Application Frauds Project

1. Data Observation

- **Data Description**
  The data is about **Application Records including Fields of Personal Identifying Information**. The data is synthetic data made from analysis on a few billion real U.S. applications to find application identity fraud. The data covers the time of **year 2017** with total **1,000,000 records** and **10 fields**.

- **Statistics Tables of Numeric and Categorical Fields**
  The followings are summary of statistics for numeric and categorical fields. We can observe from the tables that there are some **common values** in fields: 999999999 in ssn, 123 MAIN ST in address, 19070626 in dob, and 9999999999 in homephone. These common values were fixed in the next data cleaning step.

  - **Numeric Table**

Field Name	% Populated	Min	Max	Mean	Stdev	% Zero
date	100.00	2017-01-01	2017-12-31	N/A	N/A	0.00
dob	100.00	1900-01-01	2016-10-31	N/A	N/A	0.00

  - **Categorical Table**

Field Name	% Populated	# Unique Values	Most Common Field Value
record	100.00	1,000,000	N/A
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

2. Data Cleaning

- **Fix Frivolous Field Values**

    - **What**: As we mentioned above, the common values in ssn, address, dob, and homephone fields are frivolous field values that were missing but have been filled in with default values in business.

    - **Why**: Since most of the variables to be created count the linkage variables across many records, the frivolous field values included in the linkage variables will cause counts wrong. Therefore, a record with frivolous values will be incorrectly and improperly detected as a signal of identity fraud when it is not fraudulent, confusing the model.

    - **How:** We replaced these with the record number that is unique and will not link to any previous value for that field.

- **Other Data Cleaning Methods**
    - **Data Type**: Converted from date time data type into string data type.
    - **Leading Zero**: Filled in the leading zero in ssn, zip code, and homephone fields.

3. Variable Creation

- **Identity Fraud Modes**
  There are three kinds of identity frauds: identity theft, identity manipulation, and synthetic identity. A fraudster uses a real but stolen PII information in an identity theft fraud while a fraudster slightly changes his PII information in an identity manipulation fraud. For a synthetic identity fraud, a fraudster completely makes up an identity information. There are two modes for all these identity frauds:

    - **Fraudster**: An individual fraudster uses the same PII information stolen or manipulated and his own contact information for different applications. Therefore, there are many applications with same PII information and same contact information.

    - **Victim**: A victim's PII information is compromised and used by many fraudsters. Therefore, there are many applications with same PII information but different contact information.

- **Variables**
  Responding to the identity fraud modes, we created four kinds of variables to check the frequency of linkage entities (attribute) that were concatenated by two original fields: Days since, Velocity, Relative Velocity, and Counts by entities.

    - **Target Encoding**: In addition, we also had one target encoded variable that was converted from categorical date fields into numeric.

- **Summary of Independent Variables**
  The following table shows a summary of independent variables with explanations. (total **3,960** with record number and fraud_label)

Family of Variables	Description of Variables	# Variables Created
Age Variable: 'age_when_apply'	The age of an applicant on each application. It is calculated by subtracting the year of 'dob' from the year of 'date'.	1
Target Encoded Variable for day of week: 'dow_risk'	This variable is the average of the dependent variable 'fraud_label' for all records in each day of week.	1
Velocity Variables	# records with the same group/entity of attributes over the past {0,1,3,7,14,30} days.	138
Days Since Variables	# days since a record was seen with that entity/group of the attribute.	23
Relative Velocity Variables	This set of variables is calculated by taking # records with that entity of a attribute seen in the recent past {0,1} days over # records with that same entity seen in the past {3,7,14,30} days.	184
Counts by Entities Variables	This set of variables are calculated by counting unique amounts of one attribute that is linked to each group/entity of another attribute over the past {0,1,3,7,14,30,60} days. For example, count the unique 'name_dob' with each group of 'ssn' in 3 days.	3542
Age Indicator Variables	This set of variables include the maximum, minimum, and mean values of age that is linked to each group/entity of attributes.	69
	Total Variables Created	**3958**

4. Feature Selection

- **Motivation**

After removing duplicated variables, we have **2,148** independent variables, indicating a high dimensional space. As dimensionality increases, data becomes sparse quickly and all points become outliers, causing a **curse of dimensionality** in nonlinear models. Therefore, we reduced the number of independent variables, implementing the feature selection process for this supervised project 1.

