Business Report

Sheetal Srivastava

1. Executive Summary

This report presents the findings from analyzing a Card Transactions dataset, consisting of real credit card transaction information from a U.S. government organization over a one-year period. The objective of the analysis is to flag fraudulent transactions by the means of detecting anomalies in the given data. After evaluating various models, the most effective one was a Decision Tree classifier, achieving an average of 76% accuracy on training data, 73% accuracy on testing data, and **56% accuracy** on out-of-time (OOT) data. After examining the model's performance on the OOT dataset, we determined that setting the optimal Fraud Detection Rate **(FDR) at 3%** would result in potential savings of up to **$21,228,000** based on current transaction records.

2. Data Description

The data consists of real card transactions coming from a U.S. government organization. The data spans over the time period of a year, from **1st January, 2010** to **31st December, 2010**. There are a total of **10 fields** and **96753 records**. Out of these 10 fields, 2 are numeric fields and the remaining 8 are categorical. The statistics for each of these fields have been summarized in two separate tables below.

Numeric Fields Table

Field Name	# Records With Values	% Populated	# Zeros	Min	Max	Mean	Most Common	Stdev
Date	96753	100.00%	0	2010-01-01	2010-12-31	2010-06-25	2010-02-28	98 days
Amount	96753	100.00%	0	0.01	3102045.53	427.89	3.62	10006.14

Categorical Fields Tables

Field Name	# Records With Values	% Populated	# Zeros	# Unique Values	Most Common
Recnum	96753	100.00%	0	96753	1
Cardnum	96753	100.00%	0	1645	5142148452
Merchnum	93378	96.50%	0	13091	930090121224
Merch description	96753	100.00%	0	13126	GSA-FSS-ADV
Merch state	95558	98.80%	0	227	TN
Merch zip	92097	95.20%	0	4567	38118
Transtype	96753	100.00%	0	4	P
Fraud	96753	100.00%	95694	2	0

3. Data Cleaning

a. Cleaning Columns

- The first step is to remove all the empty columns with missing values. This leaves us with the 10 fields that we actually need. Out of these 10 fields, 3 fields have missing values.

b. Exclusions

- We are going to exclude all records with transaction type not equal to “P”, since the majority of records have “P” type transactions and the data is heavily skewed.
- Exclude all transactions with an amount greater than 3000000. This is to get rid of the outlier, and possibly erroneous, value.

c. Imputations / Missing Value Treatment

- Merchnum :
    - Replace “0” with NaN to get the total number of records with missing values for this field as = 3251.
    - Match the Merch Description field of these records with the Merch Description of remaining records and use their Merchnum values to impute the missing values. After this imputation, we are still left with 2094 missing values.
    - Using domain knowledge, for records that have Merch Description value as either “RETAIL CREDIT ADJUSTMENT” or “RETAIL DEBIT ADJUSTMENT”, replace the missing

Merchnum with ‘unknown’. This leaves us with 1403 missing values.
    - For the remaining missing values, create new Merchnum values based on the Merch Description field. For the 1403 records we have 508 unique Merch Description values. We use these descriptions to create 508 new Merchnum values and assign them to replace the missing values.

- Merch state :
    - Total number of records with missing values for this field = 1020.
    - Use available Merch zip, Merchnum, and Merch description values to impute the Merch State by creating a mapping dictionary.
    - Using domain knowledge, for records that have Merch Description value as either “RETAIL CREDIT ADJUSTMENT” or “RETAIL DEBIT ADJUSTMENT”, replace the missing Merchnum with ‘unknown’. This leaves us with 346 missing values.
    - For all non-US states, replace the missing values with a “foreign” tag.
    - Fill the remaining missing values with “unknown”.

- Merch zip :
    - Total number of records with missing values for this field = 4300.
    - Use Merchnum and Merch description to impute the missing values for Merch zip, by creating a mapping. After this imputation we are left with 2658 missing values.
    - Fill the remaining missing values with “unknown”.

d. Additional Preprocessing

- Add leading zeroes to Merchant zip codes.
- Delete extra whitespaces in Merchant description.

4. Variable Creation Process

Field Name / Category	Description	# Variables Created
zip3	Add leading zeroes to Merchant zip codes.	1
Dow	Day of the week (Monday to Sunday) label	1
Dow_Risk	Day of the week target encoded : average fraud percentage of that day over the training dataset (excluding OOT data)	1
state_risk	Merchant state target encoded : average fraud percentage of that particular state	1
Entities	Different combinations of different fields to create new variables (by concatenating field values, e.g. card_zip)	17
Benford's Law variables	Using Benford's Law to get the first digits of transaction amounts and bin them into low, medium, high categories. (Variables created : amount_100, first_digit, bin, etc.)	2*
Days Since	Number of days since the particular field was seen in the data	19
Counts by Entity	Number of unique entities for a particular entity over the last {0, 1, 3, 7, 14, 30, 60} days.	114
Amount by Entity	Number of unique entities for transaction amount over the last {0, 1, 3, 7, 14, 30, 60} days.	114
Velocity	Number of records with the same entity over the last {0, 1, 3, 7, 14, 30, 60} days.	114
Relative Velocity	Number of records with the same entities seen in the past {0, 1} day divided by the number of records with those same entities seen in the last {3, 7, 14, 30} days.	152
Variability	Variability of each entity over the last {0, 1, 3, 7, 14, 30} days with respect to - average, max, and med values for that entity.	342

* Many intermediate variables were created to get these 2 final variables.

5. Feature Selection

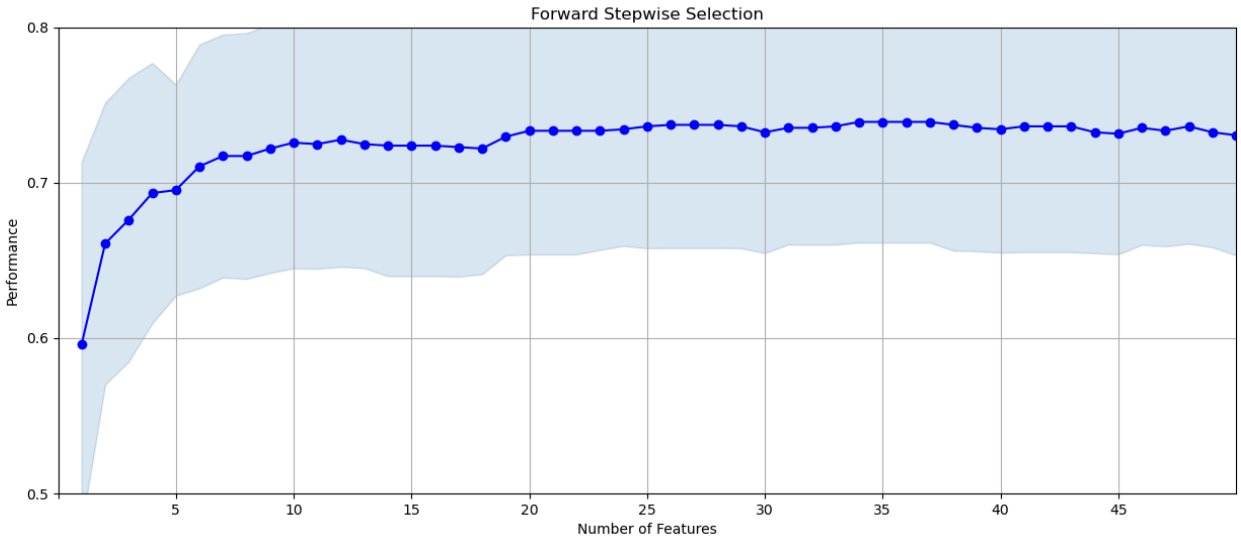
After creating new candidate variables, we must perform dimensionality reduction and retain only the most important set of features to use for modeling. We will first use filters to bring down the variables from thousands to around 200. We use filters initially since they scale linearly with the number of variables and are hence, faster. We then run wrappers to bring down the number of variables further to get our final set of features. We exclude the first few weeks as the variables are not well formed.

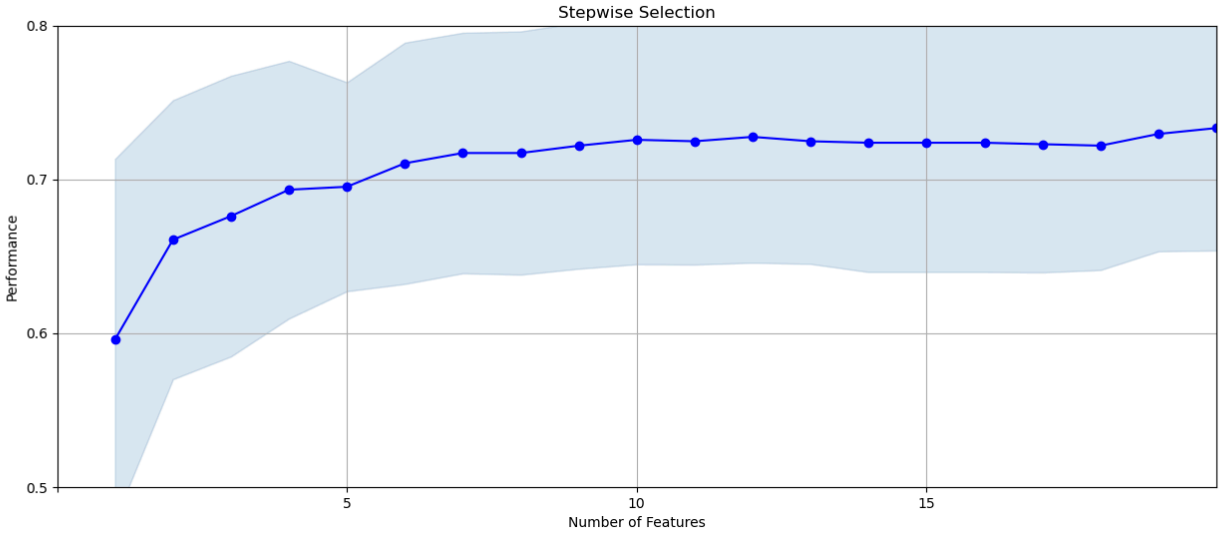
- **Filter** - we run a filter to reduce the feature size by bringing it down to about 200 candidate variables. Here we use the KS filter as a measure of variable goodness. Since the filter is univariate, it always returns the same variables. The table below shows the top 20 variables selected, sorted by filter scores.

Top 20 variables (out of 200) given by Filter Score :

	variable	filter score
0	Fraud	1
1	card_zip3_total_7	0.676549
2	card_zip_total_7	0.666816
3	card_zip3_total_3	0.66026
4	card_zip3_total_14	0.659257
5	card_zip_total_14	0.652244
6	card_zip_total_3	0.652217
7	card_merch_total_7	0.637702
8	card_zip_total_30	0.637171
9	card_zip3_max_7	0.630957
10	card_merch_total_3	0.630782
11	card_zip3_total_30	0.630295
12	card_merch_total_14	0.630048
13	card_zip3_max_14	0.629515
14	card_zip_max_14	0.62793
15	card_zip_max_7	0.625088
16	card_zip_max_30	0.624168
17	Card_Merchdesc_total_7	0.621818
18	card_zip_total_60	0.618103
19	Card_Merchnum_desc_total_7	0.618058
20	card_zip3_total_1	0.615584
21	card_merch_total_30	0.615461

- **Wrapper** - The goal is to reduce the 200 candidate variables to a final set of features. We run the wrapper multiple times since this is stochastic and can give different results. We try forward selection with both Random Forest and LGBM as the classifier. In this case the RF classifier works better. Forward stepwise selection performs better than backward stepwise selection so we focus on that here. The first plot has taken the top 50 features into account. We notice performance saturation at the point around 10 features, therefore we take the top 20 features (double) to be safe. The second plot, zooms into the first one, and shows the performance with only the top 20 features.





