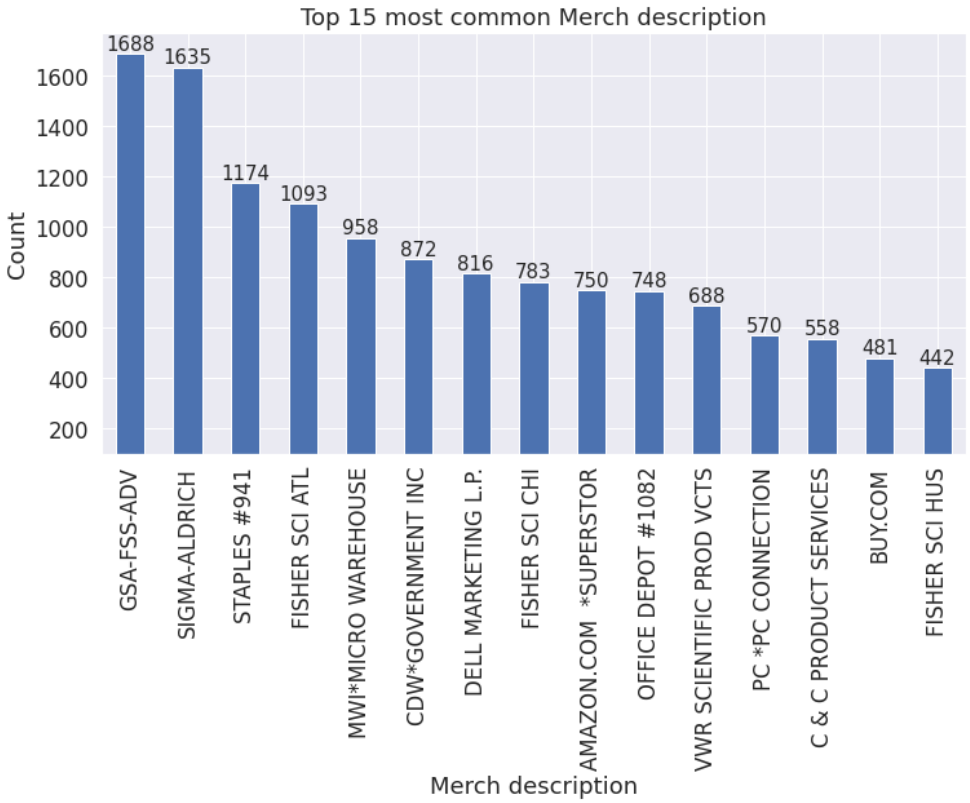
4. Merchnum

- The field records the merchandise number of each transaction.
- There are 13,092 unique entries. The most comment value is 930090121224 which appears 9,310 times in the dataset. The second most common Merchnum is blank which represents missing value so that the %populated of this field is 96.51%.



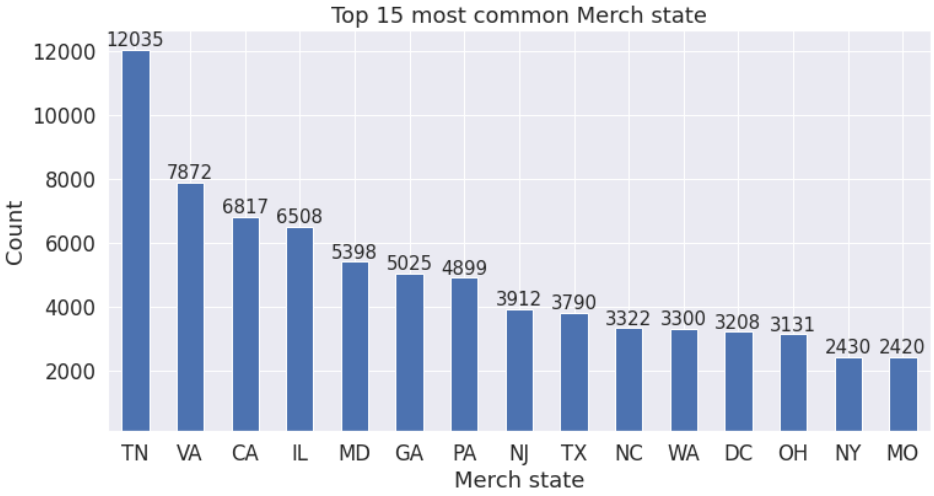
5. Merch description

- The field contains descriptions of each transaction.
- There are 13,126 unique entries in this field. The most comment value is GSA-FSS-ADV which appears 1,688 times in the dataset.