- **Feature Selection Process**

There were three steps for a feature selection: filter, wrapper, and embedded.

- **Filter**: we used KS as the univariate measure to calculate the correlation between each independent variable and fraud_label. We sorted all independent variables by the KS-filter score in a descending order and chose the first **400**.

- **Wrapper**: we used the stepwise selection to build models by adding or removing a variable at any stage until there was no significant improvement in the detection rate. I reduced the number of independent variables into **20**.

- **Embedded** (**not for this project**): we usually reduce complexity of a model by building the decision tree or regularization (lasso or ridge) model.

- **List of Final Variables**

The following is a list of final variables with **num_filter = 400 and num_wrapper = 20**

Wrapper Order	Variable	Filter Score
1	fulladdress_day_since	0.333268535624752
2	name_dob_count_30	0.228626326143987
3	address_unique_count_for_name_homephone_60	0.292437957378176
4	fulladdress_unique_count_for_dob_homephone_3	0.282977778919533
5	address_unique_count_for_homephone_name_dob_30	0.283989209761716
6	address_unique_count_for_ssn_name_dob_14	0.288127272956783
7	address_day_since	0.333268535624752
8	address_count_14	0.322436279511616
9	address_count_7	0.301735277082595
10	address_count_0_by_30	0.291922188990773
11	address_unique_count_for_homephone_name_dob_60	0.291409787536268
12	fulladdress_count_0_by_30	0.289723616808317
13	address_unique_count_for_ssn_zip5_60	0.289723616808317
14	address_unique_count_for_ssn_name_60	0.289679212026638
15	address_unique_count_for_ssn_firstname_60	0.288127272956783
16	address_unique_count_for_ssn_name_dob_60	0.287644886904071
17	address_unique_count_for_dob_homephone_60	0.287555865011610
18	address_unique_count_for_ssn_homephone_60	0.289166404580223
19	address_unique_count_for_ssn_lastname_60	0.287443596582453
20	address_unique_count_for_ssn_60	0.285913354746840

