# FINAL REPORT

## I. Executive Summary

In this report, we present a fraud detection algorithm built by a dataset of 100,000 credit card transactions in 2010. We will perform data cleaning, variable creation, feature selection, and preliminary model exploration. In the end, we will choose a final model and suggest a recommended cut-off that optimizes cost savings.

## II. Description of Data

**1. Data description**

The data includes a collection of about 100,000 Credit Card transactions. There are in total 10 fields. Including both numeric and categorical ones. The records are collected from January 1st, 2010, to December 31st ,2010, with the total of 96,753 rows. This dataset will be used to identify patterns and trends in fraudulent transactions and to develop a fraud detection algorithm that can accurately identify and prevent fraudulent transactions.

**2. Summary Tables**

a. Numerical Table

| Field | % Populated | Min | Max | Mean | Standard Deviation | %Zero |
|---|---|---|---|---|---|---|
| Date | 100% | 2010-01-01 | 2010-12-31 | n/a | n/a | 0% |
| Amount | 100% | 0.01 | 3,102,045.53 | 427.89 | 10,006.14 | 0% |

b. Categorical table

| Field | % Populated | # Blank | # Zeros | # Unique Values | Most Common Value |
|---|---|---|---|---|---|
| Recnum | 100% | 0 | 0 | 96,753 | 1 |
| Cardnum | 100% | 0 | 0 | 1,645 | 5142148452 |
| Merchnum | 96.51% | 3375 | 231 | 13,091 | 930090121224 |
| Merch description | 100% | 0 | 3 | 13,126 | GSA-FSS-ADV |
| Merch state | 100% | 1195 | 0 | 227 | TN |
| Merch zip | 100% | 4656 | 0 | 4,567 | 38118 |
| Transtype | 100% | 0 | 0 | 4 | P |
| Fraud | 100% | 0 | 0 | 2 | 0 |

## III. Data cleaning

1. Drop blank columns

2. Convert Date to datetime

3. Fill in the missing values:

a. Merchnum:
   - Replace ‘0’ into ‘NaN’ values

- Match Merchnum with corresponding Merch description and then fill in the missing Merchnum values that has a valid Merch description
- Assign 'unknown' to Merchnum if its Merch description value is 'RETAIL CREDIT ADJUSTMENT' or 'RETAIL DEBIT ADJUSTMENT'
- Find top unique values of Merch description → assign remaining missing values of Merchnum to a unique value, each will be max(merchnum) +1 → map each of them with a unique Merch description.

b. Merch state:
- Match Merch state with corresponding Merch zip → fill in missing values
- The remaining missing values map with Merchnum → Merch description
- Assign 'unknown' to Merch state if its Merch description value is 'RETAIL CREDIT ADJUSTMENT' or 'RETAIL DEBIT ADJUSTMENT'
- If there are non-US states → assign 'foreign'

c. Merch zip:
- Match Merch zip with corresponding Merchnum & Merch description → fill in missing values
- Assign 'unknown' to Merch zip if its Merch description value is 'RETAIL CREDIT ADJUSTMENT' or 'RETAIL DEBIT ADJUSTMENT'
- Assign unknown for remaining missing values

4. Remove outlier (largest transaction with 3M in Amount) and all transactions that are not 'P' type.

## IV. Variables Creation

In order to create variables, we created 13 entities where we think that they will be good combinations to identify potential fraud (list of entities are described below). Apart from the variables from those 13 entities, we also created 2 Benford's Law variable, which identifies transactions that do not follow the expected distribution of the first digit of transaction amounts. Benford's Law is a statistical phenomenon that suggests that the first digit of numbers in a dataset should follow a certain distribution, and any deviations from that distribution may indicate fraud. However, later we did decide to remove Benford's Law when we start building models as we noticed that Benford's Law does create overfitting effects for most of our models.

**The final number of variables created are 2235. After deduplication, there are 1694 left.**

| **Description of Variables** | **# Variables Created** |
|---|---|
| **Benford's Law – U* Cardnum and U* Merchnum**<br>The irregularities of the first digits found in card and merchant numbers | 2 |
| **Day of Week Target Encoding – Dow Risk**<br>Average fraud percentage of that day | 1 |
| **Days Since**<br>Number of days since a transaction with that entity was last seen | 715 |
| **Frequency and Amount (3 variables)**<br>• Count: number of times an entity appears in the past transitions over the last 0,1,3,7,214,30 days<br>• Amount: average, max, median, total amounts spent by an entity over the last 0,1,3,7,214,30 days | |

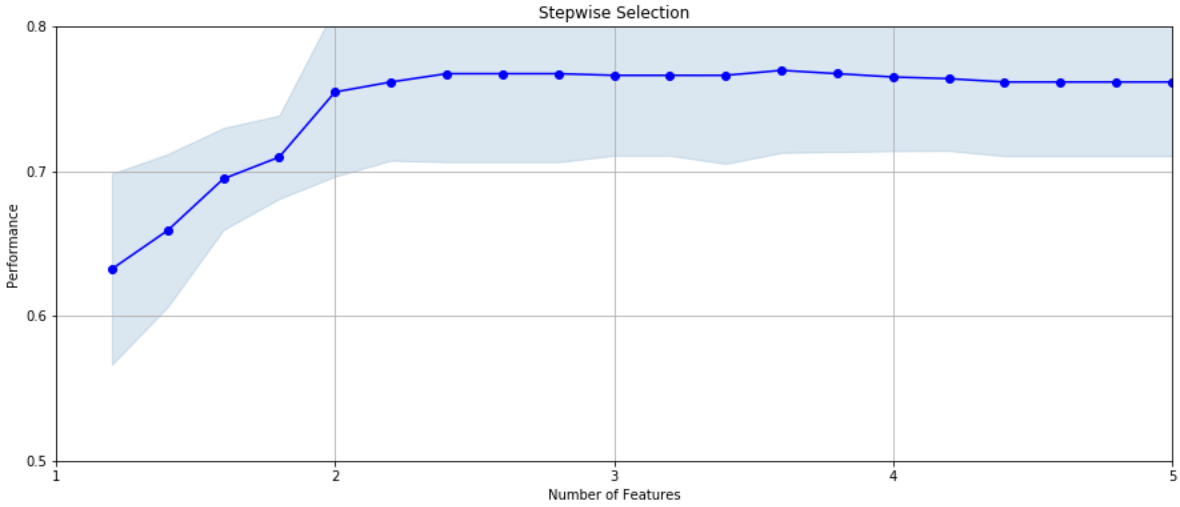
| | |
|---|---|
| • Amount Ratio: ratio of the amount spent in a transaction to the average, max, median, total amount spent by an entity over the last 0,1,3,7,214,30 days | |
| **Velocity Change**<br>Ratio of the short-term velocity (0,1 days) to a longer-term (3,7,14,30 day) velocity | 78 |
| **Velocity Days Since Ratio**<br>Ratio of short-term velocity (0, 1 day) to a longer-term (7, 14, 30 day) velocity divided by the number of days since the last transaction for that entity | 26 |
| **Cross Entity Uniqueness**<br>Number of unique values of a field that are associated with each unique value of an entity | 156 |
| **Relative Velocity Count by Cardnum**<br>Frequency of transactions made by each cardnum within 3,7,14,30 days, relative to the velocity of the transactions | 8 |
| **Variability**<br>Average, Max, Median amount difference between transactions with the same entity over the last 0,1,3,7,14,30 days | 234 |
| **Entity Unique Count**<br>Number of unique values for one entity in relation to another entity over 1,3,7,14,30,60 days | 936 |
| **Amount Bins**<br>Divide the Amount variable into 5 bins using quantiles | 1 |
| **Acceleration**<br>Acceleration of the count of transactions for each entity over different periods of time. | 78 |
| **Total Number of Variables Created** | **2235** |
| **After deduplication, there were 1694 variables left** | |

**List of entities**: 'Cardnum', 'Merchnum', 'Merch description', 'Merch state', 'Merch zip', 'cardnum_merchnum', 'cardnum_merchzip', 'cardnum_merchdescription', 'cardnum_merchstate', 'merchnum_merchstate', 'merchnum_merchdescription', 'cardnum_amount', 'merchnum_cardnum_amount'

## V. Feature selection

For the model to run faster and allow more hyperparameter tuning, feature selection is needed. After making more than 2000 variables, we ran a filter to get down to 400 candidate variables. Then, we chose a simple, fast nonlinear model for the wrapper and ran it 20 times. In the end, we have a final list of 20 variables that are sorted in terms of their univariate KS's (the order of importance). Below is the final list of variables and the plot showing model performance versus number of variables.

I ran in total 6 different times (5 forward, 1 backward & 5 LightGBM, 1 RandomForest)
The backward selection and Random Forest ones took me a lot of time (2-3 hours) and gave me results below 0.7. While LightGBM and forward selection was a lot faster to run and give way better results (> 0.7). I tried different combination of num_filter and num_wrapper for LightGBM and forward selection and chose the one that was closest to 0.75.
**The one that gave me closest to 0.75 is num_filter = 400 and num_wrapper = 20**