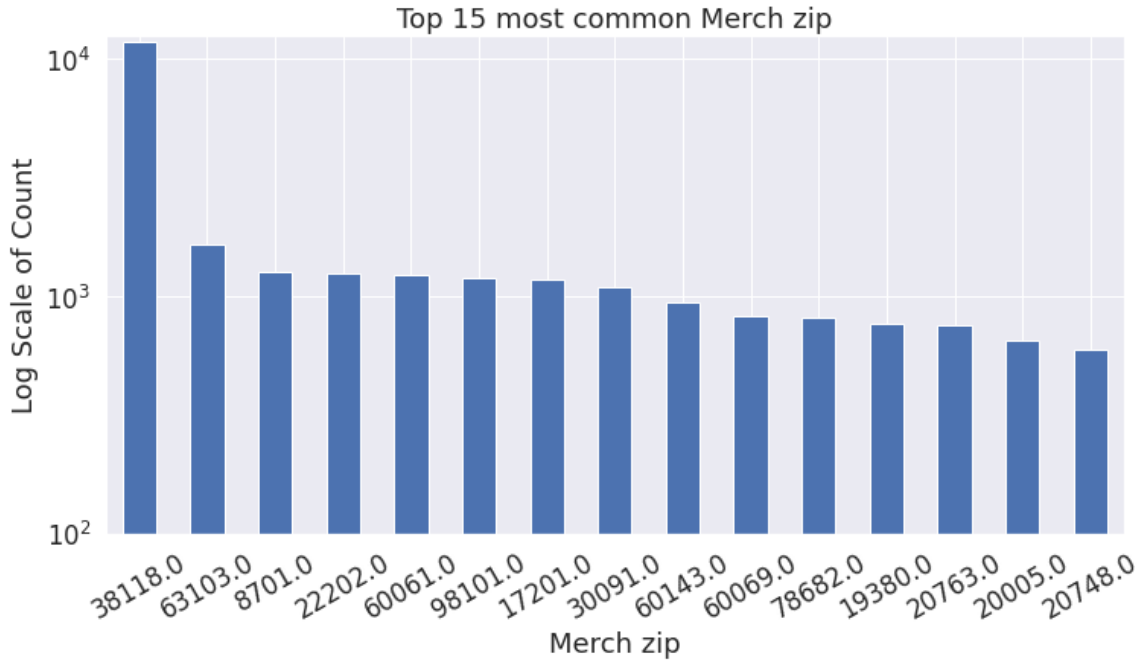
6. Merch state

- The field records the state information of each transaction, numbers are representing international locations or bad data.
- There are 228 unique values in this field. The most comment value is TN which appears 12,035 times in the dataset.



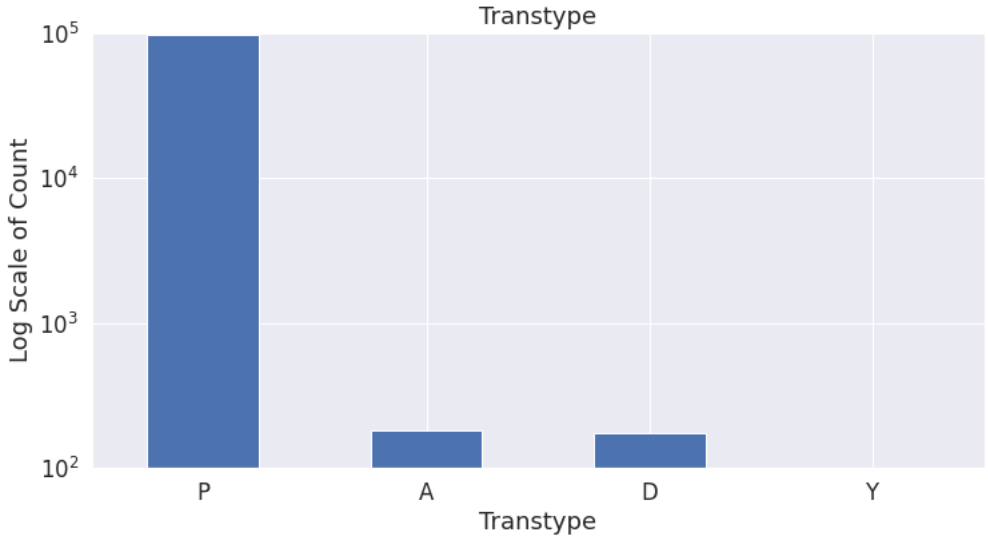
7. Merch zip

- The field records zip code of each transaction
- There are 4,568 unique values in this field. The most common value is 38118 which appears 11,868 times in the dataset.



