USC Marshall

School of Business

DSO 562

Application Fraud Analysis

Team Member: Qian Huang, Junchu Zhang, Yu Liu, Xinjin Liu, Haoyan Zhang

Date: 03/21/2021

Table of Content

Executive Summary

Financial crime is a crime committed against property, involving the unlawful conversion of the ownership of the property to one's own personal use and benefit. Credit card fraud is an inclusive term for fraud committed using a payment card, such as a credit card or debit card. The purpose may be to obtain goods or services, or to make payment to another account that is controlled by a criminal. Application fraud is a type of credit card fraud. Application fraud takes place when a person uses stolen or fake documents to open an account in another person's name. Criminals may steal or fake documents such as utility bills and bank statements to build up a personal profile. When an account is opened using fake or stolen documents, the fraudster could then withdraw cash or obtain credit in the victim's name.

In this project, we have a credit card application information dataset with label marks for fraud cases. We will first look at the data summary in section 1. Then we will clean the data in section 2. After cleaning the data, we create candidate variables in section 3. In section 4, we will do feature selection and select final variables to build models. The model algorithms are explained in section 5. Finally, we will show our results and make conclusions in section 6 and section 7.

1. Data Description

Our dataset is Product Application Data, and we are aiming to detect the application fraud for credit cards, by analyzing corresponding fields such as social security number, date of birth, address, and home phone. The dataset is synthetic data to mimic real application data over 1 year: from 2016-01-01 to 2016-12-31. There are originally 9 fields excluding the record field, and a total of 1,000,000 records (see Table 1.1).

1.1 Summary (Categorical) Table

Table 1.1 Statistics of Each Field

Column Name	# of Records	% Populated	Unique Values	Most Common Field Value
date	1000000	100%	365	2016-08-16
ssn	1000000	100%	835819	999999999
firstname	1000000	100%	78136	EAMSTRMT
lastname	1000000	100%	177001	ERJSAXA
address	1000000	100%	828774	123 MAIN ST
zip5	1000000	100%	26370	68138
dob	1000000	100%	42673	1907-06-26
homephone	1000000	100%	28244	9999999999
fraud_label	1000000	100%	2	0

1.2 Field Examples

1.2.1 Field “Address”

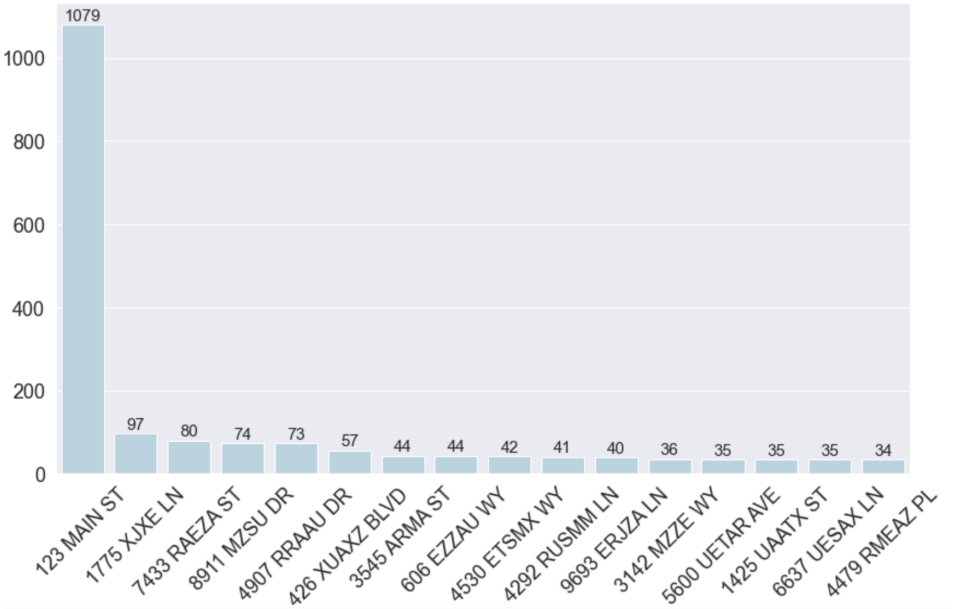


Figure 1.2.1 Histogram of Field “Address”

1.2.2 Field “SSN”

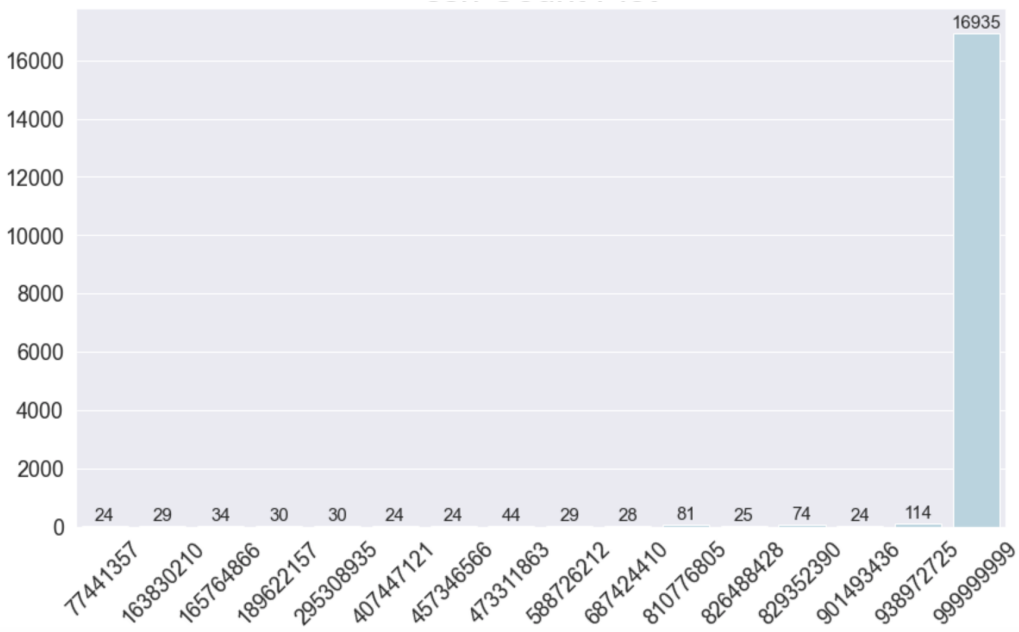


Figure 1.2.2 Histogram of Field “SSN”

