Summary of Credit Card Transaction

Executive summary

The overall goal of the project is to find a machine learning model to detect the fraud detection of credit card transactions with high accuracy to predict the fraud rate in the future. This project includes data description, data cleaning and imputation, variable creation, feature selection, model exploration, model performance analysis and business insights based on business settings. Overall, the final model can catch 53% of the frauds on the top 3% of records, and the anticipated savings for the business is $19,332,000/year.

Description of the data

The data is a collection of real card transaction records from a U.S. government organization. The data is synthetic since we modify the data by adding some fraud cases to make the data fit the course purpose. The data covers time from Jan. 2010 to Dec. 2010. There are 10 fields and 96,753 records.

1)Numeric Table

Field Name	%Populated	Min	Max	Means	St dev	%Zero
Date	100.00	2017-01-01	2017-12-31	NA	NA	0.00
DOB	100.00	19000110	20161030	NA	NA	0.00

2)Categorical Table

Field Name	%Populated	#Unique Values	Most Common Value
record	100.00	1,000,000	NA
ssn	100.00	835,819	999999999
firstname	100.00	78,136	EAMSTRMT
lastname	100.00	177,001	ERJSAXA
address	100.00	828,774	123 MAIN ST
zip5	100.00	26,370	68138
homephone	100.00	28,244	9999999999
fraud label	100.00	2	0

Data cleaning

For the **data cleaning**, I converted the type of Data to datetime and the type of Fraud to string. There is an outlier that is larger than 2,500,000, so I dropped this highest amount to remove the

effects on an outlier. I only kept the “purchased” data and dropped the other three types of trans types.

Regarding **data imputation**, for the **merch number**, firstly, I replaced 0 with null values and sum the null values. I mapped the merch description and merchnum and filled out if there are null values in merch num or merch description. Secondly, I used iteration to fulfill the merchdes_merchnum dictionary. The dictionary includes the non-null merch description and its corresponding merch number to make sure there is a unique merch description matching a merchnum value. Only the merchnum that occurs in the merch description for the first time will be stored in the dictionary. Then I used the previous dictionary to fulfill the null value in the merch number. In the case when the merch description is either ‘'RETAIL CREDIT ADJUSTMENT' or ‘RETAIL DEBIT ADJUSTMENT’, I replaced the value in merch number with ‘unknown’. Furthermore, I also added new merchnum and the new merchnumber will be max(merchnum) +1, and I filled null value of merch number by new merch number.

For the **merch state**, I checked the null value in merch zip, paired specific zip codes to corresponding states and ensure that there is a unique merch zip pairing with a merch state when I create the mapping dictionary. Then I checked the unique merch description with merch state, and merchnum with merch state and fill any missing state values in merchnum or merch description using the previous dictionary (the logic is similar to the imputation of the merch num). Furthermore, I assigned an unknown value for the adjustment transaction in case the merch state is unpracticable. If the state is not in the U.S., I assigned the merch states to foreign.

Similar to the merch state, I filled in values and assigned unknown values to merch zip.

Variable creation

Description of Variables	# Variables Created
Original fields from dataset including ‘Recnum’ and ‘Fraud’	10
Zip3: first three digits of zipcode	1
Variable of Benford’s Law:	2
Date of week target encoded	1
Amount Variables: (Average/Maximum/Median/Total/(Actual/average)/ (Actual/maximum)/(Actual/median)/(Actual/total) amount over the past (0,1,3,7,14,30,60) days	984
Velocity Change Variables: this set of variables includes (entities across fields such as [# amount] with same [card merchant] over the last 0/1 days) / (Average entities [daily # amount of transactions] with same entities like [card merchant]over the past (7,14,30,60) days	135
Acceleration Variables: this set of variables includes (entities across fields such as [# amount] with same [card merchant] over the last 0/1 days) / Power of entities [daily # amount of transactions] with same entities like [card merchant]over the past (7,14,30,60) days. This set of variables is more related to velocity change variables like the change rate of velocity change variables.	135

Variability: this set of variables includes the avg/max/median of variability of entities across fields over the past [0,1,3,7,14,30,60] days	362
Total number of variables after dropping categorical variables	1622
The total number of variables after deduplicaiton	1424

Feature selection

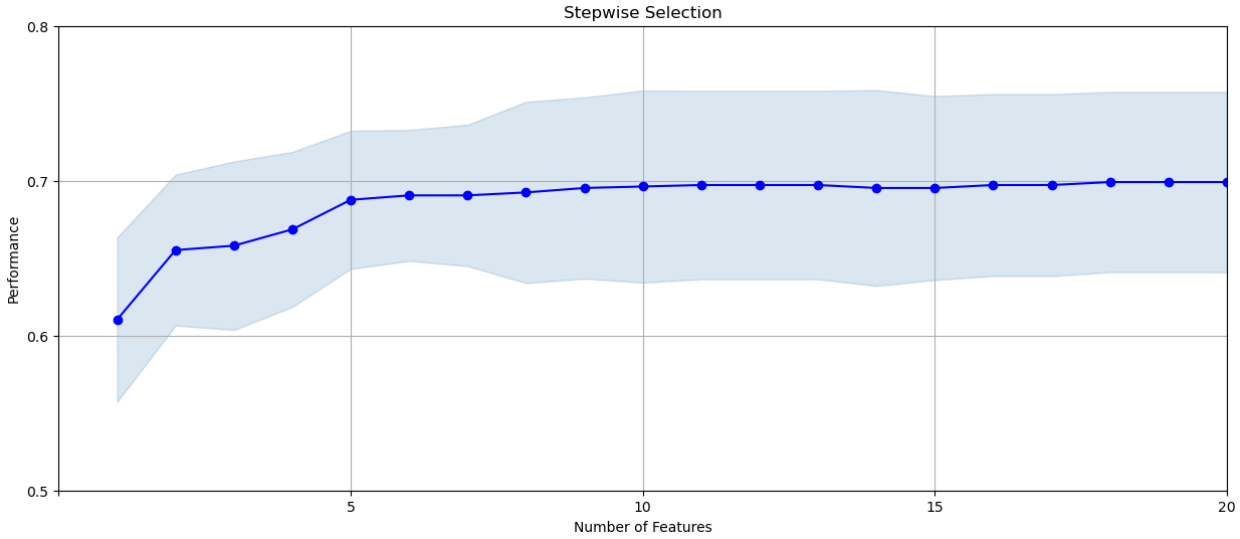
Since the performance of nonlinear model is not good in high dimensions; for instance, in high dimensions, the number of parameters in a nonlinear model can quickly exceed the number of training examples, leading to overfitting. To reduce the number of features and keep only the necessary, I've tried 8 nonlinear models, which are:

- Forward Selection: LGBMClassifier(n_estimators=20, num_leaves=4) with (filter_num = 100, wrapper_num = 20) , (filter_num = 200, wrapper_num = 20), (filter_num = 300, wrapper_num = 20)
- Forward Selection: Random Forest (n_estimators =5) with (filter_num = 100, wrapper_num = 10), (filter_num = 200, wrapper_num = 20), (filter_num =300, wrapper_num = 20)
- Backward Selection: LGBMClassifier(n_estimators=20, num_leaves=4) with (filter_num = 150, wrapper_num = 20)
- Random Forest (n_estimator = 5) with (filter_num = 100, wrapper_num = 20)

The chose model is LGBMClassifier (n_estimators=20,num_leaves=4) with filter number 300 and wrapper number 20, since it has the highest filter score of around 0.70 with relatively low running time. The table shows the list of 20 selected variables and the graph shows the model performance with the increasing number of features.

Order	VARIABLE	FILTER SCORE
1	card_zip_total_14	0.652
2	merch_zip_max_30	0.465
3	merch_zip_max_60	0.439
4	Merchnum_desc_total_0	0.505
5	Merchnum_total_30	0.395
6	card_zip_total_30	0.637
7	Merchnum_desc_variability_med_30	0.356
8	merch_zip_actual/med_60	0.373
9	zip3_avg_0	0.471
10	zip3_actual/max_30	0.394
11	Merchnum_total_14	0.444
12	amount_cat	0.477
13	zip3_actual/max_60	0.406
14	merch_zip_total_14	0.440
15	Merchnum_desc_total_1	0.529

16	Merchnum_desc_total_14	0.491
17	Merchnum_desc_total_3	0.538
18	Merchnum_max_60	0.438
19	Merchnum_desc_total_7	0.517
20	zip3_total_0	0.468



Preliminary model explore

Based on 20 variables with top KS score, I chose logistic regression model as baseline model and other our four nonlinear models including Single Decision Tree, Random Forest, LightGBM, Neural Networks to compare the training, testing, and out of time scores. I set the number of variables as 10, FDR as 3%, adjusted the parameters, ran each trial 10 times and averaged the

Model	Parameter									Average FDR at 3%		
Logistic Regression	iteration	# variables	Penalty	C	solver	l1_ratio				Train	Test	OOT
	1	10	l2	1	lbfgs	none				0.622	0.612	0.343
	2	10	l2	0.1	lbfgs	none				0.610	0.609	0.331
	3	10	l2	1	saga	none				0.620	0.627	0.343
Decision Tree	Iteration	#variables	criterion	max_depth	min_samples_leaf	min_sample_split	splitter			train	test	oot
	1	10	gini	3	5	10	best			0.6310	0.604	0.473
	2	10	gini	3	15	30	best			0.628	0.601	0.420
	3	10	gini	3	20	30	best			0.628	0.606	0.511
	4	10	gini	4	20	30	best			0.641	0.628	0.478
	5	10	entropy	4	15	25	best			0.651	0.620	0.444
	6	10	gini	8	50	100	best			0.815	0.739	0.455
Randm Forest	Iteration	# variables	# estimators	criterion	max_depth	min_sample_split	min_sample_leaf			train	test	oot
	1	10	10	gini	5	10	5			0.755	0.724	0.519
	2	10	20	gini	5	50	30			0.762	0.744	0.489
	3	10	50	gini	5	100	50			0.771	0.737	0.502
	4	10	20	gini	4	60	30			0.728	0.704	0.516
	5	10	20	entropy	4	50	30			0.770	0.740	0.527
	6	10	50	entropy	4	50	30			0.774	0.761	0.530
LightGBM	iteration	# variables	num_leaves	max_depth	learning rate	num_estimators				train	test	oot
	1	10	5	5	0.01	10				0.663	0.658	0.518
	2	10	10	5	0.01	20				0.772	0.741	0.527
	3	10	10	8	0.01	50				0.798	0.766	0.536
	4	10	50	3	0.001	50				0.6910	0.674	0.513
	5	10	30	3	0.001	50				0.6920	0.6670	0.511
	6	10	40	3	0.001	100				0.706	0.697	0.513
Neural Network	Iteration	# variables	hidden_layer_sizes	activation	alpha	learning_rate	learning_rate_init	max_iter	solver	train	test	oot
	1	10	(5,)	relu	0.1	constant	0.001	100	adam	0.644	0.631	0.373
	2	10	(5,5,20)	relu	0.1	constant	0.001	200	adam	0.676	0.675	0.455
	3	10	(10,10)	relu	0.01	adaptive	0.001	200	adam	0.725	0.696	0.438
	4	10	(5,)	relu	0.1	adaptive	0.001	300	lbfgs	0.686	0.673	0.502
	5	10	8	logistic	0.01	constant	0.001	1000	adam	0.665	0.664	0.482
	6	10	(20,)	relu	0.1	constant	0.01	1000	adam	0.681	0.673	0.485