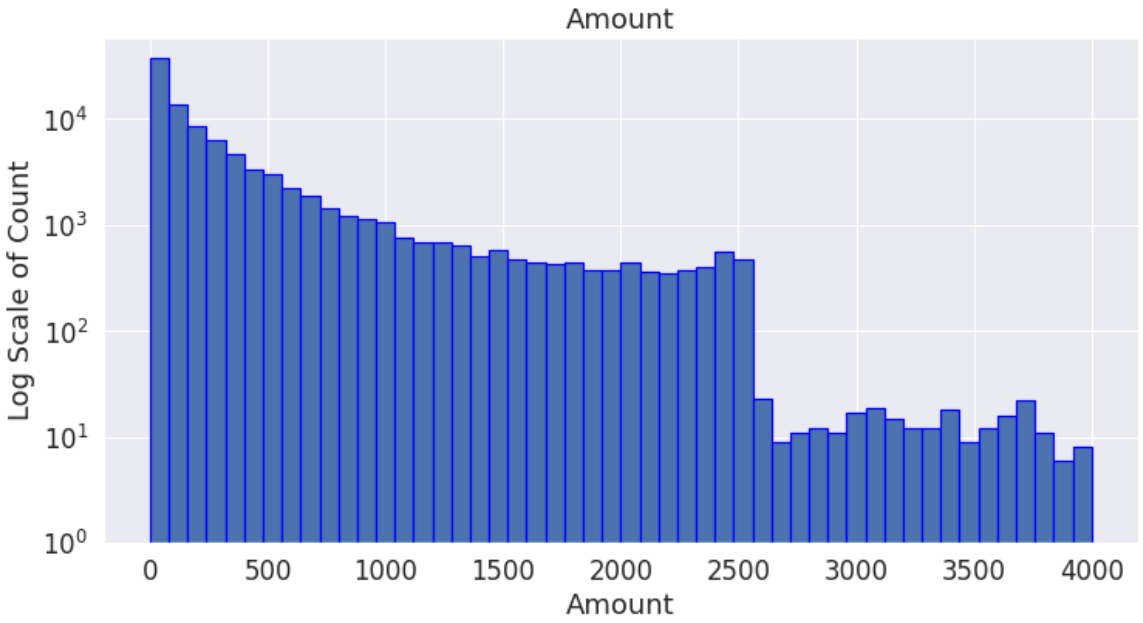
8. Transtype

- The field records the transaction type of each transaction.
- There are 4 unique values in this field: P, A, D and Y. P stands for purchase. The most common value is P which appears 96,398 times in the dataset.



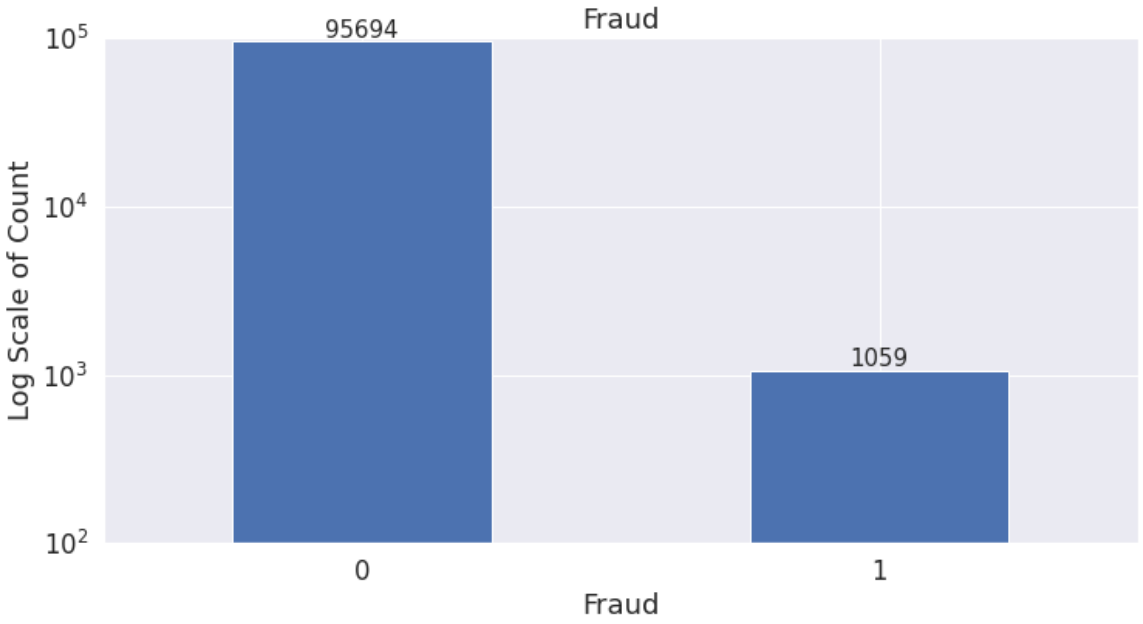
9. Amount

- The field records the amount of each transaction.



10. Fraud

- The field is a label indicating fraud.
- There are 2 unique values in this field. Only 1,059 frauds (1) in this field.



Appendix 2: List of all 804 variables

(30 variables in the final list were highlighted)

Dow_Risk	merch_zip_actual/med_0	Merchnum_desc_avg_3
state_risk	merch_zip_actual/toal_0	Merchnum_desc_max_3
benford_Cardnum	merch_zip_count_1	Merchnum_desc_med_3
benford_Merchnum	merch_zip_avg_1	Merchnum_desc_total_3
Cardnum_day_since	merch_zip_max_1	Merchnum_desc_actual/avg_3
Cardnum_count_0	merch_zip_med_1	Merchnum_desc_actual/max_3
Cardnum_avg_0	merch_zip_total_1	Merchnum_desc_actual/med_3
Cardnum_max_0	merch_zip_actual/avg_1	Merchnum_desc_actual/toal_3
Cardnum_med_0	merch_zip_actual/max_1	Merchnum_desc_count_7
Cardnum_total_0	merch_zip_actual/med_1	Merchnum_desc_avg_7
Cardnum_actual/avg_0	merch_zip_actual/toal_1	Merchnum_desc_max_7
Cardnum_actual/max_0	merch_zip_count_3	Merchnum_desc_med_7
Cardnum_actual/med_0	merch_zip_avg_3	Merchnum_desc_total_7
Cardnum_actual/toal_0	merch_zip_max_3	Merchnum_desc_actual/avg_7
Cardnum_count_1	merch_zip_med_3	Merchnum_desc_actual/max_7
Cardnum_avg_1	merch_zip_total_3	Merchnum_desc_actual/med_7
Cardnum_max_1	merch_zip_actual/avg_3	Merchnum_desc_actual/toal_7
Cardnum_med_1	merch_zip_actual/max_3	Merchnum_desc_count_14
Cardnum_total_1	merch_zip_actual/med_3	Merchnum_desc_avg_14
Cardnum_actual/avg_1	merch_zip_actual/toal_3	Merchnum_desc_max_14
Cardnum_actual/max_1	merch_zip_count_7	Merchnum_desc_med_14