**FINAL VARIABLES TABLE**

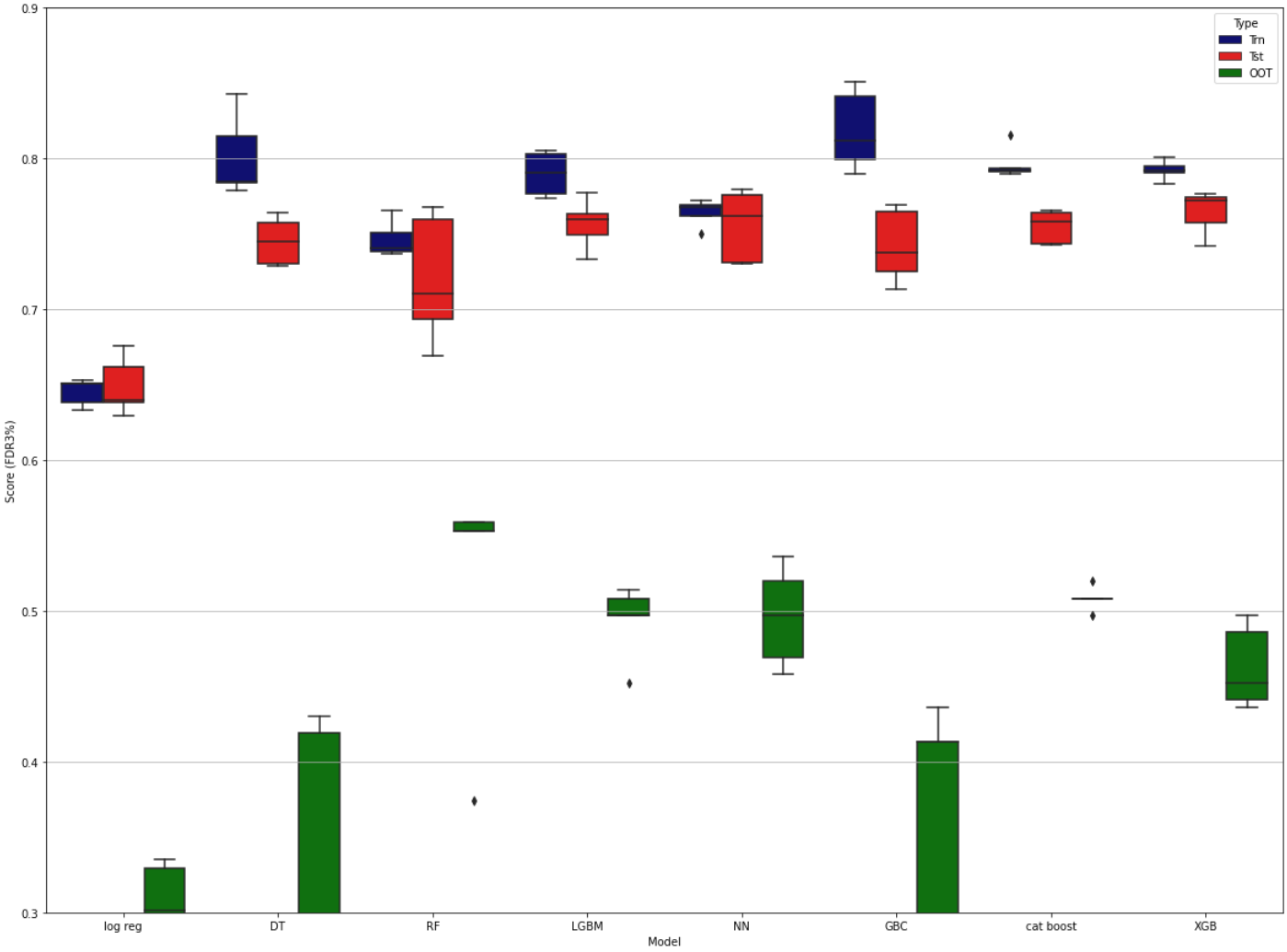
| order | variable | filter score |
|---|---|---|
| 1 | cardnum_merchnum_total_14 | 0.67609062 |
| 2 | cardnum_merchstate_max_14 | 0.63053967 |
| 3 | merchnum_merchdescription_total_3 | 0.61951278 |
| 4 | merchnum_merchstate_total_30 | 0.43528567 |
| 5 | cardnum_merchnum_cardnum_amount_nunique | 0.37809147 |
| 6 | Merch description_total_3 | 0.62488358 |
| 7 | cardnum_merchstate_variability_max_3 | 0.37010799 |
| 8 | cardnum_merchstate_total_14 | 0.66905497 |
| 9 | merchnum_merchdescription_total_30 | 0.50694727 |
| 10 | Merch zip_total_14 | 0.42526741 |
| 11 | cardnum_merchstate_total_30 | 0.63559802 |
| 12 | Cardnum_unique_count_for_merchnum_cardnum_amount_7 | 0.44158015 |
| 13 | Merchnum_total_30 | 0.43522326 |
| 14 | Merch state_variability_avg_0 | 0.3827497 |
| 15 | Cardnum_variability_med_30 | 0.35783913 |
| 16 | cardnum_amount_total_30 | 0.53445481 |
| 17 | Merch description_total_30 | 0.49915082 |
| 18 | merchnum_merchstate_total_7 | 0.58373246 |
| 19 | Merchnum_total_7 | 0.58371998 |
| 20 | Merch description_total_7 | 0.57707972 |

## VI. Preliminary models exploration

After having our final list of variables, we choose 10 out of the final 20 variables for model exploration. We first ran a linear logistic regression model as a baseline and explored with a few nonlinear models, which includes decision tree, random forest, LGBM, neural network, GBC, Catboost, XGBoost. The table below shows the results

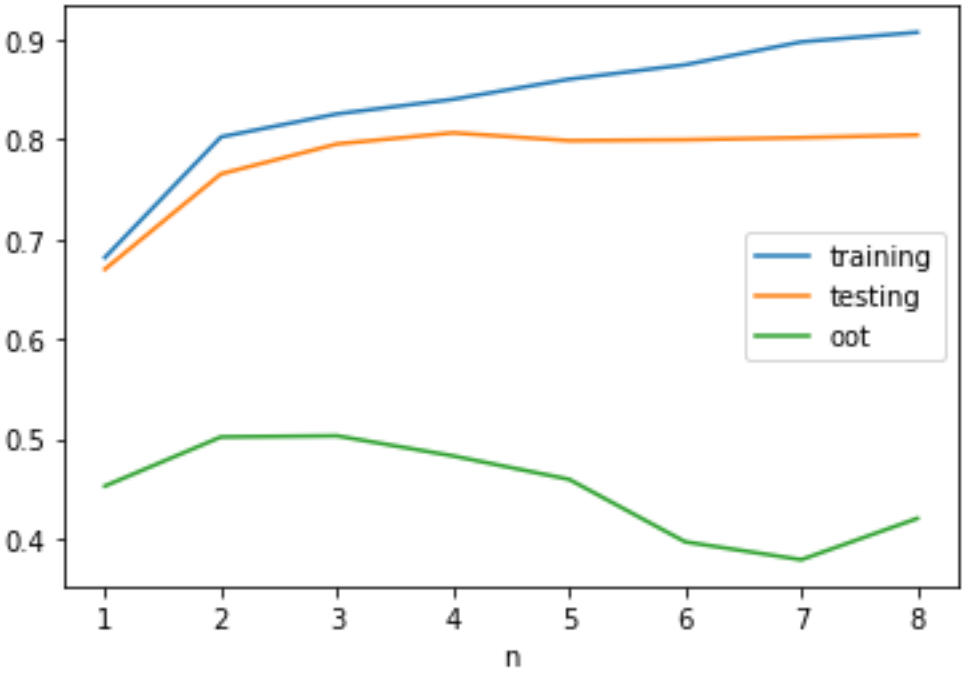
of multiple runs with different parameters and number of variables. We also attempted some overfitting results and presented it in the graphs below (including Catboost, Single DT, Random Forest, GBC and LGBM)

| Model | Parameter | | | | | | | Average FDR at 3% | | |
|---|---|---|---|---|---|---|---|---|---|---|
| **Logistic regression** | **# of variables** | **penalty** | **C** | **Solver** | | **l1_ratio** | | **Train** | **Test** | **OOT** |
| **1** | **10** | **none** | **none** | **none** | | **none** | | **0.657** | **0.626** | **0.306** |
| 2 | 10 | l1 | none | saga | | none | | 0.651 | 0.659 | 0.249 |
| 3 | 10 | l2 | 1 | saga | | none | | 0.655 | 0.645 | 0.254 |
| 5 | 10 | none | none | liblinear | | 0.4 | | 0.635 | 0.674 | 0.308 |
| **Decision Tree** | **# of variables** | **Max_depth** | | **min_sample_split** | | **min_sample_leafs** | | **Train** | **Test** | **OOT** |
| 1 | 10 | 5 | | 50 | | 30 | | 0.677 | 0.663 | 0.428 |
| **2** | **10** | **10** | | **40** | | **20** | | **0.812** | **0.772** | **0.383** |
| 3 | 10 | 15 | | 30 | | 10 | | 0.908 | 0.744 | 0.324 |
| 4 | 10 | 20 | | 20 | | 5 | | 0.980 | 0.734 | 0.291 |
| **Random Forest** | **# of variables** | **n_estimators** | **max_depth** | **min_sample_split** | | **min_samples_leaf** | **max_features** | **Train** | **Test** | **OOT** |
| **1** | **10** | **10** | **5** | **50** | | **30** | **3** | **0.748** | **0.721** | **0.512** |
| 2 | 10 | 30 | 15 | 40 | | 20 | 5 | 0.885 | 0.798 | 0.551 |
| 3 | 10 | 50 | 25 | 30 | | 10 | 8 | 0.993 | 0.797 | 0.451 |
| 4 | 10 | 100 | 30 | 20 | | 5 | 10 | 1.000 | 0.801 | 0.425 |
| **LightGBM** | **# of variables** | **n_estimators** | **max_depth** | **num_leaves** | **col_samplebytree** | **learning_rate** | **eval_metric** | **Train** | **Test** | **OOT** |
| 1 | 10 | 20 | 2 | 2 | 1 | 0.1 | none | 0.625 | 0.638 | 0.283 |
| **2** | **10** | **100** | **3** | **4** | **0.8** | **0.03** | **auc** | **0.768** | **0.767** | **0.485** |
| 3 | 10 | 500 | 5 | 8 | 0.8 | 0.01 | auc | 0.847 | 0.789 | 0.514 |
| 4 | 10 | 1000 | 6 | 10 | 0.8 | 0.01 | logloss | 0.893 | 0.828 | 0.448 |
| **Neural Network** | **# of variables** | **hidden_layer_size** | **activation** | **alpha** | **learning_rate** | **solver** | **learning_rate_init** | **Train** | **Test** | **OOT** |
| 1 | 10 | 5 | logistic | 0.1 | constant | adam | 0.01 | 0.651 | 0.659 | 0.349 |
| 2 | 10 | 10 | relu | 0.001 | constant | adam | 0.001 | 0.719 | 0.680 | 0.505 |
| 3 | 10 | 20 | relu | 0.0001 | adaptive | lbfgs | 0.0001 | 0.775 | 0.770 | 0.440 |
| **4** | **10** | (20,20,20) | **relu** | **0.0001** | **adaptive** | **lbfgs** | **0.0001** | **0.768** | **0.755** | **0.491** |
| **GBC** | **# of variables** | **n_estimators** | **max_depth** | **min_samples_leaf** | | **subsample** | | **Train** | **Test** | **OOT** |
| **1** | **10** | **10** | **5** | **1** | | **1** | | **0.779** | **0.747** | **0.379** |
| 2 | 10 | 50 | 10 | 3 | | 0.8 | | 0.975 | 0.785 | 0.326 |
| 3 | 10 | 100 | 15 | 5 | | 0.5 | | 0.974 | 0.730 | 0.324 |
| **Catboost** | **# of variables** | **bootstrap_type** | **verbose** | **max_depth** | | **iteration** | **random_state** | **Train** | **Test** | **OOT** |
| 1 | 10 | none | 0 | 2 | | 5 | None | 0.628 | 0.625 | 0.320 |
| 2 | 10 | Bayesian | 0 | 5 | | 10 | 10 | 0.738 | 0.728 | 0.444 |
| **3** | **10** | **Bayesian** | **0** | **16** | | **15** | **10** | **0.805** | **0.761** | **0.535** |
| 4 | 10 | MVS | 0 | 16 | | 20 | 5 | 0.830 | 0.773 | 0.488 |
| **XGBoost** | **# of variables** | **max_depth** | **n_estimators** | **tree_method** | | **subsample** | | **Train** | **Test** | **OOT** |
| 1 | 10 | 2 | 5 | auto | | 1 | | 0.612 | 0.625 | 0.359 |
| **6** | **10** | **5** | **25** | **auto** | | **0.8** | | **0.792** | **0.769** | **0.479** |
| 2 | 10 | 10 | 50 | approx | | 0.8 | | 0.894 | 0.813 | 0.474 |
| 3 | 10 | 20 | 100 | auto | | 0.5 | | 0.963 | 0.814 | 0.370 |
| 5 | 10 | 20 | 1000 | hist | | 1 | | 1.000 | 0.778 | 0.333 |



