Credit Card Transaction Fraud Analysis

DSO562 Team 7

Runxue Liang, Danyi Wang
Jiaxiulin Liu, Yuting Ma
Yixiao Ma, Xiaoya Li

Table of Contents

I. Executive Summary

Credit cards nowadays has been one of the major payment methods in people's lives. Yet as transaction payments get more advanced, various kinds of credit card frauds start to come into existence, including lost or stolen cards, counterfeit cards, and online account hacking. Since banks are responsible for most of the losses, credit card fraud poses great threat to banks, placing them as victims of fraudulent transactions. Therefore, it's essential for banks to identify possible fraud when authorizing credit card transactions. Our project here mainly aims to build a supervised fraud algorithm to detect credit card transaction fraud from a raw data set of 100,000 transaction records, with over 1,000 fraudulent transactions.

After data cleaning and processing, we initially created 371 variables using different combinations of fields and time windows. Then we performed feature selection procedure to select the most important and relevant variables: we first used both Kolmogorov–Smirnov (KS) score and Fraud Detection Rate (FDR) to filter out half of the variables, and then used one of the wrapper methods, Forward Sequential Feature Selection, to select top 15 variables.

When the data and variables are well-prepared, we divided the data into training and testing dataset and out-of-time dataset to construct the model. We performed four machine learning methods: Logistic Regression, Random Forest, Neural Network, and Naive Bayes. For each method, we fit the model several times and average the result to get more stable estimates on both training and testing dataset. Comparing the performance of different models, we selected Random Forest as our finalized model. Then we used the model to validate the out-of-time dataset. The FDR at 3% of our final model for out-of-time validation dataset is 46.37%, indicating a very efficient fraud detection result.

II. Description of Data

1. Data Description

This dataset provides the information about credit card transactions for the year 2010. The dataset contains 96,753 credit card transaction records and 10 fields including information on transaction identifier, card number, transaction date and type, merchant information and label of whether the transaction is fraudulent. There are 8 categorical fields: Cardnum, Date, Merchnum, Merch description, Merch state, Merch zip, Transtype and Fraud, and one numerical field: Amount.

2. Summary Table

Data Summary of Categorical Values

#	Field	%Populated	#Unique	Most common	Occurrence of Most Common Category
1	Recnum	100.0%	96753/96752	/	1
2	Cardnum	100.0%	1645/96752	5142148452	1192
3	Date	100.0%	365/96752	2010-02-28	684
4	Merchnum	96.51%	13092/93378	930090121224	9310
5	Merch description	100.0%	13125/96752	GSA-FSS-ADV	1688
6	Merch state	98.76%	228/95558	TN	12035
7	Merch zip	95.19%	4568/92097	38118.0	11868
8	Transtype	100.0%	4/96752	P	96397
9	Fraud	100.0%	2/96752	0	95693

Data Summary of Numerical Values

#	Field	Min	Max	Average	Median	SD	%Populated	# of 0	#unique
1	Amount	0.01	47,900.0	395.83	137.975	831.88	100%	0	34908/96752

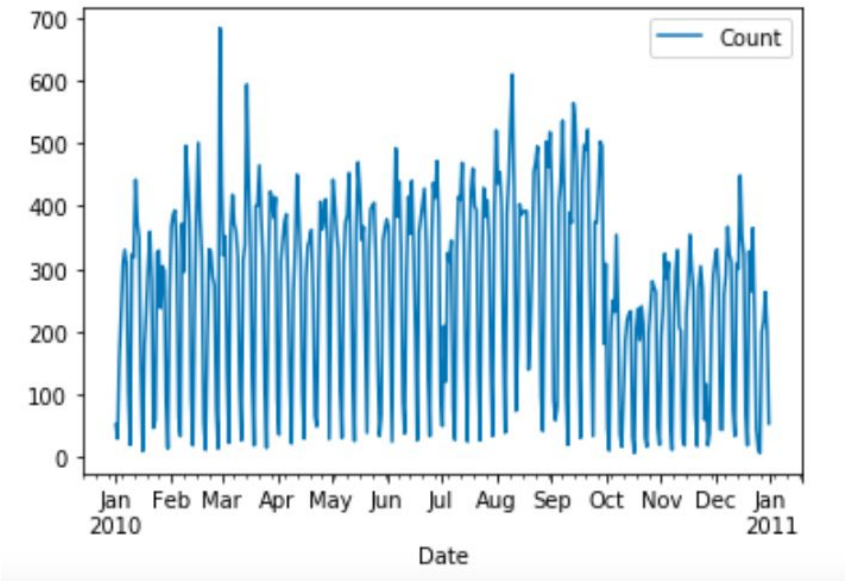
3. Important Fields

1) Field Name: Date

Description: Date, a categorical variable, representing the date on which the transaction happened. Date has 96,752 lines of records and is 100.0% populated. Date has 365 categories each is one day in the year of 2010. The most common category is 2010-02-28, when most number of transactions occurred, occurred 684 times out of 96,752 records.

Field	Number of Records	Populated %	Unique Value	Most Common Category	Occurrence of Common Category
Date	96752	100.0	365	2010-02-28	684

Below is a graph showing number of records of each date:

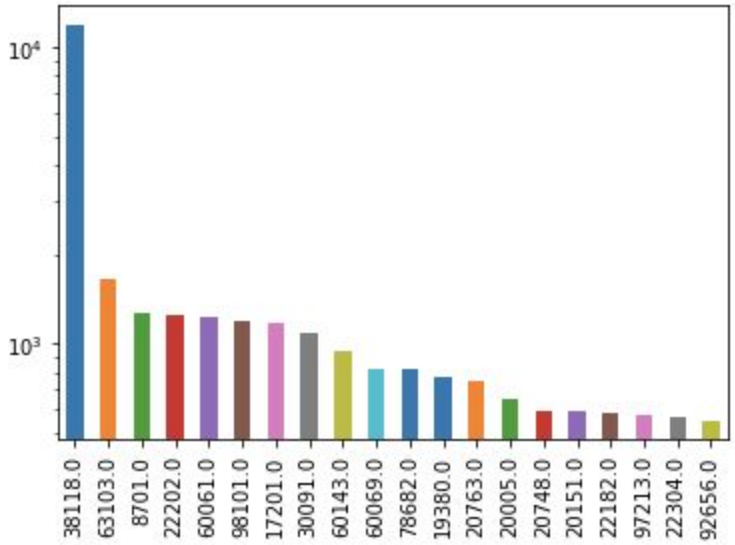


2) Field Name: Merch zip

Description: Merch zip is a categorical variable representing the zip code for merchant. Merch zip has 92,097 lines of records and is 95.19% populated. Merch zip has 4,568 categories. The most common category is 38118.0, occurred 11,868 times out of 92,097 records.

