USC Marshall
School of Business

DSO 562 Fraud Analytics Project 3
Credit Card Transaction Fraud

Team 3
Session: 16237

Boyang Han, Leon Man, Ziqing (Juno) Wen, Dakota Wu, Chutong Yan, Yangzi Zhang
May 3, 2020

Table of Contents

Executive Summary

This project is commissioned to develop a supervised fraud detection model based on data released by a federal government organization in Tennessee. The dataset covers the time period from January 1st, 2010 to December 31st, 2010, containing 10 fields and 96,753 records. Nine out of the 10 fields are real transaction data including card number, date, type and amount of the transaction, merchant number, merchant description, merchant state and merchant zip code. The last field is a binary field that indicates if the transaction is fraudulent and was artificially created to mimic the real-world signals of fraud as well as noises (e.g. unrecognized fraud or mislabeled good transactions) by Professor Coggeshall.

Two main types of fraudulent transactions exist in this dataset: 1) fraud induced by card users, and 2) fraud induced by merchants. The model learned to detect fraud by capturing common characteristics of card transaction fraud such as abnormally high transaction amount, the sudden increase of transaction amount compared to historical behavior, and the frequent occurrence of the same card or merchant. Detection Rate (FDR) at 3% rejection rate was used as the evaluation metric for hyperparameter tuning and model selection.

To get a better idea of how well the model predicts on previously unseen data, the whole dataset was divided into three parts: training, testing, and out-of-time (OOT) validation sets

- **Training data:** randomly selected 70% of the data from 1/15/2010 to 10/31/2010
- **Testing data:** randomly selected 30% of the data from 1/15/2010 to 10/31/2010
- **OOT data:** the last two months of data

Training and testing sets combined are referred to as the modeling data. Note that the first two weeks of records were included in variable creation but excluded from modeling to avoid introducing bias caused by variable values calculated from incomplete data. Moreover, the training and testing sets are dynamic in the sense that the modeling data was split every time before fitting a model and model performance was always measured by the average score across 10 independent splits.

A model would be fit on the training data and then used to predict on the testing and OOT data. The test performance provides an estimate of how well the model predicts on new data drawn from the same time period where the training data was drawn from, while the performance on the OOT data provides an estimate of model performance on new data drawn from a future time period occurred sometime after the one where the training data was drawn from.

The project was carried out following the steps below:

- **Data Exploration:** understand the characteristics of the data and recognize the necessary data transformation and manipulation to be performed
- **Data Cleaning:** fill missing values, remove outliers and transactions of undesired types
- **Variables Creation:** build 314 expert variables that quantify typical fraud behaviors
- **Variables Standardization**: bring all variables to the same scale to be comparable
- **Feature Selection:** use filter and wrapper methods to pick the top 30 variables with the best predictive potential to reduce dimensionality and speed up computation
- **Model Fitting:** try different algorithms such as Logistic Regression, Gradient Boosting Tree, Random Forest, Neural Network, Single Tree, and K-Nearest Neighbors with different hyperparameter combinations and compare model performance
- **Model Selection:** select the best model based on the performance on the testing data
- **Result Analysis & Final Decision**: select the best score cutoff point to reject transactions of high fraud risk

The final model is a Gradient Boosting Tree classifier (*learning_rate* = 0.1, *n_estimators* = 1000, *max_depth* = 3, *max_features* = 5, *min_samples_leaf* = 30, *min_samples_split* = 100, *subsample* = 1) that achieved an average FDR of 100.00% at a 3% rejection threshold on the training data, 94.85% on the testing data, and 55.53% on the OOT data. Based on the cost and saving assumptions, it is recommended to reject the top 3% of transactions with the highest fraud score outputted by the model. This policy balances the tradeoff on savings generated from catching a fraud and the loss caused by mistaking a regular transaction as fraudulent, leading to an optimal saving of $194,450 in the last two months of 2010 ($15,647 for every 1,000 transactions).

1 Data Exploration

There are nine categorical fields and one numerical field in the provided dataset. Except *Merchnum*, *Merch state*, and *Merch zip*, all other fields are fully populated. A total of 96,594 (98.91%) good records and 1,059 (1.09%) fraud records exist in the dataset. Key statistics of these fields are summarized in Table 1.1 and 1.2 below.

Table 1.1: Summary Statistics of the Categorical Fields

Field Name	# Records	% Populated	# Unique Values	Most Common Field Value
Recnum	96753	100.00%	96753	NA
Cardnum	96753	100.00%	1645	5142148452
Date	96753	100.00%	365	2010-02-28
Merchnum	93378	96.51%	13091	930090121224
Merch description	96753	100.00%	13126	GSA-FSS-ADV
Merch state	95558	98.76%	227	TN
Merch zip	92097	95.19%	4567	38118
transtype	96753	100.00%	4	P
Fraud	96753	100.00%	2	0

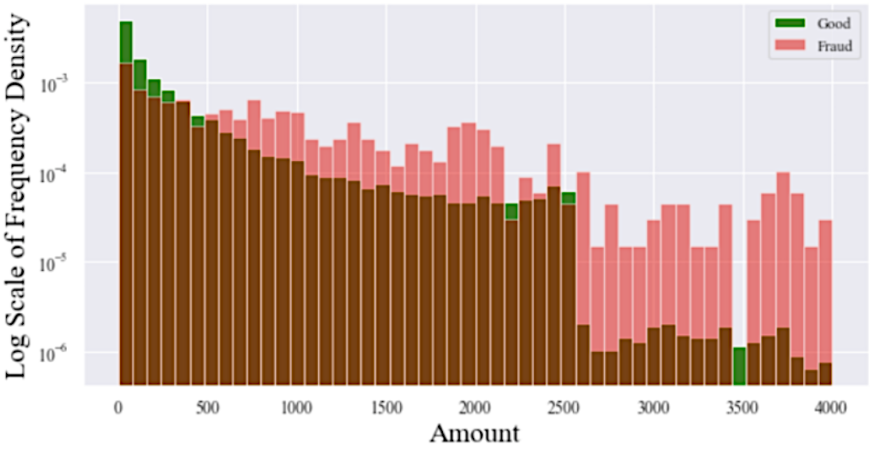
Table 1.2: Summary Statistics of the Numerical Field

Field Name	# Records	% Populated	# Unique Values	Mean	Standard Deviation	Min	Max
Amount	96753	100%	34909	427.88	10006.14	0.01	3102045.5

The team found 3,375 (3.49%) records with missing values in the *Merchnum* field, 1,195 (1.24%) in the *Merch state* field, and 4,656 (4.81%) in the *Merch zip* field. Another 231 (0.2%) records have value “0” in the *Merchnum* field, which are likely placeholders for missing values.

Three things are worth noticing in regard to the *Amount* field. Firstly, as shown in Figure 1.3, while the distribution of transaction amounts of good records are highly right skewed (most records fall in the lower range), which is common in business and economics, that of fraud records are much more balanced with only a moderate decrease of frequency as the amount value increases. This indicates that transactions with higher amounts are more likely to be fraudulent.

Figure 1.3: Distribution of Amount (Good vs. Fraud) Excluding Values over 4000



Secondly, record 52715 has an extremely large transaction amount of $3,102,046. Given that the median amount is only around $138, this record is very likely a mis-entry. Lastly, the most common type of transactions in the dataset is for FedEx shipping, which consists of over 1% of the total records. These transactions usually have a small transaction amount between three to five dollars, with the majority falling between three to four dollars.

2 Data Cleaning

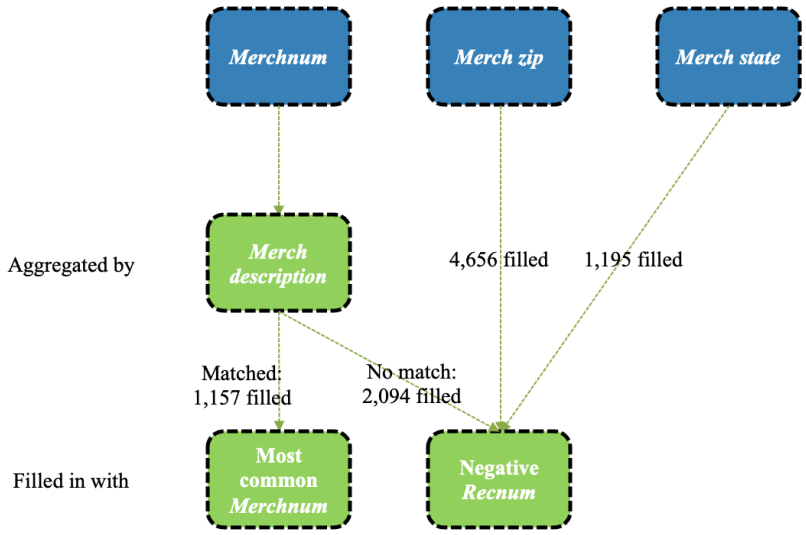
For the data cleaning process, missing values in three columns as mentioned above were filled in. After that, one outlier and all records not of the transaction type "P" were excluded from the dataset.

2.1 Impute Missing Values

Three columns, namely *Merchnum*, *Merch zip*, and *Merch state*, have missing values that need to be filled. It is common practice to use information from other fields and records to try to make a reasonable guess of what the missing value can possibly be. However, in the case of building a fraud detection algorithm, this practice is dangerous because the core idea of the algorithm is to create linkage between records based on common field values and the imputation manually creates commonality. To avoid creating unfounded linkage between records, the team intentionally avoided filling missing fields with existing values in the column as much as possible and only did it when the imputation logic is solid.

Negative *Recnum* was used to fill in the missing values in the *Merch zip and Merch state* fields. When it comes to the *Merchnum* field, missing values were first filled in with the most frequently appeared *Merchnum* that corresponds to the group of records having the same *Merch description*. If no *Merchnum* was matched, negative *Recnum* was used instead. This process was visually illustrated in Figure 2.1.1.

Figure 2.1.1: Flowchart of Filling in Missing Values



2.2 Remove Outlier

Record 52715, the transaction with an amount of over $3 million, was excluded from the dataset to avert any bias caused by its extreme value.

2.3 Remove Non-P Transactions

As instructed by the owner of the data, only the transactions with value "P" in the *transtype* field are successful transactions and they should be the sole concern of this project. Therefore, the 355 non-P transactions were removed before variable creation.

3 Variable Creation

A total of 314 candidate variables were created to quantify the characteristics of fraud behaviors. Table 3.0.1 below summarizes the description of each variable category and the number of variables created under the category. Please also see Appendix B for the full list of variables. Two entities, *Cardnum* and *Merchnum*, together with three entity combinations, *Cardnum* at a certain *Merchnum*, *Cardnum* in a certain zip code, and *Cardnum* in a certain state, were used to create linkage between records and quantify fraudulent behaviors.

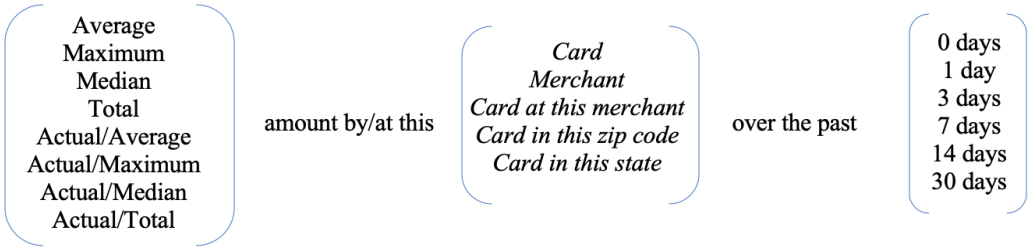
Table 3.1: Variable Creation Summary

Category	Description	# Variables Created
Amount	Absolute or relative transaction amount of the same entity or entity combination within a certain time frame before the record	240
Velocity	Occurrence of the same entity or entity combination within a certain time frame before the record	30
Days-since	Number of days since the last appearance of the same entity or entity combination	5
Velocity Change	Transaction amount or frequency of the same entity in the short term relatively to the long term	24
Benford's Law	Distribution of the first digits of transaction amounts of the same entity	14
Day-of-Week	Average likelihood of fraud on a given day of the week	1

3.1 Amount Variables

As mentioned in the data exploration section, a transaction with a higher amount is more likely to be fraud. Therefore, the amount variables were created to capture the absolute and relative dollar amount of credit card transactions by the same entity within a certain time frame before a given transaction. The time windows used were 0, 1, 3, 7, 14, and 30 days, with 0 day meaning the same day up until the time of the transaction. When creating these variables, the average, maximum, median, and total transaction amount were calculated. Moreover, the actual amount of a given transaction divided by the above four aggregated values were also calculated. As shown in Figure 3.1.1 below, a total of 240 variables were created. The greater the amount variables, the more likely a record is a fraud.

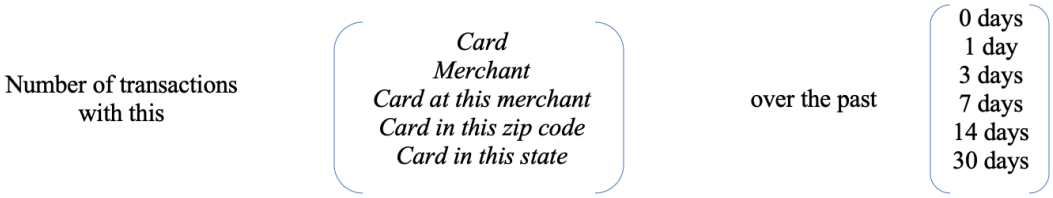
Figure 3.1.1: Amount Variables Creation



3.2 Velocity Variables

A common behavior of fraudsters is that they tend to commit a large amount of fraud within a short period of time, therefore the velocity variable is a useful indicator of fraudulent behaviors. Velocity variables represent the occurrence of the same entity or entity combinations within a certain time frame before a given transaction. The same time windows as used to create amount variables were used. As shown in Figure 3.2.1, 30 variables were created. For velocity variables, a high value indicates fraud.

