Project 1 Report

Eloise Yu

MS in Business Analytics, Marshall School of Business, University of Southern California

DSO 562: Fraud Analytics

Dr. Stephen Coggeshall

October 16, 2024

Table of Contents

Executive Summary

This project addresses the business problem of detecting fraudulent credit card transactions to minimize financial losses. Using a dataset of 97,852 card transaction records from a US government organization, the goal is to build a model that could accurately identify fraudulent activities while controlling false positives. After extensive preprocessing and feature engineering, CatBoost was selected as the final model for its top performance among all tested models. The model was evaluated across training, testing, and out-of-time datasets, and a financial analysis based on a 3% fraud detection rate recommended a 5% cutoff score. This is expected to generate up to $49,488,000 in annual savings by effectively balancing fraud detection and false positives.

Data Description

Dataset Overview

The card transactions dataset consists of real credit card transaction records collected from a US government organization during the year 2010, spanning from January 1 to December 31. This comprehensive dataset includes 10 fields, featuring essential information such as Card Number, Merchant Number, Merchant Description, and Transaction Amount. Notably, it also includes a fraud label that indicates whether each transaction is legitimate or fraudulent. In total, the dataset comprises 97,852 records, providing a substantial resource for analyzing credit card transactions and detecting fraudulent activities.

Field Summary Statistics Tables

Table 1

Numeric Field Summary Statistics

	Field Name	Field Type	# Records Have Values	% Populated	# Zeros	Min	Max	Mean	Standard Deviation	Most Common
0	Amount	numeric	97852	100.0%	0	0.01	3102045.53	425.466438	9949.8	3.62

Table 2

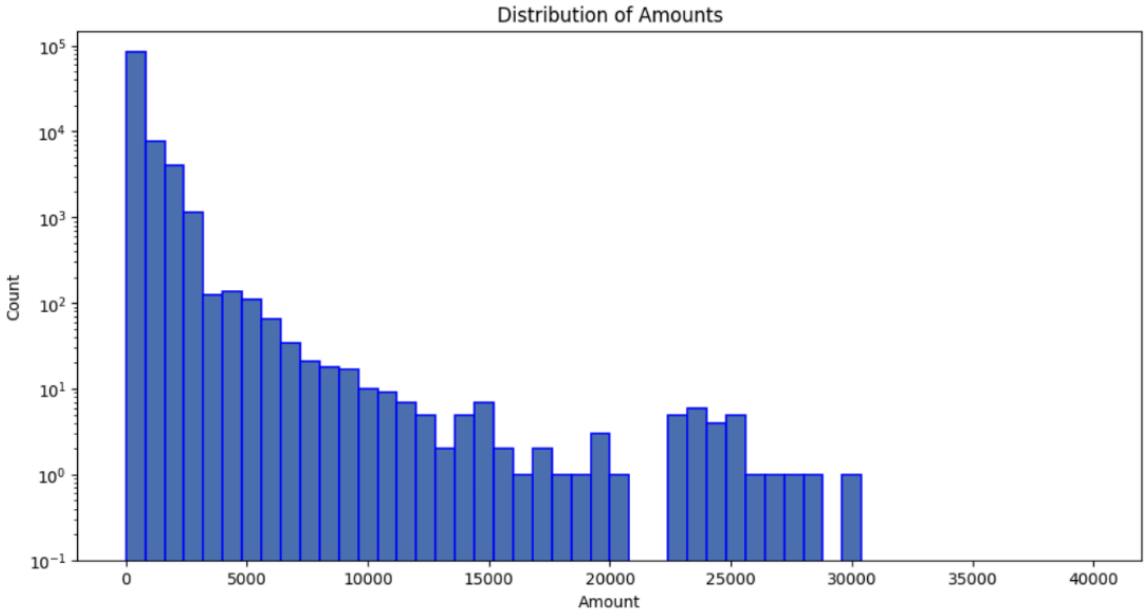
Categorical Field Summary Statistics

	Field Name	Field Type	# Records Have Values	% Populated	# Zeros	# Unique Values	Most Common
0	Date	categorical	97852	100.0%	0	365	2/28/10
1	Merchnum	categorical	94455	96.5%	0	13091	930090121224
2	Merch description	categorical	97852	100.0%	0	13126	GSA-FSS-ADV
3	Merch state	categorical	96649	98.8%	0	227	TN
4	Transtype	categorical	97852	100.0%	0	4	P
5	Recnum	categorical	97852	100.0%	0	97852	1
6	Fraud	categorical	97852	100.0%	95805	2	0
7	Cardnum	categorical	97852	100.0%	0	1645	5142148452
8	Merchnum	categorical	94455	96.5%	0	13091	930090121224
9	Merch zip	categorical	93149	95.2%	0	4567	38118.0

Field Distribution

Figure 1

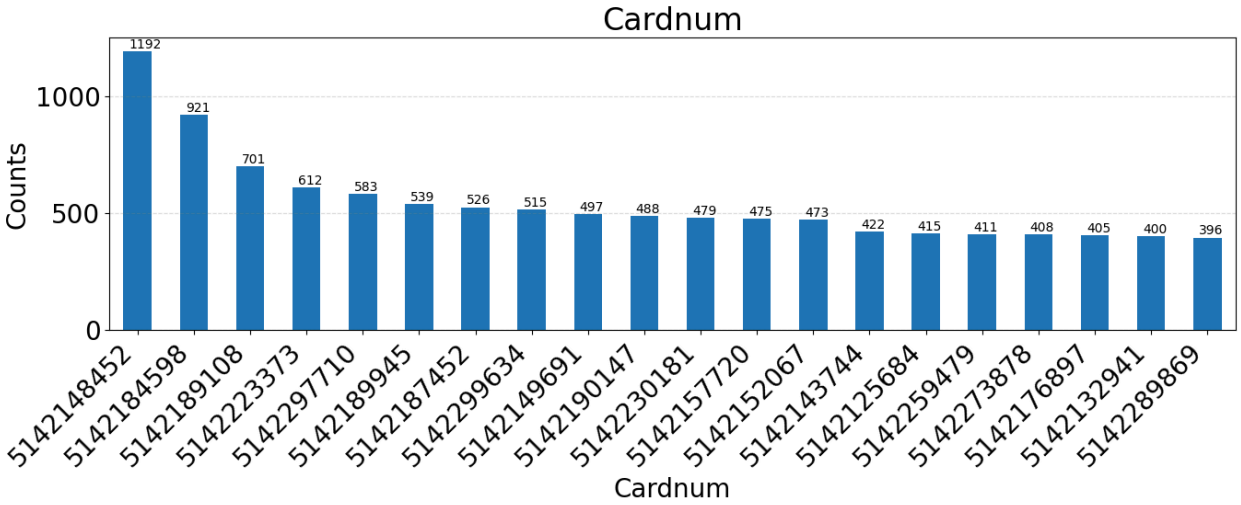
Amount Field Distribution



Note. Amount is a numerical field for the amount of each transaction in the dataset.

Figure 2

Card Number Field Distribution



Note. Cardnum is a categorical field for recording each card number that was used.

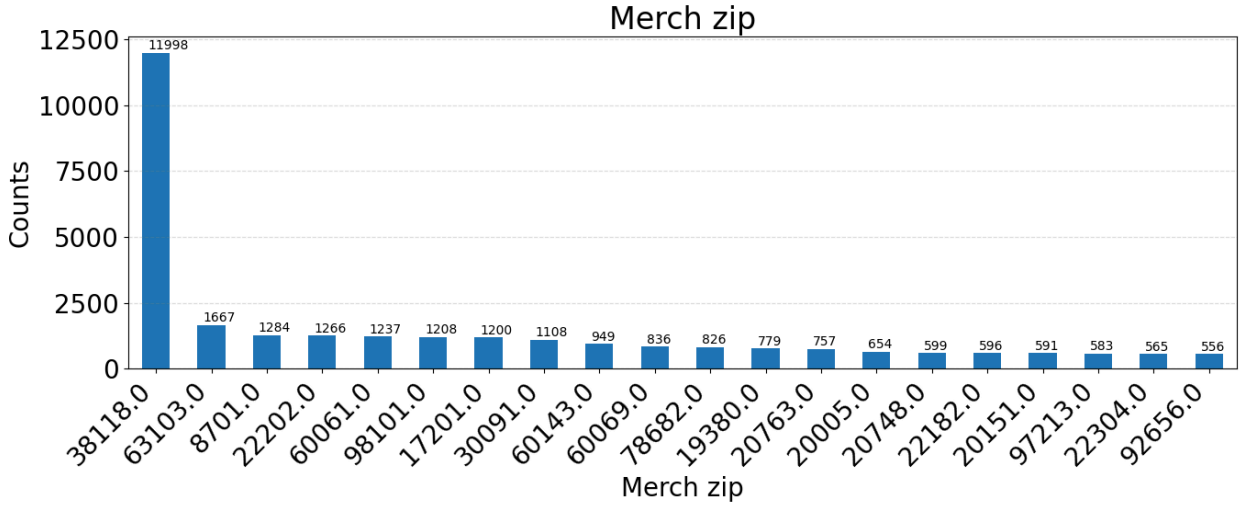
Figure 3

Fraud Field Distribution.

