Application fraud Identify

1. Data Description

- The data is synthetic data made from the analysis of a few billion real U.S. applications over about 10 years. The goal is to find application/identity fraud, so only identity fields. The data covers the time of 2017. There are 10 fields and 1000000 records.

- **Numerical Table**

Field name	% Populated	Min	Max	Mean	Stdev	% Zero
Date	100%	2017-01-01	2017-12-31	/	/	0.00

- **Categorial Table**

Field name	% Populated	# Unique Values	Most Common Value
Record	100%	1000000	N/A
ssn	100%	852745	938972725
Firstname	100%	78136	EAMSTRMT
Lastname	100%	177001	ERJSAXA
Address	100%	828774	123 MAIN ST
Zip5	100%	26370	68138
dob	100%	42673	19070626
homephone	100%	28244	9999999999
Fraud_label	100%	2	0

2. Data cleaning

To have a cleaner dataset and enable more reliable data analysis, we have to deal with missing and null values which is normal in real-life. Data cleaning is the process of fixing or removing incorrect, corrupted, incorrectly formatted, duplicate, or incomplete data within a dataset. This dataset utilized frivolous values, or dummy placeholders, to populate missing data fields, for example, 9999999999 for a blank SSN, 123 main Street for a blank address, and 9999999999 for a blank home phone. We call these filled-in values “frivolous”. This adds to the complexity.

And a good way to deal with frivolous values is to replace them with a unique record number so that they are independent of any previously recorded values in that particular field. For missing values, we have used statistical smoothing using a smooth curve.

3. Variable creation

• In particular, we discovered what is now called synthetic identity fraud. Synthetic identity fraud occurs when someone uses a combination of real and fake personal information to create an identity and commit fraud. Assign a numerical probability that the application is a synthetic identity fraud. The algorithms will score each application. We’ll focus on first name, last name, date of birth, phone number, zip code, address, SSN, and other identifying information. Thus, to explore more possibility, we scale up the number of variables.

- **Summary table of the created variable**

Description of variables	# Variables created
Original Fields from the dateset excluding ‘record’ and ‘fraud label’	8
Date of week target encoded average fraud percentage of that day within the week	1

New entities combining different original fields	18
Risk Variable # The likelihood of fraud for any of the week ('dow_risk)	1
Age when apply The current age of the applicant, which is the difference of the year of birth and application.	1
Days since Variables # days since an application with that entity has been seen. Entities list is attached below.	23
Velocity # records with the same entity over the last {0,1,3,7,14,30} days. Entities list is attached below.	138
Relative Velocity : This set of variables is the fraction of number with the same entity over the past 0 or 1 days out of the average amount with the same entity for {3, 7, 14, 30} days.	184
Amount Variables based on applications This set of variables includes unique records across entities with other fields like ssn, address, zip5, dob etc. over the last {0, 1, 3, 7,14, 30, 60} days.	3542
Maximum Indicator This set of variables include the maximum number of applications across each entity over the past {1, 3, 7, 30} days. Entities list is attached below.	92
Age Indicator This set of variables includes the average, maximum, and minimum of age across entities. The entities list is attached below.	69

Entities: ssn, address, zip5, dob, homephone, name, fulladdress, name dob, name fulladdress, name homephone, fulladdress dob, fulladdress homephone, dob homephone, homephone name dob, ssn firstname, ssn lastname, ssn address, ssn zip5, ssn dob, ssn homephone, ssn name, ssn fulladdress, ssn name dob

- **Note that:**
  The maximum indicator which is the max-type variable makes the model look much better, but it is improperly formed in that it looks into the future. The variable is able to be created for this historical data, but when you install the model the variables are built in real-time as the applications occur. There is no knowledge of the future during model use. So these variables that looked into the future will have values very different from what they had during model development. **Therefore, we have to remove the max-type variable before we continue the following process.**

4. Feature selection

- Since we build as many variables as possible without worrying about correlations or having too many variables in the last section. Now we are trying to reduce the number of input variables by feature selection to make the process more accurate and increase the prediction power of the algorithms. By doing feature selection, it allows our exploration and consideration of many, many candidate variables, essentially without limit. Then it results in a sorted list so we know which variables to add or remove in our model explorations. Also, it makes the nonlinear model runs much faster in order to optimize model architecture and hyperparameters.
- There are three ways to do feature selection: filters, wrappers, and embedded. In this case, we started with filter and wrappers and get a list of variable and their univariate KS's.