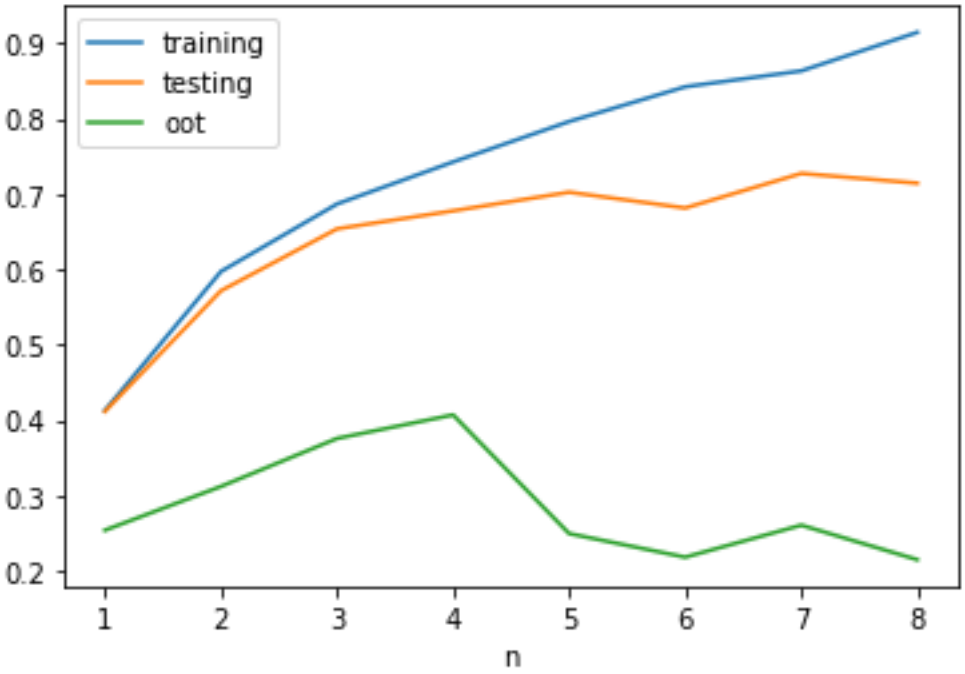

**Graph 1: Cat Boost**

Range (1,16,2)
Max_depth = 16
Iterations = 100
Verbose = 0
Random_state = None
Bootstrap_type = ‘Bayesian’




**Graph 2: Single DT**

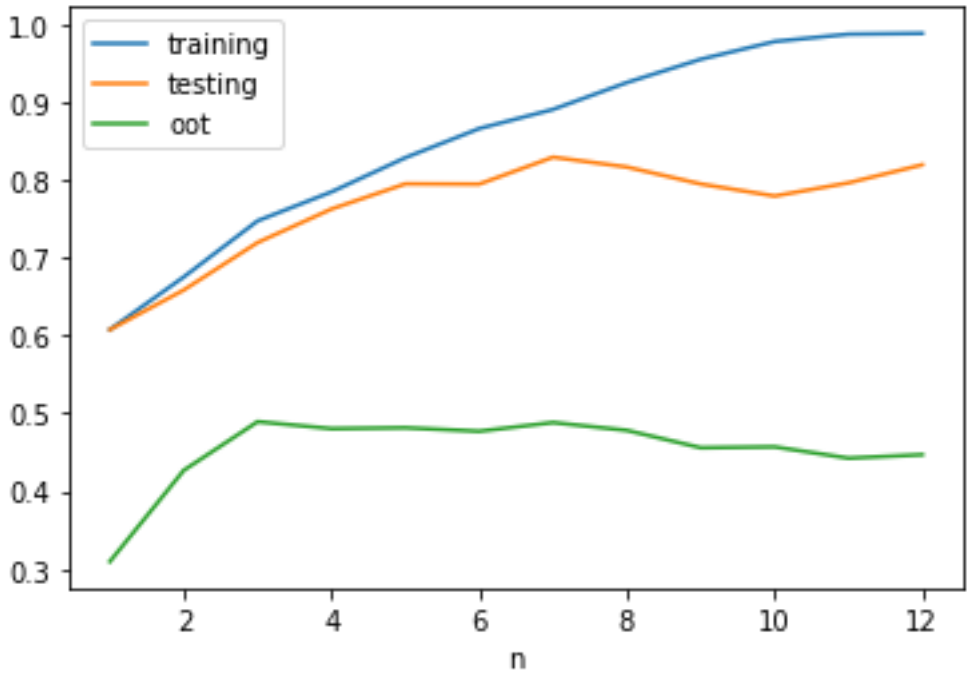
Range (1,16,2)
Max_depth = 16
Min_samples_split = 15
Min_samples_leaf = 5




**Graph 3: Random Forest**

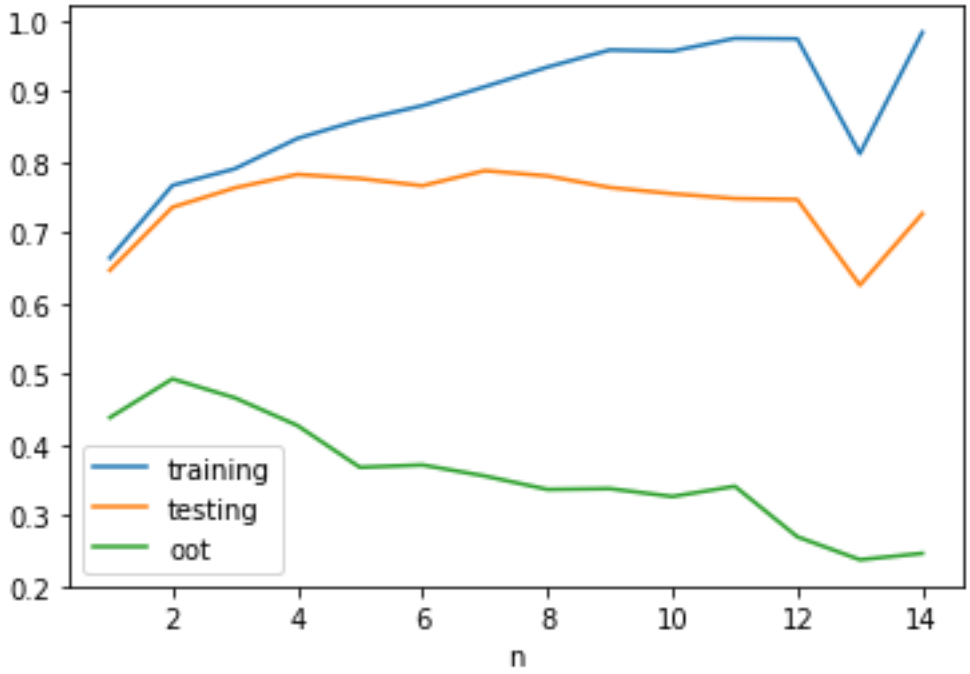
Range (1,25,2)
Max_depth = 25

N_estimators= 50
Min_samples_split = 30
Min_samples_leaf = 10
Max_features = 8



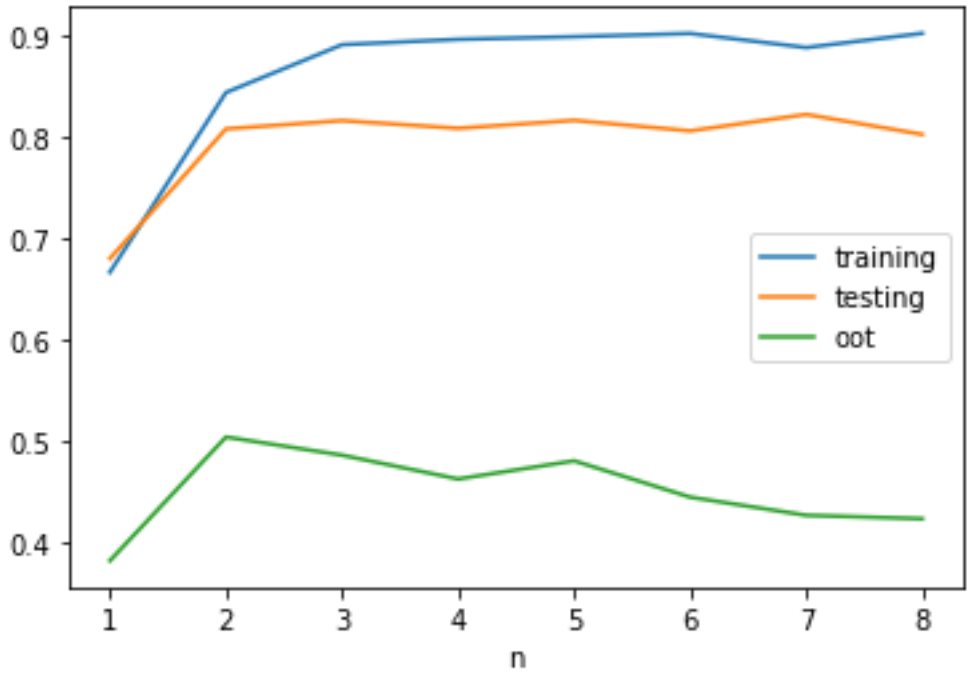

**Graph 4: GBC**

Range (1,15,1)
Max_depth = 15
N_estimators = 100
Min_samples_leaf = 5
Subsample = 0.5



