Credit Card Transaction Fraud Report

Alexander Tan

Executive summary

Transaction fraud is a common financial fraud that occurs during the account usage process. This report deals with a case of credit card transaction fraud, where an account is compromised due to it being lost, stolen, skimmed, or hacked. According to the 2019 Nilson Report, card fraud costs over $30 billion per year and continues to grow annually. Credit card transaction fraud is a burden for many organizations, and this project attempts to deal with that issue by creating an effective model to predict fraud concerning credit card purchases from a U.S. government organization. The dataset analyzed contains company credit card transaction data from a U.S. government organization in 2010 with 18 fields and 96,753 records. The data contains fields about the transactions such as card number and amount, as well as merchant information like merchant number and merchant zip code. Since the data also has a field identifying whether a transaction is fraudulent or not, the project is a supervised problem. Thus, the project will begin by cleaning the dataset and imputing missing values. Next is to create variables, select the best variables, explore a variety of linear and nonlinear algorithms, and highlight the results of the best performing machine learning model. Finally, using the project's model, we can catch 53.79% of the fraud in the top 3% of records, and we anticipate savings of $21,480,000 /year.

Data description

The dataset contains company credit card transaction data from a U.S. government organization in 2010. The data holds 18 fields with 96,753 records. The following credit card transaction fields are card number, date, transaction type, and amount. The following fields concerning the merchant are merchant number, merchant description, merchant state, and merchant zip code. A fraud field is present to label whether the transaction is legitimate. This field contains binary values of 0 or 1, where 1 is a fraudulent application record and 0 is normal. Most applicant records have a fraud of 0 with 95,694 occurrences. Applicants with fraud of 1 have 1,059 occurrences.

Summary tables

1.1 Numerical table

Field Name	% Populated	Min	Max	Mean	Stdev	% Zero
Date	100.00	2010-01-01	2010-12-31	NA	NA	0.00
Amount	100.00	0.01	3,102,045.53	427.89	10,006.14	0.00

1.2 Categorical table

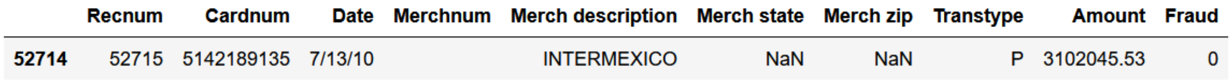
Field Name	% Populated	# Of Unique Values	Most Common Value
Recnum	100.00	96,753	NA
Cardnum	100.00	1,645	5142148452
Merchnum	96.51	13,091	930090121224
Merch description	100.00	13,126	GSA-FSS-ADV
Merch state	98.76	227	TN
Merch zip	95.19	4,567	38118
Transtype	100.00	4	P

Fraud	100.00	2	0

Data cleaning

Since the card transaction dataset is derived from real world data, the data is afflicted with noisy records and missing values. Therefore, it is necessary to exclude certain noisy records and to use data imputation before performing proper data analysis.

Although the data had 18 fields, only 10 fields possessed values in them. The eight extraneous empty fields were dropped. One outlier was removed due to its extremely high value. The extremely high value is believed to be an erroneous input with the amount value labelled in pesos instead of dollars.



	Recnum	Cardnum	Date	Merchnum	Merch description	Merch state	Merch zip	Transtype	Amount	Fraud
52714	52715	5142189135	7/13/10		INTERMEXICO	NaN	NaN	P	3102045.53	0

Transactions that were not purchases, ‘P’, were filtered out from the Transtype field to reduce noise or other unnecessary issues in the analysis.

Within this dataset, the column Merchnum contains 3251 null values, Merch state contains 1020 null values, and Merch zip contains 4300 null values. The data imputation procedure is as follows:

For Merchnum, zero or blank values were first converted to ‘NaN’. Then, missing values were filled by mapping them to the corresponding Merch description field. After this step, the Merchnum column had 2271 null values with 663 unique Merch descriptions remaining. To solve the rest of the null values, new unique Merchnum values were created by taking the maximum Merchnum and adding 1 each time. After filling in these newly created Merchnums by mapping with Merch description, Merchnum had no more null values.

For Merch state, several null values were manually filled by using the corresponding zip code to identify the state like 00926 with PR. Then the rest were mapped to Merchnum and Merch description fields. 346 null values remained.

For Merch zip, missing values were mapped to the corresponding Merchnum and Merch description fields. 2658 null values remained.

For all Merch descriptions labelled as transactions adjustments, the corresponding Merchnum, Merch state, and Merch zip were labelled as ‘unknown’. To remove all missing values, remaining null values were filled with ‘unknown’.

Variable creation

As many variables as possible are created to try to capture various signals of fraud that include bursts of activities at different merchants, card used at merchants never used before, larger than normal purchase amounts, etc. These variables are usually a consequence of how an account is compromised. Situations include a card that is lost or stolen, a common point of compromise like an ATM skimmer or a merchant stealing cards, and online accounts being hacked. Catching these situations usually involves variables that can capture bursts in activity. Example variables for credit card transaction fraud created include amount, frequency, day-since, velocity change, Benford's law, etc. A total of 1,191 variables were created. The table below contains further information on the different types of variables created.

Benford's law states that in many naturally occurring datasets, the leading digit of each number is not uniformly distributed. For example, the digit '1' appears 30% of the time and the digit '9' only appears less than 5% of the time. As a result, Benford's law could capture fraud if the fraudster is not cognizant of this fact and makes up uniformly distributed numbers. This fraudster can be exposed by looking at the distributions and seeing if it deviates from Benford's law.

Note: Benford's law variables were later removed from the analysis due to fear of overfitting during feature selection. These variables are fine for a forensic analysis to look across all the data for anomalies, but for this project they cannot be properly formed for a real time system.

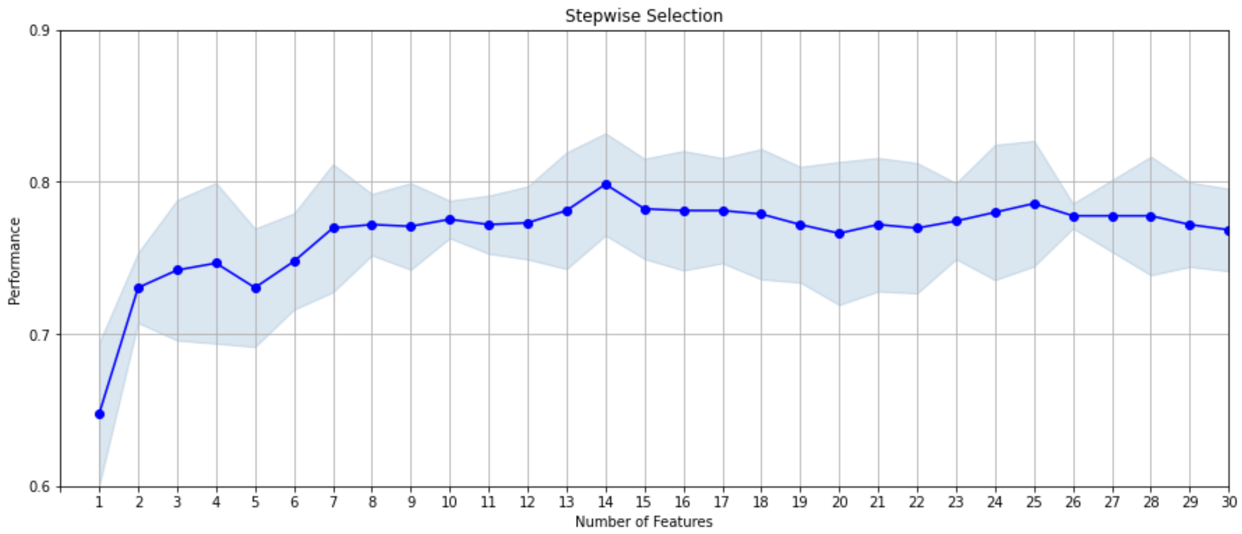
Description of Fields/Entities	**# of Fields/Entities Created**
Original fields from the dataset excluding 'Recnum' and 'Fraud' (Merch description, Merch state, etc.)	8
New entities combining/ concatenating different original fields (zip3, merch_zip, Card_merchdesc, etc.)	8
Description of Variables	**# of Variables Created**
Date of week target encoded: Average fraud percentage of that day i.e., dow_risk	1
Benford's law: Measures unusualness in Cardnum and Merchnum fields which quantifies how different the first digit distribution is from Benford's law distribution	2
Day's since: The # of days since the last time an entity was seen for a transaction record. **Frequency:** # of records with the same entity over the last {0, 1, 3, 7, 14, 30} days **Amount:** The statistics (average, max, median) of the record amount for each entity	550
Velocity Change: The # of records with the same entities in the last {0, 1} divided by the # of records with the same entities in the last {7, 14, 30} days	60

