DSO 562 FRAUD ANALYTICS
Project 3

Professor Stephen Coggeshall

Supervised Fraud Model
On Application Data

Team 1

Kalvin Tran | Aviroop Ghosal | Mohamad Ganji
Zhaojun (Pauline) Liu | Mark Orgel | Vu Duong
Jiaqi Zhu | Saif Rehman | Zhihan (Han) Li

May / 02 / 2019

Table of Contents

Executive Summary

In this paper, we outline the process used to develop an optimum model to identify fraudulent application fraud ("Application Data") in the calendar year 2016. Before cleaning and analyzing the data, an initial Data Quality Report ("DQR") was created where we reviewed the distribution and frequency of the fields (Please see **Exhibit 1** for the DQR). We then proceeded to clean the data by replacing the frivolous fields. (Please see the Data Cleaning Section for more information regarding the cleaning methodology) and created 356 candidate variables (Please see the Variable Creation Section for more information regarding the created variables).

To reduce the number of variables considered for modeling, the data set was broken into three (3) parts: training, testing, and out of time (OOT) sets. The Kolmogorov-Smirnov method (KS) and Fraud Detection Rate (FDR) were then utilized to identify useful variables for predicting fraud. However, only the KS Score was utilized to filter fields as the FDR returned similar results. Of the 356 candidate variables, 25 variables were selected to model fraud.

Next, 6 different statistical models were used to predict fraud - logistic regression, decision tree, random forest, gradient boosting, AdaBoost, and neural network. Of the 6 models, the Gradient Boosting model was considered the best model due to (i) its high FDR for both training and testing data sets, (ii) its low difference between the FDRs of both the training and testing data sets, and (iii) its highest FDR score on out of time data among all models. Although the random forest model scored quite high on the out of time data, due to the higher difference between the training and testing data, higher complexity of the model (higher number of trees and nodes), and slightly lower model performance, we didn't consider this as our best model.

Predicting frauds for the OOT data set resulted in an FDR of approximately 53.35% for 3% of the population. Based upon a saving of $6,000 for every fraud caught and a loss of $50 for every false positive, the Gradient Boosting model resulted in a total savings of $7.45M at 3%. As shown in the Model Algorithms section, the client should set a score cutoff at approximately 7%, which would save approximately $7.76M.

Description of Data

Our analysis was performed using the Excel file called “application data.csv”. This data was made available on Blackboard by Professor Stephen Coggeshall for education purpose. The data reviewed in this report originally includes 10 different categorical fields for every transaction. These fields represent information regarding the order the application was made, the identity and contact information of the applicant, and whether the application was fraudulent. Of the 1,000,000 applications, approximately 1.44% of them labeled as fraudulent applications by Professor Coggeshall using characteristics of real fraud for educational purposes. An aggregate summary of the fields’ summary is presented in **Figure 1** below.

Figure 1: Field Summary

Field	Records	Unique Values	Most Common Entry	Count of Entry	Percentage of Total
Recnum	1,000,000	1,000,000	N/A	N/A	N/A
Date	1,000,000	365	20160816	2,877	0.29%
SSN	1,000,000	835,819	999999999	16,935	1.69%
Firstname	1,000,000	78,136	EAMSTRMT	12,658	1.27%
Lastname	1,000,000	177,001	ERJSAXA	8,580	0.86%
Address	1,000,000	828,774	123 MAIN ST	1,079	0.11%
Zip5	1,000,000	26,370	68138	823	0.08%
Dob	1,000,000	43,673	19070626	126,568	12.66%
Homephone	1,000,000	28,244	9999999999	78,512	7.85%
Fraud	1,000,000	2	0	985,607	98.56%

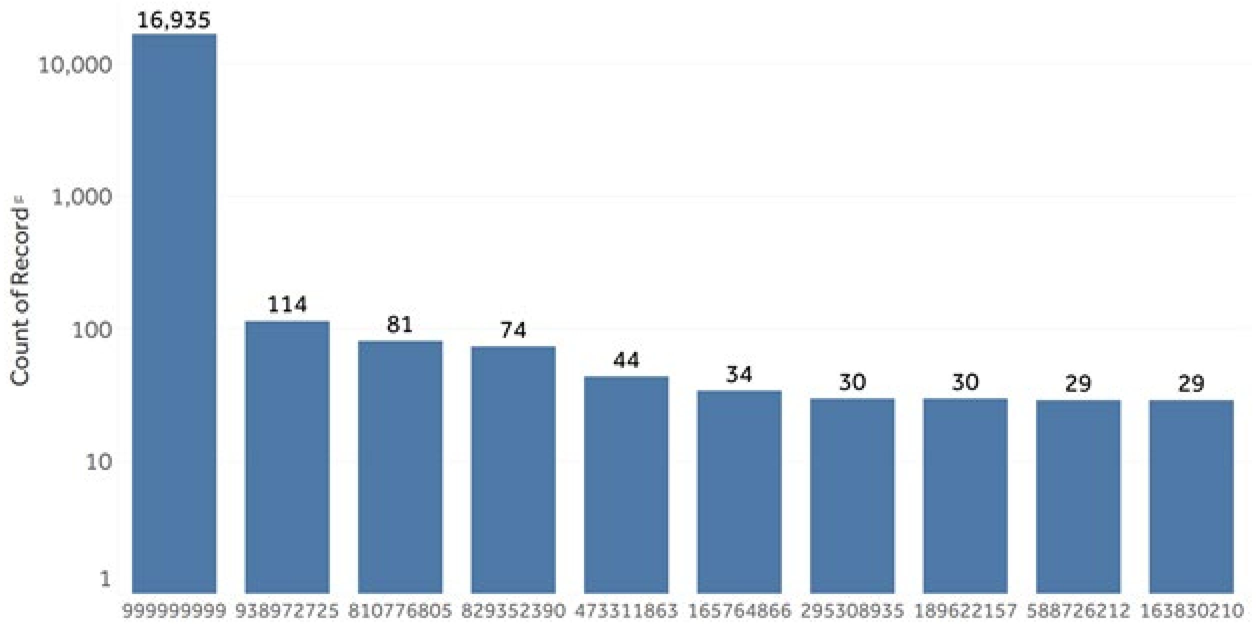
Summary Distributions of Most Important Variables

Below we have identified several fields that were critical for our analysis: Social Security Number (SSN), Address, Date of birth (Dob), and Home Phone Number (Homephone). These are the primary variables that we used to create new variables and predict fraud. Details about these values and their distributions within the dataset are as follows:

Social Security Number (SSN)

SSN is a categorical field that identifies the Social Security Number for each applicant. As shown in Figure 1, there are 1,000,000 records and among those records, there are 835,819 unique social security numbers. **Figure 2** below shows the top ten (10) applicants with the most applications.

Figure 2: Top 10 Social Security Numbers with the most applications in 2016

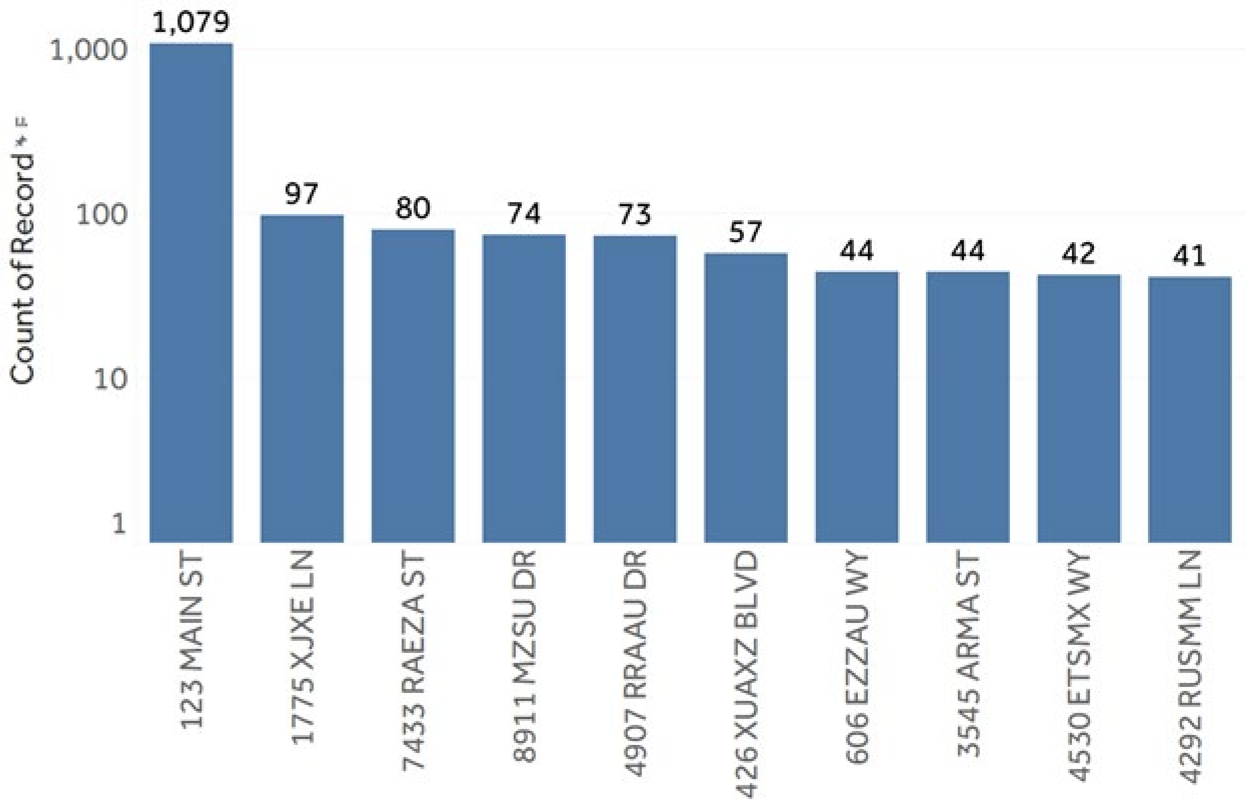


As shown in **Figure 2** above, the SSN with the most applications is 999999999 which made up approximately 1.69% of the applications made in 2016. This is more than two (2) magnitudes greater than the SSN with the second most applications. It would be likely that applications that left the SSN field blank ended up with this SSN and would make it difficult to identify fraud since it would link all applications with the SSN 999999999. Thus, SSN 999999999 is considered a frivolous value and must be cleaned. Please see the following section for more information in the cleaning process.

Address

Address is a categorical field for the mailing address of the applicant of the application. As shown in **Figure 1**, there are 1,000,000 records with an address and of those records, there are 177,001 unique addresses. **Figure 3** below shows the top ten (10) addresses with the most applications.

Figure 3: Top 10 Addresses with the most applications in 2016

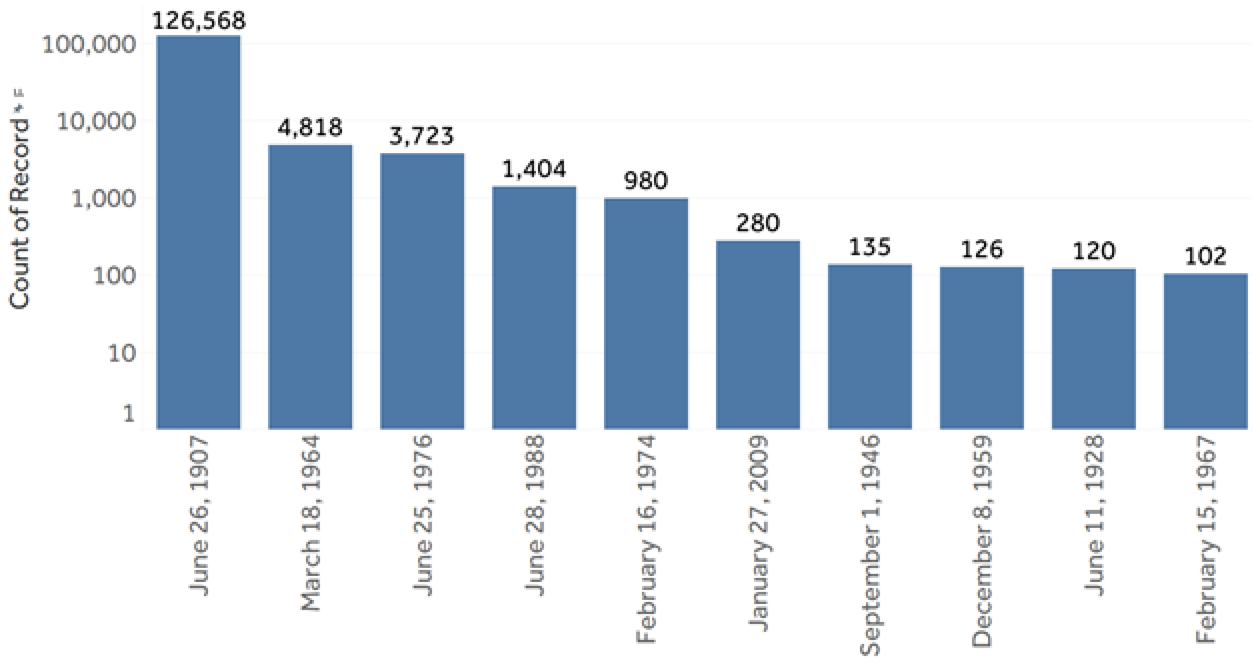


Based on **Figure 3** above, 123 MAIN ST is the address that has the most amount of applications with approximately 0.11% of all the applications made in 2016. It is about a magnitude greater than the second most common address, 1775 XJXE LN. It is likely that 123 MAIN ST is a frivolous value and was treated as such in the analysis. Please see the following section for more information in the cleaning process.

Date of Birth (Dob)

Dob is a categorical field used to identify the month, day, and year an application was made in the form of yyyymmdd. As shown in **Figure 1**, there are 1,000,000 records with a date and of those records, there are 43,673 unique dates. **Figure 4** shows the top ten (10) dates of birth with the most applications.