scores in the table. From the table, models of random forest and lightGBM have better performance scores than other models. Although lightGBM has high oot scores, it also has the problem of overfitting when training is much higher than testing. Therefore, after taking this factor into consideration, I choose random forest (# estimator = 50, max_depth = 4, min_sample_split = 50, min_sample_leaf = 30) as the final model.

Final model performance

- Random forest (# estimator = 50, max_depth = 4, min_sample_split = 50, min_sample_leaf = 30).

The following three tables are training, testing, and oot scores for the final model. The model has a train score of 0.774, test score 0.761, and oot 0.530. This indicates that the model can eliminate around 53% frauds by only rejecting 3% records.

Training

Training	#Records	#Goods	#Bads	Fraud Rate								
	59010	58387	623	0.010557533								
	Bin Statistics					Cumulative Statistics						
Populaiton Bins%	#records	#goods	#bads	%goods	%bads	Total Records	Cumulative Goods	Cumulative Bads	%Cumulative Goods	FDR(%Cumulative Bads)	KS	FPR
1	590	270	320	45.763	54.237	590	270	320	0.462	51.364	50.902	0.844
2	590	458	132	77.627	22.373	1180	728	452	1.247	72.552	71.305	1.611
3	590	565	25	95.763	4.237	1770	1293	477	2.215	76.565	74.350	2.711
4	590	568	22	96.271	3.729	2360	1861	499	3.187	80.096	76.909	3.729
5	590	576	14	97.627	2.373	2950	2437	513	4.174	82.343	78.170	4.750
6	591	578	13	97.800	2.200	3541	3015	526	5.164	84.430	79.266	5.732
7	590	579	11	98.136	1.864	4131	3594	537	6.155	86.196	80.040	6.693
8	590	585	5	99.153	0.847	4721	4179	542	7.157	86.998	79.841	7.710
9	590	587	3	99.492	0.508	5311	4766	545	8.163	87.480	79.317	8.745
10	590	582	8	98.644	1.356	5901	5348	553	9.160	88.764	79.604	9.671
11	590	583	7	98.814	1.186	6491	5931	560	10.158	89.888	79.730	10.591
12	590	582	8	98.644	1.356	7081	6513	568	11.155	91.172	80.017	11.467
13	590	587	3	99.492	0.508	7671	7100	571	12.160	91.653	79.493	12.434
14	590	588	2	99.661	0.339	8261	7688	573	13.167	91.974	78.807	13.417
15	591	588	3	99.492	0.508	8852	8276	576	14.174	92.456	78.281	14.368
16	590	589	1	99.831	0.169	9442	8865	577	15.183	92.616	77.433	15.364
17	590	588	2	99.661	0.339	10032	9453	579	16.190	92.937	76.747	16.326
18	590	589	1	99.831	0.169	10622	10042	580	17.199	93.098	75.899	17.314
19	590	586	4	99.322	0.678	11212	10628	584	18.203	93.740	75.537	18.199
20	590	587	3	99.492	0.508	11802	11215	587	19.208	94.222	75.013	19.106