- **Final Feature List** -

wrapper order	variable	filter score
1	card_merch_total_14	0.630048056206397
2	card_zip3_max_14	0.6295145774877300
3	zip3_actual/avg_60	0.5111409557513900
4	Card_Merchdesc_med_7	0.48978322433673100
5	Cardnum_total_14	0.5349291801804650
6	card_zip_total_1	0.6107732484946130
7	card_merch_total_0	0.5489019711422850
8	Merchnum_med_0	0.4712586724546150
9	card_zip3_med_3	0.49834945165204500
10	Card_Merchnum_desc_avg_1	0.511187128045461
11	merch_zip_med_0	0.4713019803084940
12	Card_Merchnum_desc_avg_3	0.5207157141968710
13	Card_Merchdesc_avg_3	0.5186539170330310
14	Card_Merchnum_desc_med_1	0.4988925164214630
15	Card_Merchdesc_med_3	0.49815075050297800
16	Card_Merchdesc_med_1	0.49739715454990900
17	Card_Merchnum_desc_med_3	0.4991383412310190
18	card_zip_total_14	0.6522444274591070
19	Merchnum_desc_avg_0	0.5105101791346440
20	card_zip_med_3	0.5032031266367770

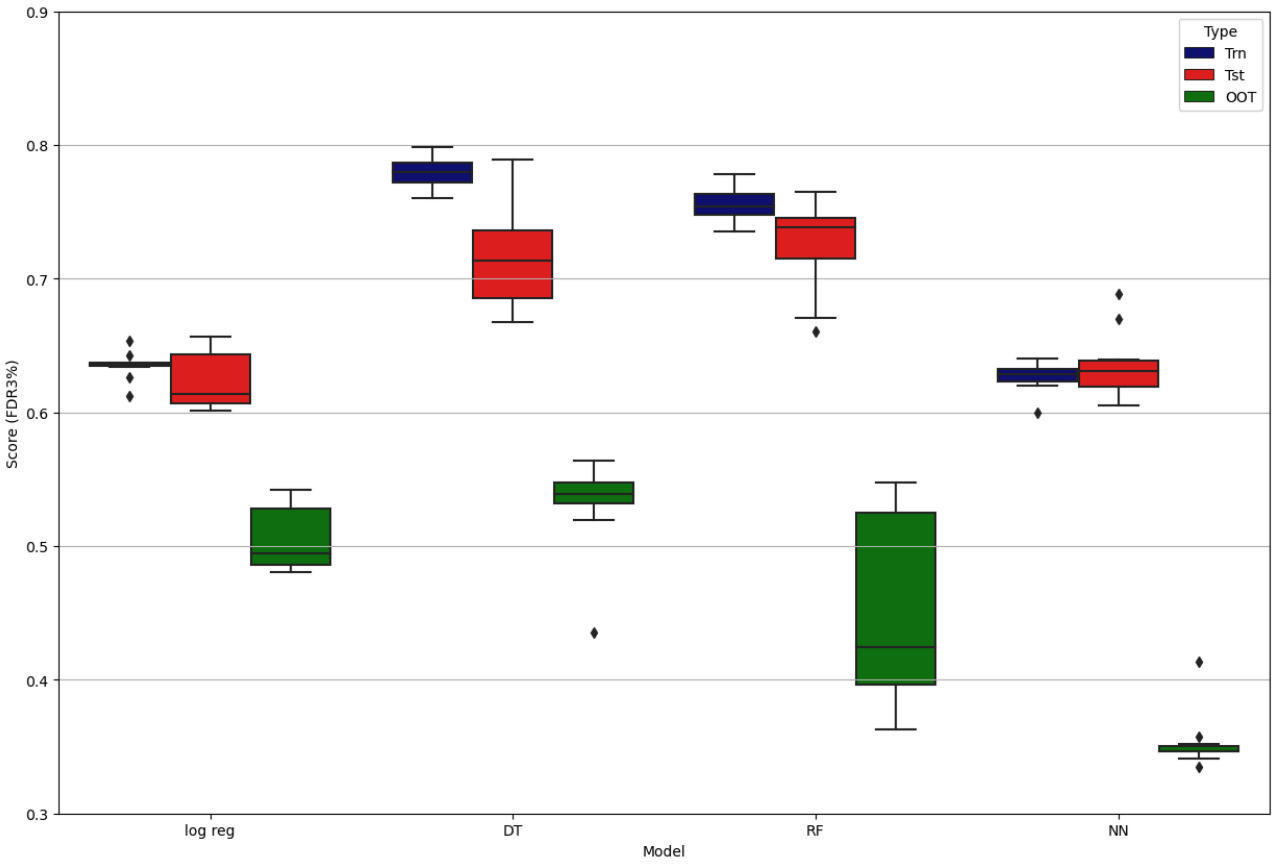
6. Model Exploration

We first start with a simple linear model, like linear/logistic regression, as a baseline model since they are robust and it's harder for them to make mistakes. Our final models should perform at least as well as this baseline model. After this we progress to non-linear models, starting with simple models and slowly increasing complexity. The aim here is to increase complexity till we start observing overfitting. This tells us where to stop, take a step back, and find the sweet spot for the model where it gives us the best performance without overfitting on training data.

Table of model comparison with hyperparameter tuning :

Model		Parameters							Average FDR at 3%		
Logistic Regression	Iteration	penalty		C	solver	l1_ratio			Train	Test	OOT
	1 (default)	l2		1	lbfgs	None			0.6308	0.6274	**0.5335**
	2	l2		0.5	lbfgs	None			0.6248	0.6374	0.5078
	3	elasticnet		1	saga	0.5			0.6261	0.6528	0.5128
	4	elasticnet		0.8	saga	0.3			0.6302	0.6455	0.5229
	5	l2		0.8	saga	None			0.6317	0.6481	0.5162
Decision Tree	Iteration	criterion	splitter	max_depth	min_samples_leaf	min_samples_split			Train	Test	OOT
	1 (default)	gini	best	None	1	2			1	0.5947	0.3016
	2	gini	best	None	40	20			0.8454	0.7486	0.4682
	3	gini	best	100	80	40			0.7998	0.7382	0.5223
	4	gini	best	None	100	50			**0.7725**	**0.7329**	**0.5642**
	5	gini	best	200	150	60			0.7354	0.7156	0.5564
	6	entropy	best	None	1	2			1	0.5768	0.2899
	7	entropy	best	None	100	50			0.7683	0.7208	0.4273
	8	entropy	best	200	100	50			0.7686	0.7195	0.4055
Random Forest	Iteration	n_estimators	criterion	max_depth	min_samples_leaf	min_samples_split			Train	Test	OOT
	1 (default)	100	gini	None	1	2			1	0.8194	0.4357
	2	100	entropy	None	1	2			1	0.8123	0.3754
	3	10	gini	None	50	20			0.8333	0.7811	0.5145
	4	50	gini	None	100	50			0.8043	0.7661	**0.5447**
	5	100	entropy	None	100	50			0.7962	0.7661	0.5547
	6	20	gini	None	200	100			0.7596	0.7444	0.4899
Light GBM	Iteration	boosting_type		num_leaves	max_depth	learning_rate	n_estimators		Train	Test	OOT
	1 (default)	gbdt		31	None (-1)	0.1	100		0.984	0.7953	0.3441
	2	gbdt		20	None (-1)	0.01	50		0.7923	0.7443	0.3731
	3	gbdt		30	None (-1)	0.05	50		0.9148	0.7939	0.3665
	4	gbdt		20	100	0.08	100		0.957	0.8008	0.3698
	6	dart		20	100	0.08	100		0.8938	0.7851	0.3888
Neural Network	Iteration	hidden_layer_sizes	activation	alpha	batch_size	solver	learning_rate	max_iter	Train	Test	OOT
	1 (default)	(100,)	relu	0.0001	auto	adam	constant	200	0.7392	0.7085	0.4408
	2	(200,)	relu	0.0001	auto	adam	constant	200	0.7363	0.7349	0.3849
	3	(100,)	logistic	0.001	auto	adam	adaptive	200	0.6545	0.6479	**0.5408**

Model Comparison (boxplot) :



Based on the model comparison results (as seen in the table), the Decision Tree Classifier works best for the given dataset, with the average highest scores for all three datasets - Train, Test, and OOT. The boxplot confirms this observation, and also shows that the Decision Tree classifier has low variability in performance on OOT data as well.

7. Final Model Performance

The final model selected is a **Decision Tree classifier** with the following model hyperparameters :

- **criterion = gini :** This is the function used to measure the quality of a split. We go with Gini Impurity here, instead of entropy or log loss.
- **splitter = best :** Instead of choosing “random” which chooses the best random split, we go with “best” which chooses the best split
- **max_depth = None :** This specifies the maximum depth of the tree.If None, then nodes are expanded until all leaves are pure or until all leaves contain less than min_samples_split samples.
- **min_samples_leaf = 100 :** This is the minimum number of samples required to be at a leaf node. In this case, a split point at any depth will only be considered if it leaves at least 100 training samples in each of the left and right branches.
- **min_samples_split = 50 :** This is the minimum number of samples required to split an internal node.

Training Summary Table :