Figure 3.2.1: Velocity Variables Creation

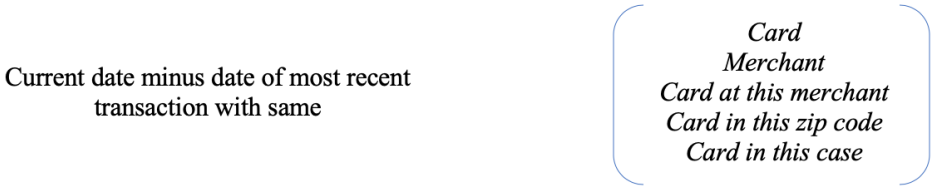


3.3 Days-since Variables

Another way to capture the frequency of appearance of the same entity is to count the number of days since its last appearance. Five days-since variables were created as shown in Figure 3.3.1. A smaller value means a closer last appearance of the same entity, and therefore a higher likelihood of fraud. If an entity has never occurred in the dataset before, its corresponding days-since variable would take the number of days since the earliest date in the dataset, 2010/1/1, as its value.

According to this policy, records on later dates would naturally have higher values in the days-since variables, which would introduce a systematic bias to the data and result in different distributions of the modeling and the validation data that could compromise model performance. Two measures were taken to mitigate such bias: removing the first two weeks of data and capping variables at ±10 after z-scaling. These processes will be discussed in detail in section 5.2 Preparation for Feature Selection.

Figure 3.3.1: Days-since Variables Creation



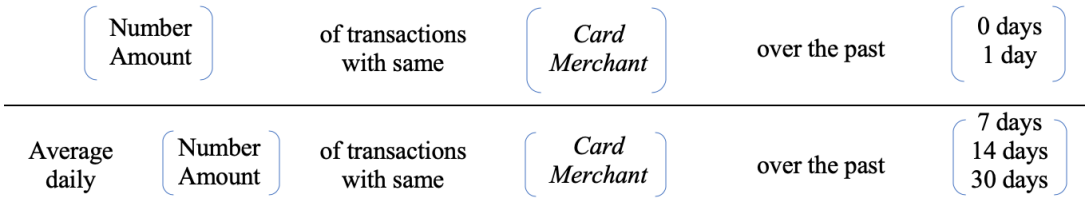
3.4 Velocity Change Variables

In addition to the velocity variables, velocity change variables were created to help capture the sudden increase of credit card transactions or transaction amounts by the same entity in the short term relative to the long term.

For number of transactions, the rationale is that given the same number of occurrences of the same entity in a set time period (i.e. a 30-day window), a situation where all the occurrences happened on the same day is more likely to be a fraud than another situation where the occurrences were spread out on different days. For transaction amounts, the rationale is that fraud is likely to occur when the total dollar amount of recent transactions is high, compared to the average daily amount over a longer time window in the past.

For entities under concern, velocity change variables were calculated as the number of the transactions or the total amount of transactions by that entity in the recent past, namely the past 0 or 1 day, divided by its average daily appearance or average daily total amount in the longer term, as shown in Figure 3.4.1. The longer-term is defined as a time frame of 7, 14 or 30 days. A total of 24 variables were created.

Figure 3.4.1: Velocity Change Variables Creation



3.5 Benford's Law Variables

Benford's Law refers to a counter-intuitive observation that the first non-zero digits of many real life numerical measures are often not uniformly distributed. Smaller numbers tend to appear more frequently than greater numbers. Table 3.5.1 shows the common distribution of the first non-zero digits.

Table 3.5.1: The Distribution of First Non-Zero Digits According to Benford's Law

Digit	Probability
1	0.301
2	0.176
3	0.125
4	0.0969
5	0.0792
6	0.0669
7	0.0580
8	0.0512
9	0.0458

When a fraudster is making up a large amount of numbers, Benford's Law is useful to detect it because a fraudster probably does not know about Benford's Law and thus the numbers he/she creates will not follow the distribution described by Benford's Law. In the case of card transaction fraud, the first digits of all the transaction amounts related to either a particular credit card number or a particular merchant number will not follow Benford's Law.

The following example uses *Cardnum* and a 30-day time frame to demonstrate how Benford's Law variables were created.

1. For each record, find all the transactions with the same *Cardnum* within the past 30 days.
2. For transactions found in step 1, extract the first non-zero digits of their dollar amount and calculate the following values:

Value	Explanation
n_low	# of first digits that are 1 and 2
n_high	# of first digits that are 3 - 9
n	n_low + n_high

3. According to Benford's Law, the ratio of *n_high* to *n_low* should be 1.096. Therefore, *R*, as shown in the formula below, should be close to 1 if the numbers do not violate Benford's Law. *R* being either too larger than or too smaller than 1 is an indication of suspicious activities. Therefore, *U = max(R, 1/R)* could be used to measure the unusualness of a group of transactions. If either *n_low* or *n_high* is zero, it will be set to 1 to avoid having zero as the denominator.

$$R = 1.096 * \frac{n\_low}{n\_high}$$

4. In the case where a specific *Cardnum* had very few transactions within a certain time window in the past, there were too few data points to form a convincing distribution and thus the value of *U* cannot accurately measure the unusualness. In order to solve the problem, a smoothing function was used to calculate the final variable value *U**:

$$U^* = 1 + \frac{U-1}{1+exp^{-t}} \; where \; t = \frac{n - n_{mid}}{c} \; (n_{mid} = 15, c = 3)$$

During the data exploration process, it was discovered that more than 1% of the transactions were for FedEx shipping and most of their transaction amounts started with "3", which is a natural violation of Benford's Law and will trigger false alarms. Therefore, those transactions were excluded when creating Benford's Law variables. If a given record itself is a FedEx shipping transaction and its corresponding *Cardnum* or *Merchnum* does not have any non-FedEx transaction within the given time frame in the past, "1" is assigned to the Benford's Law variable for that record. A total of 14 variables were created, as shown in Figure 3.5.2 below. A higher value indicates a more serious violation of Benford's Law and thus is more likely to be fraud.

Figure 3.5.2: Benford's Law Variables Creation

3.6 Day-of-Week Variables

Considering the possibility that the likelihood of someone committing fraud may vary depending on the day of the week, a categorical variable that indicates the weekday on which the application happened was added. Each category in the variable (i.e. Monday) was then replaced with the average probability of fraud in this category.

When calculating the average probabilities, the last two months of data (OOT data) were excluded to avoid overfitting. The average probabilities of fraud by weekday are summarized in Table 3.6.1 below. For example, all records that happened on a Monday will have a value of 0.008711 in this variable.

Table 3.6.1: Average Probability of Fraud by the Day of Week

Weekday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday
Probability of Fraud	0.008711	0.007127	0.009788	0.018626	0.025994	0.010095	0.009630

4 Feature Selection

Feature selection is the process of selecting the features that contribute the most to predicting the output of interest. Section 4.1 below explains why it was performed, section 4.2 lists the data preparation steps, and sections 4.3 and 4.4 elaborate in detail how it was performed. Univariate filter and backward-elimination wrapper methods were used in this project. Out of 314 candidate variables, 30 were selected for modeling.

4.1 Why is Feature Selection Necessary

When analyzing data in high-dimensional space, various phenomena make machine learning challenging. Typically, such phenomena is referred to as the curse of dimensionality. As dimensionality increases, data quickly becomes sparse and all objects appear to be dissimilar, which makes it harder for a model to learn patterns. Moreover, with higher dimensionality comes more noise, and thus more data is needed to tell the true pattern from noise. And this need for data grows exponentially as the number of features grows. On the other hand, a large number of features usually mean more irrelevant or correlated features, which could result in a dumb model with less accurate predictions. Such redundant features should be removed for both performance and efficiency concerns.

In short, having high dimensionality of excessive features in the data can decrease model performance and increase computation time, hence feature selection is a necessary step before modeling.

4.2 Preparation for Feature Selection

In order to perform feature selection, the following steps were taken to prepare the data:

1. **Remove the first two weeks:** Most variables were built by looking 1 to 30 days back into the past. Unfortunately, the first month of records do not have the luxury of a full 30 days of history to look at, thus their variables are biased. This systematic bias is inevitable when given any finite amount of data. The team removed the very first two weeks of the records, which were the most biased, and kept the second half of the first month as a compromise to include more data points for modeling.

2. **Separate the dataset into training, testing and out-of-time data (referred to as OOT):** The separation of the dataset makes it possible to evaluate model performance on new data drawn from the same or different time period. In this case, feature selection was only performed on the modeling data (training, testing) to avoid overfitting. Table 3.2.1 below summarizes how the data was split and the basic statistics of the resulting sets.

Table 4.2.1: Modeling and out-of-time data and basic statistics

Data		Criteria	# Records	# Fraud	% Fraud
Modeling	Training	randomly picked 80% of data between 2010/1/15 and 2010/10/31	64,505	697	1.08%
	Testing	randomly picked 20% of data between 2010/1/15 and 2010/10/31	16,127	171	1.06%
Out of Time/Validation		data between 2010/10/1 and 2010/12/31	12,427	175	1.41%
Total			93,059	1,043	1.12%

3. **Z-scale:** Z-scaling all variables brings them to the same scale and treats them equally so that any distance based algorithm would not be misled. When performing the standardization, scaler was fit on the modeling data and transformed on the modeling and OOT data. Fitting only on the modeling data leads to unbiased model performance on OOT data since the distribution of the OOT data was not exposed to the model.

4. **Capping:** All z-scaled variables were capped at 10. Values more than 10 standard deviations away from the mean were replaced with ±10 in order to focus on the range that matters the most. The capping process was applied to both modeling and OOT data.

5. **Z-scale again:** As suggested by experts in the field, the team z-scaled the dataset again after capping using the same technique described in step three.

4.3 Filter

A filter method is a fast feature selection method that examines how good each independent variable by itself predicts the target. It is independent of any modeling method. Common filter methods for binary classification problems include Pearson Correlation, Fisher Score, Mutual Information, univariate Kolmogorov-Smirnov (KS) Score and Fraud Detection Rate (FDR).

For the purpose of this project, the univariate KS Score and FDR were used to rank the 314 candidate variables respectively. The average ranking by two scores was then used to filter out the top 100 variables to be moved on to the next step of feature selection.

4.3.1 Univariate Kolmogorov-Smirnov (KS)

The importance of a feature in supervised fraud detection problems can be measured by how well it separates the bad records from the good ones. Kolmogorov-Smirnov score represents the maximum of the difference between the cumulative goods and bads. The higher the KS score, the more important a feature is in predicting fraud. The mathematical expression of KS is shown in the formula below:

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

4.3.2 Fraud Detection Rate (FDR)

Another way to determine the importance of a feature in fraud detection problems is to measure how well it catches frauds at a given threshold. FDR is such a robust statistic that calculates the percentage of total frauds caught at a given rejection cut-off. The higher the FDR is, the better the variable is as a predictor for fraud.

For each feature X, FDR at 3% is computed as the percentage of total frauds caught in the top 3% of the population sorted by X, with the top rows being most likely to be fraudulent and the bottom rows the opposite. The calculation of FDR is given in the formula below:

$$\text{FDR} = \frac{\#\ frauds\ caught\ at\ 3\%\ rejection\ rate}{total\ \#\ frauds\ in\ the\ dataset}$$

4.3.3 Combine KS and FDR outputs

To balance the weakness of individual filters, the outputs of KS and FDR were combined to determine the selection of variables to move on with. Records were first ranked by the KS and FDR scores respectively and then the average ranking of the two scores was used as the final filter score.

$$\text{Average Scores} = \frac{Rank\ based\ on\ KS\ (Descending) + Rank\ based\ on\ FDR\ (Descending)}{2}$$

As the desired number of input variables for modeling is between 20 and 30, the top 100 candidate variables with the highest final filter scores were selected to be passed on to the next stage of feature selection. A full list of the top 100 variables, together with their scores and rankings, can be found in Appendix C.

4.4 Wrapper

4.4.1 What is a Wrapper

A wrapper is a stepwise feature selection method that evaluates the features of a specific machine learning algorithm to find an optimal feature combination. It follows a greedy search approach by evaluating many possible combinations of features against the evaluation criterion. In this project, Random Forest was chosen as the machine learning algorithm to be wrapped around.

In addition, since the fraud dataset is very imbalanced and accuracy can be misleading in this case, ROC-AUC (Area Under the Receiver Operating Characteristic Curve) was used as the evaluation criterion. AUC is very sensitive to class imbalance. When there is a minority class in

the data, the fraud class in the case of this project, the AUC score will be strongly impacted by the minority class, meaning that it is less likely that the fraud records will be misclassified to maximize the correct classification of the majority class and to minimize overall misclassification error. Hence the results better serve the purpose of fraud detection.