wrapper order	variable	KS

1	zip5_count_1	0.221239028
2	fulladdress_count_0_by_30	0.290722131
3	ssn_firstname_day_since	0.226427511
4	fulladdress_day_since	0.333268536
5	address_unique_count_for_ssn_zip5_60	0.289723617
6	address_count_30	0.332648157
7	address_day_since	0.334139944
8	address_count_14	0.32243628
9	fulladdress_count_14	0.321952925
10	address_count_7	0.301735277
11	fulladdress_count_7	0.301666247
12	address_unique_count_for_name_homephone_60	0.292437957
13	address_count_0_by_30	0.291922189
14	address_unique_count_for_homephone_name_dob_60	0.291409788
15	fulladdress_unique_count_for_ssn_homephone_60	0.289990622
16	address_unique_count_for_ssn_name_60	0.289679212
17	fulladdress_unique_count_for_name_homephone_60	0.289535139
18	address_unique_count_for_ssn_homephone_60	0.289166405
19	fulladdress_unique_count_for_homephone_name_dob_60	0.288482719
20	fulladdress_unique_count_for_dob_homephone_60	0.288442887
21	address_unique_count_for_ssn_firstname_60	0.288127273
22	address_unique_count_for_ssn_name_dob_60	0.287644887
23	address_unique_count_for_dob_homephone_60	0.287555865
24	address_unique_count_for_ssn_lastname_60	0.287443597
25	address_unique_count_for_name_60	0.287411369
26	fulladdress_unique_count_for_ssn_name_60	0.286799367
27	fulladdress_unique_count_for_ssn_lastname_60	0.286776127
28	fulladdress_unique_count_for_ssn_60	0.286764374
29	fulladdress_unique_count_for_ssn_firstname_60	0.286763364
30	address_unique_count_for_ssn_60	0.285913355

5. Preliminary models exploration.

- In this section, we tested between 10-20 variables from the last variable sheet to explore model performance with different hyperparameters. We have tested 7 model algorithms with 10, 13, and 15 variables: Logistic Regression, Decision Tree, Random Forest, Neural Network, Boosted Tree(LGBM), Boosted Tree(Catboost), and Boosted Tree(XGB). In the table below, I was trying to reach the overfitting for example the yellow part in the table.

- **Exploring model table**

Note： yellew means overfitting, red means the highest OOT except overfitting.

Model	Parameter							Average FDR at 3%		
Logistic Regression	Iteration	ber of vari	penalty	max_iter	solver	l1 ratio		Train	Test	oot
	1 (default)	10	l2	20	lbfgs	None		0.486273	0.492734	0.473931
	2	10	l2	100	lbfgs	None		0.489318	0.486602	0.47435
	3	10	l1	100	liblinear	None		0.488685	0.487187	0.473847
	4	10	None	100	lbfgs	None		0.487016	0.491038	0.473764
	5	10	None	300	saga	None		0.487469	0.489882	0.473596
	6	13	l2	800	lbfgs	None		0.484443	0.485857	0.470662
	7	13	elasticnet	500	saga	0.2		0.479723	0.48766	0.468315
Decision Tree	Iteration	ber of vari	criterion	max_depth	_samples_	_samples_leaf		Train	Test	oot
	1 (default)	10	gini	2	1000	500		0.460889	0.455637	0.443839
	2	10	gini	10	1000	500		0.525895	0.530781	0.504191
	3	10	gini	100	10	5		0.543153	0.514857	0.499329
	4	10	gini	20	800	30		0.531493	0.52484	0.502431
	5	10	entropy	10	500	50		0.527655	0.528578	0.503856
	6	10	entropy	60	20	200		0.5327	0.522195	0.503604
	7	13	entropy	20	200	200		0.531171	0.526447	0.502934
	8	13	gini	10	500	20		0.526151	0.532454	0.506622
	9	15	gini	10	500	20		0.53031	0.519892	0.506203
	10	15	gini	12	1200	500		0.52661	0.523834	0.503018
	11	15	gini	8	200	800		0.525514	0.526305	0.50394
Random Forest	Iteration	ber of vari	_estimato	max_depth	_samples_	_samples_	ax_feature	Train	Test	oot
	1 (default)	10	3	2	1000	500	8	0.476879	0.4762	0.461945
	2	10	10	2	1000	500	8	0.474855	0.479797	0.463034
	3	10	20	2	1000	500	8	0.491585	0.489851	0.474769
	4	10	10	5	100	300	8	0.522228	0.519269	0.496479
	5	10	10	2	800	100	5	0.496837	0.50186	0.479715
	6	13	10	2	800	100	5	0.496146	0.501041	0.478793
	7	13	12	3	1000	180	10	0.50036	0.501228	0.48223
	8	13	12	3	1000	500	10	0.50346	0.503253	0.48223
	9	15	10	2	800	100	5	0.49875	0.501125	0.479799
Neural Network	Iteration	ber of vari	en_layer_	activation	alpha	earning rat	max iter	Train	Test	oot
	1 (default)	10	2	relu	0.0001	constant	200	0.484215	0.489079	0.467309
	2	10	5	relu	0.0001	constant	200	0.517586	0.512414	0.496396
	3	10	10	relu	0.0001	constant	200	0.525478	0.52243	0.503185
	4	10	3	identity	0.0001	constant	200	0.485245	0.492404	0.473596
	5	10	4	identity	0.0001	adaptive	100	0.489676	0.489909	0.475189
	6	13	4	identity	0.0001	adaptive	100	0.488596	0.493295	0.473764
	7	13	3	relu	0.0001	constant	200	0.508493	0.510174	0.488181
	8	13	4	relu	0.0001	constant	300	0.514771	0.516191	0.49254
	9	13	5	relu	0.0001	constant	300	0.51688	0.515991	0.496563
	10	13	5	relu	0.0001	constant	100	0.508959	0.509968	0.49036
Boosted Tree (LGBM)	Iteration	ber of vari	num_leave	_estimato	osting_ty	subsample	earning_ra	Train	Test	oot
	1 (default)	10	2	5	gbdt	1	0.1	0.510607	0.516869	0.489522
	2	10	20	10	gbdt	1	0.1	0.524995	0.532929	0.506622
	3	10	20	20	gbdt	0.8	0.2	0.471148	0.46239	0.445599
	4	13	5	20	gbdt	1	0.2	0.440037	0.443428	0.422297
	5	13	15	4	gbdt	1	0.1	0.526395	0.527159	0.504191
	6	13	18	10	gbdt	1	0.1	0.527909	0.525642	0.506119