Training	# Bins	# Records	# Goods	# Bads	Fraud Rate
	100	58779	58157	622	1.07%

	Bin Statistics					Cumulative Statistics									
Population Bin	# Records	# Goods	# Bads	% Goods	% Bads	Total # Records	Cumulative Goods	Cumulative Bads	Cumulative Goods %	Bads (FDR) %	KS	FPR	Fraud Savings	FP Loss	Overall Savings
0	0	0	0	0.00	0.00	0	0	0	0.00	0.00	0.00	0.00	0	0	0
1	588	259	329	44.05	55.95	588	259	329	0.45	52.89	52.45	0.79	329000	7770	321230
2	588	493	95	83.84	16.16	1176	752	424	1.29	68.17	66.87	1.77	424000	22560	401440
3	587	534	53	90.97	9.03	1763	1286	477	2.21	76.69	74.48	2.70	477000	38580	438420
4	588	552	36	93.88	6.12	2351	1838	513	3.16	82.48	79.32	3.58	513000	55140	457860
5	588	565	23	96.09	3.91	2939	2403	536	4.13	86.17	82.04	4.48	536000	72090	463910
6	588	571	17	97.11	2.89	3527	2974	553	5.11	88.91	83.79	5.38	553000	89220	463780
7	588	576	12	97.96	2.04	4115	3550	565	6.10	90.84	84.73	6.28	565000	106500	458500
8	587	574	13	97.79	2.21	4702	4124	578	7.09	92.93	85.83	7.13	578000	123720	454280
9	588	578	10	98.30	1.70	5290	4702	588	8.09	94.53	86.45	8.00	588000	141060	446940
10	588	579	9	98.47	1.53	5878	5281	597	9.08	95.98	86.90	8.85	597000	158430	438570
11	588	582	6	98.98	1.02	6466	5863	603	10.08	96.95	86.86	9.72	603000	175890	427110
12	587	580	7	98.81	1.19	7053	6443	610	11.08	98.07	86.99	10.56	610000	193290	416710
13	588	582	6	98.98	1.02	7641	7025	616	12.08	99.04	86.96	11.40	616000	210750	405250
14	588	584	4	99.32	0.68	8229	7609	620	13.08	99.68	86.59	12.27	620000	228270	391730
15	588	586	2	99.66	0.34	8817	8195	622	14.09	100.00	85.91	13.18	622000	245850	376150
16	588	588	0	100.00	0.00	9405	8783	622	15.10	100.00	84.90	14.12	622000	263490	358510
17	587	587	0	100.00	0.00	9992	9370	622	16.11	100.00	83.89	15.06	622000	281100	340900
18	588	588	0	100.00	0.00	10580	9958	622	17.12	100.00	82.88	16.01	622000	298740	323260
19	588	588	0	100.00	0.00	11168	10546	622	18.13	100.00	81.87	16.95	622000	316380	305620
20	588	588	0	100.00	0.00	11756	11134	622	19.14	100.00	80.86	17.90	622000	334020	287980
21	588	588	0	100.00	0.00	12344	11722	622	20.16	100.00	79.84	18.85	622000	351660	270340
22	587	587	0	100.00	0.00	12931	12309	622	21.17	100.00	78.83	19.79	622000	369270	252730
23	588	588	0	100.00	0.00	13519	12897	622	22.18	100.00	77.82	20.73	622000	386910	235090
24	588	588	0	100.00	0.00	14107	13485	622	23.19	100.00	76.81	21.68	622000	404550	217450
25	588	588	0	100.00	0.00	14695	14073	622	24.20	100.00	75.80	22.63	622000	422190	199810
26	588	588	0	100.00	0.00	15283	14661	622	25.21	100.00	74.79	23.57	622000	439830	182170
27	587	587	0	100.00	0.00	15870	15248	622	26.22	100.00	73.78	24.51	622000	457440	164560
28	588	588	0	100.00	0.00	16458	15836	622	27.23	100.00	72.77	25.46	622000	475080	146920
29	588	588	0	100.00	0.00	17046	16424	622	28.24	100.00	71.76	26.41	622000	492720	129280
30	588	588	0	100.00	0.00	17634	17012	622	29.25	100.00	70.75	27.35	622000	510360	111640
31	587	587	0	100.00	0.00	18221	17599	622	30.26	100.00	69.74	28.29	622000	527970	94030
32	588	588	0	100.00	0.00	18809	18187	622	31.27	100.00	68.73	29.24	622000	545610	76390
33	588	588	0	100.00	0.00	19397	18775	622	32.28	100.00	67.72	30.18	622000	563250	58750
34	588	588	0	100.00	0.00	19985	19363	622	33.29	100.00	66.71	31.13	622000	580890	41110
35	588	588	0	100.00	0.00	20573	19951	622	34.31	100.00	65.69	32.08	622000	598530	23470

Testing Summary Table :

Testing	# Bins	# Records	# Goods	# Bads	Fraud Rate
	100	25191	24933	258.00	0.01

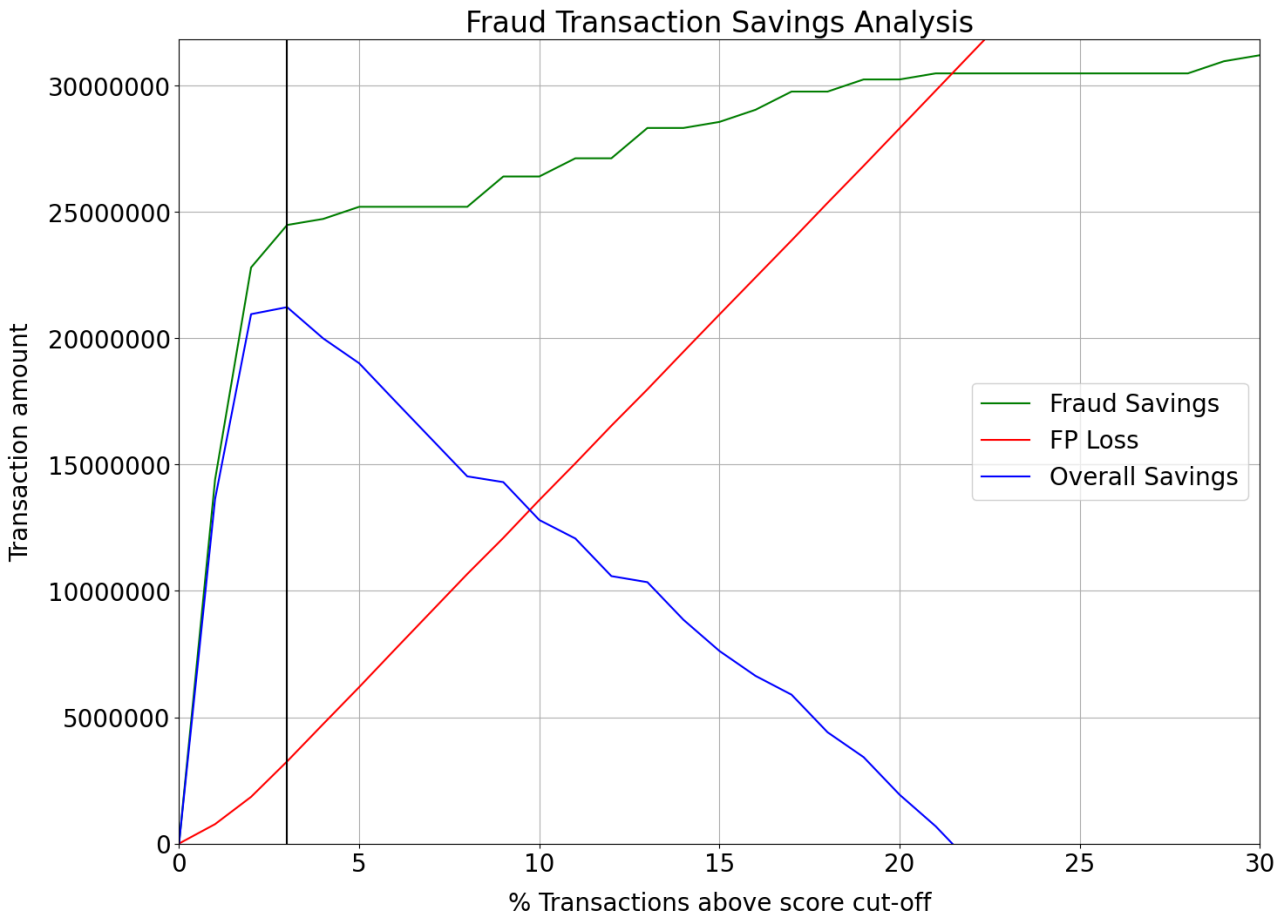
	Bin Statistics					Cumulative Statistics									
Population Bin	# Records	# Goods	# Bads	% Goods	% Bads	Total # Records	Cumulative Goods	Cumulative Bads	Cumulative Goods %	Bads (FDR) %	KS	FPR	Fraud Savings	FP Loss	Overall Savings
0	0	0	0	0.00	0.00	0	0	0	0.00	0.00	0.00	0.00	0	0	0
1	252	123	129	48.81	51.19		252	129	0.49	50.00	49.51	0.95	129000	3690	125310
2	252	213	39	84.52	15.48	504	336	168	1.35	65.12	63.77	2.00	168000	10080	157920
3	252	231	21	91.67	8.33	756	567	189	2.27	73.26	70.98	3.00	189000	17010	171990
4	252	241	11	95.63	4.37	1008	808	200	3.24	77.52	74.28	4.04	200000	24240	175760
5	252	249	3	98.81	1.19	1260	1057	203	4.24	78.68	74.44	5.21	203000	31710	171290
6	251	250	1	99.60	0.40	1511	1307	204	5.24	79.07	73.83	6.41	204000	39210	164790
7	252	247	5	98.02	1.98	1763	1554	209	6.23	81.01	74.78	7.44	209000	46620	162380
8	252	249	3	98.81	1.19	2015	1803	212	7.23	82.17	74.94	8.50	212000	54090	157910
9	252	248	4	98.41	1.59	2267	2051	216	8.23	83.72	75.49	9.50	216000	61530	154470
10	252	248	4	98.41	1.59	2519	2299	220	9.22	85.27	76.05	10.45	220000	68970	151030
11	252	250	2	99.21	0.79	2771	2549	222	10.22	86.05	75.82	11.48	222000	76470	145530
12	252	252	0	100.00	0.00	3023	2801	222	11.23	86.05	74.81	12.62	222000	84030	137970
13	252	250	2	99.21	0.79	3275	3051	224	12.24	86.82	74.58	13.62	224000	91530	132470
14	252	250	2	99.21	0.79	3527	3301	226	13.24	87.60	74.36	14.61	226000	99030	126970
15	252	252	0	100.00	0.00	3779	3553	226	14.25	87.60	73.35	15.72	226000	106590	119410
16	252	252	0	100.00	0.00	4031	3805	226	15.26	87.60	72.34	16.84	226000	114150	111850
17	251	251	0	100.00	0.00	4282	4056	226	16.27	87.60	71.33	17.95	226000	121680	104320
18	252	251	1	99.60	0.40	4534	4307	227	17.27	87.98	70.71	18.97	227000	129210	97790
19	252	252	0	100.00	0.00	4786	4559	227	18.29	87.98	69.70	20.08	227000	136770	90230
20	252	252	0	100.00	0.00	5038	4811	227	19.30	87.98	68.69	21.19	227000	144330	82670
21	252	251	1	99.60	0.40	5290	5062	228	20.30	88.37	68.07	22.20	228000	151860	76140
22	252	251	1	99.60	0.40	5542	5313	229	21.31	88.76	67.45	23.20	229000	159390	69610
23	252	251	1	99.60	0.40	5794	5564	230	22.32	89.15	66.83	24.19	230000	166920	63080
24	252	252	0	100.00	0.00	6046	5816	230	23.33	89.15	65.82	25.29	230000	174480	55520
25	252	252	0	100.00	0.00	6298	6068	230	24.34	89.15	64.81	26.38	230000	182040	47960
26	252	250	2	99.21	0.79	6550	6318	232	25.34	89.92	64.58	27.23	232000	189540	42460
27	252	252	0	100.00	0.00	6802	6570	232	26.35	89.92	63.57	28.32	232000	197100	34900
28	251	251	0	100.00	0.00	7053	6821	232	27.36	89.92	62.57	29.40	232000	204630	27370
29	252	252	0	100.00	0.00	7305	7073	232	28.37	89.92	61.55	30.49	232000	212190	19810
30	252	252	0	100.00	0.00	7557	7325	232	29.38	89.92	60.54	31.57	232000	219750	12250
31	252	252	0	100.00	0.00	7809	7577	232	30.39	89.92	59.53	32.66	232000	227310	4690
32	252	251	1	99.60	0.40	8061	7828	233	31.40	90.31	58.91	33.60	233000	234840	-1840
33	252	252	0	100.00	0.00	8313	8080	233	32.41	90.31	57.90	34.68	233000	242400	-9400
34	252	250	2	99.21	0.79	8565	8330	235	33.41	91.09	57.68	35.45	235000	249900	-14900
35	252	252	0	100.00	0.00	8817	8582	235	34.42	91.09	56.67	36.52	235000	257460	-22460