Notes. Fraud is a categorical field containing two categories, indicating whether a transaction is fraudulent or not.

Figure 4

Merchant Zip Code Field Distribution



Note. Merch zip is a categorical field for all the zip code of the merchant's location.

Data Cleaning

Exclusions

We see 4 different transaction types: P (97,497), A (181), D (173), Y (1). Most of the data is type P. Our business manager isn't really sure what these different types mean, but their best guess is that P means purchase and the others might be authorizations without a purchase, declines, etc. We are told to exclude all transactions other than type P.

Outliers

There's one record with a transaction amount above $3,000,000 that is substantially higher than the other transactions (the next highest is $47,900). When we look further into this $3 million outlier transaction we see that it was a purchase through a Mexican retailer. We see it's not labeled as fraud. After discussion with our business manager, we decided to remove this unusual transaction from our modeling.

Methods for imputation

Imputation of the field Merchnum

i) There are 3,279 records where the field Merchnum is missing (blank). We need to fill these in with our best guess at reasonable values.

ii) Use the Merch description field to fill in the most appropriate Merchnum for that Merch description.

iii) That takes care of 1,164 records and we still have 2,115 remaining missing.

iv) When the Merch description field is "RETAIL CREDIT ADJUSTMENT", set the Merchnum to "unknown". This takes care of 694, and we still have 1,421 remaining.

v) For these 1,421 records that are still missing the Merchnum, we find that there are 515 different values of Merch description. So there are lots of different miscellaneous merchants with not too many transactions each.

vi) For these 515 different Merch descriptions we'll give each one a new and unique Merchnum. Now all the records have a nonblank Merchnum field.

Imputation of the field State

vii) There are 1,028 records where the field 'Merch state' is missing. We need to fill these in with our best guess at reasonable values.

viii) Retrieve unique zip codes where the 'Merch state' is missing, but the 'Merch zip' is available. Then build a dictionary 'Zip_State' that maps each unique zip code to its associated merchant state, but only for non-null zip codes and fills in missing zip-to-state associations for specific cases. This dictionary contents 4,567 entries

ix) Then build two dictionaries, 'Merchnum_state' and 'Merchdes_state', using the same methodology. 'Merchnum_state' maps each unique Merchnum to its corresponding state and 'Merchdes_state' maps merchant descriptions to states.

x) Then fill in missing values in the 'Merchnum_state' column by using the 'Zip_State' dictionary. It maps the 'Merch zip' values to corresponding states and fills in any missing 'Merchnum_state' values using this mapping.

xi) This only handles 74 out of the total of 1,028 missing values, we still have 954 remaining.

xii) When the Merch description field is “RETAIL CREDIT ADJUSTMENT”, set the ‘Merchnum_state’ to ”unknown”. This takes care of 655, and we still have 297 remaining.

xiii) Label non-US states as foreign and fill all remaining missing values as ‘unknown’. Now all the records have a nonblank ‘Merch state’ field.

Imputation of the field Zip

xiv) There are 4,347 records where the field ‘Merch zip’ is missing (blank). We need to fill these in with our best guess at reasonable values.

xv) Create two dictionaries to map Merchnum and ‘Merch description’ to ‘Merch zip’. Then fill in missing values by mapping ‘Merch description’ to ‘Merch zip’ in ‘Merch zip’ using the two dictionaries. This handles 1,722 missing values, and we still have 2,625 remaining.

xvi) When the Merch description field is “RETAIL CREDIT ADJUSTMENT”, set the ‘Merchnum zip’ to ”unknown”. This takes care of 685, and we still have 1940 remaining.

xvii) Create a dictionary with each state’s two-letter ID and its corresponding most populous zip code. Then fill in missing values by mapping using the most populous zip code to ‘Merch zip. This handles 1,409 blanks, and we have 531 remaining.

xviii) Fill in the rest of missing values as ‘unknown’. Now all the records have a nonblank ‘Merch zip’ field.

Variable Creation

Fraud in credit card transactions typically involves unauthorized purchases made by individuals who gain access to cardholder information. Fraudsters may use stolen card details to make rapid or unusually high-value transactions, often across multiple merchants or locations within a short time frame. To detect these patterns, various features were created in the variable creation process using a list of entities shown below. These features collectively aim to differentiate normal spending patterns from potential fraudulent activities.

Entities: ['Cardnum', 'Merchnum', 'card_merch', 'card_zip', 'card_state', 'merch_zip', 'merch_state', 'state_des', 'Card_Merchdesc', 'Card_dow', 'Merchnum_desc', 'Merchnum_dow', 'Merchdesc_dow', 'Card_Merchnum_desc', 'Card_Merchnum_Zip', 'Card_Merchdesc_Zip', 'Merchnum_desc_State', 'Merchnum_desc_Zip', 'merchnum_zip', 'Merchdesc_State', 'Merchdesc_Zip', 'Card_Merchnum_State', 'Card_Merchdesc_State']

Table 3

Variable Creation & Description

Description of Variables	# Variables Created
Day since variables: number of days since a transaction with that entity was last seen.	23
Frequency variables: number of transactions with the same entity over the past 0, 1, 3, 7, 14, 30, 60 days	161
Amount variables: average, maximum, median, total, actual/average, actual/maximum, actual/median, actual/total amount by a particular	1288

entity over the past 0, 1, 3, 7, 14, 30, 60 days	
Ratio velocity change variables: the ratio of the number of transaction for that entity that occurred over the past 0, 1 days compared to the number of transaction over the past 7, 14, 30, 60 days	184
Ratio velocity change variables: the ratio of the total amount of transactions associated with that entity that occurred on the past 0, 1 days compared to the total amount of transactions over the past 7, 14, 30, 60 days	184
Ratio velocity change variables: the ratio of the number of records for a particular entity that occurred over the past 0, 1 days count to the past 7, 14, 30, 60 days compared to the number of days since the last transaction for that entity	184
Variability in amount difference variables: average, maximum, median differences in amounts between matching transactions for each entity within the time range of 0, 1, 7, 14, 30 days	414
Unique count variables: the number of unique values of one entity associated with another entity over the past 1,3,7,14,30,60 days	696
Ratio of counts by entity: the ratio of counts for an entity over the past 0, 1 days to the count over the past 7, 14, 30, 60 days and normalized by dividing the result by the square of the longer time window (7, 14,	184

30, 60 days)	
Transaction frequency variables: the number of transactions made by each CardNum over the past 7 and 30 days (new variable created)	2
Transaction since last transaction (recency) variable: the number of days since the last transaction for each CardNum (new variable created)	1
Day of the Week / Hour of the Day Variables: the day of the week and hour of day when each transaction took place (new variable created)	2
Amount bin variable: the transaction category based on the the binning amounts	1
Foreign variable: if the merchant's zip code is foreign (i.e., not in the US), indicating whether each merchant is outside the US	1

Feature Selection

In the feature selection process, various techniques were applied to narrow down the extensive set of engineered features and retain only the most relevant ones for fraud detection. By using methods such as forward and backward selection, along with wrapper models like LightGBM (LGBM) and Random Forest (RF), we evaluated feature importance and optimized the selection based on model performance. This ensured that only the most impactful variables were used in the final model, improving both accuracy and efficiency in identifying fraudulent transactions.

