Saichaitanya Katroju
Amrina Roy
Abhishek Ramji
Kotra Sree Venkata Naga Sreecharan
Swathi Manoharan

DSO 562 Fraud Analytics Homework-12
Project 2: Model Report

Executive Summary:

With the global trend towards digital currency and cashless societies, credit card fraud poses a significant financial burden for economies worldwide. As per a Nilson report, over the last five years, more than 47% of Americans have experienced credit card fraud, and the United States leads in fraud losses, accounting for 36.8% of all losses resulting from card transactions in 2018. In 2021, payment card fraud caused a loss of approximately USD 33 billion to the global economy, and it is predicted to increase to USD 40 billion by 2027.

In this report, the evaluation process for detecting credit card fraud using supervised machine learning techniques is presented in detail. The raw dataset includes approx. 97,000 rows and 10 columns, covering the period from January to December 2010. The validation data comprises the last two months of the dataset (November and December 2006), while the remaining 10 months were split into train and test datasets in a 70:30 ratio.

To detect fraud, the methodology involved examining and visualizing each variable in the dataset and cleaning the raw data to handle missing and irrelevant entries. In feature engineering, 1954 potential variables were generated by extracting features from the raw data. For feature selection, filtering was performed using boosted trees algorithm (LGBM Classifier) was utilized with forward stepwise selection to narrow down the top 20 variables. These 20 variables were then used to train various machine-learning models.

Initially, logistic regression was employed as the baseline model. Subsequently, other modeling techniques were implemented, such as decision trees, random forest, gradient boosting, and neural networks. Finally, the random forest model was selected as it demonstrated superior performance on the out-of-time dataset, with a fraud detection rate (FDR) of approximately **38.61** % at 3% FDR and **44.30** % at 5% FDR. In other words, the model can identify over 50% of fraudulent applications when analyzing the top 5% of data sorted by fraud algorithm score, which can result in annual savings of about $**12.2** million.

1. Data Description:

The given data is a CSV file. It contains Credit card transaction records of customers' data which give information about each transaction. It Contains records from 1st January 2010 to 31st December 2010 and identifies if a given transaction is fraudulent or not (Identified by the column ‘Fraud’). There are a total of 10 fields and 96,753 records(rows) in the dataset.

2. **Summary tables:**

Numerical Variable:

Field Name	Min	Max	%Populated	% NULL	Mean	Stdev
Date	1/1/10	12/31/10	100	0	N/A	N/A
Amount	0.01	3,102,045.53	100	0	427.89	10,006.14

Categorical Variable:

Field Name	% Populated	Unique Values	Most repeated values
Recnum	100.00%	96,753	-
Cardnum	100.00%	1,645	5142148452
Merchnum	96.51%	13,091	930090121224
Merch description	100.00%	13,126	GSA-FSS-ADV
Merch state	98.76%	227	TN
Merch zip	95.19%	4,567	38118
Transtype	100.00%	4	P
Fraud	100.00%	2	0

Description and visualization of each field

1. Field Name: ***Recnum***: This field represents the Record Number, which is an ordinal unique positive integer for record, ranging from 1 to 96,753.

2. Field Name: ***Date***: This field represents the Date of the transaction ranging from 1/1/10 to 12/31/10 for the year 2010

3. Field Name: ***Cardnum***: This field represents the Card Number of the customers. The most frequent value of this column is 5142148452 and there are 1192 such values.

4. Field Name: ***Merchnum***: This field represents the Merchant Numbers of the transactions. The most frequent value of this column is '930090121224' and the count of this value is 9310.

5. Field Name: ***Merch description*** : This field represents the Merchant description of the transactions. The mode (most frequent value) of this column is 'GSA-FSS-ADV' and the count of this value is 1688.

6. Field Name: ***Merch state:*** This field represents the Merchant State of the transactions. The most frequent value of this column is 'TN', and the count of this value is 12,035.

7. Field Name: ***Merch zip:*** This field represents the Merchant zip code (5 digit) of the transactions. The most frequent value of this column is 38118 and the count of this value is 11,868.

8. Field Name: ***Transtype:*** This field represents the different types of transactions used by the customers. The most frequent value of this column is 'P', and the count of this value is 96,398.

9. Field Name: ***Amount:*** This field represents the Amount of each transaction. We ca. see the transaction below 2500 $ as well as transactions above 2500 $. The mean of this column is $427.89, and the standard deviation is $10,006.14.

10. Field Name: ***Fraud:***
    Fraud: This field represents if a particular transaction is fraud.
    The Value 0 = 'Not a Fraud Transaction'
    Value 1 = 'Fraud Transaction'
    non-Fraud = 95,694
    Frauds = 1059

3. Data Cleaning:

A feature is any measurable input that can be used in a predictive model. It is the process of leveraging domain knowledge to extract new features from the raw data. It can produce new features for both supervised and unsupervised learning. It simplifies & speeds up data transformations and enhances model accuracy. We performed the following steps for feature engineering:

1. Excluding bad records
2. Removed all but "P" type of transactions
3. Removed outlier in the 'Amount' field
4. Imputed missing values in the below columns:
5. Removing duplicate columns: Last step of feature engineering is deduping duplicate columns.

- ***Merchnum:***

    - We are creating a dictionary of key-value: {Merch description: Merch num} and then filling the missing MERCHNUM by mapping with Merch description and assigning its corresponding MERCHNUM. We are also assigning unknown records for adjustments transactions

- ***Merch zip:***

    - We are creating a dictionary of key-value: {Merch num: Merch zip} and {Merch description : Merch zip}. We are then filling the missing MERCH ZIP by mapping with Merch num and Merch description and assigning its corresponding MERCH ZIP. We are also assigning 'unknown' records for adjustments transactions. The rest null values are also filled with 'unknown'

- ***Merch state:***

    - We assign state values to certain zip codes with null values in Merch State. We then create a dictionary of key-value: {Merchnum : Merch state} and {Merch description : Merch state}. We are then filling the missing MERCH STATE by mapping with Merchnum and Merch description and assigning its corresponding MERCH STATE. We are also assigning 'unknown' records for adjustments transactions.

4. Variable Creation:

Feature Engineering refers to the creation of new features from existing data. It is a crucial step in the machine learning process, where domain knowledge is used to extract meaningful characteristics, properties, and attributes from raw data. This process results in new variables that can be used in predictive models, improving their accuracy, and simplifying data transformations. Feature engineering can be applied to both supervised and unsupervised learning algorithms, and a feature is any measurable input that can be fed into the model.

In our project, we have carried out feature extraction process across the following steps:

4.1 Target Encoding
4.2 Statistical Smoothing
4.3 Attribute Creation

Using fuzzy logic, we created 1954 variables, and these variables contain velocity variables, days since variables, relative velocity variables, amount variables and Benford's law variables.