Out-of-Time Summary Table :

OOT	# Bins	# Records	# Goods	# Bads	Fraud Rate							
	100	12427	12248	179.00	0.01							
	Bin Statistics					Cumulative Statistics						
Population Bin	# Records	# Goods	# Bads	% Goods	% Bads	Total # Records	Cumulative Goods	Cumulative Bads	Cumulative Goods %	Bads (FDR) %	KS	FPR
0	0	0	0	0.00	0.00	0	0	0	0.00	0.00	0.00	0.00
1	124	64	60	51.61	48.39	124	64	60	0.52	33.52	33.00	1.07
2	125	90	35	72.00	28.00	249	154	95	1.26	53.07	51.82	1.62
3	124	117	7	94.35	5.65	373	271	102	2.21	56.98	54.77	2.66
4	124	123	1	99.19	0.81	497	394	103	3.22	57.54	54.33	3.83
5	124	122	2	98.39	1.61	621	516	105	4.21	58.66	54.45	4.91
6	125	125	0	100.00	0.00	746	641	105	5.23	58.66	53.43	6.10
7	124	124	0	100.00	0.00	870	765	105	6.25	58.66	52.41	7.29
8	124	124	0	100.00	0.00	994	889	105	7.26	58.66	51.40	8.47
9	124	119	5	95.97	4.03	1118	1008	110	8.23	61.45	53.22	9.16
10	125	125	0	100.00	0.00	1243	1133	110	9.25	61.45	52.20	10.30
11	124	121	3	97.58	2.42	1367	1254	113	10.24	63.13	52.89	11.10
12	124	124	0	100.00	0.00	1491	1378	113	11.25	63.13	51.88	12.19
13	125	120	5	96.00	4.00	1616	1498	118	12.23	65.92	53.69	12.69
14	124	124	0	100.00	0.00	1740	1622	118	13.24	65.92	52.68	13.75
15	124	123	1	99.19	0.81	1864	1745	119	14.25	66.48	52.23	14.66
16	124	122	2	98.39	1.61	1988	1867	121	15.24	67.60	52.35	15.43
17	125	122	3	97.60	2.40	2113	1989	124	16.24	69.27	53.03	16.04
18	124	124	0	100.00	0.00	2237	2113	124	17.25	69.27	52.02	17.04
19	124	122	2	98.39	1.61	2361	2235	126	18.25	70.39	52.14	17.74
20	124	124	0	100.00	0.00	2485	2359	126	19.26	70.39	51.13	18.72
21	125	124	1	99.20	0.80	2610	2483	127	20.27	70.95	50.68	19.55
22	124	124	0	100.00	0.00	2734	2607	127	21.29	70.95	49.66	20.53
23	124	124	0	100.00	0.00	2858	2731	127	22.30	70.95	48.65	21.50
24	124	124	0	100.00	0.00	2982	2855	127	23.31	70.95	47.64	22.48
25	125	125	0	100.00	0.00	3107	2980	127	24.33	70.95	46.62	23.46
26	124	124	0	100.00	0.00	3231	3104	127	25.34	70.95	45.61	24.44
27	124	124	0	100.00	0.00	3355	3228	127	26.36	70.95	44.59	25.42
28	125	125	0	100.00	0.00	3480	3353	127	27.38	70.95	43.57	26.40
29	124	122	2	98.39	1.61	3604	3475	129	28.37	72.07	43.70	26.94
30	124	123	1	99.19	0.81	3728	3598	130	29.38	72.63	43.25	27.68
31	124	123	1	99.19	0.81	3852	3721	131	30.38	73.18	42.80	28.40
32	125	125	0	100.00	0.00	3977	3846	131	31.40	73.18	41.78	29.36

8. Financial Curves

Below is a graph that plots the overall savings in dollar amount as we increase the Fraud Detection Rate (FDR) cut-off percentage. The green line represents the absolute savings assuming a **$400 gain** for every fraud that's caught. The red line represents the false positive impact assuming a **$20 loss** for every good transaction that's flagged as bad. The **overall savings** calculated is shown in blue. We project this 2 months' data to a year by using a scaling constant, to obtain the plot below.



As we see from the plot above, the overall savings spikes at about 3-4% FDR. This is represented by the straight black line. We take the lower **FDR cut-off at 3%** to try to minimize loss as much as possible while catching as many fraudulent transactions as possible.

9. Conclusion

The entire process used to perform this analysis can be summarized as follows :

- **Exploratory Data Analysis** - This is covered in the Data Quality Report (DQR) attached in the appendix of this report. This entails a preliminary analysis of the given dataset to understand the data by getting summary statistics of all the fields, visualizing each field to observe trends and correlations, performing a data sanity check, etc.

- **Data Preprocessing** - This step involves cleaning the data by excluding irrelevant columns and records, treating missing field values using suitable imputation logic, followed by any additional cleaning that may be required.
- **Variable Creation** - This involves creating as many new variables as make sense to ensure that we capture all latent relationships in the most optimal manner. This led to creation of 1456 variables from 10 given fields.
- **Feature Selection** - This step involves narrowing down the variable set from the previous step to a usable set of expert features since with each added variable we increase the compute intensity of the model. We used filters followed by wrappers to select the top 20 features.
- **Model Exploration** - After getting the final set of features we try different models on them starting with a simple linear model to establish a baseline. After this we try nonlinear models, gradually increasing the complexity and finding the point where the models start to overfit so we can stop there.
- **Model Selection & Performance Evaluation** - We select and fix the model that performs the best after multiple iterations of hyperparameter tuning.
- **Business Reconciliation** - As the final step, we reconcile the model predictions with the business goal of detecting as many frauds as possible while minimizing the loss incurred by the client. In order to do this we plot the overall savings against the FDR % cut-off to decide the optimal threshold.

Final model performance :

The final model chosen after our analysis was the Decision Tree Classifier with the following hyperparameters :

```
DecisionTreeClassifier( criterion='gini',  splitter='best',  max_depth=None,
                        min_samples_leaf=100,  min_samples_split=50 )
```

The average model accuracy over multiple runs was as follows :

- Training data : ~76%
- Testing data : ~73%
- Out of time : ~56%

The final FDR cut-off suggested here is at 3%. This would result in potential savings of up to $21,228,000 based on current transaction records.

10. Appendix : **DQR**

DATA QUALITY REPORT

1. Data Description

The dataset is **Card Transactions Data**, which contains **business transaction** information of credit cards for an organization. The data consists of real card transactions coming from a U.S. government organization. The data spans over the time period of a year, from **1st January, 2010** to **31st December, 2010**. There are **10 fields** and **96753 records**.

a. Summary Tables

Numeric Fields Table

Field Name	# Records With Values	% Populated	# Zeros	Min	Max	Mean	Most Common	Stdev
Date	96753	100.00%	0	2010-01-01	2010-12-31	2010-06-25	2010-02-28	98 days
Amount	96753	100.00%	0	0.01	3102045.53	427.89	3.62	10006.14

Categorical Fields Tables

Field Name	# Records With Values	% Populated	# Zeros	# Unique Values	Most Common
Recnum	96753	100.00%	0	96753	1
Cardnum	96753	100.00%	0	1645	5142148452
Merchnum	93378	96.50%	0	13091	930090121224
Merch description	96753	100.00%	0	13126	GSA-FSS-ADV
Merch state	95558	98.80%	0	227	TN
Merch zip	92097	95.20%	0	4567	38118
Transtype	96753	100.00%	0	4	P
Fraud	96753	100.00%	95694	2	0