Figure 4: Top 10 Dates of Birth with the most applications in 2016

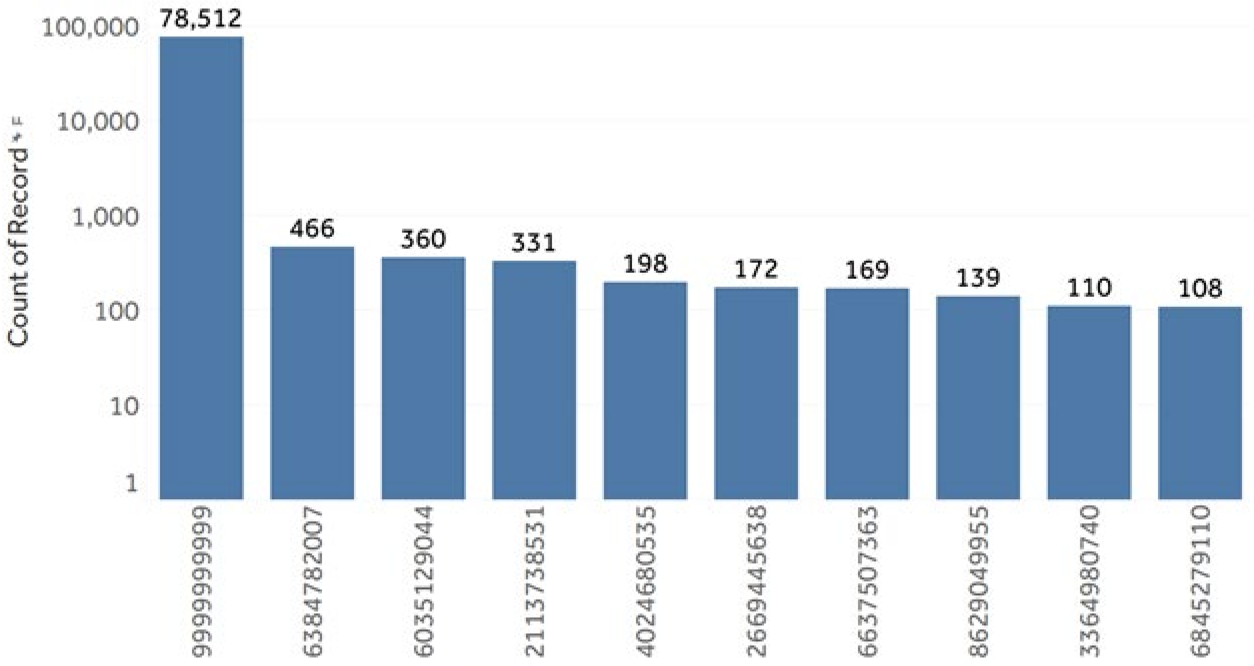


As shown in **Figure 4** above, June 26, 1907 is the most common date of birth among all the applications. This date is highly unreasonable as it makes up approximately 12.66% of the data and the is less than 0.1% of the population in the United States that is older than 110 years. Thus, this field was treated as a frivolous value in the analysis. Please see the following section for more information in the cleaning process.

Home Phone Number (Homephone)

Homephone is a categorical field used to identify the phone number of the application. As shown in **Figure 1**, there are 1,000,000 records and across those records, there are 28,244 unique phone numbers. **Figure 5** below shows the top ten (10) phone numbers with the most applications. Phone numbers are usually structured with ten (10) digits. However, it is assumed that 9-digit (or less) phone numbers in the data have a leading zero (0).

Figure 5: Top 10 Home Phone Numbers with the most applications in 2016



Based on **Figure 5** above, 9999999999 is the most common phone number, making up approximately 7.85% of all the applications. This phone number may have been entered as 9999999999 because it was left blank on the application and was thus treated as a frivolous value in this analysis. Please see the following section for more information in the cleaning process.

Data Cleaning

There is no missing value (meaningless zero value or empty value) in the original application data; however, the values with the highest frequency in fields SSN, Address, homephone, DOB are frivolous values. Extensive effort was put forth into determining a reasonable replacement for these frivolous values because an incorrect linkage can mislead our model when training to predict fraud. Thus, we utilized Python to replace all these frivolous values for each entity by using its record number, which is a unique value for each entity. It can avoid the incorrect linkage between these applications with the same frivolous value during both the variable creation and modeling stages. **Figure 6** shows the count of frivolous value for each of these fields.

Figure 6: Frivolous Value

Field	Frivolous Value	Count
SSN	999999999	16,935
Address	123 MAIN ST	1,079
Homephone	9999999999	78,512
DOB	1907626	126,568

Candidate Variables

To fully explore the hidden information and relationship between fraud and fields, we extensively created:

1. Three (3) variables indicating the application day, month and year.
2. Two (2) variables that are the combined entities: first name, last name and DOB, address and 5-digit zip codes.
3. Fourteen (14) variables to study the number of days since we last saw the entity or the entity combination.
4. One hundred and twelve (112) velocity variables to study the count of records seen for that entity or entity combination over the past 1, 2, 3, 7, 14, 21, 30 and 60 days.
5. Fifty-six (56) variables to calculate the ratio of the count of records seen for the entity combination in the past 1 day to the count of the records seen for the entity combination over the past 2, 3, 7, 14, 21, 30, 60 days.
6. Forty-eight (48) variables to calculate the ratio of the count of records seen for the entity combination in the past 2 days to the count of the records seen for the entity combination over the past 3, 7, 14, 21, 30, 60 days.
7. Forty (40) variables to calculate the ratio of the count of records seen for the entity combination in the past 3 days to the count of the records seen for the entity combination over the past 7, 14, 21, 30, 60 days.
8. Thirty-two (32) variables to calculate the ratio of the count of records seen for the entity combination in the past 7 days to the count of the records seen for the entity combination over the past 14, 21, 30, 60 days.
9. Twenty-four (24) variables to calculate the ratio of the count of records seen for the entity combination in the past 14 days to the count of the records seen for the entity combination over the past 21, 30, 60 days.
10. Sixteen (16) variables to calculate the ratio of the count of records seen for the entity combination in the past 21 days to the count of the records seen for the entity combination over the past 30, 60 days.
11. Eight (8) variables to calculate the ratio of the count of records seen for the entity combination in the past 30 days to the count of the records seen for the entity combination over the past 60 days.
12. One (1) risk table variable to calculate the average fraud rate per weekday.

Overall, we created three hundred and fifty-six (356) new variables.

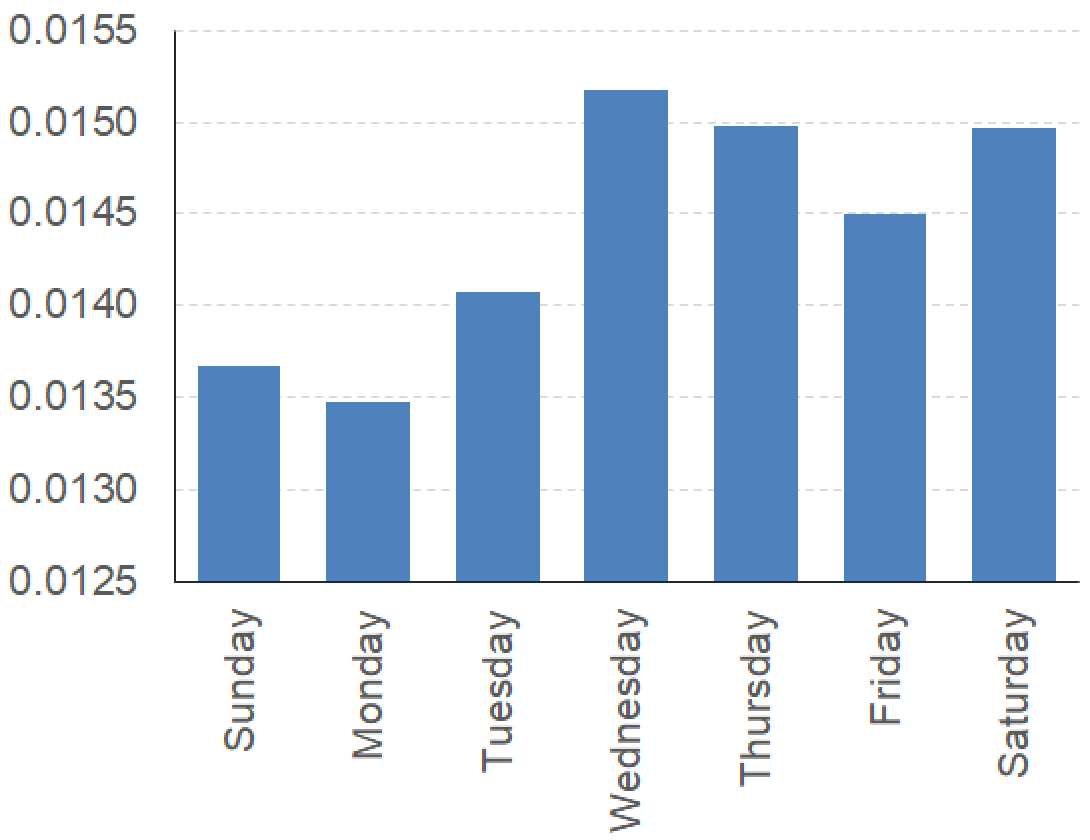
Single entity: SSN, namedob, fulladdress, phone
Entity combination groups: SSN_fulladdress, SSN_homephone, SSN_namedob, Fulladdress_namedob, Fulladdress_homephone, Namedob_homephone, Firstname_ssn, lastname_ssn

FIGURE 7: Variable Creation

Variable Group	*Count*
Application Day/Month/Year	3
Combined entity: First name, last name and DOB, Address and 5-digit Zip codes	2
Days since variable: the number of days since the last time the entity or entity combination showed	14
Record count variable: the number of records of the entity/entity combination showed in the past 1, 2, 3, 7, 14, 21, 30, 60 days	112
Record count ratio variables (1 day): The ratio of the number of records of the entity combination showed in the past 1 day to the number in the past 2, 3, 7, 14, 21, 30, 60 days	56
Record count ratio variables (2 days): The ratio of the number of records of the entity combination showed in the past 2 days to the number in the past 3, 7, 14, 21, 30, 60 days	48
Record count ratio variables (3 days): The ratio of the number of records of the entity combination showed in the past 3 days to the number in the past 7, 14, 21, 30, 60 days	40
Record count ratio variables (7 days): The ratio of the number of records of the entity combination showed in the past 7 days to the number in the past 14, 21, 30, 60 days	32
Record count ratio variables (14 days): The ratio of the number of records of the entity combination showed in the past 14 days to the number in the past 21, 30, 60 days	24
Record count ratio variables (21 days): The ratio of the number of records of the entity combination showed in the past 21 days to the number in the past 30, 60 days	16

Variable Group	***Count***
Record count ratio variables (30 days): The ratio of the number of records of the entity combination showed in the past 30 days to the number in the past 60 days	8
Risk table variable: Weekday risk - Average in sample risk by day of the week the application was made on. Please see Figure 8 for more information	1

FIGURE 8: Weekday Risk Table



Feature Selection Process

Data before November 1, 2016 was randomly divided into training and testing sets and this is the data that is used to perform the feature selection process.

Filter Stage

To reduce the dimensionality, we first performed the feature selection process including methods of filter and wrapper. The main tools used in the Filter stage are the Kolmogorov-Smirnov method (KS) and Fraud Detection Rate (FDR). KS is a robust measure of how well two distributions are separated (in this case, Fraud vs Non-fraud). When plotting the cumulative distribution of goods and bads sorted by the field we want to test, KS is the maximum of the difference of cumulative and it doesn't matter which sides of the plot we start.

Figure 9: KS Algorithm

$$KS = max \int_{x_{min}}^{x} [P_{good} - P_{bad}]dx$$

$$KS = max \sum_{x_{min}}^{x} [P_{good} - P_{bad}]$$

FDR is the other powerful tool that measures what percentage of all the frauds are caught at a particular examination cutoff location. In our case, we set the population rate at 3%. For example, FDR 75% means that the model catches 75% of overall fraud by only checking the top 3% most probable (based on probability score/propensity) fraudulent transactions of the total population.

Two fields were added into this stage as the comparison: “Fraud” from the original data and “Random” which contained only random numbers. “Fraud” should have perfect KS and FDR while “Random” is supposed to perform poorly in the test. In our case, we used only KS to filter fields while FDR was used as the reference since FDR returned a similar ranking.

159 variables were left for wrapper stage based on their KS scores. The top 25 of them are listed in **figure 10**.

FIGURE 10: Top 20 KS And FDR

No.	Field	KS	FDR
2	fraud_label	1	1
18	Days_since_fulladdress	0.328941	0.354043
16	Days_since_namedob	0.225724	0.251104
7	Days_since_ssn	0.224871	0.251437
30	Days_since_ssn_namedob	0.224777	0.250437
57	Days_since_fulladdress_homephone	0.224766	0.250021
37	Record_Count_ssn_namedob_30d	0.224689	0.250521
64	Record_Count_fulladdress_homephone_30d	0.224417	0.251437
75	Days_since_firstname_ssn	0.224166	0.250521
82	Record_Count_firstname_ssn_30d	0.224149	0.251187
84	Days_since_lastname_ssn	0.223977	0.250354
91	Record_Count_lastname_ssn_30d	0.223977	0.25077
63	Record_Count_fulladdress_homephone_21d	0.222232	0.245773
36	Record_Count_ssn_namedob_21d	0.222132	0.246356
81	Record_Count_firstname_ssn_21d	0.221838	0.24569
90	Record_Count_lastname_ssn_21d	0.22158	0.24644
38	Record_Count_ssn_namedob_60d	0.219579	0.24569
92	Record_Count_lastname_ssn_60d	0.218654	0.242442
83	Record_Count_firstname_ssn_60d	0.218235	0.245857
65	Record_Count_fulladdress_homephone_60d	0.217672	0.243191

Wrapper Stage - Stepwise Selection Methods

A wrapper method has a model "wrapped" around the process. Our chosen model is Logistic Regression. We decided to go ahead with the 25 best variables out of 159 candidate variables selected from the KS filter method, i.e., variables with highest KS scores.

Final Selected Variables

FIGURE 11: 25 Selected Variables

No.	Variable Name
1	Days_since_firstname_ssn
2	Days_since_fulladdress
3	Days_since_fulladdress_homephone
4	Days_since_homephone
5	Days_since_lastname_ssn
6	Days_since_namedob
7	Days_since_ssn
8	Days_since_ssn_namedob
9	Record_Count_lastname_14d
10	Record_Count_lastname_1d
11	Record_Count_lastname_21d
12	Record_Count_lastname_2d
13	Record_Count_lastname_30d
14	Record_Count_lastname_3d
15	Record_Count_lastname_60d
16	Record_Count_lastname_7d
17	Record_Count_firstname_ssn_14d
18	Record_Count_firstname_ssn_14d/30d
19	Record_Count_firstname_ssn_14d/60d

No.	Variable Name
20	Record_Count_firstname_ssn_1d
21	Record_Count_firstname_ssn_1d/14d
22	Record_Count_firstname_ssn_1d/21d
23	Record_Count_firstname_ssn_1d/2d
24	Record_Count_firstname_ssn_1d/30d
25	Record_Count_firstname_ssn_1d/3d

Model Algorithms

We tried multiple algorithms for classifying fraud and non-fraud: Logistic Regression, Decision Tree, Random Forest, Gradient Boosting, AdaBoost and Neural Network. The outputs of these algorithms were then combined using a table to compare their performance. We used the data after October 31, 2016 for the out-of-time validations, and randomly split the rest of the dataset into training (70%) and testing set (30%).

Model 1: Logistic Regression

Logistic regression is used to describe properties of data and to explain the relationship between a dependent binary variable and one or more independent variables, used commonly in classification problems. It fits an S-curve (sigmoid function) that can map any real value input to a value between 0 and 1. We used grid search to find the best penalty method and the inverse of regularization strength, based on the "accuracy" scoring method. We set the max iteration count to 200. Our training and testing FDRs were just under 50% for this model, and our out of time FDR was 46.35%, making this one of the weakest models we built.

