Project Informal Report

Cristian Barahona

1. Executive Summary

The following report provides valuable insights into the effectiveness of implementing a predictive model in the business' fraud detection system. The model selected utilizes a Random Forest machine learning algorithm to identify potentially fraudulent transactions and alert them in real time processing. The report highlights the accuracy and efficiency of the final model, as well as the financial implications of selecting different cutoffs scores for fraud detection. We recommend setting up a **cutoff score of 3%** for fraud detection in out-of-time transactions, as it provides a near-optimal performance, while still providing a margin of safety. The out-of-time fraud detection rate at 3% for the selected model is **0.577**. This recommendation leads to expected savings of approximately **$21,000,000** in OOT transactions when projected for the entire year. Overall, the report demonstrates the importance of utilizing advanced analytics tools to combat fraud and protect the company's financial assets.

2. Data Description

The dataset is Credit Card Transaction Data, which contains information of US government organization card transactions, in the year 2010. The data is a collection of 96,753 records including 10 fields.

The following tables show summarized statistics for the 2 numeric and 8 categorical fields in the dataset.

a. Numeric Fields Table

Field Name	# Records With Values	% Populated	# Zeros	Min	Max	Mean	Most Common	Stdev
Date	96,753	100%	0	1/1/2010	12/31/2010	6/25/2010	2/28/2010	98 days 21:38:57
Amount	96,753	100%	0	0.01	3,102,046	427.9	3.62	10,006

b. Categorical Fields Table

Field Name	# Records With Values	% Populated	# Zeros	# Unique Values	Most Common
Recnum	96,753	100.00%	0	96,753	1
Cardnum	96,753	100.00%	0	1,645	5142148452
Merchnum	93,378	96.51%	0	13,091	930090121224
Merch description	96,753	100.00%	0	13,126	GSA-FSS-ADV
Merch state	95,558	98.76%	0	227	TN
Merch zip	92,097	95.19%	0	4,567	38118
Transtype	96,753	100.00%	0	4	P
Fraud	96,753	100.00%	95,694	2	0

3. Data Cleaning

The first step in building a predictive model for fraud detection is to clean the data.

We first filter our data to consider **exclusions**. In this case, we remove the outlier transaction (<3,000,000) and include only purchases.

Now, we look at our raw data types and null value counts:

Field	# Null
Recnum	0
Cardnum	0
Date	0
Merchnum	3198
Merch description	0
Merch state	1020
Merch zip	4300
Transtype	0
Amount	0
Fraud	0

- We see that \`***Merchnum***\`, \`***Merch state***\`, \`***Merch zip***\`, have null values that need to be addressed. To do this we create a dictionary that maps merchant numbers to their descriptions, for all non-null pairs.
- Then, we fill null merchant numbers that have a mapped description. (i.e., merchant numbers that can be identified, since their description is linked to a merchant number in other observations)
- Similar to the logic used above, we map zip codes to states to identify null state values.
- We change non-US states to 'foreign', and fill all the remaining values of '**Merch state**' by 'unknown'.
- We follow the same logic to map '**Merch zip**' to their descriptions, and finally change all missing values to 'unknown'.

4. Variable Creation

The following table summarizes the variables created by category. A total of 2,227 variables were created in this process:

Description	# Variables Created
Day of the week target encoded: average fraud percentage of that day	1
Day Since: Number of days since a transaction with that entity was seen	21
Amount: {Avg,max,median,total,actual/avg,actual/max,actual/med,actual/total} transaction amount with that entity over the past {0, 1, 3, 7, 14, 30, 60} days	1176
Frequency: Number of transactions with that entity over the past {0, 1, 3, 7, 14, 30, 60} days	147
Velocity Change: Number/Amount of transactions with that entity seen in the past {0,1} days divided by the avg. number of transactions with the same entity over the past {7,14,30,60} days	336
Ratio: Number of transactions with that entity seen in the past {0,1} days divided by the number of days since the last transaction.	168
Variability: Amount variability (difference) in transactions with that entity seen in the past {7,14,30,60} days	378

5. Feature Selection

The main goal is to **reduce dimensionality** by taking the candidate variables created previously and selecting the **most relevant** variables for fraud prediction. To do so, we are going to first run a **filter,** which ranks each candidate variable individually using a score metric (in this case, we'll use KS score), and selects the X variables with highest score. Then, we will run a **wrapper,** which evaluates the entire set of variables (therefore, it accounts for correlation between variables) and narrows the candidate variables to a final selection.

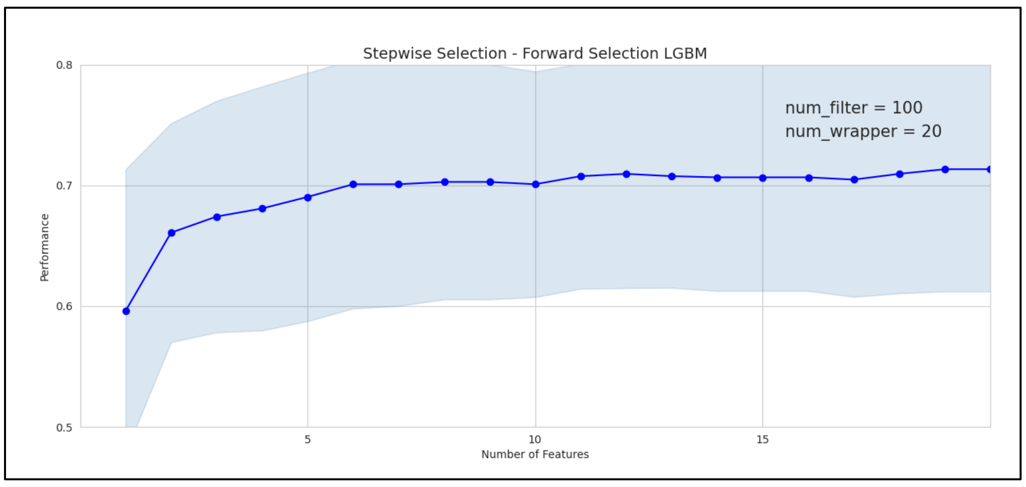
a. Exploration

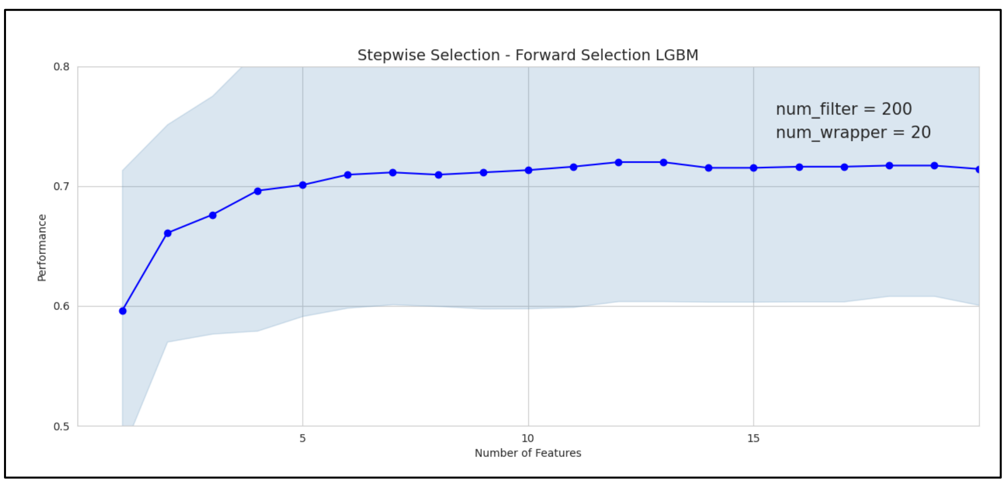
First, we are going to explore different parameters combinations and models for the filter and wrapper to see which gives us the best results. For the filter, we are going to modify the **number of variables** selected (*num_filter*) and for the wrapper we are going to test different **models** (*LGBM-FS, LGBM-BS, Random Forest).*