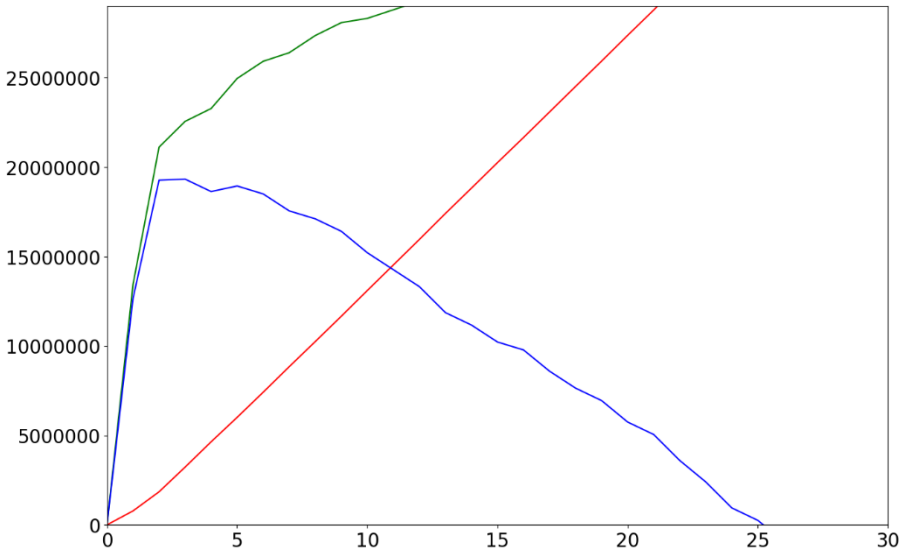
Testing

Testing	#Records	#Goods	#Bads	Fraud Rate								
	25290	25033	257	0.010162119								
	Bin Statistics					Cumulative Statistics						
Populaiton Bins%	#records	#goods	#bads	%goods	%bads	Total Records	Cumulative Goods	Cumulative Bads	%Cumulative Goods	FDR(%Cumulative Bads)	KS	FPR
1	253	118	135	46.640	53.360	253	118	135	0.471	52.529	52.058	0.874
2	253	204	49	80.632	19.368	506	322	184	1.286	71.595	70.309	1.75
3	253	241	12	95.257	4.743	759	563	196	2.249	76.265	74.016	2.872
4	253	243	10	96.047	3.953	1012	806	206	3.220	80.156	76.936	3.913
5	252	247	5	98.016	1.984	1264	1053	211	4.206	82.101	77.895	4.991
6	253	250	3	98.814	1.186	1517	1303	214	5.205	83.268	78.063	6.089
7	253	250	3	98.814	1.186	1770	1553	217	6.204	84.436	78.232	7.157
8	253	252	1	99.605	0.395	2023	1805	218	7.210	84.825	77.614	8.28
9	253	251	2	99.209	0.791	2276	2056	220	8.213	85.603	77.390	9.345
10	253	249	4	98.419	1.581	2529	2305	224	9.208	87.160	77.952	10.29
11	253	253	0	100.000	0.000	2782	2558	224	10.219	87.160	76.941	11.42
12	253	250	3	98.814	1.186	3035	2808	227	11.217	88.327	77.110	12.37
13	253	252	1	99.605	0.395	3288	3060	228	12.224	88.716	76.492	13.421
14	253	250	3	98.814	1.186	3541	3310	231	13.223	89.883	76.661	14.329
15	253	249	4	98.419	1.581	3794	3559	235	14.217	91.440	77.222	15.145
16	252	251	1	99.603	0.397	4046	3810	236	15.220	91.829	76.609	16.144
17	253	251	2	99.209	0.791	4299	4061	238	16.223	92.607	76.384	17.063
18	253	253	0	100.000	0.000	4552	4314	238	17.233	92.607	75.374	18.126
19	253	253	0	100.000	0.000	4805	4567	238	18.244	92.607	74.363	19.189
20	253	253	0	100.000	0.000	5058	4820	238	19.255	92.607	73.352	20.252

OOT

OOT	#Records	#Goods	#Bads	Fraud Rate									
	12097	11918	179	0.014797057									
	Bin Statistics					Cumulative Statistics							
Populaiton Bins%	#records	#goods	#bads	%goods	%bads	Total Records	Cumulative Goods	Cumulative Bads	%Cumulative Goods	FDR(%Cumulative Bads)	KS	FPR	
1	121	65	56	53.719	46.281	121	65	56	0.545	31.285	30.740	1.161	
2	121	89	32	73.554	26.446	242	154	88	1.292	49.162	47.870	1.750	
3	121	115	6	95.041	4.959	363	269	94	2.257	52.514	50.257	2.862	
4	121	118	3	97.521	2.479	484	387	97	3.247	54.190	50.943	3.990	
5	121	114	7	94.215	5.785	605	501	104	4.204	58.101	53.897	4.817	
6	121	117	4	96.694	3.306	726	618	108	5.185	60.335	55.150	5.722	
7	121	119	2	98.347	1.653	847	737	110	6.184	61.453	55.269	6.700	
8	121	117	4	96.694	3.306	968	854	114	7.166	63.687	56.522	7.491	
9	121	118	3	97.521	2.479	1089	972	117	8.156	65.363	57.207	8.308	
10	121	120	1	99.174	0.826	1210	1092	118	9.163	65.922	56.759	9.254	
11	121	119	2	98.347	1.653	1331	1211	120	10.161	67.039	56.878	10.092	
12	121	119	2	98.347	1.653	1452	1330	122	11.160	68.156	56.997	10.902	
13	121	121	0	100.000	0	1573	1451	122	12.175	68.156	55.982	11.893	
14	121	118	3	97.521	2.479	1694	1569	125	13.165	69.832	56.667	12.552	
15	121	119	2	98.347	1.653	1815	1688	127	14.163	70.950	56.786	13.291	
16	121	117	4	96.694	3.306	1936	1805	131	15.145	73.184	58.039	13.779	
17	120	119	1	99.167	0.833	2056	1924	132	16.144	73.743	57.599	14.576	
18	121	119	2	98.347	1.653	2177	2043	134	17.142	74.860	57.718	15.246	
19	121	118	3	97.521	2.479	2298	2161	137	18.132	76.536	58.404	15.774	
20	121	120	1	99.174	0.826	2419	2281	138	19.139	77.095	57.956	16.529	

Financial curves and recommended cutoff



Here I applied the model into a real business setting. Since it is assumed that we can have $400 (green line) in revenue for each caught fraud and $20 loss (blue line) for instances of false positive, the blue line is the overall saving. I decided to choose a cutoff of fraud detection rate at 3%, because at this point, the slope of green line is large which indicates the high rate of fraud to be caught, and at the same time, the blue line is around $19,332,000. Therefore, the estimated annual saving is $19,332,000 using the model.

Summary

In summary, in the project of credit card transaction, I included data description, data cleaning and imputation, variable creation, feature selection to choose 20 final variables, model exploration(including one linear model and four nonlinear models), model performance analysis, recommended cutoff and provided business applications and insights.

The model of RandomForest which has the number of estimator 50, max_depth 4, min_sample_split 50, min_sample_leaf 30, criterion entropy can catch 53% of the total frauds.