Model 2: Decision Tree

Decision Tree algorithm divides the data into rectangular spaces/ boxes and assigns a score to each box. The algorithm finds optimal cutoff points for each dimension that reduces impurities on either end of the split point. It is a rule-based algorithm that follows a graphical tree structure. The optimal candidate cutoff points are measured by impurities in resulting 2 boxes, that are added together to weighted by number in each box, and thus the cutoff point resulting in lowest impurity is selected. For continuous inputs, variance is used. This process of selecting the optimal cutoff point is repeated across all dimensions, and the location at a particular dimension at which the lowest impurity is obtained results in the optimal cutoff point, and then the same process is repeated in each of the resulting boxes. This continues until a stopping criterion of total impurity/minimum points/leaf is reached. The Decision Tree model parameters used were Gini impurity for split criteria and the minimum number samples in each box after one split was set to 30, the maximum depth was set to 12. Our Decision Tree model resulted in an FDR of 53.56% on training, 53.0% on testing, and 51.93% on out of time data. Although it resulted in a high FDR on the out of time data, due to its unstable and prone to overfitting nature, we decided to not select this as our best model.

Model 3: Neural Network

This model mimics the way in which human brain processes information. Neural network is a deep learning algorithm that learns data representation very well and maps the input data to output data fairly accurately. Neural network is composed of the input layer, hidden layer and output layer. Each layer consists of nodes that are interconnected with nodes in other layers. This algorithm has been inspired by biological neurons of the brain. The input data is fed into the input layers as input vector (1xn), this input vector is multiplied by weight matrix (nxm), where n is the number of features/number of nodes in the input layer and m is the number of nodes in the hidden layer, this is further added to bias in each of these nodes, hence obtaining a vector that is passed through an activation function (sigmoid/linear/tanh..), thus resulting in a vector that is passed to the output layer. The output layer nodes have the output data, and this output of the transfer function is mapped to it. The loss function is calculated (the actual labeled data values - the function output), these errors are back propagated to each node, the derivative of these errors with respect to the weights are calculated, thus weights are adjusted for each record. This is how the neural network algorithm reads through the entire data incrementally and then through multiple epochs, it achieves a fairly accurate classification or regression model. The important parameters in neural networks are the number of hidden layers, number of nodes, activation function, epochs, learning rate, and number of iterations. The number of neurons in the hidden layer of our Neural Network model is 50. L1 penalty parameter, maximum 200 iterations, learning rate of 0.001, Logistic as our activation function and solver as adam optimizer. We achieved an out of time FDR of 50.71%. Given that this result was worse than the Decision Tree model performance and challenge in model interpretability, we disregarded this model.

Model 4: Random Forest

Random Forest is an ensemble decision tree algorithm. It builds multiple trees by using random-chosen subsets of variables. The decision tree models are fit into these samples, based on higher probability score of the data point being classified in a class, each decision tree model votes on the majority class, thus the class with the highest mode for that record thus is selected as the class for that record in the random forest model. This algorithm is more robust than a decision tree model since even a slight modification in the training data would change the decision tree cutoff point and thus would provide a different result. The important parameters of this model are the number of trees, the maximum depth of each tree, and the minimum number of samples of each split, and we adjusted these parameters to improve the performance of the model. The number of trees used is 700 and we used gini impurity to measure the quality of the split. Minimum 1 sample was required to split at each node and the maximum depth of the trees was set to 15 nodes. Due to stable predictions, higher performance, reduced variance and reduced overfitting nature, this turned out to be 2nd best model, as it resulted in an FDR of 52.18% on the out of time data and performed well on training (58.19%) and testing (55.14%) without overfitting.

Models 5 & 6: Boosted Trees

Boosting is another ensemble technology. It can also improve the performance of the basic decision tree model by building many weak prediction models and eventually produce a stronger one. The parameters in boosted trees are loss function, the number of trees and maximum depth. In this project, we build adaptive boosting and gradient boosting classification models. Both these two boosting models build multiple trees by firstly randomly picking up a subset of observations with equal weights. But they perform differently when they produce the next model:

For adaptive boosting, it will use this first model to fit the whole dataset and identify the misclassified points. When creating the next model, higher weights will be given to the misclassified points so that they will be more likely to be picked up for to next model. The algorithm will repeat this process until there is no change on the loss function. Number of trees was set to 150, learning rate was adjusted to 0.5. As shown in figure 12, the FDRs for training, testing and out of time were 51.89%, 52.33% and 50.08%. However, due to low model performance, we didn't consider this model.

Contrary to adaptive boosting, which increases the weights of misclassified points at every interaction, gradient boosting tries to fit the new model to the residual errors made by the previous model. The objective is to find the bests split to minimize the error. This process will be repeated until the loss function does not change. We ran model with 200 trees and maximum depth of 5. As shown in figure 12, the FDRs for training, testing and out of time were 58.36%, 55.66% and 53.35%.

Thus, the random forest performed fairly well, but the gradient boosting model outperformed them.

Our Best Model

For each algorithm, we adjusted the parameters and a random-selected subset of variables to fit the model. The highest accuracy was attained by the Gradient Boosting model. We set the number of trees as 200, the maximum depth at 5. We used the grid search method to find the best parameters for our model. The FDRs of our best model were 58.36%, 55.66% and 53.35% for training, testing and out of time data respectively. The difference between training and testing was 2.7%. It indicated that the model was relatively robust for further predictions. Our baseline model was the Decision Tree model, FDRs of which were 53.56%, 53.0% and 51.93% for training, testing and out of time data respectively. Gradient Boosting is known to perform well in anomaly detection that typically has unbalanced datasets.

FIGURE 12: FDR at 3% of Each Model

MODEL	Parameter			Average FDR (%) at 3%		
Logistic Regression	**Max Iteration**	**Penalty**		**Train**	**Test**	**OOT**
1	100	L2		48.29	48.78	46.1
2	**200**	**L1**		**48.63**	**49.06**	**46.35**
Decision Tree	**Max Depth**	**Min_samples_leaf**		**Train**	**Test**	**OOT**
1	None	1		86.96	46.11	43.59
2	10	1		52.89	51.89	50.29
3	**12**	**30**		**53.56**	**53.0**	**51.93**
Random Forest	**# of Trees**	**Max Depth**	**Mini_sample_leaf**	**Train**	**Test**	**OOT**
1	500	10	20	54.6	54.16	51.51
2	700	10	1	54.6	54.08	51.47
3	**700**	**15**	**1**	**58.19**	**55.14**	**52.18**
AdaBoost	**# of Trees**	**Learning Rate**		**Train**	**Test**	**OOT**
1	100	0.5		51.7	51.78	49.66
2	**150**	**0.5**		**51.89**	**52.33**	**50.08**
Gradient Boosting	**# of Trees**	**Max Depth**		**Train**	**Test**	**OOT**
1	250	5		58.85	55.5	53.27
2	**200**	**5**		**58.36**	**55.66**	**53.35**
MLP	**Activation**	**Solver**	**Learning Rate**	**Train**	**Test**	**OOT**
1	relu	adam	0.001	52.29	52.47	50.29
2	**logistic**	**adam**	**0.001**	**52.83**	**52.94**	**50.71**

FIGURE 13: The Performance of The Best Model at Training

Training	# Records	# Goods	# Bads	Fraud Rate								
	583454	575049	8405	0.0140								
	Bin Statistics					Cumulative Statistics						
Population Bin %	# Records	# Goods	# Bads	% Goods	% Bads	Total # Records	Cumulative Good	Cumulative Bad	% Good	% Bad (FDR)	KS	FPR (%)
1	5835	1290	4545	22.11	77.89	5835	1290	4545	0.22	54.07	53.85	0.28
2	5835	5576	259	95.56	4.44	11670	6866	4804	1.19	57.16	55.97	1.43
3	5835	5734	101	98.27	1.73	17505	12600	4905	2.19	58.36	56.17	2.57
4	5835	5739	96	98.35	1.65	23340	18339	5001	3.19	59.5	56.31	3.67
5	5835	5748	87	98.51	1.49	29175	24087	5088	4.19	60.54	56.35	4.73
6	5835	5745	90	98.46	1.54	35010	29832	5178	5.19	61.61	56.42	5.76
7	5835	5745	90	98.46	1.54	40845	35577	5268	6.19	62.68	56.49	6.75
8	5835	5766	69	98.82	1.18	46680	41343	5337	7.19	63.5	56.31	7.75
9	5835	5753	82	98.59	1.41	52515	47096	5419	8.19	64.47	56.28	8.69
10	5835	5783	52	99.11	0.89	58350	52879	5471	9.2	65.09	55.89	9.67
11	5835	5788	47	99.19	0.81	64185	58667	5518	10.2	65.65	55.45	10.63
12	5835	5786	49	99.16	0.84	70020	64453	5567	11.21	66.23	55.02	11.58
13	5835	5797	38	99.35	0.65	75855	70250	5605	12.22	66.69	54.47	12.53
14	5835	5785	50	99.14	0.86	81690	76035	5655	13.22	67.28	54.06	13.45
15	5835	5788	47	99.19	0.81	87525	81823	5702	14.23	67.84	53.61	14.35
16	5835	5787	48	99.18	0.82	93360	87610	5750	15.24	68.41	53.17	15.24
17	5835	5788	47	99.19	0.81	99195	93398	5797	16.24	68.97	52.73	16.11
18	5835	5798	37	99.37	0.63	105030	99196	5834	17.25	69.41	52.16	17
19	5835	5794	41	99.3	0.7	110865	104990	5875	18.26	69.9	51.64	17.87
20	5835	5793	42	99.28	0.72	116700	110783	5917	19.26	70.4	51.14	18.72

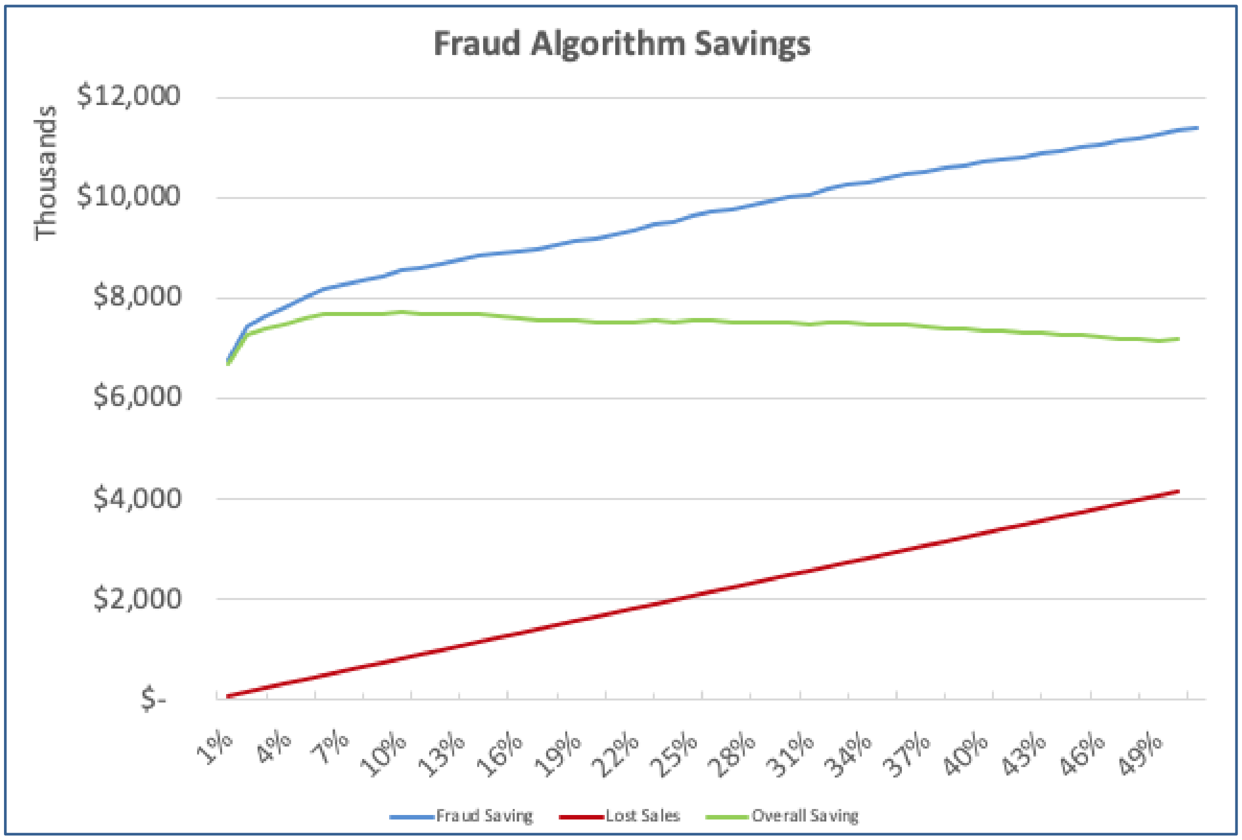
FIGURE 14: The Performance of The Best Model at Testing

Testing	# Records	# Goods	# Bads	Fraud Rate								
	250053	246451	3602	0.0144								
	Bin Statistics					Cumulative Statistics						
Population Bin %	# Records	# Goods	# Bads	% Goods	% Bads	Total # Records	Cumulative Good	Cumulative Bad	% Good	% Bad (FDR)	KS	FPR (%)
1	2501	635	1866	25.39	74.61	2501	635	1866	0.26	51.8	51.54	0.34
2	2501	2409	92	96.32	3.68	5002	3044	1958	1.24	54.36	53.12	1.55
3	2501	2454	47	98.12	1.88	7503	5498	2005	2.23	55.66	53.43	2.74
4	2501	2462	39	98.44	1.56	10004	7960	2044	3.23	56.75	53.52	3.89
5	2501	2462	39	98.44	1.56	12505	10422	2083	4.23	57.83	53.6	5
6	2501	2471	30	98.8	1.2	15006	12893	2113	5.23	58.66	53.43	6.1
7	2501	2463	38	98.48	1.52	17507	15356	2151	6.23	59.72	53.49	7.14
8	2501	2481	20	99.2	0.8	20008	17837	2171	7.24	60.27	53.03	8.22
9	2501	2474	27	98.92	1.08	22509	20311	2198	8.24	61.02	52.78	9.24
10	2501	2481	20	99.2	0.8	25010	22792	2218	9.25	61.58	52.33	10.28
11	2501	2479	22	99.12	0.88	27511	25271	2240	10.25	62.19	51.94	11.28
12	2501	2467	34	98.64	1.36	30012	27738	2274	11.25	63.13	51.88	12.2
13	2501	2489	12	99.52	0.48	32513	30227	2286	12.26	63.46	51.2	13.22
14	2501	2484	17	99.32	0.68	35014	32711	2303	13.27	63.94	50.67	14.2
15	2501	2488	13	99.48	0.52	37515	35199	2316	14.28	64.3	50.02	15.2
16	2501	2477	24	99.04	0.96	40016	37676	2340	15.29	64.96	49.67	16.1
17	2501	2485	16	99.36	0.64	42517	40161	2356	16.3	65.41	49.11	17.05
18	2501	2482	19	99.24	0.76	45018	42643	2375	17.3	65.94	48.64	17.95
19	2501	2475	26	98.96	1.04	47519	45118	2401	18.31	66.66	48.35	18.79
20	2501	2477	24	99.04	0.96	50020	47595	2425	19.31	67.32	48.01	19.63