1.2.3 Field “Homephone”

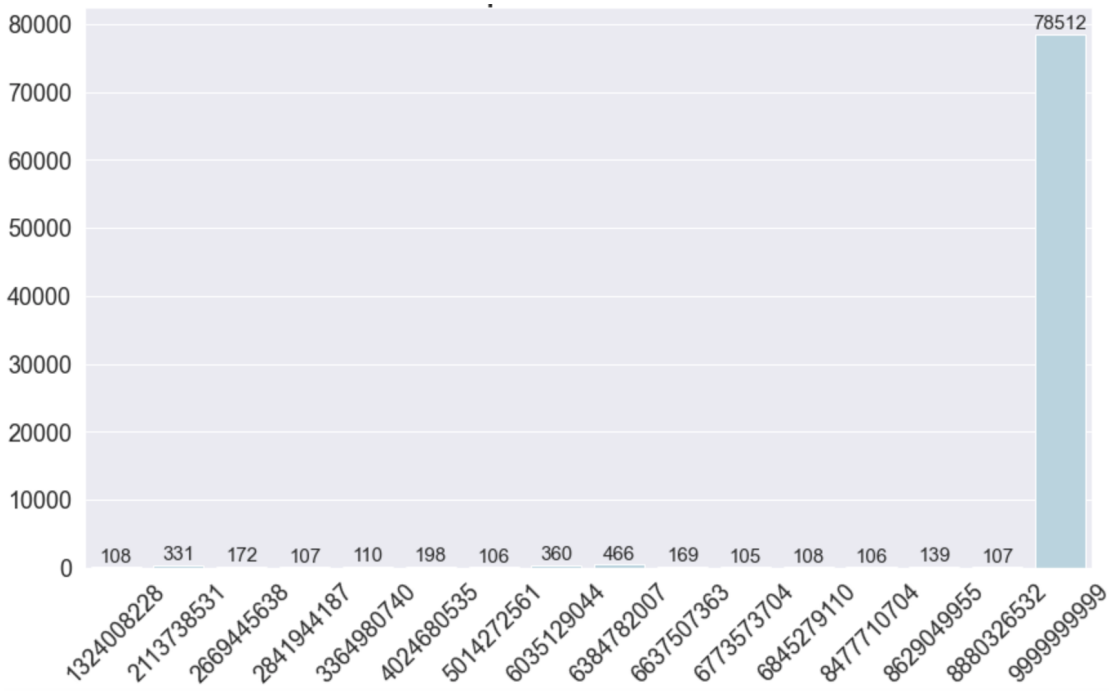


Figure 1.2.3 Histogram of Field “Homephone”

2. Data Cleaning

From the data description, we found that there are no missing values that need to be filled in. In this way, we move on to the other data cleaning steps such as fixing data type and fixing frivolous values.

We first changed the format and data type of the “date” field. The original “date” field is numeric with a “YYYYMMDD” format. We changed the field to string, applied the “YYYY-MM-DD” format, and changed it again to the DateTime format to prepare for the train-test-OOT(out-of-time) set split in the future steps. Another field that needs formatting is “zip5”, including 104526 records with zip codes less than 5 digits. We filled the zip codes less than 5 digits with 0s in the front of each record.

From the Data Quality Report attached in Appendix A, we have found that there are many values that seem unreasonable. In the “dob” field, the most frequent value is “19070626” which appeared 126568 times. It is nearly impossible for many 109-year-old people to make more than 300 transactions a day on average using credit cards. The “address” field also has a very common value of “123 MAIN ST” as a general representation of unknown addresses. Besides, in the “ssn” and “homephone” field there are records of “999999999” and “9999999999”, which are not considered valid social security numbers and phone numbers. These values may represent the unknown missing values created by the system that need to be fixed before model training to avoid correlations among records. We assume that each record is unique and adjust the frivolous “ssn”, “dob”, and “homephone” records with the 0s at the beginning and record number combined. For the “address” field, we used the format of “record number + RECORD”.

Table 2 illustrates the data cleaning step for each variable.

Table 2 Data Cleaning Step

Variable	Before Cleaning	After Cleaning
date	“20160101”	“2016-01-01”, DateTime format
zip5	Zip code below 5 digits, such as “1234”	Fill with 0s ahead, “01234”
dob	“19070626” frivolous value	“000012345”, “12345” as record number
address	“123 MAIN ST” frivolous value	“12345RECORD”, “12345” as record number
ssn	“9999999999” frivolous value	“0000012345”, “12345” as record number
homephone	“999999999” frivolous value	“000012345”, “12345” as record number

3. Candidate Variable Creation

Candidate variable creation and selection are two critical steps before the model fitting and will significantly influence the model performance. We consider that the 9 dependent variables in the original data are insufficient to build a strong model and we need to create as many relevant variables to the fraud detection as possible to prepare for the feature selection.

We created our variables based on our understanding of fraud conduction in credit card applications. Usually, fraud is conducted by stealing identity information or using leaked identity information. Fraudsters will even combine information from different persons to apply them in multiple fraud conductions. If the same combination of information is abnormally frequently appeared in credit card applications(for instance, the “same person” applies for credit card 100 times a day), it’s more likely that the record is considered fraudulent. In this way, we build the variables in several ways of indicating frequency.

We first combined the “lastname” and “firstname” variables into a new variable “name”, and “address” and “zip5” into “full address”. A single last name or first name and an address are insufficient to identify a person or a place as unique because there are so many people with the same name and so many same addresses in different cities. We want the name and address to be less ambiguous and to point out more to the individual. We also created another variable “age” subtracting “dob” from the year 2016.

It’s still possible that there are people with the exact same full name, and members in a big family living in the same place. However, it’s less likely that there are multiple people with the same name and birthday, and much less likely that there are multiple people with the same name and birthday and same phone number. In that case, it’s more convincing that the records with the same combinations of information are from the same “person” that applies for the credit card multiple times. Thus, we combined the existing 11 variables by two and three of them together to create another xxx variable and further identify the same “person”.

We used four ways to produce the frequency of our identity combinations and produce four other groups of variables.

- **Day Since**
  Day since the same record of identity combination was seen. We assume that if the record is seen more recently last time, it will have a higher possibility of conducting fraud.
- **Velocity**
  Measured by the number of same records that appeared in the past 0, 1, 3, 7, 14, and 30 days. It’s rare for a single person to apply for a credit card too many times in a short

period of time. Thus, if more records are seen in a certain time period, it's more likely that the person is conducting fraud.

- **Relative Velocity**
  Calculated by the number of records that appeared in the most recent past(0 and 1 day) divided by the average daily appearance of the record in the past 3, 7, 14, and 30 days(Total appearance in the past 3, 7, 14 and 30 days divided by number of days). There are cases of people in a family or in the same office building location using the same phone number and address. These cases might be treated as fraud by simply looking at the velocity variable. We created these variables to compare the velocity in the recent past to the velocity before to check if the appearance of certain information is constant(close to 1). Fraud is more likely to occur in the cases that certain information appears suddenly with a lot of records.
- **Uniqueness**
  Calculated by counting the unique number of other identity combinations for a particular identity combination. As is mentioned in the previous session, fraudsters may combine the information of different individuals to conduct multiple frauds, and one particular name or phone number may appear in combination with different addresses . We expect the non-fraud records to have fewer or only one unique number of other identity combinations for a certain identity combination.

Besides, we figured out the day of week for each transaction and created a "dow" variable. We used target encoding to compute the relative risk of fraud on a particular day of week and named the variable "dow risk".

The final output includes 427 variables. The full list of variables is appended in the Appendix II part of this report.

4. Feature Selection

Having too many features in the data will increase the computational cost of modeling and decrease the accuracy of model performance. So feature selection, which is the process of selecting features that contribute most to the output of interest, is a very important step before model building and model selection.

In this project, filter-based and wrapper-based feature selection methods are both used, and eventually 30 final variables are selected.

4.1 Preparation for Feature Selection

Before feature selection, the following steps are taken:

I. **Separate Data Set into In-Sample & Out-Sample:** Only use "past" data, the in-sample data that includes January to October for feature selection, leave out-of-time data for model testing.

II. **Discard the first two weeks' data:** avoid biased variables created such as *Days Since* because of insufficient previous data records

4.2 Filter

In this project, univariate Kolmogorov-Smirnov (KS) Score and Fraud Detection Rate (FDR) are used to rank the 427 candidate variables created. Then, the average ranking of these two scores are used to choose top 80 variables for the further feature selection step.