There are three commonly used wrapper techniques: forward selection, backward selection, and general stepwise selection. For this project, a backward selection wrapper was adopted. The backward selection method starts with a full model, which includes all the features, and removes one feature at a time. A feature is chosen to be removed if its removal results in the highest improvement or the least damage to the classifier performance (i.e. AUC score). Moreover, cross-validation was incorporated to ensure more reliable results. This process repeats until the model degradation, resulted by variable elimination, is below an acceptable amount and then the optimal feature subset is found.

4.4.2 How does the wrapper work

As commonly suggested in the fraud industry, a reasonable number of predictors should be kept for better dimensionality reduction. Thus all the 100 features selected from the filter method were examined through the wrapper to narrow down the top 20-30 best features. The wrapper was iterated 30 times to mitigate the variance caused by the randomness in cross validation and the Random Forest algorithm. Then, the variables selected in all iterations were ranked based on the average rank of importance. Finally, the team kept the top 30 variables with the highest average rankings (as shown in Table 4.4.2.1).

Table 4.4.2.1: Final variables with ranking

30 Iterations			
Variable	**Average_Rank**	**Final Rank**	**Description**
card_merch_lag1_tot	9.87	1	Total transaction amout using the given *card* at the given *merchant* over the past 1 day
card_lag1_tot	10.20	2	Total transaction amout using the given *card* over the past 1 day
card_lag30_max	10.33	3	Maximum transaction amout using the given *card* over the past 30 days
card_zip_lag14_tot	10.73	4	Total transaction amout using the given *card* at the given *zip* over the past 14 days
merch_lag1_tot	10.73	5	Total transaction amout at the given *merchant* over the past 1 day
card_state_lag30_max	10.83	6	Maximum transaction amout using the given *card* at the given *state* over the past 30 days
card_state_lag3_tot	11.47	7	Total transaction amout using the given *card* at the given *state* over the past 3 days
card_lag0_tot	12.23	8	Total transaction amout using the given *card* over the past 0 day
merch_lag14_max	13.40	9	Maximum transaction amout at the given *merchant* over the past 14 days
merch_lag3_max	13.60	10	Maximum transaction amout at the given *merchant* over the past 3 days
merch_lag7_tot	14.07	11	Total transaction amout at the given *merchant* over the past 7 days
card_lag7_max	14.30	12	Maximum transaction amout using the given *card* over the past 7 days
merch_lag3_tot	14.53	13	Total transaction amout at the given *merchant* over the past 3 days
card_lag30_avg	14.80	14	Average transaction amout using the given *card* over the past 30 days
card_lag3_benford	15.00	15	Distribution of the first digits of transaction amounts of the given *card* in the past 3 days
merch_lag7_max	15.63	16	Maximum transaction amout at the given *merchant* over the past 7 days
card_lag14_max	16.03	17	Maximum transaction amout using the given *card* over the past 14 days
card_merch_lag7_tot	16.97	18	Total transaction amout using the given *card* at the given *merchant* over the past 7 days
card_lag30_tot	17.50	19	Total transaction amout using the given *card* over the past 30 days
card_state_lag30_tot	17.87	20	Total transaction amout using the given *card* at the given *state* over the past 30 days
card_merch_lag14_tot	17.97	21	Total transaction amout using the given *card* at the given *merchant* over the past 14 days
card_lag3_max	18.07	22	Maximum transaction amout using the given *card* over the past 3 days
card_merch_lag30_med	18.23	23	Median transaction amout using the given *card* at the given *merchant* over the past 30 days
card_zip_lag30_tot	18.27	24	Total transaction amout using the given *card* at the given *zip* over the past 30 days
merch_lag1_max	19.23	25	Maximum transaction amout at the given *merchant* over the past 1 day
card_state_lag14_tot	19.30	26	Total transaction amout using the given *card* at the given *state* over the past 14 days
card_merch_lag30_tot	19.70	27	Total transaction amout using the given *card* at the given *merchant* over the past 30 days
card_lag14_avg	19.93	28	Average transaction amout using the given *card* over the past 14 days
card_lag14_med	20.40	29	Median transaction amout using the given *card* over the past 14 days
card_lag14_tot	23.80	30	Total transaction amout using the given *card* over the past 14 days

5 Model Fitting

This section explains the model fitting process by laying out the methodologies followed by the preliminary results. A total of six machine learning algorithms were tested, including Logistic Regression as a baseline, and more advanced models like Decision Tree, K-Nearest Neighbors, Random Forest, Gradient Boosting Tree, and Neural Network. Multiple train test splits, resampling, and hyperparameter tuning techniques were incorporated to ensure reliable model performance on future data. Fraud detection rate at 3% rejection rate (FDR at 3%) was selected as the performance metric.

5.1 Methodologies

With a relatively small dataset with only around a thousand fraud labels on hand, it is expected that the statistical variance would be huge. The model fitting methodologies were carefully designed in a way to bring down the variance and ensure a well-generalizing model of which performance on future data can be realistically estimated.

Multiple Train Test Splits

First of all, a reasonable estimate of how each algorithm with different hyperparameters performs on previously unseen data is required for model selection. A common practice in the machine learning field is splitting the data into training and testing sets and taking the performance on the testing set as a proximity of how the model performs on new data. However, given the small dataset, test set performance from one random split can be very different from another. Model A can perform better than Model B during one split, but worse in another split, which causes troubles in model selection.

To bring down the variance, the team performed 10 independent random splits on the modeling data, trained each model on 10 different training sets, and calculated the average FDR across 10 testing sets. This average testing FDR was used for model selection.

Down-sampling the Majority Class

The original modeling data has a high good-to-fraud ratio of nearly 92 to 1. Models trained on such imbalanced datasets tend to misclassify the minority class in order to maximize overall accuracy. However, the successful detection of the minority class, the fraud class, is truly the purpose of this project. Therefore, the team resampled the good records in the training set to render a good-to-fraud ratio of 10 to 1 every time a train test split happened.

Hyperparameter Tuning

A hybrid of automated grid search and randomized search techniques, in addition to manual search were used to tune the hyperparameters for each machine learning algorithm. The search usually started with wide ranges and large increments, and gradually progressed to narrower ranges around promising values and finer increments.

Variable Exclusion

Although all the 30 final variables survived the feature selection process, they bear different predictive power as indicated by their rank of importance in Table 4.4.2.1. While the team always started fitting a model on the full set of 30 variables, models of smaller sizes were also tested in which the least important variables were excluded.

5.2 Model Selection

Table 5.2.1 summarizes the performance of some typical models being evaluated. The last three columns represent the average FDR at 3% on the training, testing, and OOT data respectively. This is not an exhaustive table that covers all the models that were assessed but intends to illustrate how key hyperparameters in different value ranges affect model performance. For example, the team tried Neural Network with one layer of a handful of nodes (#7), dozens of nodes (#3) and hundreds of nodes (#8-10). It was observed that models with around 250 nodes performed the best. Gradient Boosting Tree #7 was selected as the final model for that it has the highest FDR on testing, as well as a relatively small performance degradation from the training set.

Table 5.2.1: Model Performance Summary

Model	Parameters										Average FDR at 3%		
Logistics Regression	**Iteration**	**#Variables**	**penalty**	**C**	**solver**	**r1_ratio**					**Train**	**Test**	**OOT**
	1 (default)	25	l1	1	liblinear	N/A					55.61	54.24	52.62
	2	25	l2	1	lbfgs	N/A					55.58	54.23	52.63
	3	25	l2	10	lbfgs	N/A					55.6	54.25	52.63
	4	25	l2	0.1	lbfgs	N/A					55.61	54.29	52.68
	5	25	ElasticNet	1	sage	0.2					55.6	54.29	52.68
	6	25	ElasticNet	1	sage	0.4					55.61	54.29	52.68
	7	25	ElasticNet	1	sage	0.6					55.61	54.29	52.68
	8	25	ElasticNet	1	sage	0.8					55.61	54.29	52.68
	9	20	l1	1	liblinear	N/A					55.54	54.16	52.62
	10	20	l2	1	lbfgs	N/A					55.57	54.16	52.64
	11	20	ElasticNet	1	sage	0.2					55.57	54.16	52.64
	12	20	ElasticNet	1	sage	0.4					55.56	54.16	52.64
	13	20	ElasticNet	1	sage	0.6					55.57	54.16	52.64
	14	20	ElasticNet	1	sage	0.8					55.57	54.16	52.64
Gradient Boosting Tree	**Iteration**	**learning_rate**	**n_estimators**	**max_depth**	**max_features**	**min_samples_leaf**	**min_samples_split**			**subsample**	**Train**	**Test**	**OOT**
	1 (default)	0.1	100	3	None	1	2			1	56.73	55.68	53.96
	2	**0.01**	**800**	**5**	**5**	**30**	**1500**			**0.7**	**57.17**	**55.77**	**54.07**
	3	0.05	210	5	5	30	1100			0.7	56.83	55.68	53.96
	4	0.05	240	5	25	30	500			0.7	57.05	55.64	53.94
	5	0.02	240	5	25	30	500			0.7	56.68	55.51	53.98
	6	0.001	4000	5	25	40	500			0.7	56.4	55.29	53.94
Random Forest	**Iteration**	**bootstrap**	**n_estimators**	**max_depth**	**max_features**	**min_samples_leaf**	**min_samples_split**			**criterion**	**Train**	**Test**	**OOT**
	1 (default)	TRUE	100	None	5	1	2			gini	58.61	55.01	53.65
	2	TRUE	50	20	5	30	500			entropy	56.33	54.9	53.31
	3	TRUE	50	20	5	30	300			gini	56.66	55.32	53.73
Neural Network	**Iteration**	**layer**	**nodes**	**max_iter**	**activation**	**optimizer**	**alpha**	**learning_rate**	**learning_rate_init**	**momentum**	**nesterovs_momentum**	**Train**	**Test**	**OOT**
	1 (default)	1	100	200	relu	adam	0.0001	N/A	N/A	N/A	N/A	56.78	55.71	53.88
	2	1	100	500	relu	sgd	0.0001	constant	0.005	0.9	TRUE	56.49	55.59	53.81
	3	1	100	50	relu	adam	0.0001	N/A	0.01	N/A	N/A	56.76	55.85	53.99
	4	1	5	100	relu	adam	0.0001	N/A	0.005	N/A	N/A	56.42	55.45	53.72
	5	1	50	500	relu	sgd	0.001	adaptive	0.02	0.9	TRUE	56.66	55.7	53.91
	6	1	100	200	relu	sgd	0.0001	adaptive	0.005	0.9	TRUE	56.52	55.62	53.86
	7	1	100	200	relu	sgd	0.0003	constant	0.0759	0.1	FALSE	56.59	55.53	53.86
	8	1	100	300	logistic	adam	0.0008	invscaling	0.0397	N/A	N/A	56.41	55.81	53.73
K-Nearest Neighbors	**Iteration**	**neighbors**				**weights**						**Train**	**Test**	**OOT**
	1 (default)	5				uniform						56.92	53.53	52.23
	2	5				distance						56.97	53.17	52.05
	3	7				uniform						57.27	54.65	53.24
	4	12				uniform						57.1	54.92	53.32
Decision Tree	**Iteration**	**criterion**	**max_depth**	**min_samples_leaf**	**min_samples_split**	**splitter**						**Train**	**Test**	**OOT**
	1 (default)	gini	None	1	2	"best"						58.85	53.33	52.72
	2	gini	20	60	310	random						56.39	54.89	53.55
	3	gini	20	60	300	best						56.75	55.07	53.89
	4	gini	default	79	1350	random						56.08	54.68	53.48

Table 5.2.2 below logs the hyperparameters that were tuned for each algorithm and the value ranges that were tuned through.

Table 5.2.2: Hyperparameters Tuned for Each Algorithm

Logistic Regression sklearn.linear_model.LogisticRegression	
penalty	l1, l2, elastic_net
C	0.0001 ~ 10000
r1_ratio	0 ~ 1

Gradient Boosting Tree sklearn.ensemble.GradientBoostingClassifier	
learning_rate	0.0001 ~ 0.1
n_estimators	100 ~ 20000
max_depth	2 ~ 5
max_features	5 ~ 25, None
min_samples_leaf	1 ~ 200
min_samples_split	2 ~ 500
subsample	0.5 ~ 1

Decision Tree sklearn.tree.DecisionTreeClassifier	
criterion	gini, entropy
max_depth	100 ~ 300, None
min_samples_leaf	1 ~ 100
min_samples_split	2 ~ 100

Random Forest sklearn.ensemble.RandomForestClassifier	
n_estimators	10 ~ 500
criterion	gini, entropy
max_depth	5 ~ 100, None
max_features	5 ~ 25, None
min_samples_leaf	1 ~ 200
min_samples_split	2 ~ 500

K-Nearest Neighbors sklearn.neighbors.KNeighborsClassifier	
n_neighbors	3 ~ 12
weights	uniform, distance

Neural Network sklearn.neural_network.MLPClassifier	
layer*	1, 2
nodes*	5 ~ 500
epoch	50 ~ 500
batch_size	10 ~ 200, auto
activation	relu, logistic
alpha	0.00001 ~ 0.001
learning_rate	constant, adaptive
learning_rate_imit	0.0001 ~ 0.01

* Layer and nodes were specified through the “hidden_layer_sizes” parameter.