**Graph 5: LGBM**

Range (1,16,2)
Max_depth = 16
N_estimators = 1000
Num_leaves = 10
Learning_rate = 0.01




## VII. Final Model Performance - Summary of results

**The final model we choose is CatBoost with the following parameters:**

*Number of variables = 10, max_depth = 16, iterations = 15, random_state=10, bootstrap_type = 'Bayesian'*

**RESULTS**

| | trn | tst | oot |
|---|---|---|---|
| 0 | 0.8037676609 | 0.7654320988 | 0.4581005587 |
| 1 | 0.8152350081 | 0.7604562738 | 0.4860335196 |
| 2 | 0.7961165049 | 0.7442748092 | 0.5251396648 |
| 3 | 0.8166666667 | 0.7821428571 | 0.5251396648 |
| 4 | 0.7940199336 | 0.7446043165 | 0.5139664804 |

**The reason we chose this model is because the average train of this model is close to 0.8, and test of this model is close to 0.75, and OOT is near to 0.5.**

Below are the three tables of the final model results (the first 20 bins of train, test, and OOT).

**Total of Goods =** 58408
**Total of Bads =** 602
**Fraud Rate (%Cumulative Bad) =** 0.01

**TRAIN**

| Statistics by bin | | | | | | | Cumulative statistics | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| bin | #records | #good | #bad | %good | %bad | total | cumulative good | cumulative bad | %cumulative good | FDR | KS | FPR |
| 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| 1.00 | 590.00 | 189.00 | 401.00 | 32.03 | 67.97 | 590.00 | 189.00 | 401.00 | 0.32 | 66.61 | 66.29 | 0.47 |
| 2.00 | 590.00 | 535.00 | 55.00 | 90.68 | 9.32 | 1180.00 | 724.00 | 456.00 | 1.24 | 75.75 | 74.51 | 1.59 |
| 3.00 | 590.00 | 568.00 | 22.00 | 96.27 | 3.73 | 1770.00 | 1292.00 | 478.00 | 2.21 | 79.40 | 77.19 | 2.70 |
| 4.00 | 590.00 | 576.00 | 14.00 | 97.63 | 2.37 | 2360.00 | 1868.00 | 492.00 | 3.20 | 81.73 | 78.53 | 3.80 |
| 5.00 | 590.00 | 571.00 | 19.00 | 96.78 | 3.22 | 2950.00 | 2439.00 | 511.00 | 4.18 | 84.88 | 80.71 | 4.77 |
| 6.00 | 591.00 | 584.00 | 7.00 | 98.82 | 1.18 | 3541.00 | 3023.00 | 518.00 | 5.18 | 86.05 | 80.87 | 5.84 |
| 7.00 | 590.00 | 583.00 | 7.00 | 98.81 | 1.19 | 4131.00 | 3606.00 | 525.00 | 6.17 | 87.21 | 81.04 | 6.87 |
| 8.00 | 590.00 | 581.00 | 9.00 | 98.47 | 1.53 | 4721.00 | 4187.00 | 534.00 | 7.17 | 88.70 | 81.54 | 7.84 |
| 9.00 | 590.00 | 586.00 | 4.00 | 99.32 | 0.68 | 5311.00 | 4773.00 | 538.00 | 8.17 | 89.37 | 81.20 | 8.87 |
| 10.00 | 590.00 | 587.00 | 3.00 | 99.49 | 0.51 | 5901.00 | 5360.00 | 541.00 | 9.18 | 89.87 | 80.69 | 9.91 |
| 11.00 | 590.00 | 586.00 | 4.00 | 99.32 | 0.68 | 6491.00 | 5946.00 | 545.00 | 10.18 | 90.53 | 80.35 | 10.91 |
| 12.00 | 590.00 | 589.00 | 1.00 | 99.83 | 0.17 | 7081.00 | 6535.00 | 546.00 | 11.19 | 90.70 | 79.51 | 11.97 |
| 13.00 | 590.00 | 587.00 | 3.00 | 99.49 | 0.51 | 7671.00 | 7122.00 | 549.00 | 12.19 | 91.20 | 79.00 | 12.97 |
| 14.00 | 590.00 | 584.00 | 6.00 | 98.98 | 1.02 | 8261.00 | 7706.00 | 555.00 | 13.19 | 92.19 | 79.00 | 13.88 |
| 15.00 | 591.00 | 587.00 | 4.00 | 99.32 | 0.68 | 8852.00 | 8293.00 | 559.00 | 14.20 | 92.86 | 78.66 | 14.84 |
| 16.00 | 590.00 | 588.00 | 2.00 | 99.66 | 0.34 | 9442.00 | 8881.00 | 561.00 | 15.21 | 93.19 | 77.98 | 15.83 |
| 17.00 | 590.00 | 590.00 | 0.00 | 100.00 | 0.00 | 10032.00 | 9471.00 | 561.00 | 16.22 | 93.19 | 76.97 | 16.88 |
| 18.00 | 590.00 | 589.00 | 1.00 | 99.83 | 0.17 | 10622.00 | 10060.00 | 562.00 | 17.22 | 93.36 | 76.13 | 17.90 |
| 19.00 | 590.00 | 589.00 | 1.00 | 99.83 | 0.17 | 11212.00 | 10649.00 | 563.00 | 18.23 | 93.52 | 75.29 | 18.91 |
| 20.00 | 590.00 | 586.00 | 4.00 | 99.32 | 0.68 | 11802.00 | 11235.00 | 567.00 | 19.24 | 94.19 | 74.95 | 19.81 |

**Total of Goods =** 25012
**Total of Bads =** 278
**Fraud Rate (%Cumulative Bad) =** 0.011

**TEST**

| Statistics by bin | | | | | | | Cumulative statistics | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| bin | #records | #good | #bad | %good | %bad | total | cumulative good | cumulative bad | %cumulative good | FDR | KS | FPR |
| 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| 1.00 | 253.00 | 92.00 | 161.00 | 36.36 | 63.64 | 253.00 | 92.00 | 161.00 | 0.37 | 57.91 | 57.55 | 0.57 |
| 2.00 | 253.00 | 218.00 | 35.00 | 86.17 | 13.83 | 506.00 | 310.00 | 196.00 | 1.24 | 70.50 | 69.26 | 1.58 |
| 3.00 | 253.00 | 242.00 | 11.00 | 95.65 | 4.35 | 759.00 | 552.00 | 207.00 | 2.21 | 74.46 | 72.25 | 2.67 |
| 4.00 | 253.00 | 242.00 | 11.00 | 95.65 | 4.35 | 1012.00 | 794.00 | 218.00 | 3.17 | 78.42 | 75.24 | 3.64 |
| 5.00 | 252.00 | 246.00 | 6.00 | 97.62 | 2.38 | 1264.00 | 1040.00 | 224.00 | 4.16 | 80.58 | 76.42 | 4.64 |
| 6.00 | 253.00 | 250.00 | 3.00 | 98.81 | 1.19 | 1517.00 | 1290.00 | 227.00 | 5.16 | 81.65 | 76.50 | 5.68 |
| 7.00 | 253.00 | 249.00 | 4.00 | 98.42 | 1.58 | 1770.00 | 1539.00 | 231.00 | 6.15 | 83.09 | 76.94 | 6.66 |
| 8.00 | 253.00 | 249.00 | 4.00 | 98.42 | 1.58 | 2023.00 | 1788.00 | 235.00 | 7.15 | 84.53 | 77.38 | 7.61 |
| 9.00 | 253.00 | 251.00 | 2.00 | 99.21 | 0.79 | 2276.00 | 2039.00 | 237.00 | 8.15 | 85.25 | 77.10 | 8.60 |
| 10.00 | 253.00 | 248.00 | 5.00 | 98.02 | 1.98 | 2529.00 | 2287.00 | 242.00 | 9.14 | 87.05 | 77.91 | 9.45 |
| 11.00 | 253.00 | 251.00 | 2.00 | 99.21 | 0.79 | 2782.00 | 2538.00 | 244.00 | 10.15 | 87.77 | 77.62 | 10.40 |
| 12.00 | 253.00 | 251.00 | 2.00 | 99.21 | 0.79 | 3035.00 | 2789.00 | 246.00 | 11.15 | 88.49 | 77.34 | 11.34 |
| 13.00 | 253.00 | 250.00 | 3.00 | 98.81 | 1.19 | 3288.00 | 3039.00 | 249.00 | 12.15 | 89.57 | 77.42 | 12.20 |
| 14.00 | 253.00 | 253.00 | 0.00 | 100.00 | 0.00 | 3541.00 | 3292.00 | 249.00 | 13.16 | 89.57 | 76.41 | 13.22 |
| 15.00 | 253.00 | 252.00 | 1.00 | 99.60 | 0.40 | 3794.00 | 3544.00 | 250.00 | 14.17 | 89.93 | 75.76 | 14.18 |
| 16.00 | 252.00 | 252.00 | 0.00 | 100.00 | 0.00 | 4046.00 | 3796.00 | 250.00 | 15.18 | 89.93 | 74.75 | 15.18 |
| 17.00 | 253.00 | 250.00 | 3.00 | 98.81 | 1.19 | 4299.00 | 4046.00 | 253.00 | 16.18 | 91.01 | 74.83 | 15.99 |
| 18.00 | 253.00 | 253.00 | 0.00 | 100.00 | 0.00 | 4552.00 | 4299.00 | 253.00 | 17.19 | 91.01 | 73.82 | 16.99 |
| 19.00 | 253.00 | 253.00 | 0.00 | 100.00 | 0.00 | 4805.00 | 4552.00 | 253.00 | 18.20 | 91.01 | 72.81 | 17.99 |
| 20.00 | 253.00 | 250.00 | 3.00 | 98.81 | 1.19 | 5058.00 | 4802.00 | 256.00 | 19.20 | 92.09 | 72.89 | 18.76 |