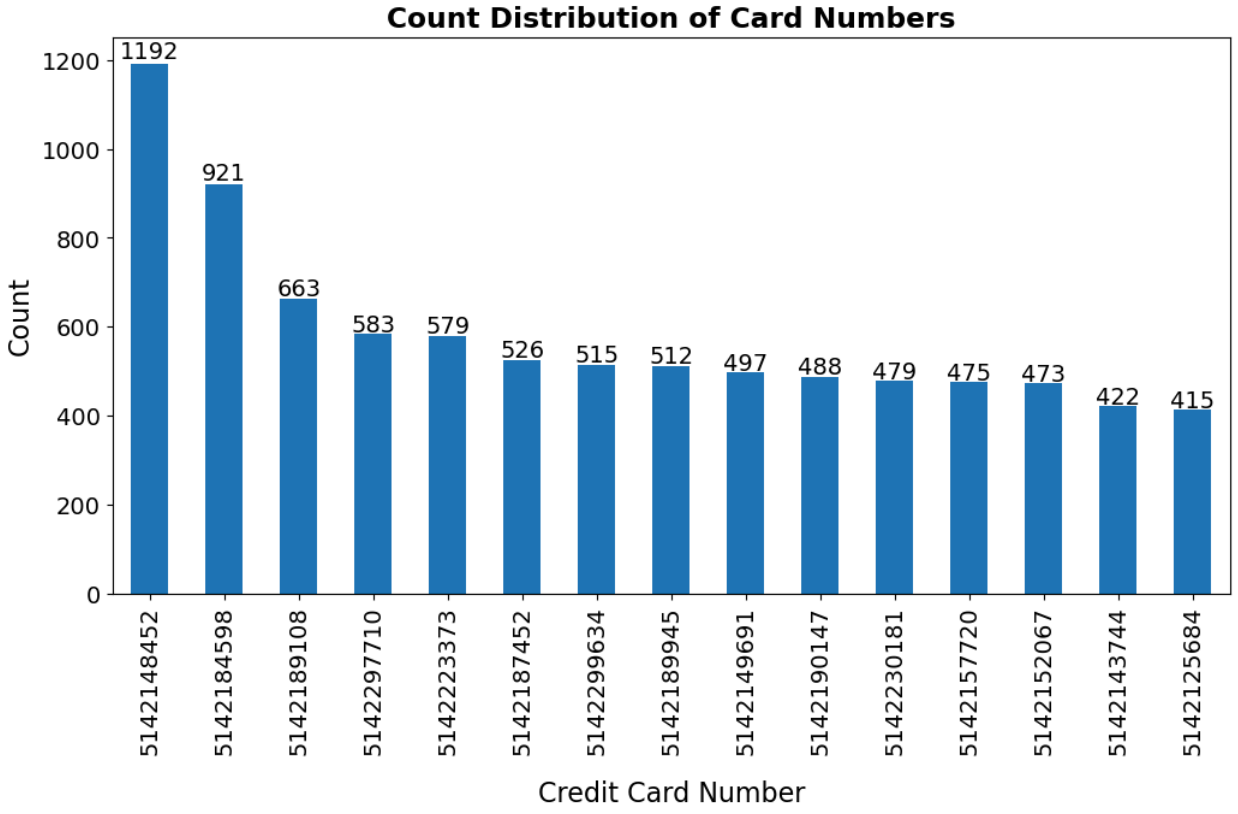
2. Visualization of Each Field

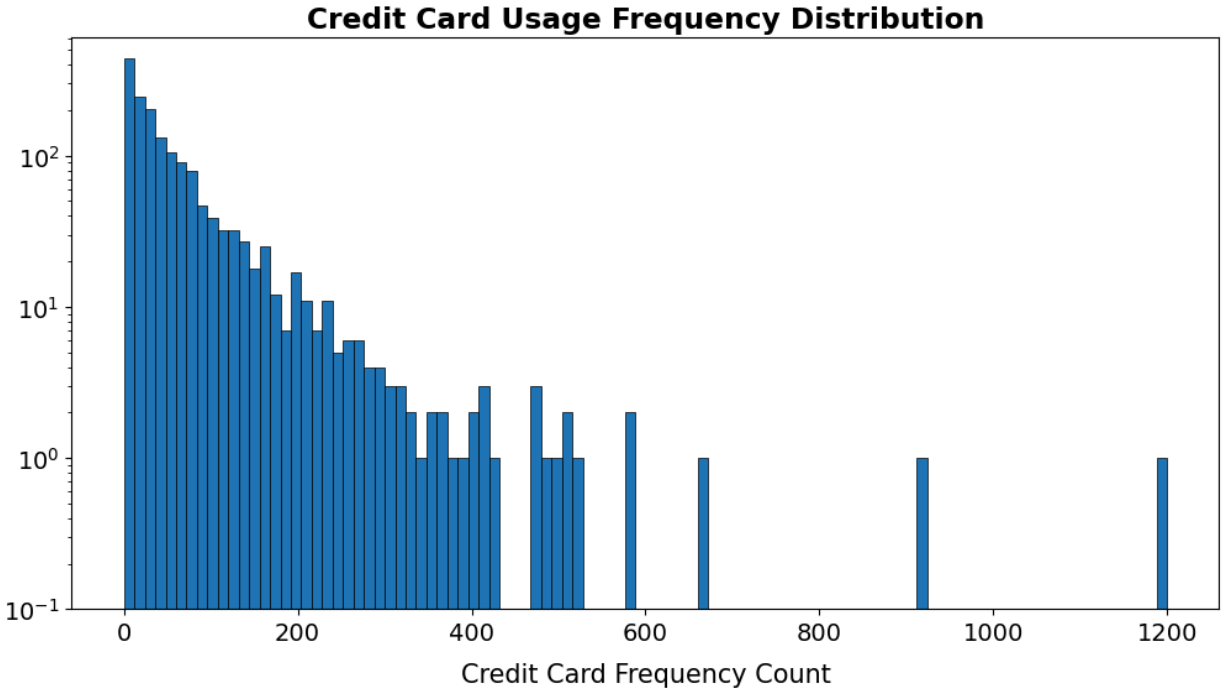
a. Field Name : **Recnum**

Description : Ordinal unique positive integer for each credit card transaction record, from 1 to 96753.

b. Field Name : **Cardnum**

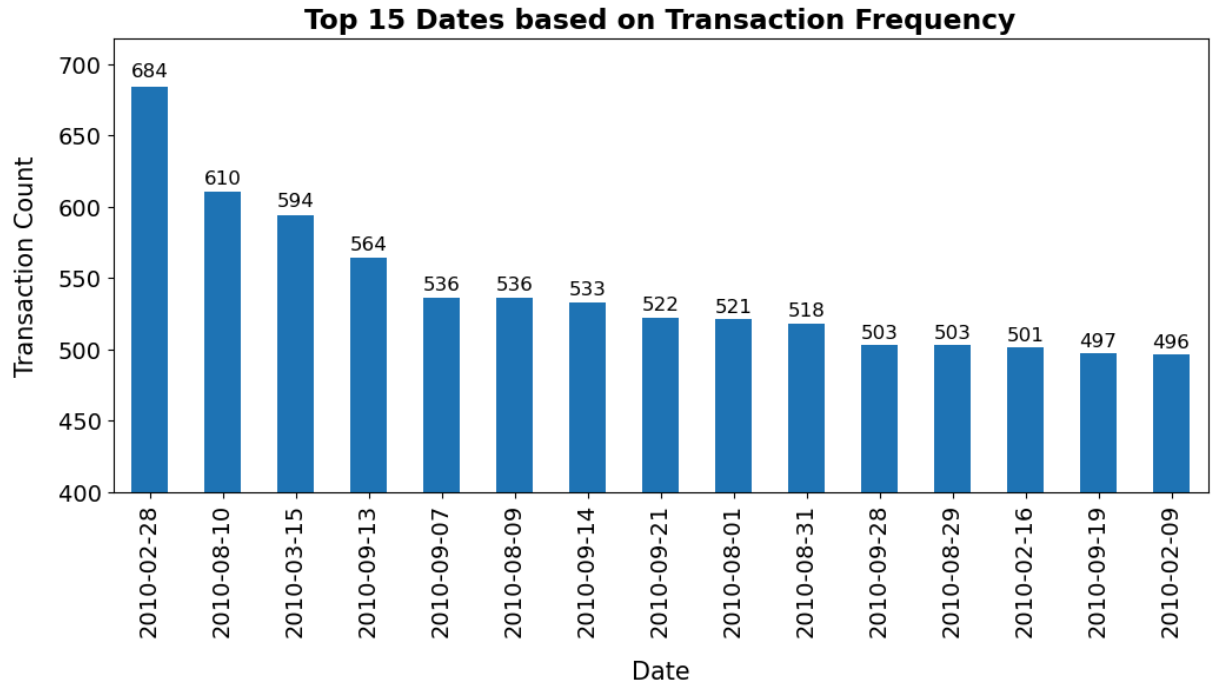
Description : Credit card number. This field has 1645 unique values.
The first graph shows the top 15 credit cards with maximum occurrences in the dataset. The credit card with the card number “5142148452” is the most common card used, with 1192 total transactions.
Observing the frequency distribution graph (second figure) we can see that only 1 card has been used more than 1000 times (1192 times, to be precise). Majority of the credit cards have been used 1 to 200 times only.