Field	Number of Records	Populated %	Unique Value	Most Common Category	Occurrence of Common Category
Merch zip	92097	95.19	4568	38118.0	11868

Below is a graph showing the most common 20 Merch zip:



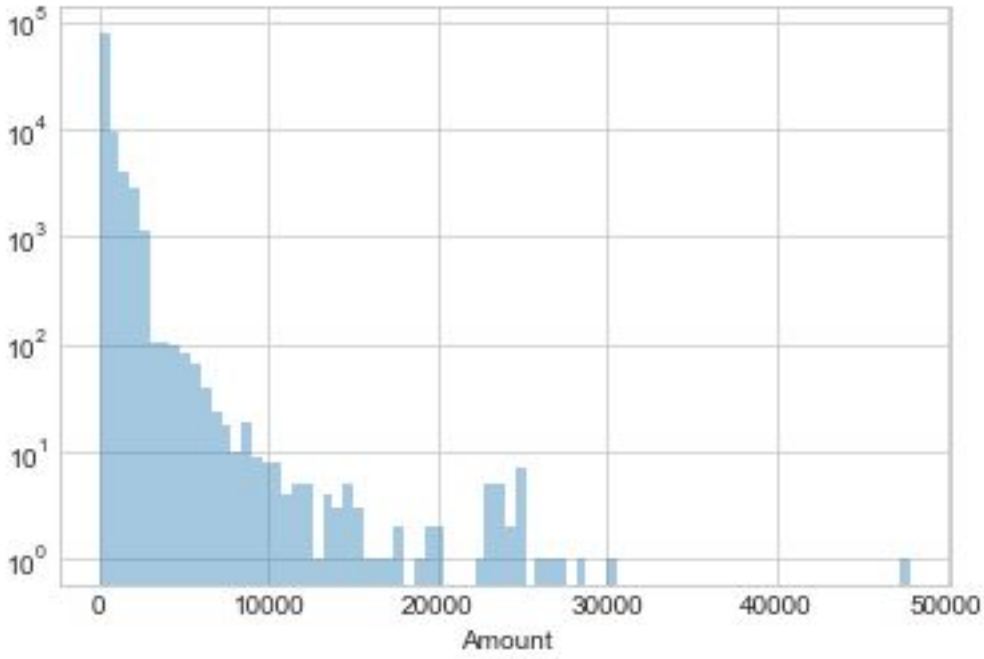
3) Field Name: Fraud

Description: Fraud is a binary variable representing fraud commission by 1, and no fraud by 0. Fraud has 96,752 lines of records and is 100.0% populated. Fraud has 2 categories. The most common category is 0, which represent non-fraudulent card transaction, occurred 95,693 times out of 96,752 records. While fraudulent card transaction, which is represented by 1, occurred 1,059 times, about 1.09% of all records.

4) Field Name: Amount

Description: Amount is a numerical variable indicating the transaction amounts in dollars. For field Amount, there are 34,908 unique values, and 100% of the records are populated. The distribution plot and the statistical summary of the field are shown below: (excluding an outlier with value of more than 3 million)

Field	Min	Max	Average	Median	SD	%Populated	# of 0	#unique
Amount	0.01	47,900.0	395.83	137.975	831.88	100%	0	34908/96752



III. Data Cleaning

1. Exclusions and Outliers:

The team focused only on the transaction type of "P", and excluded the transaction type of "A", "D", and "Y". We also excluded the outlier record with transaction amount of $3,102,406.

2. Filling Missing Values:

a. We first filled in missing values according to the relationship between the fields. We filled in Merch State from Merch Zip using US zip code package.

b. We then filled in missing values in Merchnum. We filled in the missing values from the most common Merchnum with the same Merch Description and Merch Zip. For the Merchnum that are not filled, we filled them with the most common value grouped by Merch Description and Merch State. For the Merchnum that are still not filled, we continued to fill them with the most common value grouped by Merch Description. At this step, we have completed filling in Merchnum for the records with Merch Description corresponding to at least one Merchnum.

We noticed that some Merch Description groups do not have any corresponding Merchnum. Given no related information to fill in Merchnum, we created new unique Merchnum for each of these Merch Description and assigned the Merchnum to the corresponding records.

c. We then filled in missing values in Merch Zip. We filled in the missing values with the most common Merch Zip grouped by Merch Description and Merch State. Since Merch Description is 100% populated, the records that are not yet filled are those with both

Merch Zip and Merch State missing. We noticed that these records can be divided into two groups: special transactions including fees and adjustments, and transactions at a foreign merchant. We proceeded with different approaches for the two groups. For the special transactions, we filled in the most common Merch State and Merch Zip grouped by Cardnum. For foreign transactions, we assigned these records with Merch State of "Other" and Merch Zip of "00000", indicating their foreign locations.

IV. Candidate Variables

To precisely predict fraud, we constructed 371 expert variables as candidates for the model-building process. The variables we built can be divided into 4 categories—amount variables, frequency variables, day since variables and velocity variables.

1. Amount Variables

Amount variables are constructed to detect the unusual amount of transactions occurred by cards, merchants, or geographic locations over different time windows. Therefore, we created 240 amount variables by calculating 8 measurements of Amount variables by 5 different groups over 6 time-windows:

Amount	**Groups**	**Time Windows**
Average	Card	0 Days
Maximum	Merchant	1 Day
Median	Card at this merchant	3 Days
Total	Card in this zip code	7 Days
Actual / Average of the Group	Card in this state	14 Days
Actual / Maximum of the Group		30 Days
Actual / Median of the Group		
Actual / Total of the Group		

2. Frequency Variables

Fraud can also be detected through the unusual frequency of transaction. Hence, we calculated the number of transactions by 5 groups over 6 time-windows.

Number of Transactions	**Groups**	**Time Windows**
	Card	0 Days
	Merchant	1 Day
	Card at this merchant	3 Days
	Card in this zip code	7 Days
	Card in this state	14 Days
		30 Days

3. Days Since Variables

The days since variables evaluate how long it took since today and the most frequent transaction with the same groups (Card, Merchant, Card at this Merchant, Card in this zip code, Card in this state).

4. Velocity Variables

We built variables to measure the velocity change of card transactions as well. We measured the velocity by comparing a type of transaction measurement over a recent time period over a longer time period.

The numerator is the number or amount of transactions by card or merchant, over past 0 or 1 days.

Measurement	**Group**	Time Windows
Number of transactions	Card	0 days
Amount of transactions	Merchant	1 day

The denominator is the average daily number or amount of transactions by the same card or merchant over past 7, 14 and 30 days.