Using our model, we can catch 53% of the fraud in the top 3% of records. This can help the business save $19,332,000 in a year.

Appendix – DQR

Data Quality Report

1. Data Description
   The data is a collection of real card transaction records from a U.S. government organization. The data is synthetic since we modify the data by adding some fraud cases to make the data fit the course purpose. The data covers time from **Jan**. **2010** to **Dec**. **2010.** There are **10** fields and **96,753** records.

2. Summary Tables
   1)Numeric Table

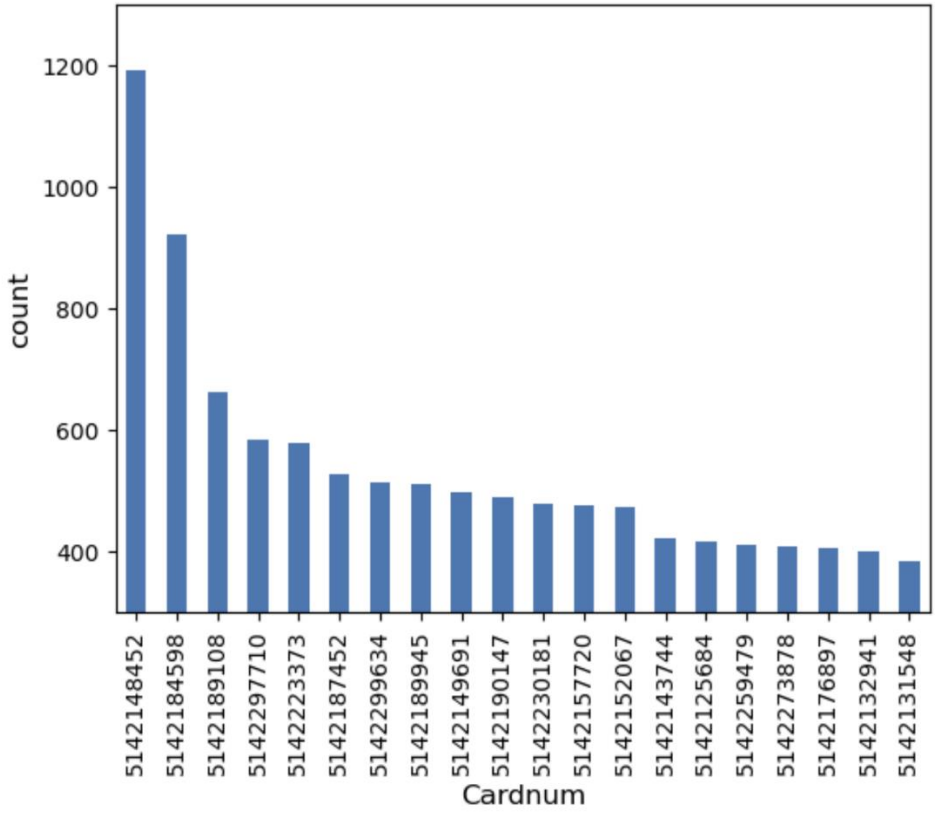
Field Name	%Populated	Min	Max	Means	St dev	%Zero
Date	100.00	2010-01-01	2010-12-31	NA	NA	0.00
Amount	100.00	0.01	3,102,046.00	427.89	10,006.14	0.00

   2)Categorical Table

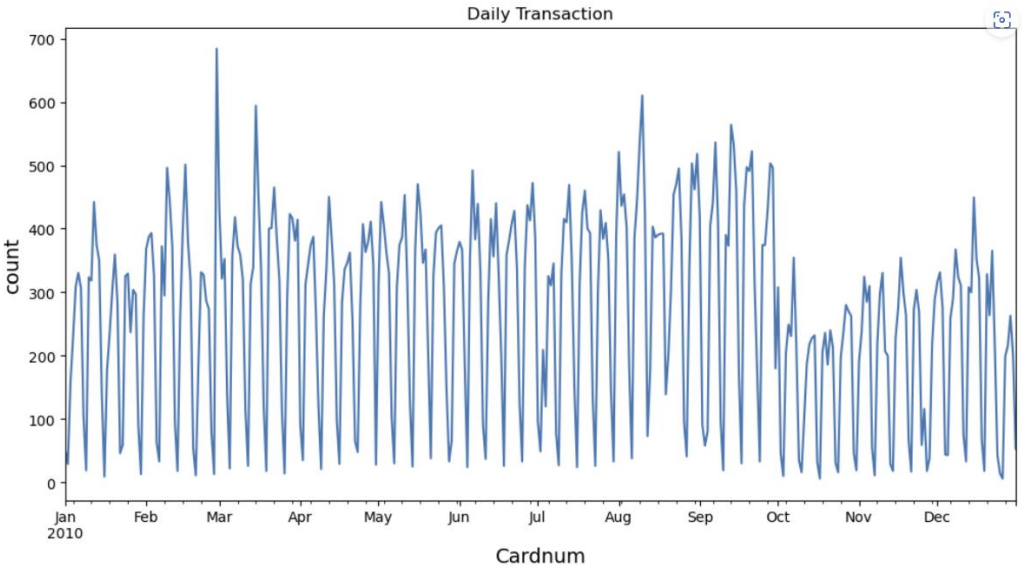
Field Name	%Populated	#Unique Values	Most Common Value
Recnum	100.00	96,753	NA
Cardnum	100.00	1,645	5142148452
Merchnum	96.51	13,092	930090121224
Merch Description	100.00	13,126	GSA-FSS-ADV
Merch State	98.76	227	TN
Merch Zip	95.19	4,567	38118
Transtype	100.00	4	P
Fraud	100.00	2	0

3. Visualization of Field
   1) Field Name: **Recnum**
      - Description: number of records: ordinal unique positive integer for each record, starting from 1, ending with 96,753.

