MGTA495 Fraud Analytics

Project 2 Report

Supervised Model on Card Transaction Data

Team 9

Gaofeng Peng

Qihuan Lin

Shihong Hu

Sihao Gu

Yuwei Tang

Ziyao Chen

Zheyu Li

Index

Executive Summary

The goal of this project is to build a supervised fraud model on the card transaction data. Our team split the project into seven parts: Description of Data, Data Cleaning, Candidate Variable Creation, Feature Selection Process, Model Algorithms, Results and Conclusion.

In Description of Data part, we explore the dataset and write a data quality report (Appendix) to get a basic understanding of variable. Then we do the Data Cleaning part includes removing outliers, filtering the data and filling in missing values. Next we create 255 candidate variables based on RFM (Recency, Frequency and Monetary) concept. In the Feature Selection Process part, we apply filter (KS, FDR score) and wrapper (backwards stepwise selection) method to get our final 20 variables. After these, we choose logistic regression as baseline and try neural network, random forest, XGBoost algorithm with different set of hyperparameter. We find that support vector machine algorithm (set radial as Kernel and the parameter: gamma=0.05 and cost = 1) is the best and it has a 0.715 FDR at 3%.

Description of Data

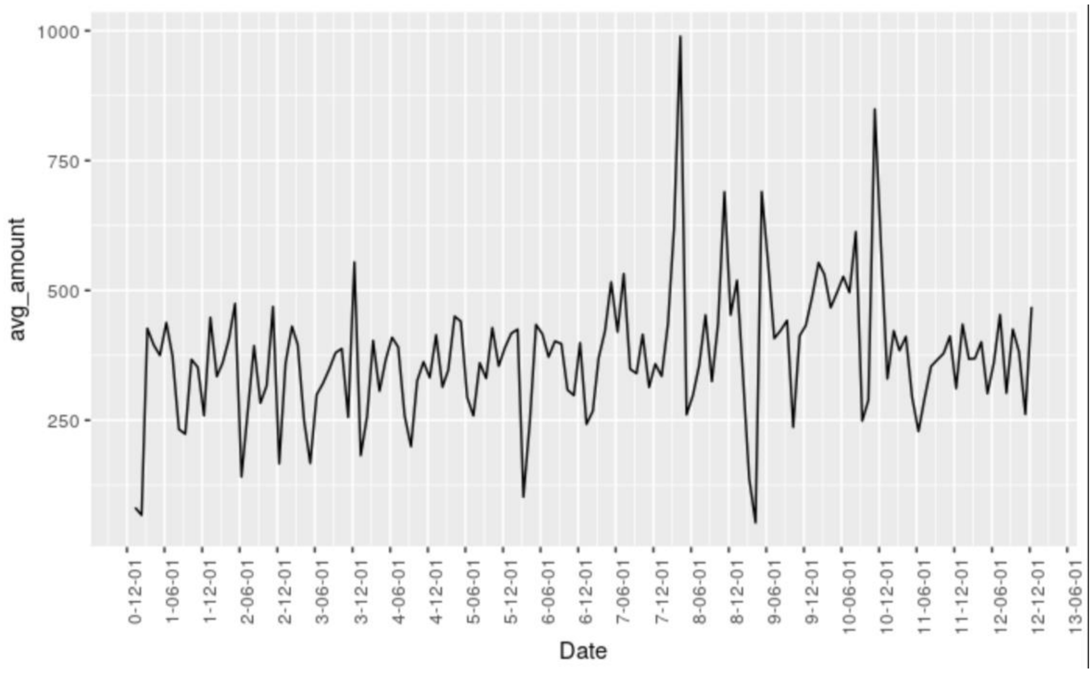
The Card Transaction Data is a real-world transaction data containing relevant merchant transaction information. The time period that it covers is within 01/2010-12/2010. It has 10 fields and 96,753 records.

There are 9 categorical variables and 1 numerical variables. Below are the most important distributions which show the characteristic of the data.

The data quality report is shown in the appendix.

Field 1 Distributions:

Field name: Amount
Description: The amount ($) of each transaction.
Explanation: I removed some outliers and only plotted semi-annual data.

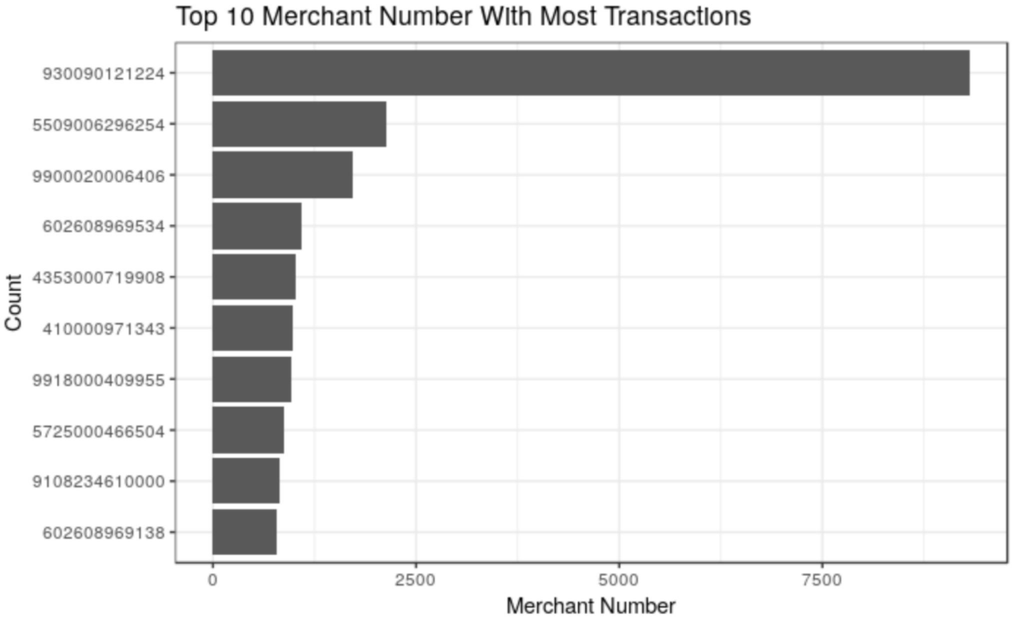


Plot 1. Distribution of Date

Field 2 Distributions:

Field name: Merchnum

Description: The number used to identify a merchant.

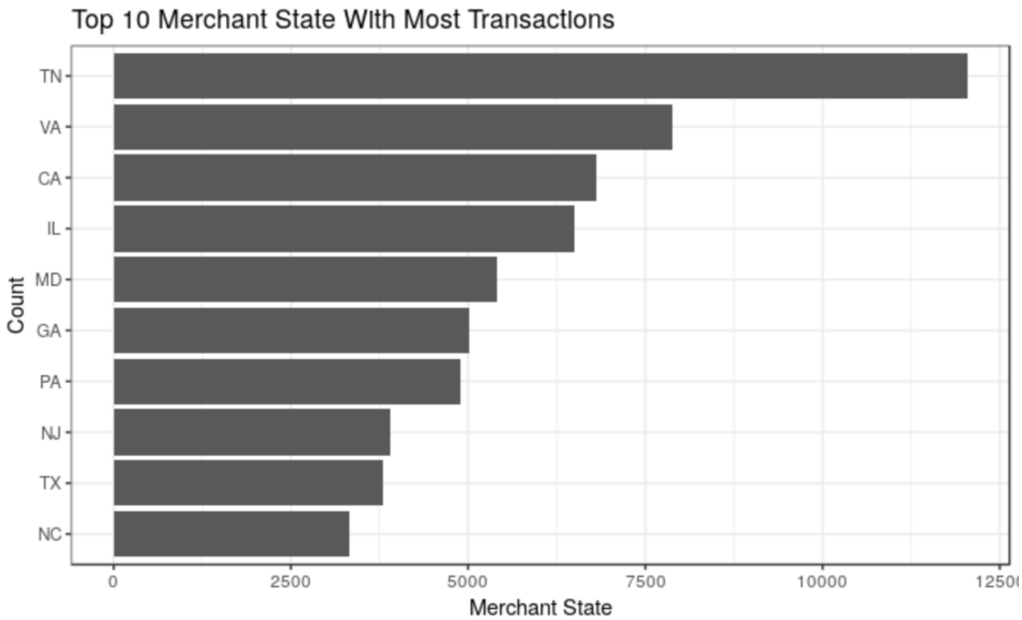


Plot 2. Top 10 Merchant Number With Most Transactions

Field 3 Distributions:

Field name: Merch state

Description: The state of the merchant. (e.g. CA)