SUMMARY TABLE

Description of variables	Attributes used	# Of variables created
Original fields from data set excluding 'record' and 'fraud_label' : count : 8		

dow: Day of week target encoded (average fraud percentage on that day) **Risk table variable:** Used to convert the DOW categorical variables into numerical		1
Customs Variables / attributes: These attributes are used to create the variables that are listed in this document card_merch card_zip card_state zip3 card_zip3 merchnum_state merchnum_zip merchnum_zip3		
Benford's Law variables: • Cardnum • Merchnum	• Cardnum • Merchnum • Amount	2
Days since variables: Number of days since that attribute has been last seen **Frequency and Amount Variables:** Frequency tracks the number of times the entity was encountered in the past n days Amount measures statistical summary metrics such as average, total, and ratios Values of n are: [0, 1, 3, 7, 14, 30, 45 ,60 ,90] Hence, Total days since variables: 13 (one for each attribute) Total frequency & amount over n days variables: 13 * 9 * 9 = 1053 Total new variables = 1053 + 13 = 1066	• Cardnum • Merchnum • Merch description • Merch state • Merch zip • card_merch • card_zip • card_state • zip3 • card_zip3 • merchnum_state • merchnum_zip • merchnum_zip3	1066

Velocity change variables: Measures the change in number of card transactions in the past 0 or 1 days over the average daily number of transactions in the past 7, 14, 30, 45, 60 & 90 days Total number of variables = 13(# of attributes) * 2 (for n in recent past {0, 1}) * 6 (for n in past {7, 14, 30,45 ,60 ,90}) = 26 * 6 = 156		156
Velocity days since entity variables: Measures the velocity change variables for an entity over the days-since variables for the same entity. Total number of variables = 13(# of attributes) * 2 (for n in recent past {0, 1}) * 6 (for n in past {7, 14, 30,45 ,60 ,90}) = 26 * 6 = 156		156
Unique variables: Calculates the occurrences of unique entities/combination groups for a particular entity/combination group		156
Acceleration: Calculating the ratio of velocity change variables for given entities over the square of average daily number of card transactions with the same entities over the past n days Total number of variables = 13(# of attributes) * 2 (for n in recent past {0, 1}) * 6 (for n in past {7, 14, 30,45 ,60 ,90}) = 26 * 6 = 156		156
Variability variables: Total number of variables = 13(# of attributes) * 3 variables (avg., max, median) * 6 (for n in past {0,1,3,7, 14, 30}) = 39 * 6 = 234		234

Frequencies & Frequency mappers:		8

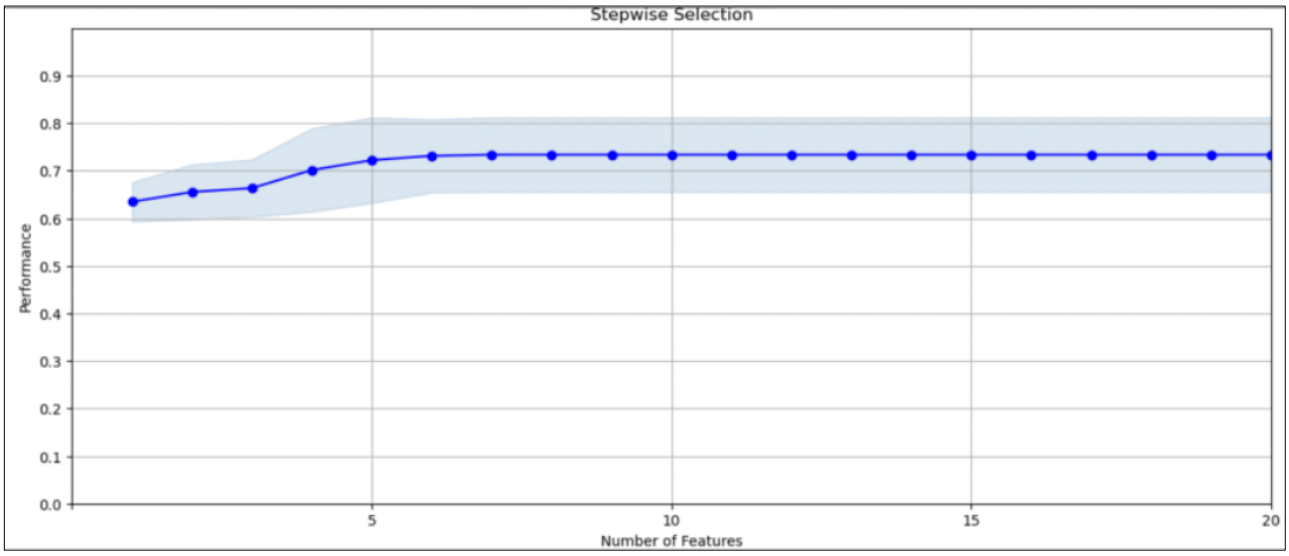
Total variables after variable creation: **1954**

5. Feature Selection

Feature selection is an important technique to improve the performance of machine learning models, especially in high-dimensional data. Due to the curse of dimensionality, it is easier to fit fewer dimensions of nonlinear models. Feature selection can also improve the model architecture. To facilitate the selection process, variables are ordered based on their relevance. For example, in our fraud detection project, we first applied filter feature selection by calculating the KS statistic through univariate tests to get the top 13% of variables. Then, we used the LGBM model to do the forward wrapper selection and identified 20 variables with a fraud detection rate at a 0.03 cutoff. LGBM model with **num_filter : 500, num_wrapper: 20** was selected as the best. Along with LGBM Forward Selection we also tried backward selection and Random Forest model for the feature selection. The faster runs achieved through feature selection help ensure that we're working with the most relevant variables and improving the efficiency of our modeling process.

Forward feature selection using LGBM model

Num_filter : 500, num_wrapper: 20, LGBM, Wrapper Performance: 0.73



Final 20 Variable list

Model chosen: LGBM model with num_filter : 500, num_wrapper: 20 Reason for choosing these final variables:

- Random Forest model was used for feature selection with Num_filter : 500, num_wrapper: 20 which gave wrapper performance of 0.68, which gives a different set of 20 variables each run.
- LGBM model was used with forward selection by increasing num_filter to 500 and the performance increases to 0.73 which is the best model.

Wrapper_order	variable	filter score
1	card_zip3_total_7	0.696009164
2	card_zip3_total_3	0.688060912
3	card_zip_total_7	0.684517116
4	card_zip3_total_14	0.681898098
5	card_merch_total_7	0.681641411
6	card_zip_total_3	0.677562597
7	card_merch_total_14	0.67609062
8	card_merch_total_3	0.675453834
9	card_zip_total_14	0.671827897
10	Card_Merchdesc_total_7	0.671266808
11	card_zip3_total_1	0.666443298
12	Card_Merchnum_desc_total_7	0.666114336
13	Card_Merchdesc_total_14	0.665537947
14	card_zip3_max_7	0.665217641
15	Card_Merchnum_desc_total_14	0.66328876
16	Card_Merchdesc_total_3	0.661331292
17	card_zip3_total_30	0.66121996
18	card_zip_total_1	0.660539487
19	card_merch_total_30	0.659999235
20	Card_Merchdesc_max_14	0.65916646

- The final set of variables that were generated with forward feature selection using LGBM with the filter count set to 380 (20% of num vars – 1887) and wrapper count to 20, we get performance close to 73%

6. Preliminary Model Exploration:

To find the optimal model, we experiment with 10, 15 and 20 variables and various hyperparameters as inputs at the 3% FDR. First, we establish a baseline performance using logistic regression and then compare its performance with several other nonlinear models such as Decision tree, Random Forest, LightGBM, Neural Network, Logistic Regression, Catboost, and XGBoost.