Forward Selection Experiments

Figure 5

Random Forest (RF): Filter 200 and Wrapper 30

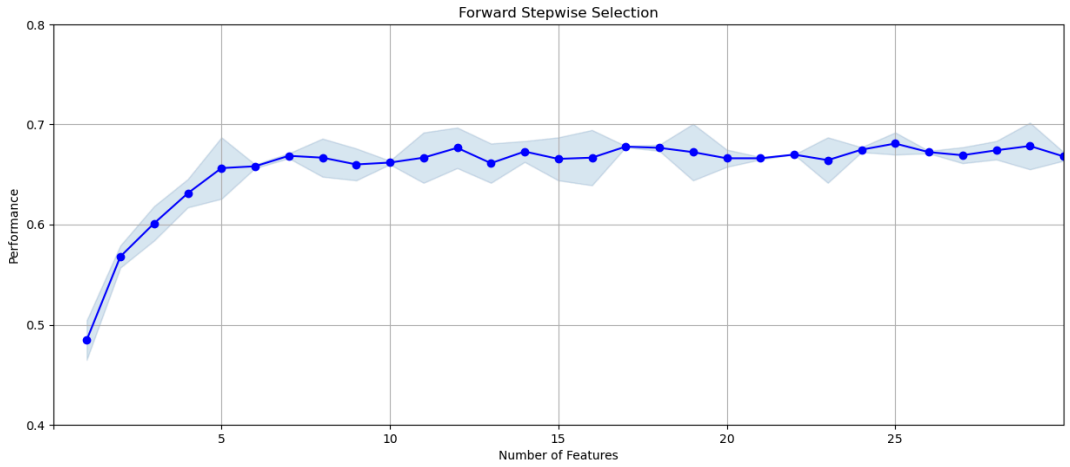


Figure 6

LightGBM (LGBM): Filter 500 and Wrapper 20

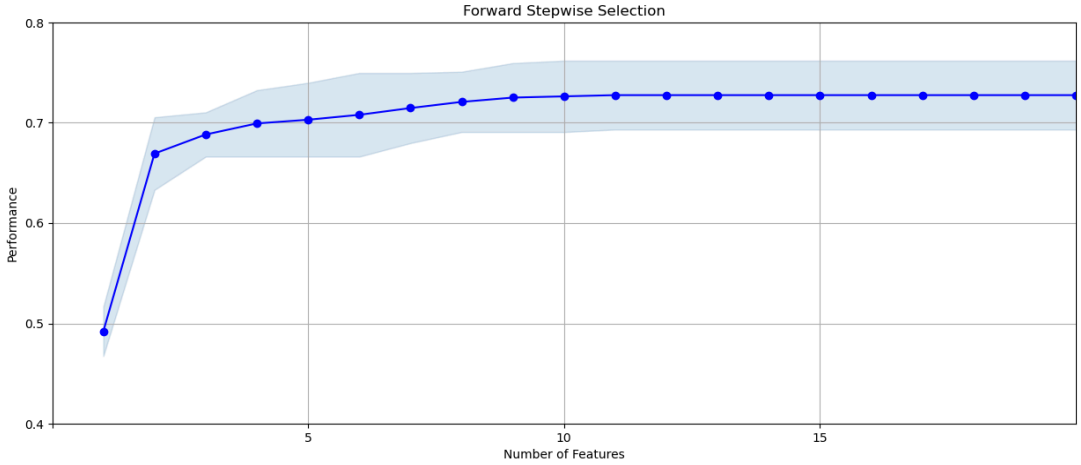
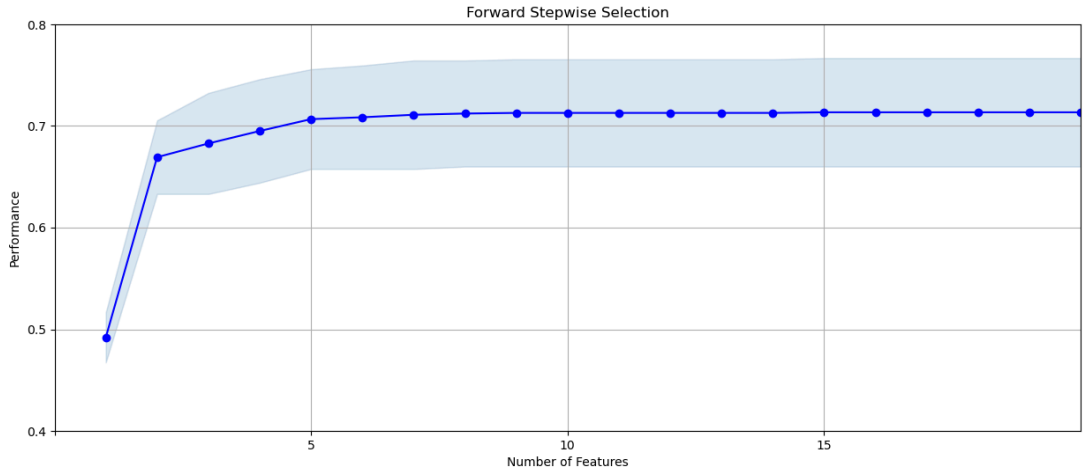


Figure 7

LightGBM (LGBM): Filter 200 and Wrapper 20



Backward Selection Experiments

Figure 8

LightGBM (LGBM): Filter 100 and Wrapper 20

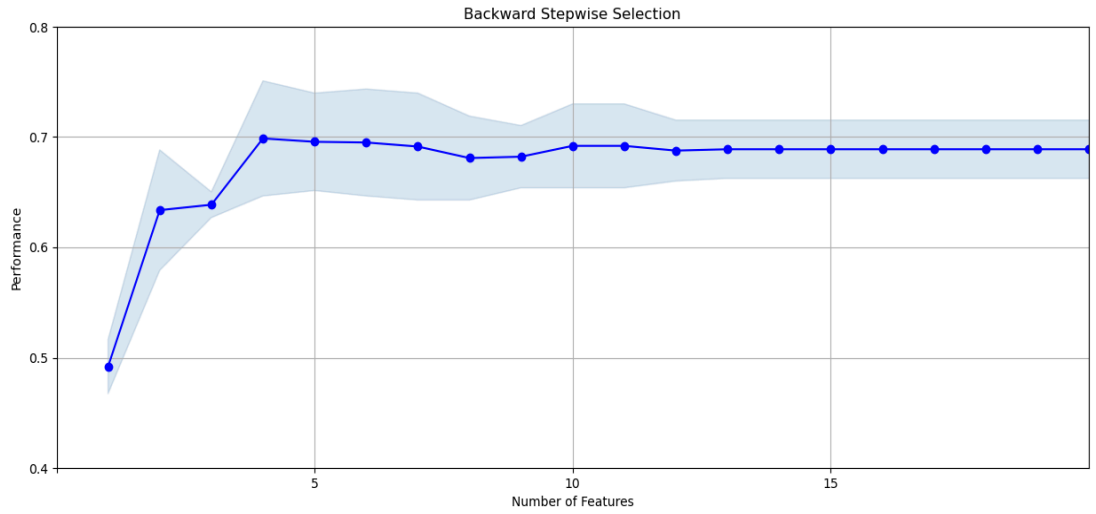
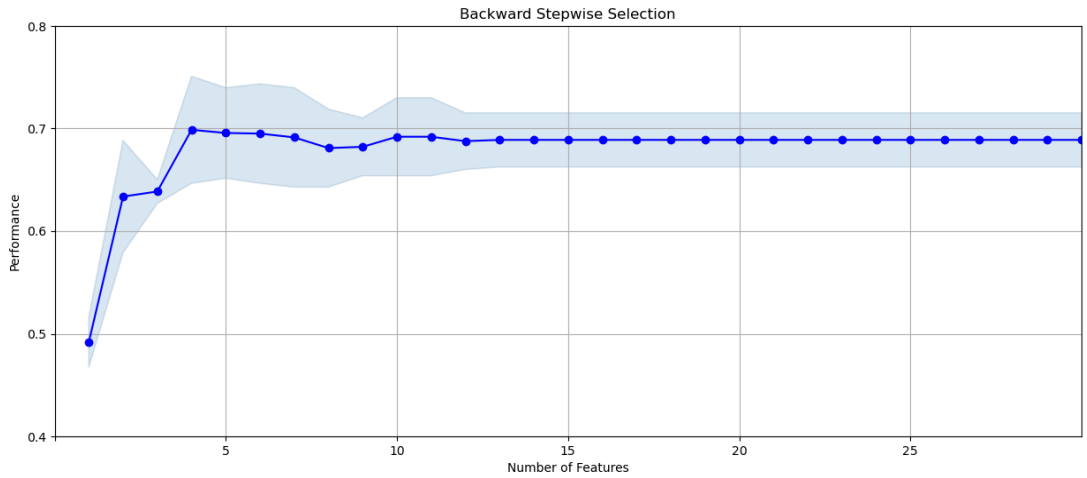


Figure 9

LightGBM (LGBM): Filter 100 and Wrapper 30



Final Selection

After conducting various feature selection experiments, the forward stepwise selection using LGBM with a filter threshold of 500 and a wrapper of 20 (Figure 6) has yielded the best

performance, slightly exceeding 0.70 in wrapper evaluation. Although another experiment with a filter of 200 and a wrapper of 20 also achieved a performance of 0.70, the final selection with the higher filter consistently shows better results. Therefore, this feature selection approach will be used for the project moving forward.