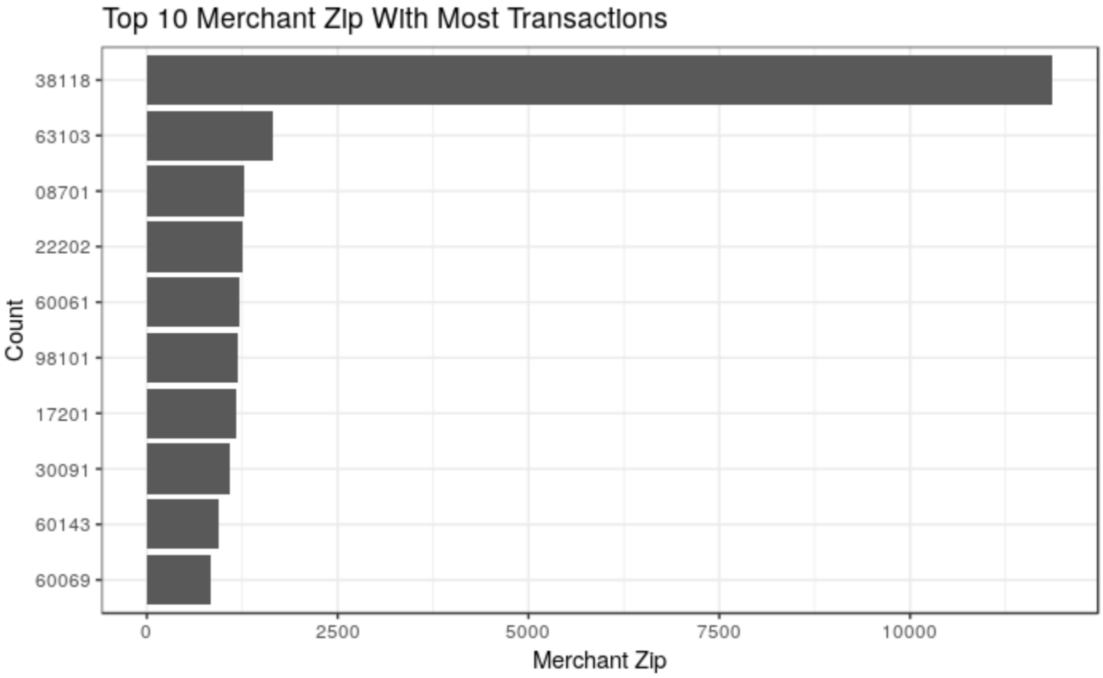


Plot 3. Top 10 Merchant State With Most Transactions

Field 4 Distributions:

Field name: Merch zip

Description: The zip code of the merchant. (5 digits)

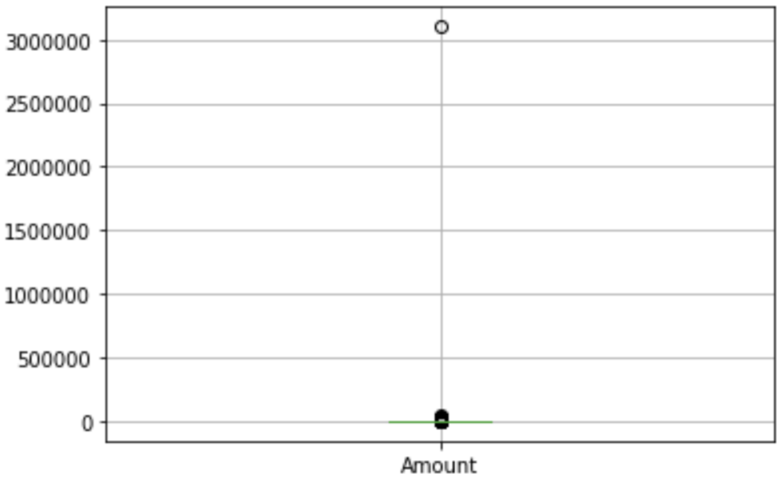


Plot 4. Top 10 Merchant Zip With Most transactions

Data Cleaning

In the data cleaning part, we have taken the following steps to clean the data.

First, we removed outliers. After drawing a boxplot on the amount column, we found there is an obvious outlier. We removed large transaction outlier by filtering data whose transaction amount is smaller than 2500000.



Plot 5. Boxplot of Amount

Second, we filtered that data and left transactions whose transaction type is “P”.

Third, we took several steps to fill in missing values. The original missing values in our dataframe are shown as below:

Column	# of missing values
Merch zip	4300
Merchnum	3198
Merch state	1020

Table 1. Count of Missing Value

- Fill in missing value in zip column

After exploring the data, we found that there are some states whose zips are all null. For these states, we filled in missing values in zip column with the name of the state. After that, there are still 2997 missing values in the zip column. For these missing values, we applied back-fill method and fill missing with the last available value in the column.

- Fill in missing value in merchnum column

Since we think merchants with the same description is likely to have the same merchant number, we first group by merchant description and fill the missing values with the last available merchant number. After that we think it is hard to fill in the correct merchant number, so we filled in other missing values with 'Other_Merchnum'.

- Fill in missing value in state column

Since we think merchants with the same zip should be in the same state, we first group by merchant zip and fill the missing values with the last available merchant state. After that we filled in other missing values with 'Other_State'.

After taking all these steps, there are no missing values in our dataframe.

Candidate Variables

The first step we took was to change the date column to datetime object for future use. To come up with an idea of variable creation, we think of the concept of RFM when R refers to recency, F refers to frequency and M refers to monetary. We consider recency, frequency and monetary as three important factors which can help detect credit card fraud.

Therefore, we created three kinds of variables: amount variables, frequency variables and days since variables. Finally, we created 255 variables in total.

- Amount variables

We calculated the average, maximum, median, total as well as the actual/average, actual/maximum, actual/median, actual/total amount at this card, merchant, card at this merchant,card in this zip code and card in this state over the past 1,3,7,14,30 days.

- Frequency variables

We calculated the number of transactions with this card, merchant, card at this merchant, card in this zip code and card in this state over the past 1,3,7,14,30 days.

- Days since variables

We calculated the current date minus the date of the most recent transaction with the same card, merchant, card in this zip code and card in this state.

*A list of all created variables is shown in appendix 2.

Feature Selection Process

In the feature selection part, we applied filter and wrapper in sequence to get our final variables.

- Filter

We applied two scores to filter out useful variables: KS, FDR

KS

KS (Kolmogorov-Smirnov) is a robust measure of how well two distributions are separated, how well goods are separated from bads. If we plot the cumulative distribution of goods and bads, then KS is the maximum of the difference of the cumulatives. The higher the KS score, the better the feature is helping to separate goods from bads.

FDR

FDR: Fraud Detection Rate represent what percentage of all the frauds are caught at a particular examination cutoff location. In our report, we are using FDR @ 3% as a threshold to judge how well our model is performing. The higher FDR the feature is getting, the better the feature is in helping to capture fraud.

Here is a table with the top 10 variables with the highest average KS and FDR scores:

Field	KS	FDR	Rank KS	Rank FDR	Average Rank
sum_Cardnum_Merchnum_7d	0.678	0.638	257	256	256.5
sum_Cardnum_Merch zip_7d	0.676	0.639	256	257	256.5
sum_Cardnum_Merchnum_14d	0.674	0.630	255	255	255
sum_Cardnum_Merch state_7d	0.670	0.626	254	253.5	253.75
sum_Cardnum_Merch zip_14d	0.665	0.626	251	253.5	252.25
sum_Cardnum_Merch state_3d	0.668	0.607	253	250	251.5
sum_Cardnum_Merch zip_3d	0.662	0.610	249	252	250.5
sum_Cardnum_Merchnum_3d	0.663	0.609	250	251	250.5
sum_Cardnum_Merchnum_30d	0.656	0.561	248	248	248
sum_Cardnum_Merch zip_30d	0.649	0.565	246	249	247.5

Table 2. the top 10 variables with the highest average KS, FDR

After applying the filter, we are removing half of the variables that we have, leaving 128 variables. Our next step is to apply the wrapper to further select variables.

- Wrapper

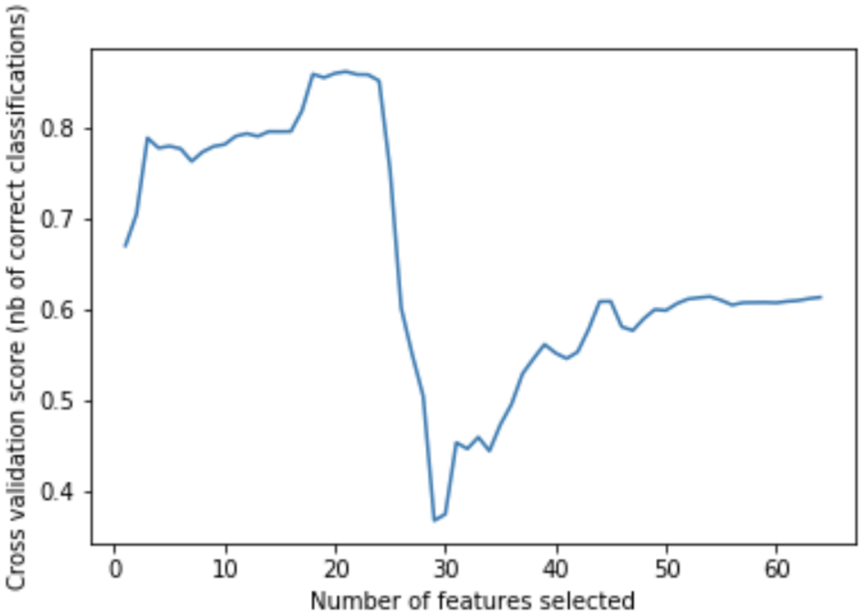
Here we use a model "wrapped" around the feature selection. And the method we are using is backwards stepwise selection.