Velocity days since ratio: The ratio containing the # of the days an entity was last seen in {0, 1} days over the # of the days an entity was last seen in {7, 14, 30} days	60
Unique entity counts: Set of variables with the # of unique records in an entity for a particular field	540
Variability: The difference in mean, max, and median for the Amount field in {0, 1, 3, 7, 14, 30} days	180
Amount bins: The scaling of the Amount field into quintile bins	1
Acceleration: The rate of change of the entities {0, 1} day velocity over {7, 14, 30} days velocity.	60

Feature selection

After variable creation, feature selection is performed to cut down the variables to a few 10's. This is necessary because the curse of dimensionality states it becomes increasingly difficult to fit nonlinear models as dimensionality increases. At the end, feature selection allows the exploration of many candidate variables without limit and faster nonlinear model runs to optimize model architecture and hyperparameters. The three ways to categorize feature selection are filter, wrapper, and embedded. Filtering is a univariate model performance measure, wrapper is multivariate, and embedded does feature selection as the model is built using decision trees or regularization. This project uses univariate KS as a filter and forward selection for the wrapper. Additionally since this is a supervised problem, filtering is done first, then a wrapper, and then regularization.

Five different feature selection experiments were conducted. Between forward and backward selection, forward selection models achieved better results. For forward selection, boosted trees had better results, obtaining a wrapper performance of near 0.75, than random forest and neural net. This could be attributed to using a higher num_filter since those models run faster. Then between light-gradient boosted machine (LGBM) and catboost, catboost had more features at around the 0.75 level. LGBM did have slightly more variability in the variable classes. However, catboost had a better wrapper performance so its results were selected to be in the final variable list. The catboost wrapper performance chart and list of final variables are placed below.

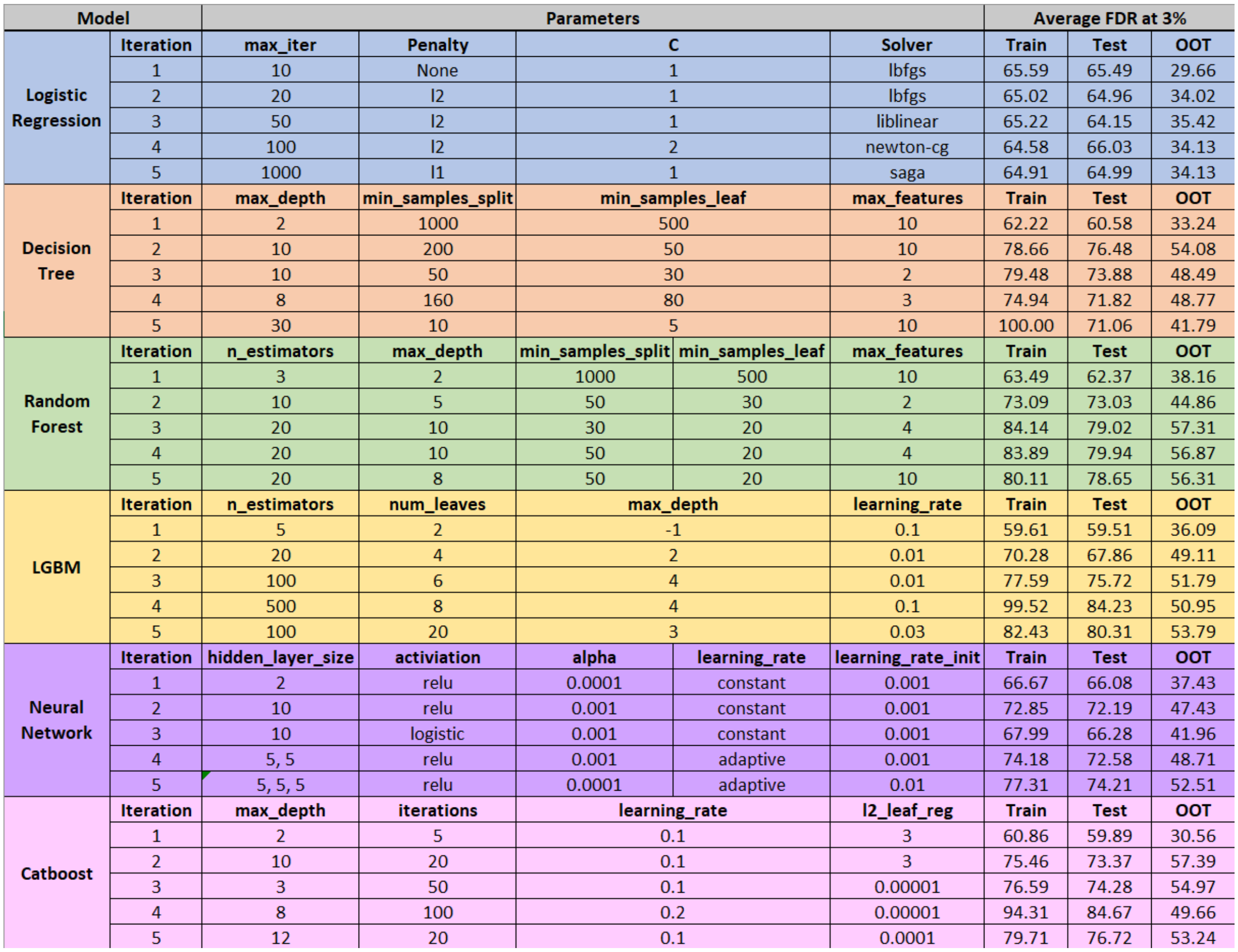


Wrapper Order	Variable	Filter Score
1	card_zip3_total_7	0.69617
2	card_zip3_max_30	0.639979
3	merch_zip_actual/med_30	0.349989
4	zip3_actual/avg_3	0.42114
5	card_merch_total_14	0.676295
6	Merchnum_desc_total_1	0.610207
7	card_zip_total_3	0.677797
8	Merchnum_desc_total_14	0.542433
9	Merchnum_max_30	0.477438
10	merch_zip_total_30	0.428641
11	card_zip3_total_3	0.688293
12	card_zip_max_3	0.64993
13	card_zip3_variability_max_3	0.367327
14	merch_zip_variability_avg_30	0.404345
15	card_zip_med_7	0.553882
16	card_zip_med_14	0.5499
17	zip3_actual/avg_30	0.534554
18	Card_Merchnum_desc_med_7	0.555093
19	Card_Merchdesc_med_30	0.559583
20	Merchnum_total_7	0.583838