6 Results

6.1 Summary of Final Model Performance

Out of all the models, Gradient Boosting Tree (*learning_rate* = 0.1, *n_estimators* = 1000, *max_depth* = 3, *max_features* = 5, *min_samples_leaf* = 30, *min_samples_split* = 100, *subsample* = 1) built on 30 variables was selected as the best and the final model for that it achieved the highest average FDR at 3% rejection rate on the test data.

After deciding on the model, the team, again, conducted train test split and resampling, fitted the best model on the training data and predicted on the training and testing set to generate Table 6.1.1 and 6.1.2. Then the model is refitted on all modeling data and used to predict on the OOT data to create Table 6.1.3. The FDR scores of this model on the training, testing, and OOT data are 100.00%, 95.00%, and 58.10% respectively. The detailed statistics of the model performance on the training, testing, and OOT data are shown in Table 6.1.1-6.1.3.

For each of the three sets, records were split into 100 bins after being sorted in the descending order based on the predicted probability of being a fraud. The first bin in each table has the highest number of frauds ("# Bads"), indicating that the model effectively captured the majority of the fraudulent transactions. The green shaded area summarizes the basic statistics within each bin, while the blue shaded area reflects cumulative statistics of all the bins above, including the current bin.

Table 6.1.1: Key Statistics of Top 20 Bins in Training Data

Training	# Records	# Goods	# Bads	Fraud Rate
	56442	55834	608	0.01077212

	Bin Statistics					Cumulative Statistics						
Population Bin %	# Records	# Goods	# Bads	% Goods	% Bads	Total # Records	Cumulative Goods	Cumulative Bads	% Goods	% Bads (FDR)	KS	FPR
1	565	60	505	10.62%	89.38%	565	60	505	0.11%	83.06%	0.83	0.12
2	565	462	103	81.77%	18.23%	1130	522	608	0.93%	100.00%	0.99	0.86
3	565	565	0	100.00%	0.00%	1695	1087	608	1.95%	100.00%	0.98	1.79
4	565	565	0	100.00%	0.00%	2260	1652	608	2.96%	100.00%	0.97	2.72
5	565	565	0	100.00%	0.00%	2825	2217	608	3.97%	100.00%	0.96	3.65
6	565	565	0	100.00%	0.00%	3390	2782	608	4.98%	100.00%	0.95	4.58
7	565	565	0	100.00%	0.00%	3955	3347	608	5.99%	100.00%	0.94	5.50
8	565	565	0	100.00%	0.00%	4520	3912	608	7.01%	100.00%	0.93	6.43
9	565	565	0	100.00%	0.00%	5085	4477	608	8.02%	100.00%	0.92	7.36
10	565	565	0	100.00%	0.00%	5650	5042	608	9.03%	100.00%	0.91	8.29
11	565	565	0	100.00%	0.00%	6215	5607	608	10.04%	100.00%	0.90	9.22
12	565	565	0	100.00%	0.00%	6780	6172	608	11.05%	100.00%	0.89	10.15
13	565	565	0	100.00%	0.00%	7345	6737	608	12.07%	100.00%	0.88	11.08
14	565	565	0	100.00%	0.00%	7910	7302	608	13.08%	100.00%	0.87	12.01
15	565	565	0	100.00%	0.00%	8475	7867	608	14.09%	100.00%	0.86	12.94
16	565	565	0	100.00%	0.00%	9040	8432	608	15.10%	100.00%	0.85	13.87
17	565	565	0	100.00%	0.00%	9605	8997	608	16.11%	100.00%	0.84	14.80
18	565	565	0	100.00%	0.00%	10170	9562	608	17.13%	100.00%	0.83	15.73
19	565	565	0	100.00%	0.00%	10735	10127	608	18.14%	100.00%	0.82	16.66
20	565	565	0	100.00%	0.00%	11300	10692	608	19.15%	100.00%	0.81	17.59

Different from the green shaded area, the "% Goods" column in the blue area represents the percentage of total good records that fall in the current bin and the bins above. The "% Bads" column calculated the percentage of total bad records caught so far, starting from the top, which essentially is the FDR at the threshold. Moreover, the "KS" column is the difference between the cumulative "% Bads" and the cumulative "% Goods", and the "FPR" (False Positive Rate) column is the ratio of "Cumulative Goods" to "Cumulative Bads".

Table 6.1.2: Key Statistics of Top 20 Bins in Testing Data

Testing	# Records	# Goods	# Bads	Fraud Rate
	24190	23930	260	0.010748243

	Bin Statistics					Cumulative Statistics						
Population Bin %	# Records	# Goods	# Bads	% Goods	% Bads	Total # Records	Cumulative Goods	Cumulative Bads	% Goods	% Bads (FDR)	KS	FPR
1	242	47	195	19.42%	80.58%	242	47	195	0.20%	75.00%	0.75	0.24
2	242	195	47	80.58%	19.42%	484	242	242	1.01%	93.08%	0.92	1.00
3	242	237	5	97.93%	2.07%	726	479	247	2.00%	95.00%	0.93	1.94
4	242	242	0	100.00%	0.00%	968	721	247	3.01%	95.00%	0.92	2.92
5	242	242	0	100.00%	0.00%	1210	963	247	4.02%	95.00%	0.91	3.90
6	242	239	3	98.76%	1.24%	1452	1202	250	5.02%	96.15%	0.91	4.81
7	242	241	1	99.59%	0.41%	1694	1443	251	6.03%	96.54%	0.91	5.75
8	242	241	1	99.59%	0.41%	1936	1684	252	7.04%	96.92%	0.90	6.68
9	242	241	1	99.59%	0.41%	2178	1925	253	8.04%	97.31%	0.89	7.61
10	242	241	1	99.59%	0.41%	2420	2166	254	9.05%	97.69%	0.89	8.53
11	242	242	0	100.00%	0.00%	2662	2408	254	10.06%	97.69%	0.88	9.48
12	242	242	0	100.00%	0.00%	2904	2650	254	11.07%	97.69%	0.87	10.43
13	242	242	0	100.00%	0.00%	3146	2892	254	12.09%	97.69%	0.86	11.39
14	242	240	2	99.17%	0.83%	3388	3132	256	13.09%	98.46%	0.85	12.23
15	242	242	0	100.00%	0.00%	3630	3374	256	14.10%	98.46%	0.84	13.18
16	242	242	0	100.00%	0.00%	3872	3616	256	15.11%	98.46%	0.83	14.13
17	242	242	0	100.00%	0.00%	4114	3858	256	16.12%	98.46%	0.82	15.07
18	242	242	0	100.00%	0.00%	4356	4100	256	17.13%	98.46%	0.81	16.02
19	242	242	0	100.00%	0.00%	4598	4342	256	18.14%	98.46%	0.80	16.96
20	242	242	0	100.00%	0.00%	4840	4584	256	19.16%	98.46%	0.79	17.91

Table 6.1.3: Key Statistics of Top 20 Bins in OOT Data

OOT	# Records	# Goods	# Bads	Fraud Rate
	12427	12248	179	0.01440412

	Bin Statistics					Cumulative Statistics						
Population Bin %	# Records	# Goods	# Bads	% Goods	% Bads	Total # Records	Cumulative Goods	Cumulative Bads	% Goods	% Bads (FDR)	KS	FPR
1	125	45	80	36.00%	64.00%	125	45	80	0.37%	44.69%	0.44	0.56
2	125	107	18	85.60%	14.40%	250	152	98	1.24%	54.75%	0.54	1.55
3	125	119	6	95.20%	4.80%	375	271	104	2.21%	58.10%	0.56	2.61
4	125	124	1	99.20%	0.80%	500	395	105	3.23%	58.66%	0.55	3.76
5	125	123	2	98.40%	1.60%	625	518	107	4.23%	59.78%	0.56	4.84
6	125	124	1	99.20%	0.80%	750	642	108	5.24%	60.34%	0.55	5.94
7	125	124	1	99.20%	0.80%	875	766	109	6.25%	60.89%	0.55	7.03
8	125	125	0	100.00%	0.00%	1000	891	109	7.27%	60.89%	0.54	8.17
9	125	124	1	99.20%	0.80%	1125	1015	110	8.29%	61.45%	0.53	9.23
10	125	124	1	99.20%	0.80%	1250	1139	111	9.30%	62.01%	0.53	10.26
11	125	122	3	97.60%	2.40%	1375	1261	114	10.30%	63.69%	0.53	11.06
12	125	122	3	97.60%	2.40%	1500	1383	117	11.29%	65.36%	0.54	11.82
13	125	122	3	97.60%	2.40%	1625	1505	120	12.29%	67.04%	0.55	12.54
14	125	125	0	100.00%	0.00%	1750	1630	120	13.31%	67.04%	0.54	13.58
15	125	125	0	100.00%	0.00%	1875	1755	120	14.33%	67.04%	0.53	14.63
16	125	123	2	98.40%	1.60%	2000	1878	122	15.33%	68.16%	0.53	15.39
17	125	122	3	97.60%	2.40%	2125	2000	125	16.33%	69.83%	0.54	16.00
18	125	124	1	99.20%	0.80%	2250	2124	126	17.34%	70.39%	0.53	16.86
19	125	125	0	100.00%	0.00%	2375	2249	126	18.36%	70.39%	0.52	17.85
20	125	125	0	100.00%	0.00%	2500	2374	126	19.38%	70.39%	0.51	18.84

In each table, the first bin in the table shows the highest number of bads, indicating that the fraud system effectively captured the majority of the fraudulent transactions. Looking across the three tables, the model performs the best on the training data (highest FDR in each bin). It then deteriorates on the testing data and performs the worst on the OOT data. This pattern is well expected as models usually perform better on data that they've seen before and on data that are drawn from a similar time period.

6.2 Journey of Fraud Detection

To illustrate the journey of fraud detection on a real-time basis, two specific examples from the perspectives of *Merchnum* and *Cardnum* are provided below.

6.2.1 An Example with *Cardnum*

In Figure 6.2.1.1, the left chart shows how fraud scores of purchases using *Cardnum 5142212038* fluctuated as the count of transactions using this card increased, and the right chart shows how the score changed over time. The spikes of fraud scores in the two figures are matched using the same shaded color.

Figure 6.2.1.1: Time Dependency Plot for *Cardnum 5142212038*

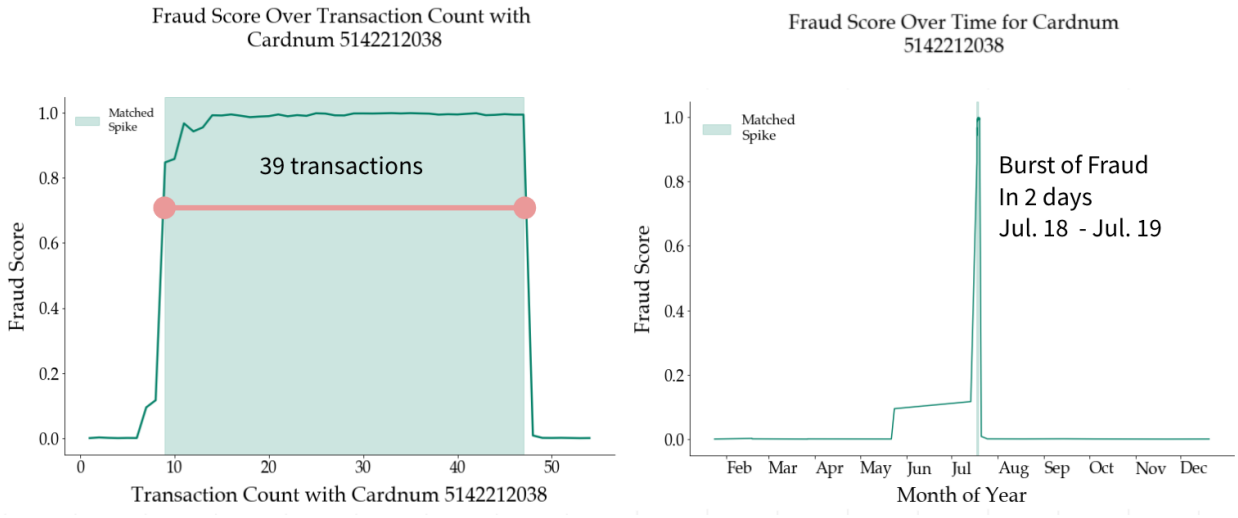


Table 6.2.1.2: First 3 Frauds for *Cardnum 5142212038*

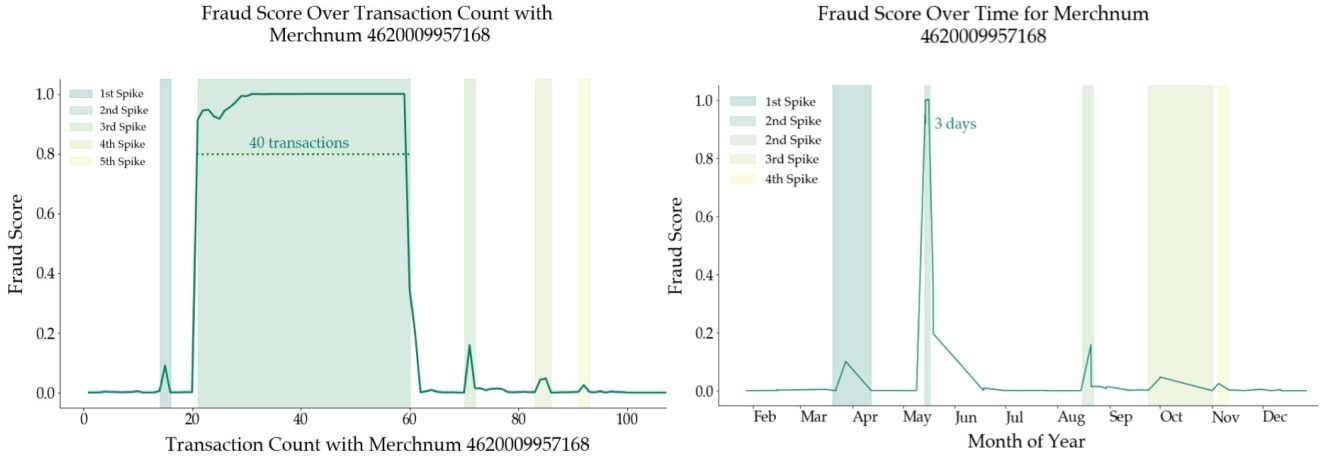
Date	fraud_score	Fraud	count
2010-01-24	0.000224	0	1
2010-02-18	0.002128	0	2
2010-02-18	0.000846	0	3
2010-03-27	0.000140	0	4
2010-03-27	0.000650	0	5
2010-05-22	0.000382	0	6
2010-05-24	0.094513	1	7
2010-07-14	0.116572	1	8
2010-07-18	0.847476	1	9

As shown in Table 6.2.1.2, the first six activities from January 24th to May 22nd are normal transactions. Starting from the 7th transaction, fraudulent activities showed up. However, the fraud score started at a fairly low point as the first two frauds scored only 0.1 in the fraud detection system. Yet this pattern is considered normal as it's inherently difficult to detect the first few fraud records. Then, as Figure 6.2.1.1 shows, when the count of fraudulent transactions continued to increase, the fraud scores quickly soared up, a phenomenon that proves the system's capability in early detection of fraud. Next, a burst of 39 fraud transactions within 2 days in July kept the scores sky-high.