Cardnum_actual/med_1	merch_zip_avg_7	Merchnum_desc_total_14
Cardnum_actual/toal_1	merch_zip_max_7	Merchnum_desc_actual/avg_14
Cardnum_count_3	merch_zip_med_7	Merchnum_desc_actual/max_14
Cardnum_avg_3	merch_zip_total_7	Merchnum_desc_actual/med_14
Cardnum_max_3	merch_zip_actual/avg_7	Merchnum_desc_actual/toal_14
Cardnum_med_3	merch_zip_actual/max_7	Merchnum_desc_count_30
Cardnum_total_3	merch_zip_actual/med_7	Merchnum_desc_avg_30
Cardnum_actual/avg_3	merch_zip_actual/toal_7	Merchnum_desc_max_30
Cardnum_actual/max_3	merch_zip_count_14	Merchnum_desc_med_30
Cardnum_actual/med_3	merch_zip_avg_14	Merchnum_desc_total_30
Cardnum_actual/toal_3	merch_zip_max_14	Merchnum_desc_actual/avg_30
Cardnum_count_7	merch_zip_med_14	Merchnum_desc_actual/max_30
Cardnum_avg_7	merch_zip_total_14	Merchnum_desc_actual/med_30
Cardnum_max_7	merch_zip_actual/avg_14	Merchnum_desc_actual/toal_30
Cardnum_med_7	merch_zip_actual/max_14	Merchnum_desc_count_60
Cardnum_total_7	merch_zip_actual/med_14	Merchnum_desc_avg_60
Cardnum_actual/avg_7	merch_zip_actual/toal_14	Merchnum_desc_max_60
Cardnum_actual/max_7	merch_zip_count_30	Merchnum_desc_med_60
Cardnum_actual/med_7	merch_zip_avg_30	Merchnum_desc_total_60
Cardnum_actual/toal_7	merch_zip_max_30	Merchnum_desc_actual/avg_60
Cardnum_count_14	merch_zip_med_30	Merchnum_desc_actual/max_60
Cardnum_avg_14	merch_zip_total_30	Merchnum_desc_actual/med_60
Cardnum_max_14	merch_zip_actual/avg_30	Merchnum_desc_actual/toal_60
Cardnum_med_14	merch_zip_actual/max_30	Card_Merchnum_desc_day_since

Cardnum_total_14	merch_zip_actual/med_30	Card_Merchnum_desc_count_0
Cardnum_actual/avg_14	merch_zip_actual/toal_30	Card_Merchnum_desc_avg_0
Cardnum_actual/max_14	merch_zip_count_60	Card_Merchnum_desc_max_0
Cardnum_actual/med_14	merch_zip_avg_60	Card_Merchnum_desc_med_0
Cardnum_actual/toal_14	merch_zip_max_60	Card_Merchnum_desc_total_0
Cardnum_count_30	merch_zip_med_60	Card_Merchnum_desc_actual/avg_0
Cardnum_avg_30	merch_zip_total_60	Card_Merchnum_desc_actual/max_0
Cardnum_max_30	merch_zip_actual/avg_60	Card_Merchnum_desc_actual/med_0
Cardnum_med_30	merch_zip_actual/max_60	Card_Merchnum_desc_actual/toal_0
Cardnum_total_30	merch_zip_actual/med_60	Card_Merchnum_desc_count_1
Cardnum_actual/avg_30	merch_zip_actual/toal_60	Card_Merchnum_desc_avg_1
Cardnum_actual/max_30	zip3_day_since	Card_Merchnum_desc_max_1
Cardnum_actual/med_30	zip3_count_0	Card_Merchnum_desc_med_1
Cardnum_actual/toal_30	zip3_avg_0	Card_Merchnum_desc_total_1
Cardnum_count_60	zip3_max_0	Card_Merchnum_desc_actual/avg_1
Cardnum_avg_60	zip3_med_0	Card_Merchnum_desc_actual/max_1
Cardnum_max_60	zip3_total_0	Card_Merchnum_desc_actual/med_1
Cardnum_med_60	zip3_actual/avg_0	Card_Merchnum_desc_actual/toal_1
Cardnum_total_60	zip3_actual/max_0	Card_Merchnum_desc_count_3
Cardnum_actual/avg_60	zip3_actual/med_0	Card_Merchnum_desc_avg_3
Cardnum_actual/max_60	zip3_actual/toal_0	Card_Merchnum_desc_max_3
Cardnum_actual/med_60	zip3_count_1	Card_Merchnum_desc_med_3
Cardnum_actual/toal_60	zip3_avg_1	Card_Merchnum_desc_total_3
Merchnum_day_since	zip3_max_1	Card_Merchnum_desc_actual/avg_3