4.2.1 Univariate Kolmogorov-Smirnov (KS)

For a binary classification, the importance of a variable can be measured by how well it separates bad records from good ones. Figure 4.2.1 shows the curve of bad records against the good ones. The more different the curves are, the better the variable is to predict the final labels.

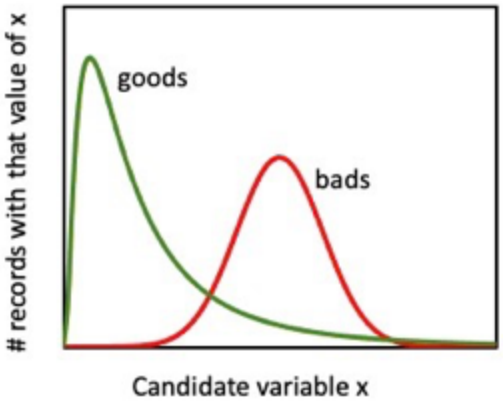


Figure 4.2.1 Illustration of the KS Score

Univariate Kolmogorov-Smirnov (KS) is a statistical measure of how well the two distributions are separated. It calculates the maximum distance between the cumulative goods and bads. The score's formula is shown as below:

$$KS = \max_{x} \int_{x_{min}}^{x} [P_{\text{goods}} - P_{\text{bads}}]\, dx$$

4.2.2 Fraud Detection Rate (FDR)

The second filter-based feature selection method is Fraud Detection Rate (FDR), calculating how much percentage of all the records can be caught at a particular examination cutoff location. For example, FDR 50% at 5% means this predictive model can catch 50% of all the frauds if specific 5% of the population are rejected and in this case, 5% is a rejection rate.

In this project, FDR at 3% is calculated as the percentage of total frauds in the top 3% or bottom 3% records of the total population sorted by the value of variable X.

4.2.3 Average KS and FDR rankings

According to KS and FDR scores, two rankings are given to candidate variables separately. To make the final selection of features more accurate, the average of the two scores is used as the final filter score and the top 80 variables with highest ranking scores are kept.

$$final\ score = \frac{KS\ ranking + FDR\ at\ 3\%\ ranking}{2}$$

4.3 Wrapper

Filter methods often evaluate features individually. In that case, some variables can be useless for prediction in isolation, but they can be quite useful when combined with other variables. To prevent that issue, a wrapper method is used here to select the best feature subsets. Wrapper methods work by evaluating a subset of features using a machine learning algorithm that employs a search strategy to look through the space of possible feature subsets, evaluating each subset based on the quality of the performance of a given algorithm.

In this project, Recursive Feature Elimination (RFE), a wrapper-type feature selection algorithm, which searches for a subset of features by starting with all features in the training dataset and successfully removing features until the desired number remains, is used. To save computation time, a logistic regression algorithm is wrapped by RFE and ROC-AUC score is used as the performance metrics.

The final selected features are listed below:

Table 4.3 Final Selected Features

Ranking	Variable	Variable Description
1	fulladdress_count_1	number of records with the same fulladdress over the same day
2	fulladdress_count_30fulladdress_count_30	number of records with the same fulladdress over the past 30 days
3	fulladdress_count_7	number of records with the same fulladdress over the past 7 days
4	name_count_14	number of records with the same name over the past 14 days
5	name_dob_count_14	number of records with the same name & date of birth pair over the past 14 days

6	name_dob_count_30	number of records with the same name & date of birth pair over the past 30 days
7	name_dob_count_7	number of records with the same name & date of birth pair over the past 7 days
8	name_dob_ssn_count_14	number of records with the same name & date of birth & ssn pair over the past 14 days
9	name_ssn_count_14	number of records with the same name & ssn pair over the past 14 days
10	ssn_count_14	number of records with the same ssn over the past 14 days
11	ssn_count_7	number of records with the same ssn over the past 7 days
12	ssn_dob_count_30	number of records with the same ssn & date of birth over the past 30 days
13	fulladdress_homephone_count_14	number of records with the same fulladdress & homephone pair over the past 14 days
14	fulladdress_count_0_by_3	the ratio of number of fulladdress seen in the recent past over number of fulladdress seen in the past 3 days
15	fulladdress_count_0_by_30	the ratio of number of fulladdress seen in the recent past over number of fulladdress seen in the past 30 days
16	ssn_count_0_by_14	the ratio of number of ssn seen in the recent past over number of ssn seen in the past 14 days
17	ssn_count_0_by_30	the ratio of number of ssn seen in the recent past over number of ssn seen in the past 30 days
18	name_dob_ssn_count_0_by_30	the ratio of number of name & date of birth & ssn seen in the recent past over number of same pair seen in the past 30 days
19	name_ssn_count_0_by_30	the ratio of number of name & ssn seen in the recent past over number of same pair seen in the past 30 days
20	fulladdress_homephone_count_0_by_30	the ratio of number of fulladdress & homephone seen in the recent past over number of same pair seen in the past 30 days
21	fulladdress_homephone_day_since	number of days since last saw the same fulladdress &_homephone pair

22	name_dob_day_since	number of days since last saw the same name & date of birth pair
23	name_ssn_day_since	number of days since last saw the same name & ssn pair
24	ssn_dob_day_since	number of days since last saw the same ssn & date of birth pair
25	fulladdress_fulladdress_dob_unique_0	number of unique fulladdress & date of birth pair for the particular fulladdress in one day
26	fulladdress_name_fulladdress_unique_0	number of unique name & fulladdres pair for the particular fulladdress in one day
27	fulladdress_name_homephone_fulladdress_unique_0	number of unique name & homephone & fulladdress pair for the particular fulladdress in one day
28	fulladdress_name_ssn_fulladdress_unique_0	number of unique name & ssn & fulladdress pair for the particular fulladdress in one day
29	fulladdress_ssn_fulladdress_unique_0	number of unique ssn & fulladdress pair for the particular fulladdress in one day
30	fulladdress_ssn_homephone_fulladdress_unique_0	number of unique fulladdress & ssn & homephone pair for the particular fulladdress in one day

5. Model Algorithm

We considered our fraud detection problem as a binary classification problem which identifies the fraud and non-fraud labels. We selected four classification models to fit and train our data, including logistic regression, neural network, random forest and gradient boosted tree. We will explain the construction process, algorithms used and results in this section.

5.1 Preparation for Model Algorithm

Our selection of modeling data is from January 15th to October 31th, and out-of-time data is from November 1st to December 31st.

Before Model Algorithm, the following steps are taken:

I. **Split data:** We split the modeling data into two parts, 70% training data to build the models, and 30% testing data to evaluate the models.

II. **Z-scale:** We performed z-scale to get rid of the extreme variate records which have z-score over 6. This step would help the model focus on the decision region of the score.

III. **Balance Data:** We tackled imbalance data by oversampling using SMOTE(Synthetic Minority Oversampling Technique). The method works by selecting examples that are close in the feature space, drawing a line between the examples in the feature space and drawing a new sample at a point along that line, which will help the model to learn the decision boundary.

IV. **Cross Validation:** We used 10-fold cross validation to evaluate estimator performance. The algorithm runs 10 times total, for each iteration, 9 folds are used to train the model, and 1 fold is for validation. We evaluate the model performance by averaging the score of each iteration. We choose average FDR at 3% as our major model performance measurement.