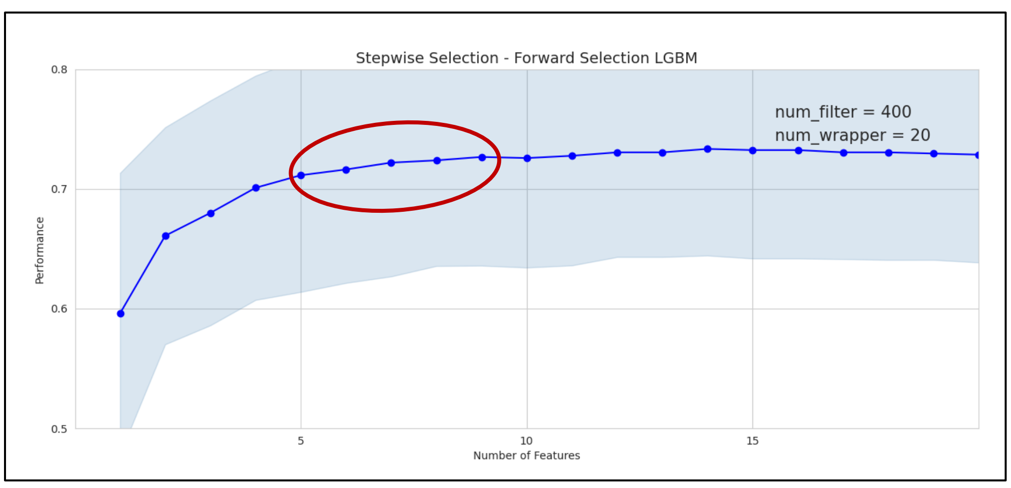
i. *LGBM – Forward Selection*

We tested this model for 3 different *num_filter* values: **100, 200, 400**.

These are around 5-20% of the total number of candidate variables (~2,000).



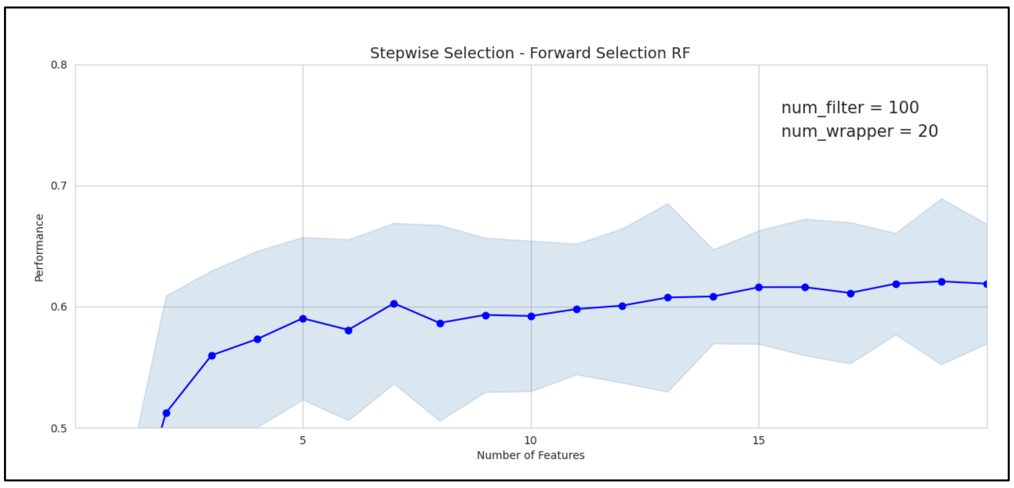


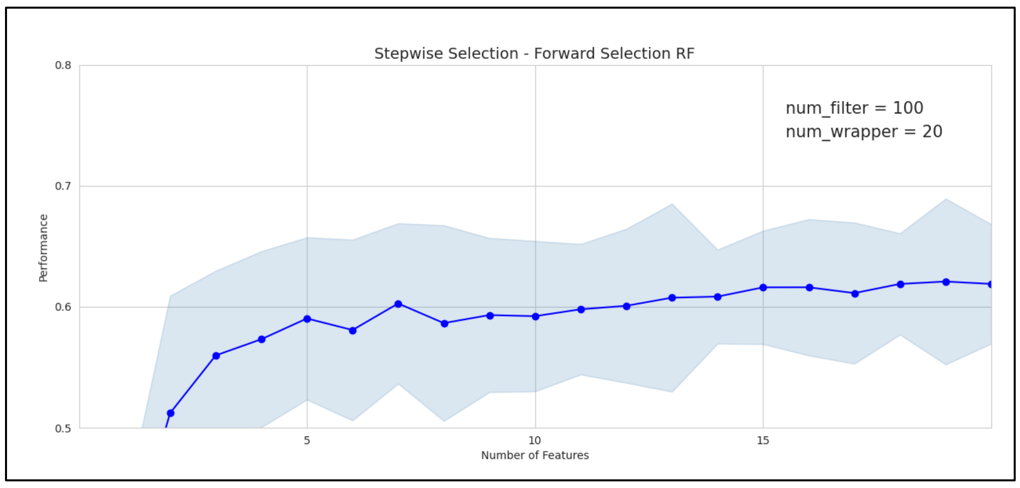


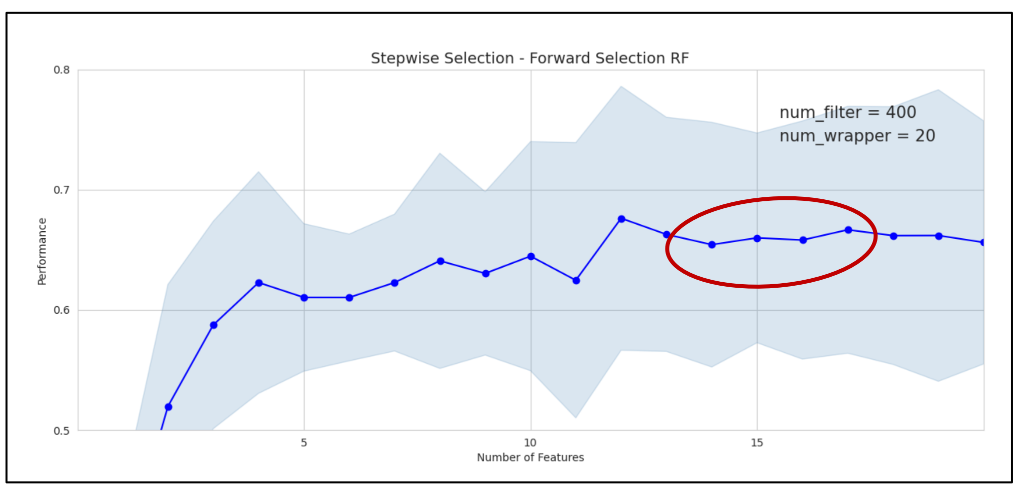
- The plots show that performance is better when we increase the number of variables (num_filter) that are given to the wrapper.
- We also see that **saturation occurs around the 8-9 features** (as shown by the red area in the 3rd plot).

ii. *Random Forest – Forward Selection*

Again, we tested this model for 3 different *num_filter* values: 100, 200, 400.