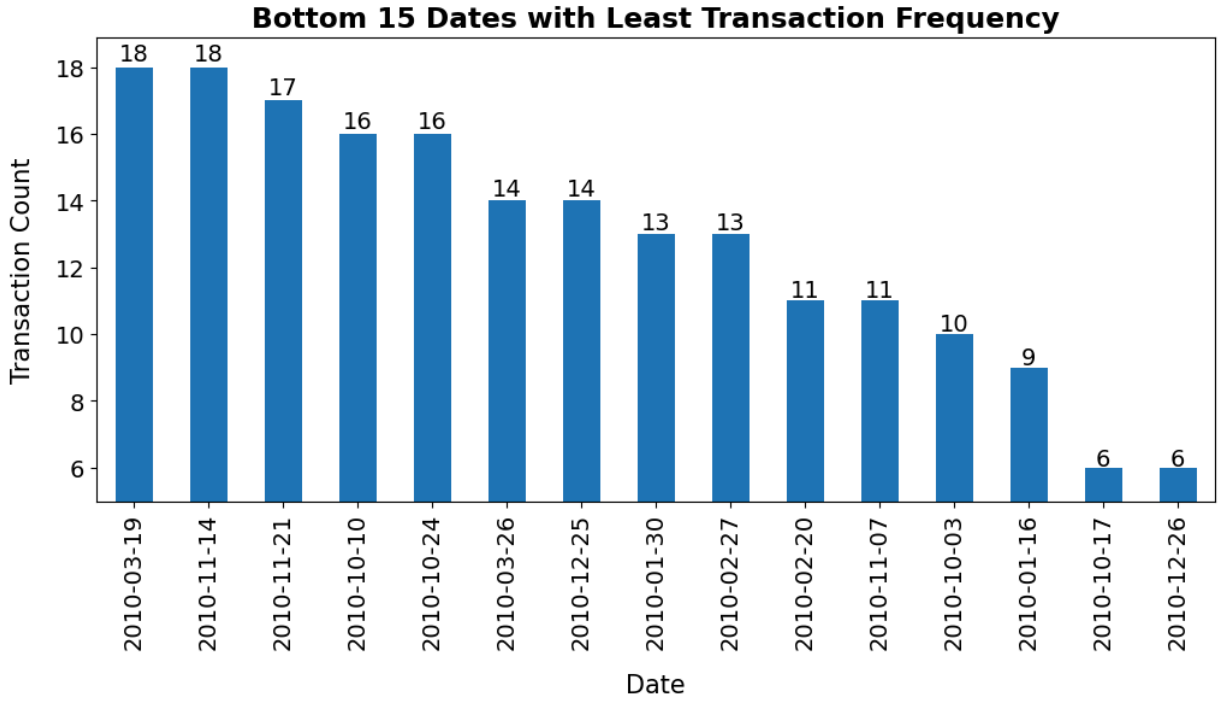


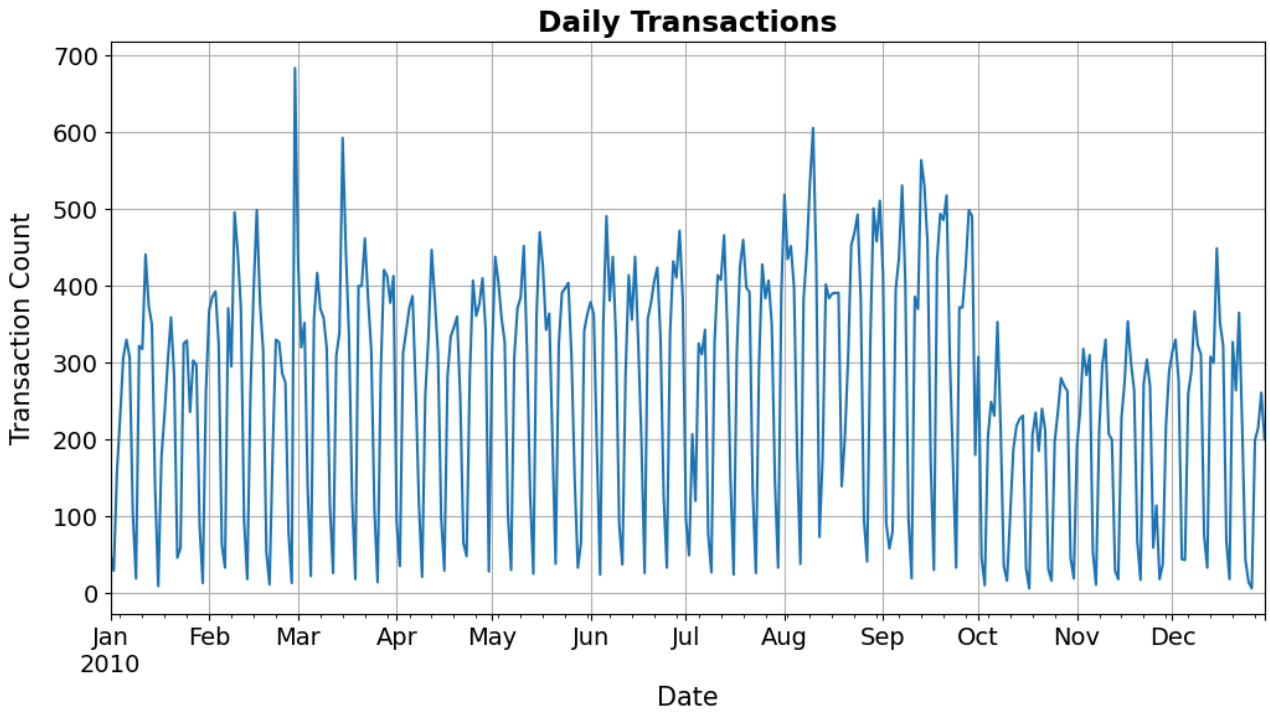


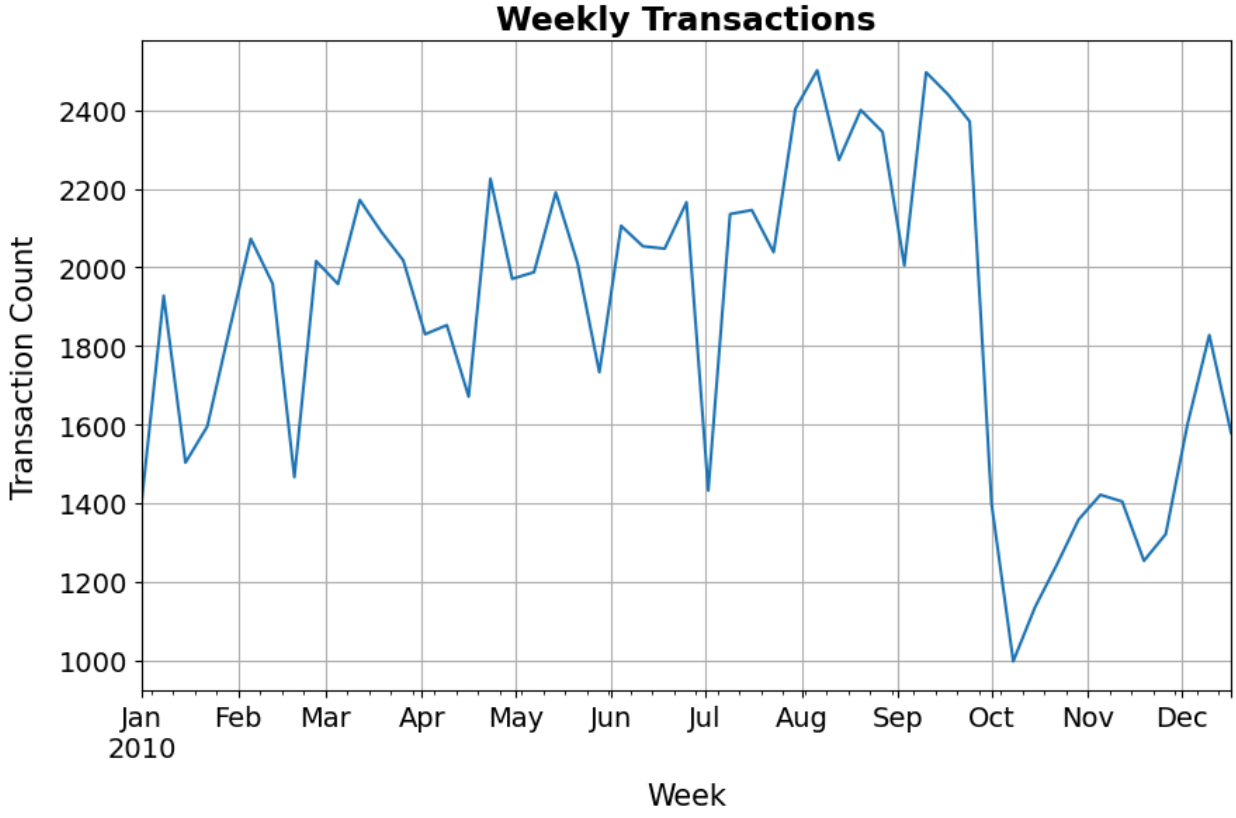
c. Field Name : **Date**

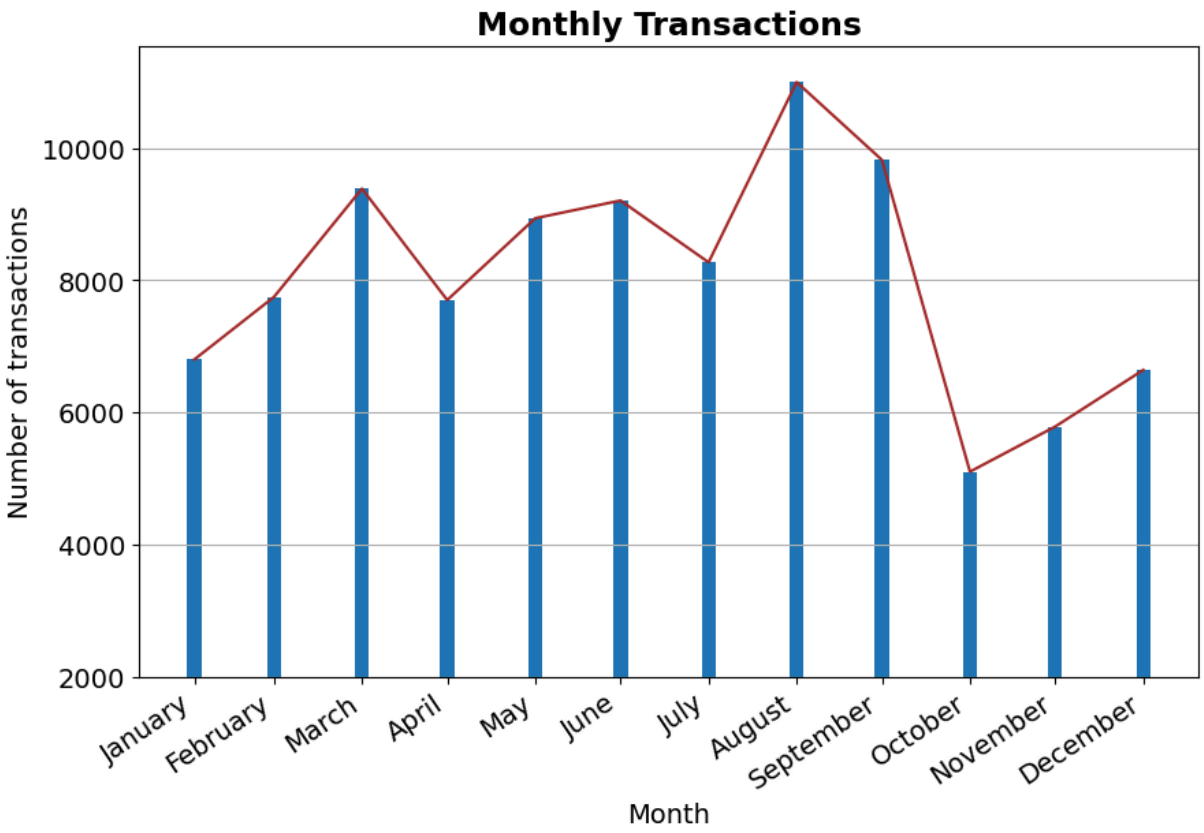
Description : Transaction date. This field has 365 unique values. We have exactly a year's worth of data, for the year 2010 (Jan 1st to Dec 31st). Based on the two Transaction Frequency graphs, we can see that the minimum number of transactions on any given date were 6, and the maximum number of transactions made on a single day were 684 (on 28th February, 2010).
The remaining three plots show the daily, weekly and monthly transaction frequency distributions over time. We can see that the month of August has recorded the highest number of transactions, in 2010.


Top 15 Dates based on Transaction Frequency
Transaction Count
Date
700
650
600
550
500
450
400
684
610
594
564
536
536
533
522
521
518
503
503
501
497
496
2010-02-28
2010-08-10
2010-03-15
2010-09-13
2010-09-07
2010-08-09
2010-09-14
2010-09-21
2010-08-01
2010-08-31
2010-09-28
2010-08-29
2010-02-16
2010-09-19
2010-02-09


Bottom 15 Dates with Least Transaction Frequency
Transaction Count
Date
18
16
14
12
10
8
6
18
18
17
16
16
14
14
13
13
11
11
10
9
6
6
2010-03-19
2010-11-14
2010-11-21
2010-10-10
2010-10-24
2010-03-26
2010-12-25
2010-01-30
2010-02-27
2010-02-20
2010-11-07
2010-10-03
2010-01-16
2010-10-17
2010-12-26

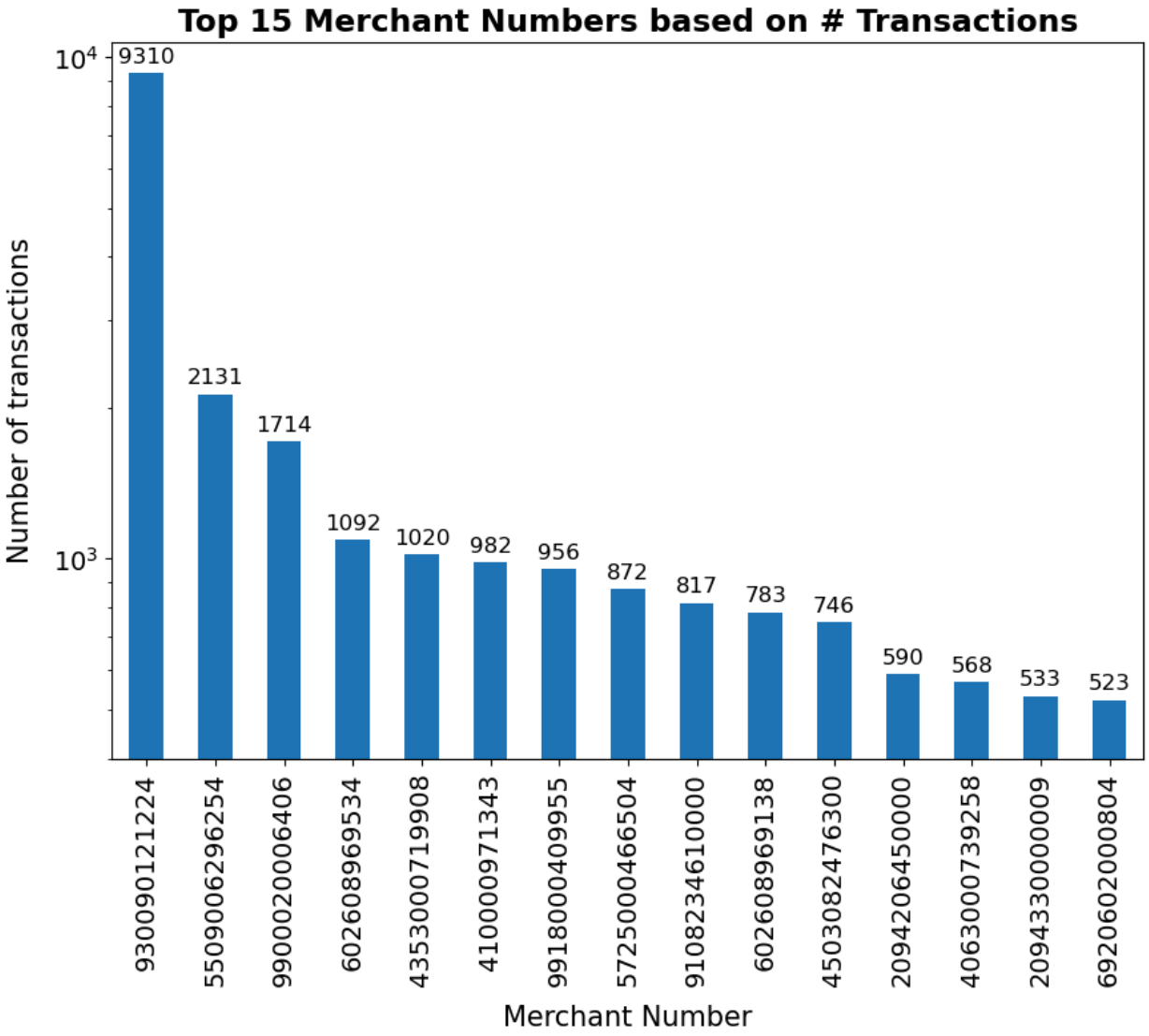

Daily Transactions
Transaction Count
700
600
500
400
300
200
100
0
Jan
2010
Feb
Mar
Apr
May
Jun
Jul
Aug
Sep
Oct
Nov
Dec
Date