6.2.2 An Example with *Merchnum*

The same fraud detection journey can be illustrated using another entity, ***Merchnum.*** As shown in the leftmost spike in figure 6.2.2.1, the first time the fraud system encountered the transaction, the fraud score only increased slightly. It was not until a burst of 40 fraudulent transactions in May that the fraud score really soared up.

Figure 6.2.2.1: Time Dependency Plot for *Merchnum 5142212038*



6.3 Decide the Cut-off Point

When deciding on the score threshold to reject a transaction, the trade-off between savings generated from catching a fraud and the loss caused by mistaking regular transactions as fraudulent should be carefully considered. To simplify the problem, the following two assumptions were made: for every fraudulent transaction the model catches, there is a $2000 gain, while for every false positive prediction, a loss of $50 incurs.

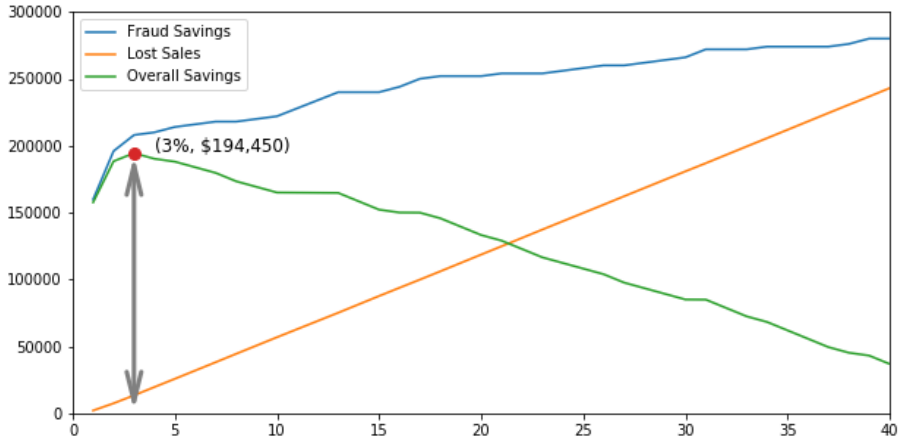
In order to find the best cut-off point, the transactions in the OOT set were sorted in a descending order based on the predicted probability of being fraud and then split into 100 bins. Next, the overall savings at various thresholds were calculated and shown in Table 6.3.1.

Table 6.3.1: Overall Savings at different thresholds

Pop Bin %	CUM Goods	CUM Bads	Fraud Savings	Lost Sales	Overall Savings
1	45	80	$ 160,000	$ 2,250	$ 157,750
2	152	98	$ 196,000	$ 7,600	$ 188,400
3	271	104	$ 208,000	$ 13,550	$ 194,450
4	395	105	$ 210,000	$ 19,750	$ 190,250
5	518	107	$ 214,000	$ 25,900	$ 188,100
6	642	108	$ 216,000	$ 32,100	$ 183,900
7	766	109	$ 218,000	$ 38,300	$ 179,700
8	891	109	$ 218,000	$ 44,550	$ 173,450
9	1015	110	$ 220,000	$ 50,750	$ 169,250
10	1139	111	$ 222,000	$ 56,950	$ 165,050
11	1261	114	$ 228,000	$ 63,050	$ 164,950
12	1383	117	$ 234,000	$ 69,150	$ 164,850
13	1505	120	$ 240,000	$ 75,250	$ 164,750
14	1630	120	$ 240,000	$ 81,500	$ 158,500
15	1755	120	$ 240,000	$ 87,750	$ 152,250
16	1878	122	$ 244,000	$ 93,900	$ 150,100
17	2000	125	$ 250,000	$ 100,000	$ 150,000
18	2124	126	$ 252,000	$ 106,200	$ 145,800
19	2249	126	$ 252,000	$ 112,450	$ 139,550
20	2374	126	$ 252,000	$ 118,700	$ 133,300

The calculation results were visualized in Figure 6.3.2 The final model effectively captured the majority of the fraudulent transactions at an early stage. The best cutoff point to reject the transaction is 3%, which leads to the highest overall savings of $194,450 and an average saving of around $15,647 for every 1,000 transactions.

Figure 6.3.2: Fraud Saving Chart



7 Conclusion

The report documents the process of building a real-time fraud detection system that captures credit card fraud, induced by either a card user or a merchant, at the moment of transaction. The project can be broken down into the following parts:

- **Data exploration**: performed exploratory data analysis to understand the given dataset
- **Data cleaning**: filled in missing values with unique numbers that would not trigger false linkage between irrelevant transactions, and removed one outlier and non-P transactions
- **Variable creation**: created 314 expert variables that captured 1) abnormally high amount of transactions, 2) the frequent occurrence of the same credit card, merchant or other entities, 3) the unusual frequency or amount change with the same credit card or the same merchant, 4) the violation of the Benford's Law, and 5) the likelihood of fraud given the day of week that the transaction happened
- **Feature selection**: applied two univariate filters (KS and FDR) to narrow down candidate variables from 314 to 100, and then a backward selection wrapper to select the final 30 variables for modeling
- **Statistical modeling**: built, tuned and tested a handful of machine learning algorithms including Logistic Regression, Gradient Boosting Tree, Random Forest, Neural Network, K-Nearest Neighbors, and Single Tree

Based on the performance on the OOT data, the final model is a Gradient Boosting Tree classifier (*learning_rate* = 0.1, *n_estimators* = 1000, *max_depth* = 3, *max_features* = 5, *min_samples_leaf* = 30, *min_samples_split* = 100, *subsample* = 1) that achieved an average FDR of 100.00% at a 3% rejection threshold on the training data, 94.85% on the testing data, and 55.53% on the OOT data. Based on the prediction results, as well as the assumptions made about the savings generated from catching a fraud and the loss caused by rejecting a normal customer, 3% was decided as the best cutoff point to reject transactions. This decision is expected to yield a saving of around $15,647 for every 1,000 transactions.

Due to the time constraint of the project, there were only a limited amount of things that could be explored. However, if more time were given, the final model could have been improved through the ways suggested below:

- Consult experts in the field to create more powerful variables that can separate fraud and non-fraud records better, which could lead to better model performance
- Conduct more thorough hyperparameter searching and try more classifiers to optimize model performance
- Fit and test on the testing and OOT data more times to further reduce variance and get more reliable results

Appendix A: Data Quality Report

Data Overview

The dataset was originated from the government agency of Tennessee, containing total 10 fields (*Recnum, Cardnum, Date, Merchnum, Merch description, Merch state, Merch zip, transtype, Amount,* and *Fraud*) and 96,753 credit card transaction records happening during the period from January 1 to December 31, 2010. The fraud records were artificially labelled by the professor based on his experience in fraud detection.

- Summary Table of Categorical Fields

Field Name	# Records	% Populated	# Unique Values	Most Common Field Value
Recnum	96753	100.00%	96753	NA
Cardnum	96753	100.00%	1645	5142148452
Date	96753	100.00%	365	2010-02-28
Merchnum	93378	96.51%	13091	930090121224
Merch description	96753	100.00%	13126	GSA-FSS-ADV
Merch state	95558	98.76%	227	TN
Merch zip	92097	95.19%	4567	38118
transtype	96753	100.00%	4	P
Fraud	96753	100.00%	2	0

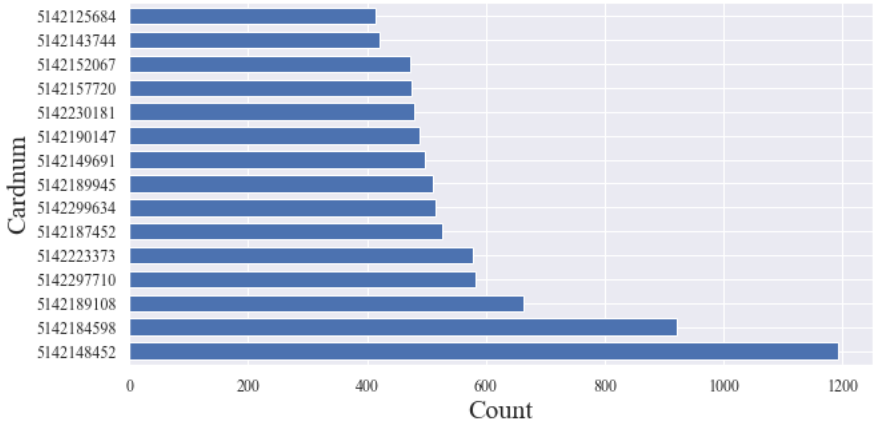
- Summary Statistics of Numerical Field

Field Name	# Records	% Populated	# Unique Values	Mean	Standard Deviation	Min	Max
Amount	96753	100%	34909	427.88	10006.14	0.01	3102045.5

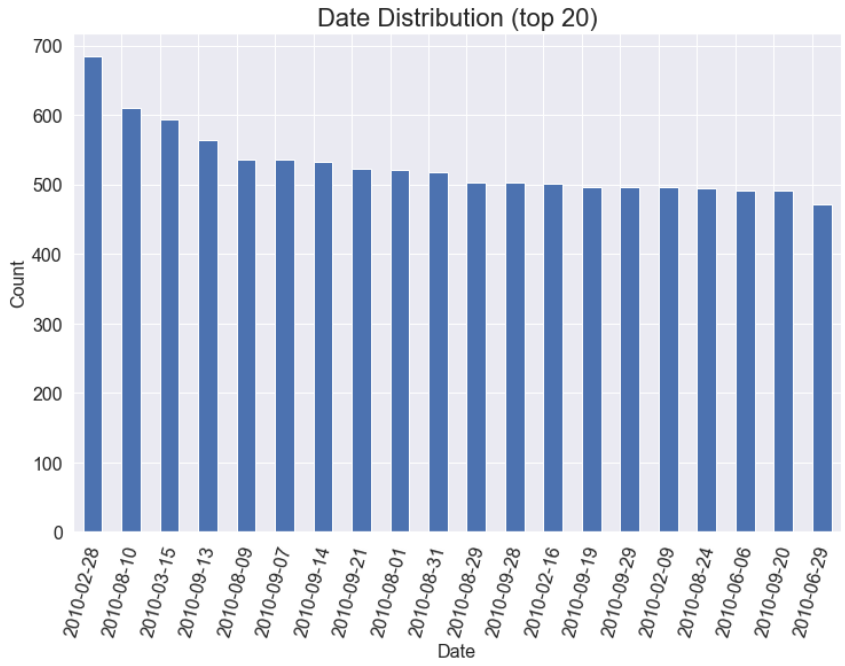
Field Descriptions

- **Recnum (categorical):** the record number for each record, which is a unique identifier for each credit card transaction. Because each record is a unique number corresponding to each row, it is not necessary to plot the distribution.

