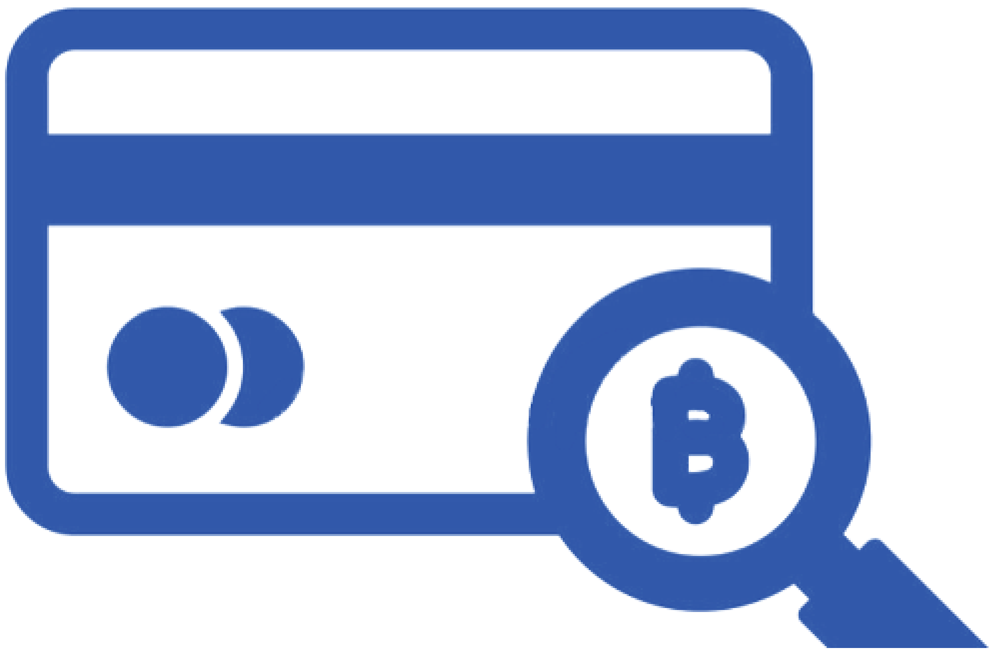
CREDIT CARD TRANSACTION FRAUD DETECTION



Rady | UC San Diego School of Management

Team 2: Nate Errez, Min Kim, Sayaka Kuwayama, Gerardo Plascencia, Yao Wan

6/11/21

Table of Contents

Card Transaction Data Executive Summary

Nate Errez, Min Kim, Sayaka Kuwayama, Gerardo Plascencia, Yao Wan

Overview

Credit card fraud is a form of fraud when using a credit card to obtain goods or services, make payments, or sign up for cards using false information. We were hired by a government entity to build a supervised fraud model to identify and predict fraudulent credit card transactions.
We were given 12 months worth of data by the US government organization which contained 96,753 records and 10 fields. After applying proper methodology such as data cleaning, variable creation, feature selection, training, and testing, we were able to build effective supervised machine learning algorithms to detect fraudulent transactions.

The report details our methodology for creating the supervised fraud model in the below steps:

1. Data Cleaning - remove exclusion and outliers, fill in missing values
2. Variable Creation - create new candidate variables that could help detect fraud
3. Feature Selection - filter and perform a wrapper to obtain our top 30 variables
4. Model Building - train, test, and tune models to obtain the most effective fraud model
5. Results and Analysis - select the final model and make recommendations for future analysis and applications

Candidate Variables and Feature Selection

A critical step of this project was building candidate variables that would help us detect credit card fraud. Original variables were used to create variables that can be categorized into 5 groups: amount variables, frequency variables, days-since variables, velocity change variables, and target encoded variables. We measured each variable by univariate KS and FDR, then sorted the variables by the average rank of these two measures. After filtering for the top 80 variables, we used a wrapper to get the top 30 variables for modeling in the order of multivariate importance.

Model Algorithms

We attempted various algorithms, including Logistic Regression, Neural Networks, Random Forest, and XGBoost. We manually tuned our model's hyperparameters to observe the best FDR scores on the training, test, and out-of-time (OOT) data. Each model was trained and evaluated 10 times, and the final score was an average of the 10 trials.

Results

After trying the different models, the Random Forest with 30 variables gave us the best results, and was selected as our final model. Using our final model, we can detect 69.4% of fraud when looking at 3% of the transactions.

1. Data Description

1.1 File description

The file “card transactions.csv” comes from actual credit card purchases from a US government organization. It holds 96,753 records with just 10 fields. It covers a time period of 12 months, from 1/1/10 to 12/31/10. Of the 10 fields, there is a column labelled ‘fraud’ to be used as a dependent variable. A ‘0’ indicates no fraud and fraud cases are recorded with ‘1’. Amongst almost 100,000 records, there are only about 1059 fraud cases which make up just a minute number. In deciding whether the case is fraudulent or not, it was ‘made up’ by Professor Coggeshall, based on his experience with card transaction fraud.

1.2 Summary Statistics

Categorical Variables

Field	Type	% Populated	# with Zero Value	# Unique values	Mode	Mode Count
Recnum	Identifier	100%	0	96,753	N/A	N/A
Cardnum	Identifier	100%	0	1,645	5142148452	1,192
Date	Datetime	100%	0	365	2/28/10	684
Merchnum	Identifier	96.51%	231	13,091	930090121224	9,310
Merch Description	Identifier	100%	0	13,126	GSA-FSS-ADV	1,688
Merch State	Categorical	98.76%	0	227	TN	12,035
Merch Zip	Categorical	95.19%	0	4,567	38118	11,868
Transtype	Categorical	100%	0	4	P	96,398
Fraud	Dependent Variable	100%	95,694	2	0	95,694

Numerical Variable

Field	Type	% Populated	# with Zero Value	Mean	Std	Max	Min
Amount	Numeric	100%	0	427.89	10,006.09	3,102,045.53	0.01

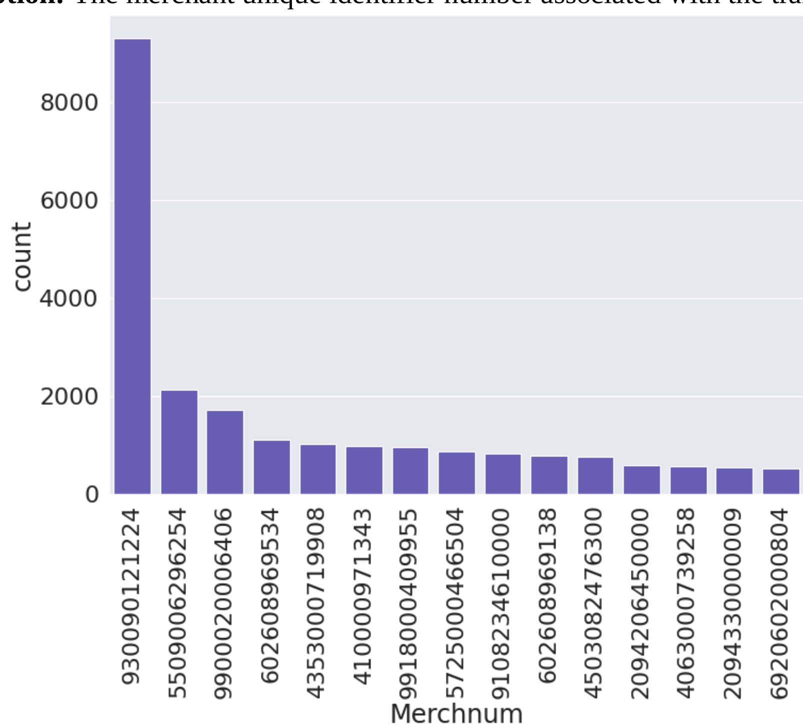
Taking a look at the summary statistics, we can see that there are a lot of identifiers in this data letting us identify each of the transactions thoroughly. Because there is a high probability that those who've purchased before will purchase again, the column labeled "number of unique values" in the categorical variables summary chart shows a variety of numbers for different fields. As for the only numerical field in the data, it represents the amount of money spent on its transaction.

1.3 Data Quality Report

To visualize our data, we built histograms to see the distribution of records for each field. Of the 10 fields, three of the fields stood out as their distributions were unbalanced. These fields held a large number of records for a specific Merch number, Merch zip code and transtype, creating a huge difference from the rest of the records.

Field Name: Merchnum

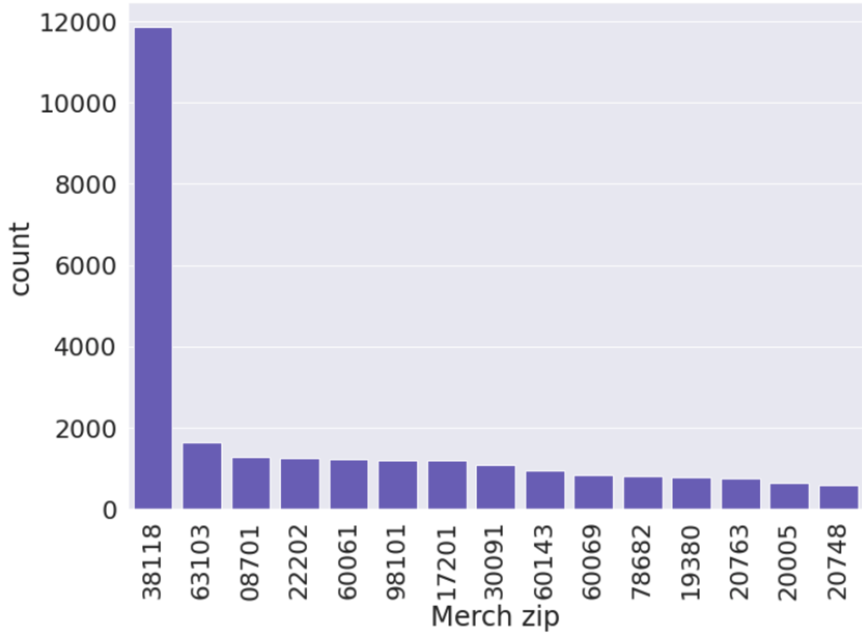
Description: The merchant unique identifier number associated with the transaction.



This field for Merch number shows that the merch number of '930090121224' holds the most transactions. In the data, this particular merchant has over 9000 transactions on its own. As it is way above all the other merchants, we can see this merchant to be a bit suspicious. However, when getting deeper into this merchant, we found out that this identification number belongs to 'FEDEX', which is a huge multinational delivery services company, which provides a reasoning for why they've got more transactions than others.

Field Name: Merch zip

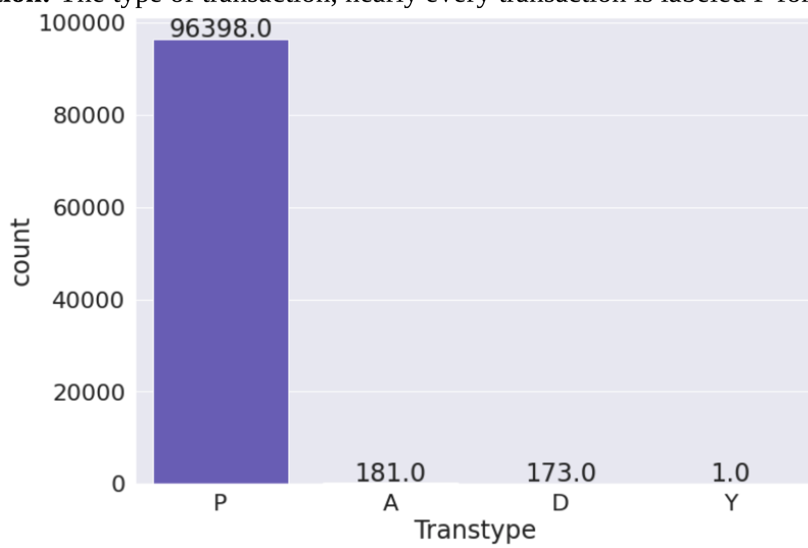
Description: The merchant's registered location state.



Same idea applies to this field, which represents the merchant's zip code. The histogram shows that there are almost 12000 records that come from the '38118' zip code. Similar to the merchant number, because this zip code is extremely higher than the rest, we can believe it needs to be looked into. Once looking through the data, this graph also became reasonable as the zip code belongs to the state of Tennessee, where all the 'FEDEX' shipments once again are from.

Field Name: Transtype

Description: The type of transaction, nearly every transaction is labeled P for purchase.



Amongst the transtype, the other types than 'P' for purchase seems to be missing but this is because the numbers for the other types are a lot lower than for that one category. The 'A' refers to authorization (funds on hold), 'D' for delayed capture (delayed authorization to make sure customers have enough credit to cover a transaction) and 'Y' for unknown. The issue for this field will be covered when we clean our data.

