Application Fraud – Credit card

1. Executive summary:

Application fraud is a type of fraud that involves using fake or stolen identities to apply for financial services or products. This can cause businesses a lot of trouble, and can also damage their reputation. Machine learning can help identify fraudulent applications, which can help prevent any losses and damage to reputation.

The report explains how supervised learning techniques were used to identify instances of application fraud, and how this was evaluated. A dataset of 1 million entries was used in the evaluation, and 10 different attributes were examined. The most important attributes were identified using the Kolmogorov-Smirnov (KS) score. Different modeling techniques were then tested, including logistic regression, decision trees, random forest, gradient boosting, and neural networks.

The challenge in detecting fraud is that fraudsters always come up with new ways to cheat, which makes it hard to build a good detection model. Additionally, there can be problems with data availability or quality, which can impact the model's accuracy.

The LightGBM model was found to have a higher fraud detection rate than the other models at 3% FDR.

2. Data Quality Report:

2.1 Summary Tables:

(1) Numerical Table

Field Name	% Populated	Min	Max	Mean	Stdev	% Zero
Date	100%	2017-01-01	2017-12-31	-	-	0.00
dob	100%	1900-01-01	2016-10-31	-	-	0.00

(2) Categorical Table

1. Categorical Table

Field Name	**%Populated**	**#Unique Values**	**Most Common Values**
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
FraudLabel	100.00	2	0

2.2 Data Description:

a. ***Record***: Record number. Ordinal unique positive integer for each record from 1 to 1000000

b. ***Date***: Date column specifying a span of 365 days throughout 2017 starting from 1st Jan – 31st Dec. Each record specifies the details of each application across these dates in 2017

c. ***SSN***: Social Security Number [SSN] field. Nominal positive integer used for determining the identity of a person.
There are 835,819 unique SSNs in the dataset while '999999999' seems to be the most commonly used one with 16,935 records. Below is the distribution of the top commonly used SSNs with and without '999999999'. A logarithmic y-axis has been used to fit the data on a linear graph.

d. ***Firstname***: First names used in the application. There are 78,136 unique first names in the application. The most commonly used name is 'EAMSTRMT' with 12,658 records.

e. ***lastname***: First names used in the application. There are 177,001 unique first names in the application. The most commonly used name is 'ERJSAXA' with 8,580 records.

f. ***address***: Addresses used in the application. There are 828,774 unique first names in the application. The most commonly used name is '123 MAIN ST' with 1,079 records

g. ***zip5***: Zipcodes used in the application. There are 26,370 unique first names in the application. The most commonly used name is '68138' with 823 records.

h. ***dob***: Date of birth of applicants in the application. There are 42,673 unique dobs in the application '1907-06-26' as the most common record [126,568 records]. After removing this date, the minimum dob is '1964-03-18'

i. ***homephone***: Home phone numbers used in the application. There are 28,244 unique first names in the application. The most commonly used number is '9999999999' with 78,512 records.

j. ***fraudlabel***: Fraud = 0 (no fraud label). Fraud = 1 (fraud label) Distribution count: Fraud = 0 [985,607].

3. **Data cleaning:**

It is required to clean the data and address any missing, irrelevant, inaccurate, or corrupted data points. Data needs to be modified, replaced, or imputed to ensure the accuracy and integrity of the model.

3.1 Handling Frivolous Values:

Frivolous fields are data fields that do not add any value to the model and may even harm model's performance. These fields can include irrelevant, incomplete, or inconsistent data that can cause noise or bias in the dataset, and result in incorrect predictions.