Measurement	**Group**	Time Windows
Number of transactions	Card	7 days

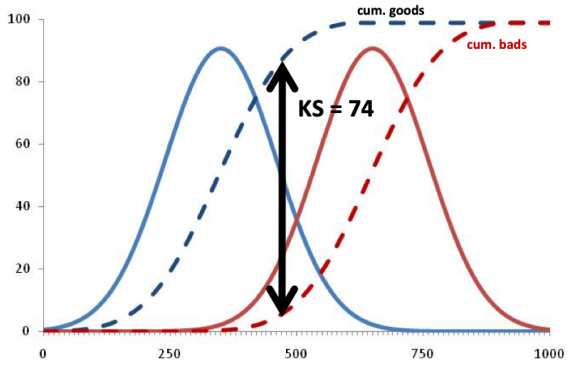
Amount of transactions	Merchant	14 days
		30 days

V. Feature Selection Process

With candidate variables built, we conducted feature selection to reduce dimensionality and reduce the variables used in the modeling process. The feature selection process consists of two steps:

1. Filter Feature Selection

First, we used Kolmogorov-Smirnov (KS) and Fraud Detection Rate (FDR) as our filter and left about half of the variables for the second step. KS is a measurement of how well two distributions are seperated. In our process, for each candidate variable, we calculated KS using scipy.stats. The larger the KS, the more different this variable value is when a record is a fraud compared to when it's not a fraud. Here is a graph illustrating this:

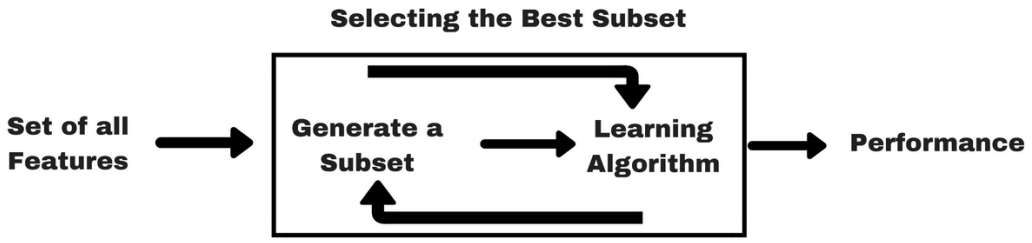


Also, for each variable, we computed the FDR at 3%. FDR at 3% is calculated as the percent of total frauds included in the top 3% population ranked by variable value. Intuitively, it shows how much proportion of frauds are captured by this variable. The higher the FDR, the more effective this variable is in capturing frauds.

After calculating KS and FDR for each candidate variable, we conducted quantile binning on both of the measurements and assigned rank orders to each candidate variable. Then we averaged the two rank orders to assign each candidate variable a combined rank order. The most significant variables indicated by KS and FDR are ranked the top. Using this filter, we kept only 180 top ranked variables for further selection.

2. Wrapper Feature Selection

Second, we did a wrapper feature selection on the 180 candidate variables we selected above. We chose to do a forward sequential feature selection. A forward feature selection is a process in which the algorithm starts from logistic regression models with no variable, adds one optimal variable at a time, until completes the selection. The algorithm constructs simple logistic regression models with one variable for each candidate variable, compares the performances of these models, and chooses the variable that has the best model performance. Including this variable into the model, the algorithm continues to add the second variable to the model, and selects the variable that gives the best performance combined with the previous variable. We repeated this process until the top 15 variables were selected. Here is a graph illustrating our forward selection process:



We used a wrapper feature selection is for two reasons. First, this process reduces linear correlation between selected variables. Each variable is selected by their contribution to the model given previous variables. In this situation, highly correlated variables are less likely to be selected since one does not give much more information than the other. Thus, the variables selected will not be linear-correlated and each of them will contribute to our models differently. Second, this process decreases the number of variables and leaves the most important variables. With fewer variables, non-linear machine learning models are able to perform better.

After these two steps, we have selected 15 variables to be used in our models. Here is a list of these variables:

1	Median__Cardnum_1
2	Total__Cardnum_1
3	Max__Cardnum_7
4	Average__Cardnum_14
5	Median__Cardnum_30
6	Total__Cardnum_30
7	Total__Cardnum_Merchnum_14

8	Max__Cardnum_Merchnum_30
9	Total__Cardnum_Merchnum_30
10	Max__Cardnum_Merch zip_1
11	Average__Cardnum_Merch zip_14
12	Total__Cardnum_Merch zip_30
13	Actual/Total__Cardnum_Merchnum_1
14	Actual/Median__Merchnum_14
15	Total__Cardnum_1Count__Cardnum_14

VI. Model Algorithms

With fraud detection model, we aimed to predict the likelihood of a credit card transaction being a fraud. When building the models, we used variables selected in the feature selection as predictor variables, and fraud label as response variable. We left out the records after 2010-10-31 as out-of-time validation dataset, and randomly separated the rest of the records into 70% training set and 30% testing dataset.

We tried four different types of model to determine the optimal model: Logistic Regression, Random Forest, Neural Network, and Naive Bayes, and used FDR at 3% to compare the performance of the models. Comparing these four types of models, the optimal model is Random Forest model with 1000 trees and maximum depth of 18. Below is the detail of the models we tried and the model performances.

1. Logistic regression

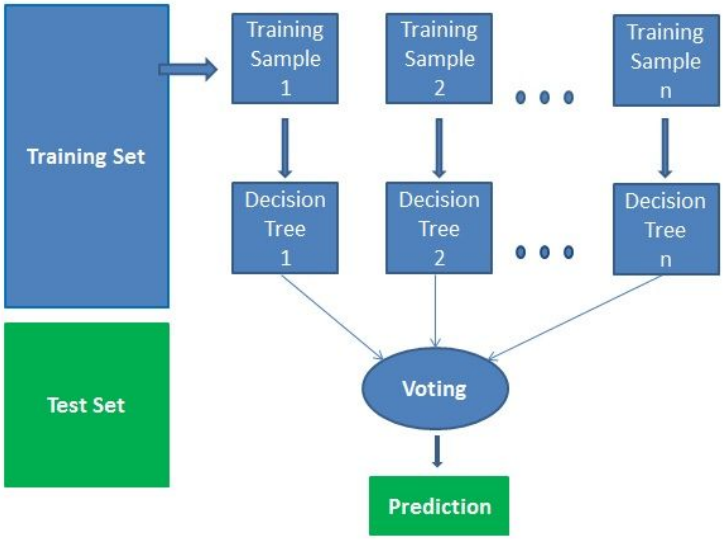
We first conducted Logistic Regression and used it as a baseline to judge other model's successes. Logistic Regression uses the logistic function to model a binary response variable, and it models log product of the odd (probability of an event happen / probability of an event not happen) as linear relationship to predictor variables. To train a Logistic Regression model, algorithm will estimate the coefficient for each predictor variable. The final output is the probability corresponding to the odd, and is bounded between 0 and 1.