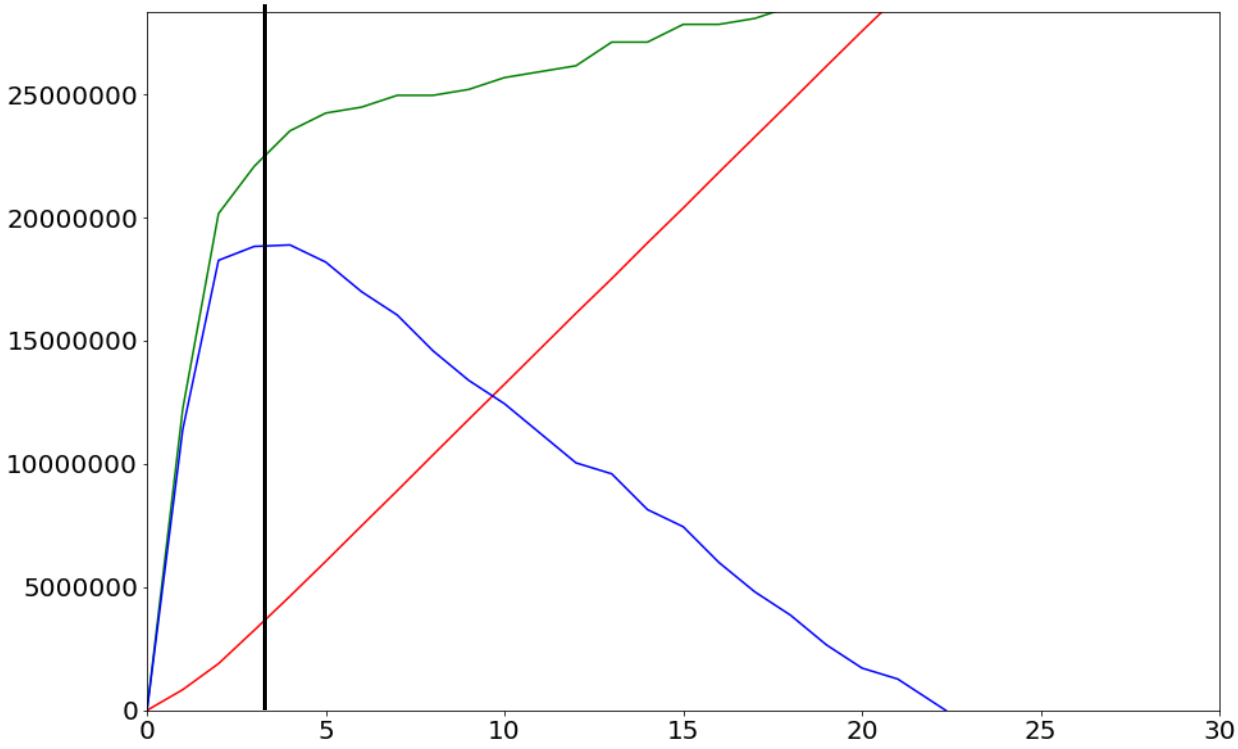
**Total of Goods =** 11918
**Total of Bads =** 179
**Fraud Rate (%Cumulative Bad) =** 0.015

**OOT**

| Statistics by bin | | | | | | | Cumulative statistics | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| bin | #records | #good | #bad | %good | %bad | total | cumulative good | cumulative bad | %cumulative good | FDR | KS | FPR |
| 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| 1.00 | 121.00 | 72.00 | 49.00 | 59.50 | 40.50 | 121.00 | 72.00 | 49.00 | 0.60 | 27.37 | 26.77 | 1.47 |
| 2.00 | 121.00 | 96.00 | 25.00 | 79.34 | 20.66 | 242.00 | 168.00 | 74.00 | 1.41 | 41.34 | 39.93 | 2.27 |
| 3.00 | 121.00 | 103.00 | 18.00 | 85.12 | 14.88 | 363.00 | 271.00 | 92.00 | 2.27 | 51.40 | 49.12 | 2.95 |
| 4.00 | 121.00 | 117.00 | 4.00 | 96.69 | 3.31 | 484.00 | 388.00 | 96.00 | 3.26 | 53.63 | 50.38 | 4.04 |
| 5.00 | 121.00 | 115.00 | 6.00 | 95.04 | 4.96 | 605.00 | 503.00 | 102.00 | 4.22 | 56.98 | 52.76 | 4.93 |
| 6.00 | 121.00 | 120.00 | 1.00 | 99.17 | 0.83 | 726.00 | 623.00 | 103.00 | 5.23 | 57.54 | 52.31 | 6.05 |
| 7.00 | 121.00 | 118.00 | 3.00 | 97.52 | 2.48 | 847.00 | 741.00 | 106.00 | 6.22 | 59.22 | 53.00 | 6.99 |
| 8.00 | 121.00 | 116.00 | 5.00 | 95.87 | 4.13 | 968.00 | 857.00 | 111.00 | 7.19 | 62.01 | 54.82 | 7.72 |
| 9.00 | 121.00 | 118.00 | 3.00 | 97.52 | 2.48 | 1089.00 | 975.00 | 114.00 | 8.18 | 63.69 | 55.51 | 8.55 |
| 10.00 | 121.00 | 120.00 | 1.00 | 99.17 | 0.83 | 1210.00 | 1095.00 | 115.00 | 9.19 | 64.25 | 55.06 | 9.52 |
| 11.00 | 121.00 | 118.00 | 3.00 | 97.52 | 2.48 | 1331.00 | 1213.00 | 118.00 | 10.18 | 65.92 | 55.74 | 10.28 |
| 12.00 | 121.00 | 121.00 | 0.00 | 100.00 | 0.00 | 1452.00 | 1334.00 | 118.00 | 11.19 | 65.92 | 54.73 | 11.31 |
| 13.00 | 121.00 | 119.00 | 2.00 | 98.35 | 1.65 | 1573.00 | 1453.00 | 120.00 | 12.19 | 67.04 | 54.85 | 12.11 |
| 14.00 | 121.00 | 119.00 | 2.00 | 98.35 | 1.65 | 1694.00 | 1572.00 | 122.00 | 13.19 | 68.16 | 54.97 | 12.89 |
| 15.00 | 121.00 | 119.00 | 2.00 | 98.35 | 1.65 | 1815.00 | 1691.00 | 124.00 | 14.19 | 69.27 | 55.09 | 13.64 |
| 16.00 | 121.00 | 121.00 | 0.00 | 100.00 | 0.00 | 1936.00 | 1812.00 | 124.00 | 15.20 | 69.27 | 54.07 | 14.61 |
| 17.00 | 120.00 | 117.00 | 3.00 | 97.50 | 2.50 | 2056.00 | 1929.00 | 127.00 | 16.19 | 70.95 | 54.76 | 15.19 |
| 18.00 | 121.00 | 118.00 | 3.00 | 97.52 | 2.48 | 2177.00 | 2047.00 | 130.00 | 17.18 | 72.63 | 55.45 | 15.75 |
| 19.00 | 121.00 | 121.00 | 0.00 | 100.00 | 0.00 | 2298.00 | 2168.00 | 130.00 | 18.19 | 72.63 | 54.43 | 16.68 |
| 20.00 | 121.00 | 120.00 | 1.00 | 99.17 | 0.83 | 2419.00 | 2288.00 | 131.00 | 19.20 | 73.18 | 53.99 | 17.47 |

## VIII. Financial Curve and Recommended Cutoff:

After plotting the financial plots, we noticed that the peak of the cost saving line (blue line) falls around 3% FDR rate. Therefore, we recommend the cutoff to be FDR@3%, where it helps to save around **18 million** per year.




## IX. SUMMARY

We have executed a comprehensive process to construct a credit card transaction fraud detection model, involving steps from data cleaning, variable creation, feature selection, to initial model exploration. Our ultimate model achieved an estimated cost savings of $18 million at FDR@3%.

Nonetheless, there is potential for further improvement in the model. Apart from the variables we built, we could explore additional variables from different entities to see if there is an improvement to the model. Even though our final model seems reasonable, we could have done more fine-tuning to the final model, which could help optimize cost savings (up to ~20M). In addition, if we could have more data for training and testing, the model will be more accurate.

## X. APPENDIX (DQR)

**Data Quality Report**

### 1. Data description

The data includes a collection of about 100,000 Credit Card transactions. There are in total 10 fields. Including both numeric and categorical ones. The records are collected from January 1st, 2010, to December 31st ,2010, with the total of 96,753 rows. This dataset will be used to identify patterns and trends in fraudulent transactions and to develop a fraud detection algorithm that can accurately identify and prevent fraudulent transactions.

