Joshua Chang
Professor Coggeshall

Homework 6: Document Project 1

Section 1: Description of Data

This dataset is about Application Identity Theft Data. The data is about credit card applications spanning from January 1, 2017 to December 31, 2017. There are 10 fields with 1,000,000 records.

Numerical Table:

Field Name	% Populated	Min	Max	Mean	Standard Deviation	% Zero
date	100%	2017-01-01	2017-12-31	N/A	N/A	0

Categorical Table:

Field Name	% Populated	# of unique values	Most Common Field Value
records	100%	1,000,000	N/A
ssn	100%	835,819	999999999
firstname	100%	78,136	EAMSTRMT
lastname	100%	177,001	ERJSAXA
address	100%	828,774	123 MAIN ST
zip5	100%	26,370	68138
dob	100%	42,673	1907-06-26
homephone	100%	28,244	(999) 999-9999
fraud_label	100%	2	0

Section 2: Data Cleaning

This dataset has frivolous fields, which are filled-in previously missing values that throw off the model. We performed data cleaning by transforming these frivolous values and changing them to something that doesn't confuse the model. For example, we fixed frivolous values such as frivolous addresses like '123 Main St' with a unique string such as that row's record number. The same goes with other fields such as ssn, date of birth, or phone number. This process helps our machine learning algorithms produce more accurate results later on.

Section 3: Variable Creation

The modes of identity fraud include identity theft, identity manipulation, and synthetic identity:

- Identity theft occurs when a fraudster uses a real but stolen identity that's different from their own. Variables such as SSN, DOB, or Name associated with multiple contact points (e.g. address, phone, email) are used to catch identity theft. Also, velocity around PII elements are indicators of identity theft.
- Identity manipulation occurs when the fraudster slightly changes their own identity. Looking for small changes to variables such as SSN, DOB, Name, or other indications of slight systematic variations in PII elements are key to catch identity manipulation.
- Synthetic identity occurs when the fraudster makes up a completely fabricated identity. Variables that link all the PII elements to be associated with multiple different identities are helpful in identifying synthetic identities.

Description of variables	# Variables Created
Original fields from the dataset excluding 'record' and 'fraud label'	8
Date of week target encoded (average fraud percentage of that day)	1
New entities combining/concatenating different original fields	9
Days since Variables: # days since an application with that entity has been seen	23
Velocity: # records with the same entity over the last 0, 1, 3, 7, 14, 30 days	138
Relative Velocity: # applications with that group/entity seen in the recent past divided by the # of applications with that same group seen in the past 1, 3, 7, 14, 30 days	184
Frequency Variables: This set of variables include the number of applications submitted with this various combinations between the fields and the combinations of fields over the past 0, 1, 3, 7, 14, 30, 60 days	3542
Risk Variable: The likelihood of fraud for any day of the week	1

Maximum Indicator Variables: This set of variables outputs the maximum number of times an attribute shows up over the past 0, 1, 3, 7, 14, 30 days	92
Age Indicator Variables: This set of variables include maximum, mean, and minimum age when application was submitted	69

Section 4: Feature Selection

After creating our variables, we want to perform feature selection. The problem is that the more variables we have in our model, the more our model suffers from the curse of dimensionality. So we used wrappers to filter out variables and therefore reduce dimensionality and complexity of our models. In a given project, sometimes halfway through we may discover some variables that can't be used and were improperly made, which was the case here. To fix the issue and get rid of the improper variables in this project, we would rerun our code in the previous section and skip over the 'max indicators' program. Due to time constraints, I did not do so. However, for section 5 and onward, the proper variables were used as they were produced from revised code files provided by the professor. The following displays variables produced with 'max indicators' code in wrapper order.

	wrapper order	variable	filter score
0	1	max_count_by_address_30	0.359215
1	2	max_count_by_ssn_dob_7	0.228401
2	3	max_count_by_homephone_3	0.224757
3	4	max_count_by_fulladdress_30	0.359914
4	5	zip5_count_3	0.224706
5	6	max_count_by_ssn_dob_30	0.240836
6	7	max_count_by_homephone_7	0.232235
7	8	fulladdress_count_0_by_30	0.290722
8	9	max_count_by_fulladdress_homephone_30	0.249724
9	10	ssn_dob_day_since	0.228626
10	11	max_count_by_address_7	0.343335
11	12	address_day_since	0.334140
12	13	fulladdress_day_since	0.333269
13	14	max_count_by_fulladdress_3	0.329538
14	15	max_count_by_address_3	0.329445
15	16	address_count_14	0.322436
16	17	fulladdress_count_14	0.321953
17	18	max_count_by_address_1	0.315332
18	19	max_count_by_fulladdress_1	0.315253
19	20	address_count_7	0.301735