This approach allows us to systematically compare and evaluate the performance of different models, considering the complexity of the model and the trade-off between bias and variance. It's important to note that the choice of hyperparameters and input variables can have a significant impact on the performance of the model, so it's important to carefully tune these parameters and conduct a thorough analysis of the results. Overall, this approach helped us build more accurate and effective machine learning models.

Based on the below model performance, we decided to select **Random Forest** as our final model.

Homework-10

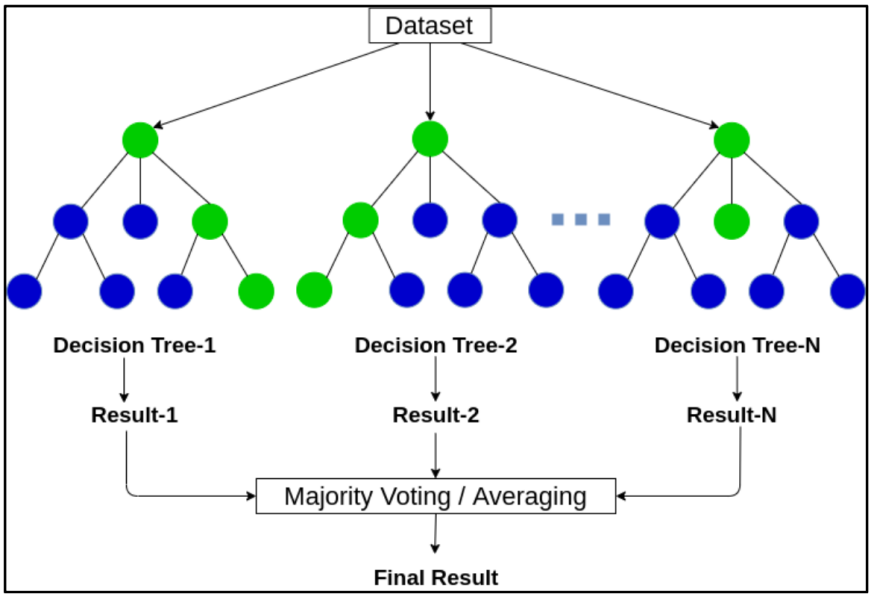
MODELS AND ITS ACCURACIES

Model	Parameter										Average FDR at 3%			Observation
Logistic Regression	Iteration	num_variables	penalty	c	solver	l1_ratio					Train	Test	OOT	
	1 (default)	10	l2	1	lbfgs	None					50.72	51.43	26.39	**Under-fitting**
	2	10	l2	0.8	lbfgs	0					53.16	53.51	27.34	**Best**
	3	10	l1	1	liblinear	None					53.44	53.76	25.31	
	4	10	elasticnet	0.5	saga	1					55.28	52.79	27.21	
	5	10	elasticnet	1	saga	0.5					53.88	54.53	27.08	
	6	15	l1	0.7	saga	None					60.43	60.56	22.40	
	7	20	l2	0.3	lbfgs	None					62.57	63.41	20.75	
	8	15	elasticnet	1	saga	0.2					59.14	60.32	21.77	
Decision Tree	Iteration	num_variables	criterion	max_depth	min_samples_split	splitter	min_samples_leaf				Train	Test	OOT	
	1 (default)	10	gini	5	2	best	1				58.68	58.58	31.13	**Under-fitting**
	2	10	gini	10	100	best	4				67.30	61.62	33.03	
	3	10	entropy	20	100	best	2				87.11	61.91	28.10	
	4	10	gini	60	50	best	2				95.91	57.86	25.44	
	5	10	entropy	10	200	random	4				60.31	60.80	34.68	**Best**
	6	15	gini	10	100	random	2				64.36	59.57	30.12	
	7	20	entropy	200	100	best	2				92.27	67.98	30.88	**Over-fitting**
Random Forest	Iteration	num_variables	n_estimators	max_depth	min_samples_split	min_samples_leaf	max_features	criterion			Train	Test	OOT	
	1(default)	10	100	5	2	1	4	gini			61.97	59.34	52.54	**Best**
	2	10	50	10	1000	10	8	gini			65.45	61.88	43.29	
	3	10	50	20	500	10	6	entropy			73.24	66.27	39.36	
	4	10	100	100	1000	2	6	gini			79.30	62.71	41.89	
	5	20	100	100	500	2	8	gini			86.70	72.01	36.45	**Over-fitting**
	6	15	100	2	100	2	5	gini			62.91	62.45	38.48	
	7	15	150	10	500	25	10	entropy			72.02	65.80	39.11	
	8	15	10	2	100	2	5	gini			58.91	56.06	28.22	**Under-fitting**
LightGBM	Iteration	num_variables	boosting_type	max_depth	n_estimators	num_leaves	subsample	colsample_bytree	learning_rate		Train	Test	OOT	
	1(default)	10	gbdt	-1	100	2	1	1	0.1		60.00	57.71	39.36	**Best**
	2	10	gbdt	10	100	10	1	1	0.1		80.30	67.61	34.30	
	3	10	dart	10	200	10	1	1	0.1		73.53	67.48	36.20	
	4	10	gbdt	20	500	10	1	0.8	0.1		96.56	64.61	31.13	
	5	10	gbdt	10	50	50	0.8	0.8	0.1		93.52	65.28	32.27	
	6	10	gbdt	20	100	50	0.8	0.8	0.1		97.22	64.33	28.60	
	7	10	gbdt	10	100	50	0.8	1	0.2		93.53	57.87	21.77	**Over-fitting**
	8	20	gbdt	10	80	70	1	0.8	0.03		96.56	74.42	34.17	
	9	15	dart	50	100	30	0.6	0.8	0.1		92.13	72.07	35.06	
	10	15	dart	30	100	70	0.8	1	0.03		90.03	70.57	33.92	
XGBoost	Iteration	num_variables	booster	max_depth	n_estimators	tree_method	min_child_weight	subsample	colsample_bytree	eta	eval_metric	Train	Test	OOT	
	1(default)	10	gbtree	6	5	auto	1	1	1	0.3	rmse	63.19	58.90	40.75	**Best**
	2	10	gbtree	10	50	hist	1	0.8	0.8	0.1	rmse	84.31	68.67	36.32	
	3	10	dart	10	50	auto	1	0.8	0.8	0.1	rmse	87.04	67.84	36.32	
	4	10	gbtree	20	50	auto	1	1	1	0.3	mae	99.74	65.66	32.27	
	5	10	gbtree	30	1000	hist	1	0.8	1	0.1	mae	99.87	62.87	24.55	**Over-fitting**
	6	10	gbtree	10	200	auto	10	1	1	0.1	rmse	85.85	67.58	35.94	
	7	20	dart	5	50	hist	1	1	1	0.1	mae	82.25	72.08	36.70	
	8	15	gbtree	30	1000	hist	5	0.6	1	0.1	mae	99.30	67.00	27.72	
CatBoost	Iteration	num_variables	depth	bootstrap_type	l2_leaf_reg	grow_policy	learning_rate	random_state	Iterations	min_data_in_leaf		Train	Test	OOT	
	1(default)	10	6	Bayesian	3	SymmetricTree	0.03	None	1000	1		55.89	56.87	42.27	**Best**
	2	10	5	Bayesian	12	Depthwise	0.02	2	500	2		69.32	67.01	37.34	**Over-fitting**
	3	10	7	Bernoulli	12	SymmetricTree	0.01	32	5	2		42.22	41.86	24.43	**Under-fitting**
	4	20	10	Bayesian	8	SymmetricTree	0.03	2	500	4		87.31	75.47	35.18	**Local Minima**
	5	20	5	Bayesian	10	SymmetricTree	0.01	2	500	10		68.15	66.90	36.70	
	6	15	10	Bayesian	8	Depthwise	0.003	32	5	10		53.22	50.89	25.44	
Neural Network	Iteration	num_variables	activation	solver	hidden_layer_size	max_iter	learning_rate	alpha				Train	Test	OOT	
	1(default)	20	relu	adam	2	200	constant	0.1				66.16	66.24	30.12	
	2	20	relu	sgd	10	300	adaptive	0.2				61.88	60.91	28.86	
	3	20	logistic	adam	10	100	adaptive	0.03				66.06	62.39	31.13	**Over-fitting**
	4	15	relu	adam	10	500	constant	0.3				62.22	62.91	33.03	**Best**
	5	10	logistic	sgd	10	500	adaptive	0.002				52.29	50.40	30.25	**Under-fitting**