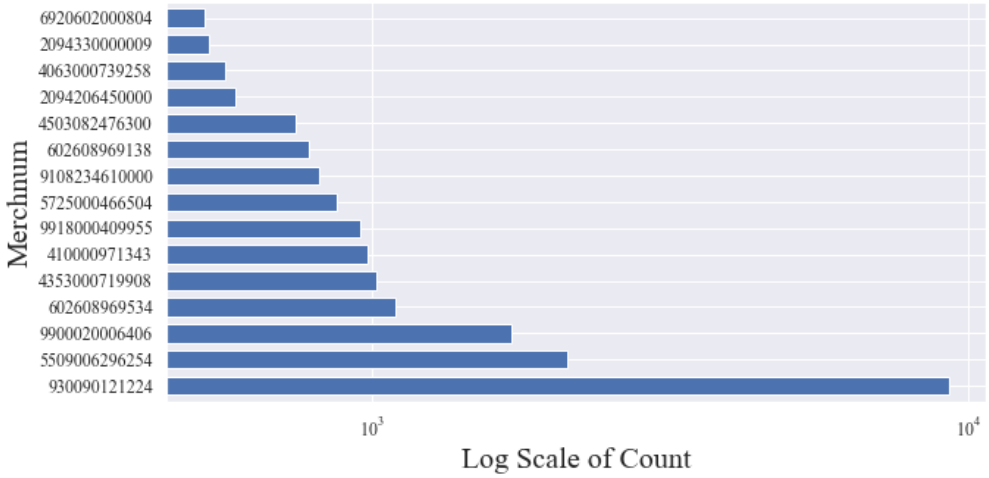
- **Cardnum (categorical):** credit card number, on which credit card transactions were made. All credit card numbers starting with 5 are issued by Mastercard. In fact, all credit card numbers, which appear in the dataset, are issued by Mastercard.



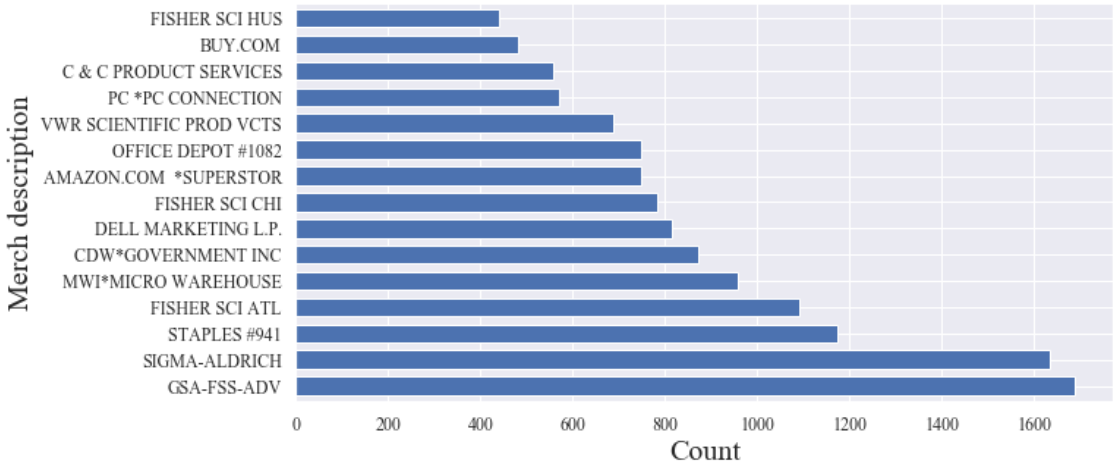
- **Date (categorical):** the date when credit card transactions occurred. This field will be an important field for creating linkage among transaction records.



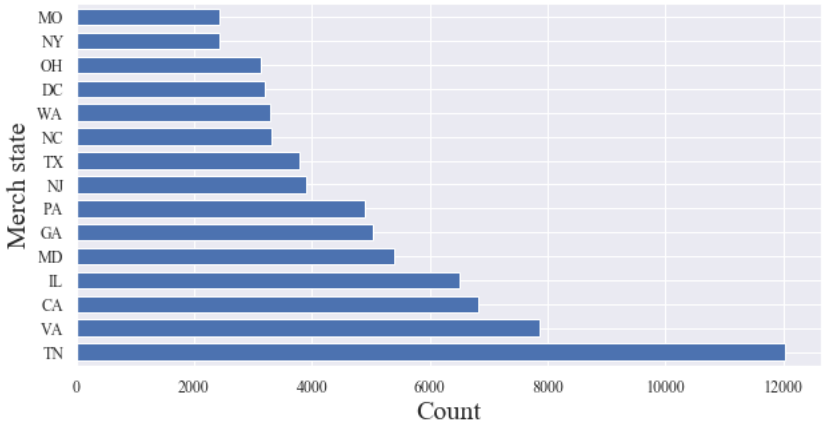
- **Merchnum (categorical):** merchant's unique identifiers to reveal merchants' identities



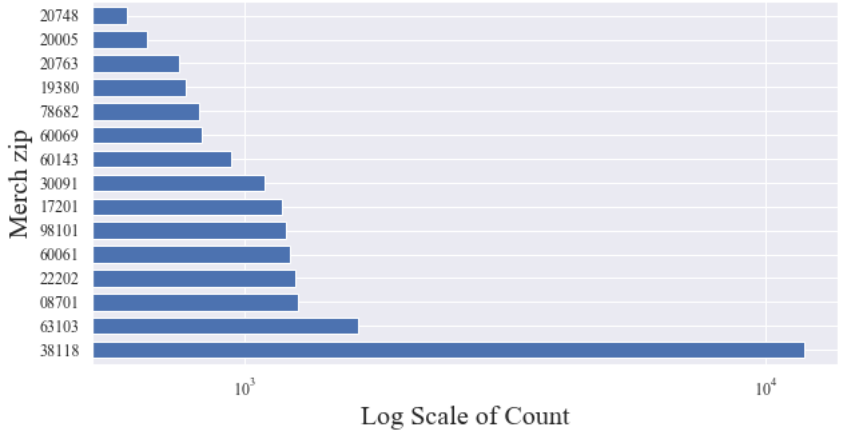
- **Merch description (categorical):** description of merchant names



- **Merch state (categorical):** states in which merchants located. Most merchants with which transactions were made situated in Tennessee (12,035 occurrences)



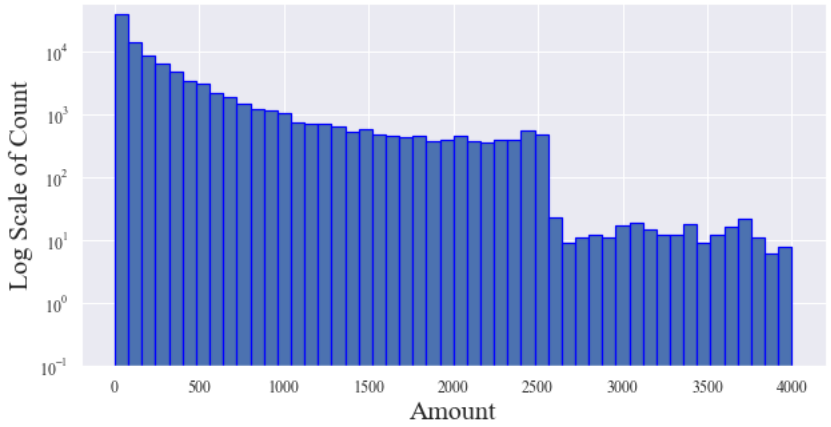
- **Merch zip (categorical):** zip codes of merchants



- **Transtype (categorical):** types of credit card transactions. One character that indicates the type of the transaction. Takes four possible values with "P" being the most common value (99.6%, see Table 2.8 below).

Transtype	Count	Pct
P	96398	99.63%
A	181	0.19%
D	173	0.18%
Y	1	0.00%

- **Amount (numerical):** dollar amount of each credit transaction. Based on the chart, the reason for why the number of transactions exceeding $3,000 dropped significantly was because beyond this threshold, a transaction must be approved and signed off by managers, which, in turn, discouraged government workers from making transactions with such high amounts.



- **Fraud (categorical):** fraud labels of credit card transactions. 1 means a fraudulent transaction (a bad record), while 0 means a transaction is a good record. The fraud labels were artificially created by the professor based on his years of experience in the field of fraud analytics .

Fraud	Count	Pct
0	95694	98.91%
1	1059	1.09%

Appendix B: 314 Candidate Variables

1. Amount Variables: 240 total variables

No.	Variable Name	Description
1	card_merch_lag0_act_avg	Average transaction amount using the given *card* at the given *merchant* over the past 0, 1, 3, 7, 14, 30 days
2	card_merch_lag1_act_avg	
3	card_merch_lag3_act_avg	
4	card_merch_lag7_act_avg	
5	card_merch_lag14_act_avg	
6	card_merch_lag30_act_avg	
7	card_state_lag0_act_avg	Average transaction amount using the given *card* in the given *state* over the past 0,1,3,7,14,30 day
8	card_state_lag1_act_avg	
9	card_state_lag3_act_avg	
10	card_state_lag7_act_avg	
11	card_state_lag14_act_avg	
12	card_state_lag30_act_avg	
13	card_zip_lag0_act_avg	Average transaction amount using the given *card* in the given *zip code* area over the past 0,1,3,7,14,30 day
14	card_zip_lag1_act_avg	
15	card_zip_lag3_act_avg	
16	card_zip_lag7_act_avg	
17	card_zip_lag14_act_avg	
18	card_zip_lag30_act_avg	
19	card_lag0_act_avg	Average transaction amount using the given *card* over the past 0,1,3,7,14,30 day
20	card_lag1_act_avg	
21	card_lag3_act_avg	
22	card_lag7_act_avg	
23	card_lag14_act_avg	
24	card_lag30_act_avg	
25	merch_lag0_act_avg	Average transaction amount at the given *merchant* over the past 0,1,3,7,14,30 day
26	merch_lag1_act_avg	
27	merch_lag3_act_avg	
28	merch_lag7_act_avg	
29	merch_lag14_act_avg	
30	merch_lag30_act_avg	

No.	Variable Name	Description
31	card_lag0_act_max	Maximum transaction amount using the given *card* over the past 0,1,3,7,14,30 day
32	card_lag1_act_max	
33	card_lag3_act_max	
34	card_lag7_act_max	
35	card_lag14_act_max	
36	card_lag30_act_max	
37	card_merch_lag0_act_max	Maximum transaction amount using the given *card* at the given *merchant* over the past 0, 1, 3, 7, 14, 30 days
38	card_merch_lag1_act_max	
39	card_merch_lag3_act_max	
40	card_merch_lag7_act_max	
41	card_merch_lag14_act_max	
42	card_merch_lag30_act_max	
43	card_state_lag0_act_max	Maximum transaction amount using the given *card* in the given *state* over the past 0,1,3,7,14,30 day
44	card_state_lag1_act_max	
45	card_state_lag3_act_max	
46	card_state_lag7_act_max	
47	card_state_lag14_act_max	
48	card_state_lag30_act_max	
49	card_zip_lag0_act_max	Maximum transaction amount using the given *card* in the given *zip code* area over the past 0,1,3,7,14,30 day
50	card_zip_lag1_act_max	
51	card_zip_lag3_act_max	
52	card_zip_lag7_act_max	
53	card_zip_lag14_act_max	
54	card_zip_lag30_act_max	
55	merch_lag0_act_max	Maximum transaction amount at the given *merchant* over the past 0,1,3,7,14,30 day
56	merch_lag1_act_max	
57	merch_lag3_act_max	
58	merch_lag7_act_max	
59	merch_lag14_act_max	
60	merch_lag30_act_max	
61	card_lag0_med	Medium transaction amount using the given *card* over the past 0,1,3,7,14,30 day
62	card_lag1_med	
63	card_lag3_med	
64	card_lag7_med	
65	card_lag14_med	
66	card_lag30_med	

No.	Variable Name	Description
67	card_merch_lag0_med	Medium transaction amount using the given *card* at the given *merchant* over the past 0, 1, 3, 7, 14, 30 days
68	card_merch_lag1_med	
69	card_merch_lag3_med	
70	card_merch_lag7_med	
71	card_merch_lag14_med	
72	card_merch_lag30_med	
73	merch_lag0_med	Medium transaction amount at the given *merchant* over the past 0,1,3,7,14,30 day
74	merch_lag1_med	
75	merch_lag3_med	
76	merch_lag7_med	
77	merch_lag14_med	
78	merch_lag30_med	
79	card_state_lag0_med	Medium transaction amount using the given *card* in the given *state* over the past 0,1,3,7,14,30 day
80	card_state_lag1_med	
81	card_state_lag3_med	
82	card_state_lag7_med	
83	card_state_lag14_med	
84	card_state_lag30_med	
85	card_zip_lag0_med	Medium transaction amount using the given *card* in the given *zip code* area over the past 0,1,3,7,14,30 day
86	card_zip_lag1_med	
87	card_zip_lag3_med	
88	card_zip_lag7_med	
89	card_zip_lag14_med	
90	card_zip_lag30_med	
91	card_lag0_tot	Total transaction amount using the given *card* over the past 0,1,3,7,14,30 day
92	card_lag1_tot	
93	card_lag3_tot	
94	card_lag7_tot	
95	card_lag14_tot	
96	card_lag30_tot	
97	card_merch_lag0_tot	Total transaction amount using the given *card* at the given *merchant* over the past 0, 1, 3, 7, 14, 30 days
98	card_merch_lag1_tot	
99	card_merch_lag3_tot	
100	card_merch_lag7_tot	
101	card_merch_lag14_tot	
102	card_merch_lag30_tot	