Section 5: Preliminary Models Exploration

Model	Parameters								Number of Variables		
Logistic Regression	**Number of Variables**	**max_iter**						**Train**	**Test**	**OOT**	**Fit**
1	5	20						0.477	0.484	0.466	underfit
2	10	20						0.487	0.491	0.474	
3	15	20						0.484	0.476	0.467	
Decision Tree	**Number of Variables**	**max_depth**	**min_samples_leaf**	**min_samples_split**	**max_features**			**Train**	**Test**	**OOT**	**Fit**
1	10	2	500	1000	none			0.461	0.461	0.444	underfit
2	10	50	250	500	none			0.531	0.523	0.503	
3	10	50	300	550	none			0.526	0.521	0.497	
4	10	100	2	5	none			0.543	0.512	0.495	overfit
5	15	50	250	500	5			0.523	0.514	0.496	
6	15	50	300	550	5			0.520	0.518	0.496	
7	20	50	250	500	8			0.526	0.521	0.498	
8	20	100	300	700	10			0.526	0.519	0.498	
Random Forest	**Number of Variables**	**max_depth**	**min_samples_leaf**	**min_samples_split**	**max_features**	**n_estimator**		**Train**	**Test**	**OOT**	**Fit**
1	10	2	500	1000	8	3		0.473	0.473	0.459	underfit
2	10	5	30	50	3	5		0.519	0.517	0.495	
3	10	10	25	45	6	10		0.529	0.527	0.505	
4	10	15	20	40	10	15		0.535	0.525	0.502	
5	15	10	25	45	6	10		0.528	0.525	0.503	
6	15	30	10	30	10	100		0.542	0.523	0.501	
7	20	10	25	45	6	10		0.530	0.523	0.505	
8	20	30	5	50	10	100		0.544	0.515	0.501	overfit
LGBM	**Number of Variables**	**n_estimators**	**max_depth**	**num_leaves**				**Train**	**Test**	**OOT**	**Fit**
1	10	2	2	2				0.464	0.452	0.444	underfit
3	10	20	2	2				0.512	0.513	0.489	
4	10	50	3	4				0.516	0.518	0.494	
5	10	50	6	10				0.528	0.527	0.504	
6	15	300	4	5				0.526	0.532	0.507	
7	15	500	6	10				0.533	0.520	0.507	
8	20	100	4	8				0.528	0.526	0.503	
9	20	75	6	10				0.527	0.528	0.505	
10	20	500	100	50				0.536	0.516	0.505	overfit
Neural Networks	**Number of Variables**	**hidden_layer_sizes**	**activation**	**alpha**	**learning_rate**	**solver**	**learning_rate_init**	**Train**	**Test**	**OOT**	**Fit**
1	10	(5)	logistic	0.1	constant	adam	0.01	0.494	0.496	0.478	underfit
3	10	(20,20,20)	relu	0.01	adaptive	lbfgs	0.01	**0.528**	**0.527**	0.505	
4	15	(5)	relu	0.01	adaptive	lbfgs	0.01	0.521	0.523	0.500	
5	15	(10,10)	relu	0.1	adaptive	lbfgs	0.0001	0.526	0.529	0.505	
6	20	(10,10)	logistic	0.01	adaptive	lbfgs	0.0001	0.516	0.516	0.496	
7	20	(20,20,20)	relu	0.01	constant	lbfgs	0.01	0.529	0.524	0.507	
GBC	**Number of Variables**	**n_estimators**	**max_depth**					**Train**	**Test**	**OOT**	**Fit**
1	10	2	2					0.487	0.483	0.469	underfit
2	10	10	4					0.524	0.513	0.499	
3	10	20	6					0.529	0.524	0.505	
4	10	1000	6					0.544	0.513	0.497	overfit
5	15	30	4					0.523	0.523	0.502	
6	15	50	6					0.527	0.529	0.504	
7	15	100	6					0.532	0.525	0.507	
8	20	10	6					0.526	0.520	0.501	
9	20	40	6					0.526	0.531	0.503	
XGB	**Number of Variables**	**n_estimators**	**max_depth**					**Train**	**Test**	**OOT**	**Fit**
1	10	2	2					0.492	0.490	0.474	underfit
2	10	10	4					0.519	0.519	0.497	
3	10	20	6					0.529	0.525	0.507	
4	15	30	4					0.528	0.525	0.505	
5	15	60	6					0.534	0.524	0.505	
6	20	50	6					0.531	0.528	0.506	
7	20	2000	5					0.544	0.518	0.498	overfit