5. Model Exploration

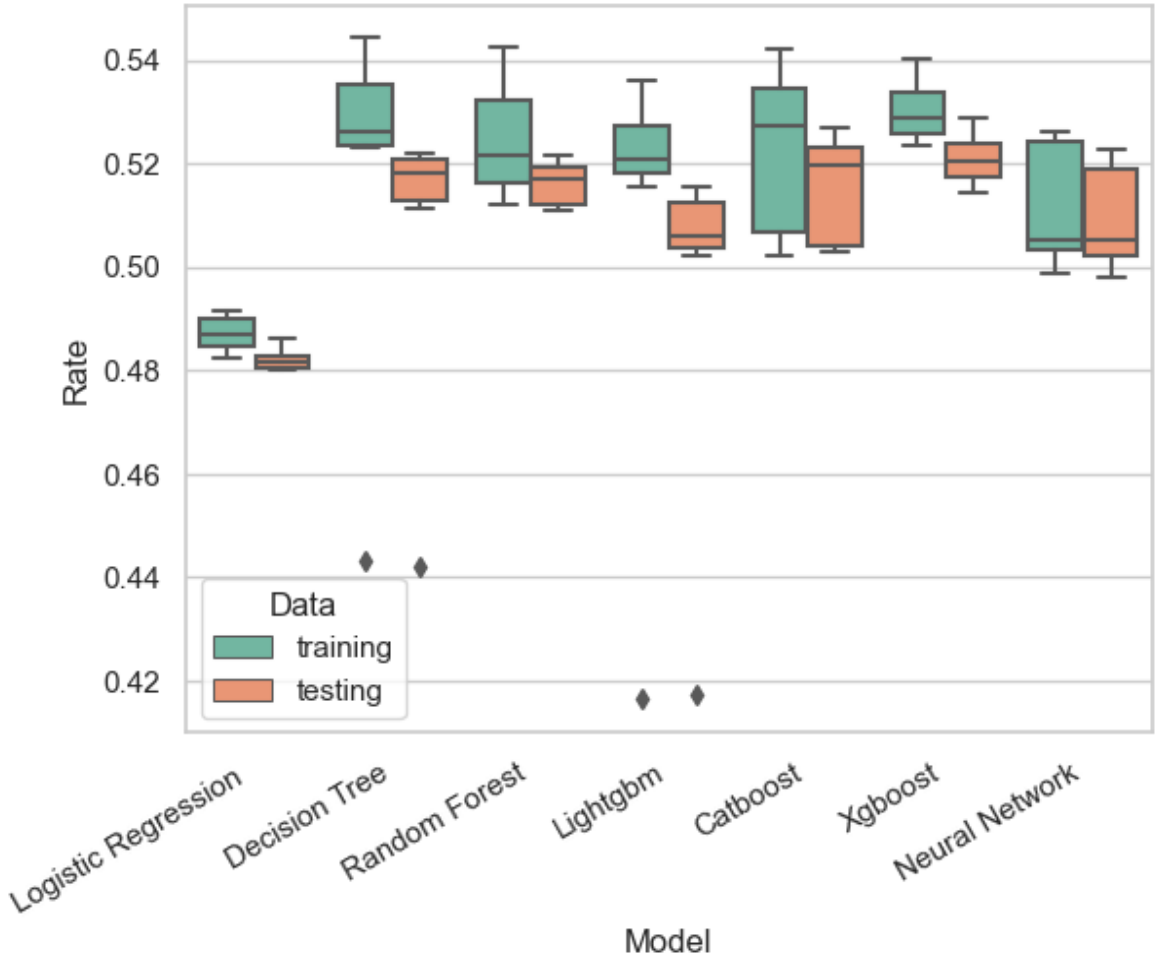
- **Hyperparameters Selection and Model Analysis**

I started from a simple model - the logistic regression and tried 6 nonlinear models with the number of variables starting from 10, 15, to 20. I firstly chose the hyperparameters that make the model overfitting (e.g., train-test > 0.02 and train over 0.535) and then tuned the hyperparameters to an underfitting performance (low train and low test).

- **Models Selection**

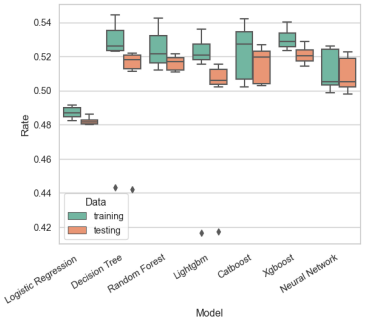
From the chart and the boxplot, I chose **CatBoost** as the final model because it generally has smaller difference between training and test rate and higher rate for training and test rate.

