Name: Amrina Roy
Supervisor: Prof. Stephen Coggeshall
Project: Credit Card Application Fraud Detection

---

1. Executive summary:

The issue of application fraud, where individuals apply for products using false identities, is a major concern in the phone application and credit card industry. In the US, it is estimated to affect 5-10 million customers and cause up to $10 billion in fraud losses every year.
This report outlines the evaluation process used to detect stolen identity application fraud using supervised learning techniques. The dataset consisted of 1 million rows and 10 columns, and involved cleaning and exploring the data, creating additional features, and selecting the most relevant features using the Kolmogorov-Smirnov (KS) score. Several modeling techniques were deployed, including logistic regression, decision trees, random forest, gradient boosting, and neural networks. The results showed that the CATBoost model had the highest fraud detection rate of ~52% 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

Field Name	% Populated	# Unique Values	Most Common Value
Record	100%	1000000	N/A
SSN	100%	835819	999999999
Firstname	100%	78136	EAMSTRMT
Lastname	100%	177001	ERJSAXA
address	100%	828774	123 MAIN ST

Zip5	100%	26370	68138
homephone	100%	28244	9999999999
fraud_label	100%	2	0

2.2 Data Description:

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

(2) ***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

(3) ***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.

(4) ***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.
(5) ***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.
(6) ***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
(7) ***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.

(8) ***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'

(9) ***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.
(10) ***fraudlabel***: Fraud = 0 (no fraud label). Fraud = 1 (fraud label)
Distribution count: Fraud = 0 [985,607]. Fraud = 1 [14,393]

3. **<u>Data cleaning:</u>**

Before implementing any machine learning model on a dataset, it's essential to clean the data and address any missing, irrelevant, inaccurate, or corrupted data points. This may involve replacing, modifying, or removing data points to ensure the accuracy and integrity of the model.

3.1 <u>Handling Frivolous Values:</u>

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. **<u>Variable creation:</u>**

Feature Engineering refers to the creation of new features from existing data. It is a crucial step in the machine learning process, where domain knowledge is used to extract meaningful characteristics, properties, and attributes from raw data. This process results in new variables that can be used in predictive models, improving their accuracy, and simplifying data transformations. Feature engineering can be applied to both supervised and unsupervised learning algorithms, and a feature is any measurable input that can be fed into the model.

In our project, we have carried out feature extraction process across the following steps:

- Target encoding

- Statistical smoothing
- Creation of attributes

4.1 Target Encoding:

Encoding categorical variables is a crucial aspect of feature engineering. The goal is to replace categorical data with numerical values, allowing for use in statistical and machine learning algorithms that only work with numerical data. There are several encoding methods available, including one-hot encoding, ordinal encoding, target encoding, and Bayesian target encoding.

Target encoding is a process of converting categorical variables into numeric variables by calculating the average of the target value for each category. This method is useful for both binary classification and regression problems. In case of a multiclass classification, m-1 new variables are created, where m is the number of classes, to represent the categorical data. This technique transforms categorical variables into numerical values that can be used by statistical and machine learning algorithms.

4.2 Statistical Smoothing:

A smoothing formula is used to smoothly transition a value from one number to another. It is used in target encoding to overcome the problem of overfitting. We have used the following logistic formula as our smoothing formula in our project.

$$\boxed{\text{Value} = Y_{\text{low}} + \frac{Y_{\text{high}} - Y_{\text{low}}}{1 + e^{-(n - n_{mid})/c}}}$$

In our project, we have used target encoding to create the variable 'DOW' which is day of the week. We used the above formula to convert each day of the week to a numerical variable. The below chart shows the values for each day of the week:

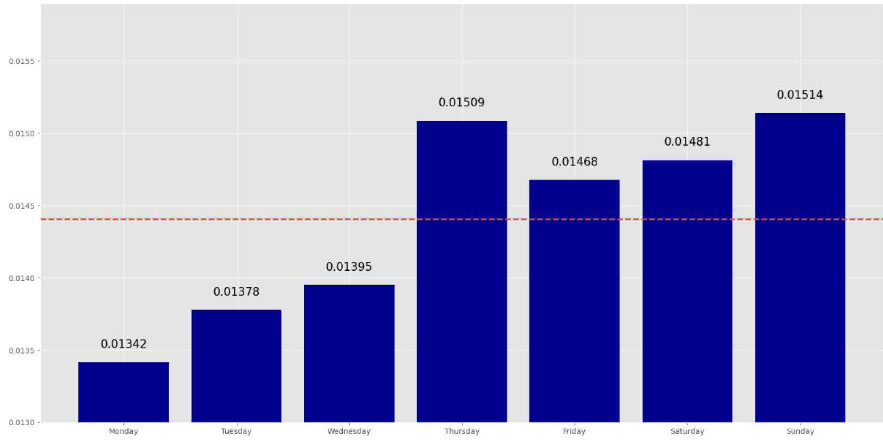


Figure 1: Day of Week - Target Encoded

4.3 Modes of fraud and creation of attributes and variables:

There are three primary modes of identity fraud: identity theft, identity manipulation, and synthetic identity.

- **Identity theft** – Identity theft involves a fraudster using a real identity that they have stolen to apply for products and services. This project strives to capture inauthentic applications that leveraged such stolen identities. To detect this type of fraud, we examine whether the stolen identity is linked to multiple contact points, and we track the frequency of use of various personal identifying information (PII) elements - the various entities, combinations of those entities, velocity, days since and relative velocity variables are created for this purpose.

- **Identity manipulation** – In this fraud, the fraudster makes slight alterations to their own identity, such as making small changes to PII. To identify this type of fraud, we search for systematic variations in PII elements.

- **Synthetic identity** – Synthetic identity fraud involves a fraudster fabricating a completely fake identity. To detect this type of fraud, we look for instances where multiple PII elements are associated with different identities.

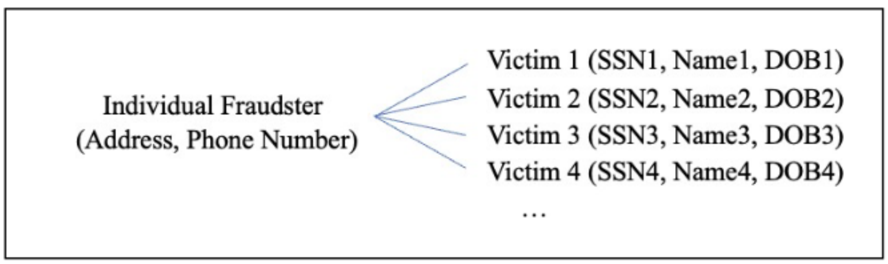
4.3.1 Rationale behind creating variables for capturing Identity theft:

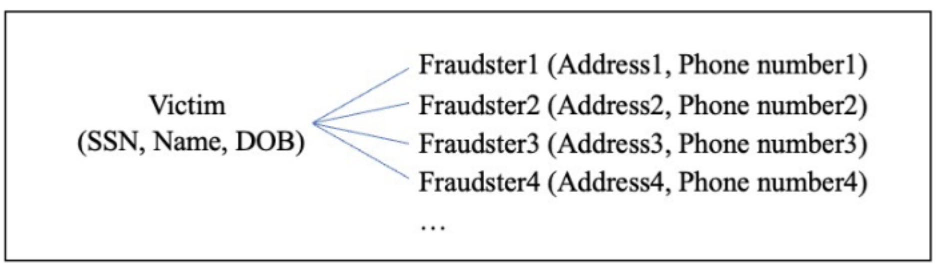
Case 1: An individual fraudster might be applying to various products or services with many stolen identities. This can be confirmed when he/she uses the same address or phone number in many applications containing different SSNs, Names, and DOBs.

Case 2: One victim's SSN, Name and DOB may be used by several fraudsters using these in multiple applications having different addresses and phone numbers.

Hence, we chose to generate our potential features by linking these characteristics. For instance, we evaluated the number of applications with the same address, phone number, and the number of SSNs associated with the same phone number or address, etc., in order to identify all possible combinations.