With fraud label as the response variable, it is natural to start with Logistic Regression since it is suitable for predicting binary output. Running the logistic regression on each split dataset, the model has FDR at 3% of 55% in the training set, 54.58% in the testing set, and 39.11% in the out-of-time validation set.

2. Random Forest

Then we tried Random Forest. Random Forest is an ensemble method of decision trees generated on a randomly split dataset. In classification problem, each tree votes and the class with most votes becomes the final result. Random Forest is considered a highly accurate and robust method because there are more decision trees in the process. One disadvantage is that it is time-consuming to generate predictions and harder to interpret than simple decision tree method. The process is to first select random samples from the dataset. And then construct a decision tree for each sample and get a prediction result from each tree. After that, it votes for each predicted result and select the one with most votes as the final prediction. The process chart is shown below.

In this case, after several attempts, we set the number of trees to be 1000, and the maximum depth to be 18. The final FDR at 3% for training, testing, and out-of-time dataset are 100%, 86.67%, and 46.37%. It shows that Random Forest has low variance while maintaining a relatively low bias. And FDR at 3% for out-of-time validation set, which is our criteria here, shows it outperforms all the other models.



3. Neural Network

We also tried fitting a Neural Network model. Neural Network maps the links from the inputs to the output. It contains input layer with all independent variables, output layer with dependent variable, and hidden layers of nodes between inputs and output. In each layer, each node takes the weighted average of all nodes from previous layer and adds a transfer function before feeding the data to the next layer. To train the Neural Network, show the model one record at a time, calculate the error to adjust the weights in each layer, and iterate through the whole dataset multiple times until finding the local optimal for the weights. To initialize a Neural Network, user needs to specify parameters including number of layers between inputs and output, and number of nodes in each layer.

In our case, the inputs for the Neural Network are the selected variables, while the output is the fraud label. The team tried different combinations of nodes in up to 2 hidden layers, and selected 1 hidden layer with 7 nodes as the optimal model. With the optimal model, the FDR at 3% is 30.77% for the training set, 30.85% for the testing set, and 39.11% for the out-of-time validation set.

4. Naive Bayes

The last model we tried was Naive Bayes. It is a classification method based on Bayes' Theorem. This is a generative model where for the given features (x) and the label (y), Naive Bayes estimates a joint probability from the training data. It assumes all the features are conditionally independent. So, if some of the features are dependent on each other, the prediction might be poor.

In this case, we simply fitted the feature variables and Fraud variable into Gaussian Naive Bayes model and called out the probabilities. The final Naive Bayes model has the FDR at 3% of 58.44% in the training set, 57.08% in the testing set, and 32.40% in the out-of-time validation set.

The below table summarises the performance in terms of FDR at 3% for the models we tried:

	FDR @ 3%		
	Training	**Testing**	**Out of Time**
Logistic Regression	55	54.5833	39.1061
Random Forest	100	86.6667	46.3687
Neural Net	30.7692	30.8511	39.1061
Naïve Bayes	58.4375	57.0833	32.4022

VII. Results

From the table in the last section, Random Forest has the highest 3% FDR for training, testing and out-of-time data. As a matter of fact, if we classify those with probability over 0.5 as frauds, Random Forest also has the highest prediction accuracy for out-of-time data among the four models. Thus, we picked Random Forest as our final model and the parameter we tried that worked best was depth of 18, which stands for 18 terminal nodes for each tree, and 1000 trees.

Final Model - Random Forest

Training	#Records	#Bads	#Goods	Fraud Rate
	58779	640	58139	0.010888

Population Bin%	Bin Statistics					Culmulative Statistics						
	# Records	# Bads	# Goods	% Bads	% Goods	Total #Records	Culmulative #Bads	Culmulative #Goods	Culmulative %Bads	Culmulative %Goods	KS	FPR
1	588	588	0	100	0	588	588	0	91.88	0.00	91.88	0.00
2	588	52	536	8.84	91.16	1176	640	536	100	0.92	99.08	0.84
3	588	0	588	0	100	1764	640	1124	100	1.93	98.07	1.76
4	588	0	588	0	100	2352	640	1712	100	2.94	97.06	2.68
5	587	0	587	0	100	2939	640	2299	100	3.95	96.05	3.59
6	588	0	588	0	100	3527	640	2887	100	4.97	95.03	4.51
7	588	0	588	0	100	4115	640	3475	100	5.98	94.02	5.43
8	588	0	588	0	100	4703	640	4063	100	6.99	93.01	6.35
9	588	0	588	0	100	5291	640	4651	100	8.00	92.00	7.27
10	587	0	587	0	100	5878	640	5238	100	9.01	90.99	8.18
11	588	0	588	0	100	6466	640	5826	100	10.02	89.98	9.10
12	588	0	588	0	100	7054	640	6414	100	11.03	88.97	10.02
13	588	0	588	0	100	7642	640	7002	100	12.04	87.96	10.94
14	587	0	587	0	100	8229	640	7589	100	13.05	86.95	11.86
15	588	0	588	0	100	8817	640	8177	100	14.06	85.94	12.78
16	588	0	588	0	100	9405	640	8765	100	15.08	84.92	13.70
17	588	0	588	0	100	9993	640	9353	100	16.09	83.91	14.61
18	588	0	588	0	100	10581	640	9941	100	17.10	82.90	15.53
19	587	0	587	0	100	11168	640	10528	100	18.11	81.89	16.45
20	588	0	588	0	100	11756	640	11116	100	19.12	80.88	17.37

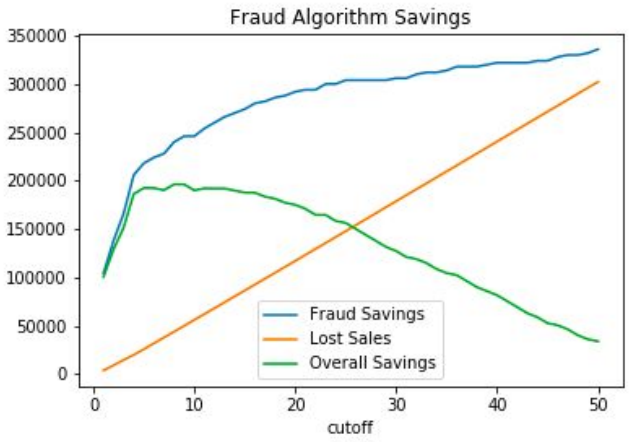
Testing	#Records	#Bads	#Goods	Fraud Rate
	25191	240	24951	0.009527