  - **Boxplot Comparing Different Models**



- Models Exploration Table

Models	Hyperparameters										Average FDR at 3%			Models Analysis	
Logistic Regression	**Number of Variables**	**max_iter**		**solver**		**penalty**		**C**			**Train**	**Test**	**OOT**	**DIFF (trn-tst)**	**Performance**
1	10	20		lbfgs		l2		1			0.492	0.480	0.473	0.011	Overfitting
2	10	20		lbfgs		none		0.25			0.490	0.483	0.473	0.008	
3	10	20		lbfgs		l2		0.1			0.489	0.486	0.473	0.003	Best Model
4	15	20		sag		none		0.5			0.485	0.480	0.469	0.005	
5	20	20		lbfgs		l2		0.1			0.485	0.482	0.470	0.002	
6	20	20		sag		l2		0.1			0.482	0.481	0.467	0.001	Underfitting
Decision Tree	**Number of Variables**	**splitter**	**max_depth**	**min_samples_split**		**min_samples_leaf**		**max_features**			**Train**	**Test**	**OOT**	**DIFF (trn-tst)**	**Performance**
1	10	best	25	20		15		10			0.543	0.514	0.502	0.028	Overfitting
2	10	random	30	25		20		5			0.528	0.521	0.498	0.007	
3	10	random	5	30		30		10			0.443	0.442	0.418	0.001	Underfitting
4	15	best	25	40		15		10			0.544	0.511	0.503	0.033	Overfitting
5	15	random	25	40		20		5			0.526	0.521	0.498	0.005	Best Model
6	20	random	15	50		30		10			0.524	0.518	0.494	0.006	
7	20	random	15	20		20		5			0.523	0.522	0.496	0.001	
Random Forest	**Number of Variables**	**n_estimators**	**max_depth**	**min_samples_split**		**min_samples_leaf**		**max_features**			**Train**	**Test**	**OOT**	**DIFF (trn-tst)**	**Performance**
1	10	15	25	50		30		10			0.542	0.512	0.501	0.031	Overfitting
2	10	100	5	50		30		5			0.525	0.520	0.499	0.004	
3	10	15	5	50		15		5			0.522	0.521	0.499	0.000	Best Model
4	15	100	5	15		15		5			0.520	0.517	0.496	0.003	
5	15	5	5	15		15		10			0.513	0.512	0.489	0.001	Underfitting
6	20	5	30	50		15		5			0.540	0.519	0.503	0.021	Overfitting
7	20	5	5	50		15		5			0.512	0.511	0.488	0.001	Underfitting
Lightgbm	**Number of Variables**	**n_estimators**	**max_depth**	**num_leaves**	**min_split_gain**	**reg_lambda**	**reg_alpha**	**learning_rate**	**subsample**		**Train**	**Test**	**OOT**	**DIFF (trn-tst)**	**Performance**
1	10	100	2	256	0	0	0	0.1	0.25		0.536	0.516	0.506	0.020	Overfitting
2	10	100	5	8	0	0	0	0.1	0.25		0.515	0.505	0.489	0.010	
3	10	100	2	32	0	0.3	0	0.25	0.25		0.522	0.514	0.493	0.008	Best Model
4	15	100	5	10	0	0.3	0.5	0.25	0.5		0.521	0.511	0.494	0.010	
5	15	100	2	2	0.5	0	0.5	0.25	0.5		0.416	0.417	0.409	-0.001	Underfitting
6	20	100	5	32	0.5	0.3	0	0.25	0.5		0.533	0.502	0.491	0.030	Overfitting
7	20	100	2	10	0	0.3	0	0.25	0.5		0.521	0.506	0.491	0.015	
Catboost	**Number of Variables**	**iterations**	**depth**	**random_strength**		**l2_leaf_reg**		**learning_rate**			**Train**	**Test**	**OOT**	**DIFF (trn-tst)**	**Performance**
1	10	1000	8	0.5		1		0.1			0.540	0.520	0.502	0.020	Overfitting
2	10	1000	6	0.5		1		0.01			0.529	0.524	0.504	0.005	
3	15	5	8	0.5		5		0.1			0.507	0.503	0.486	0.005	
4	15	1000	6	1		1		0.01			0.527	0.527	0.503	0.001	Best Model
5	15	5	8	1		5		0.1			0.506	0.504	0.483	0.001	Underfitting
6	20	1000	8	1		1		0.1			0.542	0.522	0.503	0.020	Overfitting
7	20	5	6	1		1		0.1			0.502	0.504	0.483	-0.002	Underfitting
Xgboost	**Number of Variables**	**max_depth**	**min_child_weight**	**subsample**	**reg_lambda**	**reg_alpha**	**gamma**	**learning_rate**			**Train**	**Test**	**OOT**	**DIFF (trn-tst)**	**Performance**
1	10	2	1	0.5	0.5	0.5	0	0.1			0.527	0.514	0.501	0.013	
2	10	6	1	1	1	0.5	0	0.1			0.530	0.525	0.507	0.005	
3	10	2	0.5	0.5	0.5	0	0.25	0.1			0.524	0.520	0.501	0.004	
4	15	6	0.5	0.5	1	0.5	0.25	0.25			0.538	0.516	0.503	0.021	Overfitting
5	15	6	1	1	0.5	0	0.25	0.1			0.529	0.529	0.507	0.000	Best Model
6	15	2	0.5	0.5	0.5	0	0	0.1			0.523	0.522	0.501	0.001	
7	20	6	0.5	1	0.5	0	0.25	0.25			0.540	0.518	0.506	0.022	Overfitting
Neural Network	**Number of Variables**	**hidden_layer_sizes**	**activation**	**solver**	**learning_rate**	**alpha**		**learning_rate_init**			**Train**	**Test**	**OOT**	**DIFF (trn-tst)**	**Performance**
1	10	(10,10)	relu	adam	constant	0.0001		0.01			0.526	0.520	0.502	0.006	
2	10	(20,20,20)	logistic	adam	adaptive	0.0001		0.01			0.526	0.523	0.501	0.003	Best Model
3	10	(5,)	logistic	adam	constant	0.1		0.01			0.499	0.503	0.480	-0.004	Underfitting
4	15	(5,)	relu	adam	adaptive	0.0001		0.01			0.522	0.518	0.501	0.004	
5	15	(5,)	logistic	adam	adaptive	0.1		0.01			0.502	0.501	0.482	0.001	
6	20	(5,)	logistic	adam	constant	0.1		0.01			0.505	0.498	0.483	0.007	
7	20	(5,)	logistic	adam	adaptive	0.1		0.01			0.505	0.505	0.483	-0.001	Underfitting