Merchnum_count_0	zip3_med_1	Card_Merchnum_desc_actual/max_3
Merchnum_avg_0	zip3_total_1	Card_Merchnum_desc_actual/med_3
Merchnum_max_0	zip3_actual/avg_1	Card_Merchnum_desc_actual/toal_3
Merchnum_med_0	zip3_actual/max_1	Card_Merchnum_desc_count_7
Merchnum_total_0	zip3_actual/med_1	Card_Merchnum_desc_avg_7
Merchnum_actual/avg_0	zip3_actual/toal_1	Card_Merchnum_desc_max_7
Merchnum_actual/max_0	zip3_count_3	Card_Merchnum_desc_med_7
Merchnum_actual/med_0	zip3_avg_3	Card_Merchnum_desc_total_7
Merchnum_actual/toal_0	zip3_max_3	Card_Merchnum_desc_actual/avg_7
Merchnum_count_1	zip3_med_3	Card_Merchnum_desc_actual/max_7
Merchnum_avg_1	zip3_total_3	Card_Merchnum_desc_actual/med_7
Merchnum_max_1	zip3_actual/avg_3	Card_Merchnum_desc_actual/toal_7
Merchnum_med_1	zip3_actual/max_3	Card_Merchnum_desc_count_14
Merchnum_total_1	zip3_actual/med_3	Card_Merchnum_desc_avg_14
Merchnum_actual/avg_1	zip3_actual/toal_3	Card_Merchnum_desc_max_14
Merchnum_actual/max_1	zip3_count_7	Card_Merchnum_desc_med_14
Merchnum_actual/med_1	zip3_avg_7	Card_Merchnum_desc_total_14
Merchnum_actual/toal_1	zip3_max_7	Card_Merchnum_desc_actual/avg_14
Merchnum_count_3	zip3_med_7	Card_Merchnum_desc_actual/max_14
Merchnum_avg_3	zip3_total_7	Card_Merchnum_desc_actual/med_14
Merchnum_max_3	zip3_actual/avg_7	Card_Merchnum_desc_actual/toal_14
Merchnum_med_3	zip3_actual/max_7	Card_Merchnum_desc_count_30
Merchnum_total_3	zip3_actual/med_7	Card_Merchnum_desc_avg_30
Merchnum_actual/avg_3	zip3_actual/toal_7	Card_Merchnum_desc_max_30

Merchnum_actual/max_3	zip3_count_14	Card_Merchnum_desc_med_30
Merchnum_actual/med_3	zip3_avg_14	Card_Merchnum_desc_total_30
Merchnum_actual/toal_3	zip3_max_14	Card_Merchnum_desc_actual/avg_30
Merchnum_count_7	zip3_med_14	Card_Merchnum_desc_actual/max_30
Merchnum_avg_7	zip3_total_14	Card_Merchnum_desc_actual/med_30
Merchnum_max_7	zip3_actual/avg_14	Card_Merchnum_desc_actual/toal_30
Merchnum_med_7	zip3_actual/max_14	Card_Merchnum_desc_count_60
Merchnum_total_7	zip3_actual/med_14	Card_Merchnum_desc_avg_60
Merchnum_actual/avg_7	zip3_actual/toal_14	Card_Merchnum_desc_max_60
Merchnum_actual/max_7	zip3_count_30	Card_Merchnum_desc_med_60
Merchnum_actual/med_7	zip3_avg_30	Card_Merchnum_desc_total_60
Merchnum_actual/toal_7	zip3_max_30	Card_Merchnum_desc_actual/avg_60
Merchnum_count_14	zip3_med_30	Card_Merchnum_desc_actual/max_60
Merchnum_avg_14	zip3_total_30	Card_Merchnum_desc_actual/med_60
Merchnum_max_14	zip3_actual/avg_30	Card_Merchnum_desc_actual/toal_60
Merchnum_med_14	zip3_actual/max_30	Cardnum_count_0_by_7
Merchnum_total_14	zip3_actual/med_30	Cardnum_total_amount_0_by_7
Merchnum_actual/avg_14	zip3_actual/toal_30	Cardnum_count_0_by_14
Merchnum_actual/max_14	zip3_count_60	Cardnum_total_amount_0_by_14
Merchnum_actual/med_14	zip3_avg_60	Cardnum_count_0_by_30
Merchnum_actual/toal_14	zip3_max_60	Cardnum_total_amount_0_by_30
Merchnum_count_30	zip3_med_60	Cardnum_count_0_by_60
Merchnum_avg_30	zip3_total_60	Cardnum_total_amount_0_by_60
Merchnum_max_30	zip3_actual/avg_60	Cardnum_count_1_by_7