No.	Variable Name	Description
103	card_state_lag0_tot	Total transaction amount using the given *card* in the given *state* over the past 0,1,3,7,14,30 day
104	card_state_lag1_tot	
105	card_state_lag3_tot	
106	card_state_lag7_tot	
107	card_state_lag14_tot	
108	card_state_lag30_tot	
109	card_zip_lag0_tot	Total transaction amount using the given *card* in the given *zip* code area over the past 0,1,3,7,14,30 day
110	card_zip_lag1_tot	
111	card_zip_lag3_tot	
112	card_zip_lag7_tot	
113	card_zip_lag14_tot	
114	card_zip_lag30_tot	
115	merch_lag0_tot	Total transaction amount at the given *merchant* over the past 0,1,3,7,14,30 day
116	merch_lag1_tot	
117	merch_lag3_tot	
118	merch_lag7_tot	
119	merch_lag14_tot	
120	merch_lag30_tot	
121	card_merch_lag0_act_avg	$\frac{\text{Transaction amount of the current record}}{\text{Average transaction amount using the given } card \text{ at the given } merchant \text{ over past 0, 1, 3, 7, 14, 30 days}}$
122	card_merch_lag1_act_avg	
123	card_merch_lag3_act_avg	
124	card_merch_lag7_act_avg	
125	card_merch_lag14_act_avg	
126	card_merch_lag30_act_avg	
127	card_state_lag0_act_avg	$\frac{\text{Transaction amount of the current record}}{\text{Average transaction amount using the given } card \text{ in the given } state \text{ over the past 0, 1, 3, 7, 14, 30 days}}$
128	card_state_lag1_act_avg	
129	card_state_lag3_act_avg	
130	card_state_lag7_act_avg	
131	card_state_lag14_act_avg	
132	card_state_lag30_act_avg	
133	card_zip_lag0_act_avg	$\frac{\text{Transaction amount of the current record}}{\text{Average transaction amount using the given } card \text{ } in \text{ the given } zip \text{ } code \text{ } area \text{ over the past 0, 1, 3, 7, 14, 30 days}}$
134	card_zip_lag1_act_avg	
135	card_zip_lag3_act_avg	
136	card_zip_lag7_act_avg	
137	card_zip_lag14_act_avg	
138	card_zip_lag30_act_avg	

No.	Variable Name	Description
139	card_lag0_act_avg	Transaction amount of the current record / Average transaction amount using the given *card* over the past 0, 1, 3, 7, 14, 30 days
140	card_lag1_act_avg	
141	card_lag3_act_avg	
142	card_lag7_act_avg	
143	card_lag14_act_avg	
144	card_lag30_act_avg	
145	merch_lag0_act_avg	Transaction amount of the current record / Average transaction amount at the given *merchant* over the past 0, 1, 3, 7, 14, 30 days
146	merch_lag1_act_avg	
147	merch_lag3_act_avg	
148	merch_lag7_act_avg	
149	merch_lag14_act_avg	
150	merch_lag30_act_avg	
151	card_lag0_act_max	Transaction amount of the current record / Maximum transaction amount using the given *card* over the past 0, 1, 3, 7, 14, 30 days
152	card_lag1_act_max	
153	card_lag3_act_max	
154	card_lag7_act_max	
155	card_lag14_act_max	
156	card_lag30_act_max	
157	card_merch_lag0_act_max	Transaction amount of the current record / Maximum transaction amount using the given *card* at the given *merchant* over the past 0, 1, 3, 7, 14, 30 days
158	card_merch_lag1_act_max	
159	card_merch_lag3_act_max	
160	card_merch_lag7_act_max	
161	card_merch_lag14_act_max	
162	card_merch_lag30_act_max	
163	card_state_lag0_act_max	Transaction amount of the current record / Maximum transaction amount using the given *card* at the given *state* over the past 0, 1, 3, 7, 14, 30 days
164	card_state_lag1_act_max	
165	card_state_lag3_act_max	
166	card_state_lag7_act_max	
167	card_state_lag14_act_max	
168	card_state_lag30_act_max	
169	card_zip_lag0_act_max	Transaction amount of the current record / Maximum transaction amount using the given *card* in the given *zip code area* over the past 0, 1, 3, 7, 14, 30 days
170	card_zip_lag1_act_max	
171	card_zip_lag3_act_max	
172	card_zip_lag7_act_max	
173	card_zip_lag14_act_max	
174	card_zip_lag30_act_max	

No.	Variable Name	Description
175	merch_lag0_act_max	Transaction amount of the current record / Maximum transaction amount at the given *merchant* over the past 0, 1, 3, 7, 14, 30 days
176	merch_lag1_act_max	
177	merch_lag3_act_max	
178	merch_lag7_act_max	
179	merch_lag14_act_max	
180	merch_lag30_act_max	
181	card_merch_lag0_act_med	Transaction amount of the current record / Median transaction amount using the given *card* at the given *merchant* over the past 0, 1, 3, 7, 14, 30 days
182	card_merch_lag1_act_med	
183	card_merch_lag3_act_med	
184	card_merch_lag7_act_med	
185	card_merch_lag14_act_med	
186	card_merch_lag30_act_med	
187	card_state_lag0_act_med	Transaction amount of the current record / Median transaction amount using the given *card* in the given *state* over the past 0, 1, 3, 7, 14, 30 days
188	card_state_lag1_act_med	
189	card_state_lag3_act_med	
190	card_state_lag7_act_med	
191	card_state_lag14_act_med	
192	card_state_lag30_act_med	
193	card_zip_lag0_act_med	Transaction amount of the current record / Median transaction amount using the given *card* in the given *zip code area* over the past 0, 1, 3, 7, 14, 30 days
194	card_zip_lag1_act_med	
195	card_zip_lag3_act_med	
196	card_zip_lag7_act_med	
197	card_zip_lag14_act_med	
198	card_zip_lag30_act_med	
199	card_lag0_act_med	Transaction amount of the current record / Median transaction amount using the given *card* over the past 0, 1, 3, 7, 14, 30 days
200	card_lag1_act_med	
201	card_lag3_act_med	
202	card_lag7_act_med	
203	card_lag14_act_med	
204	card_lag30_act_med	
205	merch_lag0_act_med	Transaction amount of the current record / Median transaction amount at the given *merchant* over the past 0, 1, 3, 7, 14, 30 days
206	merch_lag1_act_med	
207	merch_lag3_act_med	
208	merch_lag7_act_med	
209	merch_lag14_act_med	
210	merch_lag30_act_med	

No.	Variable Name	Description
211	card_merch_lag0_act_tot	Transaction amount of the current record / Total transaction amount using the given *card* at the given merchant over the past 0, 1, 3, 7, 14, 30 days
212	card_merch_lag1_act_tot	
213	card_merch_lag3_act_tot	
214	card_merch_lag7_act_tot	
215	card_merch_lag14_act_tot	
216	card_merch_lag30_act_tot	
217	card_state_lag0_act_tot	Transaction amount of the current record / Total transaction amount using the given *card* in the given *state* over the past 0, 1, 3, 7, 14, 30 days
218	card_state_lag1_act_tot	
219	card_state_lag3_act_tot	
220	card_state_lag7_act_tot	
221	card_state_lag14_act_tot	
222	card_state_lag30_act_tot	
223	card_zip_lag0_act_tot	Transaction amount of the current record / Total transaction amount using the given *card* in the given *zip code area* over the past 0, 1, 3, 7, 14, 30 days
224	card_zip_lag1_act_tot	
225	card_zip_lag3_act_tot	
226	card_zip_lag7_act_tot	
227	card_zip_lag14_act_tot	
228	card_zip_lag30_act_tot	
229	card_lag0_act_tot	Transaction amount of the current record / Total transaction amount using the given *card* over the past 0, 1, 3, 7, 14, 30 days
230	card_lag1_act_tot	
231	card_lag3_act_tot	
232	card_lag7_act_tot	
233	card_lag14_act_tot	
234	card_lag30_act_tot	
235	merch_lag0_act_tot	Transaction amount of the current record / Total transaction amount at the given *merchant* over the past 0, 1, 3, 7, 14, 30 days
236	merch_lag1_act_tot	
237	merch_lag3_act_tot	
238	merch_lag7_act_tot	
239	merch_lag14_act_tot	
240	merch_lag30_act_tot	

2. Velocity variables: 30 total variables

No.	Variable Name	Description
1	card_lag0_cnt	# of appearance of the given *card* in the past 0, 1, 3, 7, 14, 30 days
2	card_lag1_cnt	
3	card_lag3_cnt	
4	card_lag7_cnt	
5	card_lag14_cnt	
6	card_lag30_cnt	
7	merch_lag0_cnt	# of appearance of the given *merchant* in the past 0, 1, 3, 7, 14, 30 days
8	merch_lag1_cnt	
9	merch_lag3_cnt	
10	merch_lag7_cnt	
11	merch_lag14_cnt	
12	merch_lag30_cnt	
13	card_state_lag0_cnt	# of appearance of the given *card* in the given *state* in the past 0, 1, 3, 7, 14, 30 days
14	card_state_lag1_cnt	
15	card_state_lag3_cnt	
16	card_state_lag7_cnt	
17	card_state_lag14_cnt	
18	card_state_lag30_cnt	
19	card_zip_lag0_cnt	# of appearance of the given *card* in the given *zip code area* in the past 0, 1, 3, 7, 14, 30 days
20	card_zip_lag1_cnt	
21	card_zip_lag3_cnt	
22	card_zip_lag7_cnt	
23	card_zip_lag14_cnt	
24	card_zip_lag30_cnt	
25	card_merch_lag0_cnt	# of appearance of the given *card* at the given *merchant* in the past 0, 1, 3, 7, 14, 30 days
26	card_merch_lag1_cnt	
27	card_merch_lag3_cnt	
28	card_merch_lag7_cnt	
29	card_merch_lag14_cnt	
30	card_merch_lag30_cnt	

3. Days-since variables: 5 total variables

No.	Variable Name	Description
1	merch_#days_since	# days since the last appearance of the given *merchant*
2	card_merch_#days_since	# days since the last appearance of the given *card* at the given *merchant*
3	card_state_#days_since	# days since the last appearance of the given *card* in the given *state*
4	card_zip_#days_since	# days since the last appearance of the given *card* in the given *zip code area*
5	card_#days_since	# days since the last appearance of the given *card*

4. Velocity change variables: 24 total variables

No.	Variable Name	Description
1	card_lag0_lag7_cnt	Number of transactions using the same *card* in the past 0, 1 days / Average daily transactions using the same *card* in the past 7, 14, 30 days
2	card_lag0_lag14_cnt	
3	card_lag0_lag30_cnt	
4	card_lag1_lag7_cnt	
5	card_lag1_lag14_cnt	
6	card_lag1_lag30_cnt	
7	merch_lag0_lag7_cnt	Number of transactions at the same *merchant* in the past 0, 1 days / Average daily transactions at the same *merchant* in the past 7, 14, 30 days
8	merch_lag0_lag14_cnt	
9	merch_lag0_lag30_cnt	
10	merch_lag1_lag0_cnt	
11	merch_lag1_lag7_cnt	
12	merch_lag1_lag14_cnt	
13	card_lag0_lag7_amt	Total transaction amount using the same *card* in the past 0, 1 days / Average daily transaction amount using the same *card* in the past 0, 1 days
14	card_lag0_lag14_amt	
15	card_lag0_lag30_amt	
16	card_lag1_lag7_amt	
17	card_lag1_lag14_amt	
18	card_lag1_lag30_amt	
19	merch_lag0_lag7_amt	Total transaction amount at the same *merchant* in the past 0, 1 days / Average daily transaction amount at the same *merchant* in the past 0, 1 days
20	merch_lag0_lag14_amt	
21	merch_lag0_lag30_amt	
22	merch_lag1_lag7_amt	
23	merch_lag1_lag14_amt	
24	merch_lag1_lag30_amt	

5. Benford's Law variables: 14 total variables

No.	Variable Name	Description
1	card_lag0_benford	Distribution of the first digits of transaction amounts of the given *card* in the past 0, 1, 3, 7, 14, 30 days
2	card_lag1_benford	
3	card_lag3_benford	
4	card_lag7_benford	
5	card_lag14_benford	
6	card_lag30_benford	
7	merch_lag0_benford	Distribution of the first digits of transaction amounts of the given *merchant* in the past 0, 1, 3, 7, 14, 30 days
8	merch_lag1_benford	
9	merch_lag3_benford	
10	merch_lag7_benford	
11	merch_lag14_benford	
12	merch_lag30_benford	
13	card_benford	Distribution of the first digits of all past transaction amounts of the given *card*
14	merch_benford	Distribution of the first digits of all past transaction amounts of the given *merchant*