Preliminary models exploration

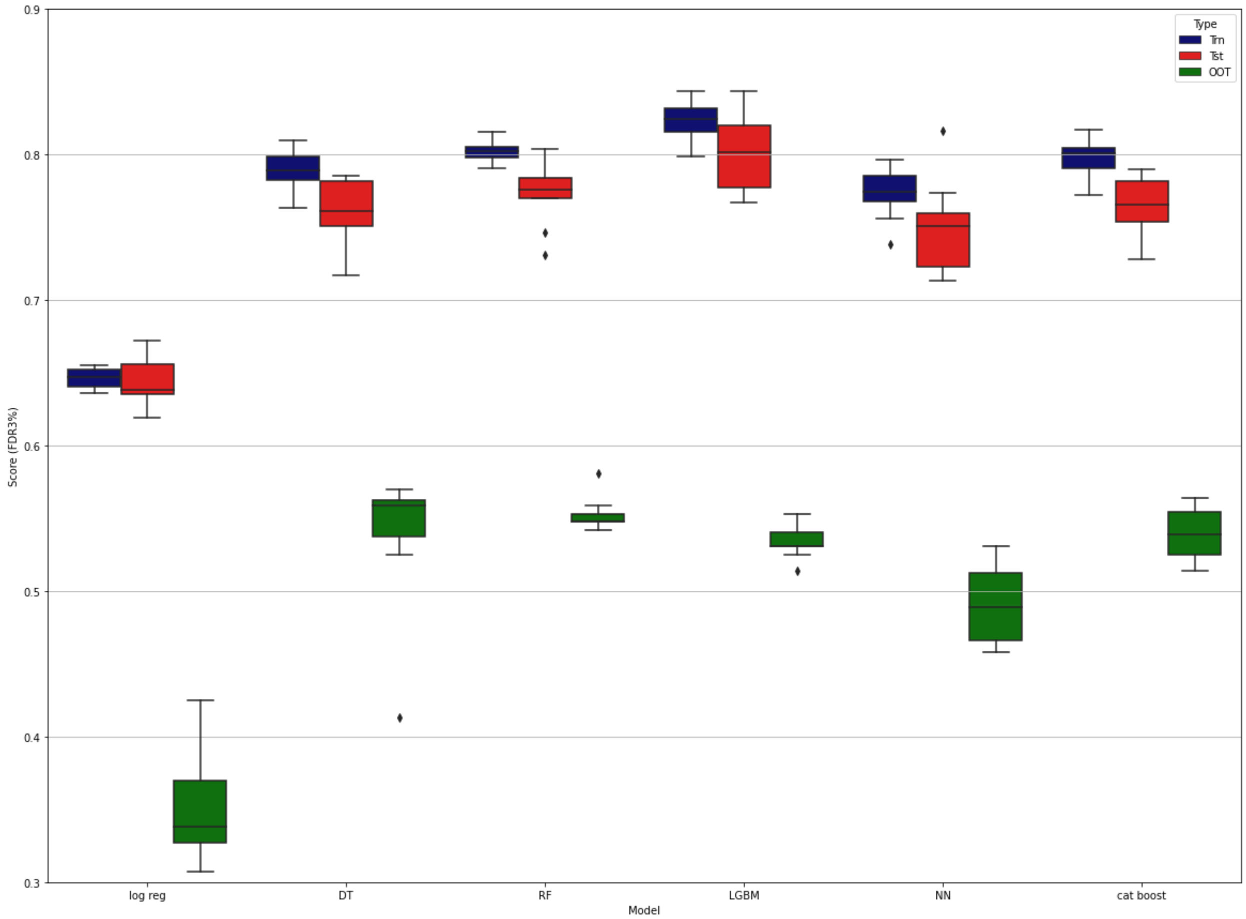
Next in the project, different classification algorithms were tested with various hyperparameters to explore training, testing, and out of time (OOT) results with a fraud detection rate of 3% as the measure of goodness. OOT is the out of time data, which is all data from the original dataset after November 1st, 2010, used to test with the training data. The results are also the average over 10 runs. Machine learning algorithms used are logistic regression, decision tree, boosted trees (LGBM, Catboost), random forest, and neural network. Since logistic regression is a linear model, it is the base model. Nonlinear models like random forest, boosted trees, and neural networks were tested after. The table below illustrates the algorithm results with different hyperparameters:

Hyperparameter Exploration



Model	Parameters						Average FDR at 3%		
Logistic Regression	Iteration	max_iter	Penalty	C		Solver	Train	Test	OOT
	1	10	None	1		lbfgs	65.59	65.49	29.66
	2	20	l2	1		lbfgs	65.02	64.96	34.02
	3	50	l2	1		liblinear	65.22	64.15	35.42
	4	100	l2	2		newton-cg	64.58	66.03	34.13
	5	1000	l1	1		saga	64.91	64.99	34.13
Decision Tree	Iteration	max_depth	min_samples_split	min_samples_leaf		max_features	Train	Test	OOT
	1	2	1000	500		10	62.22	60.58	33.24
	2	10	200	50		10	78.66	76.48	54.08
	3	10	50	30		2	79.48	73.88	48.49
	4	8	160	80		3	74.94	71.82	48.77
	5	30	10	5		10	100.00	71.06	41.79
Random Forest	Iteration	n_estimators	max_depth	min_samples_split	min_samples_leaf	max_features	Train	Test	OOT
	1	3	2	1000	500	10	63.49	62.37	38.16
	2	10	5	50	30	2	73.09	73.03	44.86
	3	20	10	30	20	4	84.14	79.02	57.31
	4	20	10	50	20	4	83.89	79.94	56.87
	5	20	8	50	20	10	80.11	78.65	56.31
LGBM	Iteration	n_estimators	num_leaves	max_depth		learning_rate	Train	Test	OOT
	1	5	2	-1		0.1	59.61	59.51	36.09
	2	20	4	2		0.01	70.28	67.86	49.11
	3	100	6	4		0.01	77.59	75.72	51.79
	4	500	8	4		0.1	99.52	84.23	50.95
	5	100	20	3		0.03	82.43	80.31	53.79
Neural Network	Iteration	hidden_layer_size	activiation	alpha	learning_rate	learning_rate_init	Train	Test	OOT
	1	2	relu	0.0001	constant	0.001	66.67	66.08	37.43
	2	10	relu	0.001	constant	0.001	72.85	72.19	47.43
	3	10	logistic	0.001	constant	0.001	67.99	66.28	41.96
	4	5, 5	relu	0.001	adaptive	0.001	74.18	72.58	48.71
	5	5, 5, 5	relu	0.0001	adaptive	0.01	77.31	74.21	52.51
Catboost	Iteration	max_depth	iterations	learning_rate		l2_leaf_reg	Train	Test	OOT
	1	2	5	0.1		3	60.86	59.89	30.56
	2	10	20	0.1		3	75.46	73.37	57.39
	3	3	50	0.1		0.00001	76.59	74.28	54.97
	4	8	100	0.2		0.00001	94.31	84.67	49.66
	5	12	20	0.1		0.0001	79.71	76.72	53.24