Finally, we selected the top 20 most informative features through feature selection methods.





4.3.2 Creating variables:

We created a set of candidate/expert variables for each entity or combination group, including:

- Velocity variables
- Days since variables: variables
- Relative velocity variables
- Entity count variables

<table>
<tr><th>Description of variables</th><th>Attributes / Entities used</th><th># variables created</th></tr>
<tr><td>Velocity and Days Since variables:

Days since variables: # of days since that attribute has been last seen

Velocity variables: # of applications with that attribute over the past n days

where values of n are: [0, 1, 3, 7, 14, 30]

Total days since variables: 23 (one for each attribute)
Total velocity over n days variables: 23 * 6 = 138
Total new variables = 138 + 23 = 161</td><td rowspan="4">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</td><td>161</td></tr>
<tr><td>Relative velocity variables (ratio):

$\frac{\text{\# Applications with that } attribute \text{ seen in the recent past } \{0, 1\} \text{ days}}{\text{\# Applications with that same } attribute \text{ seen in the past } \{3, 7, 14, 30\} \text{ days}}$

Total number of variables = 23 (# of attributes) * 2 (for n in recent past {0, 1}) * 4 (for n in past {3, 7, 14, 30}) = 23 * 8 = 184</td><td>184</td></tr>
<tr><td>Entity Count Variables (Unique):

unique entity for every particular entity over the past n days:

values of n are: [0, 1, 3, 7, 14, 30, 60]

Total new variables created: 23 (entity #) * 22 (for every entity, there are 22 entities that we count for) * 7 (# of n) = 3542</td><td>3,542</td></tr>
<tr><td>Age Indicator variables:

For each entity, the maximum, mean and minimum age of the person when applying for credit card

Total new variables created: 23 (entity #) * 3 (max, min, mean) = 69</td><td>69</td></tr>
</table>

5. **Feature selection**:

Feature selection is an important technique to improve the performance of machine learning models, especially in high-dimensional data. Due to the curse of dimensionality, it is easier to fit fewer dimensions of nonlinear models. Feature selection can also improve the model architecture. To facilitate the selection process, variables are ordered based on their relevance. For example, in our fraud detection project, we first applied filter feature selection by calculating the KS statistic through univariate tests to get the top 13% of variables. Then, we used the XG Boost model to do the forward wrapper selection and identified 20 variables with a fraud detection rate at a 0.03 cutoff. The faster runs achieved through feature selection help ensure that we're working with the most relevant variables and improving the efficiency of our modeling process.

Wrapper Order	Variable	Filter score
1	fulladdress_day_since	0.333
2	ssn_dob_unique_count_for_name_homephone_60	0.181
3	homephone_count_3	0.195
4	name_dob_count_30	0.227
5	zip5_unique_count_for_ssn_7	0.206
6	fulladdress_unique_count_for_name_7	0.272
7	zip5_unique_count_for_ssn_3	0.221
8	fulladdress_unique_count_for_ssn_name_60	0.286
9	homephone_unique_count_for_dob_14	0.178
10	homephone_unique_count_for_ssn_lastname_1	0.174
11	homephone_unique_count_for_ssn_zip5_7	0.186
12	fulladdress_homephone_unique_count_for_ssn_name_60	0.186
13	ssn_dob_unique_count_for_fulladdress_60	0.183
14	address_count_14	0.322
15	address_count_7	0.301
16	address_count_0_by_30	0.291
17	fulladdress_count_0_by_30	0.290
18	fulladdress_unique_count_for_ssn_lastname_60	0.286
19	fulladdress_unique_count_for_ssn_60	0.286
20	fulladdress_unique_count_for_ssn_firstname_60	0.286

6. **Preliminary models exploration**:

To find the optimal model, we experiment with 10 variables and various hyperparameters as inputs at the 3% FDR. First, we establish a baseline performance using logistic regression and then compare its performance with several other nonlinear models such as Decision tree, Random Forest, LightGBM, Neural Network, GBC, Catboost, and XGBoost. This helps us identify the best-performing model.