5.2 Models

We will introduce the four binary classification models we used in the following sections.

5.2.1 Logistic

The logistic method is a supervised predictive algorithm using independent variables to predict the dependent variable, like Linear Regression, but with a difference that the dependent variable

should be categorical variable. Logistic regression is mainly used in classification problems. It uses Logistic function to model the conditional probability.

We used the scikit-learn LogisticRegression package to fit the model with Python. We run the model with different numbers of independent variables to predict the fraud label. Among 5,10,15,20,25,30 candidate variables, we have 30 variables with the highest Out of Sample average FDR at 3%. Also, comparing 10, 15, 20, 25, the best number of variables to use is 10, which has a highest Out of Sample FDR at 3% of 0.494 (see Table 5.2.1).

Table 5.2.1 Hyper-parameters and Corresponding FDR at 3% of Logistic Regression Models

Logistic Regression	Number of Variables Selected	TRAIN FDR at 3%	TEST FDR at 3%	OOT FDR at 3%
1	5	0.407	0.412	0.387
2	10	0.514	0.514	0.488
3	15	0.507	0.508	0.482
4	20	0.506	0.507	0.480
5	25	0.506	0.508	0.485
6	30	0.513	0.515	0.494

5.2.2 Neural Network

A neural network is a computational learning system that uses a network of functions to understand and translate a data input of one form into a desired output, usually in another form. The concept of the artificial neural network was inspired by human biology and the way neurons of the human brain function together to understand inputs from human senses. The algorithm maps an input vector to an output scaler, typically a vector of axes into a single dependent variable y, by a series of hidden layers.

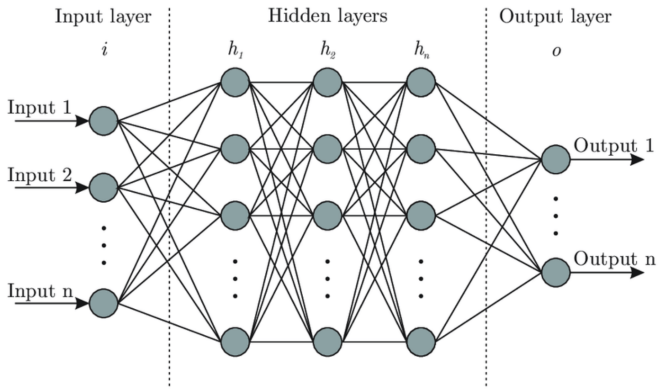


Figure 5.2.2 Illustration of Neural Network Model

We created a deep learning neural network model using Keras library. We used all 30 candidate variables to fit the model. And for each model, we set the layer parameter to 1, as it represents one hidden layer, so together it has one input layer, one hidden layer, and one output layer. The following is the group of parameter we experiment:

Epoch: An arbitrary cutoff, generally defined as "one pass over the entire dataset". Used to separate training into distinct phases. Useful for logging and periodic evaluation.
Units: Nodes in each layer

We modified the number of nodes, as well as the epochs. The parameter Node is a computational unit that has one or more weighted input connections, a transfer function that combines the inputs in some way, and an output connection. Nodes are then organized into layers to comprise a network. The parameter Epoch serves for tuning the times we want to retrain the model and catch improvement. After we tuned the hyperparameter Node for 30,60,90,120, and Epoch for 20,50, 100, we finalized the model to use Node 90, Epoch 50 that generates the highest FDR score of 0.531 (see Table 5.2.2).
Table 5.2.2 Hyper-parameters and Corresponding average FDR at 3% of Neural Network Models

Neural Net	Number of Variables	Layer	Node	Epoch	TRAIN FDR at 3%	TEST FDR at 3%	OOT FDR at 3%
1	30	1	60	50	0.542	0.545	0.528
2	30	1	90	50	0.546	0.550	0.531
3	30	1	30	20	0.545	0.549	0.527
4	30	1	120	50	0.547	0.548	0.530
5	30	1	90	100	0.546	0.550	0.529

5.2.3 Random Forest

Random forests are a type of ensemble method, meaning using an "average" of many base models to form a single, more accurate model. Each tree is built independently and is a strong, deep tree. Random Forest classifier is an ensemble method that trains several decision trees in parallel with bootstrapping followed by aggregation, jointly referred to as bagging.

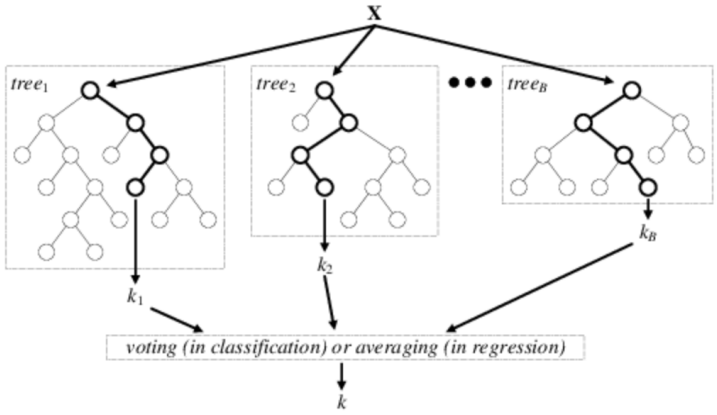


Figure 5.2.3 Illustration of Random Forest Model

Our team tried different numbers of trees, minimum sample split size, and minimum sample node sizes.The following is the detailed explanation of the parameter:

n_estimators*int:* The number of trees in the forest.
max_depth: The maximum depth of the tree. If None, then nodes are expanded until all leaves are pure or until all leaves contain less than min_samples_split samples.
min_samples_split: The minimum number of samples required to split an internal node
min_samples_leaf: The minimum number of samples required to be at a leaf node.

We finalized our best model with tried n_estimator as 400 and 500; maximum depth of the tree to 50; minimum node size to 5, minimum sample leaf to 50; n_jobs to 1 as we use all processors that are parallelized over the trees. We have the highest FDR at 3% in the Out-of-Sample group of 0.526 (see Table 5.2.3).

Table 5.2.3 Hyper-parameters and Corresponding average FDR at 3% of Random Forest Models

Random Forested	# of Variables	# of Trees	mtry	min node size	TRAIN FDR at 3%	TEST FDR at 3%	OOT FDR at 3%
1	30	400	1	2	0.545	0.531	0.504
2	30	400	2	5	0.546	0.539	0.514
3	30	400	50	5	0.548	0.549	0.526
4	30	500	50	5	0.547	0.547	0.526

5.2.4 Gradient Boosting model using decision trees

Boosting is a flexible nonlinear regression procedure that helps improve the accuracy of trees. By sequentially applying weak classification algorithms to the incrementally changed data, a series of decision trees are created that produce an ensemble of weak prediction models.

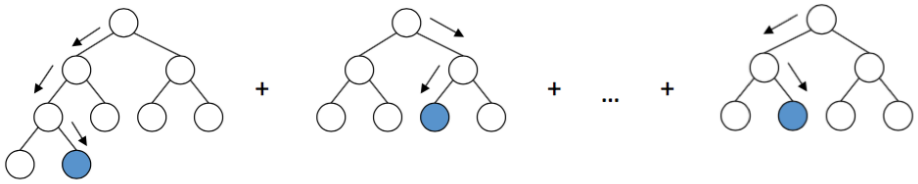


Figure 5.2.4. Illustration of Gradient Boosting model

Our team used GradientBoostingClassifier from sklearn library, and fit different choices on number of trees, maximum depth, and learning rate. Because the learning rate shrinks the contribution of each tree, and there is a trade-off between number of trees and the learning rate, we modified these two parameters to try to achieve a better model that generates the highest FDR for Out-of-Sample data. The following is the group of parameter we experiment:

n_estimators: The number of boosting stages to perform
max_depth: The maximum depth of the individual regression estimators.
min_samples_split: The minimum number of samples required to split an internal node
min_samples_leaf: The minimum number of samples required to be at a leaf node.