Final Variables with the Univariate Filter Measure (KS)

Table 4

Final Variables

wrapper order	variable	filter score
1	Cardnum_unique_count_for_card_state_1	0.4760666122777140
2	Card_Merchdesc_State_total_7	0.32466842229574100
3	card_merch_day_since	0.26669923959450200
4	Cardnum_count_1_by_30	0.42822889583921400
5	Cardnum_max_14	0.31882556436477600
6	Cardnum_count_3	0.5633564243069750
7	card_zip_total_14	0.33203885270556900
8	Card_dow_unique_count_for_merch_state_1	0.4473572977675280
9	card_merch_count_1_by_60	0.28059866469948500
10	state_des_total_3	0.3155403446112290
11	Card_Merchdesc_day_since	0.26893275690673900
12	Cardnum_count_7	0.526897283405525
13	Card_dow_total_14	0.511203068975262
14	Cardnum_total_14	0.4943749232610200
15	Card_dow_count_7	0.482384034050824
16	Cardnum_actual/toal_0	0.47955008176014900
17	Cardnum_variability_max_1	0.47783592310904000
18	Card_dow_total_30	0.4747594487486570
19	Card_dow_max_14	0.4709750625782850
20	Card_dow_vdratio_0by7	0.4679610397253980

Preliminary Model Explores

In this project, we evaluated and optimized several machine learning algorithms, as shown in Table 5. These included logistic regression, decision trees, random forests, boosted trees (LGBM), neural networks, and CatBoost. For each model, different configurations were fine-tuned to maximize performance and ensure the best fraud detection results. Figure 10 presents a boxplot of the top-performing models selected from each machine learning algorithm.

Figure 10

Performance Boxplot

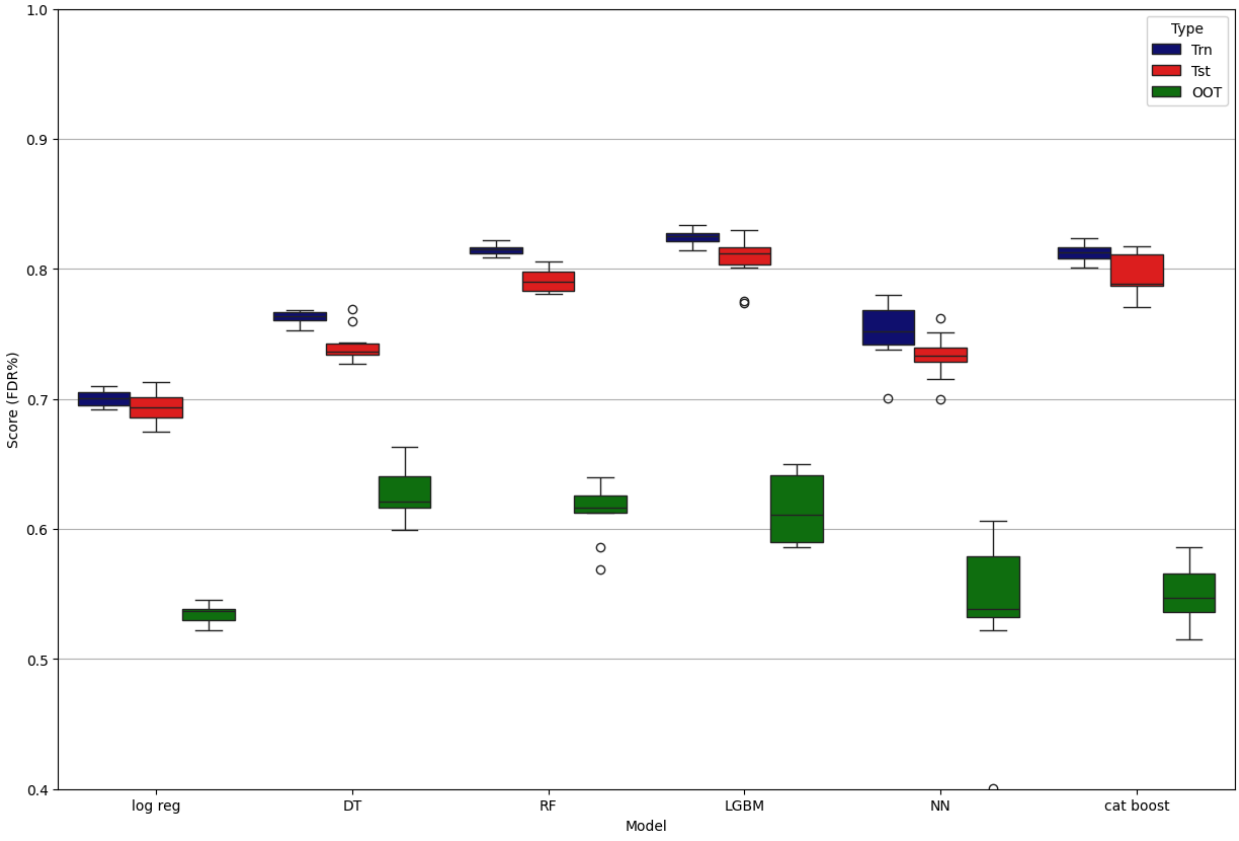


Table 5

Model Exploration

Model	Parameter						Average FDR at 3%		
Logistic Regression	**Number of Variables**	**penalty**	**C**	**solver**		**l1_ratio**	**Train**	**Test**	**OOT**
1	10	l2	1	lbfgs		None	0.697	0.701	0.54
2	10	l2	0.5	lbfgs		None	0.703	0.691	0.541
3	10	l2	0.5	saga		None	0.699	0.694	0.538
4	10	l2	1	saga		None	0.698	0.696	0.54
5	10	None	1	lbfgs		None	0.698	0.698	0.544
6	10	None	0.5	saga		None	0.698	0.699	0.54
7	10	elasticnet	0.5	saga		1	0.672	0.673	0.508
8	10	l1	0.1	saga		None	0.674	0.667	0.508
9	10	l2	0.7	lbfgs		None	0.673	0.671	0.508
10	10	elasticnet	1	saga		0.4	0.673	0.672	0.508
11	10	elasticnet	0.5	saga		0.8	0.673	0.67	0.507
DecisionTree	**Number of Variables**	**criterion**	**splitter**	**max_depth**	**min_samples_split**	**min_samples_leaf**	**Train**	**Test**	**OOT**
1	10	gini	best	200	100	50	0.734	0.696	0.519
2	10	gini	random	200	120	60	0.693	0.686	0.502
3	10	gini	random	100	120	60	0.698	0.689	0.508
4	10	gini	best	100	200	100	0.72	0.697	0.517
5	10	gini	best	100	80	40	0.74	0.692	0.517
6	10	gini	best	300	240	120	0.719	0.701	0.515
7	10	gini	best	150	140	70	0.727	0.693	0.53
8	10	gini	best	200	160	80	0.722	0.695	0.514
9	10	gini	best	100	160	80	0.718	0.7	0.52
RandomForest	**Number of Variables**	**n_estimators**	**crtierion**	**max_depth**	**min_samples_split**	**min_samples_leaf**	**Train**	**Test**	**OOT**
1	10	100	entropy	200	100	50	0.733	0.718	0.514
2	10	100	gini	200	100	50	0.729	0.72	0.512
3	10	50	gini	100	120	60	0.728	0.712	0.517
4	10	100	gini	150	120	60	0.727	0.716	0.516
5	10	50	entropy	100	120	60	0.731	0.712	0.51
6	10	50	entropy	200	180	90	0.718	0.706	0.501
7	10	100	gini	150	200	100	0.72	0.706	0.516
8	10	100	gini	200	80	40	0.739	0.714	0.514
9	10	100	entropy	200	80	40	0.745	0.713	0.511
Boosted Tree (LGBM)	**Number of Variables**	**num_leaves**	**max_depth**	**learning_rate**		**n_estimators**	**Train**	**Test**	**OOT**
1	10	31	-1	0.01		50	0.751	0.724	0.513
2	10	150	5	0.01		100	0.732	0.726	0.522
3	10	100	5	0.01		100	0.736	0.713	0.513
4	10	100	5	0.01		200	0.742	0.729	0.521
5	10	31	10	0.01		150	0.758	0.73	0.521
Neural Network	**Number of Variables**	**hidden_layer_sizes**	**activation**	**alpha**	**learning_rate**	**learning_rate_init**	**max_iter**	**Train**	**Test**	**OOT**
1	10	(100,)	relu	0.0001	constant	0.001	200	0.706	0.7	0.515
2	10	(200,)	relu	0.0001	constant	0.01	100	0.708	0.703	0.514
3	10	(200,)	relu	0.001	adaptive	0.001	50	0.818	0.793	0.65
4	10	(100,)	relu	0.0001	adaptive	0.05	50	0.755	0.739	0.561
5	10	(500,)	relu	0.0001	constant	0.005	300	0.711	0.704	0.515
6	10	(150,)	relu	0.001	adaptive	0.005	50	0.705	0.709	0.513
CatBoost	**Number of Variables**	**verbose**	**iterations**	**learning_rate**	**depth**	**l2_leaf_reg**		**Train**	**Test**	**OOT**
1	10	0	5	0.1	4	5		0.658	0.646	0.456
2	10	0	100	0.05	8	5		0.788	0.671	0.561
3	10	0	300	0.01	8	9		0.814	0.797	0.552
4	10	0	50	0.05	4	9		0.716	0.712	0.493
5	10	0	200	0.05	6	7		0.845	0.813	0.593
6	10	0	100	0.01	8	9		0.714	0.717	0.512
7	10	0	100	0.01	4	3		0.599	0.711	0.49
8	10	0	100	0.05	8	9		0.777	0.776	0.576

Final Model Performance

The final model selected for this project is the CatBoost model, configured with several key hyperparameters that influence its performance and efficiency.