The fields zip5, ssn, homephone, address, and dob contained improper data. These were corrected as follows:

1. *zip5:* Incorrect zip codes with less than 5 digits (e.g. 1362) were corrected by adding leading zeros

2. *ssn:*
   - 16,935 SSNs were listed as 999999999, which were assumed to be missing data. These were replaced with the value of the corresponding RECORD number
   - Short SSNs with less than 9 digits were corrected by adding leading zeros

3. *homephone*:
   - 78,512 homephone entries were listed as 9999999999 and were replaced with the negative value of the corresponding RECORD number
   - Phone numbers with less than 10 digits were corrected by adding leading zeros

4. *address*: 1079 entries listed as "123 MAIN ST" were assumed to be missing and were replaced with RECORD number as string

5. *dob*: 126,568 entries listed as 19070626 were assumed to be default values for missing or incorrect data and were replaced with the value of the RECORD column

4. **Variable creation:**

Feature engineering involves transforming existing data features into something that machine learning algorithms can use to better understand the data. In our project, we employed various techniques to extract features from the data.

Target encoding is a critical feature engineering technique used to transform categorical variables into continuous variables that can be used as input features in machine learning models. This technique replaces categorical variables with statistical measures like the mean of the target variable for each category. It's particularly useful for binary classification and regression problems and creates m-1 new variables for multiclass classification, where m is the number of classes.

Statistical smoothing is another technique used to smooth or regularize data by applying a statistical function to estimate the underlying trend or pattern in the data. Its aim is to reduce the noise in the data while preserving important features and patterns.

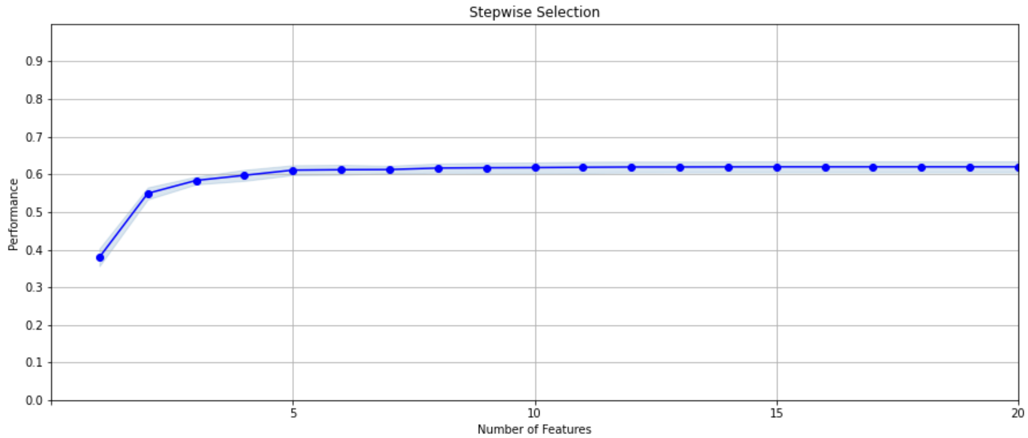
We also used fuzzy logic to create 4000 variables that include velocity, days-since, relative velocity, and entity counts variables. Velocity variables capture the frequency of encountering each entity or combination group over the past 0,1,3,7,14, and 30 days. Days-since variables record the number of days since the last encounter with each entity or combination group. Relative velocity variables indicate the relative velocity of the entity or combination group compared to its past encounters. Entity counts variables measure the number of entities or combination groups encountered. These variables help the machine learning algorithm to learn more about the data and extract insights that may not be apparent in the original data.

Description of variables	# of Variables created	Current records	Total Number of columns
Initial fields in the dataset including 'fraud_label' and 'record'--> Fields: 'record', 'date', 'ssn', 'firstname', 'lastname', 'address', 'zip5', 'dob', 'homephone','fraud_label'	10	record', 'date', 'ssn', 'firstname', 'lastname', 'address', 'zip5', 'dob', 'homephone', 'fraud_label'	10
Date of birth-DOB converted to datetime format 'dob_dt'	1		11
Age when apply field created to calculate the age of the applicant using the difference between date of applying and DOB- 'age_when_apply'	1		12
Day of Week Target Encoded- Average fraud occuring on that particular weekday	1		13
Risk in a given day of week - (risk average fraud % or probability of risk on that day) ('dow_risk')	1		14
New variables created by combining original records	9	name', 'fulladdress', 'name_dob', 'name_fulladdress', 'name_homephone', 'fulladdress_dob', 'fulladdress_homephone', 'dob_homephone', 'homephone_name_dob'	23
New variables created by combining ssn with few original records	9	'ssn_firstname', 'ssn_lastname', 'ssn_address', 'ssn_zip5', 'ssn_dob', 'ssn_homephone', 'ssn_name', 'ssn_fulladdress', 'ssn_name_dob'	32
Day since variables	23		55
Velocity variables - # of records with the same entity over the last {0,1,3,7,14,30} days	138		193
Relative velocity - Fraction of number and amount of transactions with the same acrd and the same merchant over the past 0 or 1 day out of the total number or the amount with the same card and merchant for 7,14,30 days	184		377
Count by Entity variables - number of unique records for a particular field	3542		3919
Maximum indicator variables - Maximum number of records, grouped by entities for 1,3,7,30 days	92		4011
Age indicator variables - Age of the applicant at the time of applying (current date - dob of the applicant)	69		4080
Total Number of Variables	**4080**		