Model performance summary

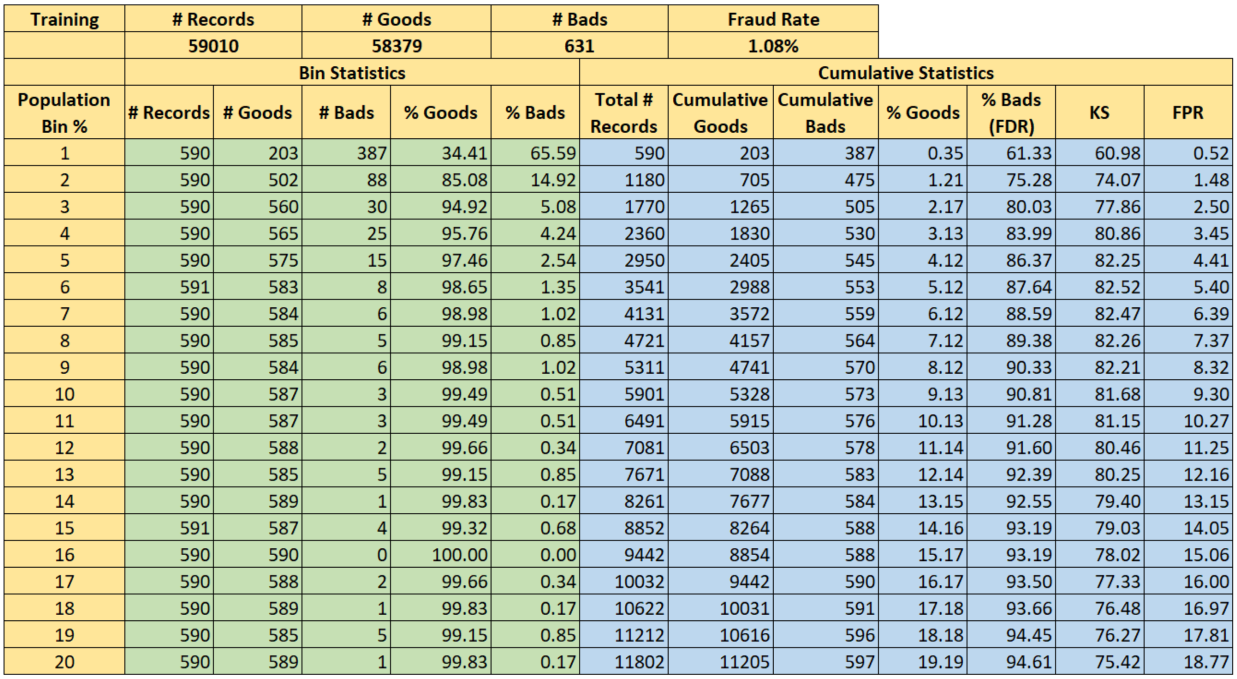


The table above contains a summary of the best performances for each model. The final model chosen is a light gradient-boosted machine with the following hyperparameters: n_estimators=100, num_leaves=20, max_depth = 3, and learning_rate = 0.03. This model and hyperparameters were chosen due to the similar training and test scores as well as having one of the highest OOT scores.

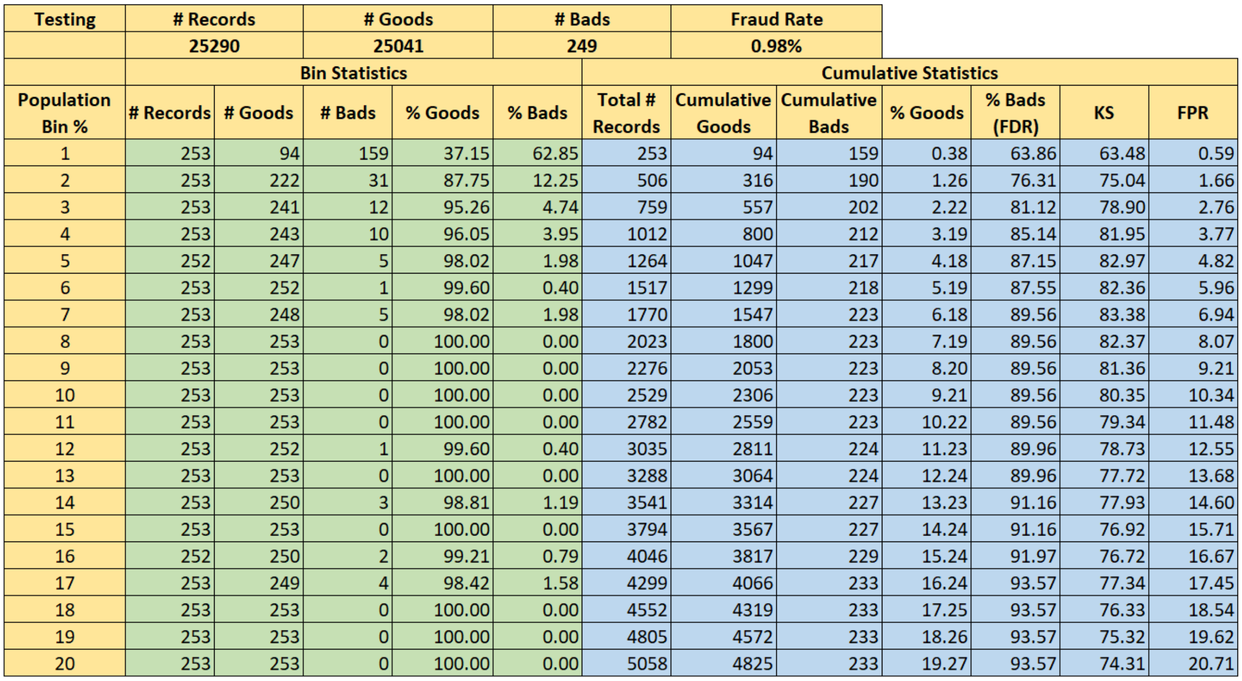
Final model performance

The final model was run on the training, testing, and OOT datasets. Training and testing were scored on data before November 1st 2010, and OOT is scored on data after. First with the training set, the model learns the correlations between characteristics and dependent variables. Then the relationship detection accuracy is assessed with the testing dataset. Finally, the model is tested with the OOT dataset to verify if the model's accuracy falters with the latest data.

As a result, the fraud rate of the training dataset is 1.08%, testing dataset is 0.98%, and OOT dataset is 1.48%. The difference in fraud rate may indicate there is slight overfitting with the training dataset and that the model is a bit too complex. The following three tables are a summary of the top 20 percentile bins from the model's performance on the three datasets. It contains the number of records detected, how many frauds were identified, their percent culmination, KS and FDR columns. The columns KS is a measure for how separate these curves between goods and bads are, and FDR is fraud detection rate which is what percent of all the frauds are caught at a particular cutoff. FDR in this project is 3%.



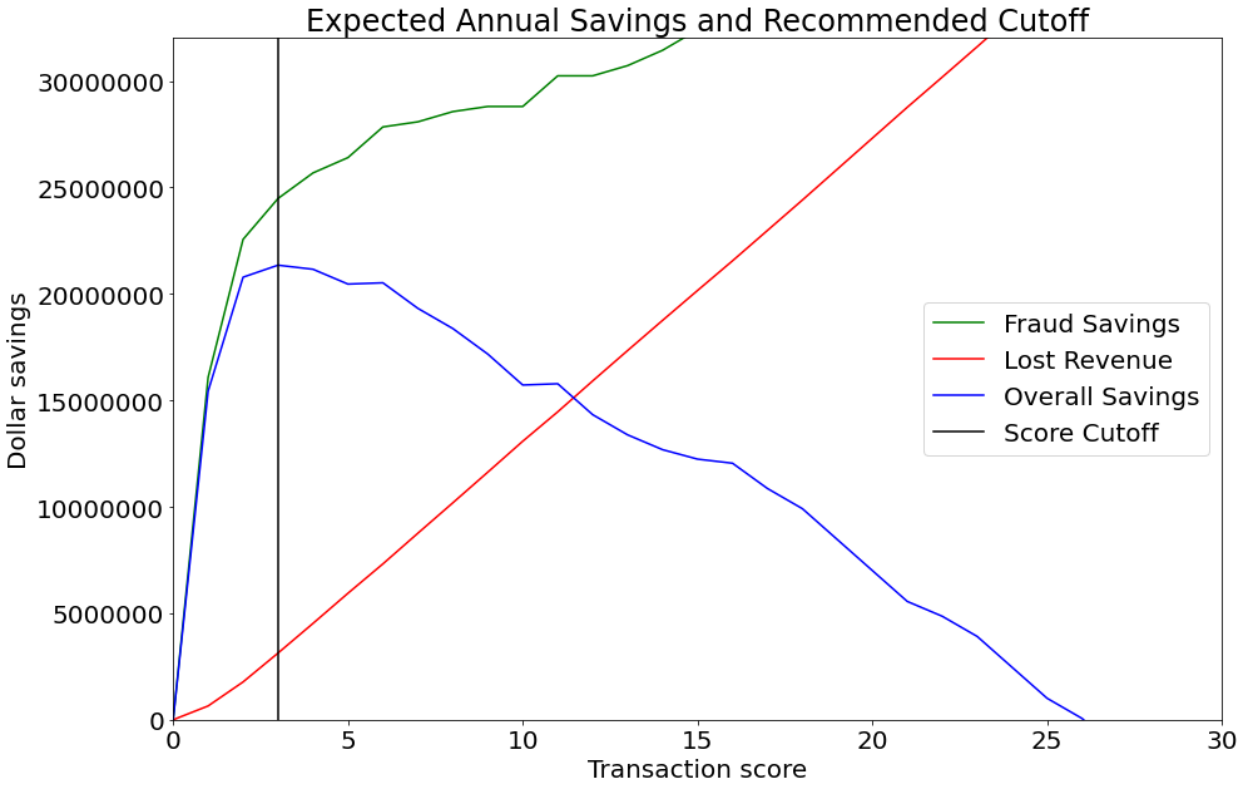
Training	# Records		# Goods		# Bads		Fraud Rate					
	59010		58379		631		1.08%					
	Bin Statistics					Cumulative Statistics						
Population Bin %	# Records	# Goods	# Bads	% Goods	% Bads	Total # Records	Cumulative Goods	Cumulative Bads	% Goods	% Bads (FDR)	KS	FPR
1	590	203	387	34.41	65.59	590	203	387	0.35	61.33	60.98	0.52
2	590	502	88	85.08	14.92	1180	705	475	1.21	75.28	74.07	1.48
3	590	560	30	94.92	5.08	1770	1265	505	2.17	80.03	77.86	2.50
4	590	565	25	95.76	4.24	2360	1830	530	3.13	83.99	80.86	3.45
5	590	575	15	97.46	2.54	2950	2405	545	4.12	86.37	82.25	4.41
6	591	583	8	98.65	1.35	3541	2988	553	5.12	87.64	82.52	5.40
7	590	584	6	98.98	1.02	4131	3572	559	6.12	88.59	82.47	6.39
8	590	585	5	99.15	0.85	4721	4157	564	7.12	89.38	82.26	7.37
9	590	584	6	98.98	1.02	5311	4741	570	8.12	90.33	82.21	8.32
10	590	587	3	99.49	0.51	5901	5328	573	9.13	90.81	81.68	9.30
11	590	587	3	99.49	0.51	6491	5915	576	10.13	91.28	81.15	10.27
12	590	588	2	99.66	0.34	7081	6503	578	11.14	91.60	80.46	11.25
13	590	585	5	99.15	0.85	7671	7088	583	12.14	92.39	80.25	12.16
14	590	589	1	99.83	0.17	8261	7677	584	13.15	92.55	79.40	13.15
15	591	587	4	99.32	0.68	8852	8264	588	14.16	93.19	79.03	14.05
16	590	590	0	100.00	0.00	9442	8854	588	15.17	93.19	78.02	15.06
17	590	588	2	99.66	0.34	10032	9442	590	16.17	93.50	77.33	16.00
18	590	589	1	99.83	0.17	10622	10031	591	17.18	93.66	76.48	16.97
19	590	585	5	99.15	0.85	11212	10616	596	18.18	94.45	76.27	17.81
20	590	589	1	99.83	0.17	11802	11205	597	19.19	94.61	75.42	18.77