**Full DQR can be found in the appendix*

1.4 Idiosyncrasies of the Data

From what we saw in the DQR, we realized that this particular data is peculiar. Starting off with the FEDEX records, these records make up a lot of the transactions, which can become an obstacle later in the process of detecting fraud. It would be helpful to remove them in the future steps to come up with thorough detection of fraudulent records.

Second, when looking at the data for the amount for transactions, we can see that there is an outlier that is way above all amounts. It belongs to the record number of ‘52715’, with an amount of $3,102,045.53. We will deal with this specific record in the data cleaning process.

Lastly, this data is a collection of transactions that accrued over time. Meaning that we have to take into consideration the dates of when the transactions took place. When coming up with variables and models of detection, it is important to remember that these are real-time, that we have to treat time correctly. In calculating this type of data, we were to only use the data of the past for that record along with that record itself. Data that happened after will not be seen when the model gets used. Taking this into account, building our models, we will hold out the last 4 months of data, known as the out of time (OOT) sample to reserve the most recent data time frame as our validation data to evaluate the implementation of the previous 8 months of test/train set data.

2. Data Cleaning

As we saw in the Data Description section, there were several idiosyncrasies in the data that needed to be addressed before building candidate variables.

2.1 Exclusions

Starting with the Transtype field, as we saw, the amount of purchase largely outweighed the other types. So, the first exclusion we made was getting rid of all the other types of transactions and just using the purchase type transactions to go forth with the data.

Then, as mentioned previously, we focused on the single large transaction outlier for the record number of '52715'. Due to the fact that the payment amount is hundreds of standard deviations above the mean, we also decided to eliminate this specific transaction.

2.2 Imputation

In order to produce effective models in detecting card transaction frauds, it would be crucial to fill in missing values for some fields. The replacements for the missing values would be recreated by us using what the data provides. To maximize our algorithm's accuracy in developing fraud scores, it would be important that we create the most appropriate and consistent values with the existing data.

The fields that contain missing values are Merch state (merchant state), Merch num (merchant number) and Merch zip (merchant zip code). Merch state has a total of 1194 values missing. To fill in values for this field, we first looked if the record had a correlating zip code. If it did, then we went ahead and used the state that the zip code is a part of. For zip codes that ranged from 00600 - 00799 and 00900 - 00999, we wrote the state to be 'PR', representing Puerto Rico. After filling the values in by these two methods, with the values that were still left unfilled, we used the mode of the merch number or merch description for the record. If there were records still left, we placed them with 'Unk' for unknown.

There were 3373 total missing for the field of merchant number. If the number was placed with a 0, it indicated that the number was not given. So, for those values, we replaced them with a NaN. For other missing values, we filled them in using the mode of merchant description. Similar to the merchant state, the rest of the values still missing were placed with an 'Unk' for unknown.

Lastly, the merchant zip code had 4616 missing values. We used the mode of merchant number to replace missing values. The rest were filled with an unknown.

There were two exceptions to imputation where we viewed them differently than the rest of the missing values. For the merchant description of ‘Retail credit adjustment’ and ‘Retail debit adjustment’, the merchant number, merchant state and merchant zip code were all null. So, we concluded to just change the three fields for all records that belong to these two descriptions as unknown.

3. Candidate Variables

With a set of cleaned data, we then continued on to building our candidate variables. These variables formed a pool of possibilities that could be implemented into our model algorithms.

3.1 Building variables

In creating our variables, we grouped some fields within our data. The entities that we came up with are 'Cardnum', 'Merchnum', 'card_merch', 'card_zip', 'card_state', 'merch_zip', 'merch_state', 'amount_bin_merch' and 'amount_bin_card'.

Aside from the original given data for 'Cardnum' ad 'Merchnum', the table below summarizes our creation in a table. Another thing to note is that for the last two fields, the amount_bin refers to the 10 bins we created for the 'Amount' variable split in ascending order. Smaller bin number would indicate the amount spent on that transaction is lower. We created these fields because we believed that by binning the amount spent with each merchnum and cardnum, it would give us a better visualization of how much was spent on transactions from each identification method.

Field Name	**Grouped Entities**
card_merch	Cardnum + merchnum
card_zip	Cardnum + merch zip
card_state	Cardnum + merch state
merch_zip	Merchnum + merch zip
merch_state	Merchnum + merch state
amount_bin_merch	Amount_bin + merchnum
amount_bin_card	Amount_bin + cardnum

Using the 9 fields we have now created, we developed 4 different kinds of variables. The **amount variable** represents the 'Average, maximum, median, total, actual/average, actual/maximum, actual/median, actual/total' amount by/at the 9 fields (card_merch, card_zip, card_state, merch_zip, merch_state, amount_bin_merch, amount_bin_card) over the past '0 days, 1 day, 3 days, 7 days, 14 days, 30 days'. We multiplied the 8 types of mathematical views of the amount by the 9 fields with the 6 groupings of numbers of certain days to get 432 amount variables.

Secondly, the **frequency variables** were calculated by taking the number of transactions with the 9 fields over the past 6 groupings of numbers of certain days mentioned in the amount variable. There were 54 frequency variables created.

Third, **days-since variables** were made up of the 9 fields, variables, that we created previously. These were produced by taking the date of the most recent transaction with the same 9 fields (card_merch, card_zip, card_state, merch_zip, merch_state, amount_bin_merch, amount_bin_card) subtracted from the current date.

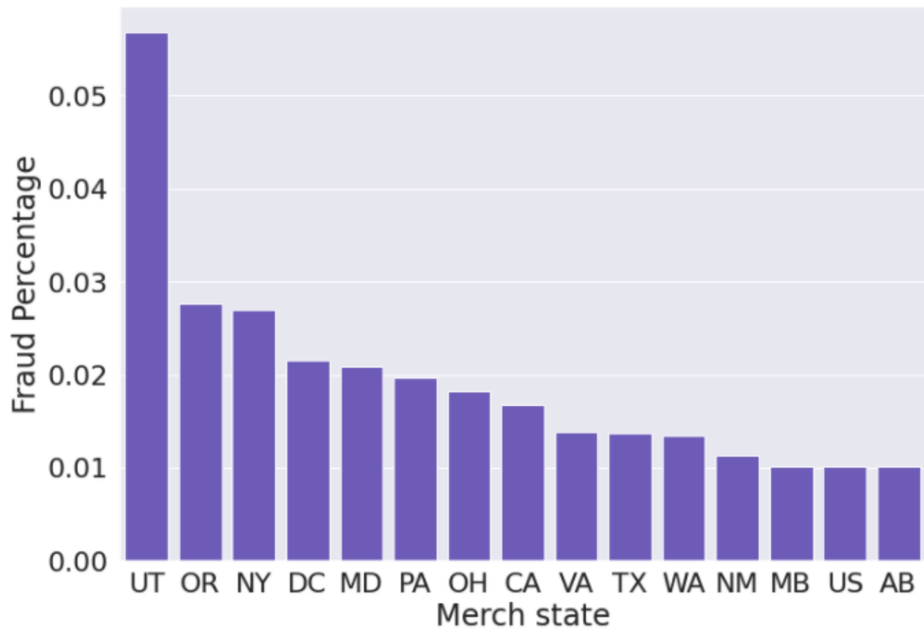
Lastly, the **velocity change variables** were built through taking the 'number, amount' of transactions with the same 9 variables over the past 6 groupings of days. As we multiplied the 2, 9 and the 6 we had produced 108 velocity change variables.

In summing up all 4 types of variables (432 + 54 + 9 + 108), we reached 603 total variables.

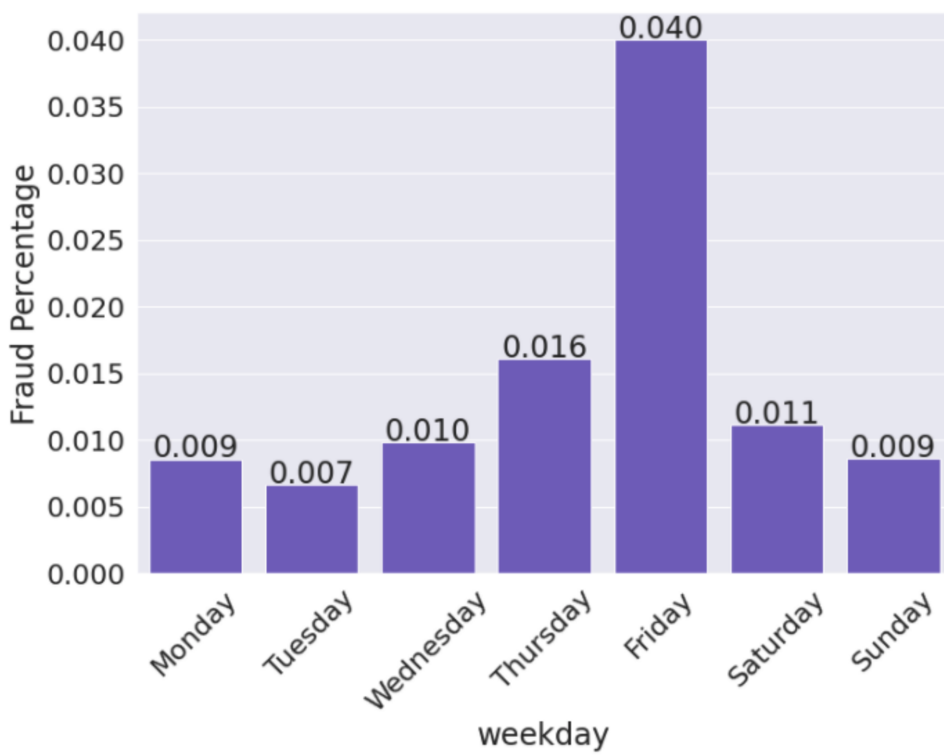
3.2 Target Encoding

Target encoding can be understood as an alternative to dummy variables when including categorical variables in a non-linear model. We can get the mean of the target variable and replace the categorical value with that result. Through this type of encoding, the benefits are that we are able to directly encode what we are trying to predict and it is the easiest on the model to figure out the relationship y = f(x). However, the drawbacks are that there could be a loss of possible interaction information along with dangers of overfitting. To make sure that we prevent the risk of overfitting, we took out the OOT sample, data for the last 4 months, when calculating the values. Also, we used a smoothing function (Appendix Table 3.) to smooth out the encoding as the feature is dependent on the target and if not smoothed, it would add on too much weight.

The first target encoded variable examined the likelihood of fraud for each of the merchant states. The graph below shows the top 15 states with the highest fraud percentage.



The second target encoded variable examined the likelihood of fraud for each day of the week. We saw how ‘Friday’s are the most likely for fraudsters to commit fraudulent transactions.



3.3 Final Candidate Variables

In our variable creation, along with the 603 variables previously developed through the 4 different types of variables, by adding in the 2 target encoded variables, our final candidate variables totaled up to 605 variables. These mentioned below are the first 10 variables. (The whole list can be found in the appendix)

1	Cardnum_day_since
2	Cardnum_count_0
3	Cardnum_avg_0
4	Cardnum_max_0
5	Cardnum_med_0
6	Cardnum_total_0
7	Cardnum_actual/avg_0
8	Cardnum_actual/max_0
9	Cardnum_actual/med_0
10	Cardnum_actual/toal_0

4. Feature Selection Process

4.1 Motivation

After creating the candidate variables, we then moved on to the process of selecting variables to use in the models. This process is highly critical when we plan to use non-linear models since their performance suffers when dealing with high dimensional data. The potentially-non intuitive phenomena that occur with high dimensions are popularly known as the *Curse of Dimensionality*, and they include:

- Data quickly becomes sparse
    - I.e., every time we add another dimension, the density of data in any particular location in space goes down by a factor of a number of bins
- All points become outliers
    - E.g., let's say 10% of data points in 1-dimensional data close to the boundaries are outliers (total of 20% data points are the outliers.)
    - Every time we add another dimension, the "inner square" (i.e., where the points are not outliers) is reduced by a factor of 0.8 (1-0.2.)
- Number of records needed to observe true nonlinearities increase exponentially
    - E.g., The minimum number of points to observe nonlinearity in 1-dimensional space is 3 (3^1), and it is 9 (3^2) for the 2-dimensional space.
    - Generalizing this behavior for a dimension S, the minimum points required to see nonlinearity is 3 to the S (3^S).
    - However, this number 3 is when the data points are placed perfectly for us to recognize the nonlinearly; in reality, there is noise in data, and it takes about 10 points to recognize the nonlinearly. Therefore, the more reasonable estimate for the number of points needed to observe true nonlinearity for S-dimensional data is 10 to the S (10^S).