Population Bin%	Bin Statistics					Culmulative Statistics						
	# Records	# Bads	# Goods	% Bads	% Goods	Total #Records	Culmulative #Bads	Culmulative #Goods	Culmulative %Bads	Culmulative %Goods	KS	FPR
1	252	166	86	65.87	34.13	252	166	86	69.17	0.34	68.82	0.52
2	252	33	219	13.10	86.90	504	199	305	82.92	1.22	81.69	1.53
3	252	9	243	3.57	96.43	756	208	548	86.67	2.20	84.47	2.63
4	252	7	245	2.78	97.22	1008	215	793	89.58	3.18	86.41	3.69
5	252	1	251	0.40	99.60	1260	216	1044	90.00	4.18	85.82	4.83
6	252	4	248	1.59	98.41	1512	220	1292	91.67	5.18	86.49	5.87
7	252	3	249	1.19	98.81	1764	223	1541	92.92	6.18	86.74	6.91
8	252	1	251	0.40	99.60	2016	224	1792	93.33	7.18	86.15	8.00
9	252	0	252	0.00	100.00	2268	224	2044	93.33	8.19	85.14	9.13
10	251	0	251	0.00	100.00	2519	224	2295	93.33	9.20	84.14	10.25
11	252	0	252	0.00	100.00	2771	224	2547	93.33	10.21	83.13	11.37
12	252	2	250	0.79	99.21	3023	226	2797	94.17	11.21	82.96	12.38
13	252	2	250	0.79	99.21	3275	228	3047	95.00	12.21	82.79	13.36
14	252	1	251	0.40	99.60	3527	229	3298	95.42	13.22	82.20	14.40
15	252	0	252	0.00	100.00	3779	229	3550	95.42	14.23	81.19	15.50
16	252	1	251	0.40	99.60	4031	230	3801	95.83	15.23	80.60	16.53
17	252	0	252	0.00	100.00	4283	230	4053	95.83	16.24	79.59	17.62
18	252	0	252	0.00	100.00	4535	230	4305	95.83	17.25	78.58	18.72
19	252	1	251	0.40	99.60	4787	231	4556	96.25	18.26	77.99	19.72
20	251	1	250	0.40	99.60	5038	232	4806	96.67	19.26	77.40	20.72

Out of Time	#Records	#Bads	#Goods	Fraud Rate
	12427	179	12248	0.014404

Population Bin%	Bin Statistics					Culmulative Statistics						
	# Records	# Bads	# Goods	% Bads	% Goods	Total #Records	Culmulative #Bads	Culmulative #Goods	Culmulative %Bads	Culmulative %Goods	KS	FPR
1	125	52	73	41.60	58.40	125	52	73	29.05	0.60	28.45	1.40
2	124	17	107	13.71	86.29	249	69	180	38.55	1.47	37.08	2.61
3	124	14	110	11.29	88.71	373	83	290	46.37	2.37	44.00	3.49
4	125	20	105	16.00	84.00	498	103	395	57.54	3.23	54.32	3.83
5	124	6	118	4.84	95.16	622	109	513	60.89	4.19	56.71	4.71
6	124	3	121	2.42	97.58	746	112	634	62.57	5.18	57.39	5.66
7	124	2	122	1.61	98.39	870	114	756	63.69	6.17	57.51	6.63
8	125	6	119	4.80	95.20	995	120	875	67.04	7.14	59.90	7.29
9	124	3	121	2.42	97.58	1119	123	996	68.72	8.13	60.58	8.10
10	124	0	124	0.00	100.00	1243	123	1120	68.72	9.14	59.57	9.11
11	124	4	120	3.23	96.77	1367	127	1240	70.95	10.12	60.83	9.76
12	125	3	122	2.40	97.60	1492	130	1362	72.63	11.12	61.51	10.48
13	124	3	121	2.42	97.58	1616	133	1483	74.30	12.11	62.19	11.15
14	124	2	122	1.61	98.39	1740	135	1605	75.42	13.10	62.31	11.89
15	124	2	122	1.61	98.39	1864	137	1727	76.54	14.10	62.44	12.61
16	125	3	122	2.40	97.60	1989	140	1849	78.21	15.10	63.12	13.21
17	124	1	123	0.81	99.19	2113	141	1972	78.77	16.10	62.67	13.99
18	124	2	122	1.61	98.39	2237	143	2094	79.89	17.10	62.79	14.64
19	124	1	123	0.81	99.19	2361	144	2217	80.45	18.10	62.35	15.40
20	125	2	123	1.60	98.40	2486	146	2340	81.56	19.11	62.46	16.03

In training data, Random Forest successfully picks out all the frauds in top 2%. The 3% FDR for testing and out-of-time data are 86.67% and 46.37%. Then we look at the fraud savings and lost under possible score cutoffs. Assuming that the savings of catching a fraud is 2000 and the lost for every false positive is 50, with step of 1% it appears 5% and 8% are optimal cutoff options for this data, resulting in 192350 overall savings. Therefore, for this particular dataset, calling top 8% records fraud (to be on the safe side) is the most profitable practice.



VIII. Conclusions

Our project aims to build supervised fraud models on credit card transaction data in 2010 to identify frauds in transaction data.

First, we generated data quality report for all 10 fields of 96,752 rows of data, excluding one outliers with unusually high transaction amount. We provided statistical summary as well as distribution or other plots for each field. Based on the data quality report, we filled the missing values and created 371 variables in preparation of building models. We filtered out half of the variables according to the average rank order of their FDR score and KS score. We then cherry-picked 15 variables after doing forward selection. The models we built are Logistic Regression, Random Forest, Neural Network and Naive Bayes. We fed the training data to our learning algorithm to learn a model and predicted the labels of our test data. For each of these machine learning methods, we ran it several times and took average to get a more robust result. Comparing different models, we chose Random Forest as our final model. We then used the out-of-time dataset to see the performance. The final 3% FDR for out-of-time data is 46.37%.

If we were given more time to work on this project, we would make improvement in the following aspects:

1. We would perform more machine learning methods on the dataset. We will take more models into consideration, such as Support Vector Machine, Boosted Trees and other methods to predict fraud labels. Thus we might find a better performer than the model we choose now.
2. We would validate our prediction and update our model to have a better predicting power if we had new data after 2010. In this project, we left out the out-of-time dataset to validate the model. However, if we continued feeding new data to the supervised model, the model could be updated and make better prediction.

3. We would utilize cross validation to lower the variability of our results. If we had more time, we would perform k-fold validation on our training and testing data instead of simply average the result and make comparison between models.

Appendix: DQR

1. Data Description

This dataset provides the information about the credit card transactions for the year 2010. The dataset contains 96753 credit card transaction record and with 10 fields including information on transaction identifier, card number, transaction time and type, merchant information and label of whether if the transaction is fraudulent.