6. Result Summary

- **Final Model**

The followings are the details of our final model.

- **Model Architecture**: **CatBoost**

- **Model Hyperparameters**

Iterations	1000
Depth	6
Random_strength	1
12_leaf_reg	1
Learning_rate	0.01

- **Final Variables: 15** independent variables from the wrapper list. Here is the list of 17 final variables (including the record number and fraud_label)

[Recnum, Fraud, fulladdress_day_since, name_dob_count_30, address_unique_count_for_name_homephone_60, fulladdress_unique_count_for_dob_homephone_3, address_unique_count_for_homephone_name_dob_30, address_unique_count_for_ssn_name_dob_14, address_day_since, address_count_14, address_count_7, address_count_0_by_30, address_unique_count_for_homephone_name_dob_60, fulladdress_count_0_by_30, address_unique_count_for_ssn_zip5_60, address_unique_count_for_ssn_name_60, address_unique_count_for_ssn_firstname_60]

- **Summary Columns**

We evaluate performance of the final model for training, testing and OOT population. The followings are explanation for each column in summary table.

Population Bin %	Percentage of population # records
# Records	Every 1% of population # records
# Goods	The increase in # goods with an increase of 1% of population records
# Bads	The increase in # bads with an increase of 1% of population records
% Goods	# Goods / # Records
% Bads	# Bads / # Records
Total # Records	Population bin % of population # records
Cumulative Goods	Total # goods in bin % of population
Cumulative Bads	Total # bads in bin % of population
% Cumulative Goods	Total # goods in bin % of population / Total # goods in 100 % of population
FDR (% Cumulative Bads)	Total # bads in bin % of population / Total # bads in 100 % of population
KS	% Cumulative Goods - % Cumulative Bads. It measures how well the goods and bads are separated.
FPR	Cumulative Goods / Cumulative Bads. It measures probability that we predict one record as bad that is actually good.