### 2. Summary Tables

c. Numerical Table

| Field | % Populated | Min | Max | Mean | Standard Deviation | %Zero |
|---|---|---|---|---|---|---|
| Date | 100% | 2010-01-01 | 2010-12-31 | n/a | n/a | 0% |

| | | | | | | |
|---|---|---|---|---|---|---|
| Amount | 100% | 0.01 | 3102045.53 | 427.89 | 10,006.14 | 0% |

d. Categorical table

| Field | % Populated | # Blank | # Zeros | # Unique Values | Most Common Value |
|---|---|---|---|---|---|
| Recnum | 100% | 0 | 0 | 96,753 | 1 |
| Cardnum | 100% | 0 | 0 | 1,645 | 5142148452 |
| Merchnum | 96.51% | 3375 | 231 | 13,091 | 930090121224 |
| Merch description | 100% | 0 | 3 | 13,126 | GSA-FSS-ADV |
| Merch state | 100% | 1195 | 0 | 227 | TN |
| Merch zip | 100% | 4656 | 0 | 4,567 | 38118 |
| Transtype | 100% | 0 | 0 | 4 | P |
| Fraud | 100% | 0 | 0 | 2 | 0 |

## 3. Data Visualization

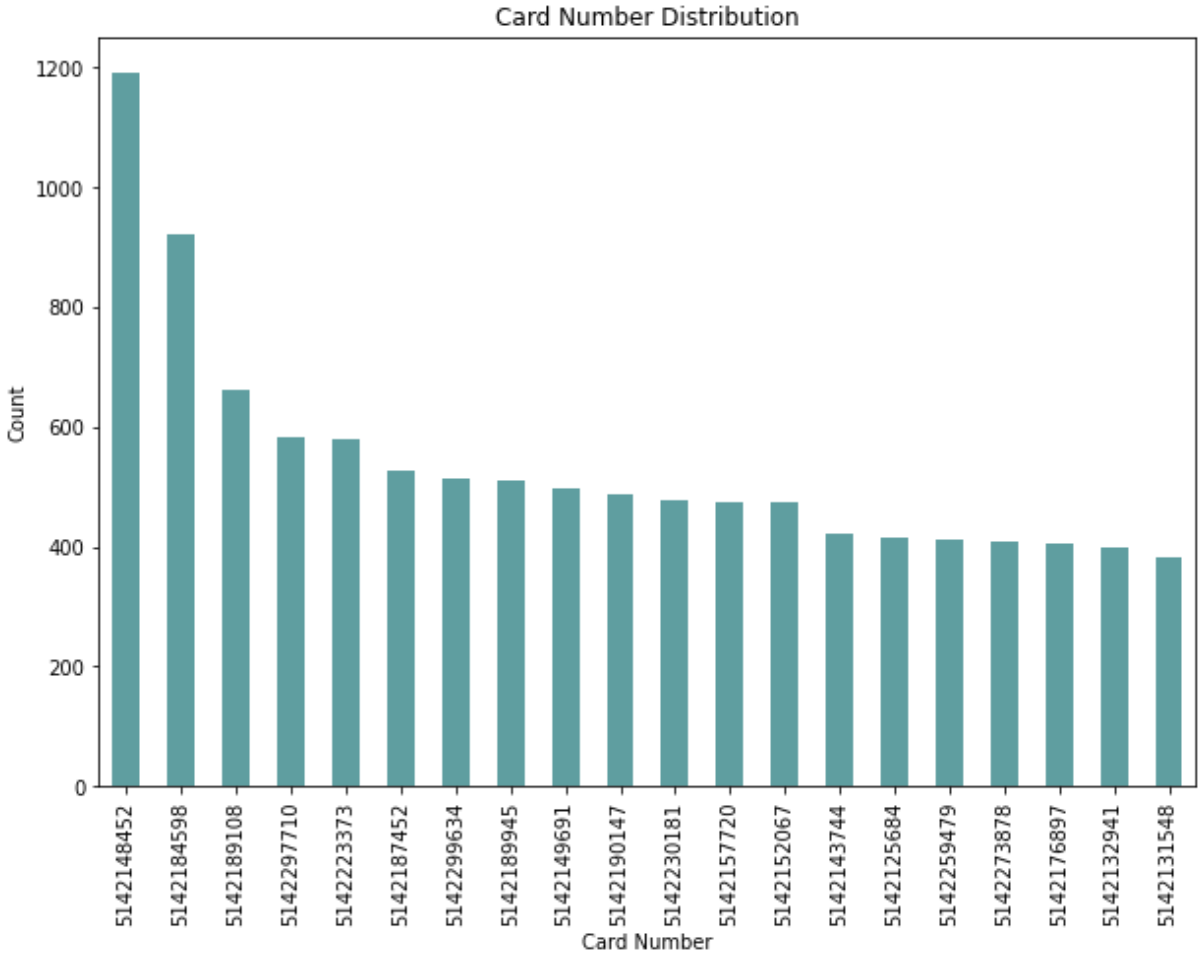
a. **Recnum**

Record numbers assigns a unique identification number to each transaction record. They are ordinal numbers from 1 to 96,753. Each row is one unique positive number.

b. **Cardnum**

Cardnum identifies the credit card used for the transaction. It can be used to track the history of transactions for a specific card and to identify any unusual spending patterns that could indicate fraud.

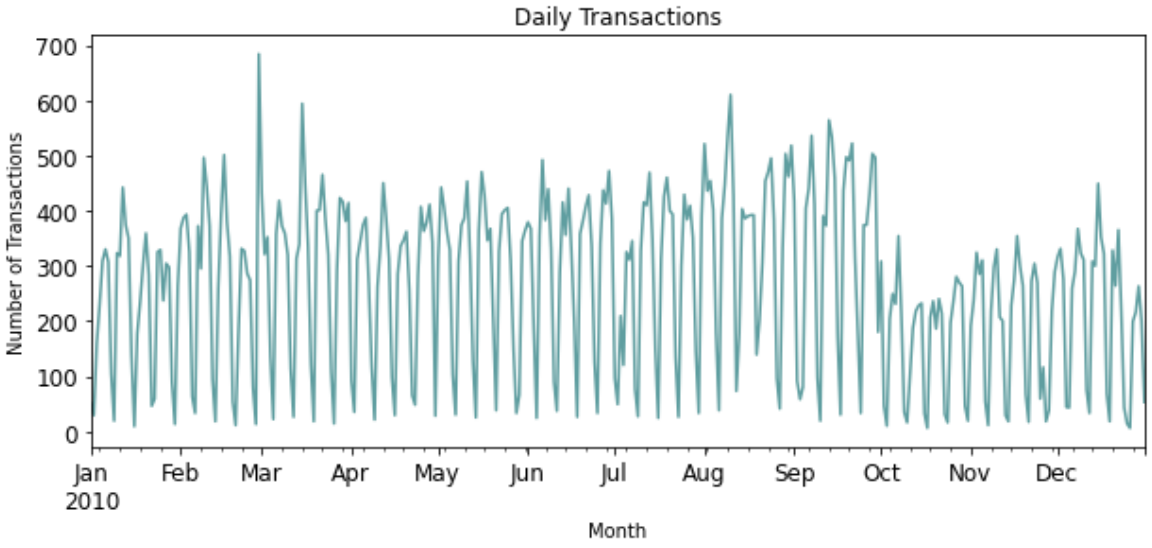
The most common value is 5142148452 with 1192 records. The chart shows the 20 most common values of card numbers, among a total of 1,645 unique ones.

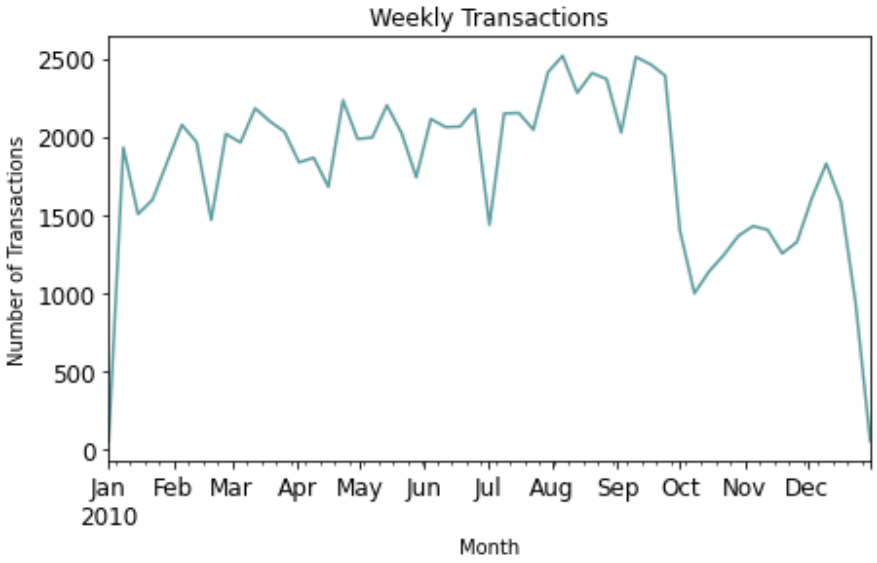



c. **Date**

The date of the transaction is recorded for each transaction. It can be used to analyze the timing of transactions and identify any unusual patterns, such as a sudden surge in transactions from a particular merchant.

From the dataset, we can observe the frequency of the transactions coming in by months. The frequency is shown in the charts below, by both daily and weekly.