2) Field Name: **Cardnum**
    - Description: Number of cards
    - The most common card number is 5142148452 and count is 1,192.

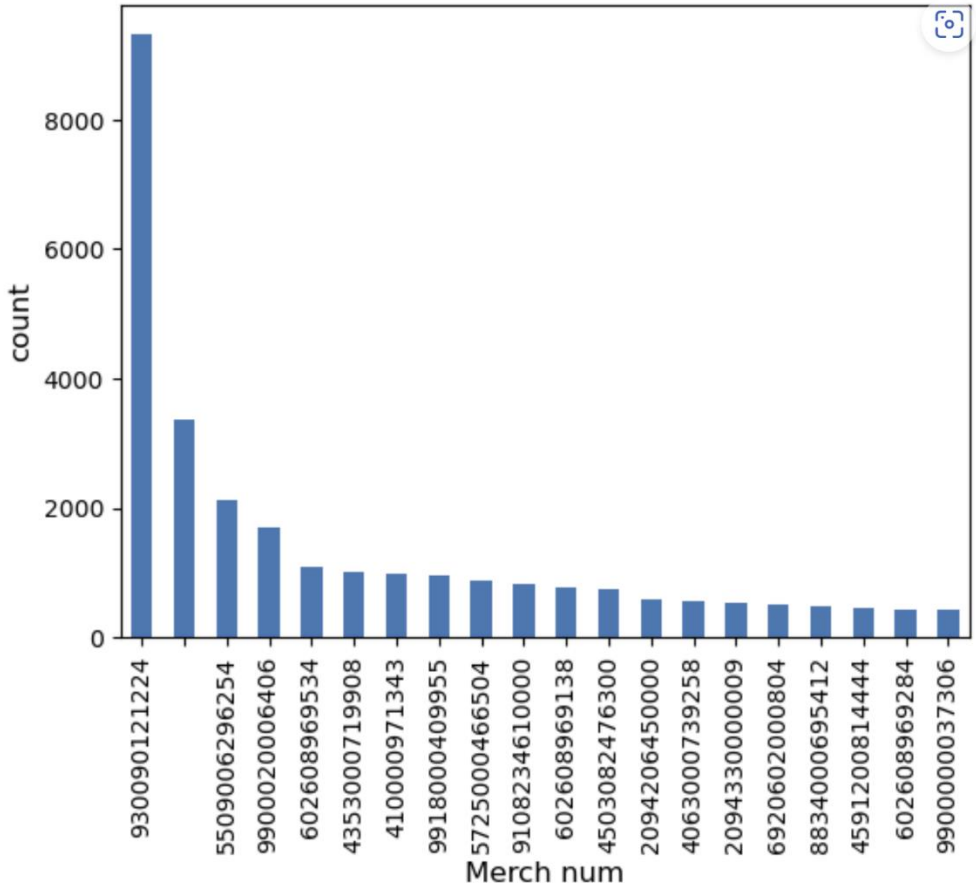


3) Field Name: **Date**
    - Description: The daily transaction from Jan. 2010 to Dec. 2010.