7. Final Model Performance

Random Forest

This is an ensemble learning technique where a collection of decision trees is used for classification or regression. Each individual tree is created through some sort of randomness for example using only a randomly chosen subset of variables or records for each tree and/or for each split iteration within a tree. Results from all trees are combined using voting or average.



Random forests improve over decision trees in terms of overfitting and stability.
In addition to decision tree hyperparameters, random forests have some more hyperparameters like:

- n_estimators: number of trees in the forest
- Bootstrap: whether bootstrap samples are used to build the trees

Random Forest is a bagging algorithm and has the following advantages:

- Random Forest can handle large datasets with high dimensionality. It can effectively work with thousands of input variables without overfitting.
- Random Forest is an ensemble model, which means that it combines multiple decision trees to make predictions. This reduces the risk of overfitting and improves the accuracy and stability of the model.
- Random forests can handle both categorical and continuous input variables, and they can handle missing data without the need for imputation.
- Random Forest is not sensitive to outliers, and it can handle imbalanced datasets.

Results:

The Random Forest model was selected as the final model after preliminary exploration, as it showed the highest average FDR for testing and relatively high FDR for OOT, training, and test datasets, with a small range of standard deviations indicating stable performance. The data was split into training, testing, and OOT sets, with similar fraud rates of around 0.01. The model could eliminate about 61.97% of fraud in the training set, 59.34% in the test set, and 52.54% in the OOT set, by declining only about 3.5% of applications. Results and top 20 percentile bins for all three datasets are summarized in a table, including the number of records, frauds caught, their percentage, cumulative KS and FPR values.

Training Summary

Train	# Record	# Goods	# Bads	Fraud Rate
	484,833	480,919	3,914	0.01422

	Bin Statistics					Cumulative Statistics						
Population Bin %	# Records	# Goods	# Bads	% Goods	% Bads	Total # Records	Cumulative Goods	Cumulative Bads	% Goods	% Bads (FDR)	KS	FPR
1	612	302	310	49.35	50.65	612	302	310	0.50	48.67	48.17	0.97
2	612	527	85	86.11	13.89	1224	829	395	1.37	62.01	60.64	2.10
3	611	553	58	90.51	9.49	1835	1382	453	2.28	71.11	68.83	3.05
4	612	579	33	94.61	5.39	2447	1961	486	3.24	76.30	73.06	4.03
5	612	598	14	97.71	2.29	3059	2559	500	4.23	78.49	74.27	5.12
6	612	596	16	97.39	2.61	3671	3155	516	5.21	81.00	75.79	6.11
7	611	601	10	98.36	1.64	4282	3756	526	6.20	82.57	76.37	7.14
8	612	599	13	97.88	2.12	4894	4355	539	7.19	84.62	77.42	8.08
9	612	594	18	97.06	2.94	5506	4949	557	8.17	87.44	79.27	8.89
10	612	607	5	99.18	0.82	6118	5556	562	9.18	88.23	79.05	9.89
11	611	603	8	98.69	1.31	6729	6159	570	10.17	89.48	79.31	10.81
12	612	608	4	99.35	0.65	7341	6767	574	11.18	90.11	78.93	11.79
13	612	607	5	99.18	0.82	7953	7374	579	12.18	90.89	78.71	12.74
14	612	607	5	99.18	0.82	8565	7981	584	13.18	91.68	78.50	13.67
15	612	607	5	99.18	0.82	9177	8588	589	14.19	92.46	78.28	14.58
16	611	606	5	99.18	0.82	9788	9194	594	15.19	93.25	78.06	15.48
17	612	610	2	99.67	0.33	10400	9804	596	16.19	93.56	77.37	16.45
18	612	610	2	99.67	0.33	11012	10414	598	17.20	93.88	76.68	17.41
19	612	608	4	99.35	0.65	11624	11022	602	18.21	94.51	76.30	18.31
20	611	608	3	99.51	0.49	12235	11630	605	19.21	94.98	75.77	19.22