Testing	# Records		# Goods		# Bads		Fraud Rate					
	25290		25041		249		0.98%					
	Bin Statistics					Cumulative Statistics						
Population Bin %	# Records	# Goods	# Bads	% Goods	% Bads	Total # Records	Cumulative Goods	Cumulative Bads	% Goods	% Bads (FDR)	KS	FPR
1	253	94	159	37.15	62.85	253	94	159	0.38	63.86	63.48	0.59
2	253	222	31	87.75	12.25	506	316	190	1.26	76.31	75.04	1.66
3	253	241	12	95.26	4.74	759	557	202	2.22	81.12	78.90	2.76
4	253	243	10	96.05	3.95	1012	800	212	3.19	85.14	81.95	3.77
5	252	247	5	98.02	1.98	1264	1047	217	4.18	87.15	82.97	4.82
6	253	252	1	99.60	0.40	1517	1299	218	5.19	87.55	82.36	5.96
7	253	248	5	98.02	1.98	1770	1547	223	6.18	89.56	83.38	6.94
8	253	253	0	100.00	0.00	2023	1800	223	7.19	89.56	82.37	8.07
9	253	253	0	100.00	0.00	2276	2053	223	8.20	89.56	81.36	9.21
10	253	253	0	100.00	0.00	2529	2306	223	9.21	89.56	80.35	10.34
11	253	253	0	100.00	0.00	2782	2559	223	10.22	89.56	79.34	11.48
12	253	252	1	99.60	0.40	3035	2811	224	11.23	89.96	78.73	12.55
13	253	253	0	100.00	0.00	3288	3064	224	12.24	89.96	77.72	13.68
14	253	250	3	98.81	1.19	3541	3314	227	13.23	91.16	77.93	14.60
15	253	253	0	100.00	0.00	3794	3567	227	14.24	91.16	76.92	15.71
16	252	250	2	99.21	0.79	4046	3817	229	15.24	91.97	76.72	16.67
17	253	249	4	98.42	1.58	4299	4066	233	16.24	93.57	77.34	17.45
18	253	253	0	100.00	0.00	4552	4319	233	17.25	93.57	76.33	18.54
19	253	253	0	100.00	0.00	4805	4572	233	18.26	93.57	75.32	19.62
20	253	253	0	100.00	0.00	5058	4825	233	19.27	93.57	74.31	20.71

OOT	# Records		# Goods		# Bads		Fraud Rate					
	12097		11918		179		1.48%					
	Bin Statistics					Cumulative Statistics						
Population Bin %	# Records	# Goods	# Bads	% Goods	% Bads	Total # Records	Cumulative Goods	Cumulative Bads	% Goods	% Bads (FDR)	KS	FPR
1	121	41	80	33.88	66.12	121	41	80	0.34	44.69	44.35	0.51
2	121	111	10	91.74	8.26	242	152	90	1.28	50.28	49.00	1.69
3	121	113	8	93.39	6.61	363	265	98	2.22	54.75	52.53	2.70
4	121	118	3	97.52	2.48	484	383	101	3.21	56.42	53.21	3.79
5	121	118	3	97.52	2.48	605	501	104	4.20	58.10	53.90	4.82
6	121	116	5	95.87	4.13	726	617	109	5.18	60.89	55.72	5.66
7	121	116	5	95.87	4.13	847	733	114	6.15	63.69	57.54	6.43
8	121	118	3	97.52	2.48	968	851	117	7.14	65.36	58.22	7.27
9	121	121	0	100.00	0.00	1089	972	117	8.16	65.36	57.21	8.31
10	121	121	0	100.00	0.00	1210	1093	117	9.17	65.36	56.19	9.34
11	121	121	0	100.00	0.00	1331	1214	117	10.19	65.36	55.18	10.38
12	121	120	1	99.17	0.83	1452	1334	118	11.19	65.92	54.73	11.31
13	121	118	3	97.52	2.48	1573	1452	121	12.18	67.60	55.41	12.00
14	121	118	3	97.52	2.48	1694	1570	124	13.17	69.27	56.10	12.66
15	121	120	1	99.17	0.83	1815	1690	125	14.18	69.83	55.65	13.52
16	121	116	5	95.87	4.13	1936	1806	130	15.15	72.63	57.47	13.89
17	120	120	0	100.00	0.00	2056	1926	130	16.16	72.63	56.47	14.82
18	121	121	0	100.00	0.00	2177	2047	130	17.18	72.63	55.45	15.75
19	121	118	3	97.52	2.48	2298	2165	133	18.17	74.30	56.14	16.28
20	121	118	3	97.52	2.48	2419	2283	136	19.16	75.98	56.82	16.79

Recommended cutoff



To best interpret this project's results for business, the above plot is generated to recommended a Fraud Detection Rate (FDR) cutoff point that would maximize savings. The chart assumes that $400 is saved every time a fraudulent transaction is caught, and $20 is lost for every false positive. Another assumption is this is a sample of 100,000 from a portfolio of 1 million. The recommended cut off point is 3%, so any transaction with a score above the cutoff point threshold is considered fraud. Therefore using the project's model, an estimated $21,480,000 can be saved every year.

Result summary

In short, the results for this credit card transaction fraud analysis are as follows. Concerning the training dataset, the model can eliminate 82.43% of fraud by declining 3% of the total applications. The model can eliminate 80.31% of fraud by declining 3% of the total applications with the testing dataset. The model can eliminate 53.79% of fraud by declining 3% of the total applications with the OOT dataset.