1. Iterations = 300 : This parameter specifies the number of boosting iterations or trees to be built.
2. Learning Rate = 0.01 : The learning rate controls how much to change the model in response to the estimated error at each iteration. A smaller learning rate means that the model will make smaller updates to the weights, which can lead to more precise convergence to the optimal solution. However, a lower learning rate requires a higher number of iterations to achieve good performance, making it essential to balance this parameter with the number of iterations.
3. Depth = 8: This parameter indicates the depth of the trees in the model. A greater depth allows the model to capture more complex relationships in the data by creating more splits in the decision trees.
4. L2 Leaf Regularization = 9: This regularization parameter helps prevent overfitting by penalizing large weights in the model. L2 regularization works by adding a penalty proportional to the square of the magnitude of the coefficients, effectively discouraging overly complex models.
5. Verbose = 0: This parameter controls the amount of output during the model training process. Setting verbose to 0 suppresses detailed logging of the training process, which can be useful for reducing clutter in the console output, especially when running multiple iterations. This setting is particularly helpful when focusing on model performance without unnecessary information during training.

These hyperparameters collectively enhance the CatBoost model's ability to accurately detect fraudulent transactions while maintaining efficiency and preventing overfitting. Below are the result tables for the training (Table 6), testing (Table 7), and OOT (Table 8) evaluations, reflecting the model's performance across these different datasets .

Table 6

Training Result

Training	# Records				# Goods		# Bads		Fraud Rate		
	59684				58462		1222		0.020474499		
	Bin Statistics						Cumulative Statistics				
Population Bin %	#Records	#Goods	#Bads	%Goods	%Bads	Total # Records	Cumulative Goods	Cumulative Bads	%Goods	%Bads (FDR)	KS	FPR
1	597	32	565	5.36%	94.64%	597	32	565	0.05%	46.24%	46.18	0.06
2	597	247	350	41.37%	58.63%	1194	279	915	0.48%	74.88%	74.40	0.30
3	597	515	82	86.26%	13.74%	1791	794	997	1.36%	81.59%	80.23	0.80
4	596	556	40	93.29%	6.71%	2387	1350	1037	2.31%	84.86%	82.55	1.30
5	597	570	27	95.48%	4.52%	2984	1920	1064	3.28%	87.07%	83.79	1.80
6	597	575	22	96.31%	3.69%	3581	2495	1086	4.27%	88.87%	84.60	2.30
7	597	592	5	99.16%	0.84%	4178	3087	1091	5.28%	89.28%	84.00	2.83
8	597	586	11	98.16%	1.84%	4775	3673	1102	6.28%	90.18%	83.90	3.33
9	597	585	12	97.99%	2.01%	5372	4258	1114	7.28%	91.16%	83.88	3.82
10	596	590	6	98.99%	1.01%	5968	4848	1120	8.29%	91.65%	83.36	4.33
11	597	588	9	98.49%	1.51%	6565	5436	1129	9.30%	92.39%	83.09	4.81
12	597	588	9	98.49%	1.51%	7162	6024	1138	10.30%	93.13%	82.82	5.29
13	597	587	10	98.32%	1.68%	7759	6611	1148	11.31%	93.94%	82.64	5.76
14	597	596	1	99.83%	0.17%	8356	7207	1149	12.33%	94.03%	81.70	6.27
15	597	595	2	99.66%	0.34%	8953	7802	1151	13.35%	94.19%	80.84	6.78
16	596	593	3	99.50%	0.50%	9549	8395	1154	14.36%	94.44%	80.08	7.27
17	597	594	3	99.50%	0.50%	10146	8989	1157	15.38%	94.68%	79.31	7.77
18	597	595	2	99.66%	0.34%	10743	9584	1159	16.39%	94.84%	78.45	8.27
19	597	593	4	99.33%	0.67%	11340	10177	1163	17.41%	95.17%	77.76	8.75
20	597	597	0	100.00%	0.00%	11937	10774	1163	18.43%	95.17%	76.74	9.26

Table 7

Testing Result

Testing	# Records				# Goods			# Bads		Fraud Rate		
	25580				25052			528		0.020641126		
	Bin Statistics							Cumulative Statistics				
Population Bin %	#Records	#Goods	#Bads	%Goods	%Bads	Total # Records	Cumulative Goods	Cumulative Bads	%Goods	%Bads (FDR)	KS	FPR
1	256	22	234	8.59%	91.41%	256	22	234	0.09%	44.32%	44.23	0.09
2	256	112	144	43.75%	56.25%	512	134	378	0.53%	71.59%	71.06	0.35
3	255	216	39	84.71%	15.29%	767	350	417	1.40%	78.98%	77.58	0.84
4	256	237	19	92.58%	7.42%	1023	587	436	2.34%	82.58%	80.23	1.35
5	256	243	13	94.92%	5.08%	1279	830	449	3.31%	85.04%	81.72	1.85
6	256	253	3	98.83%	1.17%	1535	1083	452	4.32%	85.61%	81.28	2.40
7	256	248	8	96.88%	3.13%	1791	1331	460	5.31%	87.12%	81.81	2.89
8	255	249	6	97.65%	2.35%	2046	1580	466	6.31%	88.26%	81.95	3.39
9	256	252	4	98.44%	1.56%	2302	1832	470	7.31%	89.02%	81.70	3.90
10	256	252	4	98.44%	1.56%	2558	2084	474	8.32%	89.77%	81.45	4.40
11	256	250	6	97.66%	2.34%	2814	2334	480	9.32%	90.91%	81.59	4.86
12	256	251	5	98.05%	1.95%	3070	2585	485	10.32%	91.86%	81.54	5.33
13	255	252	3	98.82%	1.18%	3325	2837	488	11.32%	92.42%	81.10	5.81
14	256	255	1	99.61%	0.39%	3581	3092	489	12.34%	92.61%	80.27	6.32
15	256	253	3	98.83%	1.17%	3837	3345	492	13.35%	93.18%	79.83	6.80
16	256	254	2	99.22%	0.78%	4093	3599	494	14.37%	93.56%	79.19	7.29
17	256	255	1	99.61%	0.39%	4349	3854	495	15.38%	93.75%	78.37	7.79
18	255	253	2	99.22%	0.78%	4604	4107	497	16.39%	94.13%	77.73	8.26
19	256	256	0	100.00%	0.00%	4860	4363	497	17.42%	94.13%	76.71	8.78
20	256	254	2	99.22%	0.78%	5116	4617	499	18.43%	94.51%	76.08	9.25

Table 8

OOT Result

OOT	# Records				# Goods			# Bads		Fraud Rate		
	12232				11935			297		0.024280576		
	Bin Statistics							Cumulative Statistics				
Population Bin %	#Records	#Goods	#Bads	%Goods	%Bads	Total # Records	Cumulative Goods	Cumulative Bads	%Goods	%Bads (FDR)	KS	FPR
1	122	18	104	14.75%	85.25%	122	18	104	0.15%	35.02%	34.87	0.17
2	123	80	43	65.04%	34.96%	245	98	147	0.82%	49.49%	48.67	0.67
3	122	101	21	82.79%	17.21%	367	199	168	1.67%	56.57%	54.90	1.18
4	122	88	34	72.13%	27.87%	489	287	202	2.40%	68.01%	65.61	1.42
5	123	99	24	80.49%	19.51%	612	386	226	3.23%	76.09%	72.86	1.71
6	122	113	9	92.62%	7.38%	734	499	235	4.18%	79.12%	74.94	2.12
7	122	120	2	98.36%	1.64%	856	619	237	5.19%	79.80%	74.61	2.61
8	123	121	2	98.37%	1.63%	979	740	239	6.20%	80.47%	74.27	3.10
9	122	115	7	94.26%	5.74%	1101	855	246	7.16%	82.83%	75.66	3.48
10	122	115	7	94.26%	5.74%	1223	970	253	8.13%	85.19%	77.06	3.83
11	123	119	4	96.75%	3.25%	1346	1089	257	9.12%	86.53%	77.41	4.24
12	122	122	0	100.00%	0.00%	1468	1211	257	10.15%	86.53%	76.39	4.71
13	122	121	1	99.18%	0.82%	1590	1332	258	11.16%	86.87%	75.71	5.16
14	122	120	2	98.36%	1.64%	1712	1452	260	12.17%	87.54%	75.38	5.58
15	123	121	2	98.37%	1.63%	1835	1573	262	13.18%	88.22%	75.04	6.00
16	122	121	1	99.18%	0.82%	1957	1694	263	14.19%	88.55%	74.36	6.44
17	122	122	0	100.00%	0.00%	2079	1816	263	15.22%	88.55%	73.34	6.90
18	123	121	2	98.37%	1.63%	2202	1937	265	16.23%	89.23%	73.00	7.31
19	122	122	0	100.00%	0.00%	2324	2059	265	17.25%	89.23%	71.97	7.77
20	122	120	2	98.36%	1.64%	2446	2179	267	18.26%	89.90%	71.64	8.16