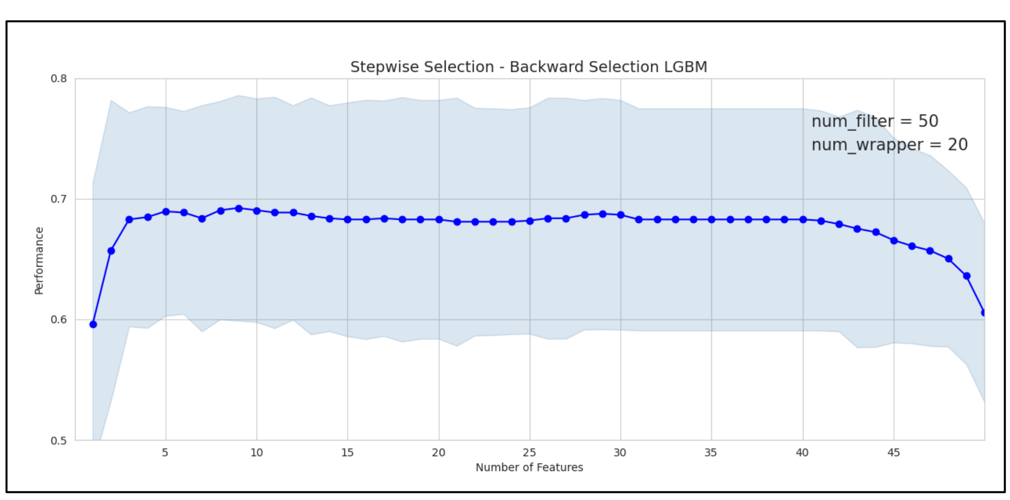


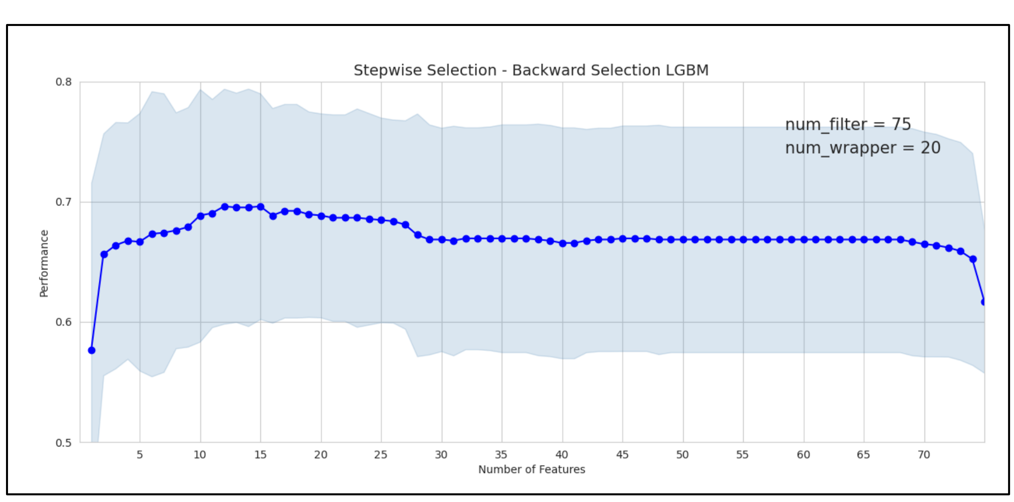


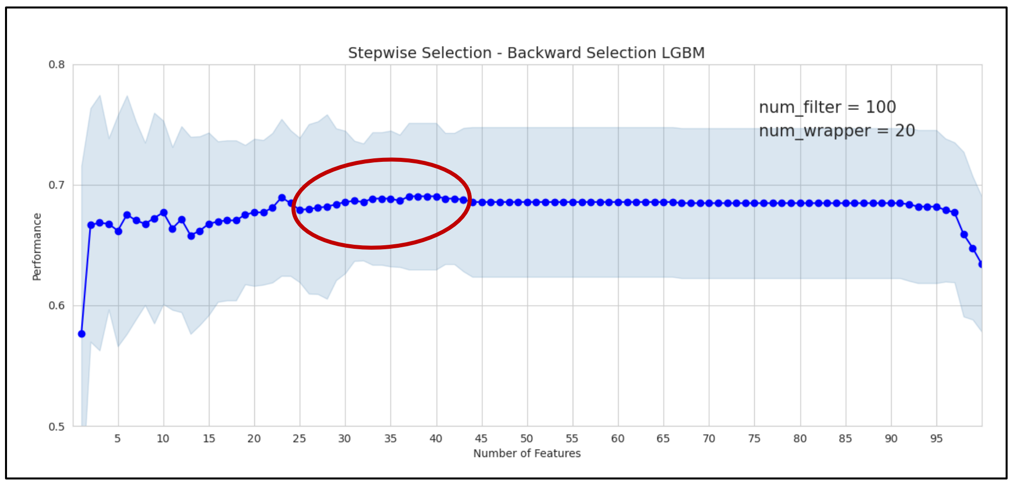
- Here, the performance is overall **lower** than the previous model, although we also see that it increases as we increase the number of input variables.
- This suggest that this model is a better choice when the input number is larger, as it doesn't reach saturation as quickly as the LGBM model.

iii. *LGBM – Backward Selection*

Finally, we try the LGBM with backward selection. Since this model takes longer to run, we tested only for these *num_filter* values: 50, 75, 100.







- This model performs worse than the LGBM – Forward Selection model, although it was tried with a fewer number of variables. With 100 variables, it reaches saturation around the 35 features.

Since our goal is **dimensionality reduction** and **model performance,** we are going to use the LGBM – Forward Selection model for feature selection, which delivers the best performance with fewer variables selection.

For the final number of variables, we'll select 20, which is about twice the number of variables where saturation is reached.

b. Variable Selection

The following table shows the **final 20 variables** selected, ordered by the wrapper, and with their respective filter (KS) scores:

Wrapper Order	Variable	Filter Score
1	card_merch_total_14	0.630048
2	card_zip3_max_14	0.629515
3	card_zip3_count_7	0.387860
4	Merchnum_desc_total_1	0.528445
5	Merchnum_desc_max_1	0.523694
6	Merchnum_desc_med_3	0.429393
7	card_zip3_variability_max_3	0.385868
8	zip3_variability_avg_3	0.405014
9	merch_zip_total_14	0.440019
10	merch_zip_max_3	0.514481
11	Card_Merchnum_desc_total_60	0.595019
12	state_des_med_3	0.425545
13	Merchnum_desc_total_7	0.517123
14	merch_zip_max_1	0.522153
15	card_merch_total_30	0.615461
16	Card_Merchnum_desc_total_30	0.606280
17	Card_Merchnum_Zip_total_30	0.612931
18	Card_Merchnum_Zip_total_14	0.627421
19	state_des_total_14	0.490872
20	Merchnum_desc_max_3	0.516808

6. Model Exploration

In this section, the main goal is to **build and tune predictive models** using the dataset with the final variable selection (20 variables) created in the previous section. Our response variable is **Fraud** classification (whether a transaction is fraudulent or not), and the **measure of goodness** used to compare and evaluate the models will be **Fraud detection rate at 3%** (FDR3%).

To do so, we are going to first prepare the data for modeling by scaling features and 'holding out' the last 2 months of data (out of time), which will be later used for evaluation (*it's important to note that in a real-world scenario, this OOT data wouldn't be available, and the model can only be evaluated using 'test' data*).

Then, we will train different models and try different hyperparameter combinations, to select a final model with the best combination.

a. Model Tuning

In the following table, we explore different hyperparameter combinations for each of the models to see which gives us the **best results**.