Weekly Transactions
Transaction Count
2400
2200
2000
1800
1600
1400
1200
1000
Jan
2010
Feb
Mar
Apr
May
Jun
Jul
Aug
Sep
Oct
Nov
Dec
Week



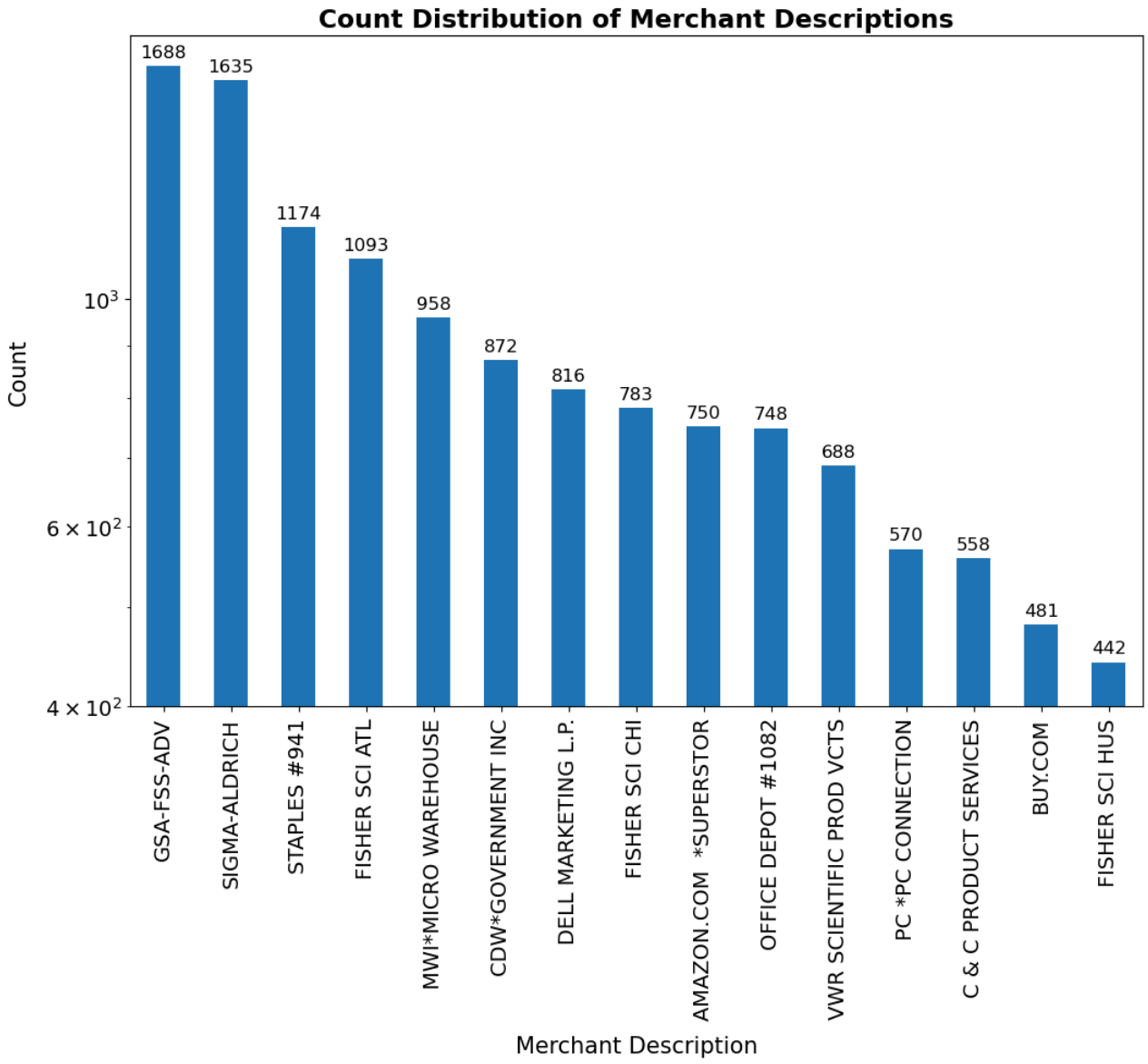
d. Field Name : **Merchnum**

Description : Merchant Number. This is a categorical field with 13092 distinct values. The most common merchant number is “930090121224”, with a total transaction count of 9310.



e. Field Name : **Merch description**

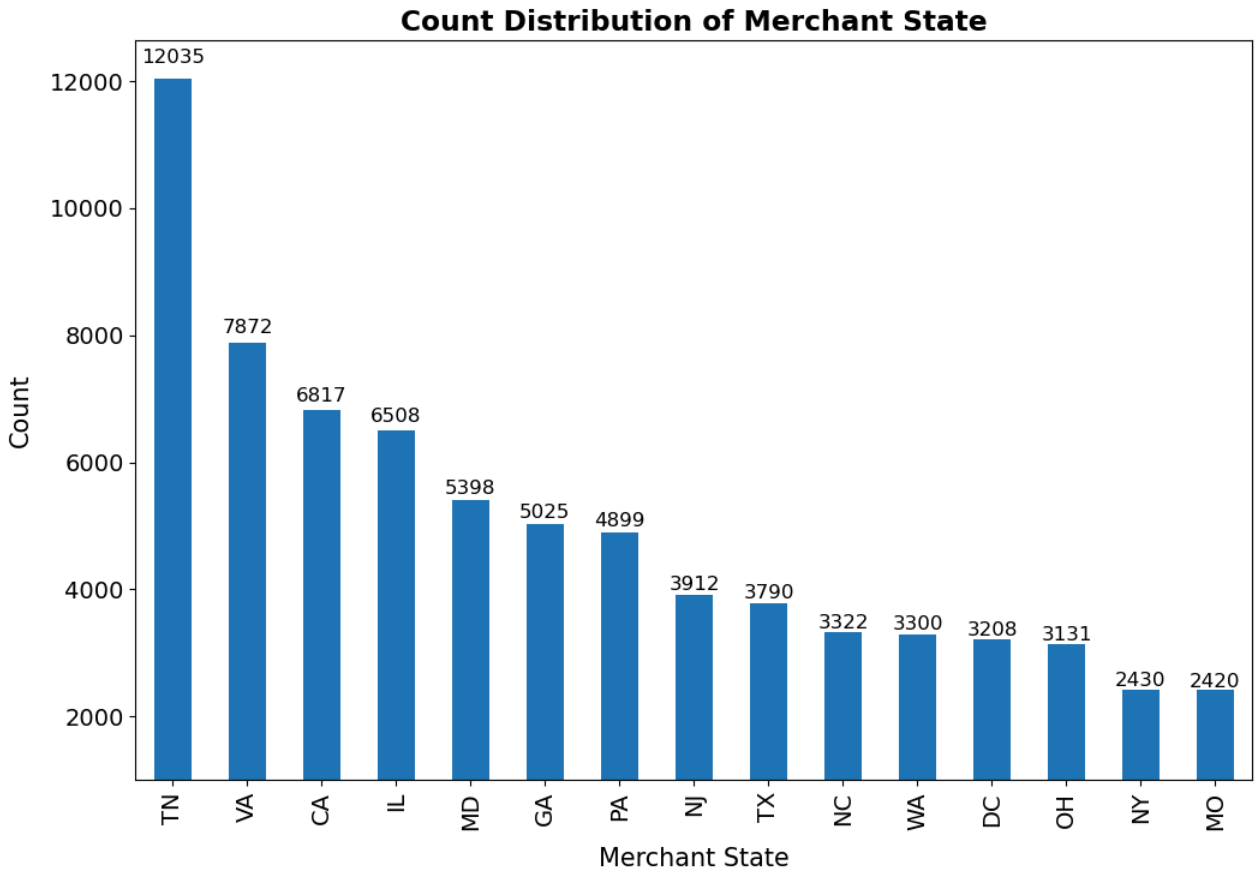
Description : Merchant Description. This a text field, giving a brief description of the merchant. This field has 13126 distinct values. The most common merchant description is “GSA-FSS-ADV” with a total count of 1688.



f. Field Name : **Merch state**

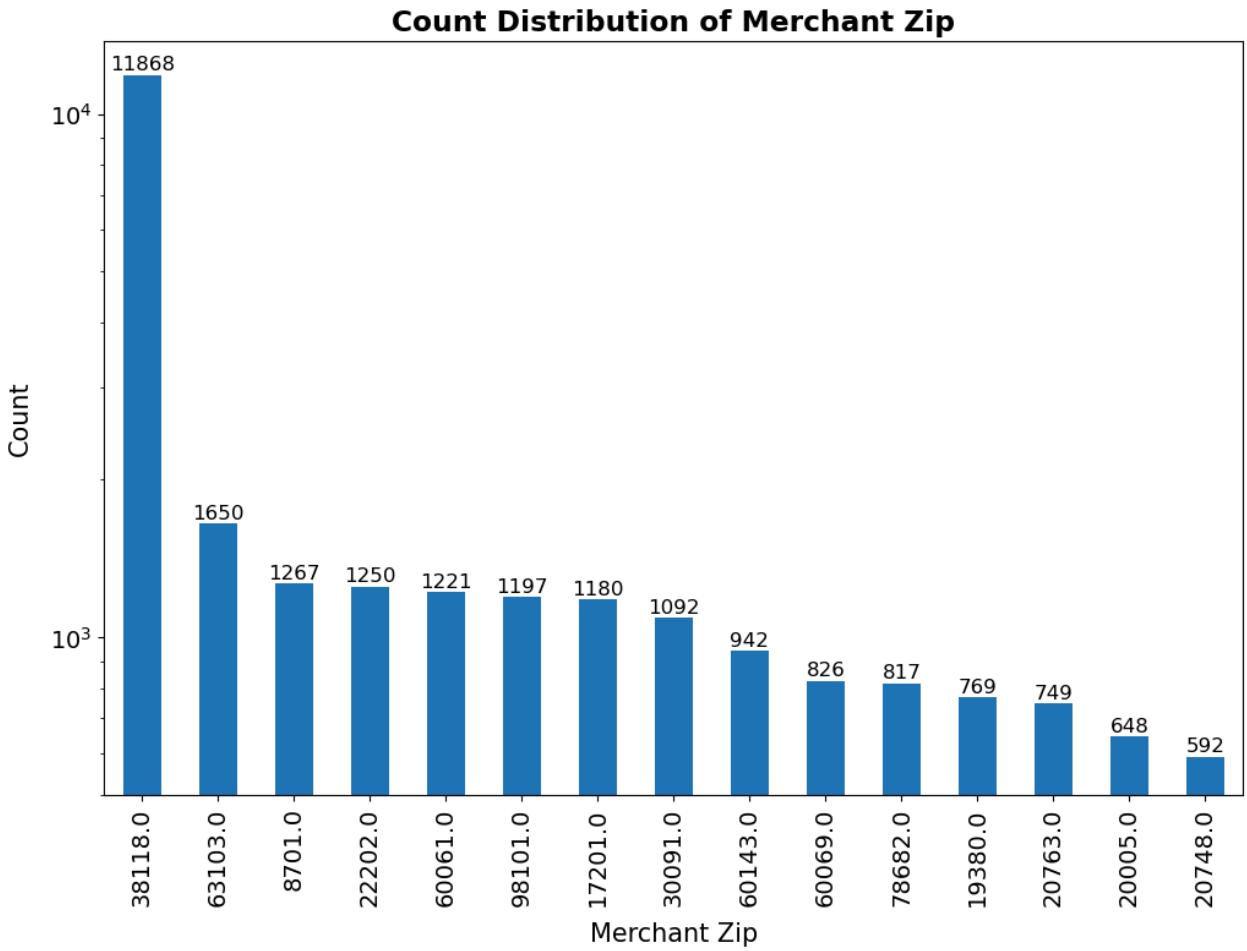
Description : Merchant State. This is a categorical field that contains the two letter state code of the state to which the merchant belongs. This field has 228 unique values.