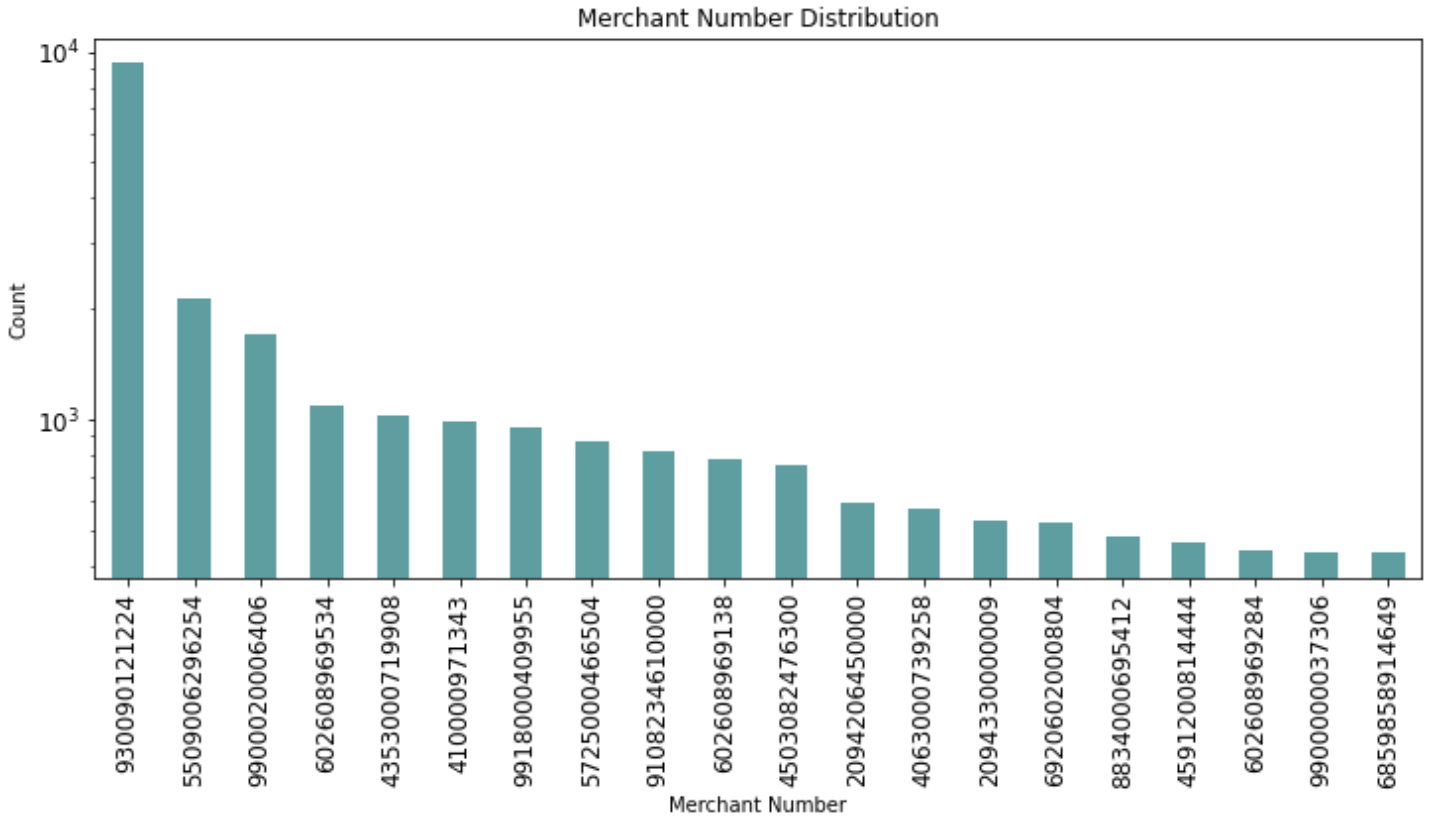





d. **Merchnum**

Merchnum identifies the merchant where the transaction took place. It can be used to track the history of transactions for a specific merchant and to identify any unusual patterns.

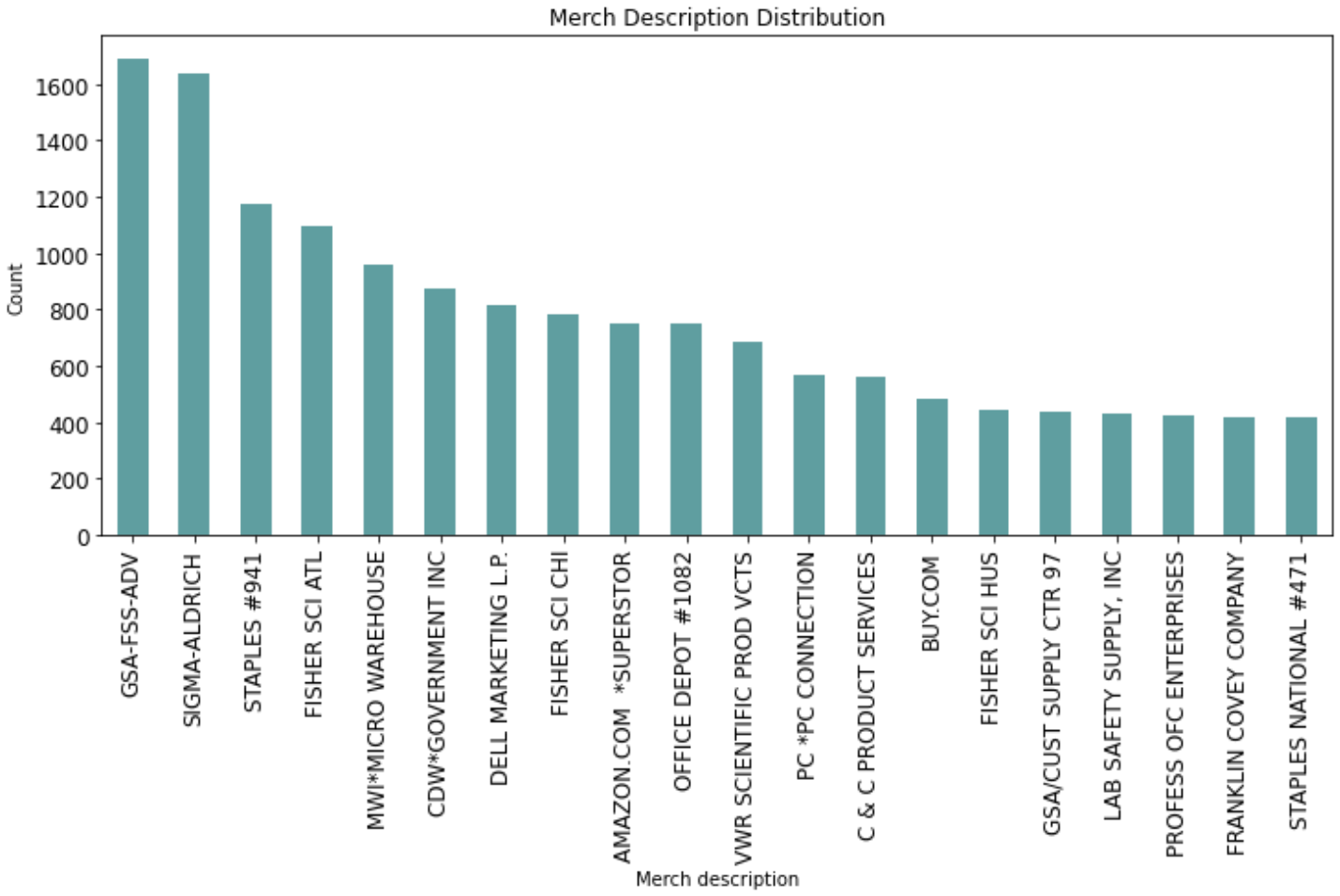
The most common merchant number is 930090121224 with 9310 records. This chart shows the 20-merchant number with the highest count, among a total of 13,091 unique merchant numbers.



e. **Merch description**

Merch description provides a brief description of the merchant where the transaction took place. It can be used to group similar transactions together and identify any unusual patterns across different merchants.

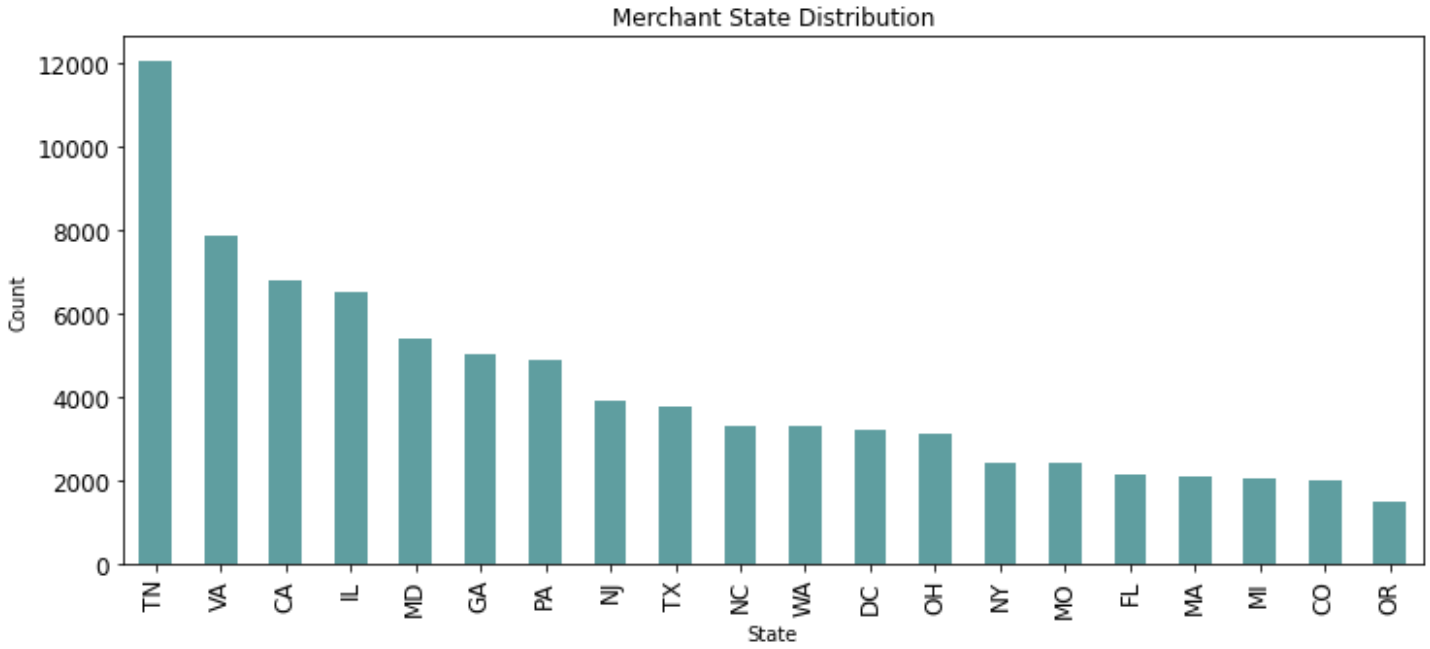
The most common merchant description is GSA-FSS-ADV with 1688 records. This chart shows the 20-merchant description with the highest count, among a total of 13,126 unique Merchant descriptions.