These phenomena cause the non-linear models to fit to noise rather than the true nonlinearities, and thus it is essential for us to select only the variables that carry substantial information and minimize the dimensionality of the data.

4.2 Implementation

There are three major ways to implement variable selection, where they can be used either by itself and/or combined. Please see below for the brief explanation of each method.

    - Filter

    - Use a mathematical measure for the importance of each independent variable toward the dependent variable, and throw away those that are not too significant
    - Common filters include Pearson correlation, univariate Kolmogorov-Smirnov, and univariate model performance (e.g., FDR)
- Wrapper
    - Use a model (linear/ nonlinear) wrapped around the process
    - Common wrappers include Forward/ Backward Stepwise Selection and General Stepwise Selection
- Embedded
    - The process of feature selection happens within the construction of machine learning models
    - The example includes the use of Decision tree and regularizations (e.g., ridge, lasso)

At this point of the project, we utilized Filter and Wrapper methods to select the features to feed in the models.

4.2.1. Filter

While there are multiple ways to implement the filter method, we decided to use the univariate Kolmogorov-Smirnov (KS) and Fraud Detection Rate (FDR) as they are some of the most commonly used methods.

For the univariate KS, although we used the *scipy* library to implement it, what happens behind the scenes is that it plots separate normalized distributions for the two populations (i.e., goods and bads.) Then, it calculates the measure of how different the two curves are; the farther the curves are from each other, the more effective the variable is for separating the two populations and thus significant. Using the *stats* from *scipy* library, we calculated this measure of importance for each independent variable and assigned a rank accordingly.

As for the FDR, we calculated the univariate FDR of 3%. In particular, what we did was for every variable, we sorted data by that variable, looked at the top and bottom 3% of data, and calculated the fraction of fraud in each of the top and bottom. Then, we took the maximum of these two to determine the fraud detection rate for each variable. After calculating the FDR of 3% for all the variables, we then ranked them based on the value of the FDR.

After we obtained two separate ranks from the univariate KS and FDR in the above steps, we took the average of these two ranks to generate the overall ranks. Then, we sorted all the variables by this overall rank and kept the top 80 variables to pass down to the next step of the feature selection, the wrapper.

4.2.2 Wrapper

Among the classic wrappers, we utilized forward selection. The process of the forward selection is the following:

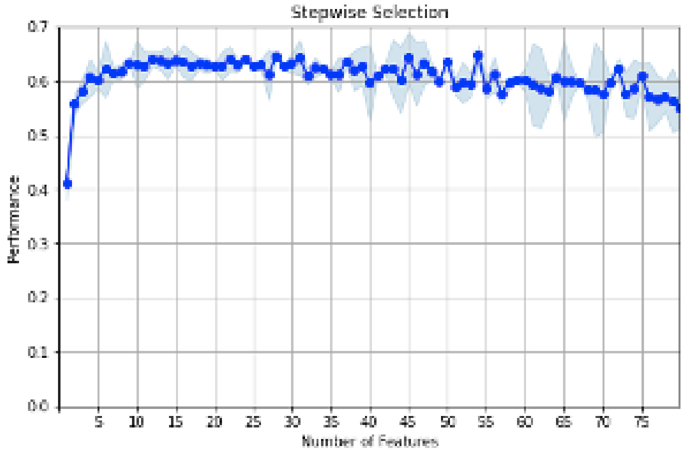
1. Start with N separate 1-dimensional models
    a. Evaluate the models (e.g., using RMSE)
    b. Keep the best variable
2. Using the best variable from the previous step, build N-1 separate 2-dimensional models
    a. Evaluate the model (e.g., using RMSE)
    b. Keep the new "best variable" among the N-1 variables that were tried
3. Repeat this process (i.e., keep the new "best variable" in each step) until we stop seeing improvements in the metrics we're using

Since the forward selection always looks for the best next thing to do given what has been chosen, this algorithm is a greedy search. While forward selection, and thus a greedy search, does not guarantee to find the global optimum, it does accomplish to find the local minimum: the good subsets of the variables.

Moreover, due to the nature of the process, forward selection removes correlation problems. This is because if a variable is highly correlated with another variable that has already been selected to be kept, that variable would not contribute to the model and consequently will not be chosen as the new best variable. Therefore, while the purpose of the forward selection initially is to reduce dimensionality, we are also able to remove correlation.

One caveat with this characteristic of forward selection is that it is expected to produce different results every time we run it. For instance, if we were to run the wrapper for multiple times and see the variable A and variable B as the first variable to be chosen over again, one might become inclined to keep both variable A and B. However, what this result suggests is that not only are they both extremely effective in predicting the dependent variable, but also, they are likely highly correlated. Therefore, it is important not to select commonly chosen variables across different runs but to use a result from a particular run.

Keeping this in mind, we ran the wrapper using the *mlxtend_feature_selection* library and a simple nonlinear model (i.e., Random Forest with 5 estimators.) The below is the plot of how the performance increases as the number of variables in the model increases.



While the plot suggests that the performance stops increasing when the number of variables is around 13, we wanted to be sure to have the most important variables. Consequently, we picked the number of variables that is short and yet conservative: 30, and the following is the list of the 30 variables we selected in the rank-order of multivariate importance.

1	'card_zip_total_7',
2	'card_state_max_7',
3	'amount_bin_card_total_7',
4	'Cardnum_total_3',
5	'card_merch_total_1',
6	'card_zip_max_14',
7	'card_state_avg_3',
8	'card_zip_total_1',
9	'card_state_avg_7',
10	'Cardnum_total_0',
11	'card_merch_total_7',

12	'Cardnum_total_7',
13	'card_merch_max_1',
14	'merch_state_total_3',
15	'card_zip_total_0',
16	'card_merch_total_3',
17	'card_merch_max_30',
18	'Cardnum_max_14',
19	'merch_zip_total_1',
20	'Merchnum_total_3',
21	'merch_zip_total_7',
22	'amount_bin_merch_total_0',
23	'card_state_max_14',
24	'card_state_total_3',
25	'card_zip_total_30',
26	'card_state_max_1',
27	'Cardnum_max_3',
28	'card_zip_max_30',
29	'amount_bin_merch_total_7',

30	'amount_bin_card_total_3'

*Before implementing the feature selection, we removed the first 2 weeks of the data since some of the variables are constructed using past days' data. We also removed the last 4 months of the data (out-of-time data) before implementing the feature selection.

5. Model Algorithms

Now that we had a promising list of variables in the rank-order of multivariate importance, we started building the models. For each of the models with a particular set of hyperparameters, we trained and evaluated the model 10 times. In particular, we split the model data into training and testing sets and computed the performance metric of Fraud Detection Rate at 3% for every trial. By taking the average of the 10 FDR results, we achieved a reasonable estimation of the performance for each model with particular sets of hyperparameters.

Below is the summary of all algorithms implemented and the corresponding performance (average FDR at 3% over 10 runs) for each algorithm and hyperparameters' combination on training, testing, and out-of-time data set.

- Logistic Regression
  - Linear models to get a baseline performance
  - *sklearn* library
  - Variations tried
    - Number of variables: 1-20, 1-25, and 1-30 in the order of multivariate importance
    - Regularization: Lasso, Ridge
    - Inverse of regularization strength: 0.1, 1
    - Solver: liblinear, lbfgs

Parameters				Average FDR at 3%		
# Variables	Regularization	C	Solver	Train	Test	OOT
20	l1	1	liblinear	0.666	0.656	0.520
20	l1	0.1	liblinear	0.669	0.660	0.523
20	l2	1	liblinear	0.663	0.656	0.511
20	l2	1	lbfgs	0.668	0.648	0.520
25	l1	1	liblinear	0.686	0.697	0.546
25	l1	0.1	liblinear	0.684	0.687	0.524

25	l2	1	liblinear	0.695	0.686	0.534
25	l2	1	lbfgs	0.692	0.685	0.537
30	l1	1	liblinear	0.709	0.688	0.548*
30	l1	0.1	liblinear	0.706	0.679	0.541
30	l2	1	liblinear	0.701	0.706	0.542
30	l2	1	lbfgs	0.697	0.690	0.528

*The highest OOT performance

- Neural Network (Multi-Layer Perceptron)
    - *sklearn* library
    - Variations tried
        - Number of variables: 1-25 and 1-30 in the order of multivariate importance
        - Layer: 1, 2
        - Node: 5, 10, 20
        - Activation function: relu, logistic
        - Ridge regularization: 0.0001, 0.001
        - Solver: stochastic gradient descent, adam,
        - Learning rate: none, constant, adaptive

Parameters							Average FDR at 3%		
# Variables	Layer	Node	Activation	Alpha	Solver	Learning rate	Train	Test	OOT
25	1	10	relu	0.0001	sgd	adaptive	0.639	0.655	0.489
25	2	10	relu	0.0001	sgd	adaptive	0.635	0.621	0.490
25	2	20	relu	0.0001	sgd	adaptive	0.654	0.651	0.503
25	2	20	relu	0.0001	adam	adaptive	0.888	0.817	0.545

25	2	20	logistic	0.0001	adam	adaptive	0.810	0.773	0.596
30	1	5	relu	0.0001	sgd	None	0.629	0.645	0.482
30	1	10	logistic	0.0001	sgd	adaptive	0.599	0.602	0.551
30	2	10	relu	0.001	adam	adaptive	0.848	0.794	0.554
30	2	10	relu	0.0001	sgd	constant	0.645	0.635	0.523
30	2	20	relu	0.0001	sgd	adaptive	0.666	0.650	0.497
30	2	20	relu	0.0001	adam	adaptive	0.895	0.813	0.558
30	2	20	logistic	0.0001	adam	adaptive	0.814	0.785	0.610*

*The highest OOT performance

- Gradient Boosted Tree
    - *sklearn* library
    - Variations tried
        - Number of variables: 1-25 and 1-30 in the order of multivariate
        - Max depth: 3, 5
        - Number of estimators: 300, 500, 800, 1000
        - Subsample: 0.7, 1
        - Max features: None, 5, 25, 30
        - Learning rate: 0.01, 0.02. 0.05, 0.1

Parameters						Average FDR at 3%		
# Variables	Max depth	Estimators	Subsample	Max features	Learning rate	Train	Test	OOT
25	3	300	1	None	0.1	0.959	0.856	0.580
25	3	300	0.7	5	0.01	0.798	0.764	0.621

25	5	300	0.7	5	0.01	0.887	0.789	0.632
25	5	500	0.7	5	0.01	0.940	0.818	0.628
25	5	800	0.7	25	0.05	0.999	0.861	0.597
30	5	800	0.7	30	0.02	0.994	0.867	0.611
30	5	1000	0.7	30	0.02	0.913	0.791	0.547
30	5	800	0.7	30	0.01	0.979	0.853	0.629*

*The highest OOT performance

- Random Forest
    - *sklearn* library
    - Variations tried
        - Number of variables: 1-30 in the order of multivariate
        - Max depth: None, 10, 20, 30
        - Number of estimators: 30, 50, 100, 150
        - Max features: 5, 10, 25, 30
        - Minimum sample split: 2, 50, 200, 300, 500
        - Minimum sample leaf: 1, 20, 30

Parameters						Average FDR at 3%		
# Variables	Max depth	Estimators	Max features	Min samples split	Min samples leaf	Train	Test	OOT
30	None	30	5	2	1	1.000	0.841	0.618
30	10	50	5	2	1	0.919	0.829	0.641
30	20	100	5	2	1	1.000	0.862	0.633
30	30	150	5	50	20	0.920	0.832	0.660

30	30	150	5	200	30	0.861	0.804	0.665
30	30	150	10	200	20	0.858	0.829	0.679
30	30	150	10	300	30	0.839	0.803	0.676
30	30	150	10	500	30	0.813	0.783	0.673
30	30	150	25	200	20	0.869	0.807	**0.680***
30	30	150	30	300	20	0.854	0.804	0.680

*The highest OOT performance out of all algorithms implemented in the model building process

- Decision Tree
  - *sklearn* library
  - Variations tried
    - Number of variables: 1-30 in the order of multivariate
    - Max depth: None, 10, 20
    - Minimum sample split: 2, 100, 300
    - Minimum sample leaf: 1, 30, 60

Parameters				Average FDR at 3%		
# Variables	Max depth	Min samples split	Min samples leaf	Train	Test	OOT
30	None	2	1	1.000	0.679	0.412
30	20	100	30	0.896	0.786	0.636
30	20	300	60	0.808	0.752	0.638
30	10	300	30	0.804	0.738	0.652*
30	10	300	60	0.800	0.776	0.617

*The highest OOT performance

Among the highest performing models with particular hyperparameter combinations (i.e., in yellow), the one from Random Forest algorithm yielded the highest fraud detection rate of 3 % on the out-of-time data.