FIGURE 15: The Performance of The Best Model at Out of Time

Out of Time	# Records	# Goods	# Bads	Fraud Rate								
	166493	164107	2386	0.0143								
	Bin Statistics					Cumulative Statistics						
Population Bin %	# Records	# Goods	# Bads	% Goods	% Bads	Total # Records	Cumulative Good	Cumulative Bad	% Good	% Bad (FDR)	KS	FPR (%)
1	1665	540	1125	32.43	67.57	1665	540	1125	0.33	47.15	46.82	0.48
2	1665	1548	117	92.97	7.03	3330	2088	1242	1.27	52.05	50.78	1.68
3	1665	1634	31	98.14	1.86	4995	3722	1273	2.27	53.35	51.08	2.92
4	1665	1637	28	98.32	1.68	6660	5359	1301	3.27	54.53	51.26	4.12
5	1665	1630	35	97.9	2.1	8325	6989	1336	4.26	55.99	51.73	5.23
6	1665	1638	27	98.38	1.62	9990	8627	1363	5.26	57.12	51.86	6.33
7	1665	1649	16	99.04	0.96	11655	10276	1379	6.26	57.8	51.54	7.45
8	1665	1654	11	99.34	0.66	13320	11930	1390	7.27	58.26	50.99	8.58
9	1665	1649	16	99.04	0.96	14985	13579	1406	8.27	58.93	50.66	9.66
10	1665	1647	18	98.92	1.08	16650	15226	1424	9.28	59.68	50.4	10.69
11	1665	1656	9	99.46	0.54	18315	16882	1433	10.29	60.06	49.77	11.78
12	1665	1651	14	99.16	0.84	19980	18533	1447	11.29	60.65	49.36	12.81
13	1665	1653	12	99.28	0.72	21645	20186	1459	12.3	61.15	48.85	13.84
14	1665	1651	14	99.16	0.84	23310	21837	1473	13.31	61.74	48.43	14.82
15	1665	1658	7	99.58	0.42	24975	23495	1480	14.32	62.03	47.71	15.88
16	1665	1659	6	99.64	0.36	26640	25154	1486	15.33	62.28	46.95	16.93
17	1665	1657	8	99.52	0.48	28305	26811	1494	16.34	62.62	46.28	17.95
18	1665	1652	13	99.22	0.78	29970	28463	1507	17.34	63.16	45.82	18.89
19	1665	1651	14	99.16	0.84	31635	30114	1521	18.35	63.75	45.4	19.8
20	1665	1655	10	99.4	0.6	33300	31769	1531	19.36	64.17	44.81	20.75

FIGURE 16: Fraud Saving Plot



Assuming a $6,000 gain for every fraud that's caught and a $50 loss for every false positive, at the cutoff point 10%, the overall savings is maximized at around $7.7M. However, at the cutoff 7%, the savings is about $7.69M, which is extremely close to the peak. The final decision depends on how the company evaluates the opportunity costs generated from rejecting more applications. If the company want to maximize the savings, 10% would be the optimal cutoff for them. If the company wants to minimize the number of applications it turns away, 7% would be the best choice.

FIGURES: 17 Fraud Score Plot of Certain SSN

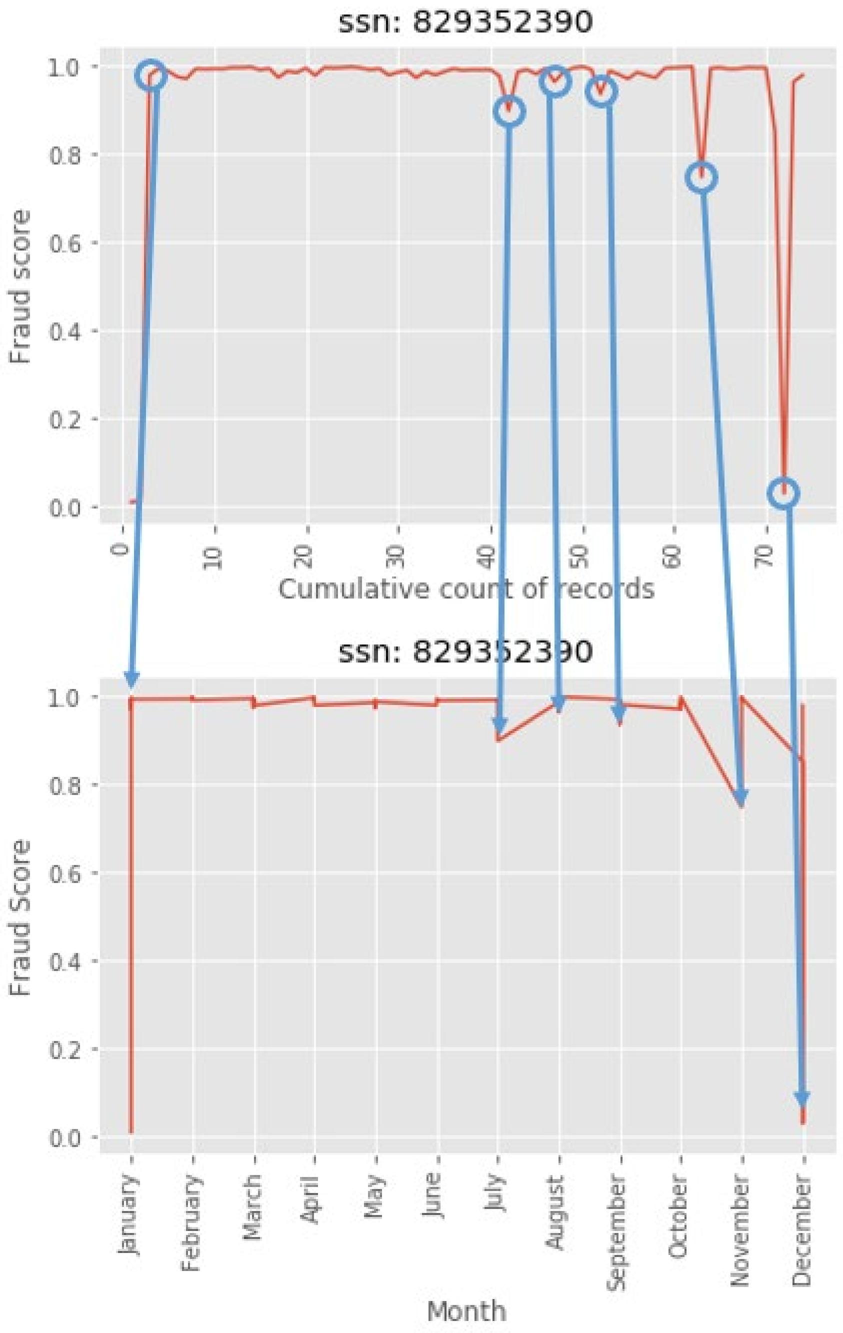
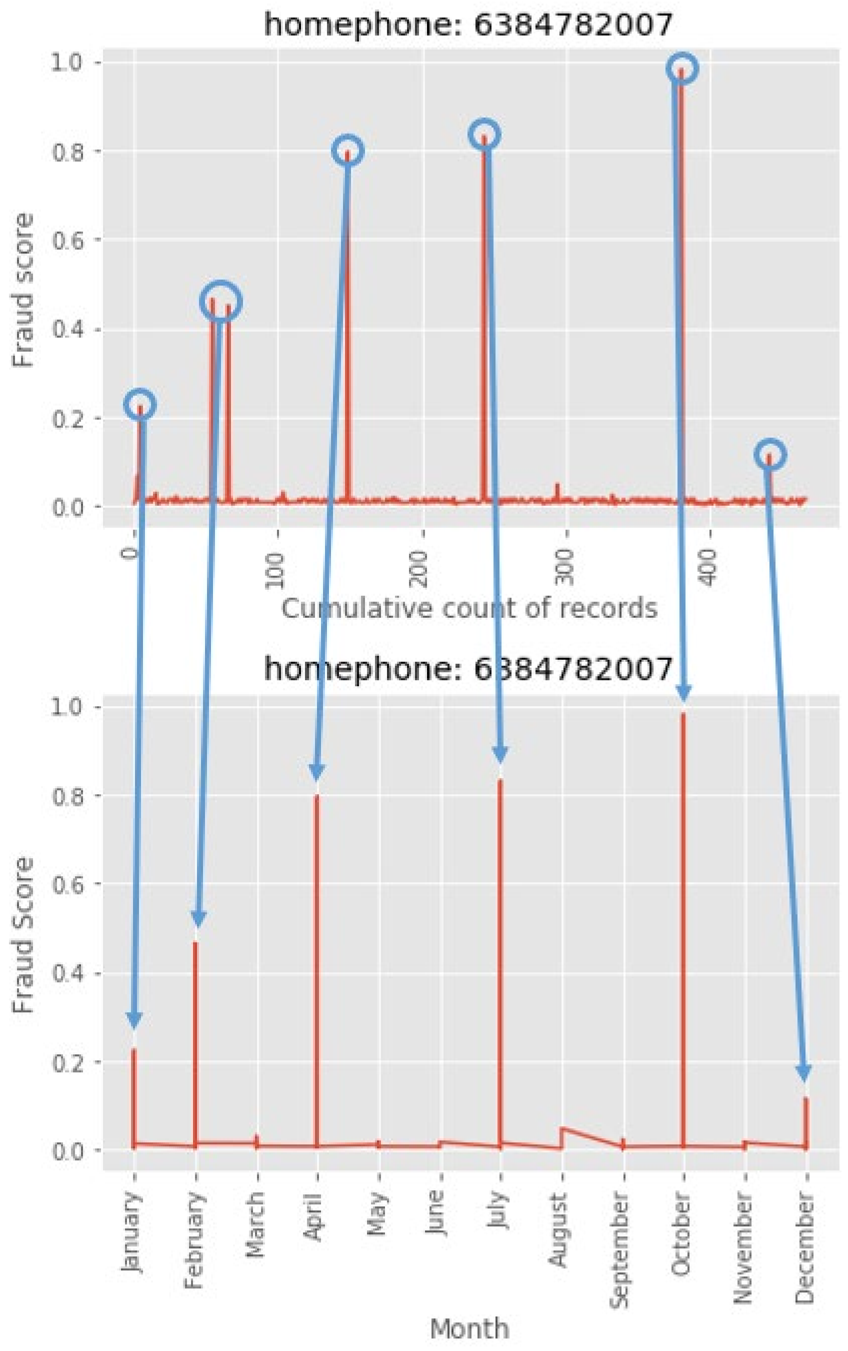


Figure 18: Fraud Score Plot of Certain Homephone



As seen in **Figures 17,** the fraud algorithm caught the application with SSN as 829352390 quite early in the process. The first few records of this application had a low fraud score, but by 3rd week of January, the fraud score spiked up to about 97% due to lot of fraud applications occurring in a short span of time and continued to be so until November, where it dropped to 75%, after which it plummeted down to 3% in December, but again soared up to a high score.

As seen in **Figures 18**, we noticed the fraud score for application with phone number as 6384782007 had been rising gradually over the year. By April, the fraud score for this application was over 50%, and eventually the chances of this application being a fraud increased, reaching a 98% fraud score on October, after which it dropped considerably.

Conclusions

In this project, we investigated 1,000,000 applications made in the calendar year 2016. Each application included information regarding the date the application was made, the applicant's identification and contact information, as well as whether or not the application was fraudulent. Using variables generated from this data and selected for predictive significance, we developed a model that can forecast whether future applications are likely to be fraudulent with a degree of accuracy high enough to (1) capture a large percentage of the instances of fraud and (2) minimize the rejection of good applications which would result in increased opportunity costs.

Of the 6 types of models tested, the Gradient Boosting model was identified as the optimal model to identify potentially fraudulent applications. The FDRs of our best model were 58.36% and 55.66% for training and testing data, respectively. The difference between training and testing was 2.7%, which indicated that the model was relatively robust for further predictions. At 3% of the total applications, the Gradient Boosting model identified approximately 53.35% of all fraudulent applications in the out of time window (November 1, 2016 to December 31, 2016). Although, based on the OOT prediction, a 10% cutoff would maximize net savings from fraud at approximately $7.78M, we recommend a cutoff of 7% in order to minimize the number of rejections. At 7%, the estimated savings would be $22,500 less at approximately $7.76M. However, approximately 5,000 more applications would have been accepted.

In order to further improve the model, other options may be considered in future analyses. Models using fewer variables resulted in limited improvement. However, balancing and scaling the data may result in improved results for both the current number of variables as well as using fewer variables. Furthermore, expanding the timeframe of data for all sets, training, testing, and particularly out of time would allow us to better identify models that will be strong predictors of future fraud. This would also give us a more definitive answer on whether the Gradient Boosting model approach did indeed produce the strongest model.

Finally, working with the project stakeholders to develop an in-depth understanding of the implications of declaring an application fraudulent in this context could help us build a model that better achieves those goals. For example, depending on the product the application is for, rejecting a good application may result in a loss of potential future earnings that would be associated with a customer's lifetime value. As a result, we might've recommended an even lower threshold in order to minimize the number of good applications rejected. However, more information provided by the domain expert managers would be required in order to better assess the consequences with rejecting a good application.

Appendix

Data Quality Report

Table of Contents

1. Introduction

This Data Quality Report ("DQR" or "Report") is a review of product application data ("Application Data") provided by Professor Stephen Coggeshall ("Professor") for the University of Southern California course, DSO 562: Fraud Analytics ("DSO562"). Although the origin and time of the data is kept unknown for anonymity, the data represents applications within calendar year 2016 (January 1, 2016 to December 31, 2016).

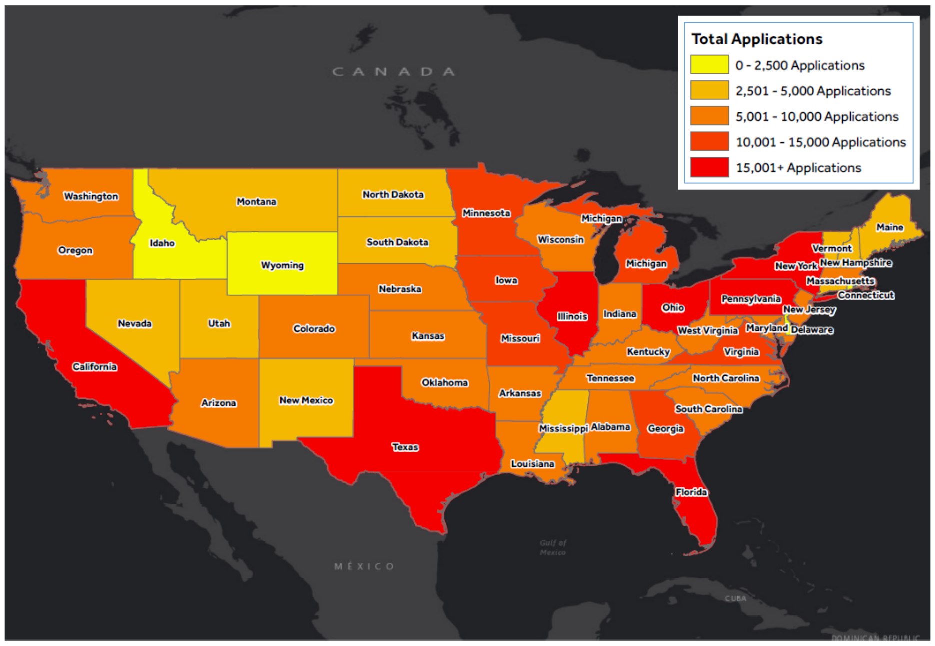
1. Purpose of Collecting Application Data

Data on applications are collected by any company or industry that needs to create a user profile or pull information from a user's history in order to do business. By taking in personal information such as social security, name, address, and/or birthday, it is possible to begin assessing whether the applicant is the person they claim to be. This can be very useful in reducing cases of identity fraud, which saves both companies and potential victims time and money.