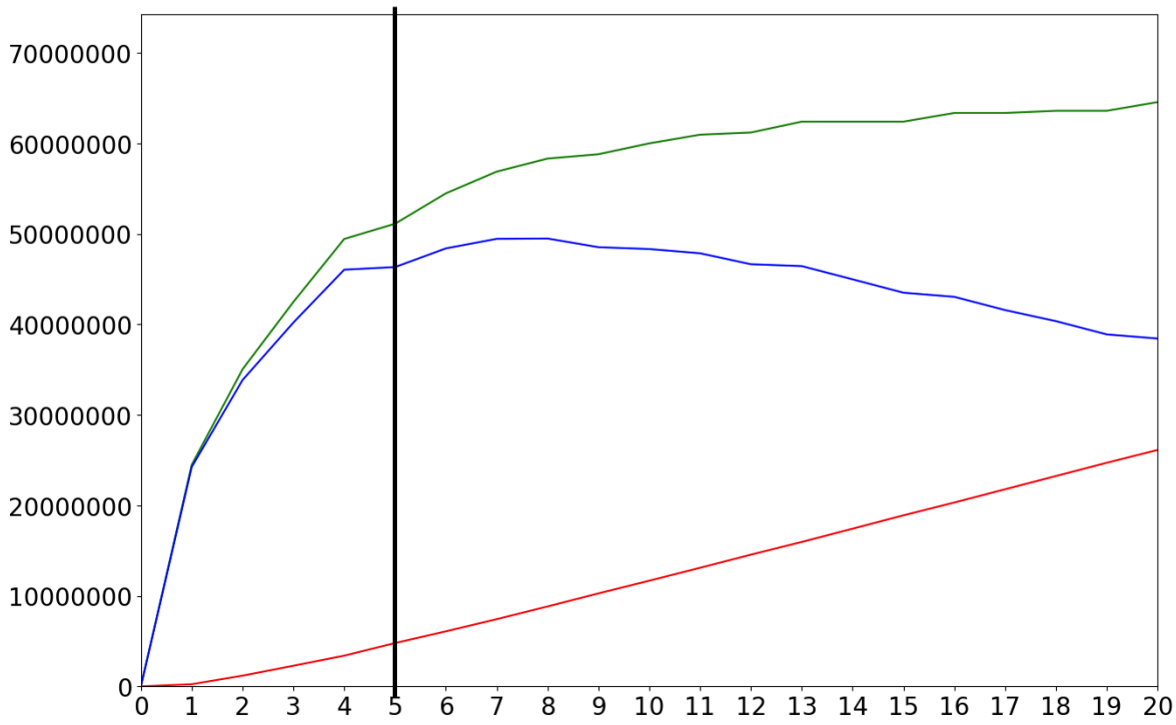
Financial Curves and Recommended Cutoff

The financial analysis is based on the following assumptions regarding the costs and benefits associated with fraud detection. For each fraudulent transaction correctly identified by the model, there would be a gain of $400. On the other hand, each false positive, an instance where a legitimate transaction is incorrectly flagged as fraudulent, incurs a loss of $20.

By analyzing the fraud detection rates and the associated costs and gains, we recommend a score cutoff of 5%. This cutoff balances the trade-off between catching fraudulent transactions and minimizing false positives, as shown in Figure 11. At this level, the model is expected to optimize the overall financial outcome, leading to an estimated annual savings of up to $49,488,000. This figure is derived from the projected number of fraudulent transactions detected multiplied by the gain per fraud, minus the expected losses from false positives. The recommendation aims to maximize the organization's financial return while maintaining a satisfactory customer experience.

Figure 11

Financial Curves



Note. The black line represents the recommended score cutoff at 5%.

Summary

Data Overview & Preprocessing

The dataset used for this analysis consists of 97,852 real credit card transaction records from a US government organization in 2010, with fraud labels marking legitimate (0) and fraudulent (1) transactions. Key fields include Card Number, Merchant Number, Merchant Description, Transaction Amount, and Merchant Location. After exploratory analysis, transactions with non-“P” types (representing purchases) were excluded, and an outlier transaction over $3 million was removed, as it was not representative of the other data. For missing values, a stepwise imputation process was applied. For missing Merchant Numbers (Merchnum), various methods were used, including assigning “unknown” where necessary. Similarly, Merchant States were imputed using zip code mappings and other logic. Remaining missing Merchant Zip fields were filled using dictionaries and, where necessary, the most populous zip codes from each state.

Feature Engineering

Extensive feature engineering was performed to derive transaction-based variables. These included time-based features such as the number of days since the last transaction and frequency of transactions in different time windows (e.g., 1, 7, 30 days). Velocity features, calculating ratios of transaction counts and amounts across different time windows, were also created, along with variability measures for transaction amounts.

Feature Selection

Various feature selection techniques were used to filter down the extensive feature set, including both forward and backward selection methods. LightGBM (LGBM) and Random

Forest (RF) models were used as wrappers, with feature importance filtering applied. The final selection, achieved through forward stepwise selection using LGBM (with 500 filter and 20 wrapper threshold), showed the best performance with a KS score exceeding 0.70.

Model Building

Multiple machine learning algorithms were evaluated, including logistic regression, decision trees, random forests, boosted trees (LGBM), neural networks, and CatBoost. Hyperparameters for each model were optimized through tuning, with CatBoost ultimately selected for its superior performance among all models. The final model was trained using CatBoost with 300 iterations, a learning rate of 0.01, depth of 8, and L2 regularization of 9.

Results & Financial Impact

The CatBoost model's performance was evaluated on training, testing, and out-of-time (OOT) datasets. For practical application, financial cost-benefit analysis was performed, assuming a gain of $400 for every correctly identified fraud and a $20 loss for each false positive. The financial savings curve recommended a cutoff of 5%, resulting in an estimated annual savings of $49,488,000.

Appendix

Data Quality Report for Card Transactions

1. Dataset Overview

   The "Card Transactions" dataset consists of real credit card transaction records collected from a US government organization during the year 2010, spanning from January 1 to December 31. This comprehensive dataset includes 10 fields, featuring essential information such as Card Number, Merchant Number, Merchant Description, and Transaction Amount. Notably, it also includes a fraud label that indicates whether each transaction is legitimate or fraudulent. In total, the dataset comprises 97,852 records, providing a substantial resource for analyzing credit card transactions and detecting fraudulent activities.

2. Field Summary Statistics Tables

   Numeric Field Summary Statistics

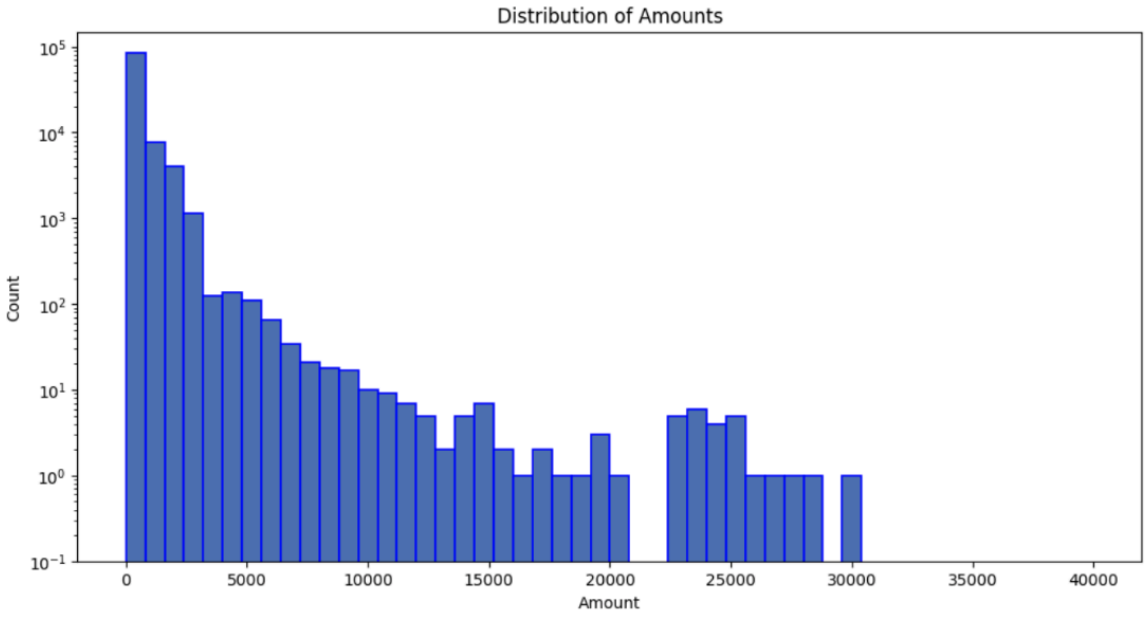
	Field Name	Field Type	# Records Have Values	% Populated	# Zeros	Min	Max	Mean	Standard Deviation	Most Common
0	Amount	numeric	97852	100.0%	0	0.01	3102045.53	425.466438	9949.8	3.62