We finalized our model with n_estimator = 500; maximum depth of tree equals to 2; learning rate as default 0.1; minimum sample leaf 70; minimum sample split 7, with the average FDR at 3% of 0.503 for Out-of Sample data (see Table 5.2.4).

Table 5.2.4 Hyper-parameters and average FDR at 3% of Gradient Boosting Tree Models

Boosted Tree	# of Variables	# of Trees	Max Depth	Learning Rate	TRAIN FDR at 3%	TEST FDR at 3%	OOT FDR at 3%
1	20	100	2	0.1	0.529	0.522	0.500
2	20	100	2	0.01	0.507	0.498	0.479
3	20	200	2	0.1	0.529	0.523	0.500
4	20	500	2	0.1	0.531	0.525	0.503
5	20	1000	1	0.01	0.519	0.513	0.492
6	25	100	2	0.1	0.530	0.523	0.501
7	25	100	2	0.01	0.507	0.498	0.479
8	25	200	2	0.1	0.508	0.499	0.482
9	25	500	2	0.01	0.527	0.518	0.497
10	25	1000	1	0.01	0.519	0.513	0.492

6. Results

We build all 4 models and tested with 10-fold cross validation. Then we calculated the FDR at 3% scores for test data, training data and out-of-time data. Logistic model is the most simple model we built, which has the lowest FDR at 3% scores compared to other models. The highest FDR at 3% scores for OOT data is 0.49, which is still lower than 0.5310. Neural networks, random forest, and decision trees are more complex models which have a better result. The FDRs at 3% scores for OOT data are all higher than 0.5.

We choose neural network as our final model. The neural network model has the highest FDR at 3% scores for OOT data. Also, the neural network model has least standard deviation in test scores among all models. Therefore, the neural network model is more stable and stronger compared to other models. That is why we choose the neural network model as our best model for further discussion.

Table 6 Hyper-parameters and average FDR at 3% of Best Neural Network Models

Neural Net	Number of Variables	Layer	Node	Epoch	TRAIN FDR at 3%	TEST FDR at 3%	OOT FDR at 3%
2	30	1	90	50	0.546	0.550	0.531

The following three tables show the result of our neural network model for train data, test data, and out-of-time data. For each dataset, the table shows the number of good data and bad data from the first 1 percent of records to the first 20 percent of records. Also, the cumulative good and cumulative bad, and KS and FPR results.

Train	# Records	# Goods	# Bads	Fraud Rate								
	556497	548457	8040	0.014447								
Population Bin%	# Records	# Goods	# Bads	% Goods	% Bads	Total # Records	Cum. Goods	Cum. Bads	% Goods	% Bads (FDR)	KS	FPR
1	5565	1351	4214	24.28%	75.72%	5565	1351	4214	0.25%	52.41%	52.17	0.32
2	5565	5432	133	97.61%	2.39%	11130	6783	4347	1.24%	54.07%	52.83	1.56
3	5565	5525	40	99.28%	0.72%	16695	12308	4387	2.24%	54.56%	52.32	2.81
4	5565	5524	41	99.26%	0.74%	22260	17832	4428	3.25%	55.07%	51.82	4.03
5	5565	5524	41	99.26%	0.74%	27825	23356	4469	4.26%	55.58%	51.33	5.23
6	5565	5528	37	99.34%	0.66%	33390	28884	4506	5.27%	56.04%	50.78	6.41
7	5565	5530	35	99.37%	0.63%	38955	34414	4541	6.27%	56.48%	50.21	7.58
8	5565	5526	39	99.30%	0.70%	44520	39940	4580	7.28%	56.97%	49.68	8.72
9	5565	5532	33	99.41%	0.59%	50085	45472	4613	8.29%	57.38%	49.08	9.86
10	5565	5516	49	99.12%	0.88%	55650	50988	4662	9.30%	57.99%	48.69	10.94
11	5565	5527	38	99.32%	0.68%	61215	56515	4700	10.30%	58.46%	48.15	12.02
12	5565	5529	36	99.35%	0.65%	66780	62044	4736	11.31%	58.91%	47.59	13.10
13	5565	5525	40	99.28%	0.72%	72345	67569	4776	12.32%	59.40%	47.08	14.15
14	5565	5524	41	99.26%	0.74%	77910	73093	4817	13.33%	59.91%	46.59	15.17
15	5565	5531	34	99.39%	0.61%	83475	78624	4851	14.34%	60.34%	46.00	16.21
16	5565	5522	43	99.23%	0.77%	89040	84146	4894	15.34%	60.87%	45.53	17.19
17	5565	5525	40	99.28%	0.72%	94604	89670	4934	16.35%	61.37%	45.02	18.17
18	5565	5527	38	99.32%	0.68%	100169	95197	4972	17.36%	61.84%	44.48	19.15
19	5565	5528	37	99.34%	0.66%	105734	100725	5009	18.37%	62.30%	43.94	20.11
20	5565	5524	41	99.26%	0.74%	111299	106249	5050	19.37%	62.81%	43.44	21.04

Testing	# Records	# Goods	# Bads	Fraud Rate								
	238499	235053	3446	0.0144								
	Bin Statistics											
Population Bin%	# Records	# Goods	# Bads	% Goods	% Bads	Total # Records	Cumu. Goods	Cumu. Bads	% Goods	% Bads (FDR)	KS	FPR
1	2385	572	1813	23.98%	76.02%	2385	572	1813	0.24%	52.61%	52.37	0.32
2	2385	2327	58	97.57%	2.43%	4770	2899	1871	1.23%	54.29%	53.06	1.55
3	2385	2359	26	98.91%	1.09%	7155	5258	1897	2.24%	55.05%	52.81	2.77
4	2385	2370	15	99.37%	0.63%	9540	7628	1912	3.25%	55.48%	52.24	3.99
5	2385	2368	17	99.29%	0.71%	11925	9996	1929	4.25%	55.98%	51.73	5.18
6	2385	2366	19	99.20%	0.80%	14310	12362	1948	5.26%	56.53%	51.27	6.35
7	2385	2368	17	99.29%	0.71%	16695	14730	1965	6.27%	57.02%	50.76	7.50
8	2385	2369	16	99.33%	0.67%	19080	17099	1981	7.27%	57.49%	50.21	8.63
9	2385	2371	14	99.41%	0.59%	21465	19470	1995	8.28%	57.89%	49.61	9.76
10	2385	2373	12	99.50%	0.50%	23850	21843	2007	9.29%	58.24%	48.95	10.88
11	2385	2375	10	99.58%	0.42%	26235	24218	2017	10.30%	58.53%	48.23	12.01
12	2385	2370	15	99.37%	0.63%	28620	26588	2032	11.31%	58.97%	47.66	13.08
13	2385	2367	18	99.25%	0.75%	31005	28955	2050	12.32%	59.49%	47.17	14.12
14	2385	2362	23	99.04%	0.96%	33390	31317	2073	13.32%	60.16%	46.83	15.11
15	2385	2369	16	99.33%	0.67%	35775	33686	2089	14.33%	60.62%	46.29	16.13
16	2385	2366	19	99.20%	0.80%	38160	36052	2108	15.34%	61.17%	45.83	17.10
17	2385	2373	12	99.50%	0.50%	40545	38425	2120	16.35%	61.52%	45.17	18.12
18	2385	2363	22	99.08%	0.92%	42930	40788	2142	17.35%	62.16%	44.81	19.04
19	2385	2372	13	99.45%	0.55%	45315	43160	2155	18.36%	62.54%	44.17	20.03
20	2385	2362	23	99.04%	0.96%	47700	45522	2178	19.37%	63.20%	43.84	20.90