The project began with a data quality report where all data from the 18 fields were analyzed. There are two numerical fields - date and amount - while the rest of the fields were categorical. A distribution is provided for each relevant field with a histogram or line plot, which can be found in the Appendix. Most importantly, the dataset had 1,059 occurrences of fraud out of 96,753 records.

Before further investigation, the data had to be properly cleaned. Eight extraneous empty fields were dropped and one outlier was removed due to its extremely high value. Transactions that were not purchases, 'P', were filtered out. The following fields: Merchnum, Merch state, and Merch zip, contain missing values. Missing values were filled with carefully designed data imputation techniques.

With the data cleaned, as many as possible candidate variables were created. These variables try to capture signals of fraud that include bursts of activities at different merchants, cards used at merchants never used before, larger than normal purchase amounts, etc. 1,191 variables were created that include amount, frequency, day-since, velocity change, Benford's law, etc. Benford's law variables were removed because while they are viable for forensic analysis, they were not properly formed for this project.

After a total of 1,191 variables were created, feature selection was performed through forward selection and a catboost wrapper to select the best 20 variables to deploy in machine learning models. Kolmogorov-Smirnov (KS) and Fraud Detection Rate (FDR) at 3% were used to filter out ineffective variables. Finally, the data is divided into three sections for model analysis: training, testing, and out-of-time (OOT). For Training and testing datasets, the model will use the data before November 1st 2010, and OOT is scored on data after that date.

To determine the best model, multiple models were tested with different hyperparameters. Logistic regression is the base linear model. Nonlinear models like random forest, boosted trees, and neural networks were tested after. Light gradient-boosted machine (LGBM) was the best model due to similar training and testing performance, and good OOT score. With light gradient-boosted machine, 53.79% of fraud can be eliminated by declining 3% of total applications to save around $21,480,000 per year.

In the future, this project can be further improved by exploring other techniques to improve fraud detection. First with feature engineering, candidate variables can be improved by consulting with domain experts to generate more or better variables. Since this dataset is imbalanced in favor of goods, the model's performance can improve by training on a more balanced dataset. This can be done by applying weights to the bads and down sampling the goods. Other methods include capping some variables, making separate models for each type of record, and iterative training with a focus on particular score areas. During feature selection, PCA can be performed next time to remove correlated variables. Finally, collecting more data would make the analysis more robust.

Appendix: Data Quality Report (DQR)

1. Dataset Description

The dataset contains **company credit card transaction data** from a **U.S. government organization** in **2010**. The data holds **18 fields** with **96,753 records**. The following credit card transaction fields are card number, date, transaction type, and amount. The following fields concerning the merchant are merchant number, merchant description, merchant state, and merchant zip code. A fraud field is present to label whether the transaction is legitimate.

2. Summary Tables

2.1 Numerical Table

Field Name	% Populated	Min	Max	Mean	Stdev	% Zero
Date	100.00	2010-01-01	2010-12-31	NA	NA	0.00
Amount	100.00	0.01	3,102,045.53	427.89	10,006.14	0.00

2.2 Categorical Table

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

3. Field Visualization

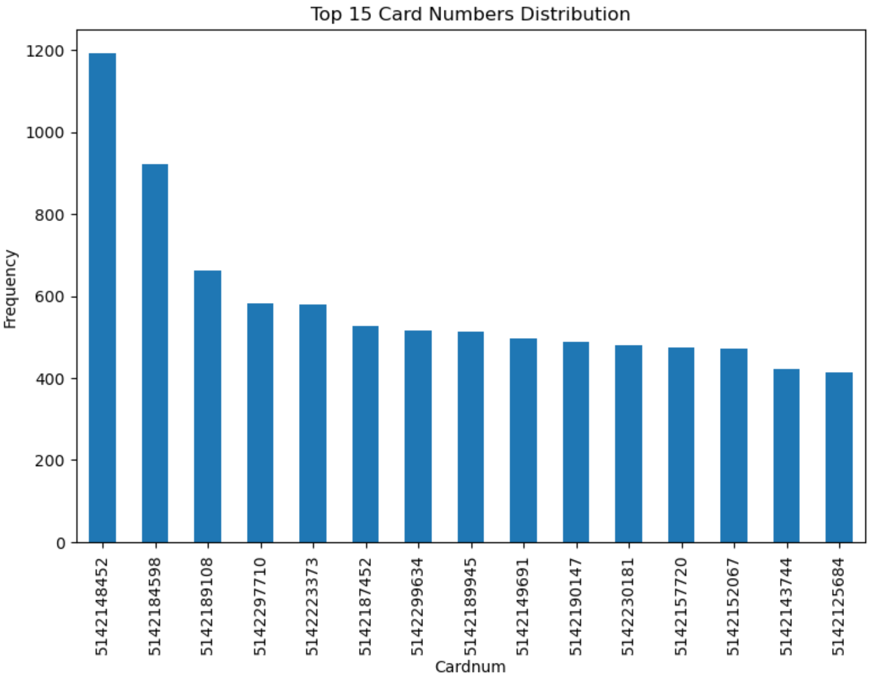
Additional detailed information about each data field in the dataset is provided below. The sections appear in order in which they appear within the dataset.

3.1 Field Name: Recnum

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

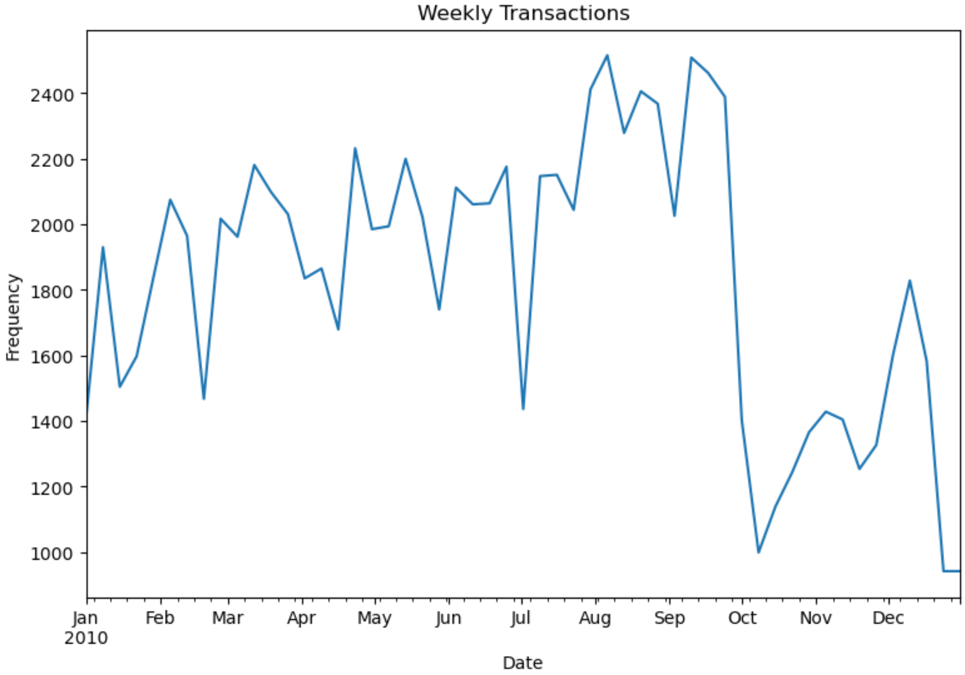
3.2 Field Name: Cardnum

Description: Categorial field of the credit card number involved in a transaction. The most common card number is 5142148452 with 1,192 occurrences. The distribution is the top 15 reoccurring card numbers.