Merchnum_med_30	zip3_actual/max_60	Cardnum_total_amount_1_by_7
Merchnum_total_30	zip3_actual/med_60	Cardnum_count_1_by_14
Merchnum_actual/avg_30	zip3_actual/toal_60	Cardnum_total_amount_1_by_14
Merchnum_actual/max_30	card_zip3_day_since	Cardnum_count_1_by_30
Merchnum_actual/med_30	card_zip3_count_0	Cardnum_total_amount_1_by_30
Merchnum_actual/toal_30	card_zip3_avg_0	Cardnum_count_1_by_60
Merchnum_count_60	card_zip3_max_0	Cardnum_total_amount_1_by_60
Merchnum_avg_60	card_zip3_med_0	Merchnum_count_0_by_7
Merchnum_max_60	card_zip3_total_0	Merchnum_total_amount_0_by_7
Merchnum_med_60	card_zip3_actual/avg_0	Merchnum_count_0_by_14
Merchnum_total_60	card_zip3_actual/max_0	Merchnum_total_amount_0_by_14
Merchnum_actual/avg_60	card_zip3_actual/med_0	Merchnum_count_0_by_30
Merchnum_actual/max_60	card_zip3_actual/toal_0	Merchnum_total_amount_0_by_30
Merchnum_actual/med_60	card_zip3_count_1	Merchnum_count_0_by_60
Merchnum_actual/toal_60	card_zip3_avg_1	Merchnum_total_amount_0_by_60
card_merch_day_since	card_zip3_max_1	Merchnum_count_1_by_7
card_merch_count_0	card_zip3_med_1	Merchnum_total_amount_1_by_7
card_merch_avg_0	card_zip3_total_1	Merchnum_count_1_by_14
card_merch_max_0	card_zip3_actual/avg_1	Merchnum_total_amount_1_by_14
card_merch_med_0	card_zip3_actual/max_1	Merchnum_count_1_by_30
card_merch_total_0	card_zip3_actual/med_1	Merchnum_total_amount_1_by_30
card_merch_actual/avg_0	card_zip3_actual/toal_1	Merchnum_count_1_by_60
card_merch_actual/max_0	card_zip3_count_3	Merchnum_total_amount_1_by_60
card_merch_actual/med_0	card_zip3_avg_3	card_merch_count_0_by_7

card_merch_actual/toal_0	card_zip3_max_3	card_merch_total_amount_0_by_7
card_merch_count_1	card_zip3_med_3	card_merch_count_0_by_14
card_merch_avg_1	card_zip3_total_3	card_merch_total_amount_0_by_14
card_merch_max_1	card_zip3_actual/avg_3	card_merch_count_0_by_30
card_merch_med_1	card_zip3_actual/max_3	card_merch_total_amount_0_by_30
card_merch_total_1	card_zip3_actual/med_3	card_merch_count_0_by_60
card_merch_actual/avg_1	card_zip3_actual/toal_3	card_merch_total_amount_0_by_60
card_merch_actual/max_1	card_zip3_count_7	card_merch_count_1_by_7
card_merch_actual/med_1	card_zip3_avg_7	card_merch_total_amount_1_by_7
card_merch_actual/toal_1	card_zip3_max_7	card_merch_count_1_by_14
card_merch_count_3	card_zip3_med_7	card_merch_total_amount_1_by_14
card_merch_avg_3	card_zip3_total_7	card_merch_count_1_by_30
card_merch_max_3	card_zip3_actual/avg_7	card_merch_total_amount_1_by_30
card_merch_med_3	card_zip3_actual/max_7	card_merch_count_1_by_60
card_merch_total_3	card_zip3_actual/med_7	card_merch_total_amount_1_by_60
card_merch_actual/avg_3	card_zip3_actual/toal_7	card_zip_count_0_by_7
card_merch_actual/max_3	card_zip3_count_14	card_zip_total_amount_0_by_7
card_merch_actual/med_3	card_zip3_avg_14	card_zip_count_0_by_14
card_merch_actual/toal_3	card_zip3_max_14	card_zip_total_amount_0_by_14
card_merch_count_7	card_zip3_med_14	card_zip_count_0_by_30
card_merch_avg_7	card_zip3_total_14	card_zip_total_amount_0_by_30
card_merch_max_7	card_zip3_actual/avg_14	card_zip_count_0_by_60
card_merch_med_7	card_zip3_actual/max_14	card_zip_total_amount_0_by_60
card_merch_total_7	card_zip3_actual/med_14	card_zip_count_1_by_7