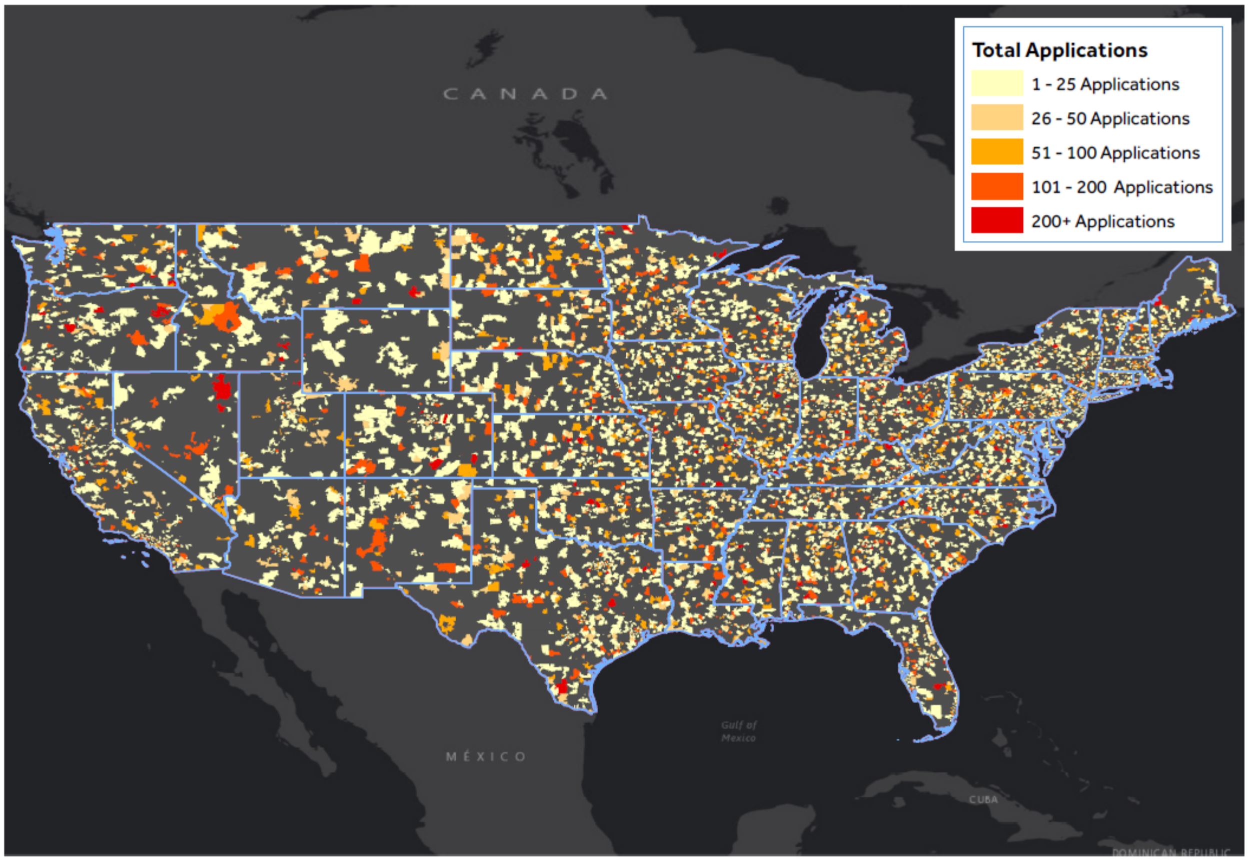
2. Data Summary

The Application Data is a record of 1,000,000 applications made in the calendar year of 2016. It contains ten (10) categorical fields that help to either identify a unique application, application, and whether the application was fraudulent. Please see **Maps 1 and 2** on the following pages for a density map of the concentration of applications by state and Zip Code, respectively. Due to limitations regarding the data and the requirements of DSO562, all maps in the Report have been drawn based on existing data prior to cleaning. For more information regarding the Mapping process, please see **Exhibit 1**.

Map 1: Density Map of Applications by State in the United States



Map 2: Density Map of Applications by ZIP Code in the United States



2. Field Summary

As mentioned in Section 1, there are ten (10) fields within the Transaction Data.

A. Aggregate Summary

Below is **Table 1**, showing a summary of each field's field type, number of records, percent populated, count of unique values, count of records with the value zero (0), and the percentage with the value zero (0).

Table 1: Aggregate Summary

Field	Field Type	Records	% Populated	Unique Values	Records with value zero	% with value zero
record	Categorical	1,000,000	100.00%	1,000,000	0	0.00%
Date	Categorical	1,000,000	100.00%	365	0	0.00%
ssn	Categorical	1,000,000	100.00%	835,819	0	0.00%
firstname	Categorical	1,000,000	100.00%	78,136	0	0.00%
lastname	Categorical	1,000,000	100.00%	177,001	0	0.00%
address	Categorical	1,000,000	100.00%	828,774	0	0.00%
Zip5	Categorical	1,000,000	100.00%	26,370	0	0.00%
dob	Categorical	1,000,000	100.00%	43,673	0	0.00%
homephone	Categorical	1,000,000	100.00%	28,244	0	0.00%
Fraud_label	Categorical	1,000,000	100.00%	2	985,607[1]	98.56%

[1] Although 985,607 tuples contain zero (0) in the Fraud field, a zero (0) indicates that the transaction was not an instance of fraud

As stated in **Section A** and **Table 1**, all the fields in the Application Data are categorical fields. Below is **Table 2**, with a summary showing the number of unique values, the most common entry, the count of the most common entry, and its percentage of the total data set.

Table 2: Unique Field Summary

Field	Records	Unique Values	Most Common Entry	Count of Entry	Percentage of Total
record	1,000,000	1,000,000	N/A[1]	N/A[1]	N/A[1]
Date	1,000,000	365	20160816	2,877	0.29%
ssn	1,000,000	835,819	999999999	16,935	1.69%
firstname	1,000,000	78,136	EAMSTRMT	12,658	1.27%
lastname	1,000,000	177,001	ERJSAXA	8,580	0.86%
address	1,000,000	828,774	123 MAIN ST	1,079	0.11%
Zip5	1,000,000	26,370	68138	823	0.08%
dob	1,000,000	43,673	19070626	126,568	12.66%
homephone	1,000,000	28,244	9999999999	78,512	7.85%
Fraud_label	1,000,000	2	0	985,607	98.56%

[1] Since every entry is a unique entry, common entry and its associated fields are not applicable.

3. Field Details

The Application Data has ten (10) different fields. Below is a short description of every field including additional information such as graphs, tables, and/or maps to provide greater insights into the data.

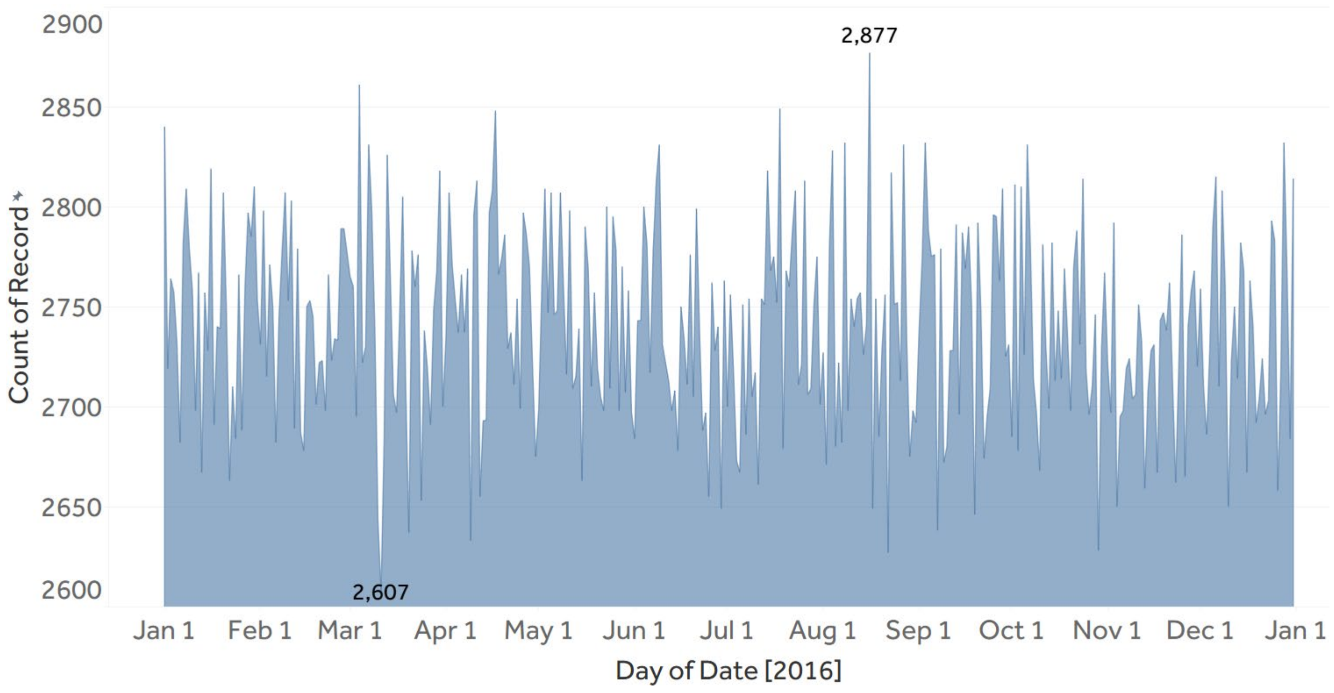
3.1. Recnum

Recnum is a nominal categorical field that provides a unique ordinal identifier for each tuple, which translates to 1,000,000 unique records. Since each tuple is assigned a unique identifier, the field follows a uniform distribution. A graph or table would not provide any greater insight into the data and thus, was not included as part of the report.

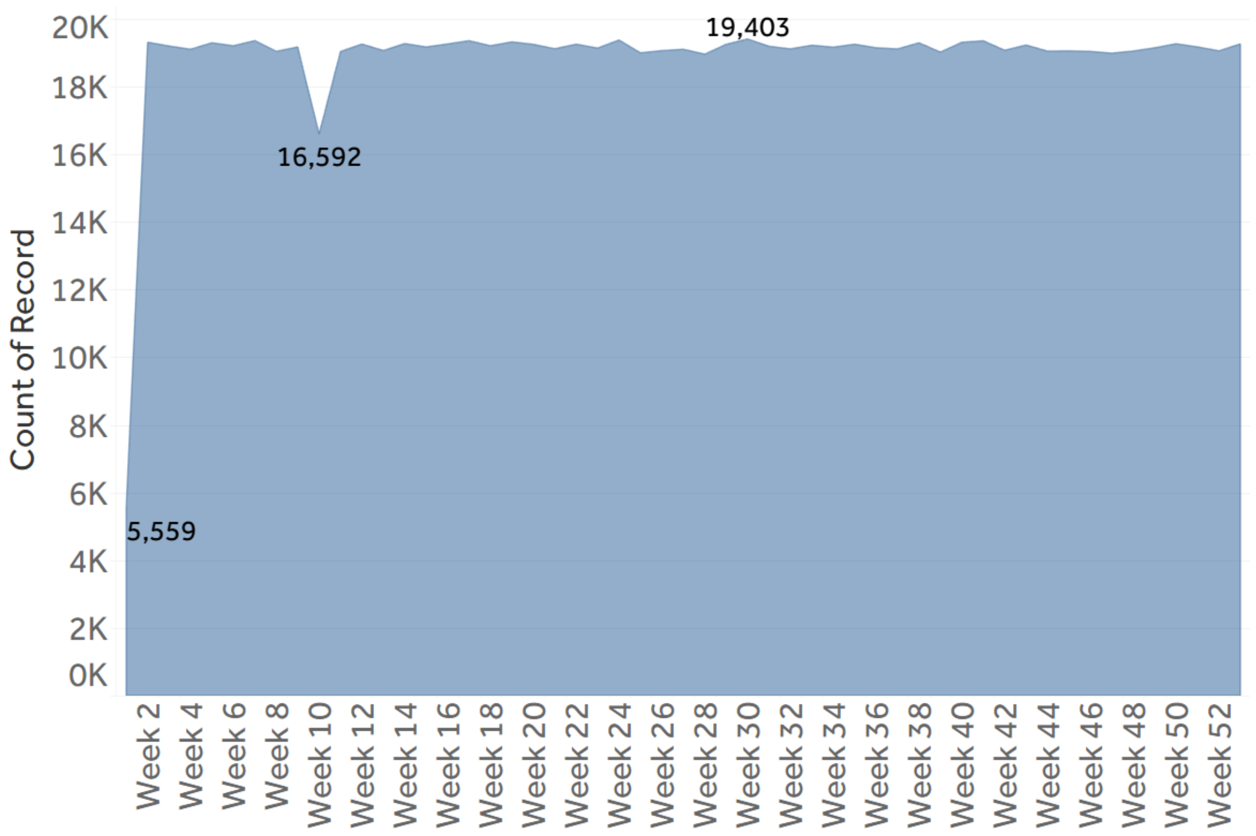
3.2. Date

Date is an ordinal categorical field used to identify the day, month, and year an application was made in the format of yyyymmdd. As shown on **Table 1**, there are 1,000,000 records across 365 days within the calendar year 2016. **Graphs 1 through 4** provide more information on transaction trends in 2010.