OOT	# Records	# Goods	# Bads	Fraud Rate								
	166493	164107	2386	0.014330								
Population Bin%	# Records	# Goods	# Bads	% Goods	% Bads	Total # Records	Cumu. Goods	Cumu. Bads	% Goods	% Bads (FDR)	KS	FPR
1	1665	476	1189	28.58%	71.41%	1665	476	1189	0.29%	49.83%	49.54	0.40
2	1665	1601	64	96.15%	3.85%	3330	2077	1253	1.27%	52.51%	51.25	1.66
3	1665	1651	14	99.15%	0.85%	4995	3728	1267	2.27%	53.10%	50.83	2.94
4	1665	1658	7	99.58%	0.42%	6660	5386	1274	3.28%	53.39%	50.11	4.23
5	1665	1652	13	99.22%	0.78%	8325	7038	1287	4.29%	53.94%	49.65	5.47
6	1665	1658	7	99.58%	0.42%	9990	8696	1294	5.30%	54.23%	48.93	6.72
7	1665	1658	7	99.58%	0.42%	11655	10354	1301	6.31%	54.53%	48.22	7.96
8	1665	1655	10	99.40%	0.60%	13319	12008	1311	7.32%	54.95%	47.63	9.16
9	1665	1653	12	99.28%	0.72%	14984	13661	1323	8.32%	55.45%	47.12	10.33
10	1665	1652	13	99.22%	0.78%	16649	15313	1336	9.33%	55.99%	46.66	11.46
11	1665	1652	13	99.22%	0.78%	18314	16965	1349	10.34%	56.54%	46.20	12.58
12	1665	1655	10	99.40%	0.60%	19979	18620	1359	11.35%	56.96%	45.61	13.70
13	1665	1651	14	99.15%	0.85%	21644	20271	1373	12.35%	57.54%	45.19	14.76
14	1665	1659	6	99.64%	0.36%	23309	21930	1379	13.36%	57.80%	44.43	15.90
15	1665	1656	9	99.46%	0.54%	24974	23586	1388	14.37%	58.17%	43.80	16.99
16	1665	1658	7	99.58%	0.42%	26639	25244	1395	15.38%	58.47%	43.08	18.10
17	1665	1649	16	99.03%	0.97%	28304	26893	1411	16.39%	59.14%	42.75	19.06
18	1665	1655	10	99.40%	0.60%	29969	28548	1421	17.40%	59.56%	42.16	20.09
19	1665	1658	7	99.58%	0.42%	31634	30206	1428	18.41%	59.85%	41.44	21.15
20	1665	1653	12	99.28%	0.72%	33299	31859	1440	19.41%	60.35%	40.94	22.12

7. Conclusions

In this report, we use machine learning models to analyze credit card application information and evaluate the data. We first explored the basic statistics of the dataset. There are originally 9 fields excluding the record field, and with a total of 1,000,000 records. Then we cleaned the data and handled exclusions, outliers, missing fields, and frivolous field values. After cleaning the data, we created candidate variables including velocity, days since, relative velocity, and uniqueness. Then we used Z-scale all the features and used KS and FDR methods to do the feature selection. We selected our final 30 variables to build machine learning models.

We used logistic algorithms, neural networks, random forest, and decision trees. The Logistic algorithm model is the simplest model we first built, which results in a 49.4% FDR at 3% for out-of-time data. Then we built more complex models such as neural networks, random forest, and decision trees. After looking at the results, the neural network model has the best performance of 53.10% FDR at 3% for out-of-time data. Therefore, we choose the neural network model as our final model and create summary tables for train data, test data, and out-of-time data.

There may still be some shortcomings in our models. If we have more time, we will try to improve our models. We will try to create more effective candidate variables. Also, we will adjust the parameters of our models and get a higher FDR at 3%. Moreover, we will use more types of machine learning models to handle the data.

Appendix A Data Quality Report

DATA QUALITY REPORT FOR PRODUCT APPLICATION DATA

Description

Dataset Name: Product Application and Fraud Data
Dataset Purpose: Application data of credit cards and cell phones Applicants, to find application/identity fraud
Data Source: From identity fraud prevention company, synthetic data set built from studying the statistical properties of more than a billion applications
Time Period: 2016
Number of Fields: #9 (exclude record field)
Number of Records: 1,000,000

Summary Table

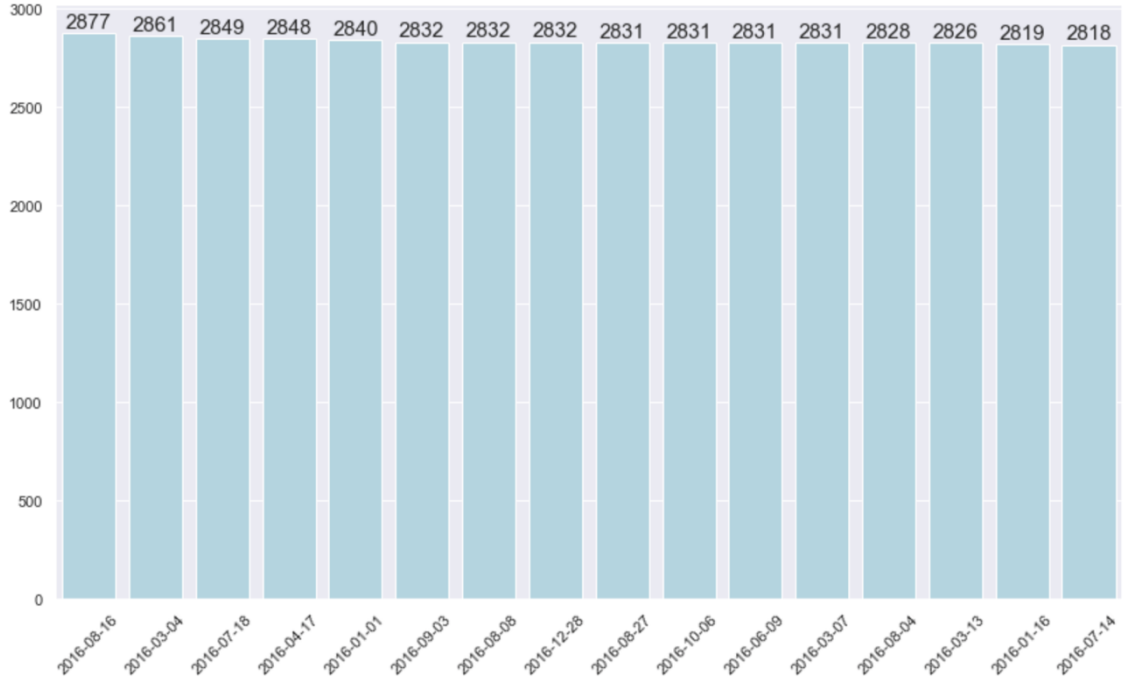
Categorical Fields:

Comuln Name	# of Records	% Populated	Unique Values	Most Common Field Value
record	1000000	100.0	1000000	2047
date	1000000	100.0	365	20160816
ssn	1000000	100.0	835819	999999999
firstname	1000000	100.0	78136	EAMSTRMT
lastname	1000000	100.0	177001	ERJSAXA
address	1000000	100.0	828774	123 MAIN ST
zip5	1000000	100.0	26370	68138
dob	1000000	100.0	42673	19070626
homephone	1000000	100.0	28244	9999999999
fraud_label	1000000	100.0	2	0

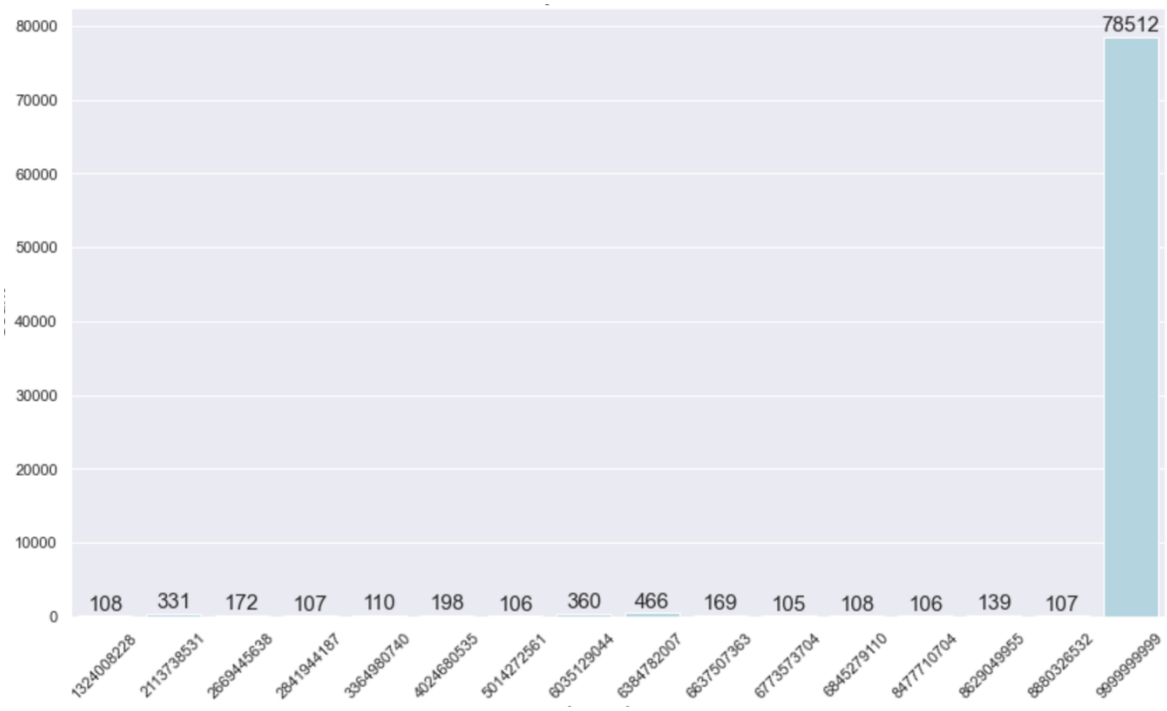
Data Field Exploration:

Field 1:
Name: Record
Description: Unique identifier of each entry in the data.

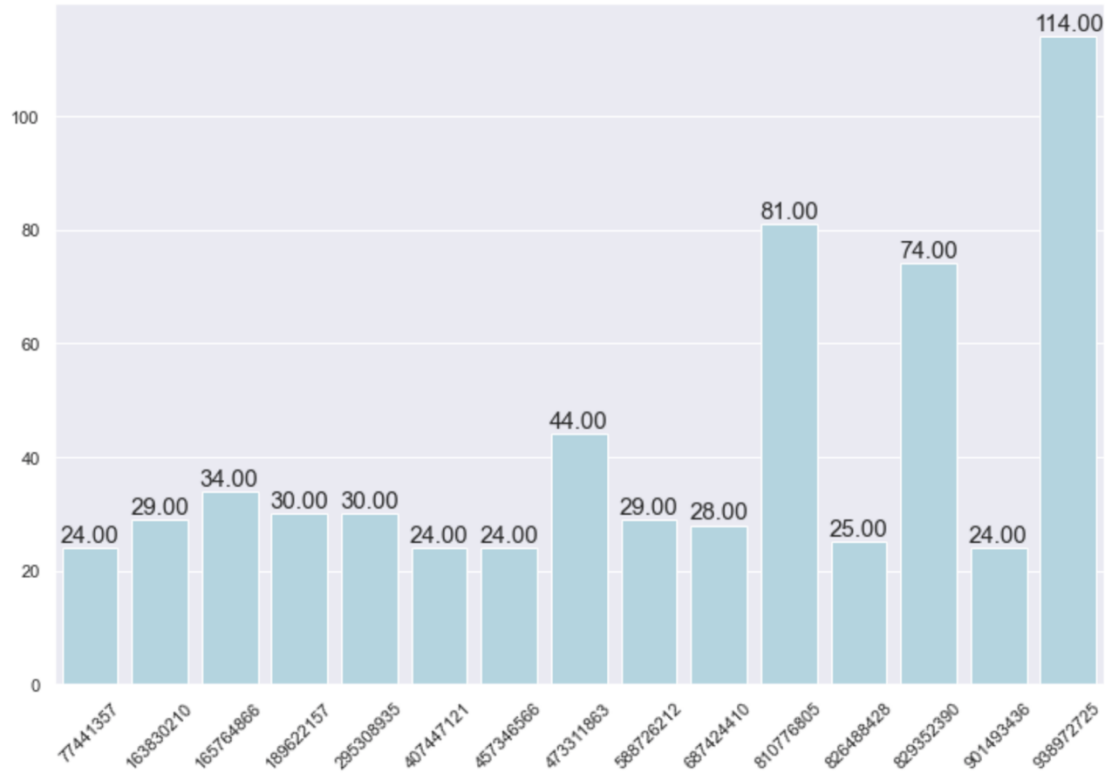
Field 2:
Name: DATE
Description: The application date.