Feature selection:

Feature selection is a crucial method to enhance the performance of machine learning models, particularly in high-dimensional data. As the complexity of models increases with the number of dimensions, fitting fewer dimensions becomes easier. Feature selection can also enhance the model's architecture. To simplify the selection process, variables are ranked based on their significance. In our fraud detection project, we initially employed filter feature selection by calculating the KS statistic through univariate tests, which helped us identify the top 13% of variables. We then employed the XG Boost model to do forward wrapper selection, which helped us pinpoint 20 variables with a fraud detection rate at a 0.03 cutoff. By using feature selection, we were able to work with the most pertinent variables and streamline our modeling process, resulting in faster runs and better efficiency.

wrapper order	variable	filter score
1	max_count_by_address_30	0.359215465
2	max_count_by_ssn_dob_7	0.228400837
3	max_count_by_homephone_3	0.224757436
4	zip5_count_1	0.221239028
5	max_count_by_fulladdress_30	0.359913969
6	max_count_by_name_30	0.222190696
7	max_count_by_homephone_7	0.232235291
8	max_count_by_ssn_dob_30	0.24083569
9	fulladdress_count_0_by_30	0.290722131
10	ssn_firstname_day_since	0.226427511
11	max_count_by_homephone_30	0.21593074
12	fulladdress_day_since	0.333268536
13	address_unique_count_for_ssn_zip5_60	0.289723617
14	max_count_by_fulladdress_homephone_30	0.249723749
15	address_count_30	0.332648157
16	max_count_by_address_7	0.343335432
17	address_day_since	0.334139944
18	max_count_by_fulladdress_3	0.329537708
19	max_count_by_address_3	0.329444706
20	address_count_14	0.32243628



5. <u>Preliminary models exploration</u>:

In order to find the best model for our task, we tested 10 different variables and various hyperparameters with a 3% false discovery rate. We started by using logistic regression as a baseline and then compared its performance with other nonlinear models such as Decision Tree, Random Forest, LightGBM, Neural Network, GBC, Catboost, and XGBoost. This allowed us to systematically evaluate the models, considering their complexity and the trade-off between bias and variance. It's important to note that the choice of hyperparameters and input variables can have a significant impact on the model's performance, so we need to carefully tune them and conduct a thorough analysis of the results. This approach can help us build more accurate and effective machine learning models.