2. Dataset Overview

The top 5 lines of dataset is shown to give a general overview of the dataset. These fields can be divided into categorical variables and numerical variables. Based on the sample values from the top 5 lines of record and the data description, categorical variables include: Recnum, Cardnum, Date, Merchnum, Merch description, Merch state, Merch zip, Transtype and Fraud; while numerical field includes Amount. The data is then cleaned according to data types.

Recnum	Cardnum	Date	Merchnum	Merch description	Merch state	Merch zip	Transtype	Amount	Fraud
1	5142190439	2010-01-01	5509006296254	FEDEX SHP 12/23/09 AB#	TN	38118	P	3.62	0
2	5142183973	2010-01-01	61003026333	SERVICE MERCHANDISE #81	MA	1803	P	31.42	0
3	5142131721	2010-01-01	4503082993600	OFFICE DEPOT #191	MD	20706	P	178.49	0
4	5142148452	2010-01-01	5509006296254	FEDEX SHP 12/28/09 AB#	TN	38118	P	3.62	0
5	5142190439	2010-01-01	5509006296254	FEDEX SHP 12/23/09 AB#	TN	38118	P	3.62	0

3. Data Summary Table

Data Summary of Categorical Values

#	Field	%Populated	#Unique	Most common	Occurrence of Most Common Category
1	Recnum	100.0%	96753/96752	/	1
2	Cardnum	100.0%	1645/96752	5142148452	1192
3	Date	100.0%	365/96752	2010-02-28	684
4	Merchnum	96.51%	13092/93378	930090121224	9310
5	Merch description	100.0%	13125/96752	GSA-FSS-ADV	1688
6	Merch state	98.76%	228/95558	TN	12035
7	Merch zip	95.19%	4568/92097	38118.0	11868
8	Transtype	100.0%	4/96752	P	96397
9	Fraud	100.0%	2/96752	0	95693

Data Summary of Numerical Values

#	Field	Min	Max	Average	Median	SD	%Populated	# of 0	#unique
1	Amount	0.01	47,900.0	395.83	137.975	831.88	100%	0	34908/96752

4. Individual Fields

1) Field Name: Recnum

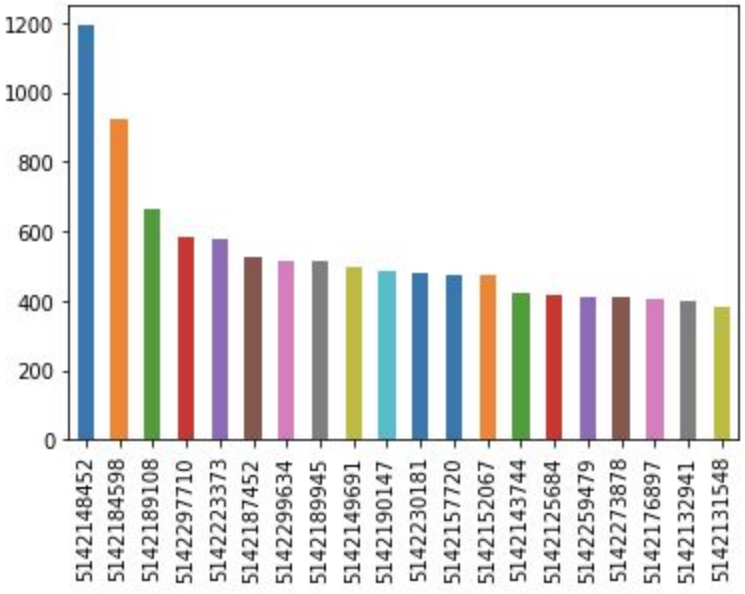
Description: Recnum is a categorical variable, representing unique identifier of each record. Recnum has 96753 lines of records, and is 100.0% populated. Recnum has 96753 unique values for each record.

2) Field Name: Cardnum
Description: Cardnum is the credit card number used in a certain transaction.

Cardnum is a categorical variable. Cardnum has 96752 lines of records and is 100.0% populated. Cardnum has 1645 categories. The most common category is 5142148452, occurred 1192 times out of 96753 records.

Field	Number of Records	Populated %	Unique Value	Most Common Category	Occurrence of Common Category
Cardnum	96752	100.0	1645	5142148452	1192

Below is a graph showing the distribution of Cardnum: (Showing top 20 categories)

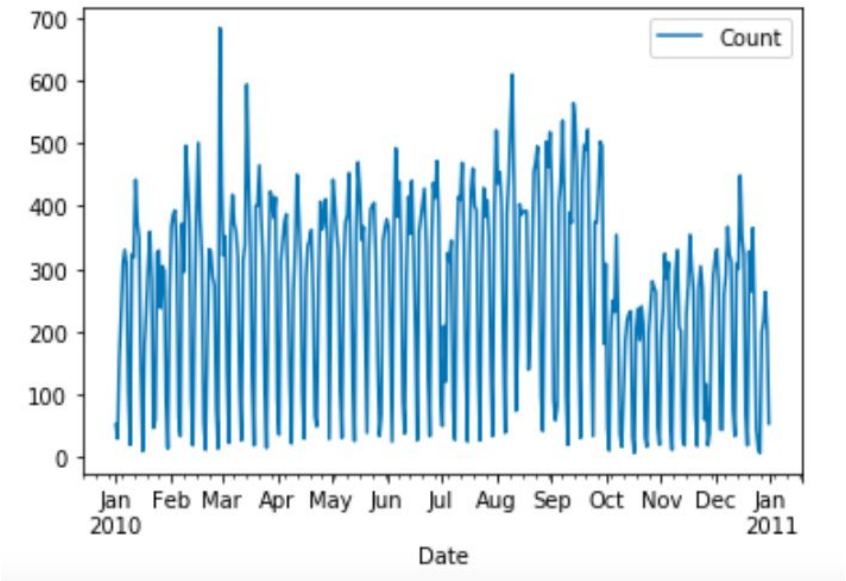


3) Field Name: Date
Description: Date represents the date on which the transaction happened.

Date is a categorical variable. Date has 96752 lines of records and is 100.0% populated. Date has 365 categories each is one day in the year of 2010. The most common category is 2010-02-28, when most number of transactions occurred, occurred 684 times out of 96752 records.