To avoid overfitting problem, we decided to use simple logistic regression model as the wrapper model. We are using recursive feature elimination with 10-fold cross-validation and ROC/AUC to measure the goodness of the model so as to help us select the best variables.

*ROC means Receiver Operating Characteristic and it is the plot of cumulative goods by cumulative bads.

*AUC means Area Under Curve, it represents the area under the ROC curve.

Here is a plot of the cross-validation result:



Plot 6. Cross Validation of Number of Features Selected

From the plot we can see that when 20 features are selected, we are getting the highest cross-validation score. Therefore, we are keeping 20 variables as our final variables for model input.

A list of final variables we are keeping:

- 'Actual/count_Cardnum_1d',
- 'Actual/mean_Cardnum_30d',
- 'Actual/mean_Merchnum_14d',
- 'Actual/mean_Merchnum_30d',
- 'Actual/mean_Merchnum_7d',
- 'Actual/median_Merchnum_14d',
- 'Actual/median_Merchnum_30d',
- 'Actual/median_Merchnum_7d',
- 'Actual/sum_Cardnum_Merch state_3d',

- 'Actual/sum_Cardnum_Merch zip_3d',
- 'Actual/sum_Cardnum_Merchnum_3d',
- 'count_Cardnum_1d',
- 'count_Cardnum_3d',
- 'count_Cardnum_7d',
- 'count_Cardnum_Merch state_3d',
- 'count_Cardnum_Merch zip_3d',
- 'count_Cardnum_Merchnum_3d',
- 'count_Cardnum_Merchnum_7d',
- 'max_Cardnum_Merchnum_3d',
- 'mean_Cardnum_7d'

Model Algorithms

After feature selection, we have 20 fields (except **Date** and **Recnum**) and then we decide to z-scale all the numeric fields. After scaling the dataset, we filter out those data with date after 2010-11-01 as the oot set and random split the rest set with a fraction of 0.7 and 0.3. This is our train set and test set. For the train set, we applied up-sample method to deal with the unbalance of data and the final train set had the same number of positive and negative fraud data while the test set and oot set remain. The model we used included: Logistic Regression, Support Vector Machine, Random Forest, XGB, Neural Network and we found the SVM was our best model.

- Logistic Regression

Logistic Regression is a regression method used when the dependent variable is dichotomous (binary). It is used to describe data and to explain the relationship between one dependent binary variable and one or more nominal, ordinal, interval or ratio-level independent variables. Logistic regression is using logit function which "squashes" the linear regression function at high and low values, restricting it to between 0 and 1. The parameter we can adjust is the regularization parameter C.

- Support Vector Machine

SVM is a supervised machine learning algorithm which can be used for classification or regression problems. It uses a linear classifier separator, but with the data projected to a higher dimension. Describing in a professional way, it uses a technique called the kernel trick to transform your data and then based on these transformations it finds an optimal boundary between the possible outputs. Simply put, it does data transformations, then figures out how to separate your data based on the labels or outputs you've defined. The parameters we can adjust include kernel type, regularization parameter C, probability etc.

- Random Forest

Random forests are an ensemble learning method for classification, regression and other tasks that operates by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees. For classification problems, the response takes the form of a class membership, which associates, or classifies, a set of independent predictor values with one of the categories present in the dependent variable. Alternatively, for regression problems, the response is an estimate of the dependent variable given the predictors. Random forest builds many trees, each uses only a randomly-chosen subset of variables. Then it combines all the results by averaging or voting. In training the model, the parameters we can adjust include number of estimators, max depth, maximum number of features, etc.

- XGB

XGBoost is an implementation of gradient boosted decision trees designed for speed and performance. Boosting is a way of training a series of weak learners to result in a strong learner. And each weak learner is trained to predict the residual error of the current sum. XFBoost looks at the distribution of features across all data points in a leaf and using this information to reduce the search space of possible feature splits. The parameters we can adjust include # of estimators, maximum depth, learning rate, l1 and l2 regularization terms etc.

- Neural Network

Neural net is an algorithm that is designed to recognize patterns and can either be supervised or unsupervised. Neural networks are typically organized in several layers. Layers are made up of a number of interconnected nodes which contain an activation function. Patterns are presented to the network via the input layer, which communicates to one or more hidden layers where the actual processing is done via a system of weighted connections. Then the hidden layer is linked to an output layer which represent the output of the model. In training the model, we and adjust number of nodes, number of hidden layers, learning rate, etc.

The table below shows how we tuned the model and the results we are getting.

	Parameter		FDR @ 3%		
Logistic Regression	Number of variables		TRAIN	TEST	OOT
1	10 variables		0.543	0.549	0.268
2	15 variables		0.595	0.574	0.263
3	20 variables		0.597	0.594	0.264
Neural Network	Nodes	Decay	TRAIN	TEST	OOT
1	5	0.1	0.764	0.701	0.324
2	5	0.3	0.775	0.701	0.37
3	5	0.5	0.775	0.712	0.36
4	6	0.1	0.761	0.678	0.35
5	6	0.3	0.774	0.62	0.32
6	6	0.5	0.75	0.65	0.3
7	7	0.1	0.787	0.69	0.35
8	7	0.3	0.77	0.72	0.43
9	7	0.5	0.78	0.72	0.34
10	8	0.1	0.798	0.7	0.4
11	8	0.3	0.782	0.64	0.35
12	8	0.5	0.783	0.72	0.34
13	9	0.1	0.8	0.689	0.39
14	9	0.3	0.776	0.71	0.31
15	9	0.5	0.801	0.72	0.32
16	10	0.1	0.85	0.69	0.36
17	10	0.3	0.8	0.69	0.34
18	10	0.5	0.79	0.69	0.38
19	11	0.1	0.85	0.74	0.34
20	11	0.3	0.79	0.74	0.34
21	11	0.5	0.83	0.67	0.4
Random Forest	# of Trees	Max Nodes	TRAIN	TEST	OOT
1	100	24	0.737	0.307	0.089
2	100	23	0.959	0.389	0.1
3	200	24	0.737	0.307	0.089
4	200	23	0.967	0.396	0.1
5	300	24	0.737	0.307	0.093
6	300	23	0.97	0.404	0.1
Xg Boost	Eta	Max Depth	TRAIN	TEST	OOT
1	0.2	4	0.906	0.252	0.201
2	0.2	5	0.955	0.17	0.117
3	0.3	4	0.962	0.126	0.117
4	0.3	5	0.987	0.13	0.123
5	0.4	4	0.974	0.115	0.14
6	0.4	5	0.995	0.111	0.101
Support Vector Machine	Kernel	Cost	TRAIN	TEST	OOT
1	radial	1	0.844	0.714	0.715
2	linear	1	0.518	0.506	0.386
3	poly	1	0.745	0.659	0.603
4	sigmoid	1	0.205	0.17	0.145
5	radial	0.9	0.839	0.719	0.709
6	radial	1.1	0.857	0.709	0.698

Table 3. Model Performance Under Different Parameter

Results

The best algorithm we used is support vector machine and we use radial as Kernel and the parameter: gamma=0.05, cost = 1.

Training	# Records		# Goods		# Bads		Fraud Rate					
	41325		40871		454		0.0110					
	Bin Statistics					Cumulative Statistics						
Population Bin %	# Records	# Goods	# Bads	% Goods	% Bads	Total # Records	Cumulative good	Cumulative Bad	% Good	% Bad(FDR)	KS	FPR
1	414	219	195	52.9	47.1	414	219	195	0.5	43.0	42.4	1.1
2	413	247	166	59.8	40.2	827	466	361	1.1	79.5	78.4	1.3
3	413	391	22	94.7	5.3	1240	857	383	2.1	84.4	82.3	2.2
4	413	405	8	98.1	1.9	1653	1262	391	3.1	86.1	83.0	3.2
5	414	407	7	98.3	1.7	2067	1669	398	4.1	87.7	83.6	4.2
6	413	407	6	98.6	1.5	2480	2076	404	5.1	89.0	83.9	5.1
7	413	408	5	98.8	1.2	2893	2484	409	6.1	90.1	84.0	6.1
8	413	412	1	99.8	0.2	3306	2896	410	7.1	90.3	83.2	7.1
9	414	411	3	99.3	0.7	3720	3307	413	8.1	91.0	82.9	8
10	413	406	7	98.3	1.7	4133	3713	420	9.1	92.5	83.4	8.8
11	413	411	2	99.5	0.5	4546	4124	422	10.1	93.0	82.9	9.8
12	413	411	2	99.5	0.5	4959	4535	424	11.1	93.4	82.3	11
13	414	413	1	99.8	0.2	5373	4948	425	12.1	93.6	81.5	12
14	413	412	1	99.8	0.2	5786	5360	426	13.1	93.8	80.7	13
15	413	412	1	99.8	0.2	6199	5772	427	14.1	94.1	79.9	14
16	413	413	0	100.0	0.0	6612	6185	427	15.1	94.1	78.9	15
17	414	414	0	100.0	0.0	7026	6599	427	16.1	94.1	77.9	16
18	413	413	0	100.0	0.0	7439	7012	427	17.2	94.1	76.9	16
19	413	413	0	100.0	0.0	7852	7425	427	18.2	94.1	75.9	17
20	413	413	0	100.0	0.0	8265	7838	427	19.2	94.1	74.9	18

Table 4. Performance Table of SVM on Training Set

We can see from the result that in the 1% to 3% part FDR rises the fastest. Later, from 3%, the speed of FDR slows down and from 15%, the FDR nearly stops growing. That means from 15% and on, there exist some records which are hard for machine to recognize.