Testing Summary

Test	# Record	# Goods	# Bads	Fraud Rate
	207,787	206,138	1,649	0.01422

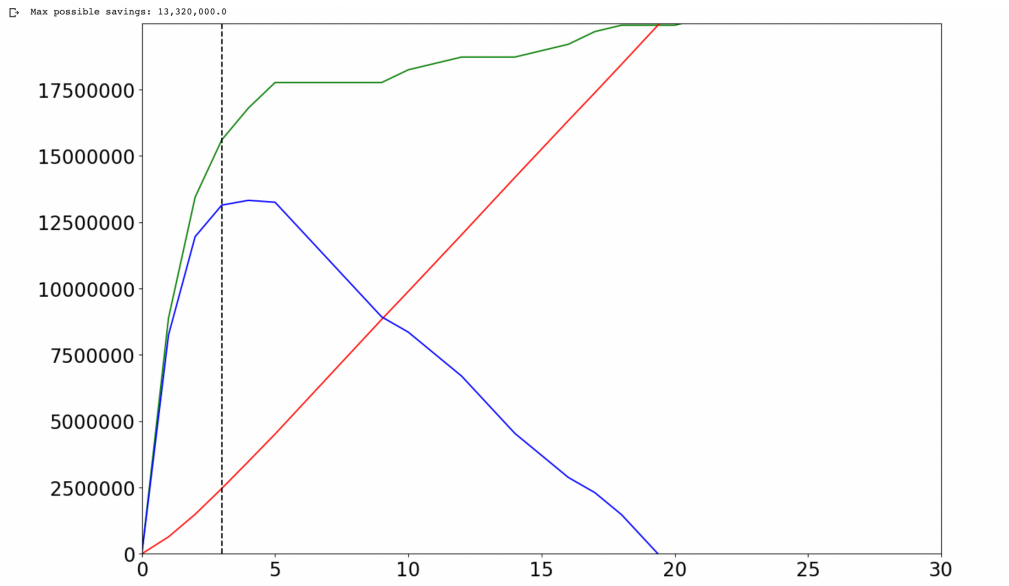
	Bin Statistics					Cumulative Statistics						
Population Bin %	# Records	# Goods	# Bads	% Goods	% Bads	Total # Records	Cumulative Goods	Cumulative Bads	% Goods	% Bads (FDR)	KS	FPR
1	262	130	132	49.62	50.38	262	130	132	0.50	50.00	49.50	0.98
2	262	227	35	86.64	13.36	524	357	167	1.38	63.26	61.88	2.14
3	263	246	17	93.54	6.46	787	603	184	2.32	69.70	67.37	3.28
4	262	252	10	96.18	3.82	1049	855	194	3.29	73.48	70.19	4.41
5	262	251	11	95.80	4.20	1311	1106	205	4.26	77.65	73.39	5.40
6	262	256	6	97.71	2.29	1573	1362	211	5.25	79.92	74.68	6.45
7	262	258	4	98.47	1.53	1835	1620	215	6.24	81.44	75.20	7.53
8	263	263	0	100.00	0.00	2098	1883	215	7.25	81.44	74.18	8.76
9	262	261	1	99.62	0.38	2360	2144	216	8.26	81.82	73.56	9.93
10	262	260	2	99.24	0.76	2622	2404	218	9.26	82.58	73.31	11.03
11	262	261	1	99.62	0.38	2884	2665	219	10.27	82.95	72.69	12.17
12	262	261	1	99.62	0.38	3146	2926	220	11.27	83.33	72.06	13.30
13	263	261	2	99.24	0.76	3409	3187	222	12.28	84.09	71.81	14.36
14	262	262	0	100.00	0.00	3671	3449	222	13.29	84.09	70.80	15.54
15	262	260	2	99.24	0.76	3933	3709	224	14.29	84.85	70.56	16.56
16	262	261	1	99.62	0.38	4195	3970	225	15.30	85.23	69.93	17.64
17	262	259	3	98.85	1.15	4457	4229	228	16.29	86.36	70.07	18.55
18	263	263	0	100.00	0.00	4720	4492	228	17.31	86.36	69.06	19.70
19	262	262	0	100.00	0.00	4982	4754	228	18.32	86.36	68.05	20.85
20	262	259	3	98.85	1.15	5244	5013	231	19.31	87.50	68.19	21.70

OOT Summary

OOT	# Record	# Goods	# Bads	Fraud Rate
	136,659	135,596	1,063	0.01422

	Bin Statistics					Cumulative Statistics						
Population Bin %	# Records	# Goods	# Bads	% Goods	% Bads	Total # Records	Cumulative Goods	Cumulative Bads	% Goods	% Bads (FDR)	KS	FPR
1	90	55	35	61.11	38.89	90	55	35	0.62	22.15	21.53	1.57
2	90	71	19	78.89	21.11	180	126	54	1.43	34.18	32.75	2.33
3	90	83	7	92.22	7.78	270	209	61	2.36	38.61	36.24	3.43
4	90	87	3	96.67	3.33	360	296	64	3.35	40.51	37.16	4.63
5	90	84	6	93.33	6.67	450	380	70	4.30	44.30	40.01	5.43
6	90	89	1	98.89	1.11	540	469	71	5.30	44.94	39.63	6.61
7	90	88	2	97.78	2.22	630	557	73	6.30	46.20	39.90	7.63
8	90	89	1	98.89	1.11	720	646	74	7.31	46.84	39.53	8.73
9	90	90	0	100.00	0.00	810	736	74	8.32	46.84	38.51	9.95
10	90	87	3	96.67	3.33	900	823	77	9.31	48.73	39.43	10.69
11	90	88	2	97.78	2.22	990	911	79	10.30	50.00	39.70	11.53
12	90	89	1	98.89	1.11	1080	1000	80	11.31	50.63	39.32	12.50
13	90	90	0	100.00	0.00	1170	1090	80	12.33	50.63	38.31	13.63
14	90	88	2	97.78	2.22	1260	1178	82	13.32	51.90	38.58	14.37
15	90	89	1	98.89	1.11	1350	1267	83	14.33	52.53	38.20	15.27
16	90	89	1	98.89	1.11	1440	1356	84	15.34	53.16	37.83	16.14
17	90	90	0	100.00	0.00	1530	1446	84	16.35	53.16	36.81	17.21
18	90	90	0	100.00	0.00	1620	1536	84	17.37	53.16	35.79	18.29
19	90	88	2	97.78	2.22	1710	1624	86	18.37	54.43	36.06	18.88
20	90	90	0	100.00	0.00	1800	1714	86	19.38	54.43	35.05	19.93

8. Final curve and recommended cutoff



To recommend a cut off we should be as far to the left as possible but away from the sharp increase. We can see a sharp increase from 8.5 M to around 12.5 M. So, we should select a cutoff after that but not very near to the sharp increase. We can see there is not much deviation in overall saving between 2.5 and 4.0, though there is a small peak. We have maximum earnings at 12.25M and we can consider the cutoff at 3.0.

9. Summary – describe everything you did, FDR@3% for oot, $ savings, other things you could do

This project focused on detecting fraudulent credit card transactions using machine learning techniques. We commenced by generating a Data Quality Report that presented the distribution of data. Next, we performed data cleaning, where we handled missing and irrelevant values using imputation methodologies and eliminated certain data points. Additionally, we conducted exploratory data analysis to develop a comprehensive comprehension of the dataset.

Feature engineering was pursued, generating 1954 potential variables based on various combinations of the raw features. We created many candidate variables as we believe it to be a crucial step in identifying the appropriate set of features that can yield excellent results even with linear models. Following that, we focused on reducing the dimensionality of our data by performing feature selection. We utilized boosted tree technique as the wrapper, with forward stepwise selection to arrive at the top 20 variables.

Following this, we ran linear and non-linear models with various combinations of hyperparameters on our data. The modeling techniques involved logistic regression as the base model, followed by decision trees, random forest, lightGBM, XGBoost, CatBoost and neural networks. We selected the random forest technique with specific hyperparameters as our final algorithm. Our model could achieve a Fraud Detection Rate (FDR) of 38.61% by rejecting the top 3% of the population and 44.30% by rejecting the top 5% of the population based on the fraud algorithm score when tested on the out-of-time dataset. With the above model we can have maximum earnings at **12.25M** and we can consider the cutoff at 3.0.

Although every project has its constraints, given more time and access to larger datasets, we would recommend the following enhancements to our project:

- Engage with domain experts to create additional high-quality candidate variables
- Reduce the imbalance in fraud labels by utilizing semi-supervised learning techniques such as co-training, Yarowsky algorithm, active learning, etc., and examining techniques such as SMOTE to generate more labels for the 'fraud' category
- Investigate other model algorithms like k-nearest neighbors, SVM
- Research grid search CV and other boosting techniques to enhance model performance.

10. APPENDIX

Data Quality Report

1. Data Description:

The given data is a CSV file. It contains Credit card transactions records of customers data which gives information about each transaction. It Contains of records from 1st January 2010 to 31st December 2010 and identifies if a given transaction is fraudulent or not (Identified by the column 'Fraud'). There are a total of **10 fields and 96,753 records**(rows) in the dataset.