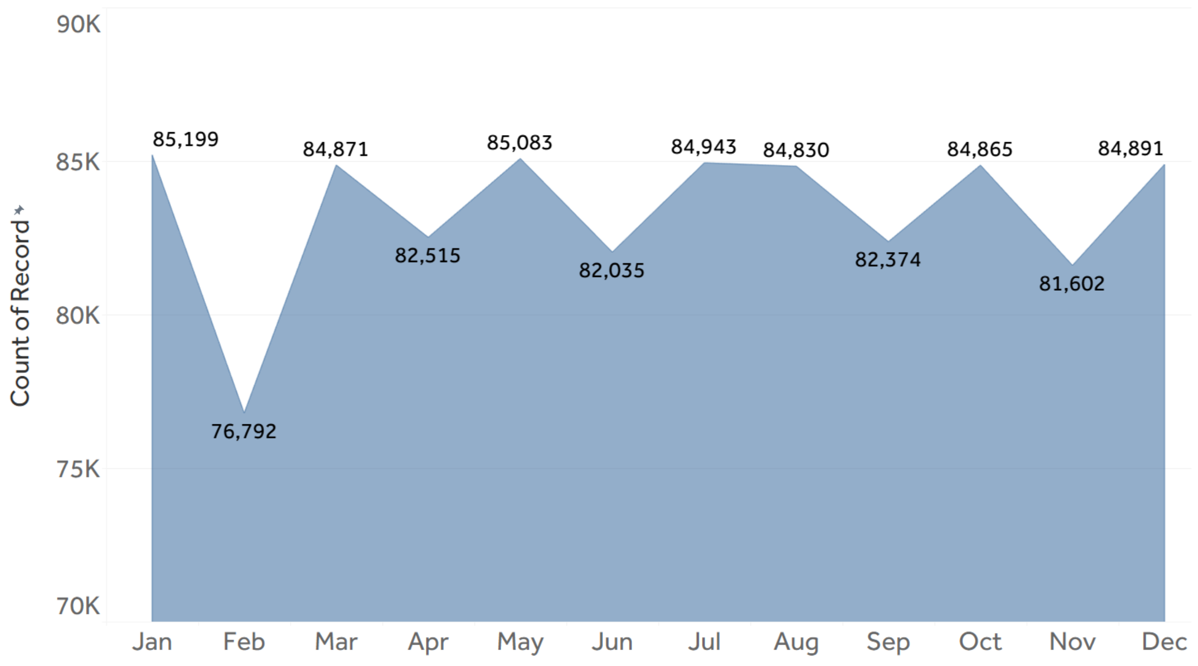
Graph 1: Total Application Count by Day in 2016



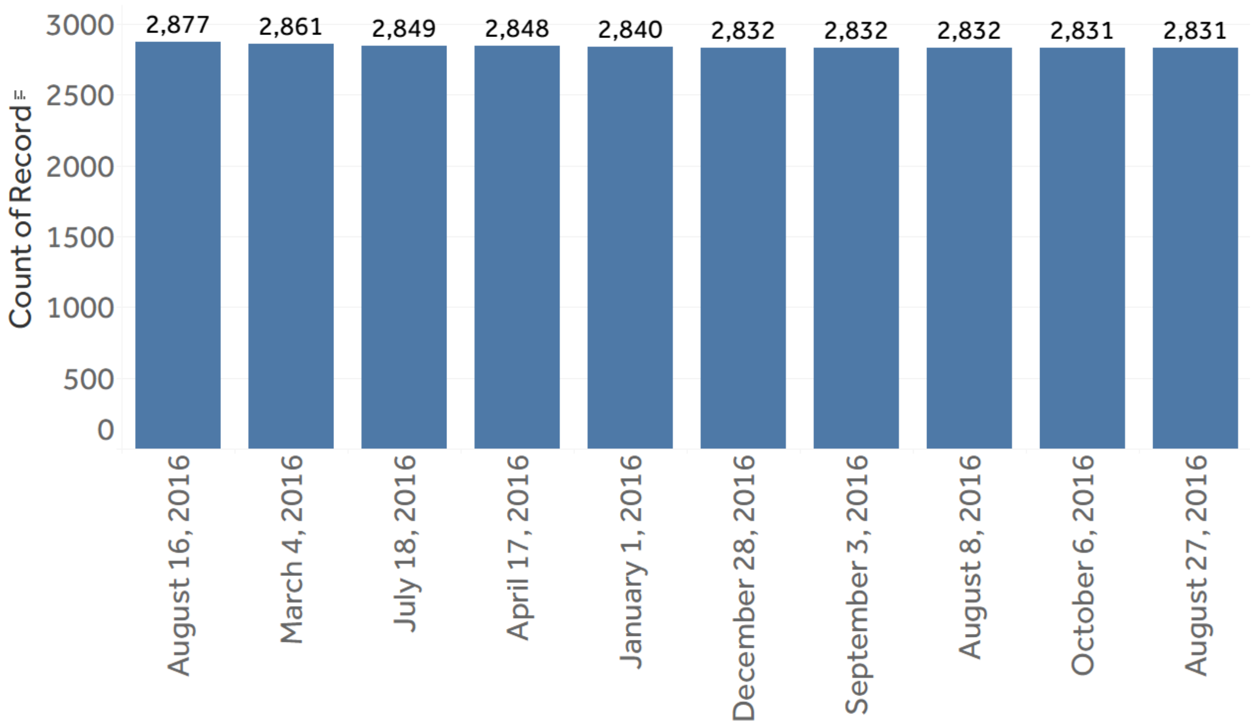
Graph 2: Total Application Count by Week in 2016



Graph 3: Total Application Count by Month in 2016



Graph 4: Top 10 days for Applications in 2016



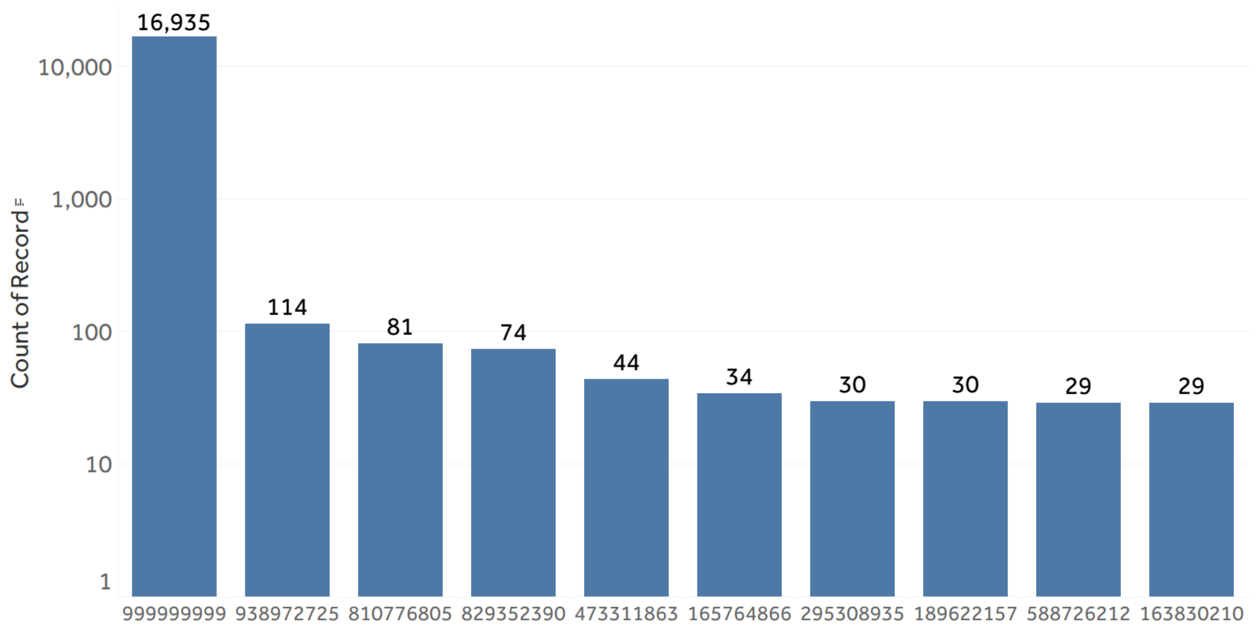
Based on **Graphs 1 and 2** on the prior page, there doesn't appear to be any daily or weekly cyclical relationship for when applications were made. Interestingly, **Graph 3** shows a slight cyclical relationship where the number of applications made drops about every other month before coming back up again. However, more research is required to verify whether it is just a coincidence. In **Graphs 1, 2, and 3**, there appears to be a sharp drop in applications within the first quarter of the year. More information is required to determine the cause for the decrease.

Lastly, based on **Graph 4** above, the date with the most applications is 20160816 (August 16, 2016) making up approximately 0.29% of the applications made in 2016. It does not appear that the date of the application has any anomalies.

3.3. SSN

Ssn is a categorical field that identifies the social security number for each applicant. As shown in **Table 1**, there are 1,000,000 records and among those records, there are 835,819 unique social security numbers. **Graph 5** below shows the top ten (10) applicants with the most applications.

Graph 5: Top 10 Ssn with the most applications in 2016

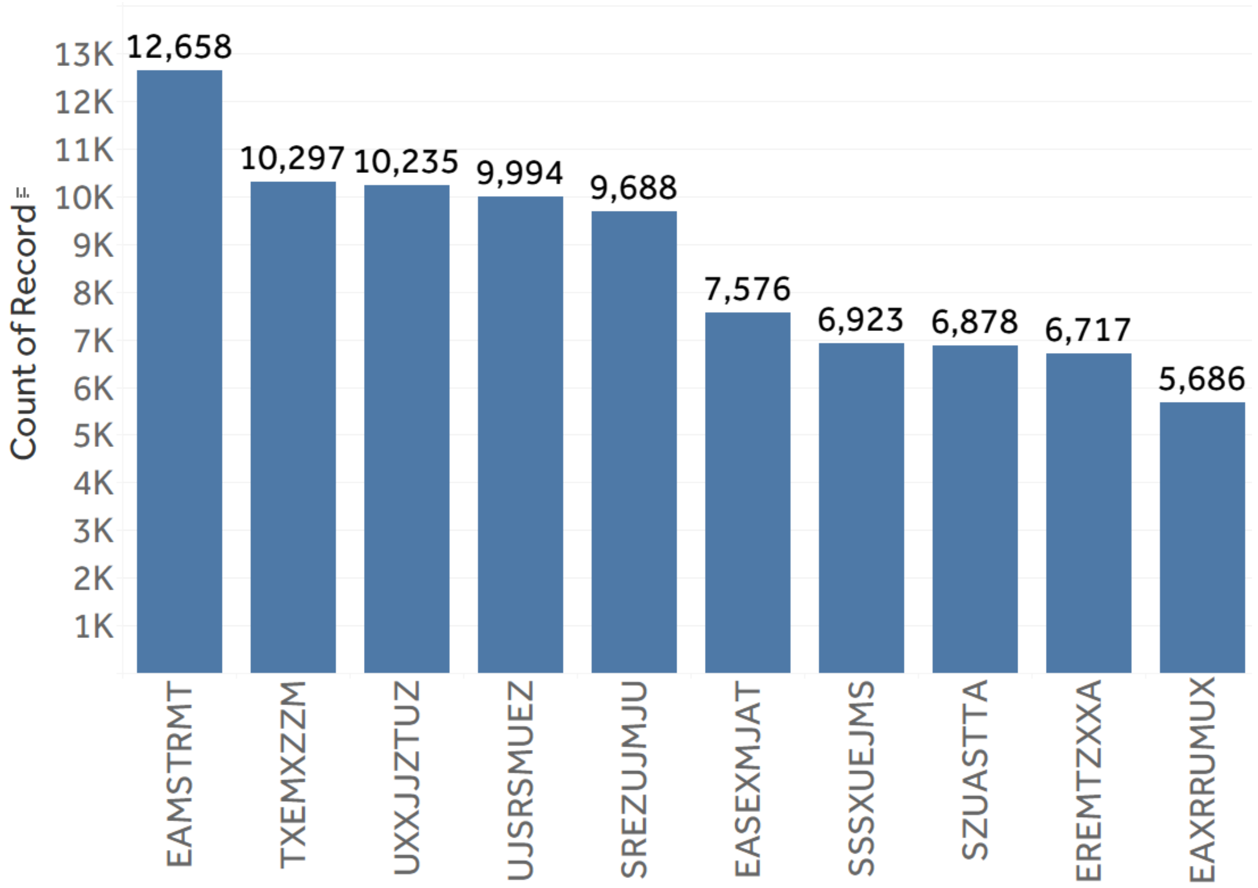


As shown in **Graph 5** above, the Ssn with the most applications is 999999999 which made up approximately 1.69% of the applications made in 2016. This is more than two (2) magnitudes greater than the Ssn with the second most applications.

3.4. Firstname

Firstname is a categorical field for the first name of the applicant of the application. As shown in **Table 1**, there are 1,000,000 records with a first name and of those records, there are 78,136 unique first names. **Graph 6** below shows the top ten (10) first names with the most applications.

Graph 6: Top 10 Firstnames With the Most Applications in 2016

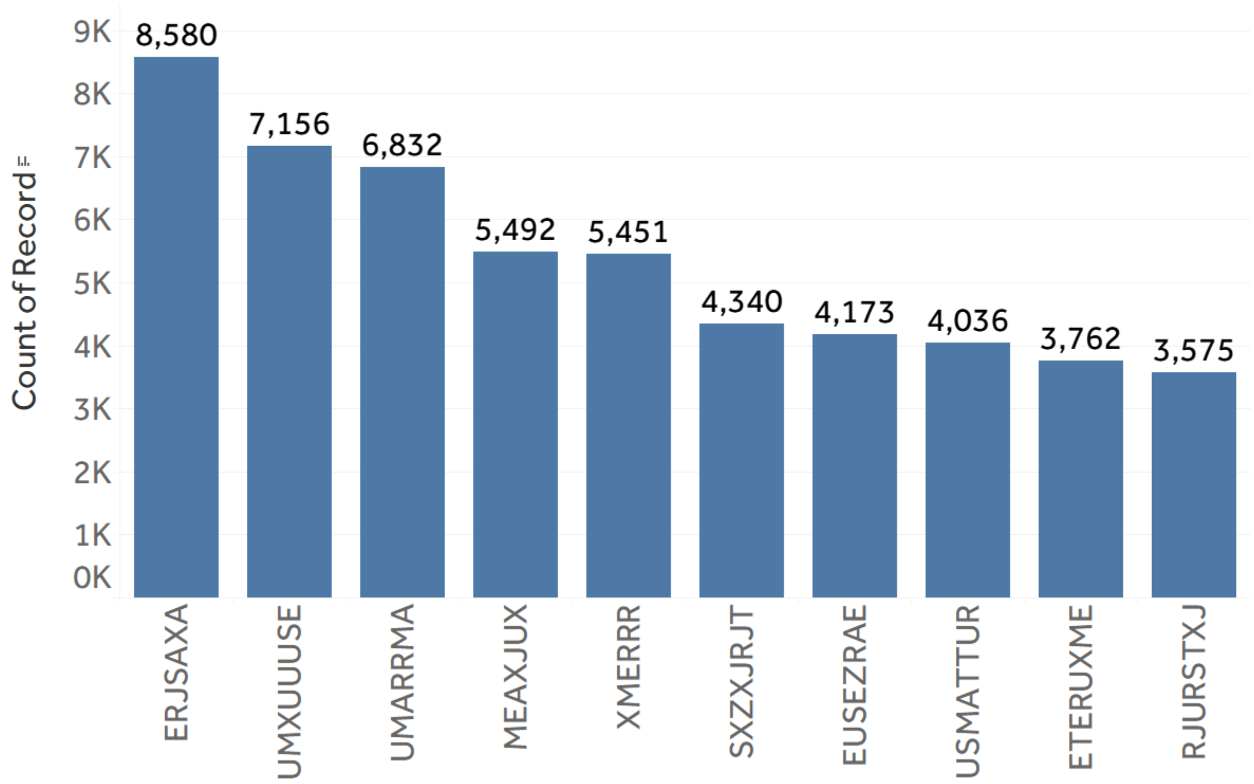


As shown in **Graph 6** above, EAMSTRMT is the first name that has the most amount of applications with approximately 1.27% of the applications made in 2016.

3.5 Lastname

Lastname is a categorical field for the last name of the applicant of the application. As shown in **Table 1**, there are 1,000,000 records with a last name and of those records, there are 177,001 unique last names. **Graph 7** below shows the top ten (10) last names with the most applications.