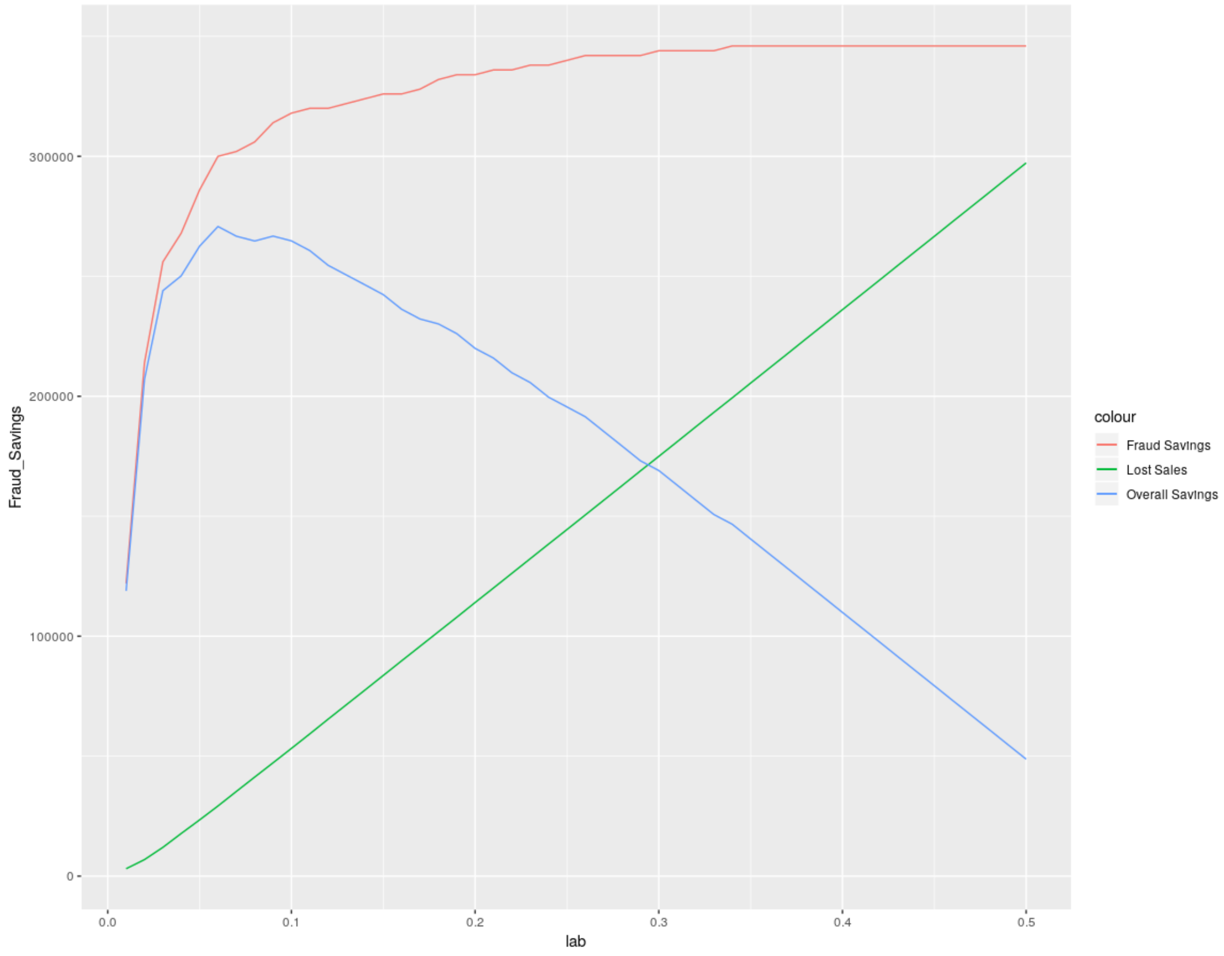
Testing	# Records		# Goods			# Bads		Fraud Rate				
	17712		17530			182		0.0103				
	Bin Statistics					Cumulative Statistics						
Population Bin %	# Records	# Goods	# Bads	% Goods	% Bads	Total # Records	Cumulative good	Cumulative Bad	% Good	% Bad(FDR)	KS	FPR
1	178	97	81	54.5	45.5	178	97	81	0.6	44.5	44.0	1.2
2	177	145	32	81.9	18.1	355	242	113	1.4	62.1	60.7	2.1
3	177	160	17	90.4	9.6	532	402	130	2.3	71.4	69.1	3.1
4	177	169	8	95.5	4.5	709	571	138	3.3	75.8	72.6	4.1
5	177	173	4	97.7	2.3	886	744	142	4.2	78.0	73.8	5.2
6	177	172	5	97.2	2.8	1063	916	147	5.2	80.8	75.5	6.2
7	177	176	1	99.4	0.6	1240	1092	148	6.2	81.3	75.1	7.4
8	177	172	5	97.2	2.8	1417	1264	153	7.2	84.1	76.9	8.3
9	177	175	2	98.9	1.1	1594	1439	155	8.2	85.2	77.0	9.3
10	178	177	1	99.4	0.6	1772	1616	156	9.2	85.7	76.5	10
11	177	176	1	99.4	0.6	1949	1792	157	10.2	86.3	76.0	11
12	177	177	0	100.0	0.0	2126	1969	157	11.2	86.3	75.0	13
13	177	176	1	99.4	0.6	2303	2145	158	12.2	86.8	74.6	14
14	177	174	3	98.3	1.7	2480	2319	161	13.2	88.5	75.2	14
15	177	176	1	99.4	0.6	2657	2495	162	14.2	89.0	74.8	15
16	177	175	2	98.9	1.1	2834	2670	164	15.2	90.1	74.9	16
17	177	177	0	100.0	0.0	3011	2847	164	16.2	90.1	73.9	17
18	177	177	0	100.0	0.0	3188	3024	164	17.3	90.1	72.9	18
19	178	178	0	100.0	0.0	3366	3202	164	18.3	90.1	71.8	20
20	177	176	1	99.4	0.6	3543	3378	165	19.3	90.7	71.4	21

Table 5. Performance Table of SVM on Testing Set

OOT	# Records		# Goods		# Bads	Fraud Rate						
	12236		12057		179	0.0146						
	Bin Statistics					Cumulative Statistics						
Population Bin %	# Records	# Goods	# Bads	% Goods	% Bads	Total # Records	Cumulative good	Cumulative Bad	% Good	% Bad(FDR)	KS	FPR
1	123	62	61	50.4	49.6	123	62	61	0.5	34.1	33.6	1
2	122	76	46	62.3	37.7	245	138	107	1.1	59.8	58.6	1.3
3	123	102	21	82.9	17.1	368	240	128	2.0	71.5	69.5	1.9
4	122	116	6	95.1	4.9	490	356	134	3.0	74.9	71.9	2.7
5	122	113	9	92.6	7.4	612	469	143	3.9	79.9	76.0	3.3
6	123	116	7	94.3	5.7	735	585	150	4.9	83.8	78.9	3.9
7	122	121	1	99.2	0.8	857	706	151	5.9	84.4	78.5	4.7
8	122	120	2	98.4	1.6	979	826	153	6.9	85.5	78.6	5.4
9	123	119	4	96.8	3.3	1102	945	157	7.8	87.7	79.9	6
10	122	120	2	98.4	1.6	1224	1065	159	8.8	88.8	80.0	6.7
11	122	121	1	99.2	0.8	1346	1186	160	9.8	89.4	79.5	7.4
12	123	123	0	100.0	0.0	1469	1309	160	10.9	89.4	78.5	8.2
13	122	121	1	99.2	0.8	1591	1430	161	11.9	89.9	78.1	8.9
14	122	121	1	99.2	0.8	1713	1551	162	12.9	90.5	77.6	9.6
15	123	122	1	99.2	0.8	1836	1673	163	13.9	91.1	77.2	10
16	122	122	0	100.0	0.0	1958	1795	163	14.9	91.1	76.2	11
17	122	121	1	99.2	0.8	2080	1916	164	15.9	91.6	75.7	12
18	123	121	2	98.4	1.6	2203	2037	166	16.9	92.7	75.8	12
19	122	121	1	99.2	0.8	2325	2158	167	17.9	93.3	75.4	13
20	123	123	0	100.0	0.0	2448	2281	167	18.9	93.3	74.4	14

Table 4. Performance Table of SVM on OOT Set

From the table of testing set and oot set, we can see though FDR at 3% is nearly the same, the speed of detecting fraud is different. In the testing set, it can detect 44% fraud at the first bin. In oot, it can only detect 34% at the first bin. Later, in the second bin, the speed of detecting fraud in testing data slows down but the speed of detecting fraud in oot set nearly remains the same. This means in the oot set there exist some records which is more difficult to classify.



Plot 7. Overall Fraud Saving at Different Fraud Percentage

Here's the table used to decide which percent of data should we choose as ***Fraud***. Assume $2000 gain for every fraud that's caught (red curve) and assume $50 loss for every false positive (green curve). The result overall saving is shown in the plot above (blue curve).