Model	Parameter											Average FDR at 3%			OBSERVATIONS
Logistic Regression	Iteration	num_variables	penalty	C	solver	l1_ratio						Train	Test	OOT	
	1	10	l1	0.5	lbfgs	None						48.7	49.1	47.4	
	2	10	l2	0.8	lbfgs	0						48.9	48.7	47.4	
	3	20	l1	0.3	saga	None						48.7	48.3	47.1	
	4	10	elasticnet	0.7	liblinear	0.6						48.8	48.9	47.4	
	5	10	l2	1	lbfgs	None						49.1	48.3	47.6	Over-fitting
Decision Tree	Iteration	num_variables	criterion	max_depth	min_samples_split	splitter	min_samples_leaf					Train	Test	OOT	
	1	10	gini	None	2	best	1					46.1	45.7	44.4	Under-fitting
	2	10	gini	60	50	best	2					54	52.1	50	
	3	10	entropy	20	100	best	2					53.8	52.2	50.1	
	4	10	entropy	10	200	random	4					53	52.4	50.4	
	5	20	gini	100	50	best	2					54.6	51.6	50.2	Over-fitting
Random Forest	Iteration	num_variables	n_estimators	max_depth	min_samples_split	min_samples_leaf	max_features	criterion				Train	Test	OOT	
	1	10	100	None	2	1	4	gini				54.5	51.1	49.9	Over-fitting
	2	10	50	10	500	20	8	gini				52	52.4	50.7	
	3	10	50	20	500	10	6	entropy				53.4	52.2	50.4	
	4	15	10	2	1000	2	8	gini				47.8	48.4	46.4	Under-fitting
	5	20	100	100	500	2	8	gini				52.8	51.1	50.3	
Light Gbm	Iteration	num_variables	boosting_type	max_depth	n_estimators	num_leaves	subsample	colsample_bytree	learning_rate			Train	Test	OOT	
	1	10	gbdt	20	100	50	0.8	0.8	0.1			52.9	53.1	50.5	
	2	10	gbdt	10	100	50	0.8	1	0.2			49.3	49.2	46.5	Under-fitting
	3	20	gbdt	10	80	70	1	0.8	0.03			53.2	52.8	50.1	
	4	15	dart	50	100	30	0.6	0.8	0.1			51.5	52	50.5	
	5	15	dart	30	100	70	0.8	1	0.03			53.3	52.2	50.8	
Xgboost	Iteration	num_variables	booster	max_depth	n_estimators	tree_method	min_child_weight	subsample	colsample_bytree	eta	eval_metric	Train	Test	OOT	
	1	10	gbtree	6	5	auto	1	1	1	0.3	rmse	52.4	52.4	50.2	
	2	10	gbtree	10	50	hist	1	0.8	0.8	0.1	rmse	53.3	52.3	50.5	
	3	10	dart	10	50	auto	1	0.8	0.8	0.1	rmse	53.2	52.4	50.6	
	4	10	gbtree	20	50	auto	1	1	1	0.3	mae	54	52.3	50.1	
	5	10	gbtree	30	1000	hist	1	0.8	1	0.1	mae	54.1	51.5	49.9	Over-fitting
CatBoost	Iteration	num_variables	depth	bootstrap_type	l2_leaf_reg	grow_policy	learning_rate	random_state	Iterations	min_data_in_leaf		Train	Test	OOT	
	1	10	6	Bayesian	3	SymmetricTree	0.03	None	1000	1		52.8	52.7	50.4	
	2	10	5	Bayesian	12	Depthwise	0.02	2	500	2		52.1	52.8	50.2	
	3	10	7	Bernoulli	12	SymmetricTree	0.01	32	5	2		51.7	50.8	49.9	
	4	20	10	Bayesian	8	SymmetricTree	0.03	2	500	4		53.1	51.5	50.3	
	5	20	20	Bayesian	10	SymmetricTree	0.01	2	500	10		53.1	52.8	50	Over-fitting
Neural Network	Iteration	num_variables	activation	solver	hidden_layer_size	max_iter	learning_rate	alpha				Train	Test	OOT	
	1	20	relu	adam	2	200	constant	0.1				50.4	49.9	48.5	Over-fitting
	2	20	relu	sgd	10	300	adaptive	0.2				51.4	51.5	49.2	
	3	20	logistic	adam	10	100	adaptive	0.03				51.4	51.6	49.1	
	4	15	relu	adam	10	500	constant	0.3				50.6	51.1	49.5	
	5	10	logistic	sgd	10	500	adaptive	0.002				51.4	51.6	49.1	

6. **<u>Summary of results</u>:**

After conducting some initial exploratory analysis, we chose the LGBM model as the final model because it had the highest average FDR for testing and relatively high FDR for the OOT, training, and test datasets. Additionally, the standard deviation was small, indicating that the model's performance was stable. The data was split into training, testing, and OOT sets with similar fraud rates of around 0.014. The LGBM model was able to identify around 53% of fraudulent cases in the training set, 52.3% in the test set, and 51% in the OOT set, while only declining approximately 3% of applications. We summarized the results in a table, including the number of records, the percentage of fraud caught, cumulative KS and FPR values, and the top 20 percentile bins for all three datasets.