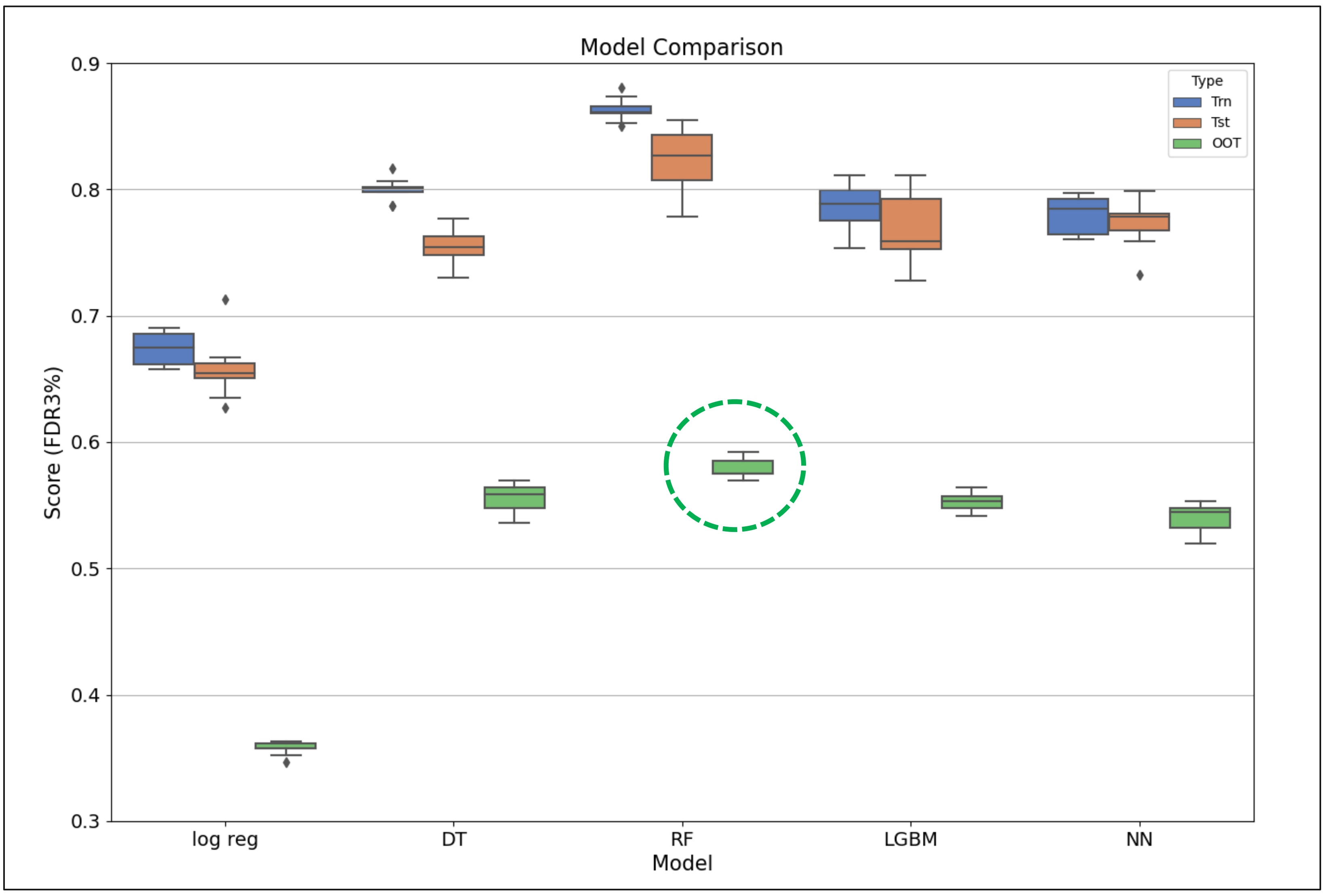
MODEL	PARAMETERS					Average FDR @ 3%		
	# Vars	penaty		C	solver	Train	Test	OOT
Logistic Regression	20	L2		1	lbgs	0.636	0.649	0.305
	20	L2		0.1	lbgs	0.656	0.648	0.334
	20	L2		0.01	lbgs	0.652	0.657	0.341
	20	L1		1	saga	0.646	0.633	0.328
	20	L1		0.1	saga	0.654	0.640	0.330
	20	L2		0.01	saga	0.658	0.655	0.345
	10	L2		1	lbgs	0.657	0.654	0.353
	10	L1		0.1	saga	0.659	0.655	0.245
	10	L2		0.01	saga	0.676	0.656	0.355
	# Vars	max_depth		min_samples_split	min_samples_leaf	Train	Test	OOT
Decision Tree	20	20		2	4	0.963	0.717	0.334
	20	20		4	100	0.805	0.764	0.549
	20	10		4	10	0.829	0.724	0.372
	20	10		4	100	0.797	0.769	0.560
	20	8		2	100	0.782	0.753	0.544
	10	None		2	1	0.799	0.741	0.549
	10	10		4	100	0.794	0.749	0.562
	10	20		2	10	0.974	0.764	0.359
	10	20		2	100	0.803	0.755	0.556
	# Vars	n_estimators	max_depth	min_samples_split	min_samples_leaf	Train	Test	OOT
Random Forest	20	100	None	2	1	1.000	0.862	0.530
	20	100	20	2	100	0.803	0.778	0.552
	20	200	20	2	100	0.799	0.773	0.554
	20	200	10	4	100	0.786	0.759	0.546
	20	400	10	4	50	0.817	0.792	0.560
	20	400	20	8	50	0.863	0.803	0.566
	20	400	20	2	50	0.865	0.807	0.566
	10	100	10	2	1	0.954	0.851	0.554
	10	200	10	4	100	0.812	0.782	0.570
	10	400	20	2	50	0.862	0.821	0.577
	10	400	10	2	100	0.807	0.794	0.571
	# Vars	n_estimators	max_depth	num_leaves	learning_rate	Train	Test	OOT
Boosted Tree (LGBM)	20	20	2	31	0.1	0.773	0.760	0.546
	20	100	None	31	0.1	1.000	0.848	0.388
	20	100	20	50	0.1	1.000	0.849	0.386
	20	200	10	50	0.01	0.959	0.861	0.423
	20	400	10	50	0.001	0.881	0.821	0.476
	20	400	20	20	0.0005	0.805	0.765	0.550
	20	400	50	20	0.0001	0.768	0.752	0.543
	10	20	2	31	0.1	0.768	0.758	0.553
	10	100	10	50	0.0001	0.807	0.768	0.436
	10	200	20	20	0.0005	0.792	0.764	0.555
	# Vars	activation	hidden_layer_sizes	alpha	solver	Train	Test	OOT
Neural Network	20	relu	(100,)	0.00001	adam	0.879	0.809	0.406
	20	relu	(10,)	0.00001	adam	0.763	0.741	0.513
	20	relu	(20,)	0.0001	adam	0.798	0.765	0.508
	20	tanh	(20,)	0.00001	adam	0.810	0.789	0.524
	20	tanh	(100,)	0.0001	sgd	0.683	0.661	0.366
	20	relu	(100, 2)	0.0001	sgd	0.716	0.699	0.462
	20	tanh	(25,)	0.00001	lbfgs	0.913	0.787	0.376
	20	tanh	(10,)	0.000001	lbfgs	0.836	0.813	0.431
	20	tanh	(25,)	0.000001	adam	0.824	0.803	0.509
	10	relu	(3,)	0.00001	adam	0.691	0.667	0.437
	10	relu	(100,)	0.0001	adam	0.837	0.817	0.476
	10	tanh	(20,)	0.00001	adam	0.785	0.763	0.544

- After exploring different parameter combinations for **20 variables,** the models were tested with **10 variables,** and all of them **performed better** in the OOT data when the number of variables was reduced.
- The 'best' (OOT) performance for each model is highlighted in red, and was the combination selected for model comparison.

b. Model Comparison

- The following plot shows the comparison for all 5 models tuned using the **best hyperparameter combination** found in part 1.

- The boxplot shows the variation of the **FDR3% score** in train, test, and OOT data, across the 10 different iterations for each model.
- Overall, a good model will have **similar train/test** scores (no overfitting), and **good test/OOT** scores (performs well on unseen data).



- We observe that the **Random Forest** model performs best in the 3 data subsets, and all models except for Logistic Regression have a reasonably good performance in OOT.

7. Final Model Performance

The final model selected is a **Random Forest** model with the **10 best variables** and the following hyperparameters:

- *n_estimators* (# of trees) = 400
- *max_depth* (tree depth) = 20
- *min_samples_split (*min # of samples in each node*) = 2*
- *min_samples_leaves = 50*

We will now summarize the final model's performance in the **training, testing,** and **validation (OOT)** sets, by grouping each set into bins (*100 bins total*) and calculating the number of **'goods'** and **'bads'** in each bin.

The following tables show these summarized statistics for the **first 20 bins**, and the total records are displayed in the top (green).

a. Training Performance

Training	# Records		# Goods		# Bads		Fraud Rate					
	58779		58196		583		0.009918508					
	Bin Statistics					Cumulative Statistics						
Population Bin %	# Records	# Goods	# Bads	% Goods	% Bads	Total # Records	Cumulative Goods	Cumulative Bads	% Goods	% Bads (FDR)	KS	FPR
1	588	238	350	40.48%	59.52%	588	238	350	0.41%	60.03%	59.63	0.68
2	588	468	120	79.59%	20.41%	1176	706	470	1.21%	80.62%	79.40	1.50
3	587	553	34	94.21%	5.79%	1763	1259	504	2.16%	86.45%	84.29	2.50
4	588	566	22	96.26%	3.74%	2351	1825	526	3.14%	90.22%	87.09	3.47
5	588	574	14	97.62%	2.38%	2939	2399	540	4.12%	92.62%	88.50	4.44
6	588	573	15	97.45%	2.55%	3527	2972	555	5.11%	95.20%	90.09	5.35
7	588	580	8	98.64%	1.36%	4115	3552	563	6.10%	96.57%	90.47	6.31
8	587	577	10	98.30%	1.70%	4702	4129	573	7.09%	98.28%	91.19	7.21
9	588	583	5	99.15%	0.85%	5290	4712	578	8.10%	99.14%	91.05	8.15
10	588	583	5	99.15%	0.85%	5878	5295	583	9.10%	100.00%	90.90	9.08
11	588	588	0	100.00%	0.00%	6466	5883	583	10.11%	100.00%	89.89	10.09
12	587	587	0	100.00%	0.00%	7053	6470	583	11.12%	100.00%	88.88	11.10
13	588	588	0	100.00%	0.00%	7641	7058	583	12.13%	100.00%	87.87	12.11
14	588	588	0	100.00%	0.00%	8229	7646	583	13.14%	100.00%	86.86	13.11
15	588	588	0	100.00%	0.00%	8817	8234	583	14.15%	100.00%	85.85	14.12
16	588	588	0	100.00%	0.00%	9405	8822	583	15.16%	100.00%	84.84	15.13
17	587	587	0	100.00%	0.00%	9992	9409	583	16.17%	100.00%	83.83	16.14
18	588	588	0	100.00%	0.00%	10580	9997	583	17.18%	100.00%	82.82	17.15
19	588	588	0	100.00%	0.00%	11168	10585	583	18.19%	100.00%	81.81	18.16
20	588	588	0	100.00%	0.00%	11756	11173	583	19.20%	100.00%	80.80	19.16