The plot shows that fraud saving reached its maximum of $270,750 when set score cut-off at the sweet point of 6%.

Conclusions

For data cleaning, we remove one outlier and fill in missing values in Merchnum, Merch Zip and Merch state fields.

We select features first based on filter using the average of KS and FDR scores, then we applied a wrapper using logistic regression model and backwards stepwise selection function. After the feature selection, we select 20 variables for our final model.

For modeling algorithm, we start with logistic regression with 20 variables, and later use the same algorithm and backward select 15 variables, 10 variables to train models respectively. The result is 3% FDR of 0.268 in OOT set. We take this as a baseline. Afterward, we try neural network, random forest, xgboost algorithm with different set of hyperparameter but only neural network works better than baseline. However, we find the best model using support vector machine algorithm when set kernel type as radial, gamma as 0.05, cost as 1. The result is 0.715 FDR at 3% in OOT set.

For result, using the support vector machine model, we could save a maximum of $270,750 by defining top 6% highest fraud score record as fraud.

Although the result is quite favorable to company in first glance, we don’t know how will it affect the fraud detection rate in next time after applying this model since we will have already label top 6% as fraud and we only have actual data in bias region. If we have any chance and have more time to access data next time, we would like to retrain a model to relabel them and train another model to get the optimize result.

Appendix

Appendix 1

Data Quality Report

1. Data Description

The Card Transaction Data is a real-world transaction data containing relevant merchant transaction information. The time period it covers is within 01/2010-12/2010. It has 10 fields and 96,753 records.

2. Summary

2.1 Numerical Value:

Field name	# records that have value	% populated	# unique values	# records with values zero	Mean	Standard deviation	Min	Max
Amount	96,753	100	34,909	0	427.9	10,006.14	0	3,102,045.5

2.2 Categorical Value:

Field name	# records that have a value	% populated	# unique values	Most common field value
Recnum	96,753	100	96,753	NA
Cardnum	96,753	100	1,645	5142148452

Date	96,753	100	365	2/28/10
Merchnum	93,378	97	13,092	930090121224
Merch description	96,753	100	13,126	GSA-FSS-ADV
Merch state	95,558	99	228	TN
Merch zip	92,097	95	4,568	38118
Transtype	96,753	100	4	P
Fraud	96,753	100	2	0

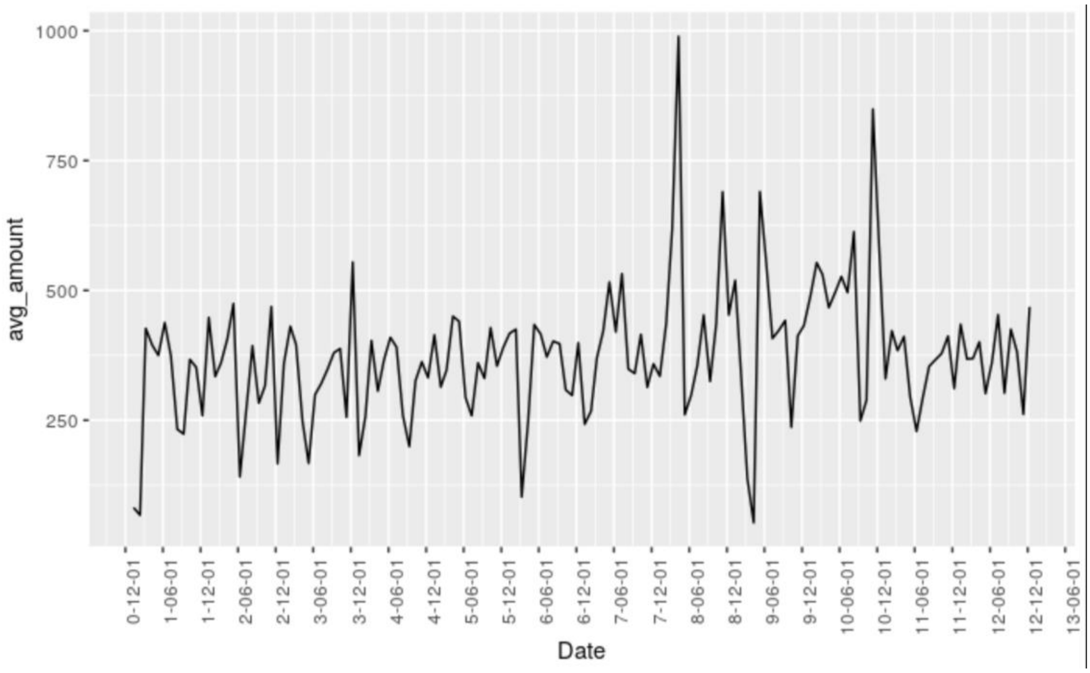
3. Data Field Exploration

Field 1

Field name: Amount

Description: The amount ($) of each transaction.

Explanation: I removed some outliers and only plotted semi-annual data.



Field 2

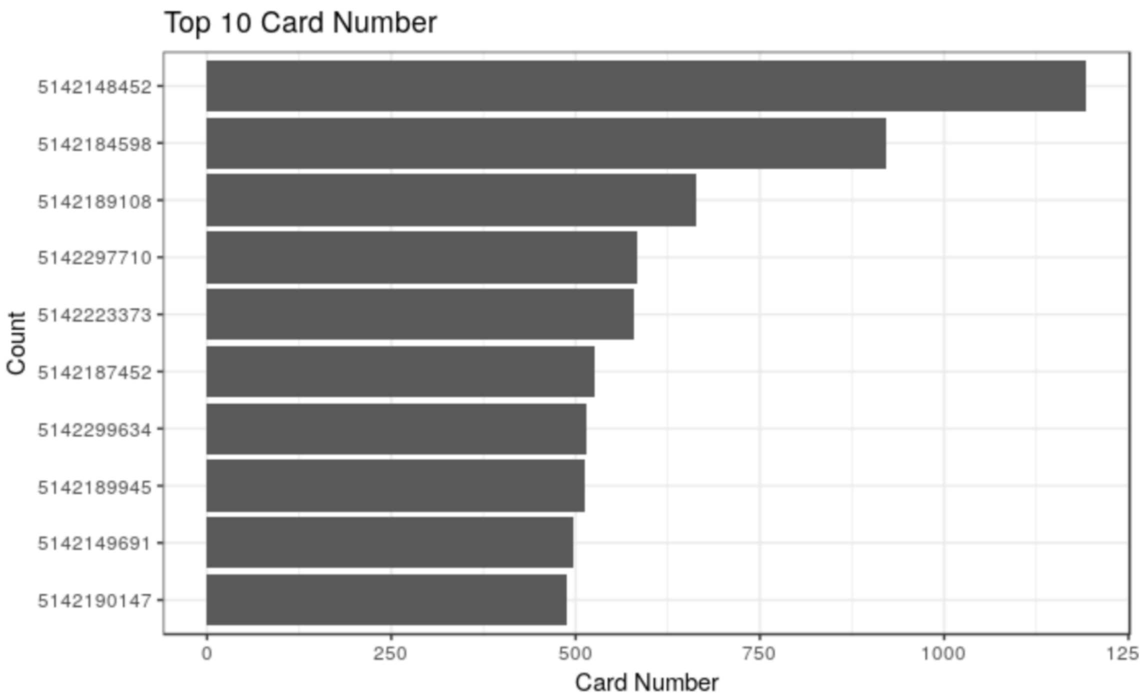
Field name: Recnum

Description: A unique identification for each record.

Field 3

Field name: Cardnum

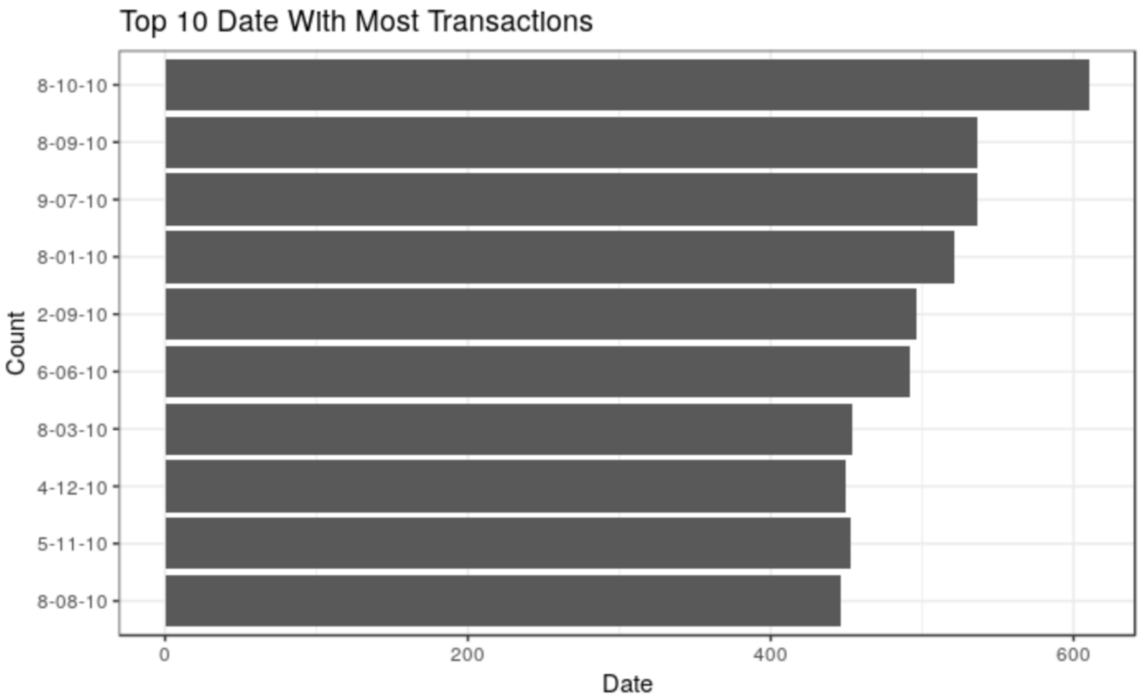
Description: The corresponding card number for that transaction.