Model	Parameters								Average FDR at 3%		
Logistic Regression	Iteration	c	solver	l1 ratio	max_iter			penalty	Train	Test	OOT
	1	1	lbfgs	None	20			l2	48.64	48.50	47.15
	2	1	lbfgs	None	100			l2	48.93	48.53	47.18
	3	1	liblinear	None	100			l1	48.81	48.92	47.46
	4	0.5	newton-cg	None	100			l2	49.06	48.35	47.44
	5	1	saga	0.5	100			elasctic	48.79	48.92	47.44
	6	0.5	saga	0.7	1000			elasctic	48.96	48.44	47.34
Decision tree	Iteration	criterion	max_depth	min_samples_split	min_samples_leaf			splitter	Train	Test	OOT
	1	gini	2	1000	500			best	45.96	46.03	44.38
	2	gini	10	2000	200			best	52.81	52.59	50.51
	3	entropy	15	2500	200			best	53.07	52.13	50.46
	4	entropy	15	2500	150			random	52.15	51.60	49.73
	5	gini	20	2500	170			random	52.29	52.55	50.11
	6	gini	30	3000	250			best	52.88	52.51	50.46
Random forest classifier	Iteration	criterion	max_depth	min_samples_split	min_samples_leaf		n_estimators	max_features	Train	Test	OOT
	1	gini	2	1000	500		3	8	47.76	48.37	46.44
	2	gini	10	2000	200		50	8	52.87	52.27	50.36
	3	gini	20	2500	200		100	8	52.80	52.31	50.41
	3	entropy	25	2500	250		50	8	52.99	51.94	50.36
	4	entopy	30	1500	300		150	8	52.77	52.61	50.43
	5	gini	100	1000	300		250	8	52.76	52.71	50.42
LGBM	Iteration	max_depth	n_estimators	num_leaves	colsample_bytree		subsample	learning_rate	Train	Test	OOT
	1	-1	5	2	1		1	0.1	51.24	51.22	51.19
	2	2	20	10	0.8		0.8	0.01	50.02	50.36	47.93
	3	10	500	20	0.8		0.8	0.1	53.20	52.68	50.47
	4	15	1000	25	0.8		0.8	0.1	53.21	52.41	50.27
	5	15	20000	20	0.8		0.8	0.1	53.25	51.81	49.89
Neural networks	Iteration	hidden_layer_sizes	activation	alpha	learning_rate	learning_rate_imit	max_iter	solver	Train	Test	OOT
	1	2	relu	0.01	constant	0.001	200	adam	50.23	50.15	48.54
	2	10	relu	0.001	constant	0.001	300	sgd	51.15	50.86	49.14
	3	10	logistic	0.001	adaptive	0.001	500	adam	52.19	51.51	49.55
	4	10,10,10	relu	0.001	adaptive	0.001	500	adam	52.59	52.94	50.55
	5	20,15,10	relu	0.001	constant	0.0001	600	sgd	50.73	51.22	48.89
NN-PCA	Iteration	hidden_layer_sizes	activation	alpha	learning_rate	learning_rate_init	max_iter	solver	Train	Test	OOT
	1	10	logistic	0.001	adaptive	0.001	500	adam	52.29	52.22	49.80
	2	10,10,10	relu	0.001	adaptive	0.001	600	adam	52.61	52.81	50.51
Gradient Boosting Classifier	Iteration	max_depth	n_estimators	min_samples_split	min_samples_leaf	max_features	learning_rate	subsample	Train	Test	OOT
	1	2	5	2	1	None	0.1	1	48.97	49.64	47.40
	2	10	50	100	20	5	0.01	0.8	52.95	52.49	50.37
	3	7	100	200	10	7	0.001	0.7	52.70	52.35	50.20
	4	8	500	150	11	8	0.1	0.9	53.90	51.87	50.19
XGBoost	Iteration	max_depth	n_estimators	learning_rate	criterion	min_samples_split	min_samples_leaf	max_features	Train	Test	OOT
	1	2	5	0.1	friedman_mse	2	1	None	49.60	49.30	47.72
	2	5	8	0.01	squared_errror	100	10	auto	50.98	50.70	48.60
	3	8	100	0.1	friedman_mse	150	8	5	53.17	52.56	50.51
	4	7	500	0.01	friedman_mse	180	9	8	52.84	52.87	50.51
Catboost	Iteration		max_depth		iterations		learning_rate		Train	Test	OOT
	1		2		5		0.1		50.04	50.21	47.92
	2		5		50		0.01		51.57	50.25	48.85
	3		7		200		0.1		52.70	53.08	50.63
	4		6		300		0.001		50.39	50.60	48.13