- **Summary Tables**

Training	Population Total # Records	Population Total # Goods	Population Total # Bads	Actual Fraud Rate
	583,454	575,081	8,373	0.014559688

	Bin Statistics					Cumulative Statistics				Model Performance		
Population Bin %	# Records	# Goods	# Bads	% Goods	% Bads	Total # Records	Cumulative Goods	Cumulative Bads	% Cumulative Goods	% Cumulative Bads (FDR)	KS	FPR
1	5835	1604	4231	27.49%	72.51%	5835	1604	4231	0.28%	50.53%	50.25%	0.38
2	5834	5716	118	97.98%	2.02%	11669	7320	4349	1.27%	51.94%	50.67%	1.68
3	5835	5761	74	98.73%	1.27%	17504	13081	4423	2.27%	52.82%	50.55%	2.96
4	5834	5787	47	99.19%	0.81%	23338	18868	4470	3.28%	53.39%	50.10%	4.22
5	5835	5799	36	99.38%	0.62%	29173	24667	4506	4.29%	53.82%	49.53%	5.47
6	5834	5782	52	99.11%	0.89%	35007	30449	4558	5.29%	54.44%	49.14%	6.68
7	5835	5792	43	99.26%	0.74%	40842	36241	4601	6.30%	54.95%	48.65%	7.88
8	5834	5790	44	99.25%	0.75%	46676	42031	4645	7.31%	55.48%	48.17%	9.05
9	5835	5791	44	99.25%	0.75%	52511	47822	4689	8.32%	56.00%	47.69%	10.20
10	5834	5786	48	99.18%	0.82%	58345	53608	4737	9.32%	56.57%	47.25%	11.32
11	5835	5792	43	99.26%	0.74%	64180	59400	4780	10.33%	57.09%	46.76%	12.43
12	5834	5783	51	99.13%	0.87%	70014	65183	4831	11.33%	57.70%	46.36%	13.49
13	5835	5797	38	99.35%	0.65%	75849	70980	4869	12.34%	58.15%	45.81%	14.58
14	5835	5786	49	99.16%	0.84%	81684	76766	4918	13.35%	58.74%	45.39%	15.61
15	5834	5798	36	99.38%	0.62%	87518	82564	4954	14.36%	59.17%	44.81%	16.67
16	5835	5800	35	99.40%	0.60%	93353	88364	4989	15.37%	59.58%	44.22%	17.71
17	5834	5791	43	99.26%	0.74%	99187	94155	5032	16.37%	60.10%	43.73%	18.71
18	5835	5788	47	99.19%	0.81%	105022	99943	5079	17.38%	60.66%	43.28%	19.68
19	5834	5790	44	99.25%	0.75%	110856	105733	5123	18.39%	61.18%	42.80%	20.64
20	5835	5799	36	99.38%	0.62%	116691	111532	5159	19.39%	61.61%	42.22%	21.62

Testing	Population Total # Records	Population Total # Goods	Population Total # Bads	Actual Fraud Rate
	250,053	246,419	3,634	0.014747239