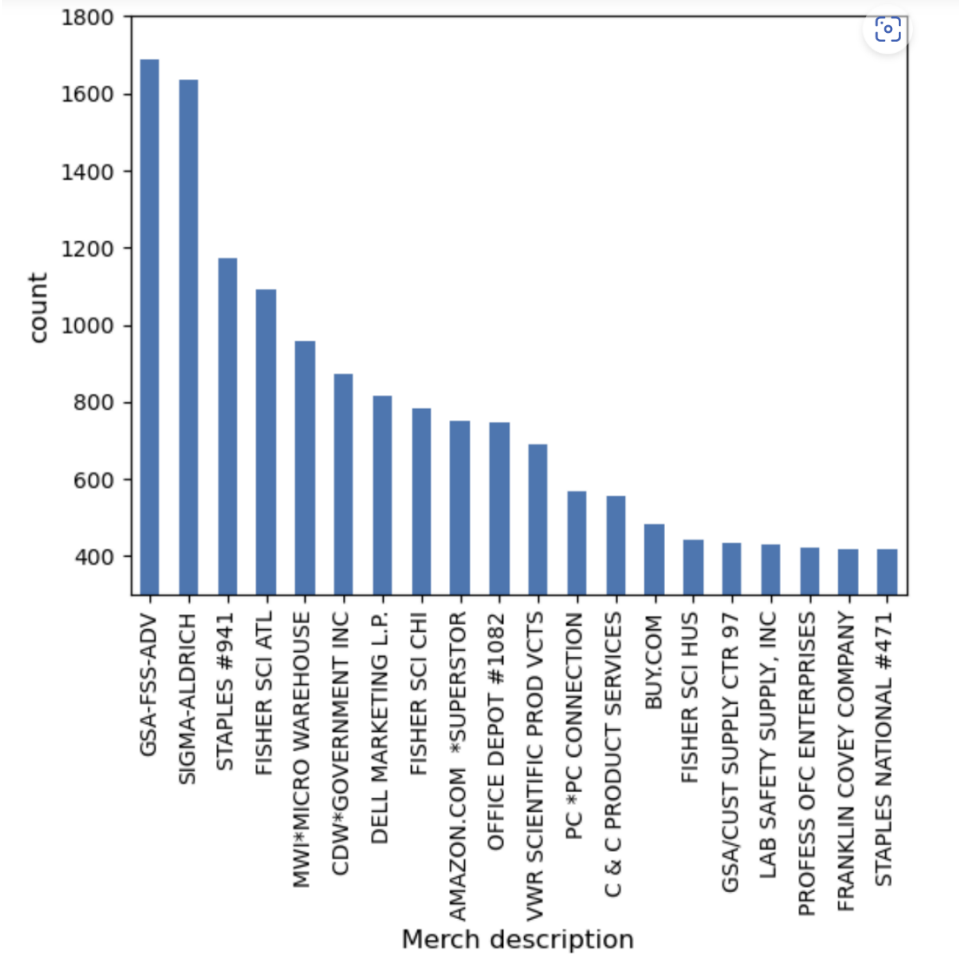


4) Field Name: **Merchnum**

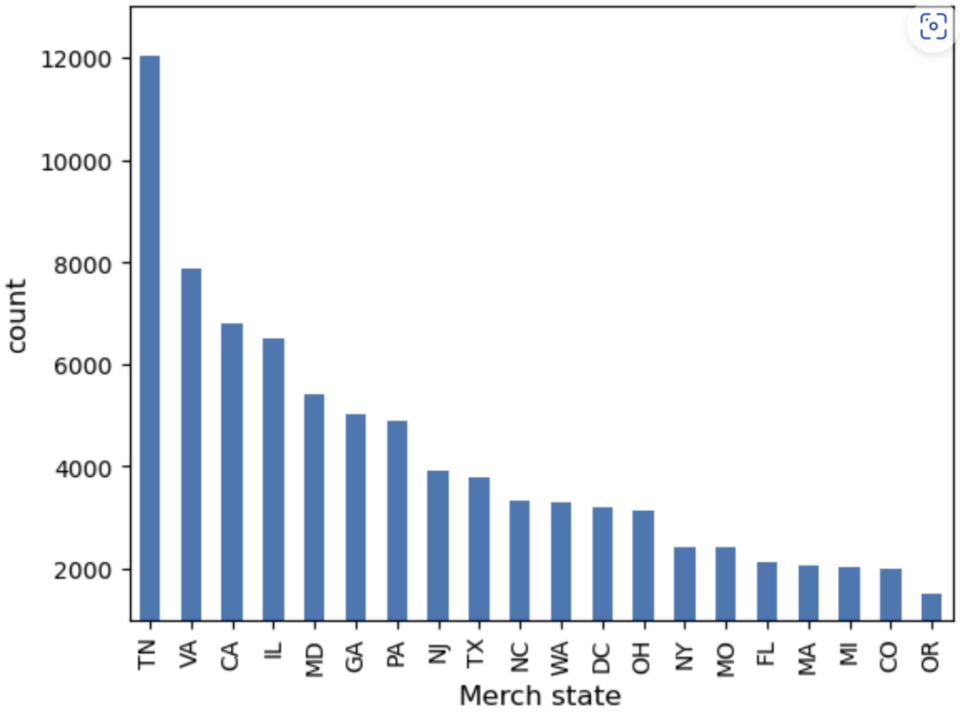
- Description: the number of merchants. The distribution is top 20 field values of all merchant numbers.
- The most common number of merchants is 930090121224 and count is 9,310. In addition, the second common merchant number is null.



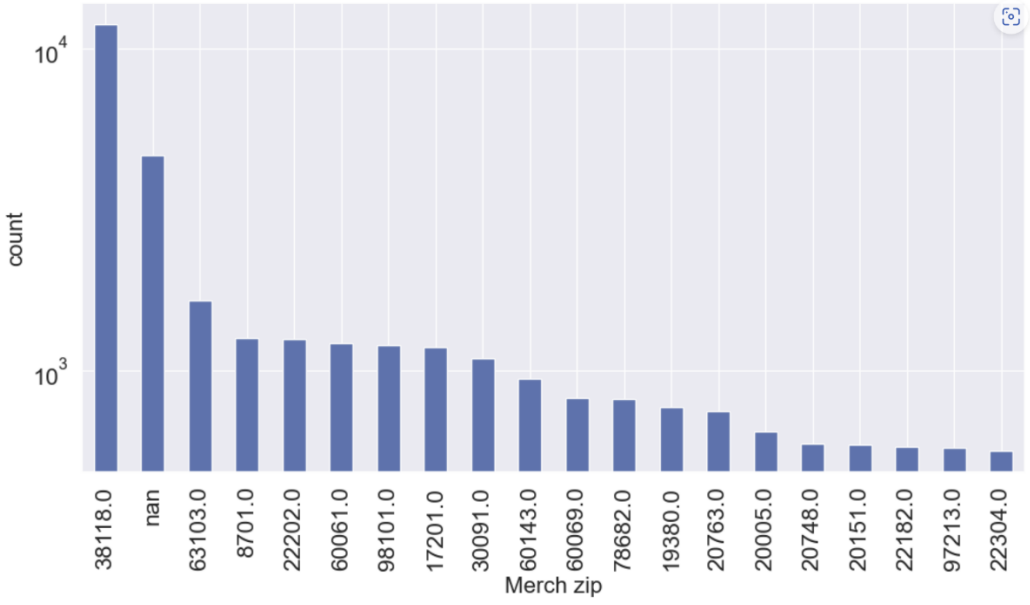
5) Field Name: **Merch description**
   - Description: the description of merchant. The distribution is top 20 field values of merchant description.
   - The most common description of merchant is GSA-FSS-ADV and the count is 1,688.