	7	13	17	5	gbdt	1	0.1	0.527124	0.527918	0.507209
	8	15	20	20	gbdt	1	0.2	0.476225	0.476348	0.457502
	9	15	15	11	gbdt	1	0.1	0.528008	0.525259	0.505029
Boosted Tree(Catboost)	Iteration	ber of vari	verbose	max_depth	iterations	earning rate		Train	Test	oot
	1 (default)	10	0	2	5	none		0.497639	0.499842	0.479715
	2	10	0	4	10	none		0.521369	0.520262	0.499162
	3	10	0	3	5	none		0.495309	0.500939	0.478877
	4	10	1	2	5	0.2		0.510109	0.517737	0.487846
	5	13	1	2	5	1		0.497729	0.50089	0.480721
	6	13	0	3	7	1		0.511754	0.511144	0.487678
	7	13	0	2	10	None		0.507373	0.516481	0.488097
	8	15	0	2	8	None		0.500153	0.500604	0.481643
Boosted Tree(XGB)	Iteration	ber of vari	max_depth	_estimato	booster	eta	_child_we	Train	Test	oot
	1 (default)	10	2	5	gbtree	0.3	1	0.496826	0.495903	0.477871
	2	10	4	10	gbtree	0.3	1	0.517816	0.521845	0.496815
	3	10	8	15	gbtree	0.3	10	0.529238	0.526027	0.506287
	4	10	5	15	dart	0.3	1	0.526727	0.528323	0.506203
	5	13	5	15	dart	0.3	1	0.530294	0.52109	0.506538
	6	13	3	15	gbtree	0.2	1	0.512629	0.51557	0.490612
	7	15	5	15	gbtree	0.3	1	0.526395	0.529384	0.505868

- I adjust the hyperparameters to find the best model. After tests, I found decision tree and boosted tree perform relatively well. I decided to go with boosted tree(LGBM). As the table, in iterations 6 and 7, I changed hyperparameters *n_estimators* from 18 to 17 and *n_estimators* from 10 to 5 which resulted in changing the model from overfitting to well-performance with 0.507209 oot. Based on the definition of overfitting and model performance, I decided the final model would be boosted tree(LGBM) with 13 variables and the model's *num_leaves* is 17, n_estimators is 5, *boosting_type* is gbdt, *subsample* is 1, the *learning rate* is 0.1.