Field	Number of Records	Populated %	Unique Value	Most Common Category	Occurrence of Common Category
Date	96752	100.0	365	2010-02-28	684

Below is a graph showing number of records of each date:

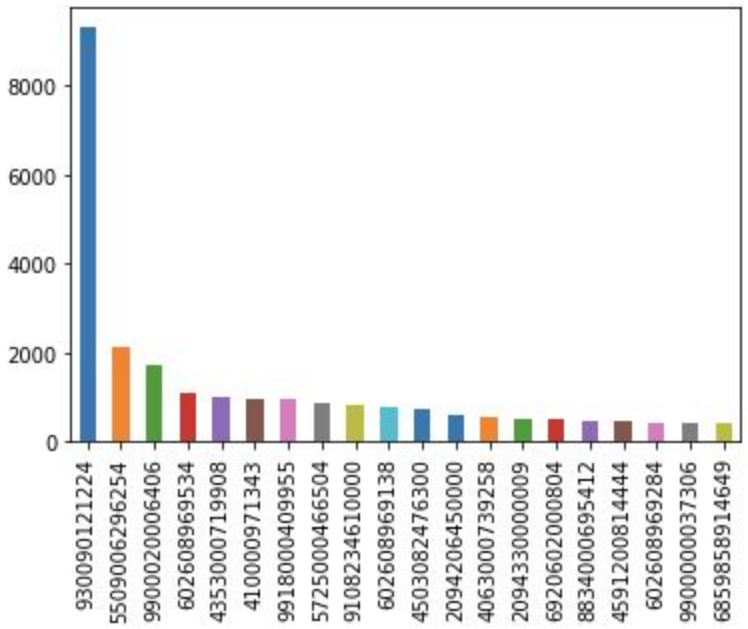


4) Field Name: Merchnum
Description: Merchnum is the merchant number, representing the identifier for merchant.

Merchnum is a categorical variable. Merchnum has 93378 lines of records and is 96.51% populated. Merchnum has 13092 categories. The most common category is 930090121224, occurred 9310 times out of 93378 records.

Field	Number of Records	Populated %	Unique Value	Most Common Category	Occurrence of Common Category
Merchnum	93378	96.51	13092	930090121224	9310

Below is a graph showing the distribution of Merchnum: (Showing top 20 categories)

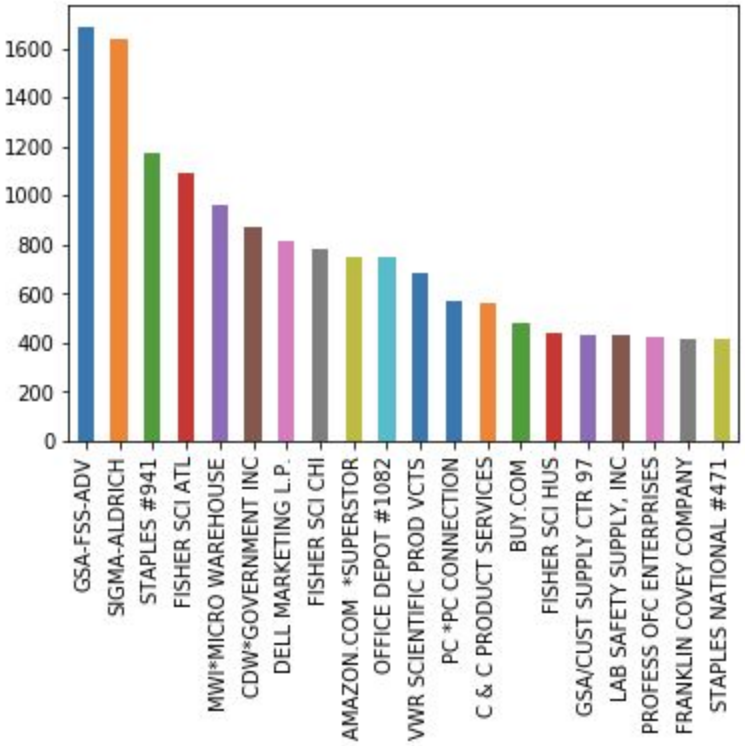


5) Field Name: Merch description
Description: Merch description describes certain merchant.

Merch description is a categorical variable. Merch description has 96752 lines of records and is 100.0% populated. Merch description has 13125 categories. The most common category is GSA-FSS-ADV, occurred 1688 times out of 96752 records.

Field	Number of Records	Populated %	Unique Value	Most Common Category	Occurrence of Common Category
Merch description	96752	100.0	13125	GSA-FSS-ADV	1688

Below is a graph showing the distribution of Merch description: (Showing top 20 categories)

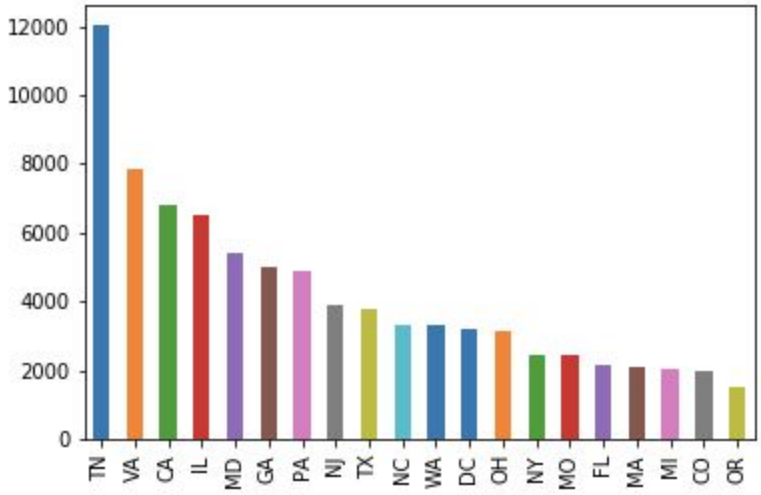


6) Field Name: Merch state
Description: Merch state represents the state where certain merchant locates in.

Merch state is a categorical variable. Merch state has 95558 lines of records and is 98.77% populated. Merch state has 228 categories. The most common category is TN, occurred 12035 times out of 95558 records.

Field	Number of Records	Populated %	Unique Value	Most Common Category	Occurrence of Common Category
Merch state	95558	98.77	228	TN	12305

Below is a graph showing the distribution of Merch state: (Showing top 20 categories)

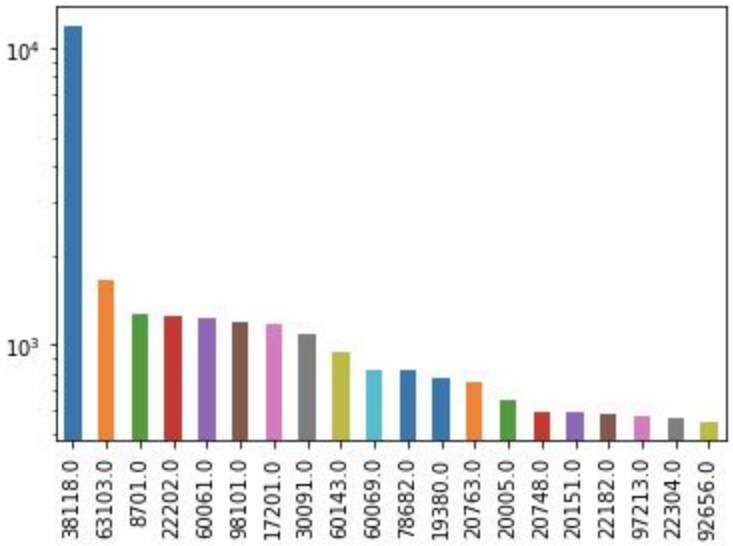


7) Field Name: Merch zip
Description: Merch zip is the zip code for merchant.