6. Results

We selected a Random Forest (30 variables, max depth = 30, n_estimator = 150, max_features = 25, min_samples_split = 200, min_samples_leaf = 20) as our final model since it obtained the highest FDR at 3% of the transactions. It achieved an FDR at 3% of 88.19% on the training data and 69.4% on the OOT data.

We sorted the training, testing, and OOT results of our model by the Fraud Detection Rate. Each set of results were split into 100 bins to build detailed model performance tables. The tables display the number of records in each population, the number of good records, number of bad records, and the fraud rate. The green section shows the statistics within each bin, and the blue section shows the cumulative statistics for everything up to and including that bin. The top 20 bins (20% of the population) are shown below.

The highest fraud scoring bin contains the highest number of bads detected in each of the three tables. Approximately 60% of the bad records were captured in the 1st bin, and we see a nice monotonically decreasing number of bad records and bads % as we go down the table. Our cumulative KS score stays high and the cumulative False Positive Ratio remains relatively low in the top 20 bins. This shows our model does a good job at differentiating between good records and bad records, and detecting fraudulent card transactions.

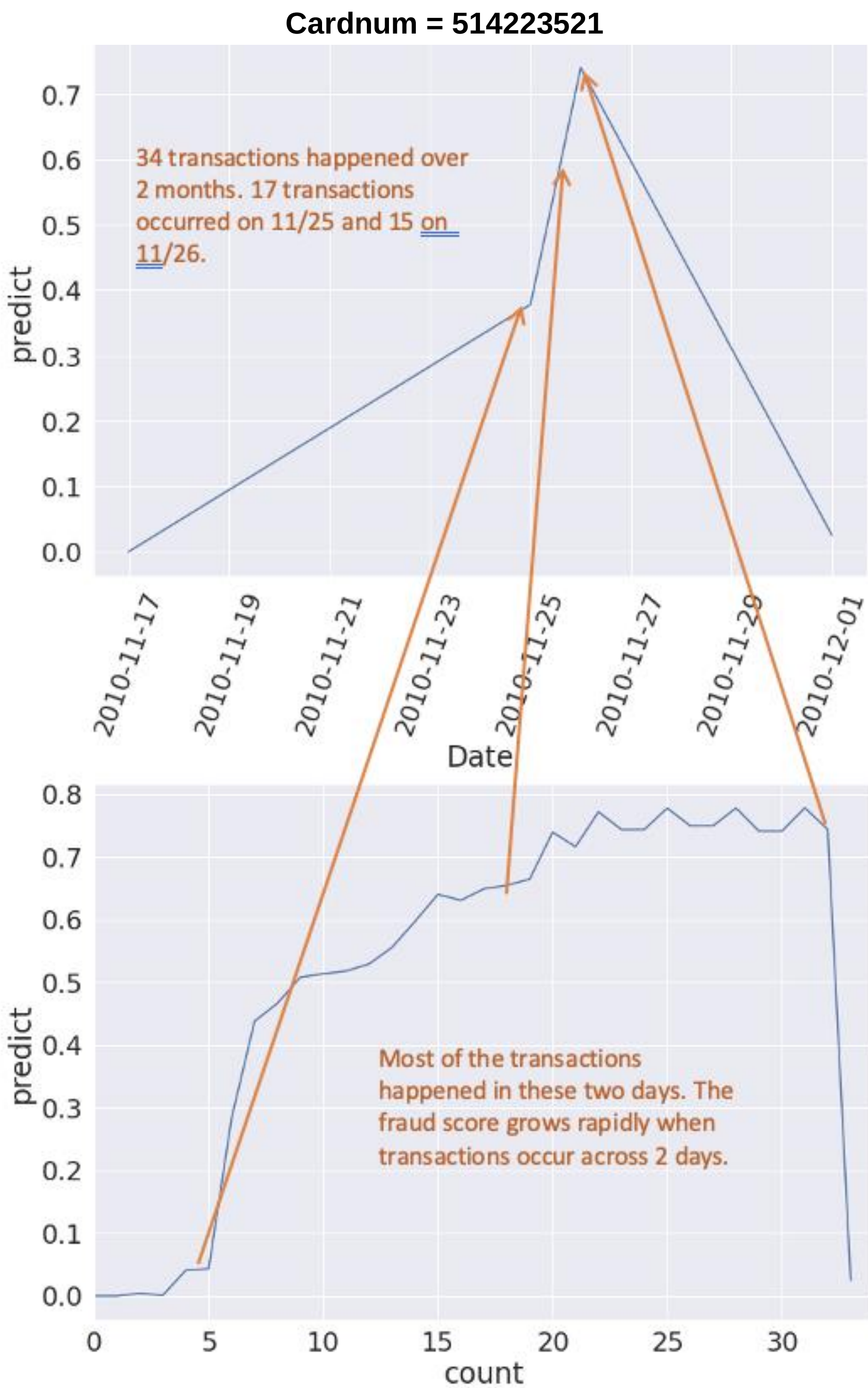
Training	# Records		# Goods		# Bads		Fraud Rate					
	48,332		47,836		496		0.0103					
	Bin Statistics					Cumulative Statistics						
Population Bin %	# Records	# Goods	# Bads	% Goods	% Bads	Cumulative Records	Cumulative Goods	Cumulative Bads	% Good	FDR	KS	FPR
1	484	176	308	36%	64%	484	176	308	0.4	62.7	62.3	0.57
2	483	410	73	85%	15%	967	586	381	1.2	77.6	76.4	1.54
3	483	431	52	89%	11%	1450	1017	433	2.1	88.2	86.1	2.35
4	484	466	18	96%	4%	1934	1483	451	3.1	91.9	88.8	3.29
5	483	459	24	95%	5%	2417	1942	475	4.1	96.7	92.6	4.09
6	483	471	12	98%	2%	2900	2413	487	5	99.2	94.2	4.95
7	484	482	2	100%	0%	3384	2895	489	6.1	99.6	93.5	5.92
8	483	482	1	100%	0%	3867	3377	490	7.1	99.8	92.7	6.89
9	483	483	0	100%	0%	4350	3860	490	8.1	99.8	91.7	7.88
10	484	484	0	100%	0%	4834	4344	490	9.1	99.8	90.7	8.87
11	483	483	0	100%	0%	5317	4827	490	10.1	99.8	89.7	9.85
12	483	483	0	100%	0%	5800	5310	490	11.1	99.8	88.7	10.84
13	484	483	1	100%	0%	6284	5793	491	12.1	100	87.9	11.8
14	483	483	0	100%	0%	6767	6276	491	13.1	100	86.9	12.78
15	483	483	0	100%	0%	7250	6759	491	14.1	100	85.9	13.77
16	483	483	0	100%	0%	7733	7242	491	15.1	100	84.9	14.75
17	484	484	0	100%	0%	8217	7726	491	16.1	100	83.9	15.74
18	483	483	0	100%	0%	8700	8209	491	17.2	100	82.8	16.72
19	483	483	0	100%	0%	9183	8692	491	18.2	100	81.8	17.7
20	484	484	0	100%	0%	9667	9176	491	19.2	100	80.8	18.69

Testing	# Records		# Goods		# Bads		Fraud Rate					
	20,714		20,507		207		0.0100					
	Bin Statistics					Cumulative Statistics						
Population Bin %	# Records	# Goods	# Bads	% Goods	% Bads	Cumulative Records	Cumulative Goods	Cumulative Bads	% Good	FDR	KS	FPR
1	208	91	117	44%	56%	208	91	117	0.4	55.2	54.8	0.78
2	207	170	37	82%	18%	415	261	154	1.3	72.6	71.3	1.69
3	207	194	13	94%	6%	622	455	167	2.2	78.8	76.6	2.72
4	207	197	10	95%	5%	829	652	177	3.2	83.5	80.3	3.68
5	207	202	5	98%	2%	1036	854	182	4.2	85.8	81.6	4.69
6	207	205	2	99%	1%	1243	1059	184	5.2	86.8	81.6	5.76
7	207	197	10	95%	5%	1450	1256	194	6.1	91.5	85.4	6.47
8	208	206	2	99%	1%	1658	1462	196	7.1	92.5	85.4	7.46
9	207	206	1	100%	0%	1865	1668	197	8.1	92.9	84.8	8.47
10	207	204	3	99%	1%	2072	1872	200	9.1	94.3	85.2	9.36
11	207	207	0	100%	0%	2279	2079	200	10.1	94.3	84.2	10.4
12	207	205	2	99%	1%	2486	2284	202	11.1	95.3	84.2	11.31
13	207	207	0	100%	0%	2693	2491	202	12.2	95.3	83.1	12.33
14	207	207	0	100%	0%	2900	2698	202	13.2	95.3	82.1	13.36
15	207	206	1	100%	0%	3107	2904	203	14.2	95.8	81.6	14.31
16	208	208	0	100%	0%	3315	3112	203	15.2	95.8	80.6	15.33
17	207	207	0	100%	0%	3522	3319	203	16.2	95.8	79.6	16.35
18	207	206	1	100%	0%	3729	3525	204	17.2	96.2	79	17.28
19	207	207	0	100%	0%	3936	3732	204	18.2	96.2	78	18.29
20	207	207	0	100%	0%	4143	3939	204	19.2	96.2	77	19.31

Out of Time	# Records		# Goods		# Bads		Fraud Rate					
	27,351		26,995		356		0.0130					
	Bin Statistics					Cumulative Statistics						
Population Bin %	# Records	# Goods	# Bads	% Goods	% Bads	Cumulative Records	Cumulative Goods	Cumulative Bads	% Good	FDR	KS	FPR
1	274	112	162	41%	59%	274	112	162	0.4	45.5	45.1	0.69
2	274	225	49	82%	18%	548	337	211	1.2	59.3	58.1	1.6
3	273	240	33	88%	12%	821	577	244	2.1	68.5	66.4	2.36
4	274	265	9	97%	3%	1095	842	253	3.1	71.1	68	3.33
5	273	264	9	97%	3%	1368	1106	262	4.1	73.6	69.5	4.22
6	274	261	13	95%	5%	1642	1367	275	5.1	77.2	72.1	4.97
7	273	267	6	98%	2%	1915	1634	281	6.1	78.9	72.8	5.81
8	274	272	2	99%	1%	2189	1906	283	7.1	79.5	72.4	6.73
9	273	268	5	98%	2%	2462	2174	288	8.1	80.9	72.8	7.55
10	274	269	5	98%	2%	2736	2443	293	9	82.3	73.3	8.34
11	273	271	2	99%	1%	3009	2714	295	10.1	82.9	72.8	9.2
12	274	273	1	100%	0%	3283	2987	296	11.1	83.1	72	10.09
13	273	270	3	99%	1%	3556	3257	299	12.1	84	71.9	10.89
14	274	271	3	99%	1%	3830	3528	302	13.1	84.8	71.7	11.68
15	273	272	1	100%	0%	4103	3800	303	14.1	85.1	71	12.54
16	274	273	1	100%	0%	4377	4073	304	15.1	85.4	70.3	13.4
17	273	271	2	99%	1%	4650	4344	306	16.1	86	69.9	14.2
18	274	272	2	99%	1%	4924	4616	308	17.1	86.5	69.4	14.99
19	273	273	0	100%	0%	5197	4889	308	18.1	86.5	68.4	15.87
20	274	271	3	99%	1%	5471	5160	311	19.1	87.4	68.3	16.59