- In the **training** data, the model labels approx. 0.99% (583 out of 58,779) of transactions as '**bads**'.

b. Testing Performance

Testing	# Records		# Goods		# Bads		Fraud Rate					
	25191		24894		297		0.011789925					
	Bin Statistics					Cumulative Statistics						
Population Bin %	# Records	# Goods	# Bads	% Goods	% Bads	Total # Records	Cumulative Goods	Cumulative Bads	% Goods	% Bads (FDR)	KS	FPR
1	252	94	158	37.30%	62.70%	252	94	158	0.38%	53.20%	52.82	0.59
2	252	190	62	75.40%	24.60%	504	284	220	1.14%	74.07%	72.93	1.29
3	252	239	13	94.84%	5.16%	756	523	233	2.10%	78.45%	76.35	2.24
4	252	242	10	96.03%	3.97%	1008	765	243	3.07%	81.82%	78.75	3.15
5	252	244	8	96.83%	3.17%	1260	1009	251	4.05%	84.51%	80.46	4.02
6	251	248	3	98.80%	1.20%	1511	1257	254	5.05%	85.52%	80.47	4.95
7	252	249	3	98.81%	1.19%	1763	1506	257	6.05%	86.53%	80.48	5.86
8	252	250	2	99.21%	0.79%	2015	1756	259	7.05%	87.21%	80.15	6.78
9	252	250	2	99.21%	0.79%	2267	2006	261	8.06%	87.88%	79.82	7.69
10	252	250	2	99.21%	0.79%	2519	2256	263	9.06%	88.55%	79.49	8.58
11	252	252	0	100.00%	0.00%	2771	2508	263	10.07%	88.55%	78.48	9.54
12	252	248	4	98.41%	1.59%	3023	2756	267	11.07%	89.90%	78.83	10.32
13	252	251	1	99.60%	0.40%	3275	3007	268	12.08%	90.24%	78.16	11.22
14	252	252	0	100.00%	0.00%	3527	3259	268	13.09%	90.24%	77.14	12.16
15	252	250	2	99.21%	0.79%	3779	3509	270	14.10%	90.91%	76.81	13.00
16	252	250	2	99.21%	0.79%	4031	3759	272	15.10%	91.58%	76.48	13.82
17	251	251	0	100.00%	0.00%	4282	4010	272	16.11%	91.58%	75.47	14.74
18	252	251	1	99.60%	0.40%	4534	4261	273	17.12%	91.92%	74.80	15.61
19	252	252	0	100.00%	0.00%	4786	4513	273	18.13%	91.92%	73.79	16.53
20	252	251	1	99.60%	0.40%	5038	4764	274	19.14%	92.26%	73.12	17.39

- In the **testing** data, the model labels approx. 1.18% (297 out of 25,191) of transactions as '**bads**'.

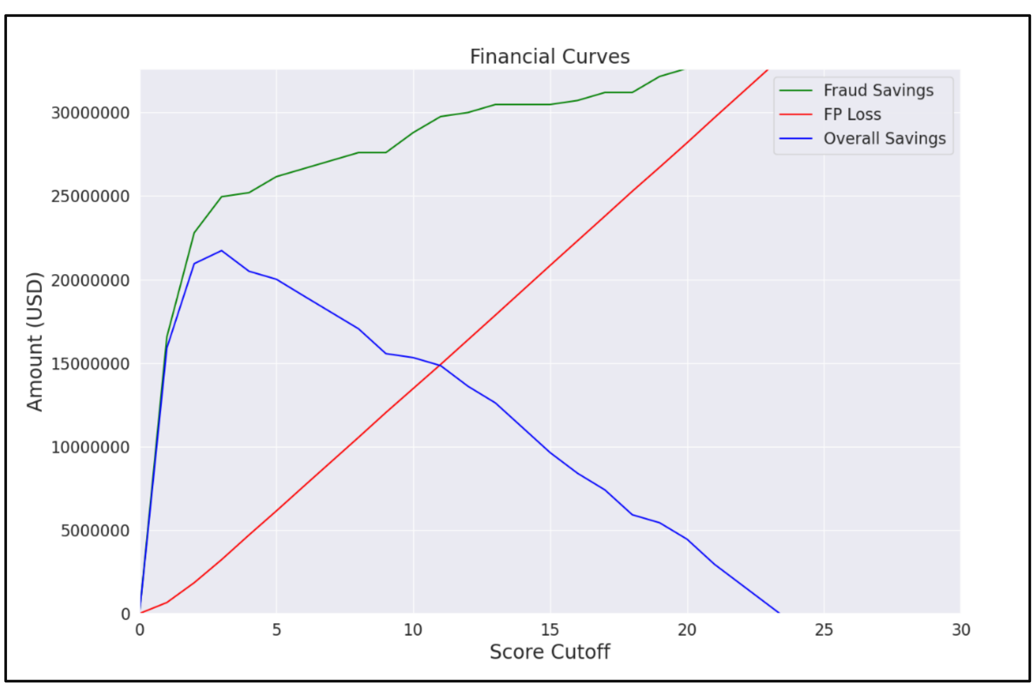
c. Evaluation Performance

OOT	# Records		# Goods		# Bads		Fraud Rate					
	12427		12248		179		0.01440412					
	Bin Statistics					Cumulative Statistics						
Population Bin %	# Records	# Goods	# Bads	% Goods	% Bads	Total # Records	Cumulative Goods	Cumulative Bads	% Goods	% Bads (FDR)	KS	FPR
1	124	55	69	44.35%	55.65%	124	55	69	0.45%	38.55%	38.10	0.80
2	125	99	26	79.20%	20.80%	249	154	95	1.26%	53.07%	51.82	1.62
3	124	115	9	92.74%	7.26%	373	269	104	2.20%	58.10%	55.90	2.59
4	124	123	1	99.19%	0.81%	497	392	105	3.20%	58.66%	55.46	3.73
5	124	120	4	96.77%	3.23%	621	512	109	4.18%	60.89%	56.71	4.70
6	125	123	2	98.40%	1.60%	746	635	111	5.18%	62.01%	56.83	5.72
7	124	122	2	98.39%	1.61%	870	757	113	6.18%	63.13%	56.95	6.70
8	124	122	2	98.39%	1.61%	994	879	115	7.18%	64.25%	57.07	7.64
9	124	124	0	100.00%	0.00%	1118	1003	115	8.19%	64.25%	56.06	8.72
10	125	120	5	96.00%	4.00%	1243	1123	120	9.17%	67.04%	57.87	9.36
11	124	120	4	96.77%	3.23%	1367	1243	124	10.15%	69.27%	59.13	10.02
12	124	123	1	99.19%	0.81%	1491	1366	125	11.15%	69.83%	58.68	10.93
13	125	123	2	98.40%	1.60%	1616	1489	127	12.16%	70.95%	58.79	11.72
14	124	124	0	100.00%	0.00%	1740	1613	127	13.17%	70.95%	57.78	12.70
15	124	124	0	100.00%	0.00%	1864	1737	127	14.18%	70.95%	56.77	13.68
16	124	123	1	99.19%	0.81%	1988	1860	128	15.19%	71.51%	56.32	14.53
17	125	123	2	98.40%	1.60%	2113	1983	130	16.19%	72.63%	56.44	15.25
18	124	124	0	100.00%	0.00%	2237	2107	130	17.20%	72.63%	55.42	16.21
19	124	120	4	96.77%	3.23%	2361	2227	134	18.18%	74.86%	56.68	16.62
20	124	122	2	98.39%	1.61%	2485	2349	136	19.18%	75.98%	56.80	17.27

- In the **evaluation** data, the model labels approx. 1.44% (179 out of 12,427) of transactions as '**bads**'.