card_merch_actual/avg_7	card_zip3_actual/toal_14	card_zip_total_amount_1_by_7
card_merch_actual/max_7	card_zip3_count_30	card_zip_count_1_by_14
card_merch_actual/med_7	card_zip3_avg_30	card_zip_total_amount_1_by_14
card_merch_actual/toal_7	card_zip3_max_30	card_zip_count_1_by_30
card_merch_count_14	card_zip3_med_30	card_zip_total_amount_1_by_30
card_merch_avg_14	card_zip3_total_30	card_zip_count_1_by_60
card_merch_max_14	card_zip3_actual/avg_30	card_zip_total_amount_1_by_60
card_merch_med_14	card_zip3_actual/max_30	merch_zip_count_0_by_7
card_merch_total_14	card_zip3_actual/med_30	merch_zip_total_amount_0_by_7
card_merch_actual/avg_14	card_zip3_actual/toal_30	merch_zip_count_0_by_14
card_merch_actual/max_14	card_zip3_count_60	merch_zip_total_amount_0_by_14
card_merch_actual/med_14	card_zip3_avg_60	merch_zip_count_0_by_30
card_merch_actual/toal_14	card_zip3_max_60	merch_zip_total_amount_0_by_30
card_merch_count_30	card_zip3_med_60	merch_zip_count_0_by_60
card_merch_avg_30	card_zip3_total_60	merch_zip_total_amount_0_by_60
card_merch_max_30	card_zip3_actual/avg_60	merch_zip_count_1_by_7
card_merch_med_30	card_zip3_actual/max_60	merch_zip_total_amount_1_by_7
card_merch_total_30	card_zip3_actual/med_60	merch_zip_count_1_by_14
card_merch_actual/avg_30	card_zip3_actual/toal_60	merch_zip_total_amount_1_by_14
card_merch_actual/max_30	Card_Merchdesc_day_since	merch_zip_count_1_by_30
card_merch_actual/med_30	Card_Merchdesc_count_0	merch_zip_total_amount_1_by_30
card_merch_actual/toal_30	Card_Merchdesc_avg_0	merch_zip_count_1_by_60
card_merch_count_60	Card_Merchdesc_max_0	merch_zip_total_amount_1_by_60

card_merch_avg_60	Card_Merchdesc_med_0	zip3_count_0_by_7
card_merch_max_60	Card_Merchdesc_total_0	zip3_total_amount_0_by_7
card_merch_med_60	Card_Merchdesc_actual/avg_0	zip3_count_0_by_14
card_merch_total_60	Card_Merchdesc_actual/max_0	zip3_total_amount_0_by_14
card_merch_actual/avg_60	Card_Merchdesc_actual/med_0	zip3_count_0_by_30
card_merch_actual/max_60	Card_Merchdesc_actual/toal_0	zip3_total_amount_0_by_30
card_merch_actual/med_60	Card_Merchdesc_count_1	zip3_count_0_by_60
card_merch_actual/toal_60	Card_Merchdesc_avg_1	zip3_total_amount_0_by_60
card_zip_day_since	Card_Merchdesc_max_1	zip3_count_1_by_7
card_zip_count_0	Card_Merchdesc_med_1	zip3_total_amount_1_by_7
card_zip_avg_0	Card_Merchdesc_total_1	zip3_count_1_by_14
card_zip_max_0	Card_Merchdesc_actual/avg_1	zip3_total_amount_1_by_14
card_zip_med_0	Card_Merchdesc_actual/max_1	zip3_count_1_by_30
card_zip_total_0	Card_Merchdesc_actual/med_1	zip3_total_amount_1_by_30
card_zip_actual/avg_0	Card_Merchdesc_actual/toal_1	zip3_count_1_by_60
card_zip_actual/max_0	Card_Merchdesc_count_3	zip3_total_amount_1_by_60
card_zip_actual/med_0	Card_Merchdesc_avg_3	card_zip3_count_0_by_7
card_zip_actual/toal_0	Card_Merchdesc_max_3	card_zip3_total_amount_0_by_7
card_zip_count_1	Card_Merchdesc_med_3	card_zip3_count_0_by_14
card_zip_avg_1	Card_Merchdesc_total_3	card_zip3_total_amount_0_by_14
card_zip_max_1	Card_Merchdesc_actual/avg_3	card_zip3_count_0_by_30
card_zip_med_1	Card_Merchdesc_actual/max_3	card_zip3_total_amount_0_by_30
card_zip_total_1	Card_Merchdesc_actual/med_3	card_zip3_count_0_by_60

card_zip_actual/avg_1	Card_Merchdesc_actual/toal_3	card_zip3_total_amount_0_by_60
card_zip_actual/max_1	Card_Merchdesc_count_7	card_zip3_count_1_by_7
card_zip_actual/med_1	Card_Merchdesc_avg_7	card_zip3_total_amount_1_by_7
card_zip_actual/toal_1	Card_Merchdesc_max_7	card_zip3_count_1_by_14
card_zip_count_3	Card_Merchdesc_med_7	card_zip3_total_amount_1_by_14
card_zip_avg_3	Card_Merchdesc_total_7	card_zip3_count_1_by_30
card_zip_max_3	Card_Merchdesc_actual/avg_7	card_zip3_total_amount_1_by_30
card_zip_med_3	Card_Merchdesc_actual/max_7	card_zip3_count_1_by_60
card_zip_total_3	Card_Merchdesc_actual/med_7	card_zip3_total_amount_1_by_60
card_zip_actual/avg_3	Card_Merchdesc_actual/toal_7	Card_Merchdesc_count_0_by_7
card_zip_actual/max_3	Card_Merchdesc_count_14	Card_Merchdesc_total_amount_0_by_7
card_zip_actual/med_3	Card_Merchdesc_avg_14	Card_Merchdesc_count_0_by_14
card_zip_actual/toal_3	Card_Merchdesc_max_14	Card_Merchdesc_total_amount_0_by_14
card_zip_count_7	Card_Merchdesc_med_14	Card_Merchdesc_count_0_by_30
card_zip_avg_7	Card_Merchdesc_total_14	Card_Merchdesc_total_amount_0_by_30
card_zip_max_7	Card_Merchdesc_actual/avg_14	Card_Merchdesc_count_0_by_60
card_zip_med_7	Card_Merchdesc_actual/max_14	Card_Merchdesc_total_amount_0_by_60
card_zip_total_7	Card_Merchdesc_actual/med_14	Card_Merchdesc_count_1_by_7
card_zip_actual/avg_7	Card_Merchdesc_actual/toal_14	Card_Merchdesc_total_amount_1_by_7
card_zip_actual/max_7	Card_Merchdesc_count_30	Card_Merchdesc_count_1_by_14
card_zip_actual/med_7	Card_Merchdesc_avg_30	Card_Merchdesc_total_amount_1_by_14
card_zip_actual/toal_7	Card_Merchdesc_max_30	Card_Merchdesc_count_1_by_30
card_zip_count_14	Card_Merchdesc_med_30	Card_Merchdesc_total_amount_1_by_30
card_zip_avg_14	Card_Merchdesc_total_30	Card_Merchdesc_count_1_by_60