2. Summary tables:

Numerical Variable:

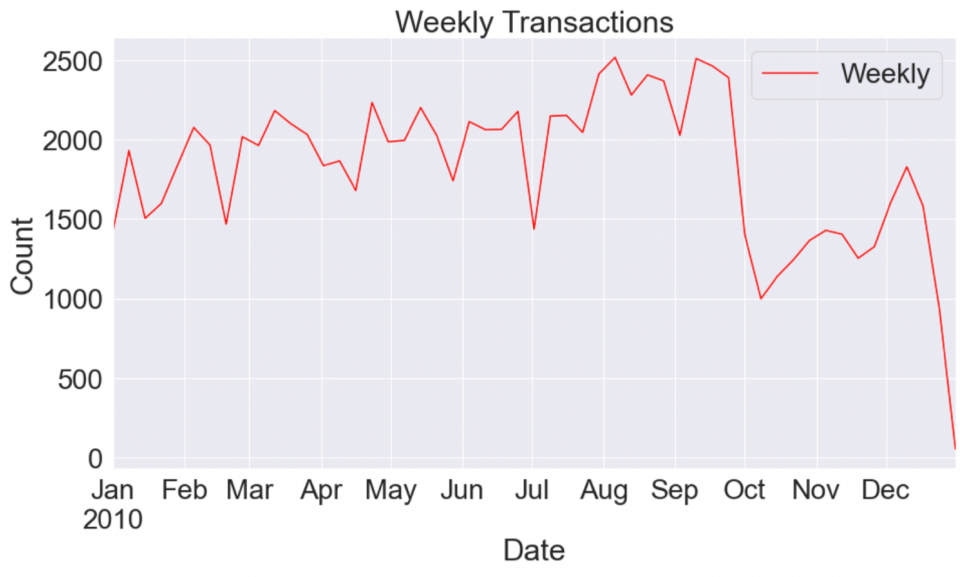
Field Name	Min	Max	%Populated	% NULL	Mean	Stdev
Date	1/1/10	12/31/10	100	0	N/A	N/A
Amount	0.01	3,102,045.53	100	0	427.89	10,006.14

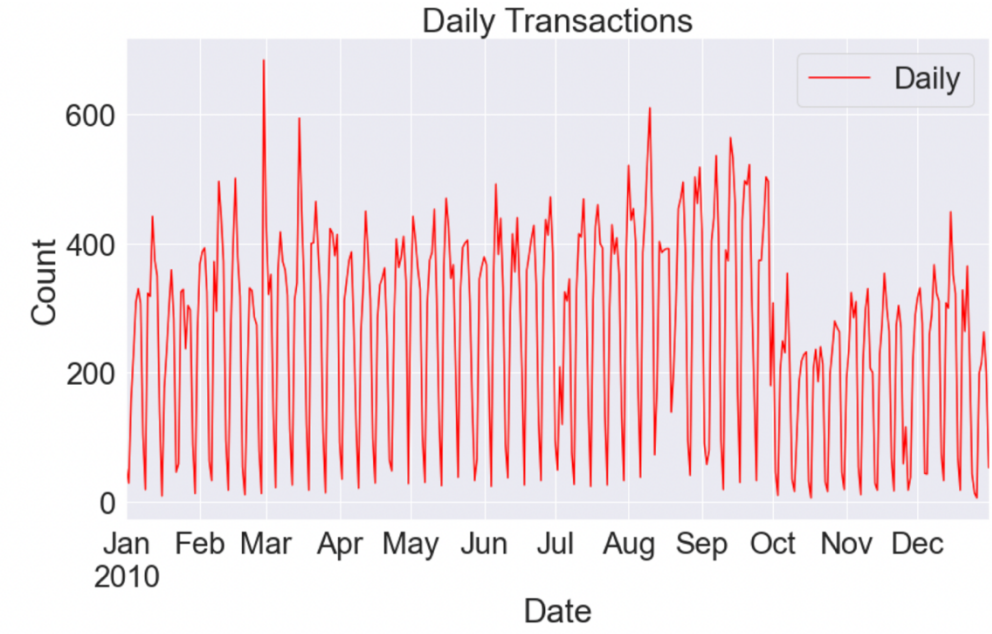
Categorical Variable:

Field Name	% Populated	Unique Values	Most repeated values
Recnum	100.00%	96,753	-
Cardnum	100.00%	1,645	5142148452
Merchnum	96.51%	13,091	930090121224
Merch description	100.00%	13,126	GSA-FSS-ADV
Merch state	98.76%	227	TN
Merch zip	95.19%	4,567	38118
Transtype	100.00%	4	P
Fraud	100.00%	2	0

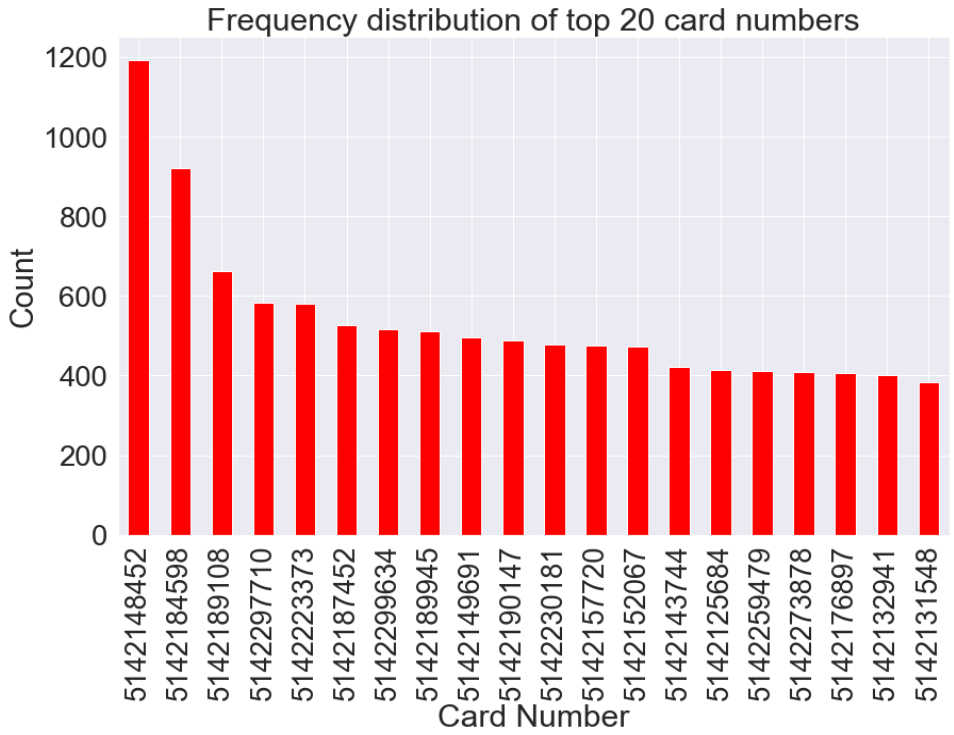
3. Description and visualization of each field

- **Field Name: Recnum:**
  This field represents the Record Number which is an ordinal unique positive integer for each record, ranging from 1 to 96,753.
- **Field Name: Date:** The below is the distributions of transactions over time (daily and we ekly).