Graph 7: Top 10 Lastnames With the Most Applications in 2016

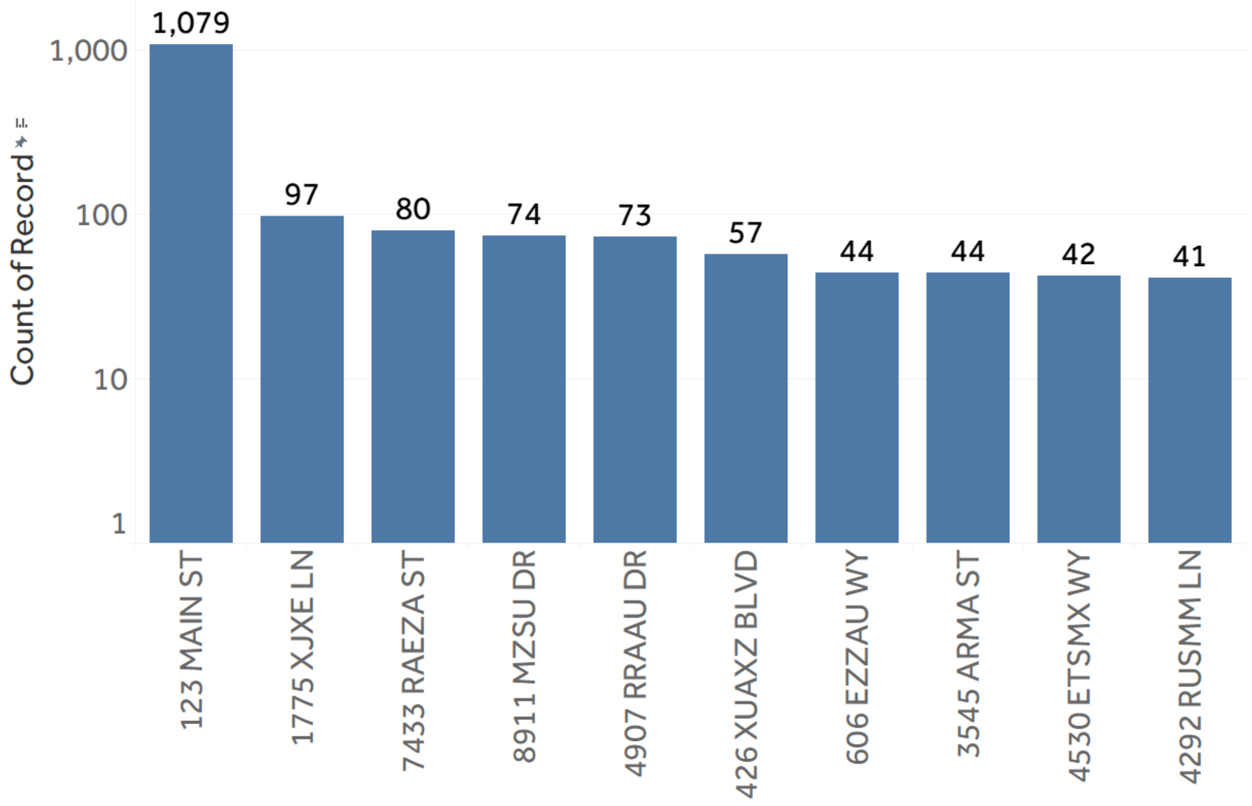


Based on **Graph 7** above, ERJSAXA is the last name that has the most amount of applications with approximately 0.86% of all applications made in 2016.

3.6 Address

Address is a categorical field for the mailing address of the applicant of the application. As shown in **Table 1**, there are 1,000,000 records with an address and of those records, there are 177,001 unique addresses. **Graph 8** below shows the top ten (10) addresses with the most applications.

Graph 8: Top 10 Addresses With the Most Applications in 2016

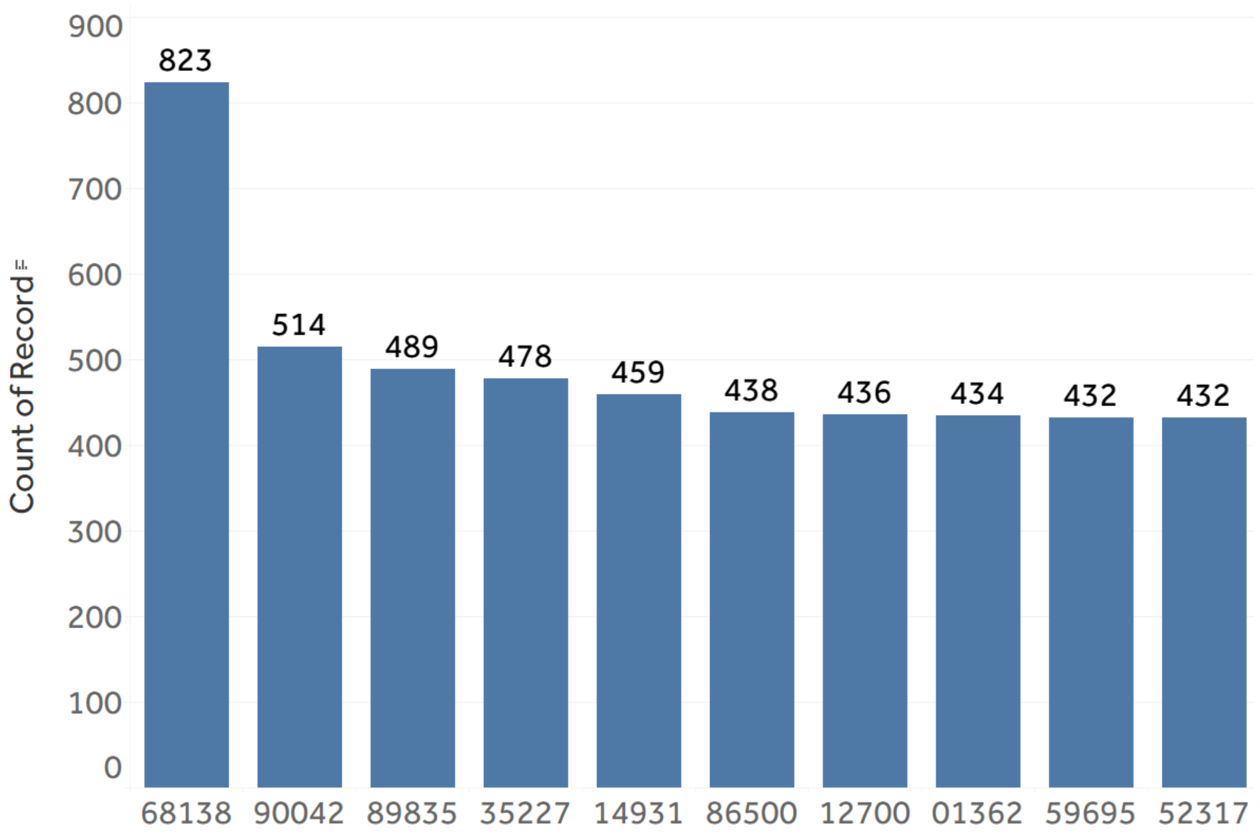


Based on **Graph 8** above, 123 MAIN ST is the address that has the most amount of applications with approximately 0.11% of all the applications made in 2016. It is about a magnitude greater than the second most common address, 1775 XJXE LN.

3.7 Zip5

Zip5 is a categorical field that identifies the ZIP Code for the mailing address of the applicant. As shown in **Table 1**, there are 1,000,000 records with a ZIP Code and of those records, there are 26,370 unique ZIP Codes. **Graph 9** below shows the top ten (10) ZIP Codes with the most applications. Please see **Map 2** on page 3 for a map of the concentration of applications by ZIP Code.

Graph 9: Top 10 ZIP Codes (Zip5) With the Most Applications in 2016

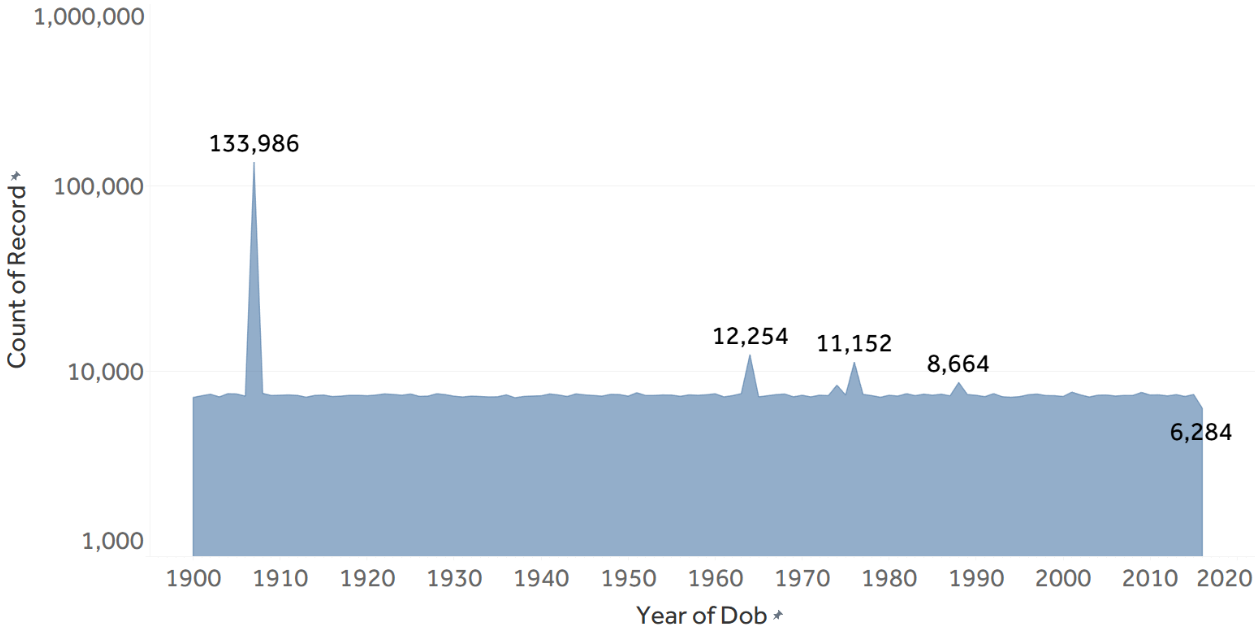


As shown in **Graph 9** above, ZIP Code 68138 has the most amount of applications making up approximately 0.08% of the applications made in 2016.

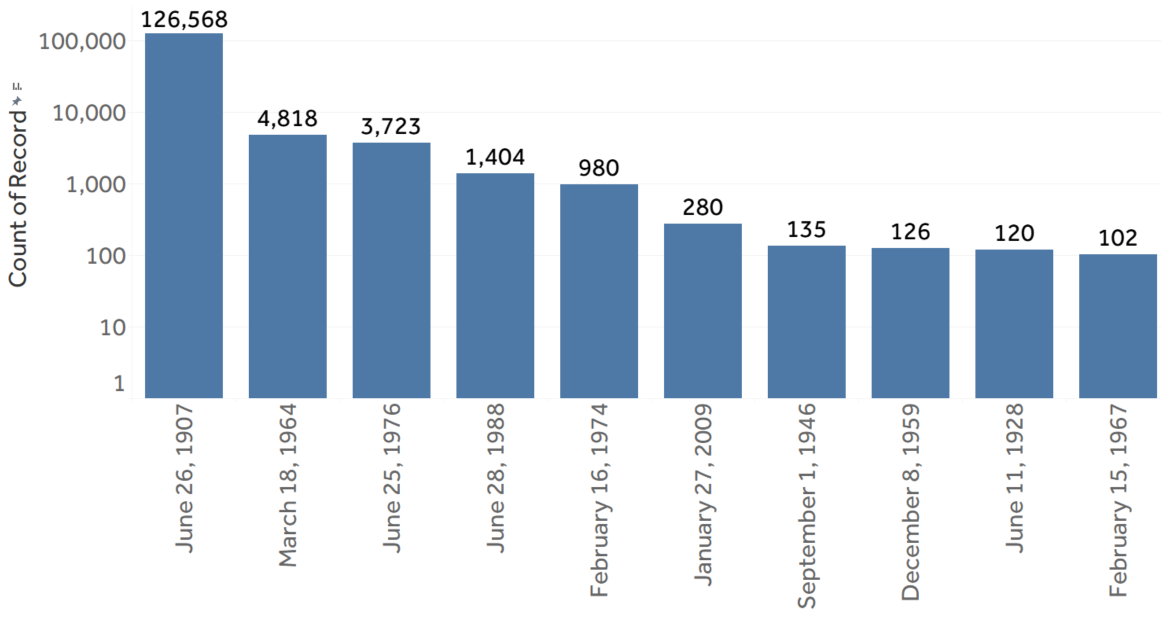
3.8 DOB

Dob is a categorical field used to identify the month, day, and year an application was made in the form of yyyymmdd. As shown in **Table 1**, there are 1,000,000 records with a date and of those records, there are 43,673 unique dates. **Graph 10** below shows the distribution of births between the years 1900 and 2017 while **Graph 11** shows the top ten (10) dates of birth with the most applications.

Graph 10: Distribution of Birthdays between 1900 and 2017 for Applications in 2016



Graph 11: Top 10 Dates of Birth (Dob) With the Most Applications in 2016

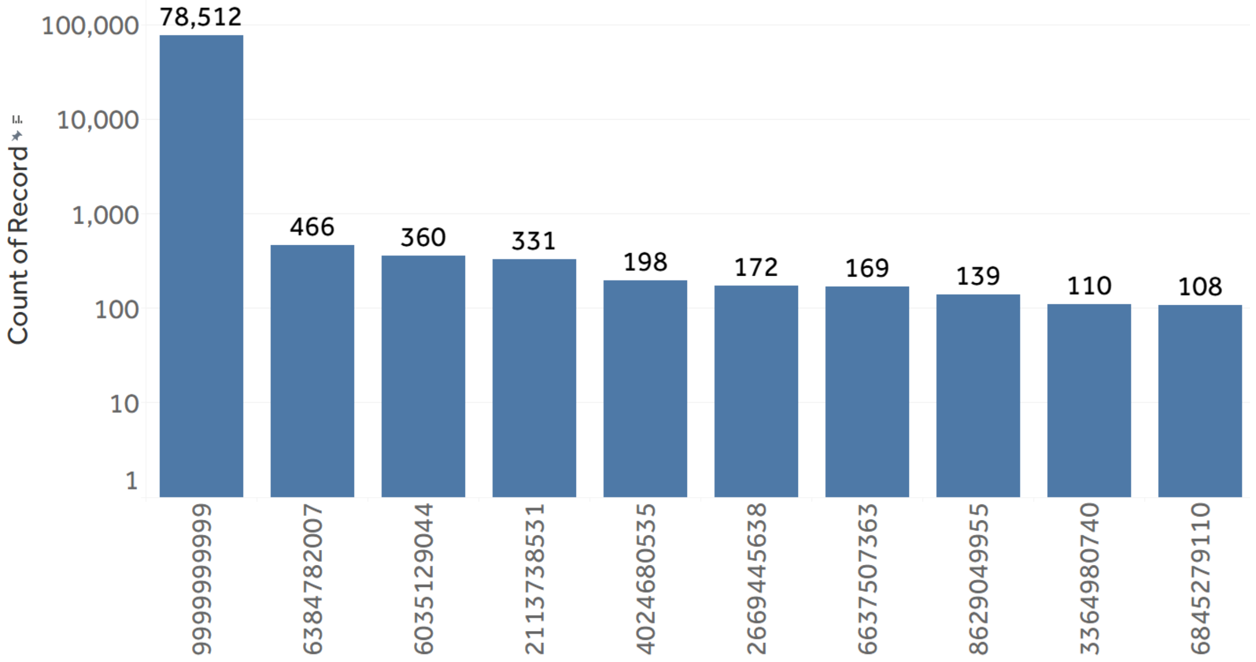


As show in **Graphs 10 and 11** on the prior page, there appears to be a massive spike in the date of birth for applications made around 1907. This is due to 19070626 have 126,568 applications making it 12.66% of all the applications made in 2016. Not only is the sheer number of applications made by this birthday suspicious, an individual borne on this date would've been approximately 109 years old. The probability of having many individuals at that age making applications or few individuals making many applications is unlikely. A closer look reveals that this birthday is associated with 107,467 Social Security Numbers, 18,149 first names, 40,335 last names, 111,901 Addresses, and 15,999 ZIP Codes. It is highly suspicious to have this many 109-year old applicants as they alone would make up approximately 12.97% of all the applicants (using Ssn). More research is required to investigate the possibility of sampling bias.