   Categorical Field Summary Statistics

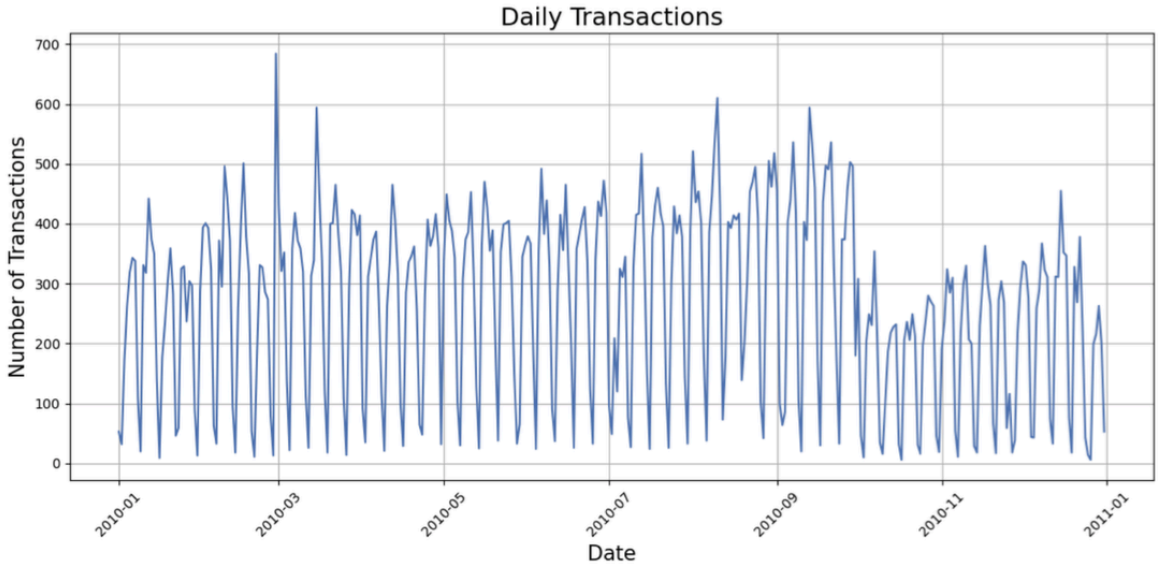
	Field Name	Field Type	# Records Have Values	% Populated	# Zeros	# Unique Values	Most Common
0	Date	categorical	97852	100.0%	0	365	2/28/10
1	Merchnum	categorical	94455	96.5%	0	13091	930090121224
2	Merch description	categorical	97852	100.0%	0	13126	GSA-FSS-ADV
3	Merch state	categorical	96649	98.8%	0	227	TN
4	Transtype	categorical	97852	100.0%	0	4	P
5	Recnum	categorical	97852	100.0%	0	97852	1
6	Fraud	categorical	97852	100.0%	95805	2	0
7	Cardnum	categorical	97852	100.0%	0	1645	5142148452
8	Merchnum	categorical	94455	96.5%	0	13091	930090121224
9	Merch zip	categorical	93149	95.2%	0	4567	38118.0

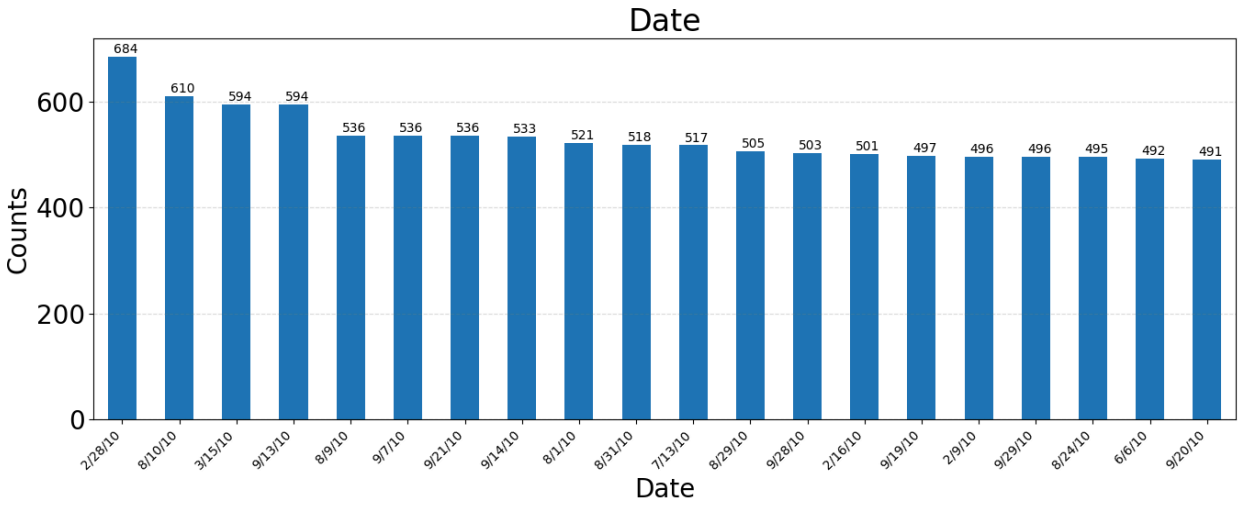
3. Field Distribution

   a. Amount: A numerical field for the amount of the transaction.