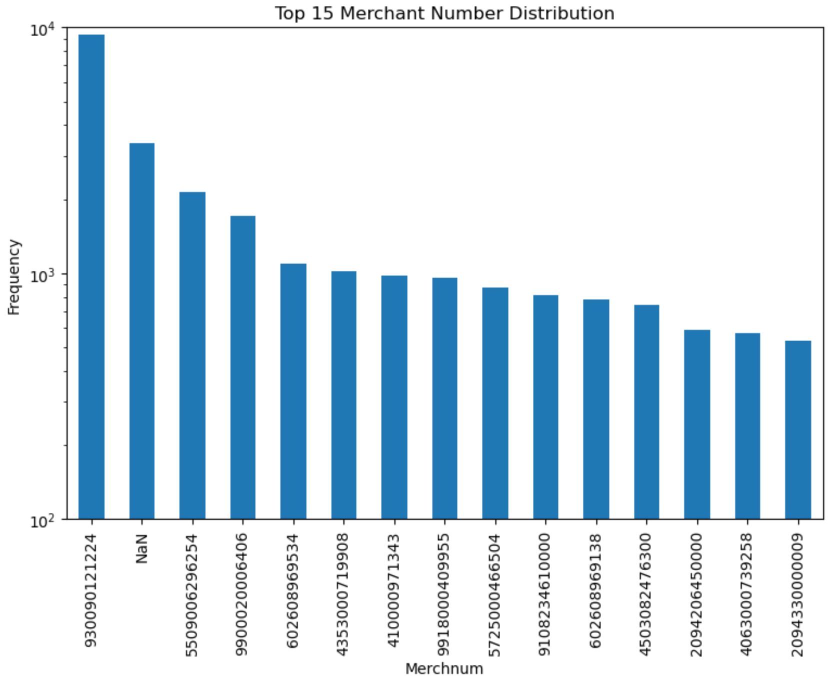
3.3 Field Name: Date

Description: Datetime field containing the date of each credit card transaction. The day with the most transactions was 2010-02-28 with 684 submissions. Weekly transactions are distributed across time (the year 2010).



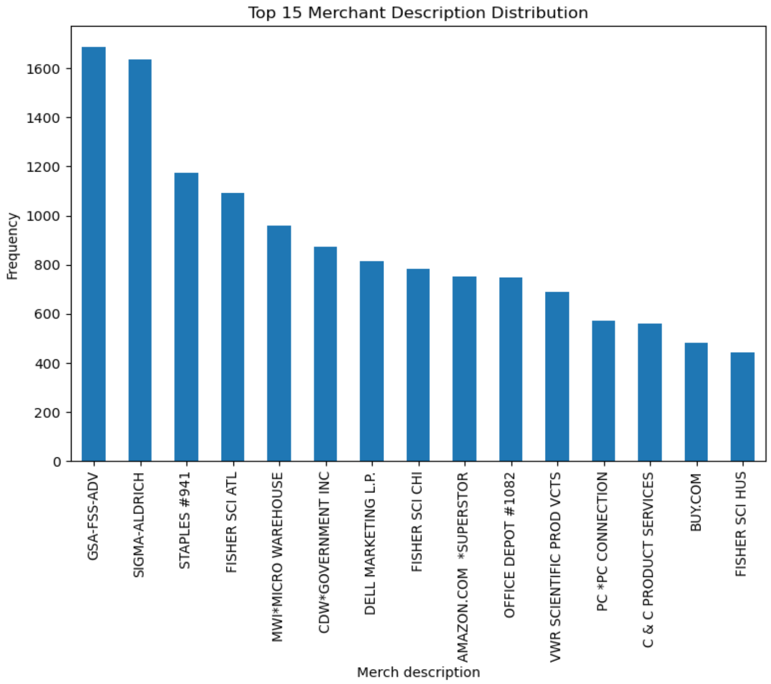
3.4 Field Name: Merchnum

Description: Categorial field of merchant identifying number for every credit card transaction. The most common merchant number is 930090121224 with 9,310 occurrences. This field also contains 3,375 null values. The distribution is the top 15 reoccurring merchant numbers.



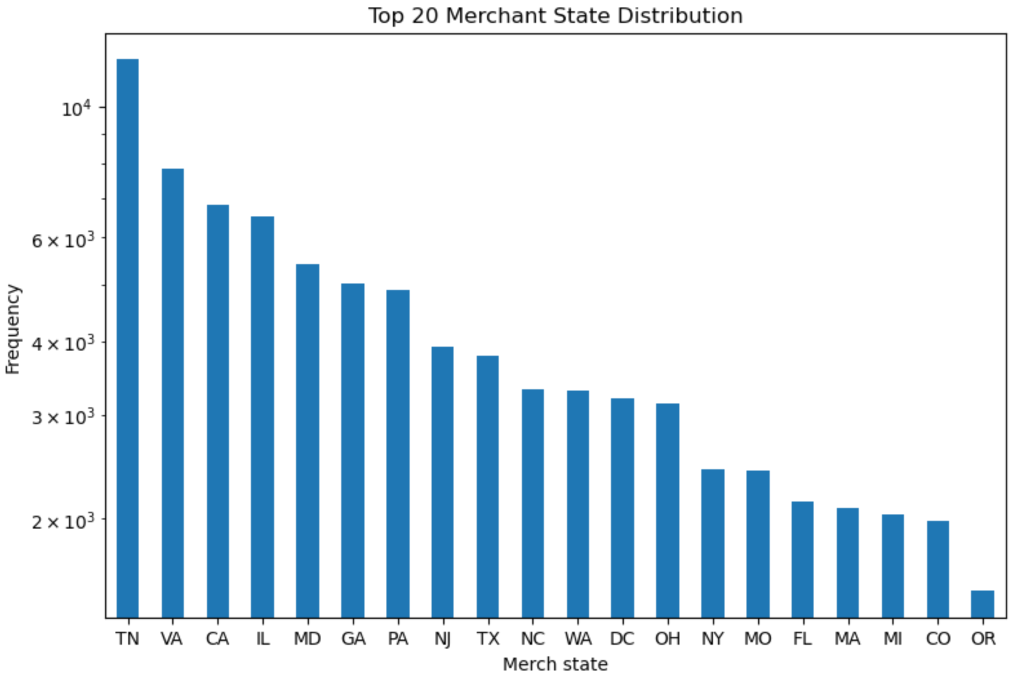
3.5 Field Name: Merch description

Description: Categorial field of the merchant's description for each credit card transaction. The most merchant description is GSA-FSS-ADV with 1,688 occurrences. The distribution is the top reoccurring 15 merchant descriptions.



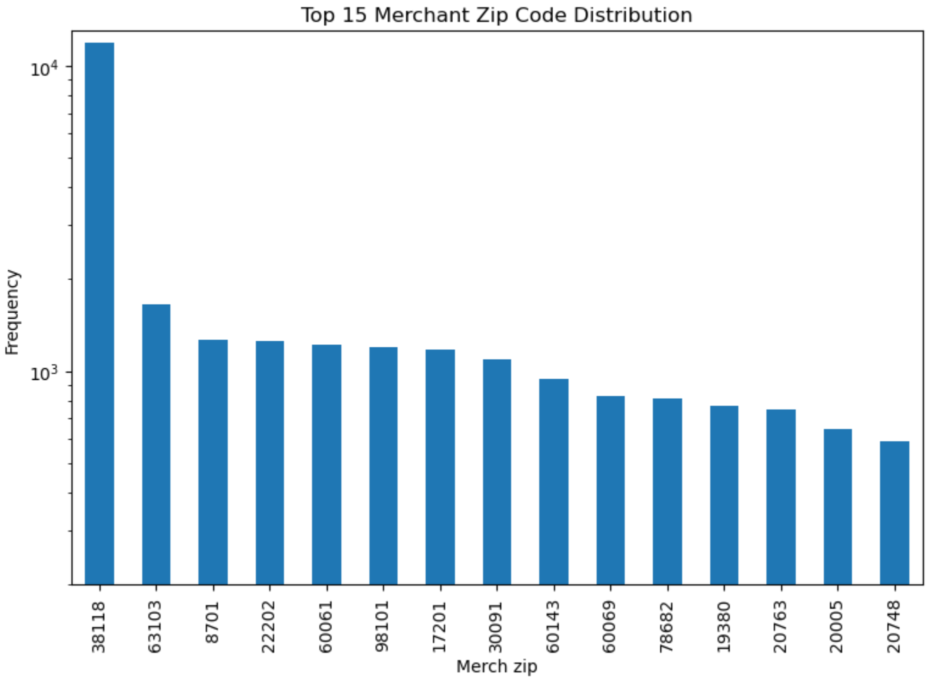
3.6 Field Name: Merch state

Description: Categorial field of the state where each merchant from a transaction is located. The most common state among applicants is TN with 12,035 occurrences. Additionally, this field contains 1,195 null values. The distribution is the top 20 merchant state occurrences.