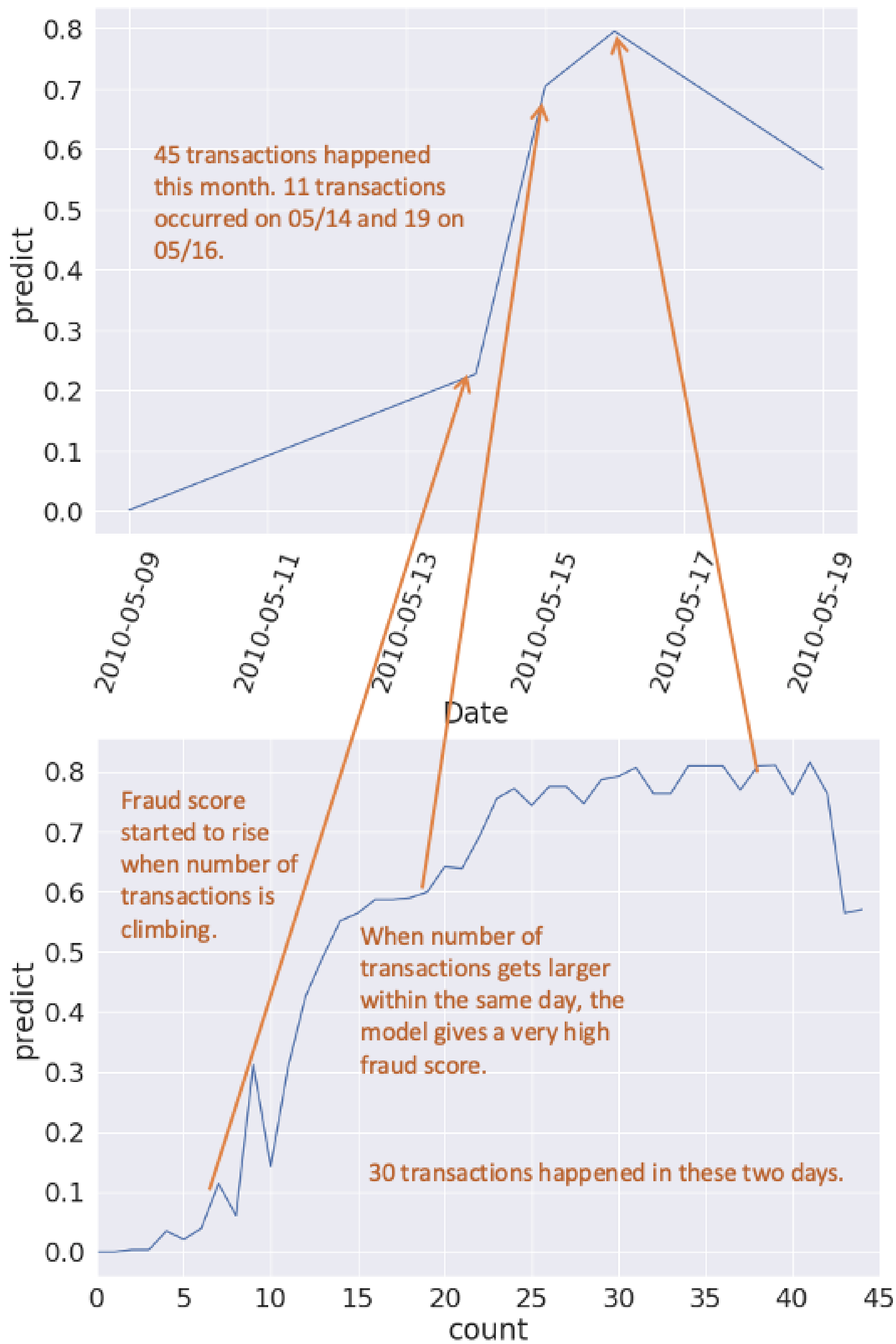
The plot below is a Fraud Savings Financial plot. This is how financial managers decide where they want to set the cut off. The green curve is the difference between the fraud savings and lost sales, which is the expected annual savings. We assume a $2000 gain for every fraud that is caught, and a $50 loss for every false positive. In this case, Our expected annual savings using this model is $1,377,450, and we recommend a score cutoff at 3 %.

Fraud scores rise as they see more activity at the entity level. We looked at the fraud score for cardnumber 5142235211 and merchant number 4620009957157 as a function across time and again as a function of number of transactions. We see that as the number of transactions increases, the fraud score increases. For this card number, most transactions in November happened on 11/25 and 11/26 (17 of 34 transactions), which caused the fraud score to rise.



The fraud score also increased rapidly for merchant number 4620009957157 across two days, as the number of transactions started climbing. There were 45 transactions in May for this merchant number, and 11 over 5/14 and 5/16.

Merchnum = 4620009957157



7. Summary and Conclusion

In this project, we created supervised machine learning models to help identify fraudulent credit card transactions. Before building these supervised models, we sought to understand the basic characteristics of the Card Transaction data. We wrote out the definition of each variable and performed Exploratory Data Analysis (EDA) on every variable to create a Data Quality Report (DQR). This served as a check to make sure the data is correct and helped us familiarize ourselves with the data.

After understanding the data, we cleaned the data by removing exclusions and filling in null values that were present in the fields that were used to create our candidate variables. Our candidate variables were created to help us detect credit card fraud, and these variables can be classified into 5 groups: Amount, Frequency, Days-since, Velocity change, and Target encoded variables. After creating 605 variables, our feature selection process allowed us to reduce the number of features to the top 30 based on multivariate importance, which is an average ranking of the KS score and FDR score.

Using our top 30 features, we created a series of models by utilizing different machine learning algorithms and different hyperparameters. Each model was trained and evaluated 10 times, and the final score for each model was the average of the 10 scores. We decided to use a Random Forest with 30 variables, max depth = 30, n_estimator = 150, max_features = 25, min_samples_split = 200, and min_samples_leaf = 20, as this performed the best on the OOT data. The Fraud Detection Rate of our final model was 69.4% at 3% of the population.

These fraud algorithms worked well and helped identify credit card fraud. However, there is room for improvement. A lot of our processes were manual and time consuming, so additional time would have allowed us to try some improvements. We could have performed more research and communicated with more domain experts to come up with finer candidate variables. We could have tested different selection processes for our wrapper, which could ultimately lead to a different set of variables after the feature selection process. Our model building took a long time as we had to manually test different hyper parameters. More combinations of hyperparameters or hyper parameter tuning using cross validation and a grid search could provide a better performing model. However, manually tuning our models taught us the importance of evaluating the model each time and having more control over the number of trials, while also maintaining the size of our testing data set. Lastly, our data only spanned over 12 months, so more data could have possibly provided more accurate and effective model results.

Analysts often jump right into modeling and neglect the design, understanding of the data, and the engineering of the expert variables. The design of an analytics project is the most important part to obtaining the desired outcome. In our case this was detecting and predicting credit card fraud.

Appendix

I) Data Quality Report

1. Data Description

Dataset Name: Card Transactions

Dataset Source: Fraud Analytics by Professor Stephen Coggeshall

Dataset Purpose: Data represent credit card purchases from a US government organization for the purpose of building a supervised fraud model.

Time Period: From January 01, 2010 to December 31, 2010

Number of Fields: 10

Number of Records: 96,753

2. Data Summary

2.1. Numeric Fields:

Field Name	Field Type	# Records	% Populated	Mean	Standard Deviation	Min	Max	# Zeros
Amount	Numeric	96753	100.00%	427.89	10006.14	0.01	3102046	0

2.2. Categorical Fields:

Field Name	Field Type	# Records	% Populated	# Unique	Most Common
Recnum	Categorical	1070994	100.00%	96753	N/A
Cardnum	Categorical	1070994	100.00%	1645	5142148452
Date	Datetime	1070994	100.00%	365	40237
Merchnum	Categorical	1067619	96.65%	13092	930090121224
Merch descri	Categorical	1070994	100.00%	13126	GSA-FSS-ADV
Merch state	Categorical	1069799	98.76%	228	TN
Merch zip	Categorical	1066338	95.19%	4568	38118
Transtype	Categorical	1070994	100.00%	4	P
Fraud	Categorical	1070994	100.00%	2	0

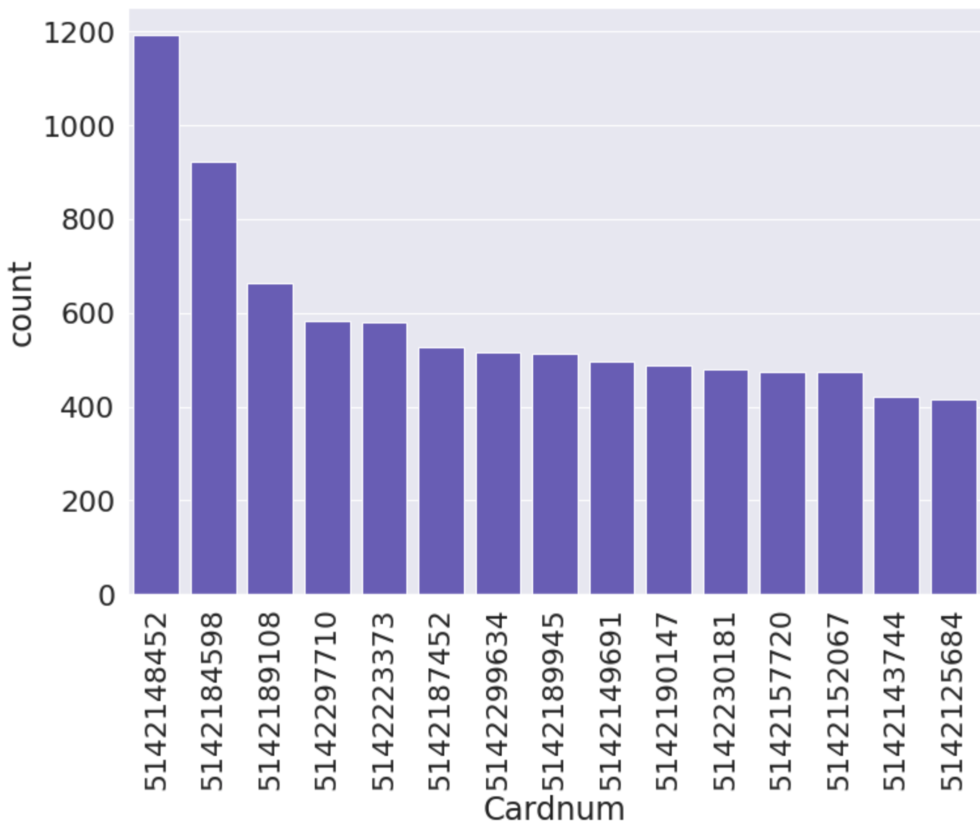
3. Data Field Exploration

3.1. Recnum

The record number is a unique identifier of each entry in the data. It consists of numbers from 1 to 96753.

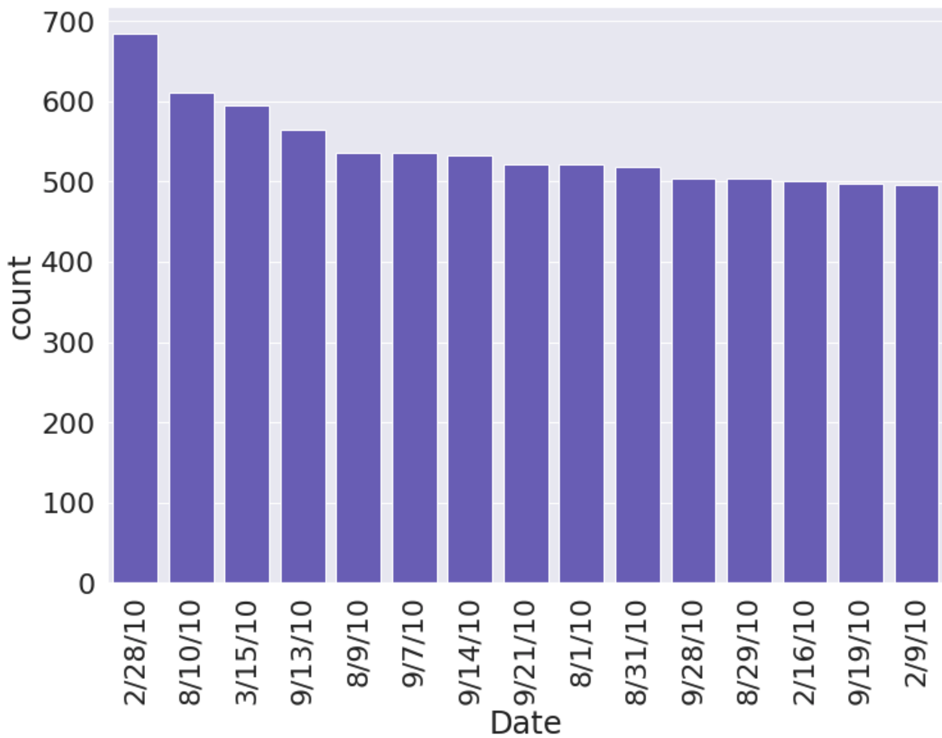
3.2. Cardnum

The credit card number associated with each purchase. It consists of ten-digit numbers. The 15 most common values of the field:



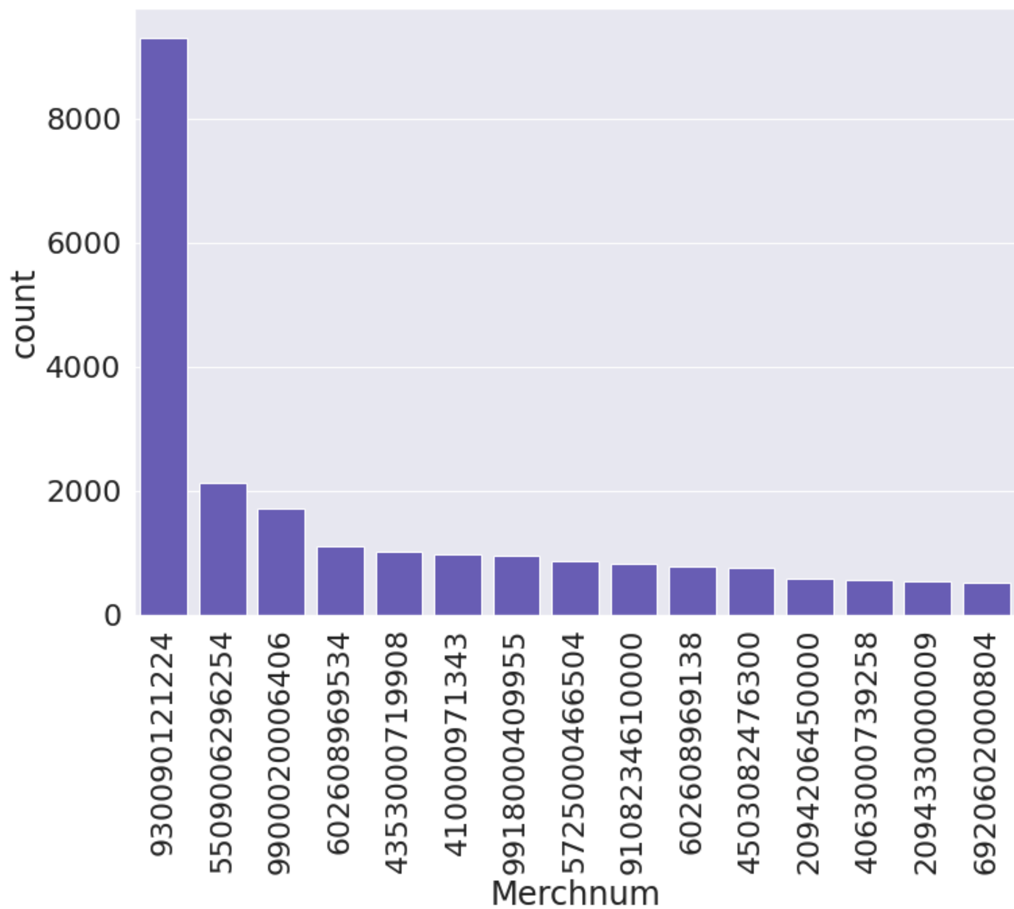
3.3. Date

The date of the transaction occurred. Ranges from 1/1/10 to 12/31/10. The 15 most common values of the field:



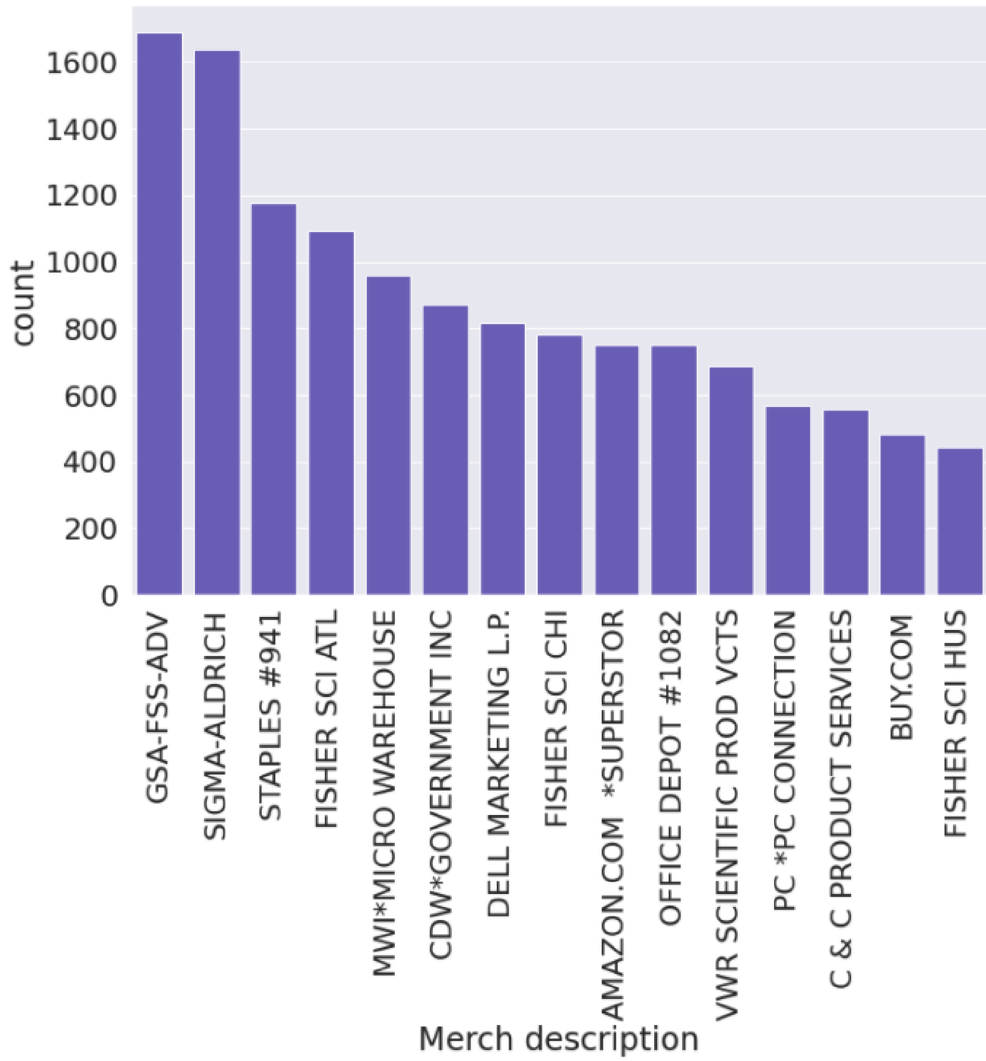
3.4. Merchnum

The merchant numbers. It is a unique identifier of the merchant that processed the transaction. The 15 most common values of the field:



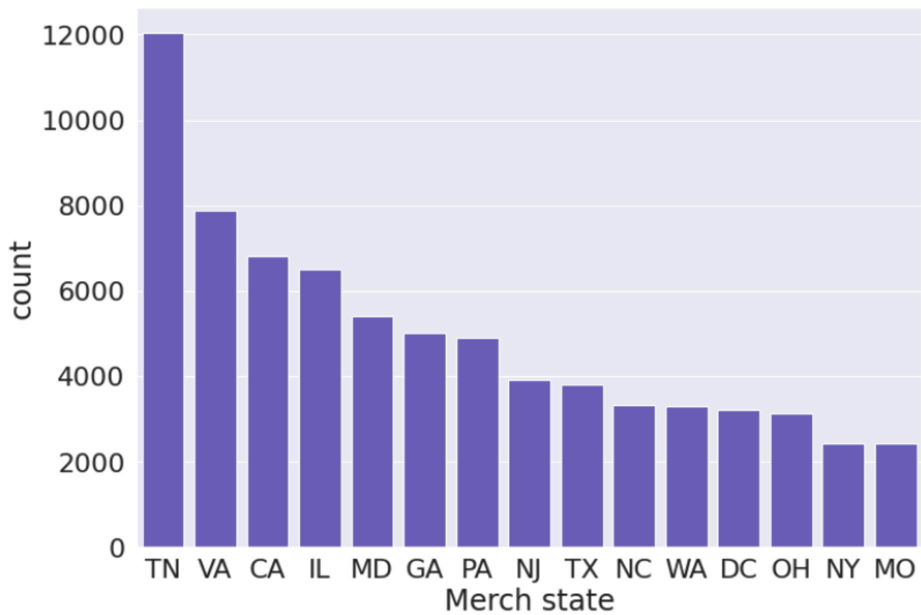
3.5. Merch description

The text description of the merchant that processed the transaction. The 15 most common values of the field:



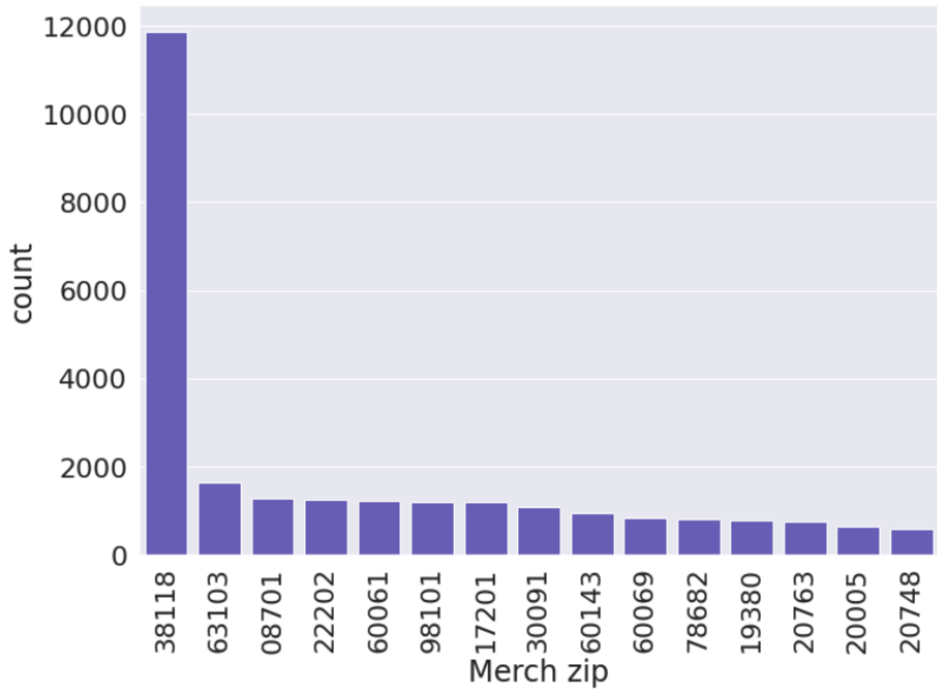
3.6. Merch state

The two-letter state abbreviations of where the merchant is in. The 15 most common values of the field:



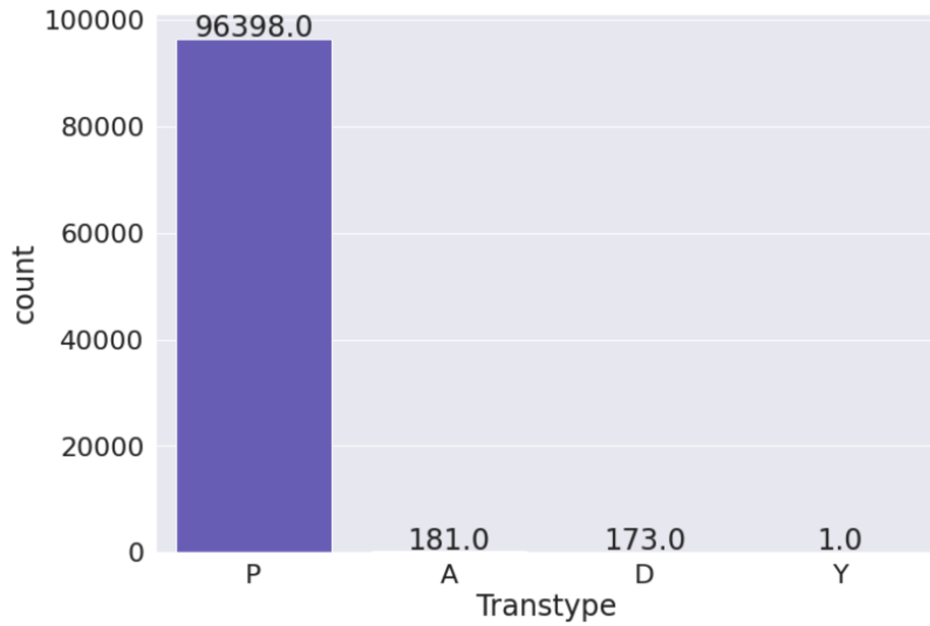
3.7. Merch zip

The five-digit zip code of the Merchant's address. The 15 most common values of the field:



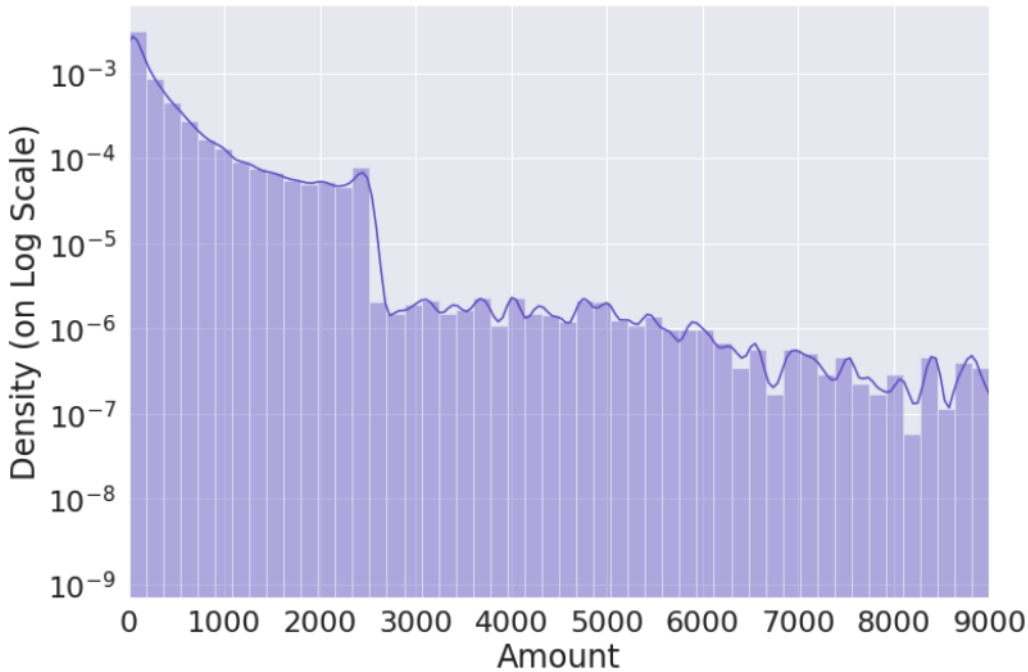
3.8. Transtype

The one-letter code indicating transaction types. The most common value of this field is "P" and it stands for purchase. The count of each transaction types:



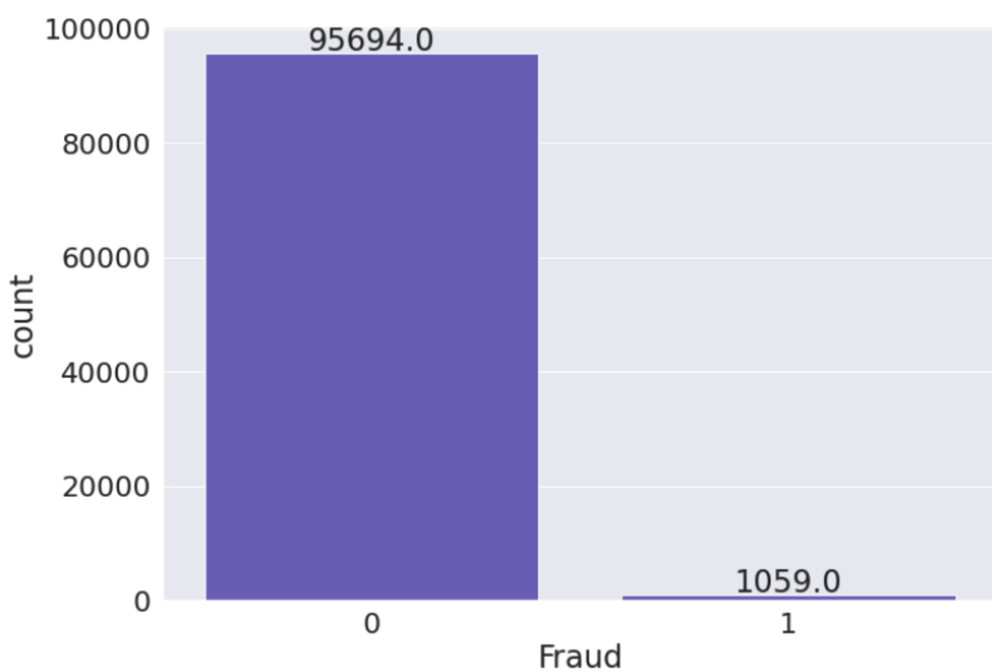
3.9. Amount

The amount of the transaction. The distribution of 99.91% of the data in log scale, after excluding outliers greater than 9000:



3.10. Fraud

The fraud score of the transaction, where 1 indicates a fraud transaction and 0 indicates a normal transaction.



2) List of All Candidate Variables

1	Cardnum_day_since
2	Cardnum_count_0
3	Cardnum_avg_0
4	Cardnum_max_0
5	Cardnum_med_0
6	Cardnum_total_0
7	Cardnum_actual/avg_0
8	Cardnum_actual/max_0
9	Cardnum_actual/med_0
10	Cardnum_actual/toal_0
11	Cardnum_count_1
12	Cardnum_avg_1
13	Cardnum_max_1
14	Cardnum_med_1
15	Cardnum_total_1
16	Cardnum_actual/avg_1
17	Cardnum_actual/max_1
18	Cardnum_actual/med_1
19	Cardnum_actual/toal_1
20	Cardnum_count_3
21	Cardnum_avg_3
22	Cardnum_max_3
23	Cardnum_med_3
24	Cardnum_total_3
25	Cardnum_actual/avg_3
26	Cardnum_actual/max_3
27	Cardnum_actual/med_3
28	Cardnum_actual/toal_3
29	Cardnum_count_7
30	Cardnum_avg_7
31	Cardnum_max_7
32	Cardnum_med_7
33	Cardnum_total_7
34	Cardnum_actual/avg_7
35	Cardnum_actual/max_7

36	Cardnum_actual/med_7
37	Cardnum_actual/toal_7
38	Cardnum_count_14
39	Cardnum_avg_14
40	Cardnum_max_14
41	Cardnum_med_14
42	Cardnum_total_14
43	Cardnum_actual/avg_14
44	Cardnum_actual/max_14
45	Cardnum_actual/med_14
46	Cardnum_actual/toal_14
47	Cardnum_count_30
48	Cardnum_avg_30
49	Cardnum_max_30
50	Cardnum_med_30
51	Cardnum_total_30
52	Cardnum_actual/avg_30
53	Cardnum_actual/max_30
54	Cardnum_actual/med_30
55	Cardnum_actual/toal_30
56	Merchnum_day_since
57	Merchnum_count_0
58	Merchnum_avg_0
59	Merchnum_max_0
60	Merchnum_med_0
61	Merchnum_total_0
62	Merchnum_actual/avg_0
63	Merchnum_actual/max_0
64	Merchnum_actual/med_0
65	Merchnum_actual/toal_0
66	Merchnum_count_1
67	Merchnum_avg_1
68	Merchnum_max_1
69	Merchnum_med_1
70	Merchnum_total_1
71	Merchnum_actual/avg_1
72	Merchnum_actual/max_1
73	Merchnum_actual/med_1
74	Merchnum_actual/toal_1
75	Merchnum_count_3

76	Merchnum_avg_3
77	Merchnum_max_3
78	Merchnum_med_3
79	Merchnum_total_3
80	Merchnum_actual/avg_3
81	Merchnum_actual/max_3
82	Merchnum_actual/med_3
83	Merchnum_actual/toal_3
84	Merchnum_count_7
85	Merchnum_avg_7
86	Merchnum_max_7
87	Merchnum_med_7
88	Merchnum_total_7
89	Merchnum_actual/avg_7
90	Merchnum_actual/max_7
91	Merchnum_actual/med_7
92	Merchnum_actual/toal_7
93	Merchnum_count_14
94	Merchnum_avg_14
95	Merchnum_max_14
96	Merchnum_med_14
97	Merchnum_total_14
98	Merchnum_actual/avg_14
99	Merchnum_actual/max_14
100	Merchnum_actual/med_14
101	Merchnum_actual/toal_14
102	Merchnum_count_30
103	Merchnum_avg_30
104	Merchnum_max_30
105	Merchnum_med_30
106	Merchnum_total_30
107	Merchnum_actual/avg_30
108	Merchnum_actual/max_30
109	Merchnum_actual/med_30
110	Merchnum_actual/toal_30
111	card_merch_day_since
112	card_merch_count_0
113	card_merch_avg_0
114	card_merch_max_0
115	card_merch_med_0

116	card_merch_total_0
117	card_merch_actual/avg_0
118	card_merch_actual/max_0
119	card_merch_actual/med_0
120	card_merch_actual/toal_0
121	card_merch_count_1
122	card_merch_avg_1
123	card_merch_max_1
124	card_merch_med_1
125	card_merch_total_1
126	card_merch_actual/avg_1
127	card_merch_actual/max_1
128	card_merch_actual/med_1
129	card_merch_actual/toal_1
130	card_merch_count_3
131	card_merch_avg_3
132	card_merch_max_3
133	card_merch_med_3
134	card_merch_total_3
135	card_merch_actual/avg_3
136	card_merch_actual/max_3
137	card_merch_actual/med_3
138	card_merch_actual/toal_3
139	card_merch_count_7
140	card_merch_avg_7
141	card_merch_max_7
142	card_merch_med_7
143	card_merch_total_7
144	card_merch_actual/avg_7
145	card_merch_actual/max_7
146	card_merch_actual/med_7
147	card_merch_actual/toal_7
148	card_merch_count_14
149	card_merch_avg_14
150	card_merch_max_14
151	card_merch_med_14
152	card_merch_total_14
153	card_merch_actual/avg_14
154	card_merch_actual/max_14
155	card_merch_actual/med_14

156	card_merch_actual/toal_14
157	card_merch_count_30
158	card_merch_avg_30
159	card_merch_max_30
160	card_merch_med_30
161	card_merch_total_30
162	card_merch_actual/avg_30
163	card_merch_actual/max_30
164	card_merch_actual/med_30
165	card_merch_actual/toal_30
166	card_zip_day_since
167	card_zip_count_0
168	card_zip_avg_0
169	card_zip_max_0
170	card_zip_med_0
171	card_zip_total_0
172	card_zip_actual/avg_0
173	card_zip_actual/max_0
174	card_zip_actual/med_0
175	card_zip_actual/toal_0
176	card_zip_count_1
177	card_zip_avg_1
178	card_zip_max_1
179	card_zip_med_1
180	card_zip_total_1
181	card_zip_actual/avg_1
182	card_zip_actual/max_1
183	card_zip_actual/med_1
184	card_zip_actual/toal_1
185	card_zip_count_3
186	card_zip_avg_3
187	card_zip_max_3
188	card_zip_med_3
189	card_zip_total_3
190	card_zip_actual/avg_3
191	card_zip_actual/max_3
192	card_zip_actual/med_3
193	card_zip_actual/toal_3
194	card_zip_count_7
195	card_zip_avg_7

196	card_zip_max_7
197	card_zip_med_7
198	card_zip_total_7
199	card_zip_actual/avg_7
200	card_zip_actual/max_7
201	card_zip_actual/med_7
202	card_zip_actual/toal_7
203	card_zip_count_14
204	card_zip_avg_14
205	card_zip_max_14
206	card_zip_med_14
207	card_zip_total_14
208	card_zip_actual/avg_14
209	card_zip_actual/max_14
210	card_zip_actual/med_14
211	card_zip_actual/toal_14
212	card_zip_count_30
213	card_zip_avg_30
214	card_zip_max_30
215	card_zip_med_30
216	card_zip_total_30
217	card_zip_actual/avg_30
218	card_zip_actual/max_30
219	card_zip_actual/med_30
220	card_zip_actual/toal_30
221	card_state_day_since
222	card_state_count_0
223	card_state_avg_0
224	card_state_max_0
225	card_state_med_0
226	card_state_total_0
227	card_state_actual/avg_0
228	card_state_actual/max_0
229	card_state_actual/med_0
230	card_state_actual/toal_0
231	card_state_count_1
232	card_state_avg_1
233	card_state_max_1
234	card_state_med_1
235	card_state_total_1

236	card_state_actual/avg_1
237	card_state_actual/max_1
238	card_state_actual/med_1
239	card_state_actual/toal_1
240	card_state_count_3
241	card_state_avg_3
242	card_state_max_3
243	card_state_med_3
244	card_state_total_3
245	card_state_actual/avg_3
246	card_state_actual/max_3
247	card_state_actual/med_3
248	card_state_actual/toal_3
249	card_state_count_7
250	card_state_avg_7
251	card_state_max_7
252	card_state_med_7
253	card_state_total_7
254	card_state_actual/avg_7
255	card_state_actual/max_7
256	card_state_actual/med_7
257	card_state_actual/toal_7
258	card_state_count_14
259	card_state_avg_14
260	card_state_max_14
261	card_state_med_14
262	card_state_total_14
263	card_state_actual/avg_14
264	card_state_actual/max_14
265	card_state_actual/med_14
266	card_state_actual/toal_14
267	card_state_count_30
268	card_state_avg_30
269	card_state_max_30
270	card_state_med_30
271	card_state_total_30
272	card_state_actual/avg_30
273	card_state_actual/max_30
274	card_state_actual/med_30
275	card_state_actual/toal_30