7. Summary of results:

The CATBoost model was selected as the final model after preliminary exploration, as it showed the highest average FDR for testing and relatively high FDR for OOT, training, and test datasets, with a small range of standard deviations indicating stable performance. The data was split into training, testing, and OOT sets, with similar fraud rates of around 0.014. The model could eliminate about 53% of fraud in the training set, 52.3% in the test set, and 51% in the OOT set, by declining only about 3% of applications. Results and top 20 percentile bins for all three datasets are summarized in a table, including the number of records, frauds caught, their percentage, cumulative KS and FPR values.

Training Results:

Train	# Record		# Goods		# Bads		Fraud Rate					
	583,454		575,005		8,449		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	5835	1561	4274	26.75	73.25	5835	1561	4274	0.27	50.59	50.31	0.37
2	5834	5689	145	97.51	2.49	11669	7250	4419	1.26	52.30	51.04	1.64
3	5835	5769	66	98.87	1.13	17504	13019	4485	2.26	53.08	50.82	2.90
4	5834	5787	47	99.19	0.81	23338	18806	4532	3.27	53.64	50.37	4.15
5	5835	5794	41	99.30	0.70	29173	24600	4573	4.28	54.12	49.85	5.38
6	5834	5788	46	99.21	0.79	35007	30388	4619	5.28	54.67	49.38	6.58
7	5835	5795	40	99.31	0.69	40842	36183	4659	6.29	55.14	48.85	7.77
8	5834	5796	38	99.35	0.65	46676	41979	4697	7.30	55.59	48.29	8.94
9	5835	5791	44	99.25	0.75	52511	47770	4741	8.31	56.11	47.81	10.08
10	5834	5800	34	99.42	0.58	58345	53570	4775	9.32	56.52	47.20	11.22
11	5835	5787	48	99.18	0.82	64180	59357	4823	10.32	57.08	46.76	12.31
12	5834	5791	43	99.26	0.74	70014	65148	4866	11.33	57.59	46.26	13.39
13	5835	5791	44	99.25	0.75	75849	70939	4910	12.34	58.11	45.78	14.45
14	5835	5796	39	99.33	0.67	81684	76735	4949	13.35	58.57	45.23	15.51
15	5834	5800	34	99.42	0.58	87518	82535	4983	14.35	58.98	44.62	16.56
16	5835	5800	35	99.40	0.60	93353	88335	5018	15.36	59.39	44.03	17.60
17	5834	5793	41	99.30	0.70	99187	94128	5059	16.37	59.88	43.51	18.61
18	5835	5802	33	99.43	0.57	105022	99930	5092	17.38	60.27	42.89	19.62
19	5834	5786	48	99.18	0.82	110856	105716	5140	18.39	60.84	42.45	20.57
20	5835	5780	55	99.06	0.94	116691	111496	5195	19.39	61.49	42.10	21.46

Test Results:

Test	# Record		# Goods		# Bads		Fraud Rate					
	250,053		246,495		3,558		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	2501	726	1775	29.03	70.97	2501	726	1775	0.29	49.89	49.59	0.41
2	2500	2447	53	97.88	2.12	5001	3173	1828	1.29	51.38	50.09	1.74
3	2501	2468	33	98.68	1.32	7502	5641	1861	2.29	52.30	50.02	3.03
4	2500	2486	14	99.44	0.56	10002	8127	1875	3.30	52.70	49.40	4.33
5	2501	2487	14	99.44	0.56	12503	10614	1889	4.31	53.09	48.79	5.62
6	2500	2480	20	99.20	0.80	15003	13094	1909	5.31	53.65	48.34	6.86
7	2501	2475	26	98.96	1.04	17504	15569	1935	6.32	54.38	48.07	8.05
8	2500	2482	18	99.28	0.72	20004	18051	1953	7.32	54.89	47.57	9.24
9	2501	2481	20	99.20	0.80	22505	20532	1973	8.33	55.45	47.12	10.41
10	2500	2486	14	99.44	0.56	25005	23018	1987	9.34	55.85	46.51	11.58
11	2501	2482	19	99.24	0.76	27506	25500	2006	10.35	56.38	46.03	12.71
12	2500	2486	14	99.44	0.56	30006	27986	2020	11.35	56.77	45.42	13.85
13	2501	2487	14	99.44	0.56	32507	30473	2034	12.36	57.17	44.80	14.98
14	2500	2485	15	99.40	0.60	35007	32958	2049	13.37	57.59	44.22	16.08
15	2501	2481	20	99.20	0.80	37508	35439	2069	14.38	58.15	43.77	17.13
16	2500	2486	14	99.44	0.56	40008	37925	2083	15.39	58.54	43.16	18.21
17	2501	2480	21	99.16	0.84	42509	40405	2104	16.39	59.13	42.74	19.20
18	2501	2481	20	99.20	0.80	45010	42886	2124	17.40	59.70	42.30	20.19
19	2500	2484	16	99.36	0.64	47510	45370	2140	18.41	60.15	41.74	21.20
20	2501	2491	10	99.60	0.40	50011	47861	2150	19.42	60.43	41.01	22.26

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	509	1156	30.57	69.43	1665	509	1156	0.31	48.45	48.14	0.44
2	1665	1638	27	98.38	1.62	3330	2147	1183	1.31	49.58	48.27	1.81
3	1665	1636	29	98.26	1.74	4995	3783	1212	2.31	50.80	48.49	3.12
4	1665	1653	12	99.28	0.72	6660	5436	1224	3.31	51.30	47.99	4.44
5	1665	1655	10	99.40	0.60	8325	7091	1234	4.32	51.72	47.40	5.75
6	1665	1651	14	99.16	0.84	9990	8742	1248	5.33	52.31	46.98	7.00
7	1665	1649	16	99.04	0.96	11655	10391	1264	6.33	52.98	46.64	8.22
8	1664	1648	16	99.04	0.96	13319	12039	1280	7.34	53.65	46.31	9.41
9	1665	1654	11	99.34	0.66	14984	13693	1291	8.34	54.11	45.76	10.61
10	1665	1656	9	99.46	0.54	16649	15349	1300	9.35	54.48	45.13	11.81
11	1665	1655	10	99.40	0.60	18314	17004	1310	10.36	54.90	44.54	12.98
12	1665	1651	14	99.16	0.84	19979	18655	1324	11.37	55.49	44.12	14.09
13	1665	1653	12	99.28	0.72	21644	20308	1336	12.37	55.99	43.62	15.20
14	1665	1650	15	99.10	0.90	23309	21958	1351	13.38	56.62	43.24	16.25
15	1665	1660	5	99.70	0.30	24974	23618	1356	14.39	56.83	42.44	17.42
16	1665	1644	21	98.74	1.26	26639	25262	1377	15.39	57.71	42.32	18.35
17	1665	1652	13	99.22	0.78	28304	26914	1390	16.40	58.26	41.86	19.36
18	1665	1654	11	99.34	0.66	29969	28568	1401	17.41	58.72	41.31	20.39
19	1665	1657	8	99.52	0.48	31634	30225	1409	18.42	59.05	40.63	21.45
20	1665	1656	9	99.46	0.54	33299	31881	1418	19.43	59.43	40.00	22.48