Training Results:

Train	# Record		# Goods		# Bads		Fraud Rate					
		583,454		575,000		8,454	0.01422					
	Bin Statistics					Cumulative Statistics						
Population Bin %	# Records	# Goods	# Bads	% Goods	% Bads	Total # Records	Cumulative Goods	Cumulative Bads	% Goods	% Bads (FDR)	KS	FPR
0	0	0	0	0	0	0	0	0	0	0	0	0
1	5835	1952	3883	33.45	66.55	5835	1952	3883	0.34	45.93	45.59	0.50
2	5834	5552	282	95.17	4.83	11669	7504	4165	1.31	49.27	47.96	1.80
3	5835	5660	175	97.00	3.00	17504	13164	4340	2.29	51.34	49.05	3.03
4	5834	5795	39	99.33	0.67	23338	18959	4379	3.30	51.80	48.50	4.33
5	5835	5795	40	99.31	0.69	29173	24754	4419	4.31	52.27	47.97	5.60
6	5834	5795	39	99.33	0.67	35007	30549	4458	5.31	52.73	47.42	6.85
7	5835	5797	38	99.35	0.65	40842	36346	4496	6.32	53.18	46.86	8.08
8	5834	5801	33	99.43	0.57	46676	42147	4529	7.33	53.57	46.24	9.31
9	5835	5783	52	99.11	0.89	52511	47930	4581	8.34	54.19	45.85	10.46
10	5834	5793	41	99.30	0.70	58345	53723	4622	9.34	54.67	45.33	11.62
11	5835	5788	47	99.19	0.81	64180	59511	4669	10.35	55.23	44.88	12.75
12	5834	5802	32	99.45	0.55	70014	65313	4701	11.36	55.61	44.25	13.89
13	5835	5799	36	99.38	0.62	75849	71112	4737	12.37	56.03	43.67	15.01
14	5835	5793	42	99.28	0.72	81684	76905	4779	13.37	56.53	43.15	16.09
15	5834	5787	47	99.19	0.81	87518	82692	4826	14.38	57.09	42.70	17.13
16	5835	5787	48	99.18	0.82	93353	88479	4874	15.39	57.65	42.27	18.15
17	5834	5787	47	99.19	0.81	99187	94266	4921	16.39	58.21	41.82	19.16
18	5835	5797	38	99.35	0.65	105022	100063	4959	17.40	58.66	41.26	20.18
19	5834	5790	44	99.25	0.75	110856	105853	5003	18.41	59.18	40.77	21.16
20	5835	5795	40	99.31	0.69	116691	111648	5043	19.42	59.65	40.24	22.14

Test Results:

Test	# Record		# Goods		# Bads		Fraud Rate					
	250,053		246,500		3,553		0.01422					
	Bin Statistics					Cumulative Statistics						
Population Bin %	# Records	# Goods	# Bads	% Goods	% Bads	Total # Records	Cumulative Goods	Cumulative Bads	% Goods	% Bads (FDR)	KS	FPR
0	0	0	0	0.00	0.00	0	0	0	0	0	0	0
1	2501	886	1615	35.43	64.57	2501	886	1615	0.36	45.45	45.10	0.55
2	2500	2385	115	95.40	4.60	5001	3271	1730	1.33	48.69	47.36	1.89
3	2501	2418	83	96.68	3.32	7502	5689	1813	2.31	51.03	48.72	3.14
4	2500	2478	22	99.12	0.88	10002	8167	1835	3.31	51.65	48.33	4.45
5	2501	2487	14	99.44	0.56	12503	10654	1849	4.32	52.04	47.72	5.76
6	2500	2478	22	99.12	0.88	15003	13132	1871	5.33	52.66	47.33	7.02
7	2501	2488	13	99.48	0.52	17504	15620	1884	6.34	53.03	46.69	8.29
8	2500	2486	14	99.44	0.56	20004	18106	1898	7.35	53.42	46.07	9.54
9	2501	2479	22	99.12	0.88	22505	20585	1920	8.35	54.04	45.69	10.72
10	2500	2487	13	99.48	0.52	25005	23072	1933	9.36	54.40	45.04	11.94
11	2501	2477	24	99.04	0.96	27506	25549	1957	10.36	55.08	44.72	13.06
12	2500	2480	20	99.20	0.80	30006	28029	1977	11.37	55.64	44.27	14.18
13	2501	2479	22	99.12	0.88	32507	30508	1999	12.38	56.26	43.89	15.26
14	2500	2487	13	99.48	0.52	35007	32995	2012	13.39	56.63	43.24	16.40
15	2501	2489	12	99.52	0.48	37508	35484	2024	14.40	56.97	42.57	17.53
16	2500	2483	17	99.32	0.68	40008	37967	2041	15.40	57.44	42.04	18.60
17	2501	2488	13	99.48	0.52	42509	40455	2054	16.41	57.81	41.40	19.70
18	2501	2478	23	99.08	0.92	45010	42933	2077	17.42	58.46	41.04	20.67
19	2500	2482	18	99.28	0.72	47510	45415	2095	18.42	58.96	40.54	21.68
20	2501	2482	19	99.24	0.76	50011	47897	2114	19.43	59.50	40.07	22.66

OOT Results:

OOT	# Record		# Goods		# Bads		Fraud Rate					
	166,493		164,107		2,386		0.01422					
	Bin Statistics					Cumulative Statistics						
Population Bin %	# Records	# Goods	# Bads	% Goods	% Bads	Total # Records	Cumulative Goods	Cumulative Bads	% 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	1665	627	1038	37.66	62.34	1665	627	1038	0.38	43.50	43.12	0.60
2	1665	1580	85	94.89	5.11	3330	2207	1123	1.34	47.07	45.72	1.97
3	1665	1620	45	97.30	2.70	4995	3827	1168	2.33	48.95	46.62	3.28
4	1665	1650	15	99.10	0.90	6660	5477	1183	3.34	49.58	46.24	4.63
5	1665	1652	13	99.22	0.78	8325	7129	1196	4.34	50.13	45.78	5.96
6	1665	1654	11	99.34	0.66	9990	8783	1207	5.35	50.59	45.23	7.28
7	1665	1655	10	99.40	0.60	11655	10438	1217	6.36	51.01	44.65	8.58
8	1664	1654	10	99.40	0.60	13319	12092	1227	7.37	51.42	44.06	9.85
9	1665	1655	10	99.40	0.60	14984	13747	1237	8.38	51.84	43.47	11.11
10	1665	1652	13	99.22	0.78	16649	15399	1250	9.38	52.39	43.01	12.32
11	1665	1657	8	99.52	0.48	18314	17056	1258	10.39	52.72	42.33	13.56
12	1665	1643	22	98.68	1.32	19979	18699	1280	11.39	53.65	42.25	14.61
13	1665	1655	10	99.40	0.60	21644	20354	1290	12.40	54.07	41.66	15.78
14	1665	1658	7	99.58	0.42	23309	22012	1297	13.41	54.36	40.95	16.97
15	1665	1659	6	99.64	0.36	24974	23671	1303	14.42	54.61	40.19	18.17
16	1665	1645	20	98.80	1.20	26639	25316	1323	15.43	55.45	40.02	19.14
17	1665	1656	9	99.46	0.54	28304	26972	1332	16.44	55.83	39.39	20.25
18	1665	1649	16	99.04	0.96	29969	28621	1348	17.44	56.50	39.06	21.23
19	1665	1650	15	99.10	0.90	31634	30271	1363	18.45	57.12	38.68	22.21
20	1665	1647	18	98.92	1.08	33299	31918	1381	19.45	57.88	38.43	23.11