276	merch_zip_day_since
277	merch_zip_count_0
278	merch_zip_avg_0
279	merch_zip_max_0
280	merch_zip_med_0
281	merch_zip_total_0
282	merch_zip_actual/avg_0
283	merch_zip_actual/max_0
284	merch_zip_actual/med_0
285	merch_zip_actual/toal_0
286	merch_zip_count_1
287	merch_zip_avg_1
288	merch_zip_max_1
289	merch_zip_med_1
290	merch_zip_total_1
291	merch_zip_actual/avg_1
292	merch_zip_actual/max_1
293	merch_zip_actual/med_1
294	merch_zip_actual/toal_1
295	merch_zip_count_3
296	merch_zip_avg_3
297	merch_zip_max_3
298	merch_zip_med_3
299	merch_zip_total_3
300	merch_zip_actual/avg_3
301	merch_zip_actual/max_3
302	merch_zip_actual/med_3
303	merch_zip_actual/toal_3
304	merch_zip_count_7
305	merch_zip_avg_7
306	merch_zip_max_7
307	merch_zip_med_7
308	merch_zip_total_7
309	merch_zip_actual/avg_7
310	merch_zip_actual/max_7
311	merch_zip_actual/med_7
312	merch_zip_actual/toal_7
313	merch_zip_count_14
314	merch_zip_avg_14
315	merch_zip_max_14

316	merch_zip_med_14
317	merch_zip_total_14
318	merch_zip_actual/avg_14
319	merch_zip_actual/max_14
320	merch_zip_actual/med_14
321	merch_zip_actual/toal_14
322	merch_zip_count_30
323	merch_zip_avg_30
324	merch_zip_max_30
325	merch_zip_med_30
326	merch_zip_total_30
327	merch_zip_actual/avg_30
328	merch_zip_actual/max_30
329	merch_zip_actual/med_30
330	merch_zip_actual/toal_30
331	merch_state_day_since
332	merch_state_count_0
333	merch_state_avg_0
334	merch_state_max_0
335	merch_state_med_0
336	merch_state_total_0
337	merch_state_actual/avg_0
338	merch_state_actual/max_0
339	merch_state_actual/med_0
340	merch_state_actual/toal_0
341	merch_state_count_1
342	merch_state_avg_1
343	merch_state_max_1
344	merch_state_med_1
345	merch_state_total_1
346	merch_state_actual/avg_1
347	merch_state_actual/max_1
348	merch_state_actual/med_1
349	merch_state_actual/toal_1
350	merch_state_count_3
351	merch_state_avg_3
352	merch_state_max_3
353	merch_state_med_3
354	merch_state_total_3
355	merch_state_actual/avg_3

356	merch_state_actual/max_3
357	merch_state_actual/med_3
358	merch_state_actual/toal_3
359	merch_state_count_7
360	merch_state_avg_7
361	merch_state_max_7
362	merch_state_med_7
363	merch_state_total_7
364	merch_state_actual/avg_7
365	merch_state_actual/max_7
366	merch_state_actual/med_7
367	merch_state_actual/toal_7
368	merch_state_count_14
369	merch_state_avg_14
370	merch_state_max_14
371	merch_state_med_14
372	merch_state_total_14
373	merch_state_actual/avg_14
374	merch_state_actual/max_14
375	merch_state_actual/med_14
376	merch_state_actual/toal_14
377	merch_state_count_30
378	merch_state_avg_30
379	merch_state_max_30
380	merch_state_med_30
381	merch_state_total_30
382	merch_state_actual/avg_30
383	merch_state_actual/max_30
384	merch_state_actual/med_30
385	merch_state_actual/toal_30
386	amount_bin_merch_day_since
387	amount_bin_merch_count_0
388	amount_bin_merch_avg_0
389	amount_bin_merch_max_0
390	amount_bin_merch_med_0
391	amount_bin_merch_total_0
392	amount_bin_merch_actual/avg_0
393	amount_bin_merch_actual/max_0
394	amount_bin_merch_actual/med_0
395	amount_bin_merch_actual/toal_0

396	amount_bin_merch_count_1
397	amount_bin_merch_avg_1
398	amount_bin_merch_max_1
399	amount_bin_merch_med_1
400	amount_bin_merch_total_1
401	amount_bin_merch_actual/avg_1
402	amount_bin_merch_actual/max_1
403	amount_bin_merch_actual/med_1
404	amount_bin_merch_actual/toal_1
405	amount_bin_merch_count_3
406	amount_bin_merch_avg_3
407	amount_bin_merch_max_3
408	amount_bin_merch_med_3
409	amount_bin_merch_total_3
410	amount_bin_merch_actual/avg_3
411	amount_bin_merch_actual/max_3
412	amount_bin_merch_actual/med_3
413	amount_bin_merch_actual/toal_3
414	amount_bin_merch_count_7
415	amount_bin_merch_avg_7
416	amount_bin_merch_max_7
417	amount_bin_merch_med_7
418	amount_bin_merch_total_7
419	amount_bin_merch_actual/avg_7
420	amount_bin_merch_actual/max_7
421	amount_bin_merch_actual/med_7
422	amount_bin_merch_actual/toal_7
423	amount_bin_merch_count_14
424	amount_bin_merch_avg_14
425	amount_bin_merch_max_14
426	amount_bin_merch_med_14
427	amount_bin_merch_total_14
428	amount_bin_merch_actual/avg_14
429	amount_bin_merch_actual/max_14
430	amount_bin_merch_actual/med_14
431	amount_bin_merch_actual/toal_14
432	amount_bin_merch_count_30
433	amount_bin_merch_avg_30
434	amount_bin_merch_max_30
435	amount_bin_merch_med_30

436	amount_bin_merch_total_30
437	amount_bin_merch_actual/avg_30
438	amount_bin_merch_actual/max_30
439	amount_bin_merch_actual/med_30
440	amount_bin_merch_actual/toal_30
441	amount_bin_card_day_since
442	amount_bin_card_count_0
443	amount_bin_card_avg_0
444	amount_bin_card_max_0
445	amount_bin_card_med_0
446	amount_bin_card_total_0
447	amount_bin_card_actual/avg_0
448	amount_bin_card_actual/max_0
449	amount_bin_card_actual/med_0
450	amount_bin_card_actual/toal_0
451	amount_bin_card_count_1
452	amount_bin_card_avg_1
453	amount_bin_card_max_1
454	amount_bin_card_med_1
455	amount_bin_card_total_1
456	amount_bin_card_actual/avg_1
457	amount_bin_card_actual/max_1
458	amount_bin_card_actual/med_1
459	amount_bin_card_actual/toal_1
460	amount_bin_card_count_3
461	amount_bin_card_avg_3
462	amount_bin_card_max_3
463	amount_bin_card_med_3
464	amount_bin_card_total_3
465	amount_bin_card_actual/avg_3
466	amount_bin_card_actual/max_3
467	amount_bin_card_actual/med_3
468	amount_bin_card_actual/toal_3
469	amount_bin_card_count_7
470	amount_bin_card_avg_7
471	amount_bin_card_max_7
472	amount_bin_card_med_7
473	amount_bin_card_total_7
474	amount_bin_card_actual/avg_7
475	amount_bin_card_actual/max_7

476	amount_bin_card_actual/med_7
477	amount_bin_card_actual/toal_7
478	amount_bin_card_count_14
479	amount_bin_card_avg_14
480	amount_bin_card_max_14
481	amount_bin_card_med_14
482	amount_bin_card_total_14
483	amount_bin_card_actual/avg_14
484	amount_bin_card_actual/max_14
485	amount_bin_card_actual/med_14
486	amount_bin_card_actual/toal_14
487	amount_bin_card_count_30
488	amount_bin_card_avg_30
489	amount_bin_card_max_30
490	amount_bin_card_med_30
491	amount_bin_card_total_30
492	amount_bin_card_actual/avg_30
493	amount_bin_card_actual/max_30
494	amount_bin_card_actual/med_30
495	amount_bin_card_actual/toal_30
496	Cardnum_count_0_by_7
497	Cardnum_amount_0_by_7
498	Cardnum_count_0_by_14
499	Cardnum_amount_0_by_14
500	Cardnum_count_0_by_30
501	Cardnum_amount_0_by_30
502	Cardnum_count_1_by_7
503	Cardnum_amount_1_by_7
504	Cardnum_count_1_by_14
505	Cardnum_amount_1_by_14
506	Cardnum_count_1_by_30
507	Cardnum_amount_1_by_30
508	Merchnum_count_0_by_7
509	Merchnum_amount_0_by_7
510	Merchnum_count_0_by_14
511	Merchnum_amount_0_by_14
512	Merchnum_count_0_by_30
513	Merchnum_amount_0_by_30
514	Merchnum_count_1_by_7
515	Merchnum_amount_1_by_7

516	Merchnum_count_1_by_14
517	Merchnum_amount_1_by_14
518	Merchnum_count_1_by_30
519	Merchnum_amount_1_by_30
520	card_merch_count_0_by_7
521	card_merch_amount_0_by_7
522	card_merch_count_0_by_14
523	card_merch_amount_0_by_14
524	card_merch_count_0_by_30
525	card_merch_amount_0_by_30
526	card_merch_count_1_by_7
527	card_merch_amount_1_by_7
528	card_merch_count_1_by_14
529	card_merch_amount_1_by_14
530	card_merch_count_1_by_30
531	card_merch_amount_1_by_30
532	card_zip_count_0_by_7
533	card_zip_amount_0_by_7
534	card_zip_count_0_by_14
535	card_zip_amount_0_by_14
536	card_zip_count_0_by_30
537	card_zip_amount_0_by_30
538	card_zip_count_1_by_7
539	card_zip_amount_1_by_7
540	card_zip_count_1_by_14
541	card_zip_amount_1_by_14
542	card_zip_count_1_by_30
543	card_zip_amount_1_by_30
544	card_state_count_0_by_7
545	card_state_amount_0_by_7
546	card_state_count_0_by_14
547	card_state_amount_0_by_14
548	card_state_count_0_by_30
549	card_state_amount_0_by_30
550	card_state_count_1_by_7
551	card_state_amount_1_by_7
552	card_state_count_1_by_14
553	card_state_amount_1_by_14
554	card_state_count_1_by_30
555	card_state_amount_1_by_30

556	merch_zip_count_0_by_7
557	merch_zip_amount_0_by_7
558	merch_zip_count_0_by_14
559	merch_zip_amount_0_by_14
560	merch_zip_count_0_by_30
561	merch_zip_amount_0_by_30
562	merch_zip_count_1_by_7
563	merch_zip_amount_1_by_7
564	merch_zip_count_1_by_14
565	merch_zip_amount_1_by_14
566	merch_zip_count_1_by_30
567	merch_zip_amount_1_by_30
568	merch_state_count_0_by_7
569	merch_state_amount_0_by_7
570	merch_state_count_0_by_14
571	merch_state_amount_0_by_14
572	merch_state_count_0_by_30
573	merch_state_amount_0_by_30
574	merch_state_count_1_by_7
575	merch_state_amount_1_by_7
576	merch_state_count_1_by_14
577	merch_state_amount_1_by_14
578	merch_state_count_1_by_30
579	merch_state_amount_1_by_30
580	amount_bin_merch_count_0_by_7
581	amount_bin_merch_amount_0_by_7
582	amount_bin_merch_count_0_by_14
583	amount_bin_merch_amount_0_by_14
584	amount_bin_merch_count_0_by_30
585	amount_bin_merch_amount_0_by_30
586	amount_bin_merch_count_1_by_7
587	amount_bin_merch_amount_1_by_7
588	amount_bin_merch_count_1_by_14
589	amount_bin_merch_amount_1_by_14
590	amount_bin_merch_count_1_by_30
591	amount_bin_merch_amount_1_by_30
592	amount_bin_card_count_0_by_7
593	amount_bin_card_amount_0_by_7
594	amount_bin_card_count_0_by_14
595	amount_bin_card_amount_0_by_14

596	amount_bin_card_count_0_by_30
597	amount_bin_card_amount_0_by_30
598	amount_bin_card_count_1_by_7
599	amount_bin_card_amount_1_by_7
600	amount_bin_card_count_1_by_14
601	amount_bin_card_amount_1_by_14
602	amount_bin_card_count_1_by_30
603	amount_bin_card_amount_1_by_30
604	state_risk
605	weekday_risk

3) Smoothing Formula

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