Section 6: Summary of Results

Final Model Selection:

Model	# of Variables	n_estimators	max_depth	Train	Test	OOT
XGB	10	20	6	0.529	0.525	0.507

My final model uses an XGB, or xg boost, architecture, with n_estimators set to 20 and a max_depth of 6. The following is the list of the final variables used in this model:

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'

The 3 Results Tables for My Final Model

Training:

Training	# Records		# Goods		# Bads		Fraud Rate						
	583454		575058		8396		0.01439016615						
	Bin Statistics					Cumulative Statistics							
Population Bin %	# Records	# Goods	# Bads	% Goods	% Bads	Total # Records	Cumulative Goods	Cumulative Bads	% Goods	% Bads (FDR)	KS	FPR	
1	5835	1613	4222	27.64%	72.36%	5835	1613	4222	0.28%	50.29%	50.01	0.38	
2	5834	5698	136	97.67%	2.33%	11669	7311	4358	1.27%	51.91%	50.63	1.68	
3	5835	5761	74	98.73%	1.27%	17504	13072	4432	2.27%	52.79%	50.51	2.95	
4	5834	5776	58	99.01%	0.99%	23338	18848	4490	3.28%	53.48%	50.20	4.20	
5	5835	5797	38	99.35%	0.65%	29173	24645	4528	4.29%	53.93%	49.64	5.44	
6	5834	5784	50	99.14%	0.86%	35007	30429	4578	5.29%	54.53%	49.23	6.65	
7	5835	5793	42	99.28%	0.72%	40842	36222	4620	6.30%	55.03%	48.73	7.84	
8	5834	5794	40	99.31%	0.69%	46676	42016	4660	7.31%	55.50%	48.20	9.02	
9	5835	5803	32	99.45%	0.55%	52511	47819	4692	8.32%	55.88%	47.57	10.19	
10	5834	5790	44	99.25%	0.75%	58345	53609	4736	9.32%	56.41%	47.09	11.32	
11	5835	5797	38	99.35%	0.65%	64180	59406	4774	10.33%	56.86%	46.53	12.44	
12	5834	5801	33	99.43%	0.57%	70014	65207	4807	11.34%	57.25%	45.91	13.57	
13	5835	5790	45	99.23%	0.77%	75849	70997	4852	12.35%	57.79%	45.44	14.63	
14	5835	5800	35	99.40%	0.60%	81684	76797	4887	13.35%	58.21%	44.85	15.71	
15	5834	5791	43	99.26%	0.74%	87518	82588	4930	14.36%	58.72%	44.36	16.75	
16	5835	5790	45	99.23%	0.77%	93353	88378	4975	15.37%	59.25%	43.89	17.76	
17	5834	5795	39	99.33%	0.67%	99187	94173	5014	16.38%	59.72%	43.34	18.78	
18	5835	5799	36	99.38%	0.62%	105022	99972	5050	17.38%	60.15%	42.76	19.80	
19	5834	5788	46	99.21%	0.79%	110856	105760	5096	18.39%	60.70%	42.30	20.75	
20	5835	5784	51	99.13%	0.87%	116691	111544	5147	19.40%	61.30%	41.91	21.67	

Testing:

Testing	# Records		# Goods		# Bads		Fraud Rate					
	250053		246442		3611		0.01444093852					
	Bin Statistics					Cumulative Statistics						
Population Bin %	# Records	# Goods	# Bads	% Goods	% Bads	Total # Records	Cumulative Goods	Cumulative Bads	% Goods	% Bads (FDR)	KS	FPR
1	2501	681	1820	27.23%	72.77%	2501	681	1820	0.28%	50.40%	50.13	0.37
2	2500	2446	54	97.84%	2.16%	5001	3127	1874	1.27%	51.90%	50.63	1.67
3	2501	2467	34	98.64%	1.36%	7502	5594	1908	2.27%	52.84%	50.57	2.93
4	2500	2475	25	99.00%	1.00%	10002	8069	1933	3.27%	53.53%	50.26	4.17
5	2501	2485	16	99.36%	0.64%	12503	10554	1949	4.28%	53.97%	49.69	5.42
6	2500	2485	15	99.40%	0.60%	15003	13039	1964	5.29%	54.39%	49.10	6.64
7	2501	2481	20	99.20%	0.80%	17504	15520	1984	6.30%	54.94%	48.65	7.82
8	2500	2483	17	99.32%	0.68%	20004	18003	2001	7.31%	55.41%	48.11	9.00
9	2501	2488	13	99.48%	0.52%	22505	20491	2014	8.31%	55.77%	47.46	10.17
10	2500	2472	28	98.88%	1.12%	25005	22963	2042	9.32%	56.55%	47.23	11.25
11	2501	2484	17	99.32%	0.68%	27506	25447	2059	10.33%	57.02%	46.69	12.36
12	2500	2485	15	99.40%	0.60%	30006	27932	2074	11.33%	57.44%	46.10	13.47
13	2501	2492	9	99.64%	0.36%	32507	30424	2083	12.35%	57.68%	45.34	14.61
14	2500	2478	22	99.12%	0.88%	35007	32902	2105	13.35%	58.29%	44.94	15.63
15	2501	2477	24	99.04%	0.96%	37508	35379	2129	14.36%	58.96%	44.60	16.62
16	2500	2477	23	99.08%	0.92%	40008	37856	2152	15.36%	59.60%	44.23	17.59
17	2501	2485	16	99.36%	0.64%	42509	40341	2168	16.37%	60.04%	43.67	18.61
18	2501	2477	24	99.04%	0.96%	45010	42818	2192	17.37%	60.70%	43.33	19.53
19	2500	2479	21	99.16%	0.84%	47510	45297	2213	18.38%	61.28%	42.90	20.47
20	2501	2486	15	99.40%	0.60%	50011	47783	2228	19.39%	61.70%	42.31	21.45

OOT:

OOT	# Records		# Goods		# Bads		Fraud Rate					
	166493		164107		2386		0.01433093283					
	Bin Statistics					Cumulative Statistics						
Population Bin %	# Records	# Goods	# Bads	% Goods	% Bads	Total # Records	Cumulative Goods	Cumulative Bads	% Goods	% Bads (FDR)	KS	FPR
1	1665	508	1157	30.51%	69.49%	1665	508	1157	0.31%	48.49%	48.18	0.44
2	1665	1639	26	98.44%	1.56%	3330	2147	1183	1.31%	49.58%	48.27	1.81
3	1665	1637	28	98.32%	1.68%	4995	3784	1211	2.31%	50.75%	48.45	3.12
4	1665	1646	19	98.86%	1.14%	6660	5430	1230	3.31%	51.55%	48.24	4.41
5	1665	1653	12	99.28%	0.72%	8325	7083	1242	4.32%	52.05%	47.74	5.70
6	1665	1657	8	99.52%	0.48%	9990	8740	1250	5.33%	52.39%	47.06	6.99
7	1665	1656	9	99.46%	0.54%	11655	10396	1259	6.33%	52.77%	46.43	8.26
8	1664	1645	19	98.86%	1.14%	13319	12041	1278	7.34%	53.56%	46.23	9.42
9	1665	1657	8	99.52%	0.48%	14984	13698	1286	8.35%	53.90%	45.55	10.65
10	1665	1656	9	99.46%	0.54%	16649	15354	1295	9.36%	54.27%	44.92	11.86
11	1665	1653	12	99.28%	0.72%	18314	17007	1307	10.36%	54.78%	44.41	13.01
12	1665	1654	11	99.34%	0.66%	19979	18661	1318	11.37%	55.24%	43.87	14.16
13	1665	1655	10	99.40%	0.60%	21644	20316	1328	12.38%	55.66%	43.28	15.30
14	1665	1653	12	99.28%	0.72%	23309	21969	1340	13.39%	56.16%	42.77	16.39
15	1665	1651	14	99.16%	0.84%	24974	23620	1354	14.39%	56.75%	42.35	17.44
16	1665	1653	12	99.28%	0.72%	26639	25273	1366	15.40%	57.25%	41.85	18.50
17	1665	1653	12	99.28%	0.72%	28304	26926	1378	16.41%	57.75%	41.35	19.54
18	1665	1648	17	98.98%	1.02%	29969	28574	1395	17.41%	58.47%	41.05	20.48
19	1665	1653	12	99.28%	0.72%	31634	30227	1407	18.42%	58.97%	40.55	21.48
20	1665	1660	5	99.70%	0.30%	33299	31887	1412	19.43%	59.18%	39.75	22.58

Description of Results:

We see that we can achieve a fraud detection rate at 3% of 0.529, 0.525, and 0.507 for training, testing, and out of time, respectively. That means that our model is able to capture 50.7% of all the fraud in the top 3%. This means that our model allows us to reject only 3% of the applications and catch 50.7% of the fraud in those rejected applications. As mentioned previously, our final model choice here is a boosted tree, particularly using the xgboost architecture. In terms of hyperparameters set, we used 20 estimators and set our max depth to 6.