6. Summary of results.

In the table, we have conducted 3 results tables (trn, tst, oot) for my final model which is **boosted tree(LGBM)** with 13 variables and the model's hyperparameters is:

Number of variables	num_leaves	n_estimators	boosting_type	subsample	learning_rate
13	17	5	gbdt	1	0.1

And result is:

Train	Test	oot
0.527124	0.527918	0.507209

And we are using these 13 variables below:

1. fulladdress_day_since
2. name_dob_count_30
3. address_unique_count_for_name_homephone_60
4. fulladdress_unique_count_for_dob_homephone_3
5. address_unique_count_for_homephone_name_dob_30
6. address_unique_count_for_ssn_name_dob_14
7. address_day_since
8. address_count_14
9. address_count_7
10. address_count_0_by_30
11. address_unique_count_for_homephone_name_dob_60
12. fulladdress_count_0_by_30
13. address_unique_count_for_ssn_zip5_60

Trainning	#Records		#Good		#Bads		Fraud Rate					
	583454		575093		8361		0.01433018					
	Bin Statistics					Cumulative Statistics						
Population Bin %	#Records	#Good	#Bads	%Good	%Bads	Total # Record	Cumulative Goods	Cumulative Bads	%Good	%Bads (FDR)	KS	FPR
1	5835	1614	4221	27.66067	72.33933	5835	1614	4221	0.28065	50.48439	50.20374	0.382374
2	5834	5697	137	97.6517	2.348303	11669	7311	4358	1.271273	52.12295	50.85168	1.677604
3	5835	5760	75	98.71465	1.285347	17504	13071	4433	2.27285	53.01997	50.74712	2.948568
4	5834	5793	41	99.29722	0.702777	23338	18864	4474	3.280165	53.51035	50.23018	4.216361
5	5835	5805	30	99.48586	0.514139	29173	24669	4504	4.289567	53.86915	49.57959	5.477131
6	5834	5784	50	99.14296	0.857045	35007	30453	4554	5.295317	54.46717	49.17185	6.687088
7	5835	5794	41	99.29734	0.702656	40842	36247	4595	6.302807	54.95754	48.65473	7.888357
8	5834	5789	45	99.22866	0.77134	46676	42036	4640	7.309426	55.49575	48.18633	9.059483
9	5835	5777	58	99.006	0.994002	52511	47813	4698	8.31396	56.18945	47.87549	10.17731
10	5834	5792	42	99.28008	0.719918	58345	53605	4740	9.321101	56.69178	47.37068	11.30907
11	5835	5805	30	99.48586	0.514139	64180	59410	4770	10.3305	57.05059	46.72009	12.45493
12	5834	5798	36	99.38293	0.617072	70014	65208	4806	11.33869	57.48116	46.14247	13.56804
13	5835	5805	30	99.48586	0.514139	75849	71013	4836	12.34809	57.83997	45.49188	14.68424
14	5835	5784	51	99.12596	0.874036	81684	76797	4887	13.35384	58.44995	45.09611	15.71455
15	5834	5793	41	99.29722	0.702777	87518	82590	4928	14.36116	58.94032	44.57916	16.75933
16	5835	5793	42	99.28021	0.719794	93353	88383	4970	15.36847	59.44265	44.07418	17.7833
17	5834	5798	36	99.38293	0.617072	99187	94181	5006	16.37666	59.87322	43.49657	18.81362
18	5835	5793	42	99.28021	0.719794	105022	99974	5048	17.38397	60.37555	42.99158	19.80468
19	5834	5789	45	99.22866	0.77134	110856	105763	5093	18.39059	60.91377	42.52318	20.76635
20	5835	5791	44	99.24593	0.75407	116691	111554	5137	19.39756	61.44002	42.04246	21.71579