8. Financial Curves

The following chart shows the financial curves for different score cutoffs (i.e., what percentage of transactions are flagged as fraudulent). The **green line** shows the gains (savings) for every fraud that is correctly detected. The **red line** shows losses for false positives (non-fraudulent transaction flagged as fraudulent). The **blue line** shows the overall savings (total gains – losses).



- From the chart, we see that the score cutoff that maximizes **overall savings** is approximately 2.7%.
- We recommend using a score cutoff of **3%,** which is still close to the optimal, but "safer", as it will flag a higher number of transactions.

9. Summary

This report offers valuable insights into the effectiveness of implementing a predictive model in the business's fraud detection system.

In the first section, a description of the dataset used is provided, including summary tables for both categorical and numerical fields. The raw dataset included a total of 96,753 records across 10 fields. Since the original dataset included some null values and records that we didn't want our model to train on, the data cleaning process handled both the exclusion of these records (non-purchase transactions, different currency) and the imputation of missing values for the 'Merch zip', 'Merch num', and 'Zip code' fields.

After cleaning the data, we created as many variables as possible by combining different entities, and calculating frequencies, variability, velocity change, amount, days since, and other ratios to create a large number of variables. A total of 2,227 variables was created in this step. In the next section, the total number of variables was reduced to only 20, by running a filter and then a wrapper. Here, the main objective was dimensionality reduction and model performance, so we used the LGBM – Forward Selection model, which delivered the best performance with fewer variables selection.

Once the features were selected, different models were trained and tested in the training, testing, and OOT datasets, with a wide variety of hyperparameters. Here, our response variable was Fraud classification (whether a transaction is fraudulent or not), and the measure of goodness used to compare and evaluate the models was Fraud detection rate at 3% (FDR3%). The model comparison showed that the Random Forest model performs best in the 3 data subsets.

We finally evaluated the selected model's performance. In the evaluation data, the model labeled approximately 1.44% (179 out of 12,427) of transactions as fraudulent. In OOT data, the model has a FDR @3% of 0.577. The score cutoff recommendation leads to expected savings of approximately $21,000,000 in OOT transactions when projected for the entire year.

10. Annex: Data Quality Report

1. Data Description

The dataset is **Credit Card Transaction Data**, which contains information of US government organization card transactions, in the **year 2010**. The data is a collection of **96,753** records including **10 fields**.

2. Summary Tables

Numeric Fields Table

Field Name	# Records With Values	% Populated	# Zeros	Min	Max	Mean	Most Common	Stdev
Date	96,753	100%	0	1/1/2010	12/31/2010	6/25/2010	2/28/2010	98 days 21:38:57
Amount	96,753	100%	0	0.01	3,102,046	427.9	3.62	10,006

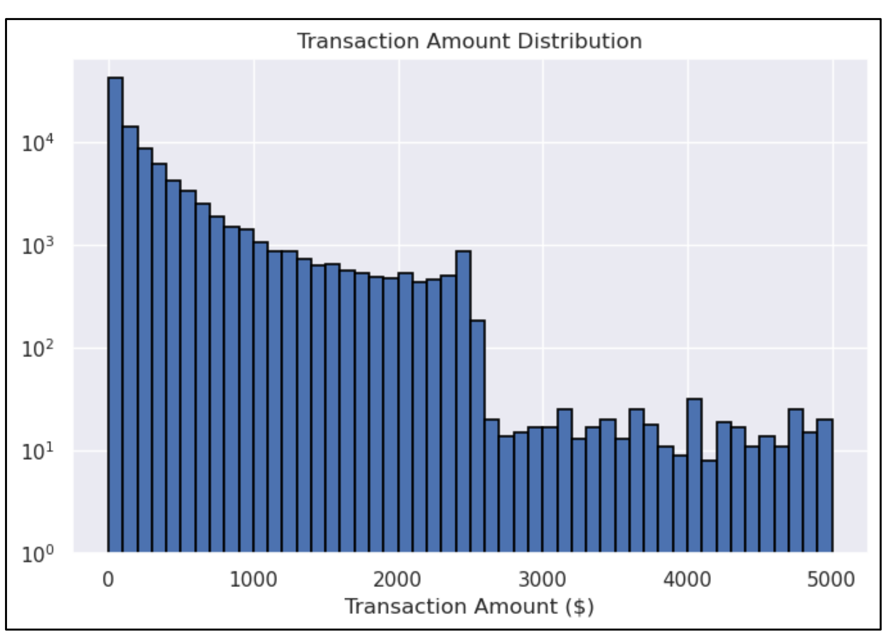
Categorical Fields Table

Field Name	# Records With Values	% Populated	# Zeros	# Unique Values	Most Common
Recnum	96,753	100.00%	0	96,753	1
Cardnum	96,753	100.00%	0	1,645	5142148452
Merchnum	93,378	96.51%	0	13,091	930090121224
Merch description	96,753	100.00%	0	13,126	GSA-FSS-ADV
Merch state	95,558	98.76%	0	227	TN
Merch zip	92,097	95.19%	0	4,567	38118
Transtype	96,753	100.00%	0	4	P
Fraud	96,753	100.00%	95,694	2	0

3. Visualization of Each Field

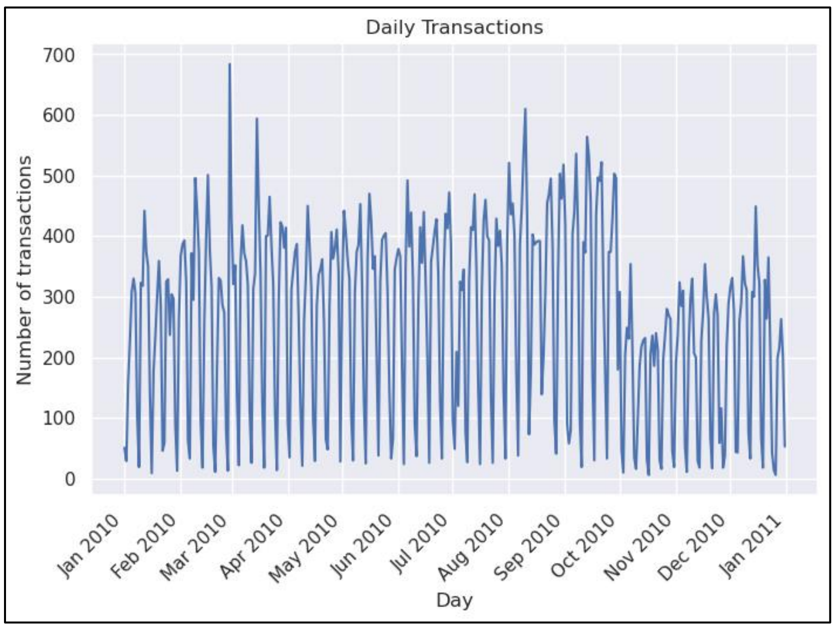
1) Field Name: Amount

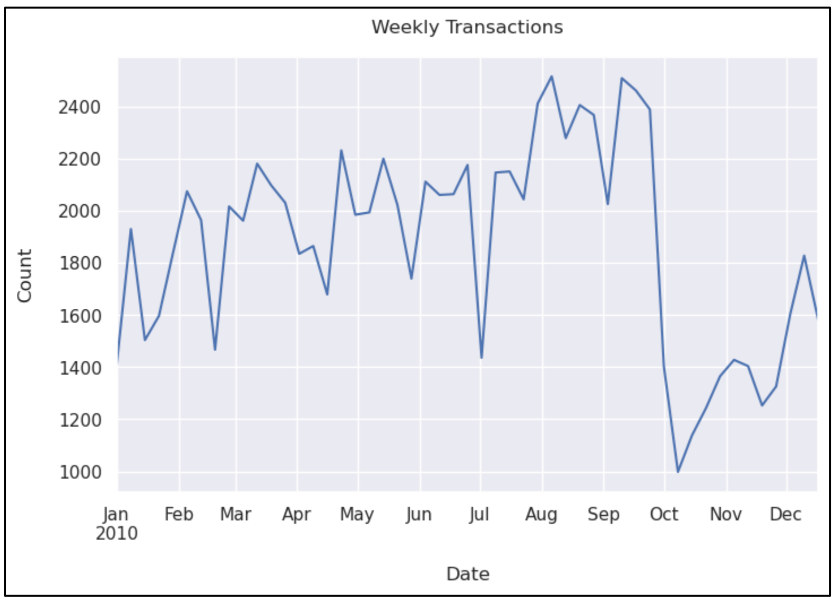
Description: Transaction amount. The histogram shows the distribution of transaction amounts. The frequencies are shown in log scale, and only transactions up to $5,000 are shown in the chart (99.68% of total transactions).