6. Day-of-Week variable: 1 total (average likelihood of fraud on a given day of week)

Appendix C: 100 Variables Selected by the Filters

No.	Variable Name	KS	FDR	KS Rank	FDR Rank	Avg Rank
1	fraud	1.000	1.000	316	316	316
2	card_merch_lag7_tot	0.683	0.634	315	314	314.5
3	card_zip_lag7_tot	0.679	0.632	314	313	313.5
4	card_merch_lag14_tot	0.678	0.632	313	313	313
5	card_merch_lag3_tot	0.677	0.631	312	311	311.5
6	card_state_lag3_tot	0.676	0.630	311	310	310.5
7	card_zip_lag3_tot	0.671	0.635	310	315	312.5
8	card_state_lag7_tot	0.671	0.600	309	307	308
9	card_state_lag14_tot	0.670	0.524	308	295	301.5
10	card_zip_lag14_tot	0.668	0.629	307	309	308
11	card_merch_lag30_tot	0.662	0.562	306	301	303.5
12	card_state_lag1_tot	0.660	0.606	305	308	306.5
13	card_merch_lag1_tot	0.660	0.599	304	306	305
14	card_merch_lag14_max	0.656	0.475	303	284	293.5
15	card_zip_lag1_tot	0.654	0.592	302	305	303.5
16	card_merch_lag30_max	0.652	0.474	301	281	291
17	card_merch_lag7_max	0.652	0.465	300	279	289.5
18	card_zip_lag30_tot	0.651	0.565	299	303	301
19	card_zip_lag14_max	0.650	0.475	298	284	291
20	card_zip_lag7_max	0.650	0.464	297	278	287.5
21	card_state_lag3_max	0.648	0.474	296	281	288.5
22	card_state_lag7_max	0.648	0.490	295	291	293
23	card_merch_lag3_max	0.645	0.476	294	286	290
24	card_zip_lag30_max	0.645	0.480	293	288	290.5
25	card_zip_lag3_max	0.641	0.475	292	284	288
26	card_state_lag30_tot	0.636	0.446	291	273	282
27	card_state_lag14_max	0.631	0.488	290	290	290
28	card_state_lag1_max	0.626	0.442	289	270	279.5
29	card_merch_lag1_max	0.621	0.457	288	276	282
30	card_zip_lag1_max	0.618	0.455	287	274	280.5
31	merch_lag3_tot	0.617	0.422	286	266	276
32	card_merch_lag0_tot	0.613	0.561	285	300	292.5
33	card_state_lag0_tot	0.612	0.563	284	302	293
34	merch_lag1_tot	0.610	0.491	283	292	287.5
35	merch_lag0_max	0.609	0.445	282	271	276.5

No.	Variable Name	KS	FDR	KS Rank	FDR Rank	Avg Rank
36	card_zip_lag0_tot	0.606	0.554	281	299	290
37	card_state_lag0_max	0.603	0.419	280	265	272.5
38	card_lag3_tot	0.602	0.553	279	298	288.5
39	card_merch_lag0_max	0.601	0.416	278	264	271
40	card_lag7_tot	0.600	0.518	277	294	285.5
41	card_state_lag30_max	0.599	0.478	276	287	281.5
42	card_zip_lag0_max	0.598	0.415	275	263	269
43	card_zip_lag30_avg	0.596	0.297	274	223	248.5
44	merch_lag1_max	0.595	0.441	273	269	271
45	card_merch_lag30_avg	0.594	0.291	272	215	243.5
46	card_merch_lag3_avg	0.590	0.297	271	223	247
47	card_zip_lag3_avg	0.590	0.300	270	228	249
48	card_merch_lag14_avg	0.590	0.293	269	217	243
49	card_zip_lag7_avg	0.589	0.297	268	223	245.5
50	merch_lag7_tot	0.589	0.346	267	250	258.5
51	card_state_lag3_avg	0.589	0.305	266	235	250.5
52	card_zip_lag14_avg	0.589	0.294	265	219	242
53	card_merch_lag7_avg	0.588	0.294	264	219	241.5
54	card_state_lag7_avg	0.587	0.309	263	239	251
55	card_lag0_max	0.585	0.425	262	267	264.5
56	merch_lag3_max	0.583	0.446	261	273	267
57	merch_lag0_tot	0.583	0.567	260	304	282
58	merch_lag0_avg	0.583	0.309	259	239	249
59	card_state_lag1_avg	0.583	0.305	258	235	246.5
60	merch_lag1_avg	0.577	0.291	257	215	236
61	card_lag1_tot	0.577	0.545	256	296	276
62	card_merch_lag1_avg	0.577	0.300	255	228	241.5
63	card_zip_lag0_avg	0.576	0.319	254	243	248.5
64	card_zip_lag1_avg	0.575	0.302	253	230	241.5
65	card_merch_lag0_avg	0.574	0.319	252	243	247.5
66	card_state_lag14_avg	0.573	0.316	251	241	246
67	card_state_lag0_avg	0.573	0.324	250	245	247.5
68	card_lag1_avg	0.572	0.355	249	252	250.5
69	card_lag0_tot	0.571	0.552	248	297	272.5
70	card_merch_lag30_med	0.570	0.278	247	198	222.5
71	card_lag1_max	0.570	0.431	246	268	257
72	card_lag0_avg	0.570	0.328	245	248	246.5

No.	Variable Name	KS	FDR	KS Rank	FDR Rank	Avg Rank
73	card_zip_lag3_med	0.570	0.289	244	210	227
74	card_lag3_avg	0.570	0.361	243	253	248
75	card_state_lag30_avg	0.567	0.327	242	246	244
76	merch_lag3_avg	0.566	0.274	241	195	218
77	card_merch_lag3_med	0.566	0.287	240	208	224
78	card_merch_lag1_med	0.564	0.291	239	215	227
79	card_zip_lag0_med	0.564	0.298	238	225	231.5
80	card_merch_lag0_med	0.564	0.300	237	228	232.5
81	card_zip_lag30_med	0.562	0.280	236	203	219.5
82	card_zip_lag1_med	0.562	0.291	235	215	225
83	card_lag3_max	0.561	0.456	234	275	254.5
84	card_state_lag0_med	0.561	0.303	233	231	232
85	card_merch_lag14_med	0.561	0.279	232	200	216
86	card_state_lag3_med	0.560	0.287	231	208	219.5
87	card_state_lag1_med	0.560	0.296	230	220	225
88	card_lag7_max	0.559	0.488	229	290	259.5
89	card_lag0_med	0.558	0.309	228	239	233.5
90	card_lag1_med	0.557	0.332	227	249	238
91	card_zip_lag7_med	0.557	0.283	226	206	216
92	card_merch_lag7_med	0.555	0.280	225	203	214
93	card_zip_lag14_med	0.553	0.280	224	203	213.5
94	card_state_lag7_med	0.552	0.290	223	211	217
95	merch_lag7_max	0.551	0.463	222	277	249.5
96	card_state_lag30_med	0.550	0.298	221	225	223
97	card_lag14_tot	0.548	0.476	220	286	253
98	card_lag7_avg	0.547	0.403	219	258	238.5
99	card_lag3_med	0.545	0.324	218	245	231.5
100	merch_lag0_med	0.541	0.288	217	209	213

Appendix D: 30 Features Rankings in 30 Iterations

Variable	re-rank_1	re-rank_2	re-rank_3	re-rank_4	re-rank_5	re-rank_6	re-rank_7	re-rank_8	re-rank_9	re-rank_10	re-rank_11	re-rank_12	re-rank_13	re-rank_14	re-rank_15	avg_rank	final_rank
card_merch_lag1_tot	4	4	4	28	23	3	3	30	1	5	2	1	2	1	7	9.87	1
card_lag1_tot	12	1	1	2	4	8	1	1	3	1	1	2	9	3	5	10.20	2
card_lag30_max	3	7	7	4	5	7	8	5	5	8	6	4	7	5	11	10.33	3
card_zip_lag14_tot	5	6	25	6	22	1	2	17	15	4	23	6	16	6	4	10.73	4
merch_lag1_tot	18	2	2	19	3	5	6	8	4	9	3	9	4	12	8	10.73	5
card_state_lag30_max	7	11	6	16	12	6	4	22	14	14	8	17	8	28	6	10.83	6
card_state_lag3_tot	1	14	5	14	20	15	28	4	7	2	30	3	3	2	28	11.47	7
card_lag0_tot	25	30	3	1	1	2	16	24	2	3	5	25	24	4	12	12.23	8
merch_lag14_max	2	8	20	5	11	12	7	7	11	16	9	8	10	14	23	13.40	9
merch_lag3_max	10	16	11	18	15	17	14	27	19	18	26	13	17	27	24	13.60	10
merch_lag7_tot	15	9	13	10	19	14	26	11	12	10	10	5	11	16	16	14.07	11
card_lag7_max	6	5	10	3	9	9	23	13	9	19	25	24	20	15	17	14.30	12
merch_lag3_tot	28	18	14	17	18	4	15	9	16	7	7	18	5	7	3	14.53	13
card_lag30_avg	24	22	24	9	17	13	13	23	10	13	13	10	12	17	18	14.80	14
card_lag3_benford	9	24	9	7	29	30	11	2	20	27	20	14	14	25	2	15.00	15
merch_lag7_max	27	19	17	22	13	23	12	18	26	26	21	21	22	9	15	15.63	16
card_lag14_max	11	12	16	13	8	16	5	6	13	15	11	12	6	11	14	16.03	17
card_merch_lag7_tot	14	17	29	11	7	20	21	3	28	6	29	29	1	8	1	16.97	18
card_lag30_tot	29	27	18	24	10	24	20	16	17	22	17	28	21	10	22	17.50	19
card_state_lag30_tot	21	13	27	23	28	28	25	19	23	30	22	16	25	23	20	17.87	20
card_merch_lag14_tot	26	21	30	20	27	26	19	12	6	24	4	23	26	21	30	17.97	21
card_lag3_max	20	15	15	8	24	22	22	14	30	25	14	30	13	18	26	18.07	22
card_merch_lag30_med	8	3	21	12	6	10	29	15	8	20	19	26	29	13	9	18.23	23
card_zip_lag30_tot	23	23	28	15	14	18	27	25	18	17	16	22	28	19	19	18.27	24
merch_lag1_max	19	20	26	30	16	25	17	21	27	23	27	19	18	26	25	19.23	25
card_state_lag14_tot	13	26	19	26	2	27	30	20	29	11	15	27	27	20	10	19.30	26
card_merch_lag30_tot	17	29	8	29	21	19	24	26	25	28	18	11	15	30	27	19.70	27
card_lag14_avg	16	10	12	25	25	29	10	10	24	12	12	20	30	22	21	19.93	28
card_lag14_med	22	28	23	21	30	11	9	29	21	29	28	7	19	24	13	20.40	29
card_lag14_tot	30	25	22	27	26	21	18	28	22	21	24	15	23	29	29	23.80	30

Variable	re-rank_16	re-rank_17	re-rank_18	re-rank_19	re-rank_20	re-rank_21	re-rank_22	re-rank_23	re-rank_24	re-rank_25	re-rank_26	re-rank_27	re-rank_28	re-rank_29	re-rank_30	avg_rank	final_rank
card_merch_lag1_tot	22	15	10	6	17	8	20	11	25	6	3	6	25	1	3	9.87	1
card_lag1_tot	2	28	30	1	13	16	1	30	14	27	18	17	18	17	20	10.20	2
card_lag30_max	16	3	23	23	4	23	13	29	6	11	15	2	10	30	10	10.33	3
card_zip_lag14_tot	12	10	22	16	3	4	2	7	20	15	5	7	16	12	4	10.73	4
merch_lag1_tot	4	6	13	25	25	10	23	15	23	22	9	24	1	7	8	10.73	5
card_state_lag30_max	8	4	14	8	15	15	14	4	13	7	10	14	2	16	6	10.83	6
card_state_lag3_tot	28	17	6	14	11	11	10	2	18	5	7	3	9	3	7	11.47	7
card_lag0_tot	1	16	27	3	8	7	30	28	5	16	13	13	17	2	21	12.23	8
merch_lag14_max	15	26	26	28	2	6	21	23	30	20	11	1	4	4	12	13.40	9
merch_lag3_max	20	9	7	19	16	3	12	8	2	8	8	18	6	6	2	13.60	10
merch_lag7_tot	23	20	9	5	9	19	6	5	19	25	12	15	11	13	28	14.07	11
card_lag7_max	10	22	25	18	7	25	9	20	11	2	20	5	13	14	1	14.30	12
merch_lag3_tot	13	23	24	11	1	26	22	18	17	19	24	27	19	9	5	14.53	13
card_lag30_avg	9	12	1	7	23	17	3	6	21	21	26	4	26	25	24	14.80	14
card_lag3_benford	7	7	19	15	26	2	18	24	12	12	21	16	3	11	15	15.00	15
merch_lag7_max	19	14	3	12	20	14	17	3	7	10	6	9	12	19	13	15.63	16
card_lag14_max	11	19	28	22	6	24	25	16	24	26	14	23	29	15	30	16.03	17
card_merch_lag7_tot	3	30	15	13	30	9	19	27	27	29	16	12	8	24	14	16.97	18
card_lag30_tot	14	27	11	2	27	27	5	21	9	9	17	20	14	10	16	17.50	19
card_state_lag30_tot	27	11	12	4	10	21	8	19	4	4	25	8	7	5	18	17.87	20
card_merch_lag14_tot	6	21	4	9	19	13	28	1	28	3	30	10	24	29	9	17.97	21
card_lag3_max	17	5	16	27	12	5	26	26	16	1	1	22	15	27	25	18.07	22
card_merch_lag30_med	25	1	2	21	22	20	7	22	26	23	22	26	23	21	29	18.23	23
card_zip_lag30_tot	30	13	18	24	24	12	15	13	15	13	23	19	20	8	23	18.27	24
merch_lag1_max	24	2	21	10	5	1	16	12	22	18	19	25	21	23	19	19.23	25
card_state_lag14_tot	21	8	8	17	21	22	24	17	3	24	27	11	22	26	26	19.30	26
card_merch_lag30_tot	5	29	20	20	14	30	4	14	8	17	4	30	27	20	22	19.70	27
card_lag14_avg	29	25	17	26	18	28	27	9	29	28	2	28	5	22	17	19.93	28
card_lag14_med	26	18	29	30	28	18	11	10	1	14	29	21	28	18	27	20.40	29
card_lag14_tot	18	24	5	29	29	29	29	25	10	30	28	29	30	28	11	23.80	30