f. **Merch state**

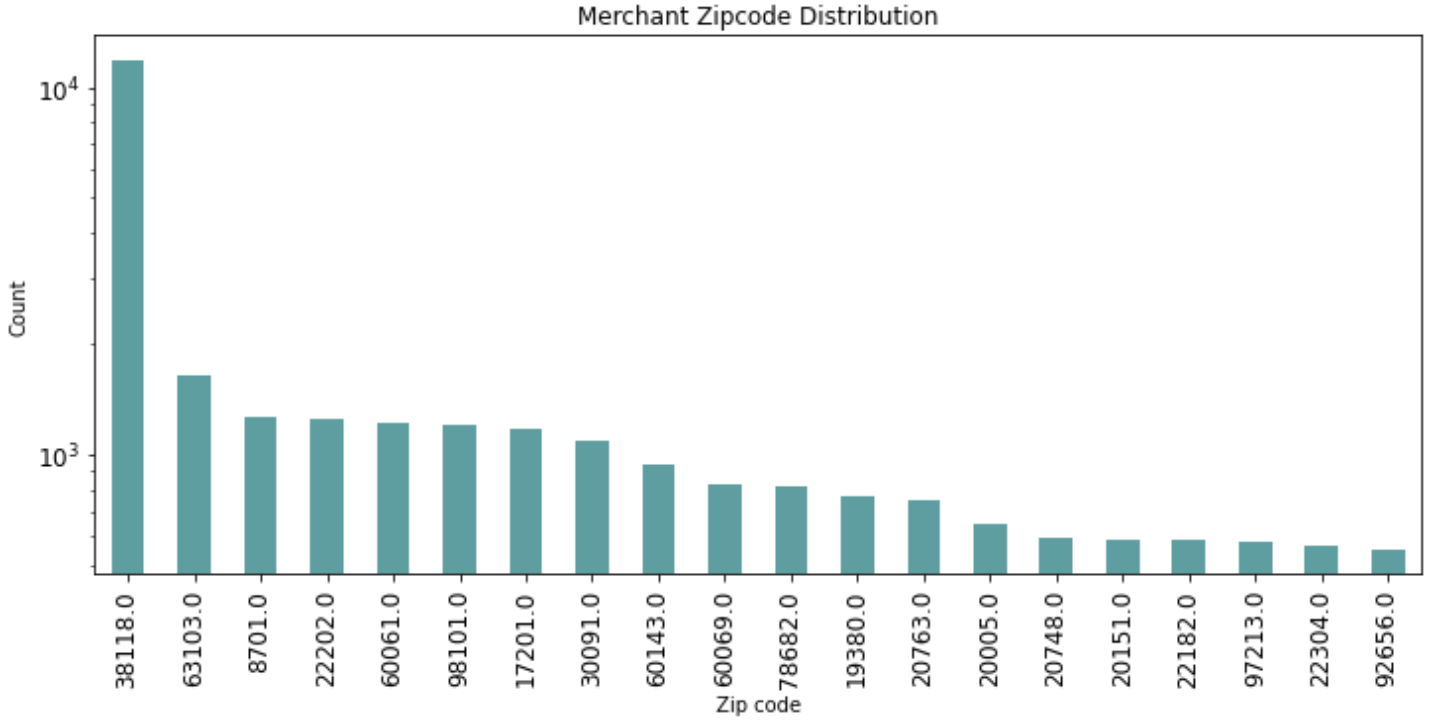
Merch state specifies the state where the merchant is located, which could be used to identify any regional trends or patterns in fraud activity.

TN is the most common value with 12035 records. This organization appears to be based in Tennessee (TN). This chart shows the 20 states with the highest count, among a total of 227 unique states.



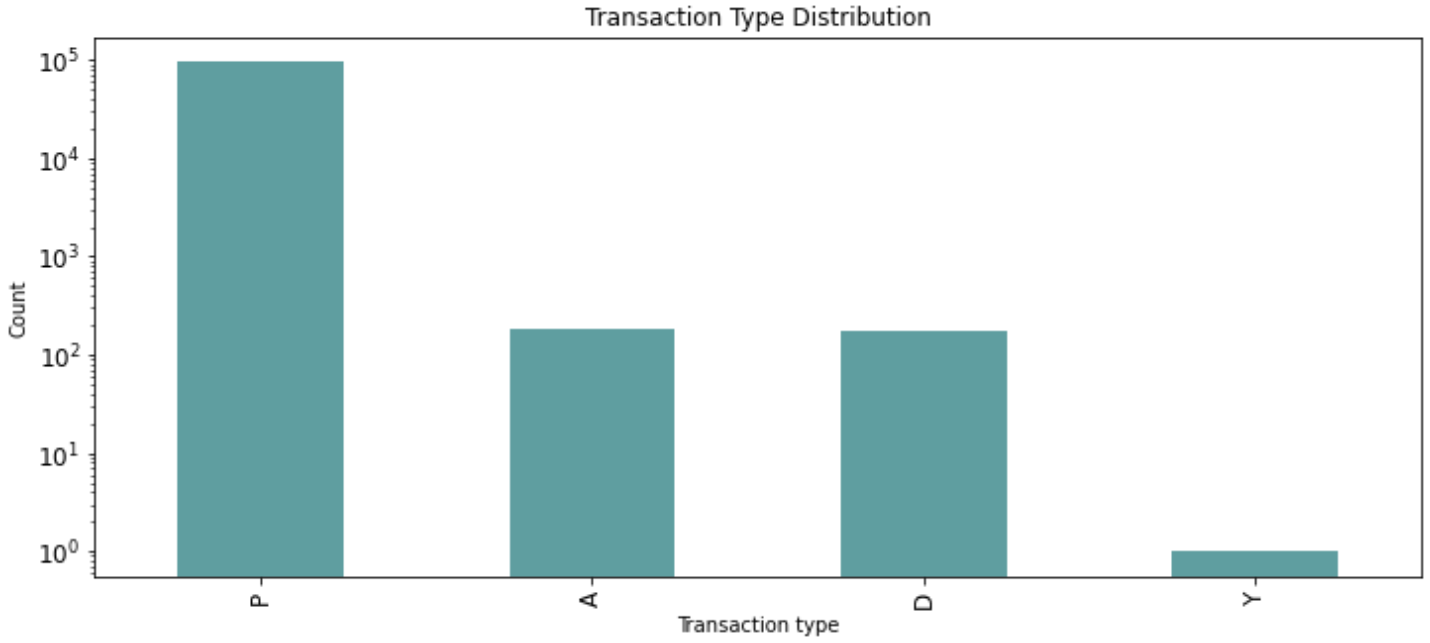

g. **Merch zip**

Merch zip provides the zip code where the merchant is located. Zip code '38118' is the most common value with a total of 11868 records. This chart shows the 20 zip codes with the highest count, among a total of 4567 unique zip codes.




h. **Transtype**

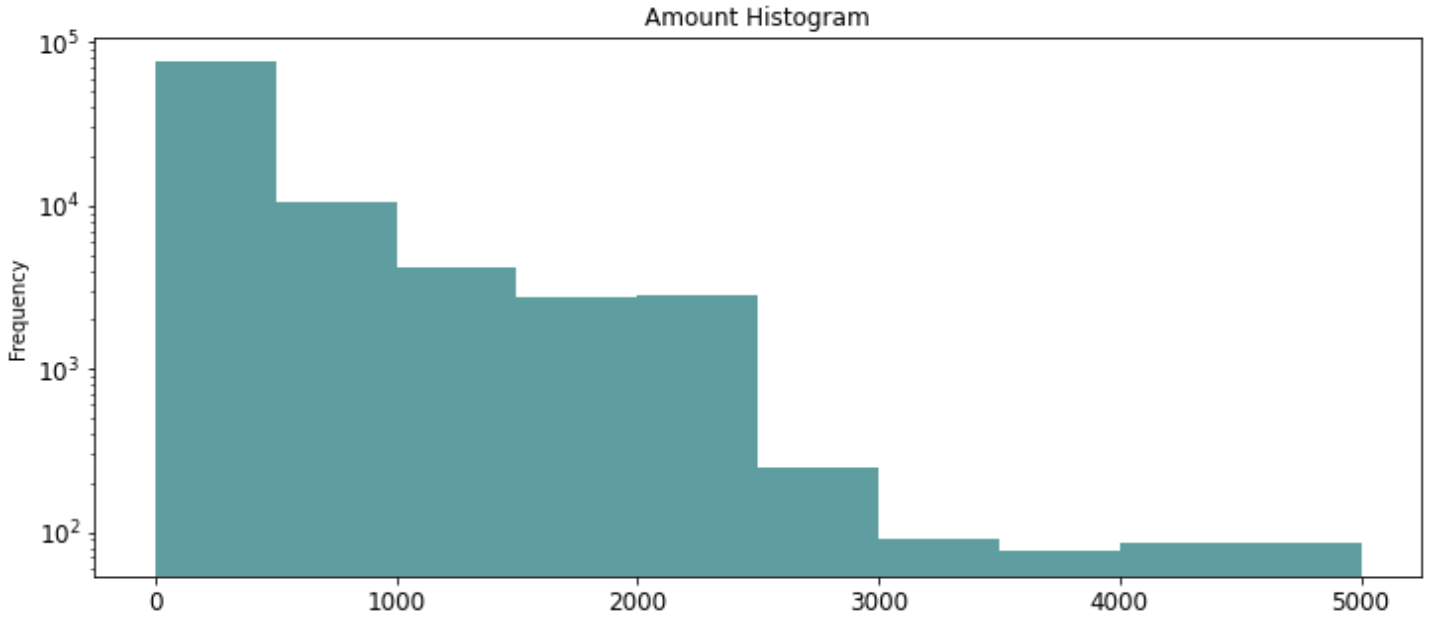
There are 4 main types of transactions (P, A, D, Y). ‘P’ is the most common value with 96,398 records.




i. **Amount**

Amount indicates the dollar amount of the transaction. It can be used to identify any unusually high-value transactions or patterns of transactions that fall outside the typical spending habits of the cardholder.

Below is the histogram of the amount field. The minimum amount is 0.01 and the maximum amount is 3102045.53, with a mean of 427.89. We only chose to show the distribution of different amounts ranging from 0 to 5000.




j. **Fraud**

Fraud value is a Boolean field with a value of 0 or 1, where 1 represents a potential fraud and 0 means it is not a potential fraud. The percentage of the applications identified as a potential fraud is around 1.09%


Below charts demonstrates monthly frequency of 'good' and 'bad' transactions; bad indicates a potential fraud and good indicates otherwise. In the chart, red line represents the frequency of a 'bad' transaction and green line represents the frequency of a 'good' transaction. 'Good' transactions appear to have a relatively steady frequency while 'bad' transactions tend to fluctuate more.