Field 4

Field name: Date

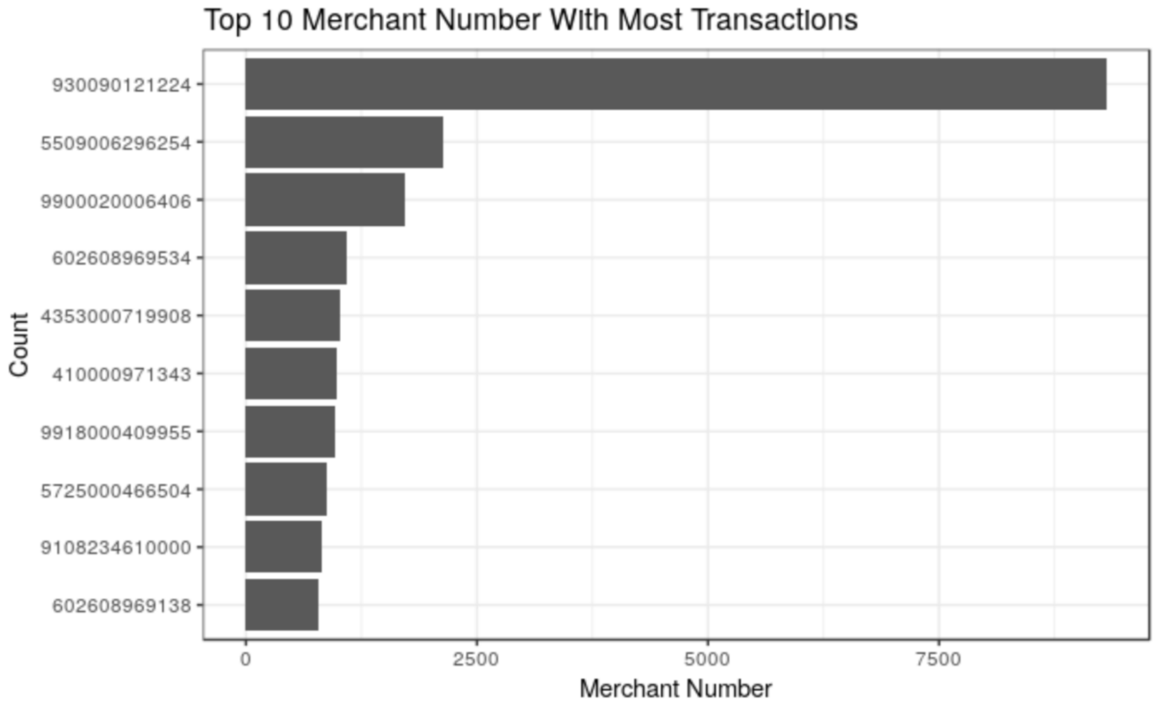
Description: The date of the transaction.



Field 5

Field name: Merchnum

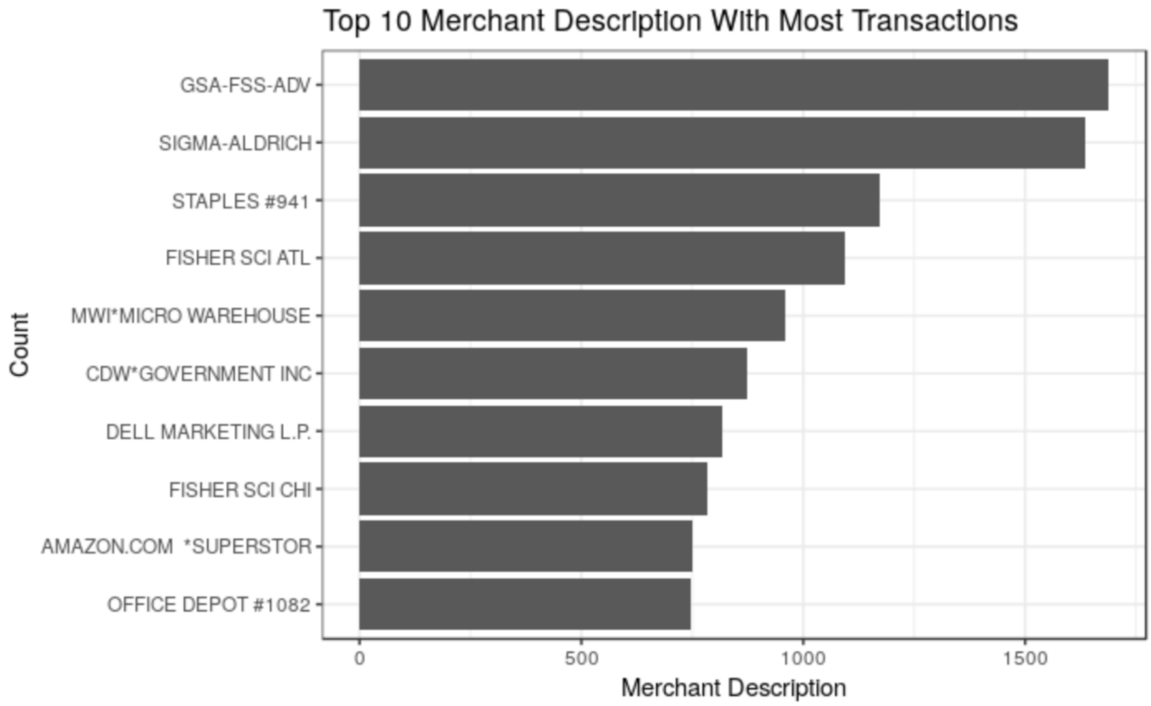
Description: The number used to identify a merchant.



Field 6

Field name: Merch description

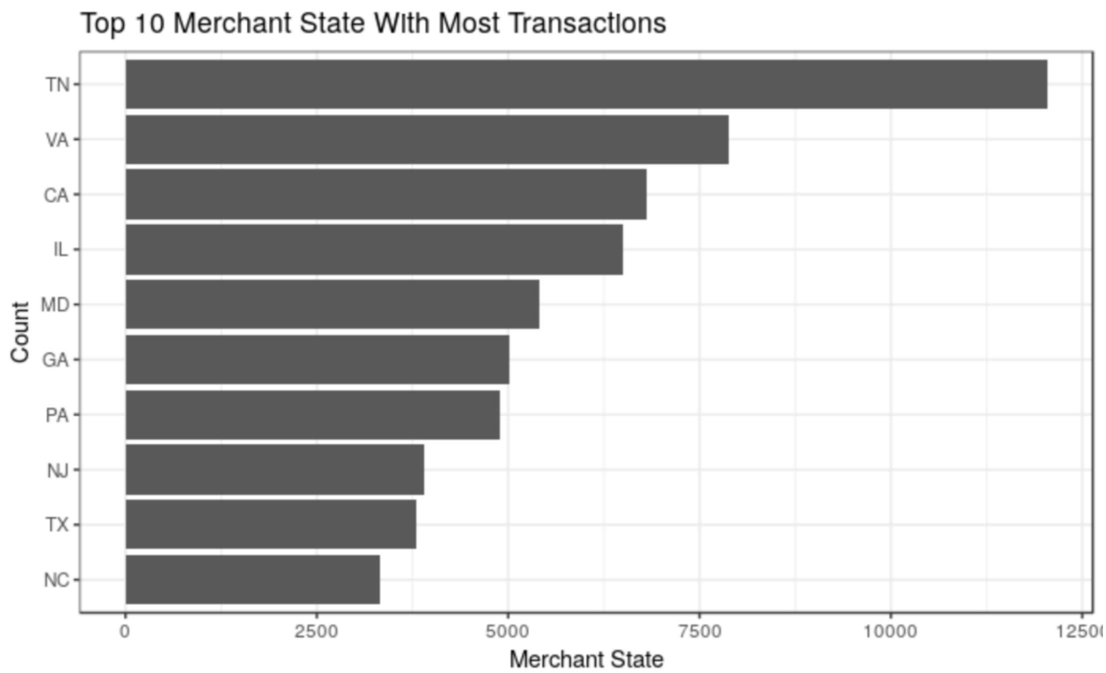
Description: The description of the merchant where the transaction is going on.