6) Field Name: **Merch state**
    - Description: the state where is merchant locates. The distribution is top 20 field values of states.
    - The most common state is TN, and the count is 12,035.



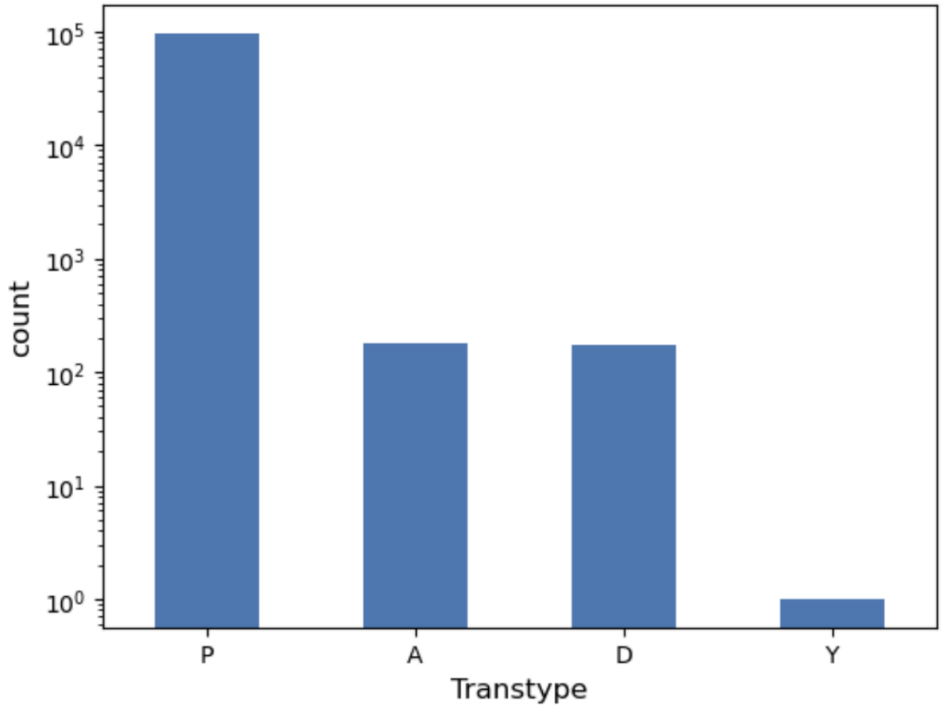
7) Field Name: **Merch zip**
   - Description: the 5-digit zip code of merchants. The distribution is top 20 field values of zip codes.
   - The most common zip code is 38118, and the second common one is nan. The count of zip code 38118 is 11868.



8) Field Name: **Transtype**
   - Description: the type of transaction. The distribution is 4 types of transaction, which are P: purchase , A: approval, D: debit from a account, Y: year-end

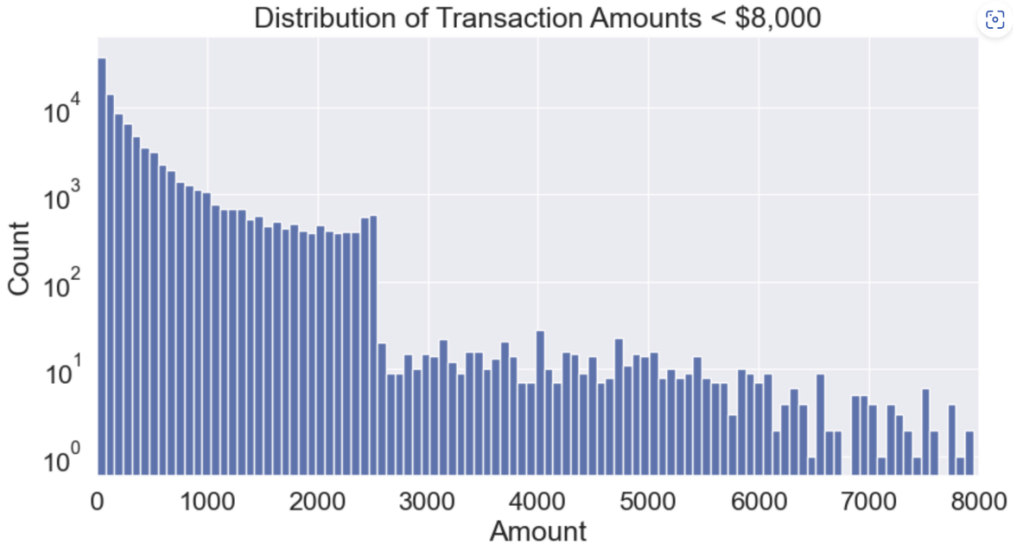
processing. However, we can only confirm the type of purchase, and three other types are based on research results.

- The most common type of transaction is P and the count is 96,398.



9) Field Name: **Amount**

- Description: amount of transaction. The distribution is counts of amounts which is less than $8,000.
- The general trend of amounts is decreasing from $0 to $8,000. Also, there is a gap around $2,500.



10) Field Name: **Fraud**

Description: the label of fraud, fraud_label = 0 (not frand); fraud_label =1 (fraud)
Count (fraud_label=0)= 95,694; count(fraud_label=1) =1,059