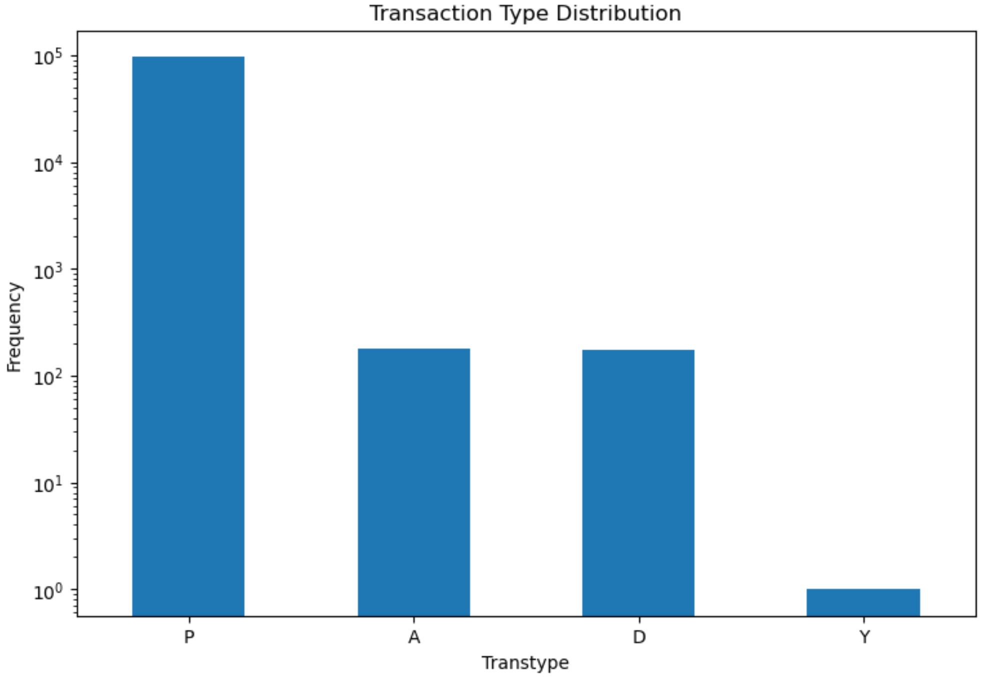
3.7 Field Name: Merch zip

Description: Categorial field of each merchant's zip code from a transaction. The most common zip code among merchants is 38118 with 11,868 occurrences. Additionally, this field contains 4,656 null values. The distribution is the top 15 reoccurring merchant zip codes.



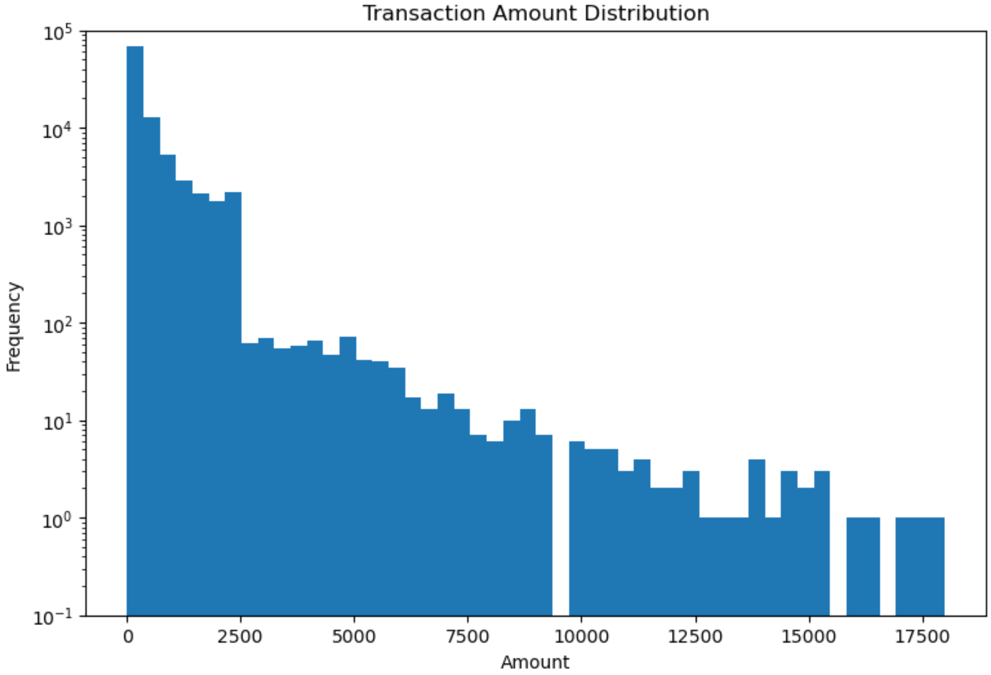
3.8 Field Name: Transtype

Description: Categorial field of each credit card transaction type. The most common transaction type is P, purchases, with 96,398 occurrences.



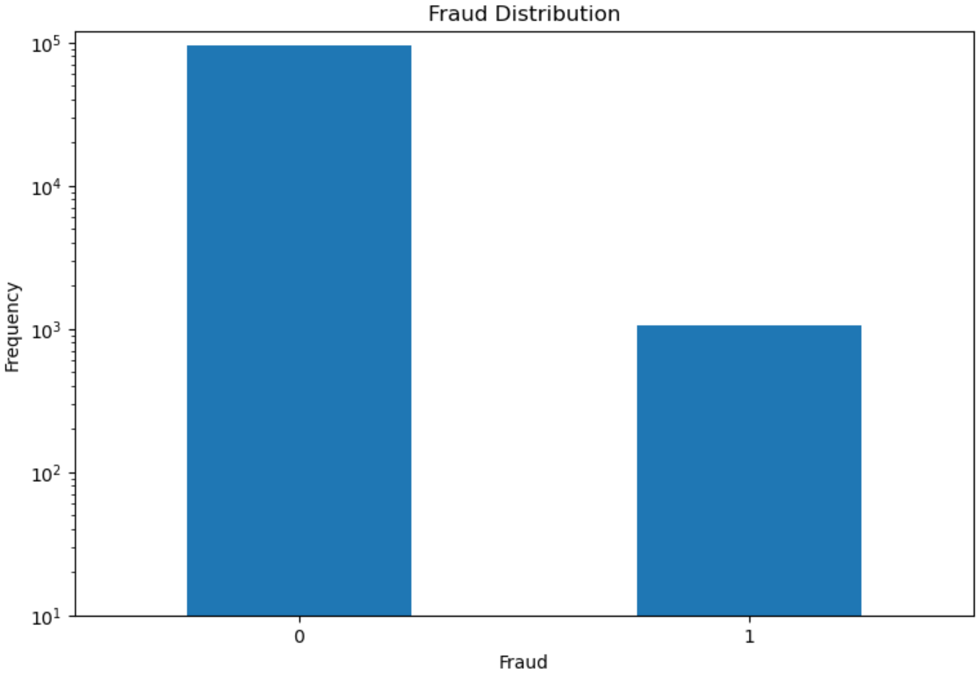
3.9 Field Name: Amount

Description: Numerical field of the amount spent for each credit card transaction. The average amount spent is 427.89 with a standard deviation of 10,006.14. The distribution is the reoccurring transaction amounts.



3.10 Field Name: Fraud

Description: Categorial field determining whether a credit card transaction is legitimate (0) or fraudulent (1). Most transactions are labelled as 0 with 95,694 occurrences. Transactions with fraud as 1 have 1,059 occurrences.


Fraud Distribution
Frequency
10^5
10^4
10^3
10^2
10^1
Fraud