Testing	#Records		#Good		#Bads		Fraud Rate					
	250053		246407		3646		0.014580909					
	Bin Statistics					Cumulative Statistics						
Population Bin %	#Records	#Good	#Bads	%Good	%Bads	Total # Record	Cumulative Goods	Cumulative Bads	%Good	%Bads (FDR)	KS	FPR
1	2501	674	1827	26.94922	73.05078	2501	674	1827	0.273531	50.10971	49.83618	0.368911
2	2500	2449	51	97.96	2.04	5001	3123	1878	1.267415	51.5085	50.24109	1.662939
3	2501	2469	32	98.72051	1.279488	7502	5592	1910	2.269416	52.38618	50.11676	2.927749
4	2500	2485	15	99.4	0.6	10002	8077	1925	3.27791	52.79759	49.51968	4.195844
5	2501	2486	15	99.40024	0.59976	12503	10563	1940	4.28681	53.209	48.92219	5.444845
6	2500	2475	25	99	1	15003	13038	1965	5.291246	53.89468	48.60343	6.635115
7	2501	2481	20	99.20032	0.79968	17504	15519	1985	6.298117	54.44323	48.14511	7.818136
8	2500	2485	15	99.4	0.6	20004	18004	2000	7.306611	54.85464	47.54802	9.002
9	2501	2488	13	99.48021	0.519792	22505	20492	2013	8.316322	55.21119	46.89487	10.17983
10	2500	2474	26	98.96	1.04	25005	22966	2039	9.320352	55.9243	46.60395	11.26336
11	2501	2485	16	99.36026	0.639744	27506	25451	2055	10.32885	56.36314	46.03429	12.38491
12	2500	2480	20	99.2	0.8	30006	27931	2075	11.33531	56.91168	45.57637	13.46072
13	2501	2484	17	99.32027	0.679728	32507	30415	2092	12.3434	57.37795	45.03455	14.53872
14	2500	2484	16	99.36	0.64	35007	32899	2108	13.35149	57.81679	44.4653	15.60674
15	2501	2478	23	99.08037	0.919632	37508	35377	2131	14.35714	58.44761	44.09047	16.60113
16	2500	2482	18	99.28	0.72	40008	37859	2149	15.36442	58.94131	43.57689	17.61703
17	2501	2487	14	99.44022	0.559776	42509	40346	2163	16.37372	59.32529	42.95156	18.6528
18	2501	2487	14	99.44022	0.559776	45010	42833	2177	17.38303	59.70927	42.32624	19.67524
19	2500	2479	21	99.16	0.84	47510	45312	2198	18.38909	60.28524	41.89616	20.6151
20	2501	2482	19	99.2403	0.759696	50011	47794	2217	19.39636	60.80636	41.41	21.55796

OOT	#Records		#Good		#Bads		Fraud Rate					
	166493		164107		2386		0.014330933					
	Bin Statistics					Cumulative Statistics						
Population Bin %	#Records	#Good	#Bads	%Good	%Bads	Total # Record	Cumulative Goods	Cumulative Bads	%Good	%Bads (FDR)	KS	FPR
1	1665	509	1156	30.57057	69.42943	1665	509	1156	0.310163	48.44929	48.13912	0.440311
2	1665	1635	30	98.1982	1.801802	3330	2144	1186	1.306465	49.70662	48.40016	1.807757
3	1665	1641	24	98.55856	1.441441	4995	3785	1210	2.306422	50.71249	48.40607	3.128099
4	1665	1648	17	98.97898	1.021021	6660	5433	1227	3.310645	51.42498	48.11433	4.427873
5	1665	1655	10	99.3994	0.600601	8325	7088	1237	4.319133	51.84409	47.52496	5.729992
6	1665	1656	9	99.45946	0.540541	9990	8744	1246	5.328231	52.22129	46.89306	7.017657
7	1665	1654	11	99.33934	0.660661	11655	10398	1257	6.33611	52.68231	46.3462	8.272076
8	1664	1653	11	99.33894	0.661058	13319	12051	1268	7.34338	53.14334	45.79996	9.503943
9	1665	1655	10	99.3994	0.600601	14984	13706	1278	8.351868	53.56245	45.21058	10.72457
10	1665	1654	11	99.33934	0.660661	16649	15360	1289	9.359747	54.02347	44.66372	11.91621
11	1665	1656	9	99.45946	0.540541	18314	17016	1298	10.36884	54.40067	44.03183	13.1094
12	1665	1655	10	99.3994	0.600601	19979	18671	1308	11.37733	54.81978	43.44245	14.27446
13	1665	1654	11	99.33934	0.660661	21644	20325	1319	12.38521	55.2808	42.89559	15.4094
14	1665	1658	7	99.57958	0.42042	23309	21983	1326	13.39553	55.57418	42.17865	16.57843
15	1665	1649	16	99.03904	0.960961	24974	23632	1342	14.40036	56.24476	41.8444	17.60954
16	1665	1656	9	99.45946	0.540541	26639	25288	1351	15.40946	56.62196	41.2125	18.71799
17	1665	1650	15	99.0991	0.900901	28304	26938	1366	16.4149	57.25063	40.83573	19.72035
18	1665	1649	16	99.03904	0.960961	29969	28587	1382	17.41973	57.92121	40.50147	20.68524
19	1665	1653	12	99.27928	0.720721	31634	30240	1394	18.427	58.42414	39.99714	21.69297
20	1665	1650	15	99.0991	0.900901	33299	31890	1409	19.43244	59.05281	39.62036	22.63307

The OOT (0.507) shows that the boosted tree(LGBM) model can eliminate about 50% of the fraud by declining only about 3% of the applications.