3.9 Homephone

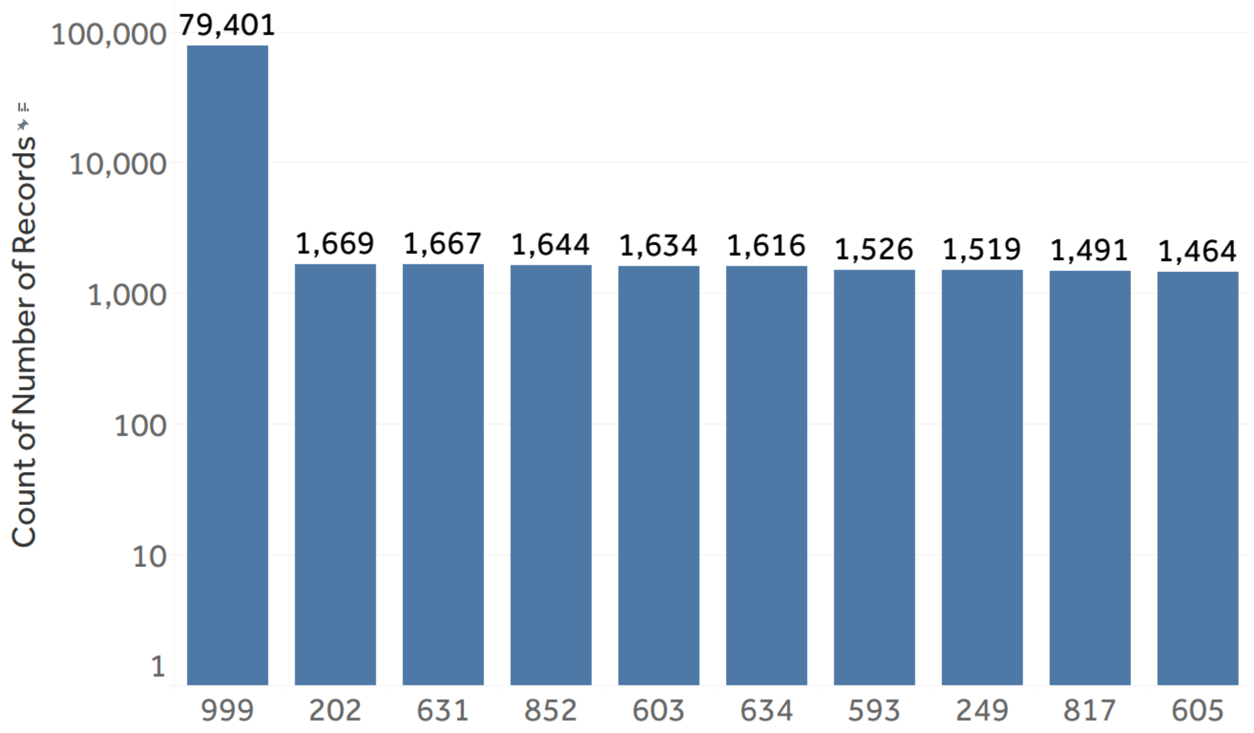
Homephone is a categorical field used to identify the phone number of the application. As shown on **Table 1**, there are 1,000,000 records and across those records, there are 28,244 unique phone numbers. **Graph 12** below shows the top ten (10) phone numbers with the most applications. Phone numbers are usually structured with ten (10) digits. However, it is assumed that 9-digit (or less) phone numbers in the data have a leading zero (0). The first three (3) digits of any phone number is assumed to be the area code for the applicant. **Graph 13** on the following page shows the top ten (10) area codes with the most applications. **Map 3** on page 19 shows a density map of the top area codes within the US (Please see **Exhibit 1** for more information regarding the mapping process).

Graph 12: Top 10 Phone Numbers (Homephone) With the Most Applications in 2016



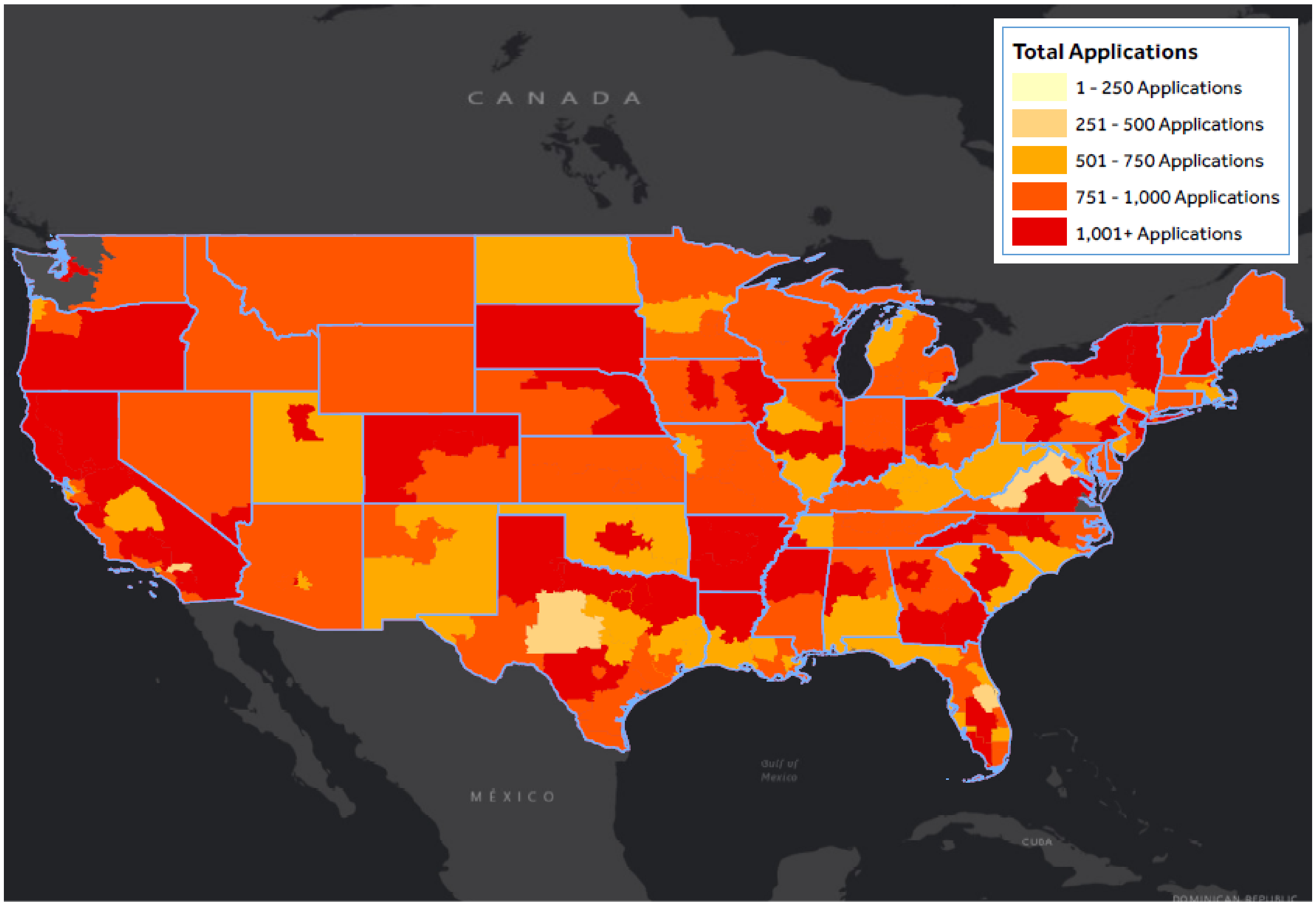
Based on **Graph 12** above, 9999999999 is the most common phone number, making up approximately 7.85% of all the applications. This phone number may have been entered as 9999999999 because it was left blank on the application.

Graph 13: Top 10 Area Codes With the Most Applications in 2016



Based on **Graph 13** above, there does not appear to be a bias with regards the area code outside of the area code 999. Of which, most of those entries were from 9999999999.

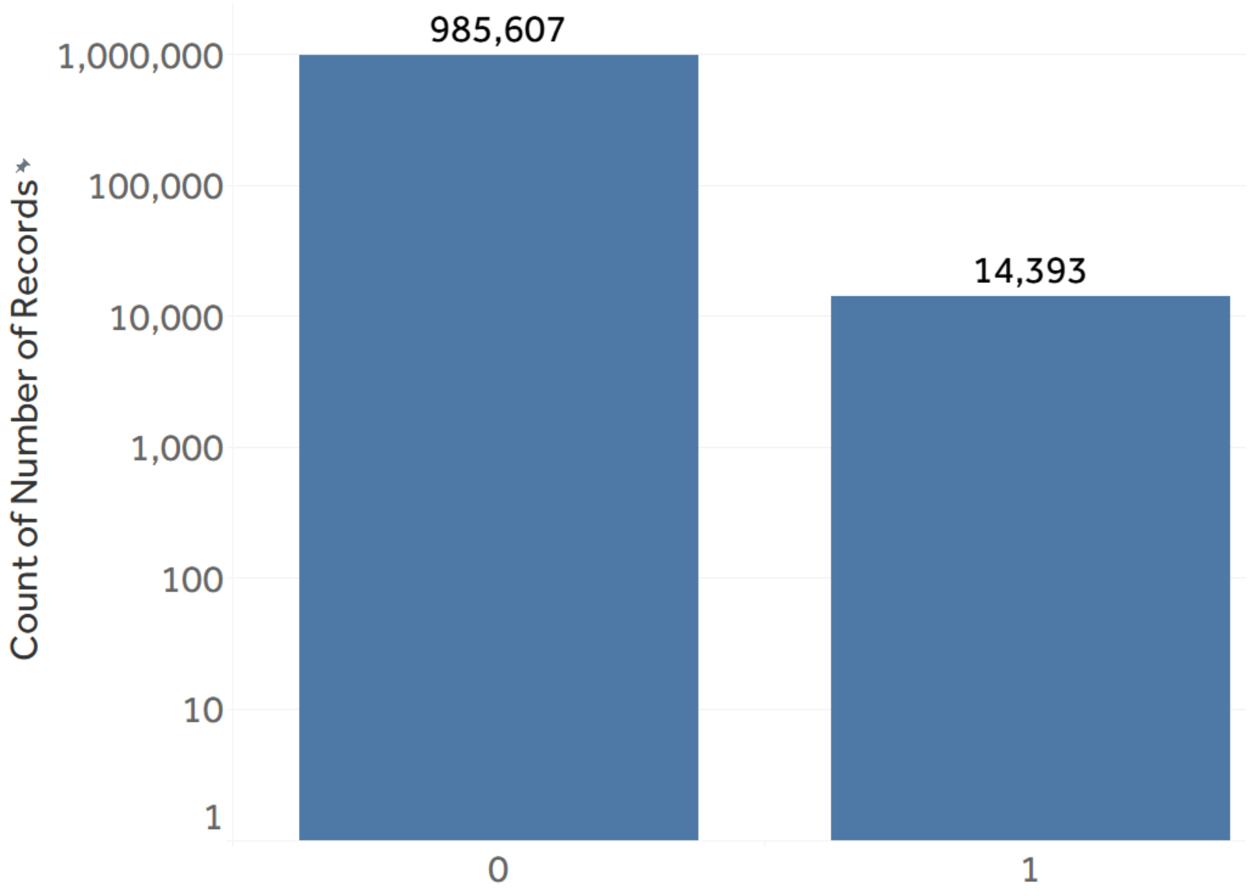
Map 3 - Density Map of Applications by Telephone Area Code in the United States



3.10 Fraud_label

Fraud_label is a categorical field that indicates whether an application was fraudulent (value = 1, the specificity of the type of fraud was not identified in the data). **Graph 14** below show the count of fraudulent and non-fraudulent applications.

Graph 14: Count of Fraud for Applications in 2016



Based on **Graph 14**, approximately 1.44% of applications in the year 2016 were fraudulent.

4. Exhibits

Exhibit 1: Information Regarding Density Maps

In order to map the application data by telephone area code, ZIP Code, and State, databases and shapefiles regarding the geographic regions above were required in ordered capture all regions of the United States accurately. Below is a list of the data sources acquired:

Telephone Area Code

- UCLA Geoportal. North America Telephone Area Code Boundaries. Dated Oct. 2015. Web. Apr. 3, 2019. https://apps.gis.ucla.edu/geodata/dataset/north-america-telephone-area-code-boundaries

ZIP Code

- federalgovernmentzipcodes.us Dated Jan. 2012. Web, Apr 3, 2019. http://federalgovernmentzipcodes.us/
    - This data set incorporates both active and decommissioned ZIP Codes from the US Postal Service, MPSA 2008 Election Ballot, IRS, and the National Weather Service
    - Unfortunately, this data set was last updated in 2012. However, it does include decommissioned ZIP codes, which may prove useful
- USA ZIP Code Areas. ESRI. Dec. 2017. Web. Apr. 3, 2019. https://www.arcgis.com/home/item.html?id=8d2012a2016e484dafaac0451f9aea24
    - This data set only contains the ZIP Codes used by the US Postal Service. Unlike the previous ZIP Code data set, this file was last updated in December 2017

As taken from ESRI, the following is an explanation to understanding the ZIP Code:

> "The first digit of a five-digit ZIP Code divides the United States into 10 large groups of states numbered from 0 in the Northeast to 9 in the far West. Within these areas, each state is divided into an average of 10 smaller geographical areas, identified by the second and third digits. These digits, in conjunction with the first digit, represent a sectional center facility or a mail processing facility area. The fourth and fifth digits identify a post office, station, branch or local delivery area."

Using all the information to match the ZIP Codes, less than half of the Application Data was matched to an actual ZIP Code (Approximately 43%). This may be due to the modifications made to the data in order to make it anonymous.

Exhibit 2: Sources

federalgovernmentzipcodes.us Dated Jan. 2012. Web, Apr 3, 2019. http://federalgovernmentzipcodes.us/

UCLA Geoportal. North America Telephone Area Code Boundaries. Dated Oct. 2015. Web. Apr. 3, 2019. https://apps.gis.ucla.edu/geodata/dataset/north-america-telephone-area-code-boundaries

Understanding Area codes. Verizon. 2019. Web. Apr. 3, 2019. https://www.verizon.com/support/residential/homephone/area-international-info/area-code-lookup

USA ZIP Code Areas. ESRI. Dec. 2017. Web. Apr. 3, 2019. https://www.arcgis.com/home/item.html?id=8d2012a2016e484dafaac0451f9aea24