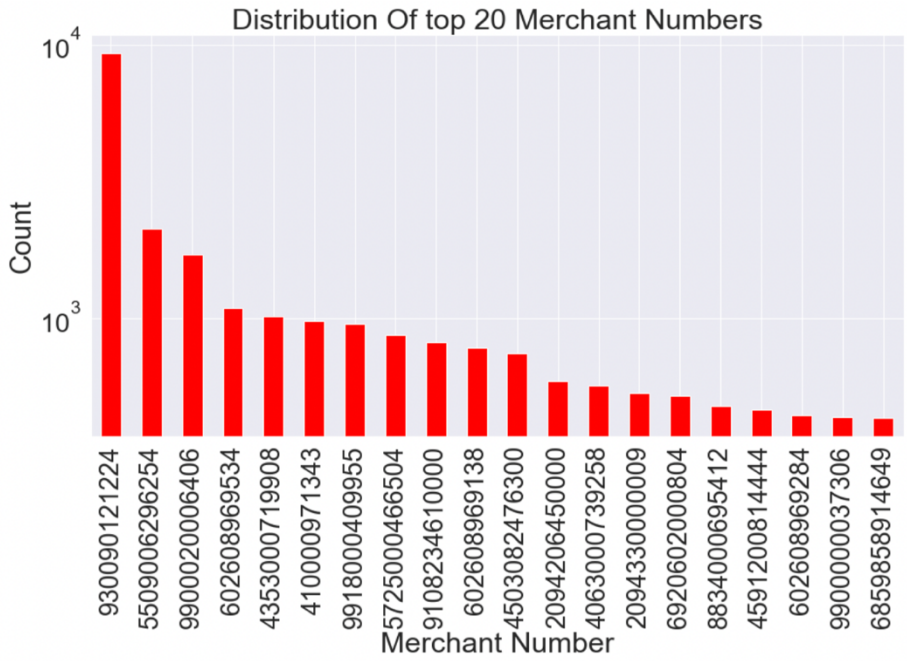




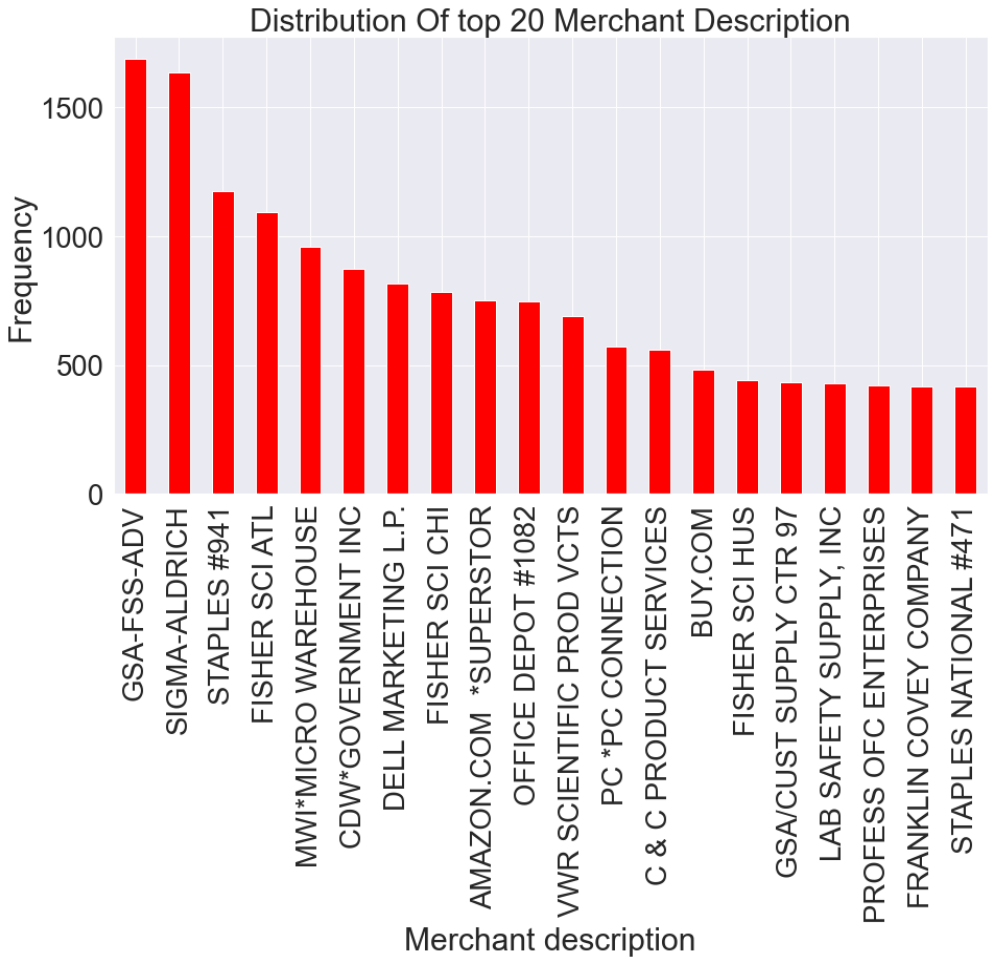
- **Field Name: Cardnum:**
  This field represents the Card Number of the customers. The most frequent value of this column is 5142148452 and there are 1192 such values.



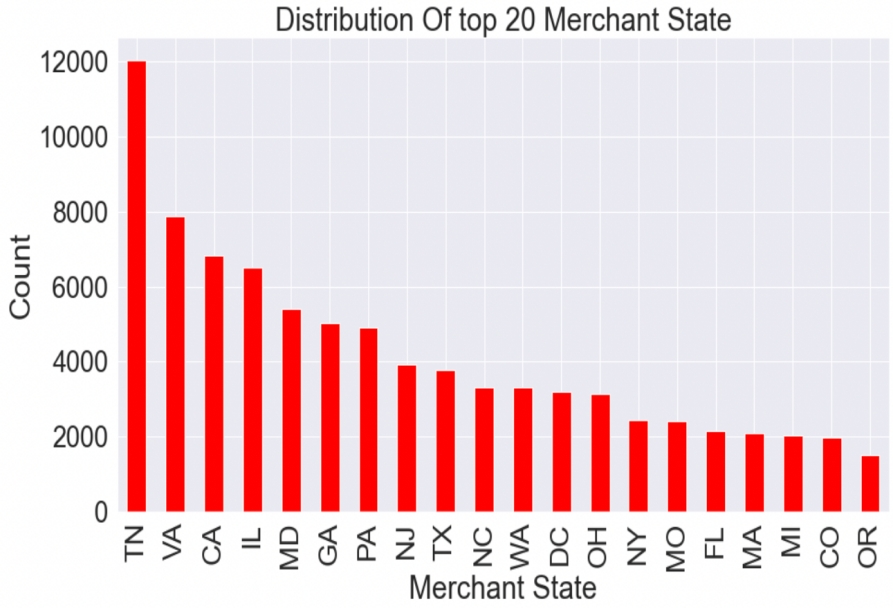
- **Field Name: Merchnum:**
  Merchnum: This field represents the Merchant Numbers of the transactions. The most frequent value of this column is '930090121224' and the count of this value is 9310.