The figure below shows the top 15 merchant states based on the number of occurrences. The most common merchant state is “TN” with a total count of 12035 transactions.



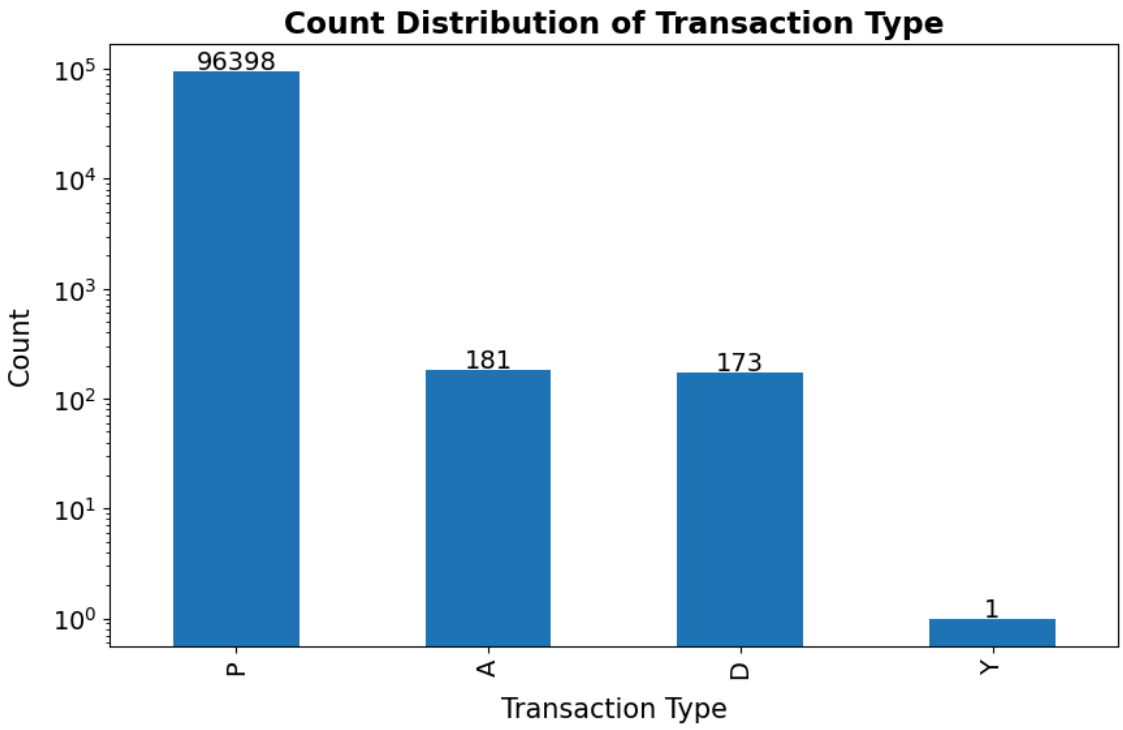
g. Field Name : **Merch zip**

Description : Merchant zip code. This is a categorical field consisting of the zip code values of the merchant. This field has 4568 distinct values.
The most common Merchant Zip is “38118” with 11868 total occurrences.



h. Field Name : **Transtype**

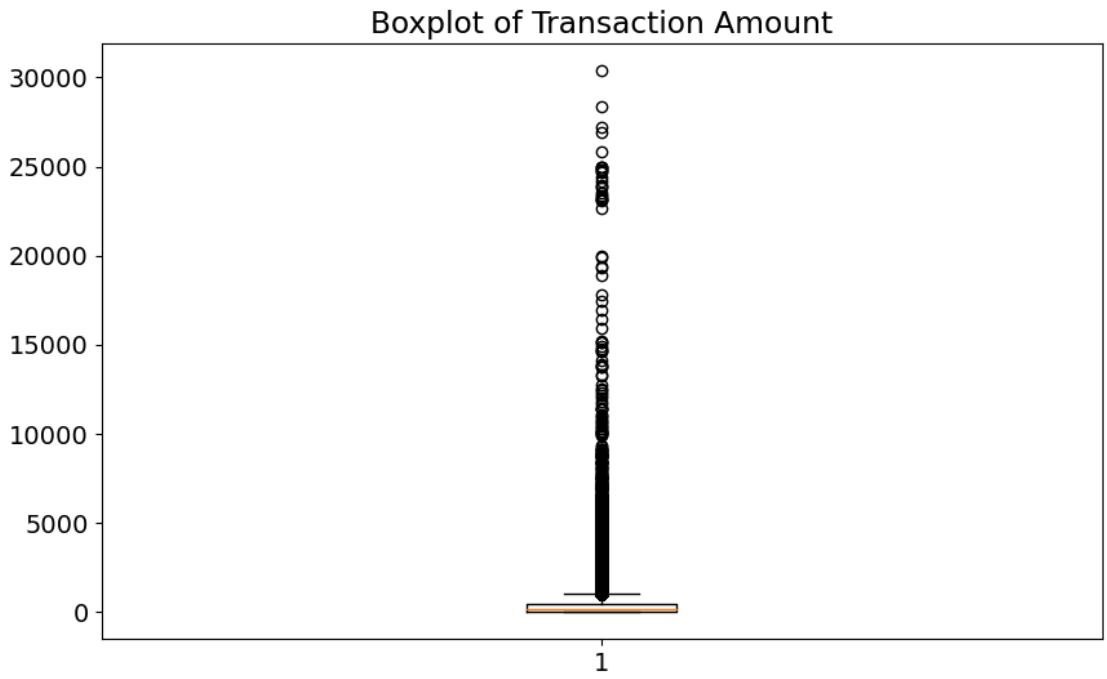
Description : Transaction type. This is a categorical field classifying the type of transaction. This field has 3 distinct values which are - P, A, D, and Y. The most common transaction type is “P” (purchase) with a total of 96398 occurrences out of a total of 96753 records.

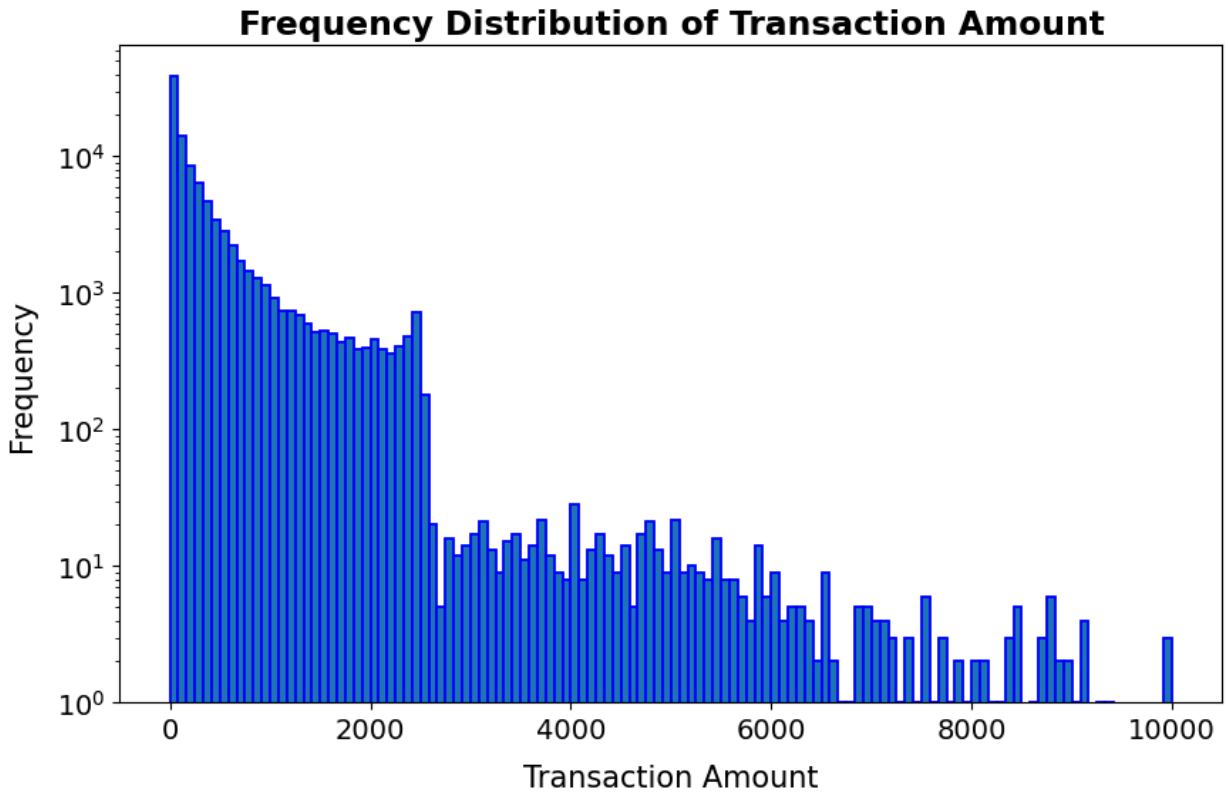


i. Field Name : **Amount**

Description : This numerical field contains the value of the transaction amount. The amount value ranges from 0.01 to 3102045.53 with a mean value of 427.89 and mode value of 3.62.
Examination of this data shows that the max value (3102045.53) is an outlier. Removing this outlier value, we get the first figure, a boxplot, showing that the majority of the transactions fall within the range of 0.01 to 10,000.

Therefore, we plot the second graph, which gives us the distribution of the transaction amount, keeping the amount limit as 10,000. This plot shows us that the bulk of the transactions are of an amount ranging from 0.01 to 2700.


Boxplot of Transaction Amount
30000
25000
20000
15000
10000
5000
0
1


Frequency Distribution of Transaction Amount
Frequency
10^4
10^3
10^2
10^1
10^0
0
2000
4000
6000
8000
10000
Transaction Amount

j. Field Name : **Fraud**

Description : This is a binary categorical field that specifies the label of the records, classifying it as fraud if the value = 1, and not a fraud transaction if value = 0. Since it's a binary field, it only takes 2 values : 0 and 1.
The total count of fraudulent transactions = 1,059.
The total count of non-fraudulent transactions = 95,694.