Field 7

Field name: Merch state

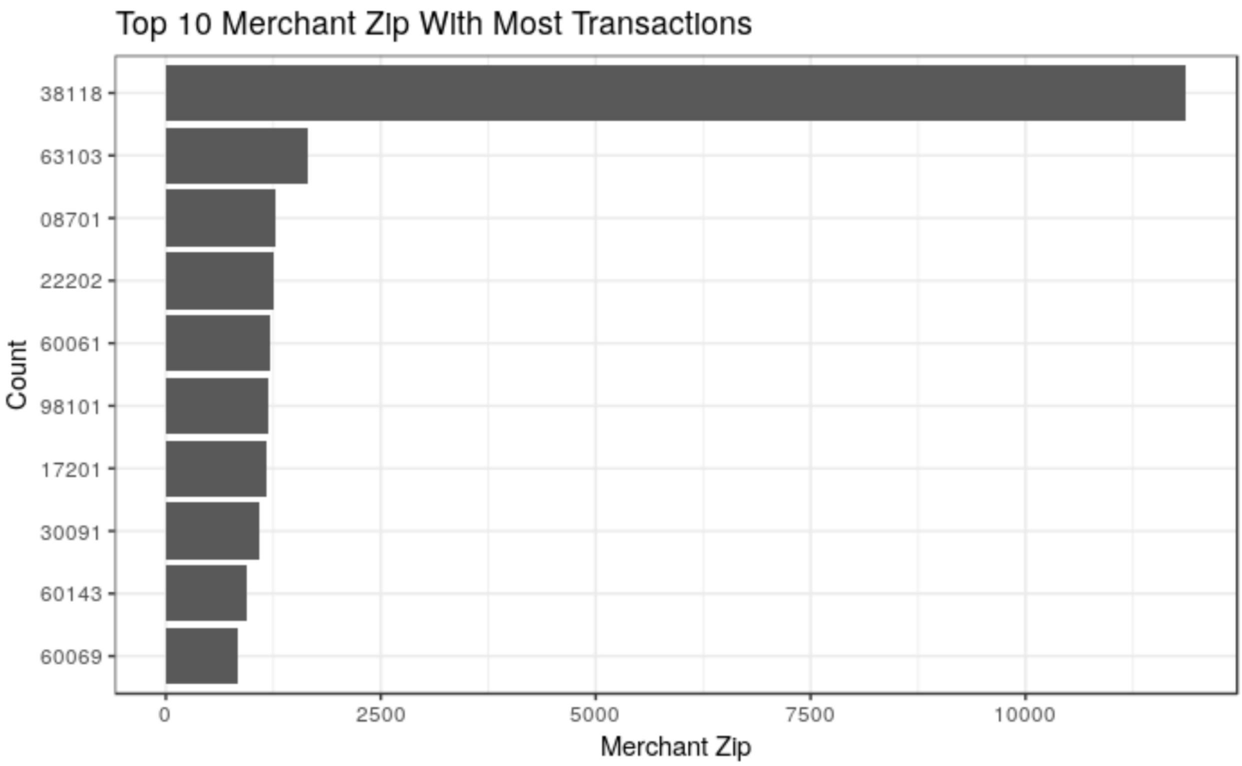
Description: The state of the merchant. (e.g. CA)



Field 8

Field name: Merch zip

Description: The zip code of the merchant. (5 digits)



Field 9

Field name: Transtype

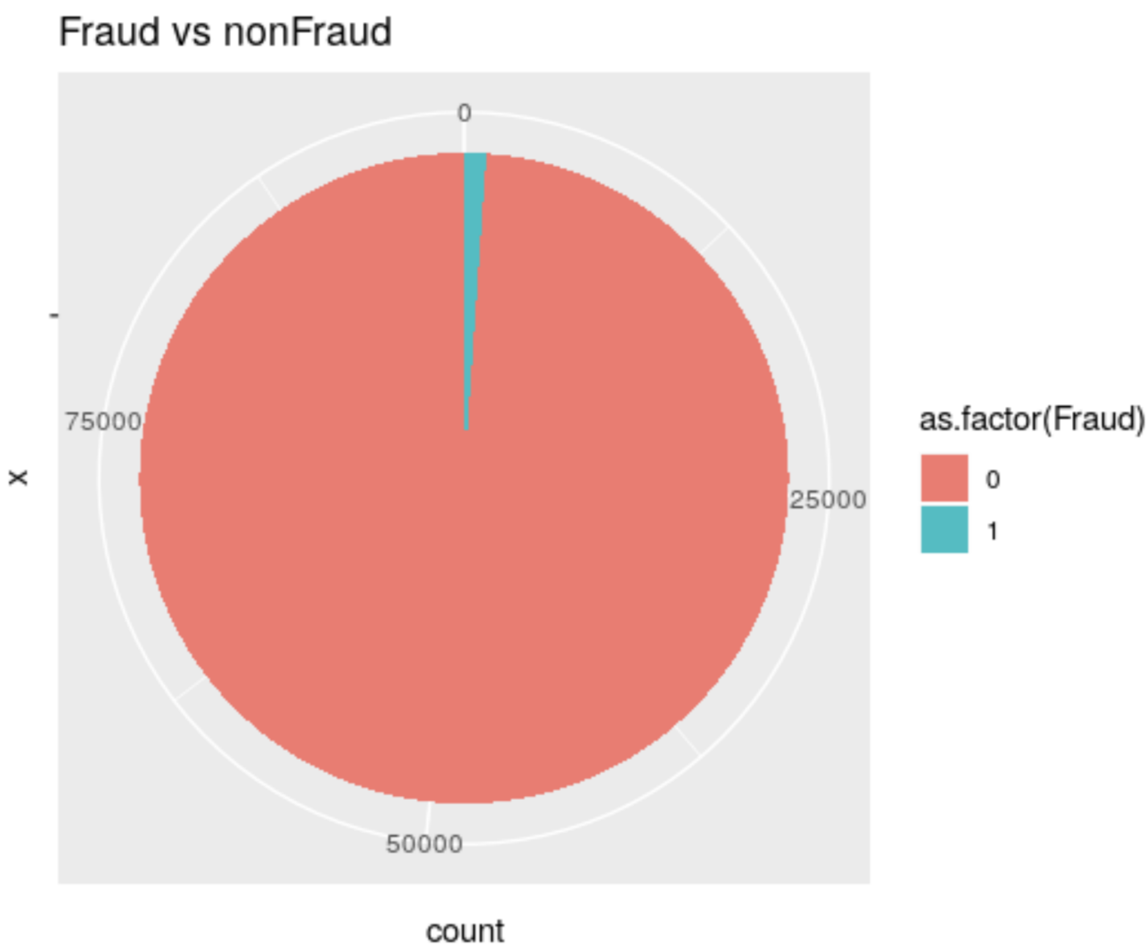
Description: Type of the transaction.

Transtype <chr>	count <int>
P	96398
A	181
D	173
Y	1

Field 10

Field name: Fraud

Description: Whether the transaction is identified as a fraud. (Fraud is flagged as 1)



Appendix 2

List of all variables:

['mean_Cardnum_1d', 'Actual/mean_Cardnum_1d', 'mean_Cardnum_3d', 'Actual/mean_Cardnum_3d', 'mean_Cardnum_7d', 'Actual/mean_Cardnum_7d', 'mean_Cardnum_14d', 'Actual/mean_Cardnum_14d', 'mean_Cardnum_30d', 'Actual/mean_Cardnum_30d', 'max_Cardnum_1d', 'Actual/max_Cardnum_1d', 'max_Cardnum_3d', 'Actual/max_Cardnum_3d', 'max_Cardnum_7d', 'Actual/max_Cardnum_7d', 'max_Cardnum_14d', 'Actual/max_Cardnum_14d', 'max_Cardnum_30d', 'Actual/max_Cardnum_30d', 'median_Cardnum_1d', 'Actual/median_Cardnum_1d', 'median_Cardnum_3d', 'Actual/median_Cardnum_3d', 'median_Cardnum_7d', 'Actual/median_Cardnum_7d', 'median_Cardnum_14d', 'Actual/median_Cardnum_14d', 'median_Cardnum_30d', 'Actual/median_Cardnum_30d', 'sum_Cardnum_1d', 'Actual/sum_Cardnum_1d', 'sum_Cardnum_3d', 'Actual/sum_Cardnum_3d', 'sum_Cardnum_7d', 'Actual/sum_Cardnum_7d', 'sum_Cardnum_14d', 'Actual/sum_Cardnum_14d', 'sum_Cardnum_30d', 'Actual/sum_Cardnum_30d', 'count_Cardnum_1d', 'Actual/count_Cardnum_1d', 'count_Cardnum_3d', 'Actual/count_Cardnum_3d', 'count_Cardnum_7d', 'Actual/count_Cardnum_7d', 'count_Cardnum_14d', 'Actual/count_Cardnum_14d', 'count_Cardnum_30d', 'Actual/count_Cardnum_30d', 'mean_Merchnum_1d', 'Actual/mean_Merchnum_1d', 'mean_Merchnum_3d', 'Actual/mean_Merchnum_3d', 'mean_Merchnum_7d', 'Actual/mean_Merchnum_7d', 'mean_Merchnum_14d', 'Actual/mean_Merchnum_14d', 'mean_Merchnum_30d', 'Actual/mean_Merchnum_30d', 'max_Merchnum_1d', 'Actual/max_Merchnum_1d', 'max_Merchnum_3d', 'Actual/max_Merchnum_3d', 'max_Merchnum_7d', 'Actual/max_Merchnum_7d', 'max_Merchnum_14d', 'Actual/max_Merchnum_14d', 'max_Merchnum_30d', 'Actual/max_Merchnum_30d', 'median_Merchnum_1d', 'Actual/median_Merchnum_1d', 'median_Merchnum_3d', 'Actual/median_Merchnum_3d', 'median_Merchnum_7d', 'Actual/median_Merchnum_7d', 'median_Merchnum_14d', 'Actual/median_Merchnum_14d', 'median_Merchnum_30d', 'Actual/median_Merchnum_30d', 'sum_Merchnum_1d', 'Actual/sum_Merchnum_1d', 'sum_Merchnum_3d', 'Actual/sum_Merchnum_3d', 'sum_Merchnum_7d', 'Actual/sum_Merchnum_7d', 'sum_Merchnum_14d', 'Actual/sum_Merchnum_14d', 'sum_Merchnum_30d', 'Actual/sum_Merchnum_30d', 'count_Merchnum_1d', 'Actual/count_Merchnum_1d', 'count_Merchnum_3d', 'Actual/count_Merchnum_3d', 'count_Merchnum_7d', 'Actual/count_Merchnum_7d', 'count_Merchnum_14d', 'Actual/count_Merchnum_14d', 'count_Merchnum_30d', 'Actual/count_Merchnum_30d', 'mean_Cardnum_Merchnum_1d', 'Actual/mean_Cardnum_Merchnum_1d', 'mean_Cardnum_Merchnum_3d', 'Actual/mean_Cardnum_Merchnum_3d', 'mean_Cardnum_Merchnum_7d', 'Actual/mean_Cardnum_Merchnum_7d', 'mean_Cardnum_Merchnum_14d', 'Actual/mean_Cardnum_Merchnum_14d', 'mean_Cardnum_Merchnum_30d', 'Actual/mean_Cardnum_Merchnum_30d', 'max_Cardnum_Merchnum_1d', 'Actual/max_Cardnum_Merchnum_1d', 'max_Cardnum_Merchnum_3d', 'Actual/max_Cardnum_Merchnum_3d', 'max_Cardnum_Merchnum_7d', 'Actual/max_Cardnum_Merchnum_7d', 'max_Cardnum_Merchnum_14d', 'Actual/max_Cardnum_Merchnum_14d', 'max_Cardnum_Merchnum_30d', 'Actual/max_Cardnum_Merchnum_30d', 'median_Cardnum_Merchnum_1d', 'Actual/median_Cardnum_Merchnum_1d', 'median_Cardnum_Merchnum_3d', 'Actual/median_Cardnum_Merchnum_3d', 'median_Cardnum_Merchnum_7d', 'Actual/median_Cardnum_Merchnum_7d', 'median_Cardnum_Merchnum_14d', 'Actual/median_Cardnum_Merchnum_14d', 'median_Cardnum_Merchnum_30d', 'Actual/median_Cardnum_Merchnum_30d', 'sum_Cardnum_Merchnum_1d', 'Actual/sum_Cardnum_Merchnum_1d', 'sum_Cardnum_Merchnum_3d', 'Actual/sum_Cardnum_Merchnum_3d', 'sum_Cardnum_Merchnum_7d', 'Actual/sum_Cardnum_Merchnum_7d', 'sum_Cardnum_Merchnum_14d', 'Actual/sum_Cardnum_Merchnum_14d', 'sum_Cardnum_Merchnum_30d', 'Actual/sum_Cardnum_Merchnum_30d', 'count_Cardnum_Merchnum_1d', 'Actual/count_Cardnum_Merchnum_1d', 'count_Cardnum_Merchnum_3d', 'Actual/count_Cardnum_Merchnum_3d', 'count_Cardnum_Merchnum_7d', 'Actual/count_Cardnum_Merchnum_7d', 'count_Cardnum_Merchnum_14d', 'Actual/count_Cardnum_Merchnum_14d', 'count_Cardnum_Merchnum_30d', 'Actual/count_Cardnum_Merchnum_30d', 'mean_Cardnum_Merch zip_1d', 'Actual/mean_Cardnum_Merch zip_1d', 'mean_Cardnum_Merch zip_3d', 'Actual/mean_Cardnum_Merch zip_3d', 'mean_Cardnum_Merch zip_7d', 'Actual/mean_Cardnum_Merch zip_7d', 'mean_Cardnum_Merch zip_14d', 'Actual/mean_Cardnum_Merch zip_14d', 'mean_Cardnum_Merch zip_30d', 'Actual/mean_Cardnum_Merch zip_30d', 'max_Cardnum_Merch zip_1d', 'Actual/max_Cardnum_Merch zip_1d', 'max_Cardnum_Merch zip_3d', 'Actual/max_Cardnum_Merch zip_3d', 'max_Cardnum_Merch zip_7d', 'Actual/max_Cardnum_Merch zip_7d', 'max_Cardnum_Merch zip_14d', 'Actual/max_Cardnum_Merch zip_14d', 'max_Cardnum_Merch zip_30d', 'Actual/max_Cardnum_Merch zip_30d', 'median_Cardnum_Merch zip_1d', 'Actual/median_Cardnum_Merch zip_1d', 'median_Cardnum_Merch zip_3d', 'Actual/median_Cardnum_Merch zip_3d', 'median_Cardnum_Merch zip_7d', 'Actual/median_Cardnum_Merch zip_7d', 'median_Cardnum_Merch zip_14d', 'Actual/median_Cardnum_Merch zip_14d', 'median_Cardnum_Merch zip_30d', 'Actual/median_Cardnum_Merch zip_30d', 'sum_Cardnum_Merch zip_1d', 'Actual/sum_Cardnum_Merch zip_1d', 'sum_Cardnum_Merch zip_3d', 'Actual/sum_Cardnum_Merch zip_3d', 'sum_Cardnum_Merch zip_7d', 'Actual/sum_Cardnum_Merch zip_7d', 'sum_Cardnum_Merch zip_14d', 'Actual/sum_Cardnum_Merch zip_14d', 'sum_Cardnum_Merch zip_30d', 'Actual/sum_Cardnum_Merch zip_30d', 'count_Cardnum_Merch zip_1d', 'Actual/count_Cardnum_Merch zip_1d', 'count_Cardnum_Merch zip_3d', 'Actual/count_Cardnum_Merch zip_3d', 'count_Cardnum_Merch zip_7d', 'Actual/count_Cardnum_Merch zip_7d', 'count_Cardnum_Merch zip_14d', 'Actual/count_Cardnum_Merch zip_14d', 'count_Cardnum_Merch zip_30d', 'Actual/count_Cardnum_Merch zip_30d', 'mean_Cardnum_Merch state_1d', 'Actual/mean_Cardnum_Merch state_1d', 'mean_Cardnum_Merch state_3d', 'Actual/mean_Cardnum_Merch state_3d', 'mean_Cardnum_Merch state_7d', 'Actual/mean_Cardnum_Merch state_7d', 'mean_Cardnum_Merch state_14d', 'Actual/mean_Cardnum_Merch state_14d', 'mean_Cardnum_Merch state_30d', 'Actual/mean_Cardnum_Merch state_30d', 'max_Cardnum_Merch state_1d', 'Actual/max_Cardnum_Merch state_1d', 'max_Cardnum_Merch state_3d', 'Actual/max_Cardnum_Merch state_3d', 'max_Cardnum_Merch state_7d', 'Actual/max_Cardnum_Merch state_7d', 'max_Cardnum_Merch state_14d', 'Actual/max_Cardnum_Merch state_14d', 'max_Cardnum_Merch state_30d', 'Actual/max_Cardnum_Merch state_30d', 'median_Cardnum_Merch state_1d', 'Actual/median_Cardnum_Merch state_1d', 'median_Cardnum_Merch state_3d', 'Actual/median_Cardnum_Merch state_3d', 'median_Cardnum_Merch state_7d', 'Actual/median_Cardnum_Merch state_7d', 'median_Cardnum_Merch state_14d', 'Actual/median_Cardnum_Merch state_14d', 'median_Cardnum_Merch state_30d', 'Actual/median_Cardnum_Merch state_30d', 'sum_Cardnum_Merch state_1d', 'Actual/sum_Cardnum_Merch state_1d', 'sum_Cardnum_Merch state_3d', 'Actual/sum_Cardnum_Merch state_3d', 'sum_Cardnum_Merch state_7d', 'Actual/sum_Cardnum_Merch state_7d', 'sum_Cardnum_Merch state_14d', 'Actual/sum_Cardnum_Merch state_14d', 'sum_Cardnum_Merch state_30d', 'Actual/sum_Cardnum_Merch state_30d', 'count_Cardnum_Merch state_1d', 'Actual/count_Cardnum_Merch state_1d', 'count_Cardnum_Merch state_3d', 'Actual/count_Cardnum_Merch state_3d', 'count_Cardnum_Merch state_7d', 'Actual/count_Cardnum_Merch state_7d', 'count_Cardnum_Merch state_14d', 'Actual/count_Cardnum_Merch state_14d', 'count_Cardnum_Merch state_30d', 'Actual/count_Cardnum_Merch state_30d', 'Days_since_per_Cardnum', 'Days_since_per_Merchnum', 'Days_since_per_Cardnum_Merch zip', 'Days_since_per_Cardnum_Merch state']