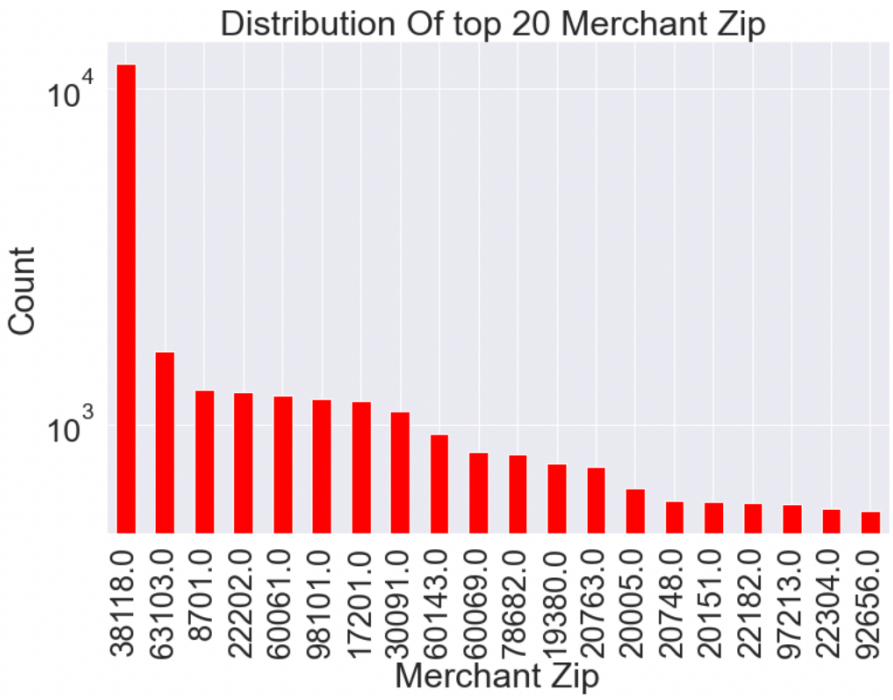
- **Field Name: Merch description:**
  Merch description: This field represents the Merchant description of the transactions. The mode (most frequent value) of this column is ‘GSA-FSS-ADV’ and the count of this value is 1688.



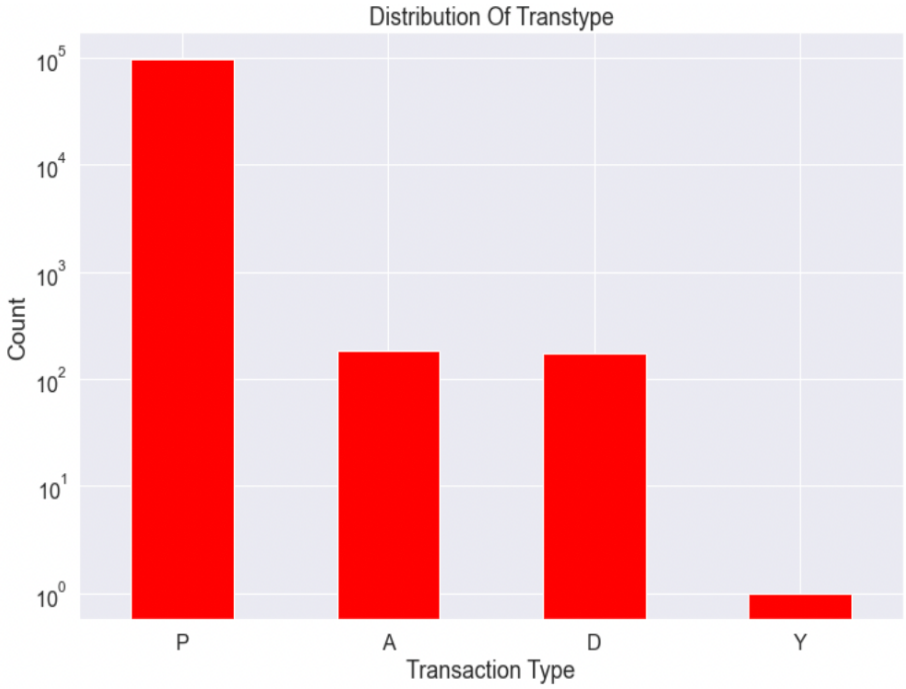
- **Field Name: Merch state:**
  Merch state: This field represents the Merchant State of the transactions. The most frequent value of this column is 'TN' and the count of this value is 12,035.



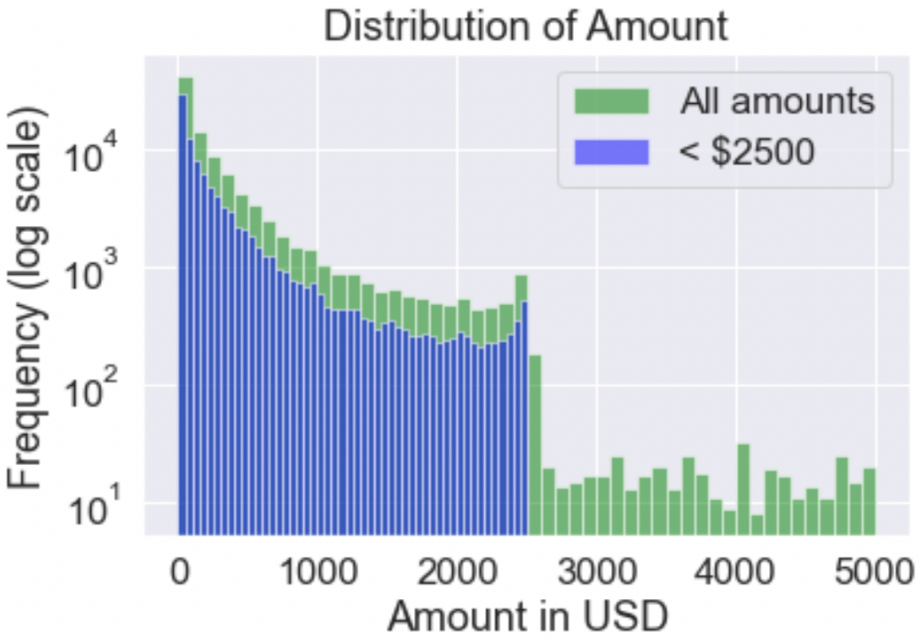
- **Field Name: Merch zip:**
  Merch zip: This field represents the Merchant zip code (5 digit) of the transactions. The most frequent value of this column is 38118 and the count of this value is 11,868.



- **Field Name: Transtype:**
  Transtype: This field represents the different types of transactions used by the customers. The most frequent value of this column is ‘P’ and the count of this value is 96,398.



- **Field Name: Amount:**
  Amount: This field represents the Amount of each transaction. We ca. see the transaction below 2500 $ as well as transactions above 2500 $. The mean of this column is $427.89 and the standard deviation is $10,006.14.



- **Field Name: Fraud:**
  Fraud: This field represents if a particular transaction is fraud. The
  Value 0 = ‘Not a Fraud Transaction’
  Value 1 = ‘Fraud Transaction’
  non-Fraud = 95,694
  Frauds = 1059