card_zip_max_14	Card_Merchdesc_actual/avg_30	Card_Merchdesc_total_amount_1_by_60
card_zip_med_14	Card_Merchdesc_actual/max_30	Merchnum_desc_count_0_by_7
card_zip_total_14	Card_Merchdesc_actual/med_30	Merchnum_desc_total_amount_0_by_7
card_zip_actual/avg_14	Card_Merchdesc_actual/toal_30	Merchnum_desc_count_0_by_14
card_zip_actual/max_14	Card_Merchdesc_count_60	Merchnum_desc_total_amount_0_by_14
card_zip_actual/med_14	Card_Merchdesc_avg_60	Merchnum_desc_count_0_by_30
card_zip_actual/toal_14	Card_Merchdesc_max_60	Merchnum_desc_total_amount_0_by_30
card_zip_count_30	Card_Merchdesc_med_60	Merchnum_desc_count_0_by_60
card_zip_avg_30	Card_Merchdesc_total_60	Merchnum_desc_total_amount_0_by_60
card_zip_max_30	Card_Merchdesc_actual/avg_60	Merchnum_desc_count_1_by_7
card_zip_med_30	Card_Merchdesc_actual/max_60	Merchnum_desc_total_amount_1_by_7
card_zip_total_30	Card_Merchdesc_actual/med_60	Merchnum_desc_count_1_by_14
card_zip_actual/avg_30	Card_Merchdesc_actual/toal_60	Merchnum_desc_total_amount_1_by_14
card_zip_actual/max_30	Merchnum_desc_day_since	Merchnum_desc_count_1_by_30
card_zip_actual/med_30	Merchnum_desc_count_0	Merchnum_desc_total_amount_1_by_30
card_zip_actual/toal_30	Merchnum_desc_avg_0	Merchnum_desc_count_1_by_60
card_zip_count_60	Merchnum_desc_max_0	Merchnum_desc_total_amount_1_by_60
card_zip_avg_60	Merchnum_desc_med_0	Card_Merchnum_desc_count_0_by_7
card_zip_max_60	Merchnum_desc_total_0	Card_Merchnum_desc_total_amount_0_by_7
card_zip_med_60	Merchnum_desc_actual/avg_0	Card_Merchnum_desc_count_0_by_14
card_zip_total_60	Merchnum_desc_actual/max_0	Card_Merchnum_desc_total_amount_0_by_14
card_zip_actual/avg_60	Merchnum_desc_actual/med_0	Card_Merchnum_desc_count_0_by_30
card_zip_actual/max_60	Merchnum_desc_actual/toal_0	Card_Merchnum_desc_total_amount_0_by_30
card_zip_actual/med_60	Merchnum_desc_count_1	Card_Merchnum_desc_count_0_by_60

card_zip_actual/toal_60	Merchnum_desc_avg_1	Card_Merchnum_desc_total_amount_0_by_60
merch_zip_day_since	Merchnum_desc_max_1	Card_Merchnum_desc_count_1_by_7
merch_zip_count_0	Merchnum_desc_med_1	Card_Merchnum_desc_total_amount_1_by_7
merch_zip_avg_0	Merchnum_desc_total_1	Card_Merchnum_desc_count_1_by_14
merch_zip_max_0	Merchnum_desc_actual/avg_1	Card_Merchnum_desc_total_amount_1_by_14
merch_zip_med_0	Merchnum_desc_actual/max_1	Card_Merchnum_desc_count_1_by_30
merch_zip_total_0	Merchnum_desc_actual/med_1	Card_Merchnum_desc_total_amount_1_by_30
merch_zip_actual/avg_0	Merchnum_desc_actual/toal_1	Card_Merchnum_desc_count_1_by_60
merch_zip_actual/max_0	Merchnum_desc_count_3	Card_Merchnum_desc_total_amount_1_by_60