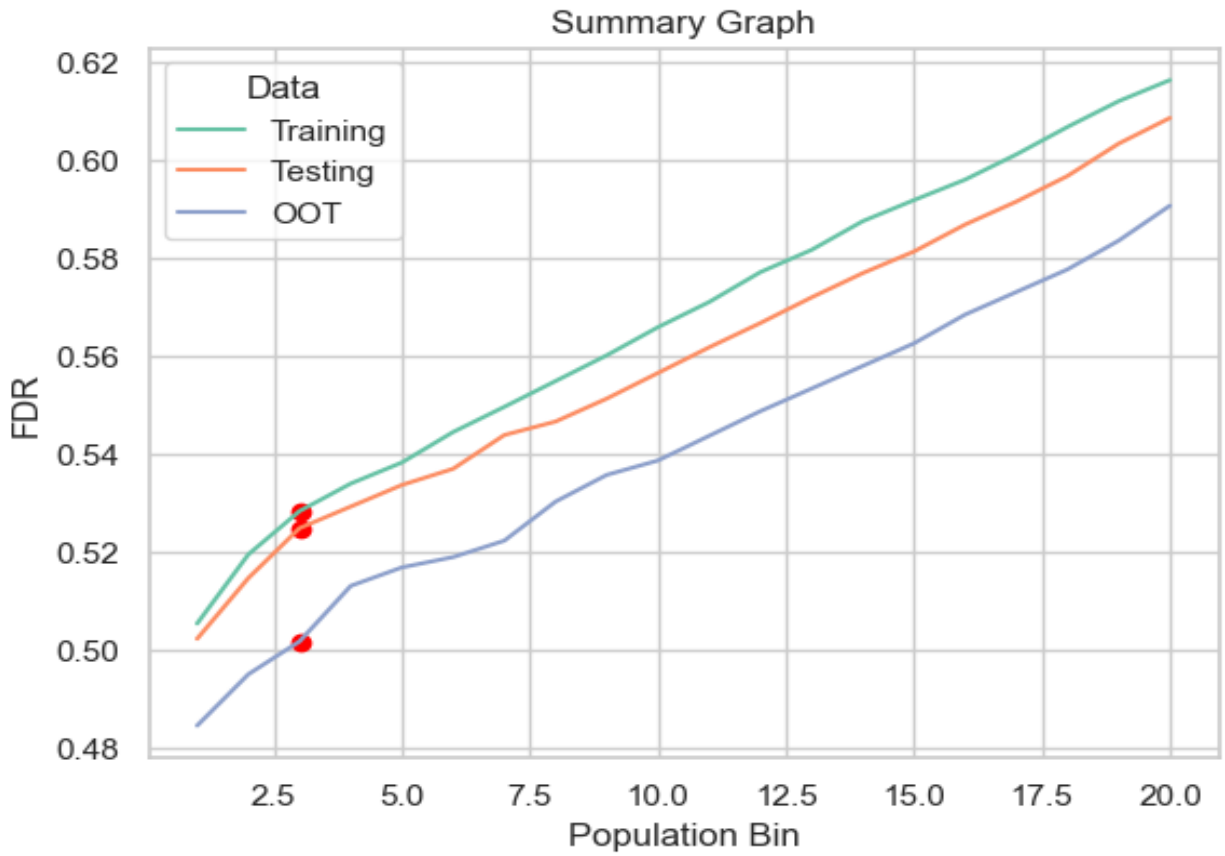
	Bin Statistics					Cumulative Statistics				Model Performance		
Population Bin %	# Records	# Goods	# Bads	% Goods	% Bads	Total # Records	Cumulative Goods	Cumulative Bads	% Cumulative Goods	% Cumulative Bads (FDR)	KS	FPR
1	2501	676	1825	27.03%	72.97%	2501	676	1825	0.27%	50.22%	49.95%	0.37
2	2500	2455	45	98.20%	1.80%	5001	3131	1870	1.27%	51.46%	50.19%	1.67
3	2501	2464	37	98.52%	1.48%	7502	5595	1907	2.27%	52.48%	50.21%	2.93
4	2500	2484	16	99.36%	0.64%	10002	8079	1923	3.28%	52.92%	49.64%	4.20
5	2501	2485	16	99.36%	0.64%	12503	10564	1939	4.29%	53.36%	49.07%	5.45
6	2500	2488	12	99.52%	0.48%	15003	13052	1951	5.30%	53.69%	48.39%	6.69
7	2501	2476	25	99.00%	1.00%	17504	15528	1976	6.30%	54.38%	48.07%	7.86
8	2500	2490	10	99.60%	0.40%	20004	18018	1986	7.31%	54.65%	47.34%	9.07
9	2501	2484	17	99.32%	0.68%	22505	20502	2003	8.32%	55.12%	46.80%	10.24
10	2500	2481	19	99.24%	0.76%	25005	22983	2022	9.33%	55.64%	46.31%	11.37
11	2501	2482	19	99.24%	0.76%	27506	25465	2041	10.33%	56.16%	45.83%	12.48
12	2500	2482	18	99.28%	0.72%	30006	27947	2059	11.34%	56.66%	45.32%	13.57
13	2501	2482	19	99.24%	0.76%	32507	30429	2078	12.35%	57.18%	44.83%	14.64
14	2500	2482	18	99.28%	0.72%	35007	32911	2096	13.36%	57.68%	44.32%	15.70
15	2501	2485	16	99.36%	0.64%	37508	35396	2112	14.36%	58.12%	43.75%	16.76
16	2500	2480	20	99.20%	0.80%	40008	37876	2132	15.37%	58.67%	43.30%	17.77
17	2501	2484	17	99.32%	0.68%	42509	40360	2149	16.38%	59.14%	42.76%	18.78
18	2501	2482	19	99.24%	0.76%	45010	42842	2168	17.39%	59.66%	42.27%	19.76
19	2500	2476	24	99.04%	0.96%	47510	45318	2192	18.39%	60.32%	41.93%	20.67
20	2501	2482	19	99.24%	0.76%	50011	47800	2211	19.40%	60.84%	41.44%	21.62