Merch zip is a categorical variable. Merch zip has 92097 lines of records and is 95.19% populated. Merch zip has 4568 categories. The most common category is 38118.0, occurred 11868 times out of 92097 records.

Field	Number of Records	Populated %	Unique Value	Most Common Category	Occurrence of Common Category
Merch zip	92097	95.19	4568	38118.0	11868

Below is a graph showing the distribution of Merch zip: (Showing top 20 categories)

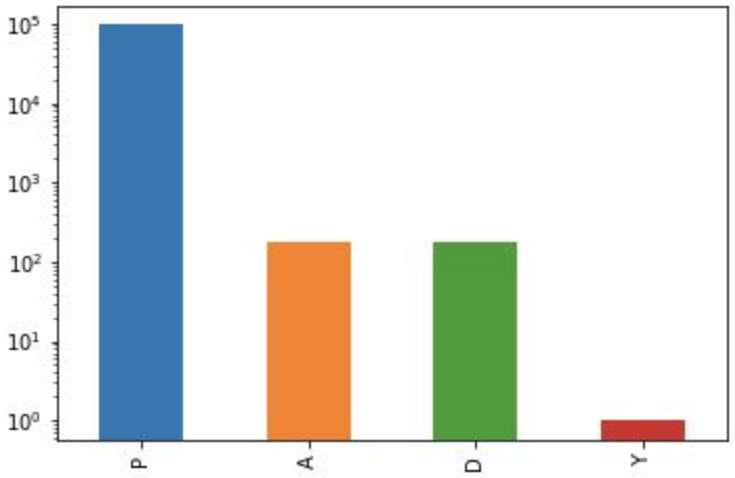


8) Field Name: Transtype
Description: Transtype indicates the type of transaction.

Transtype is a categorical variable. Transtype has 96752 lines of records and is 100.0% populated. Transtype has 4 categories. The most common category is P, occurred 963987 times out of 96752 records.

Field	Number of Records	Populated %	Unique Value	Most Common Category	Occurrence of Common Category
Transtype	96752	100.0	4	P	96397

Below is a graph showing the distribution of Transtype: (Showing top 20 categories)

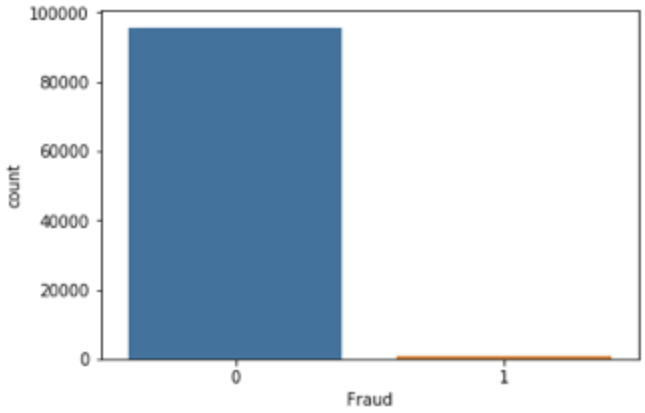


9) Field Name: Fraud
Description: Fraud is a binary variable representing fraud commission by 1, and no fraud by 0.

Fraud is a categorical variable. Fraud has 96752 lines of records and is 100.0% populated. Fraud has 2 categories. The most common category is 0, occurred 956943times out of 96752 records.

Field	Number of Records	Populated %	Unique Value	Most Common Category	Occurrence of Common Category
Fraud	96752	100.0	2	0	95693

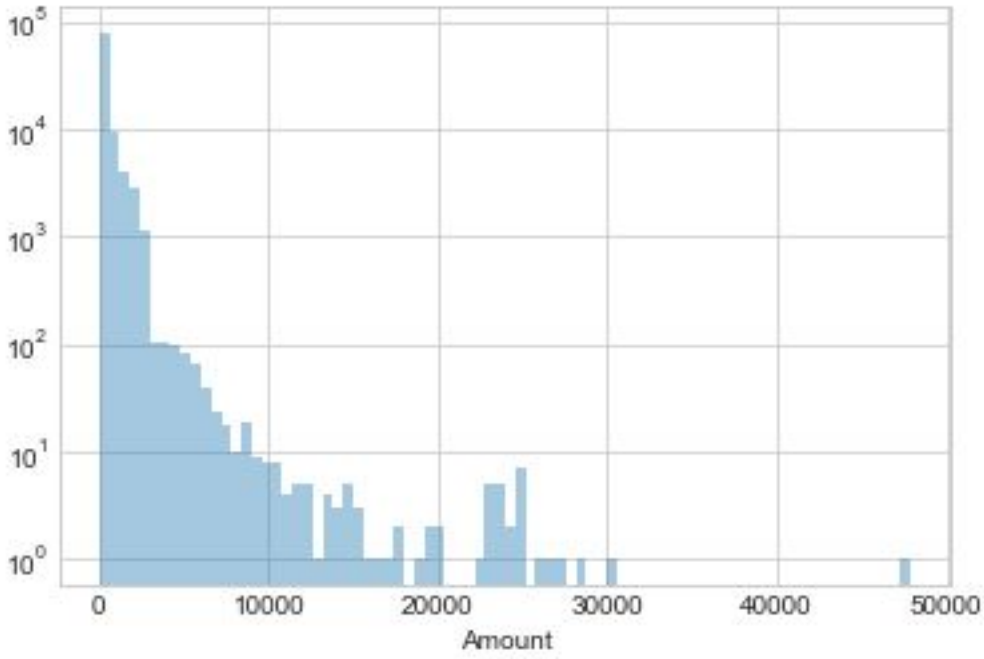
Below is a graph showing the distribution of Fraud:



10) Field Name: Amount
Description: Amount indicates the transaction amounts in dollars.

For field Amount, there are 34908 unique values, and 100% of the records are populated. The distribution plot and the statistical summary of the field are shown below: (excluding null value and an outlier with value of 3 million)

#	Field	Min	Max	Average	Median	SD	%Populated	# of 0	#unique
1	Amount	0.01	47,900.0	395.83	137.975	831.88	100%	0	34908/96752


10^5
10^4
10^3
10^2
10^1
10^0
0
10000
20000
30000
40000
50000
Amount