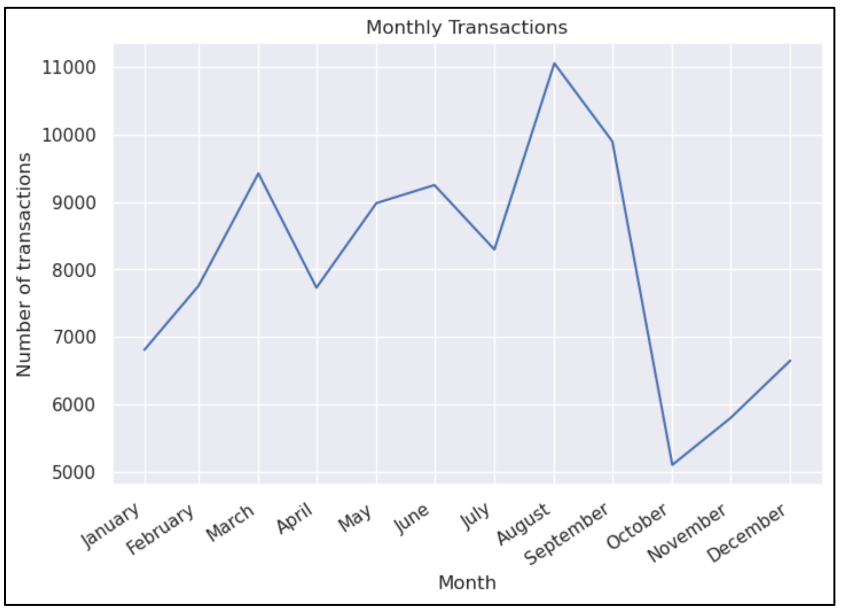


2) Field Name: Date

Description: Transaction date. The first distribution shows the number of **daily** transactions across the year 2010. The second distribution shows the number of **weekly** transactions across the year 2010. The third distribution shows the number of **monthly** transactions across the year 2010.





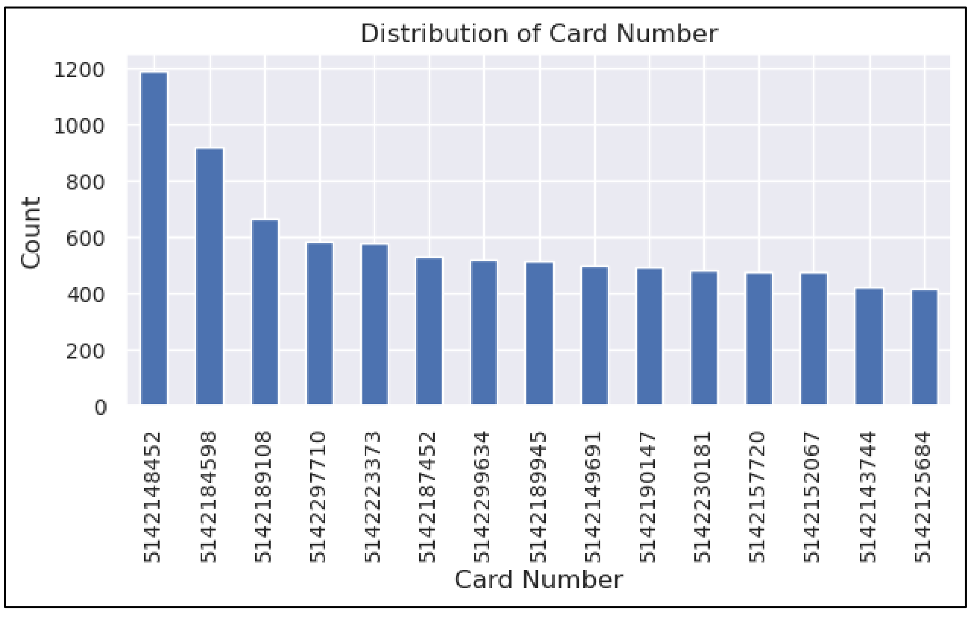


3) Field Name: Recnum

Description: Ordinal unique positive integer for each transaction record, from 1 to 96,753.

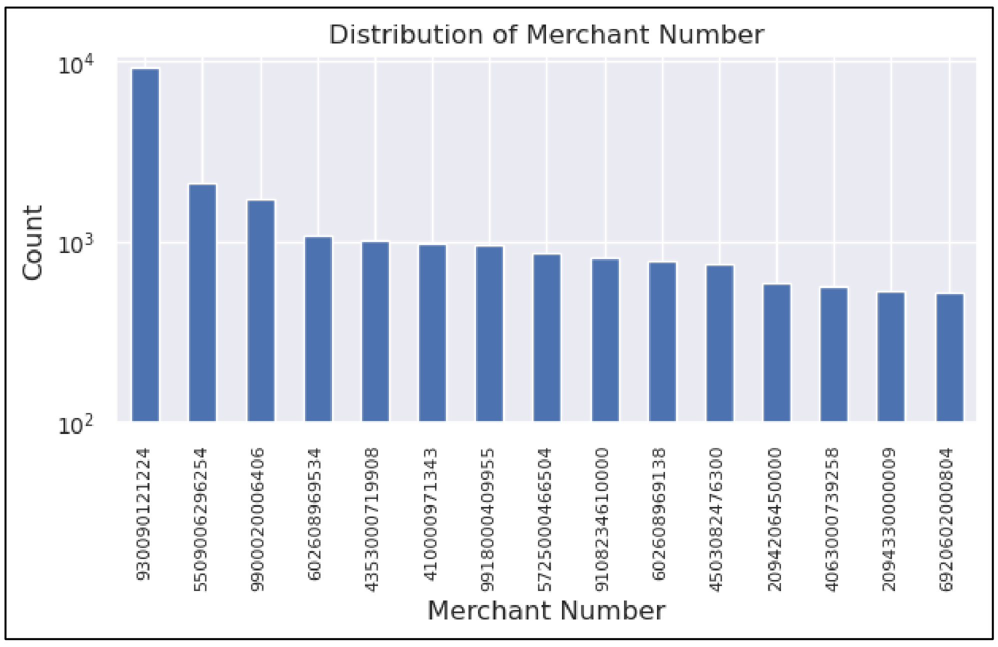
4) Field Name: Cardnum

Description: Credit card number. There are a total of 1,645 unique card numbers. The distribution shows the top 15 field values. The most common number is '5142148452', with a total count of 1,192.