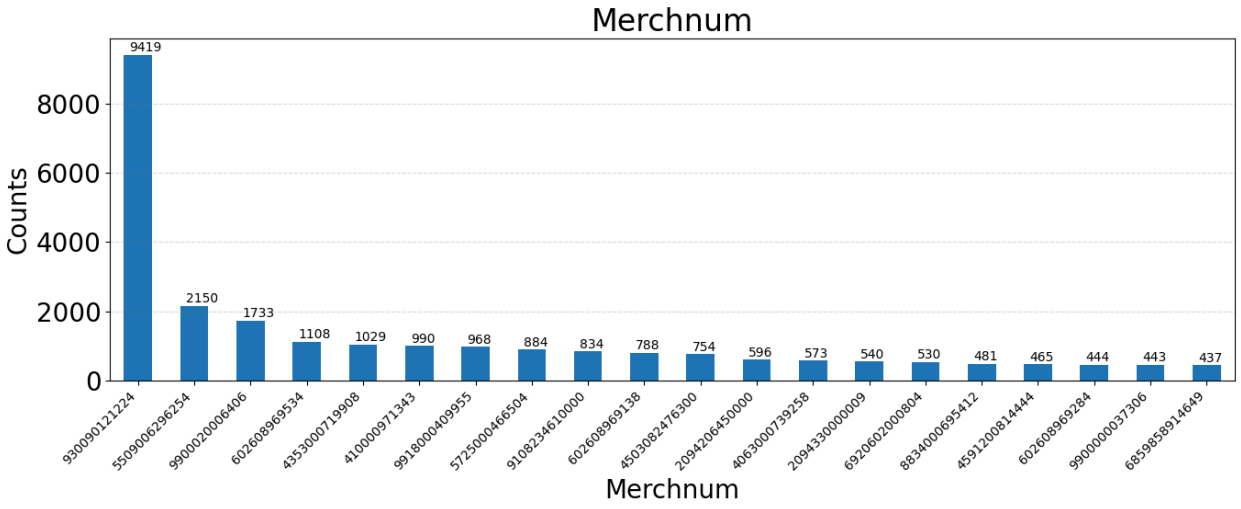


   b. Date: A categorical field containing the date of the transaction

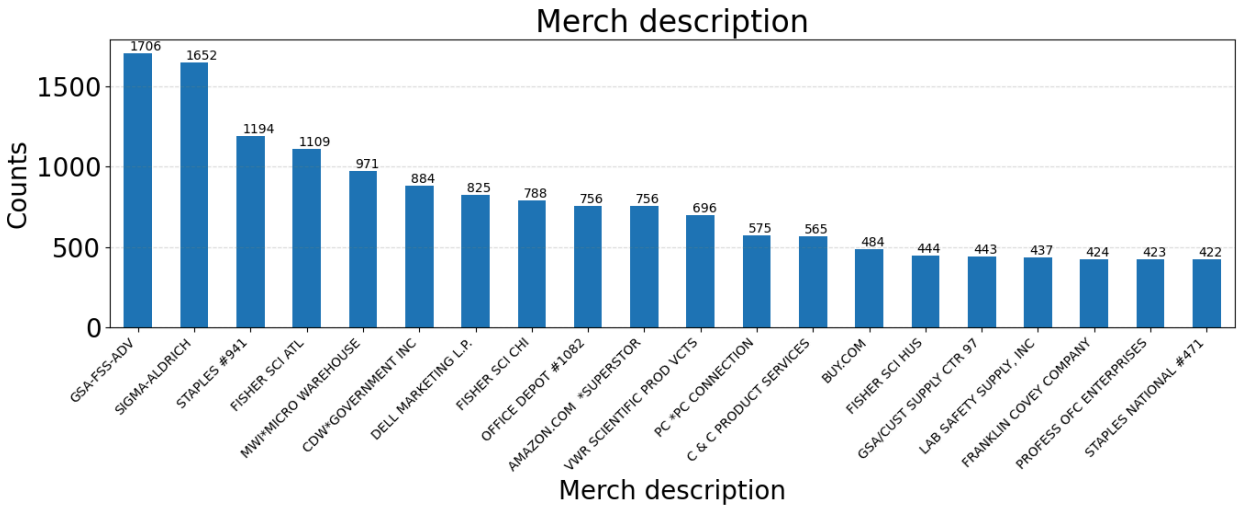




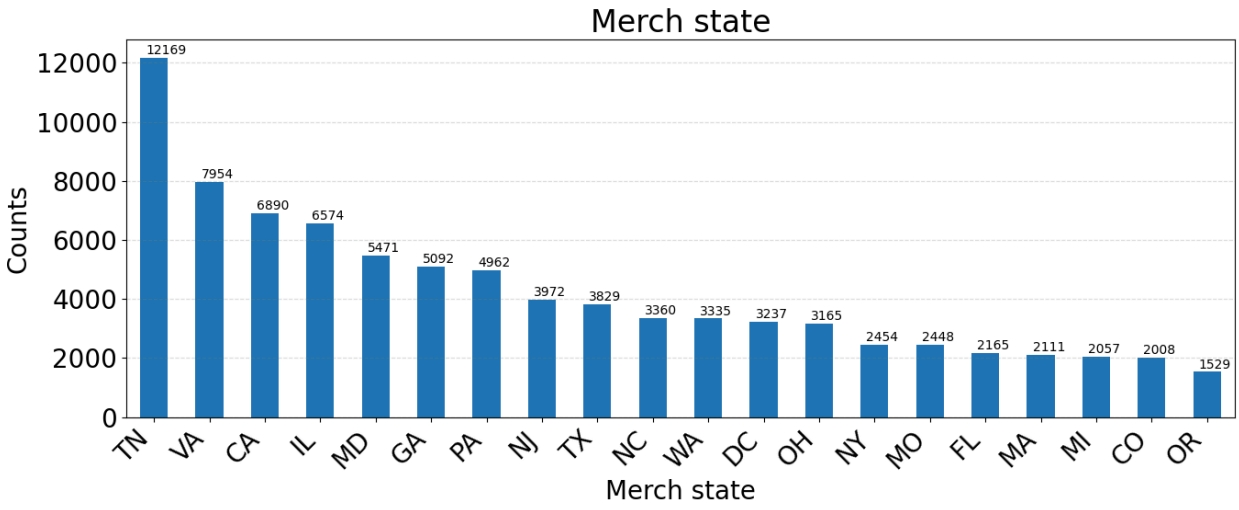
c. Merch num (Merchant Number): A categorical field containing the merchant number which helps in identifying the merchant.



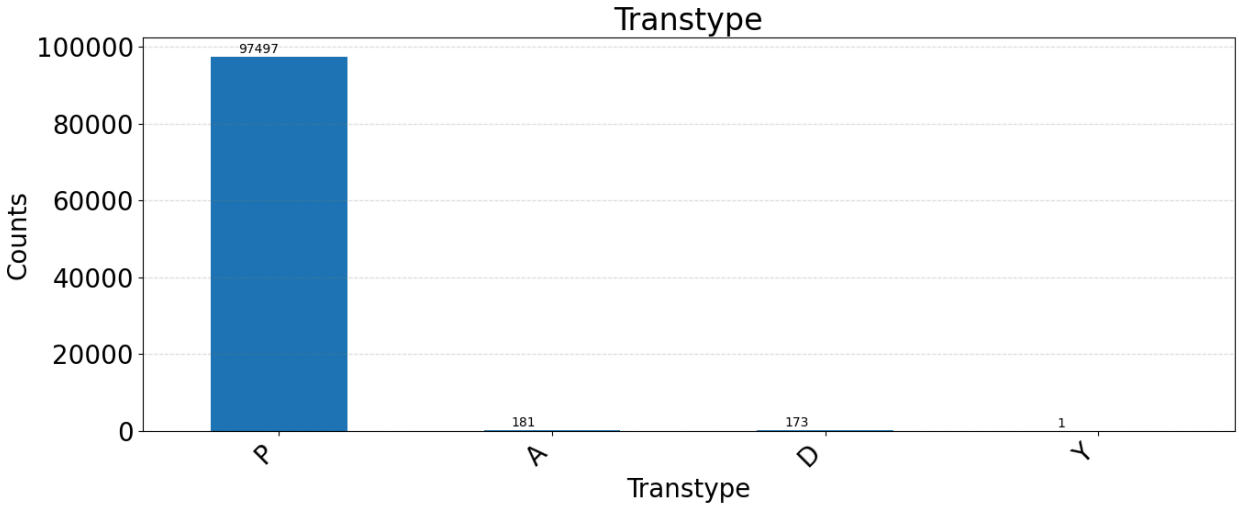
d. Merch description: A categorical field containing details of the merchant.



e. Merch state: A categorical field with the abbreviation of the state where the merchant is based.



f. Transtype (Transaction Type): A categorical field containing the type of transaction. This field consists of 4 translation types: ‘P’, ‘A’, ‘D’, and ‘Y’.



g. Recnum (Record Number): A categorical field containing a unique integer for each record from 1 to 97,852.

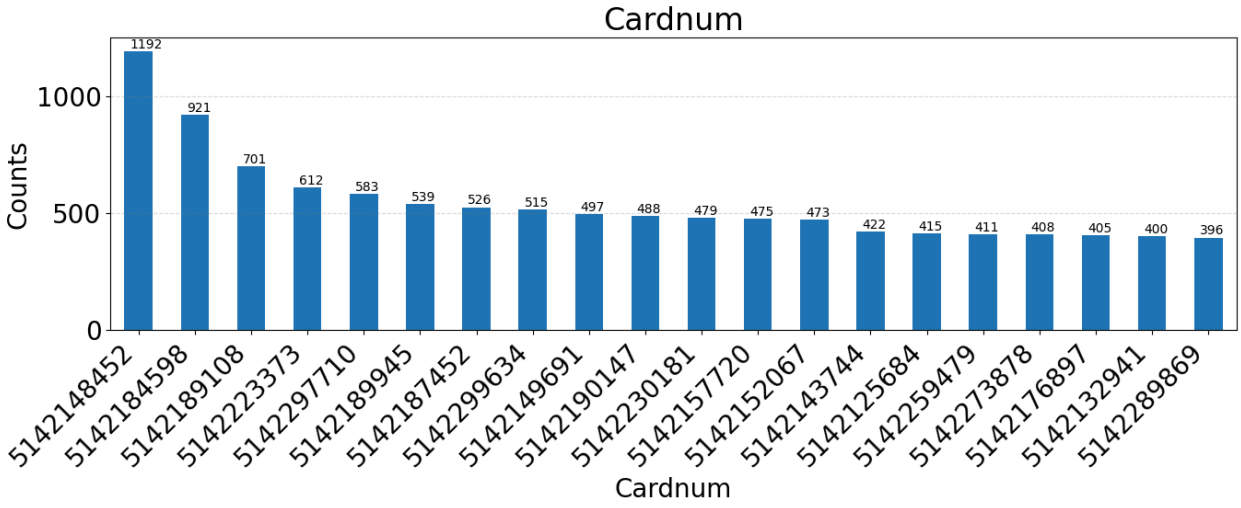
Recnum	Amount
1	3.62
2	31.42
3	178.49
4	3.62
5	3.62
6	3.67
7	3.62
8	230.32
9	62.11
10	3.62

Table containing first 10 values of Recnum and Amount.

h. Fraud: A categorical field containing two categories: 0 – legitimate transaction

1 – fraudulent transaction

i. Cardnum (Card Number): A categorical field for recording the card number that was used for every transaction.



j. Merch Zip (Merchant Zip Code): A categorical field for all the ZIP code of the merchant's location.