OOT	Population Total # Records	Population Total # Goods	Population Total # Bads	Actual Fraud Rate
	166,493	164,107	2,386	0.014539294

	Bin Statistics					Cumulative Statistics				Model Performance		
Population Bin %	# Records	# Goods	# Bads	% Goods	% Bads	Total # Records	Cumulative Goods	Cumulative Bads	% Cumulative Goods	% Cumulative Bads (FDR)	KS	FPR
1	1665	509	1156	30.57%	69.43%	1665	509	1156	0.31%	48.45%	48.14%	0.44
2	1665	1640	25	98.50%	1.50%	3330	2149	1181	1.31%	49.50%	48.19%	1.82
3	1665	1649	16	99.04%	0.96%	4995	3798	1197	2.31%	50.17%	47.85%	3.17
4	1665	1638	27	98.38%	1.62%	6660	5436	1224	3.31%	51.30%	47.99%	4.44
5	1665	1656	9	99.46%	0.54%	8325	7092	1233	4.32%	51.68%	47.35%	5.75
6	1665	1660	5	99.70%	0.30%	9990	8752	1238	5.33%	51.89%	46.55%	7.07
7	1665	1657	8	99.52%	0.48%	11655	10409	1246	6.34%	52.22%	45.88%	8.35
8	1664	1645	19	98.86%	1.14%	13319	12054	1265	7.35%	53.02%	45.67%	9.53
9	1665	1652	13	99.22%	0.78%	14984	13706	1278	8.35%	53.56%	45.21%	10.72
10	1665	1658	7	99.58%	0.42%	16649	15364	1285	9.36%	53.86%	44.49%	11.96
11	1665	1653	12	99.28%	0.72%	18314	17017	1297	10.37%	54.36%	43.99%	13.12
12	1665	1653	12	99.28%	0.72%	19979	18670	1309	11.38%	54.86%	43.48%	14.26
13	1665	1654	11	99.34%	0.66%	21644	20324	1320	12.38%	55.32%	42.94%	15.40
14	1665	1654	11	99.34%	0.66%	23309	21978	1331	13.39%	55.78%	42.39%	16.51
15	1665	1654	11	99.34%	0.66%	24974	23632	1342	14.40%	56.24%	41.84%	17.61
16	1665	1651	14	99.16%	0.84%	26639	25283	1356	15.41%	56.83%	41.43%	18.65
17	1665	1654	11	99.34%	0.66%	28304	26937	1367	16.41%	57.29%	40.88%	19.71
18	1665	1654	11	99.34%	0.66%	29969	28591	1378	17.42%	57.75%	40.33%	20.75
19	1665	1651	14	99.16%	0.84%	31634	30242	1392	18.43%	58.34%	39.91%	21.73
20	1665	1648	17	98.98%	1.02%	33299	31890	1409	19.43%	59.05%	39.62%	22.63

- **Summary Graph**

The following is the plot for training, testing, and OOT FDR as population bin increases.



- **Conclusion**

Based on the table above, we can get **52.82%** FDR at 3% population for **training** data, **52.48%** FDR at 3% population for **testing** data, and **50.17%** FDR at 3% population for **OOT** data. In conclusion, OOT FDR shows that the final model can eliminate about **50%** of the fraud by declining only about **3%** of the applications without any overfitting or underfitting.