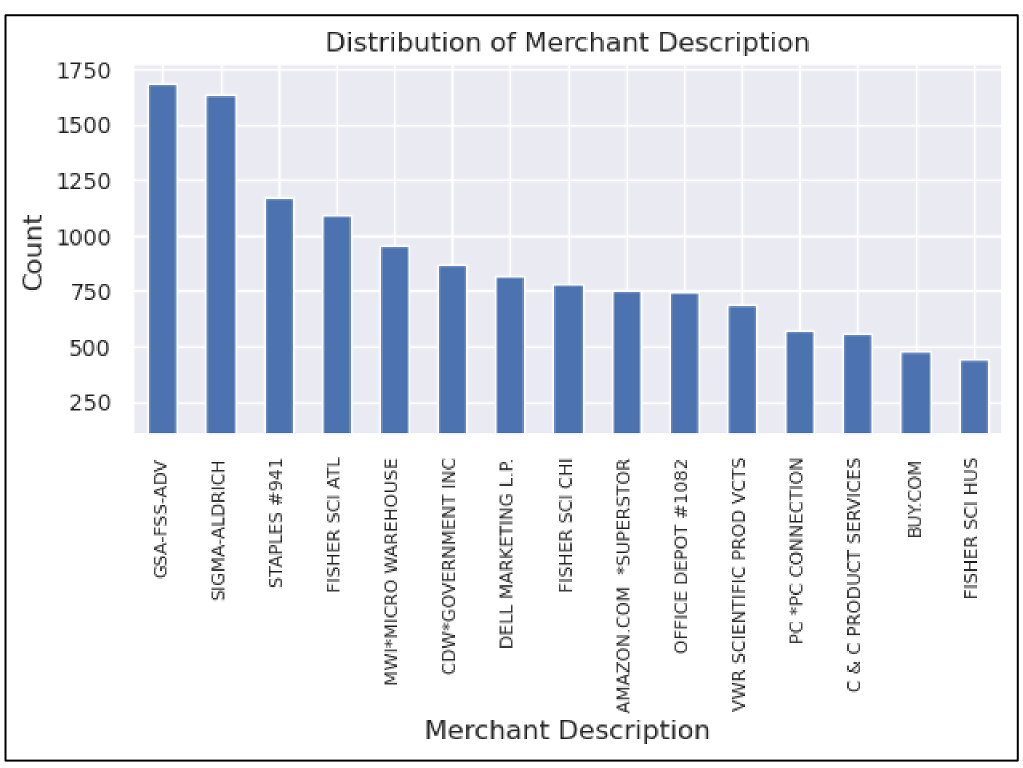
5) Field Name: Merchnum

Description: Merchant (business) number. There are a total of 13,092 unique merchant numbers. The distribution shows the top 15 field values. The most common number is '930090121224', with a total count of 9,310.



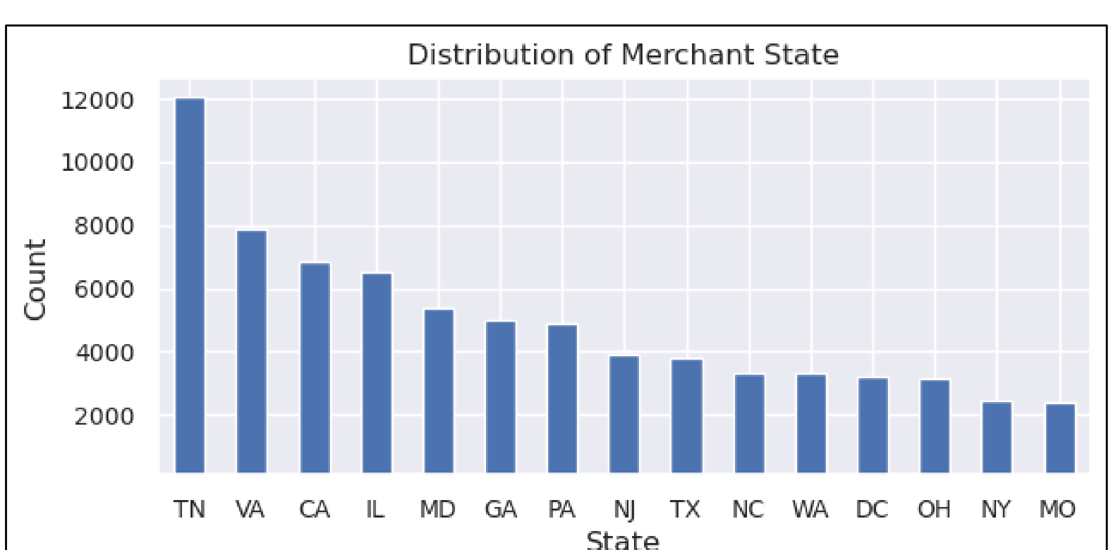
6) Field Name: Merch description

Description: Merchant (business) description. There are a total of 13,126 unique merchant descriptions. The distribution shows the top 15 field values. The most common description is 'GSA-FSS-ADV', with a total count of 1,688.



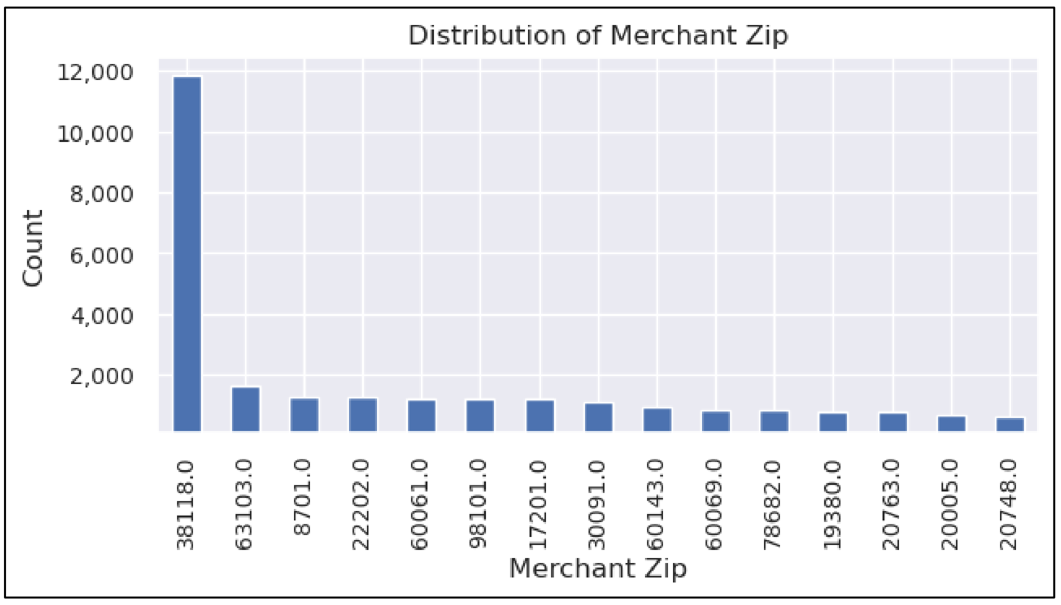
7) Field Name: Merch state

Description: Merchant's state. The distribution shows the top 15 field values of business' state. The most common state is Tennessee (TN), with a total count of 12,035.



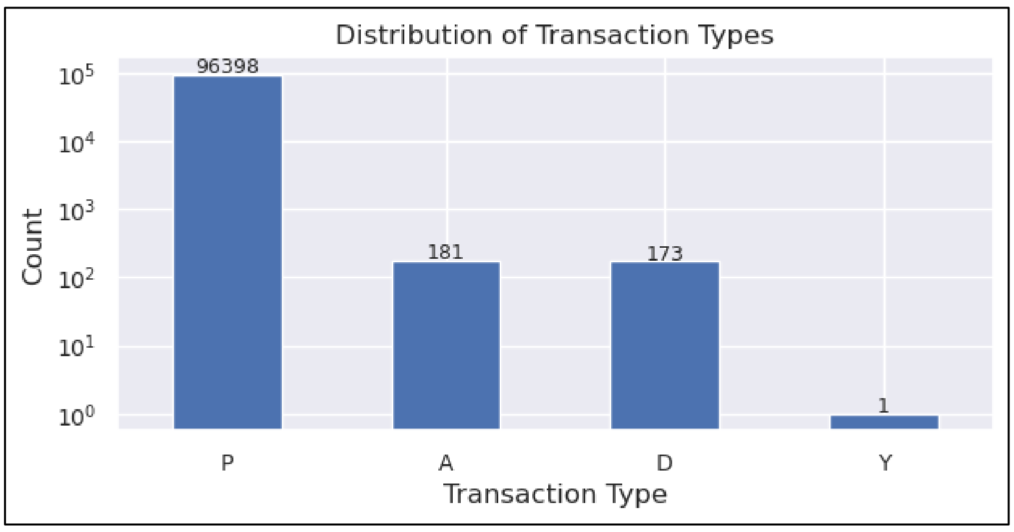
8) Field Name: Merch zip

Description: Merchant's zip code. The distribution shows the top 15 field values of business' zip code. The most common zip code is '38118', with a total count of 11,868.



9) Field Name: Transtype

Description: Transaction types. The distribution shows the top 15 field values of transaction types. The most common transaction type is 'P', with a total count of 96,398.



10) Field Name: fraud

Description: Fraud identification label. Fraud = 0 (Not fraudulent), Fraud =1 (Fraud identified). The total count of fraud = 0 is 95,694. The total count of fraud = 1 is 1,059.