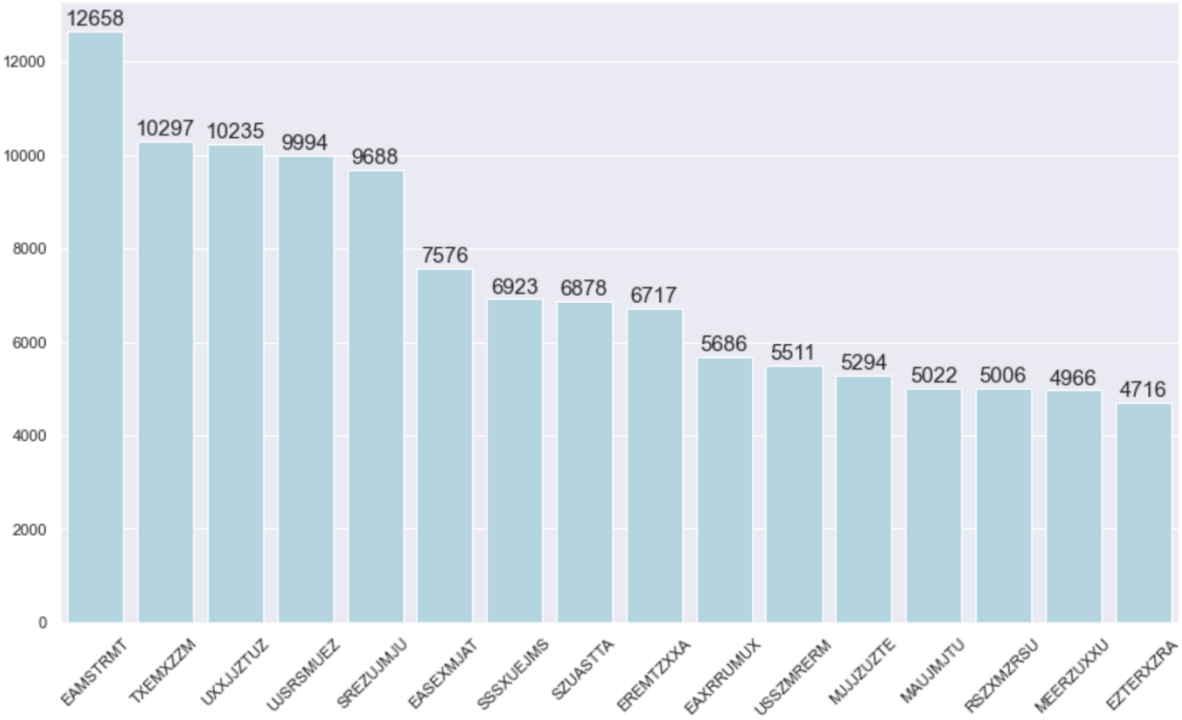
Field 3:
Name: SSN
Description: The Social Security Number of the applicant.
Graph with 999999999



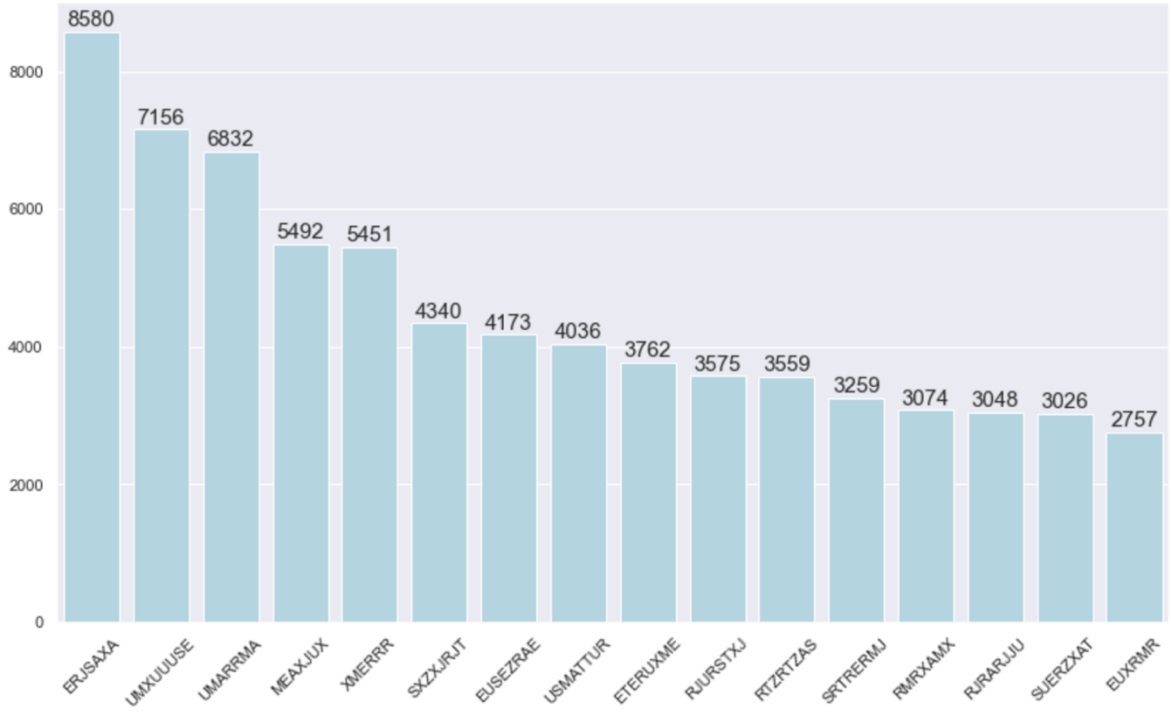
Graph without 999999999:



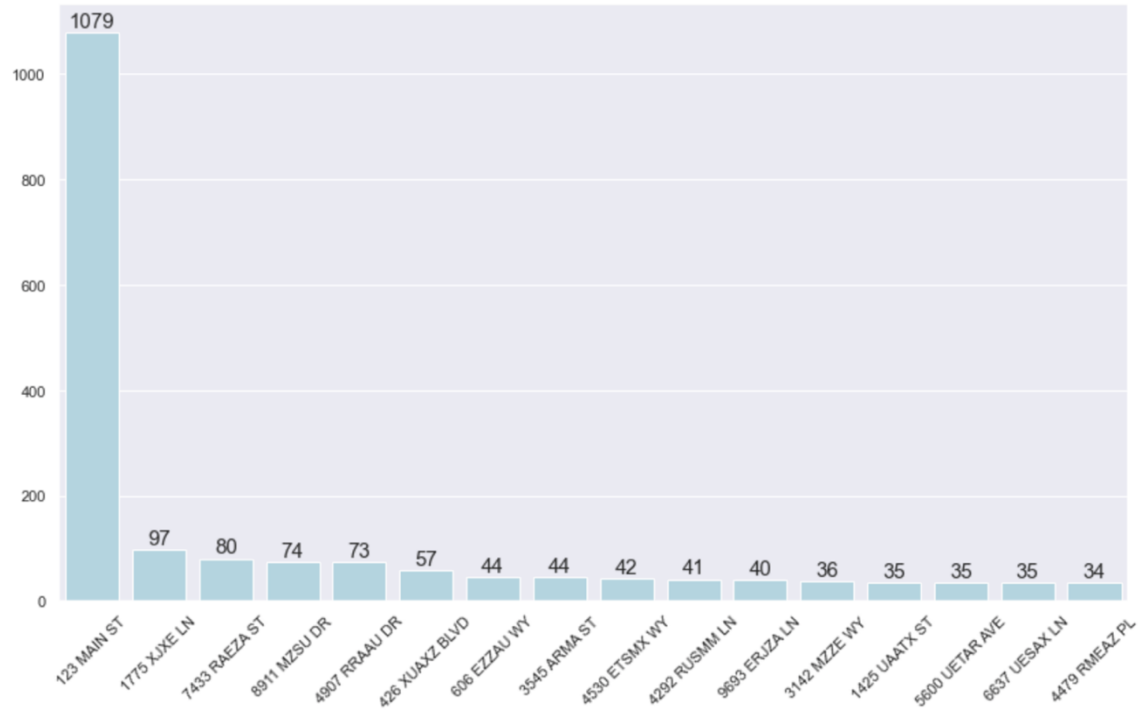
Field 4:
Name: FIRSTNAME
Description: The first name of the applicant.



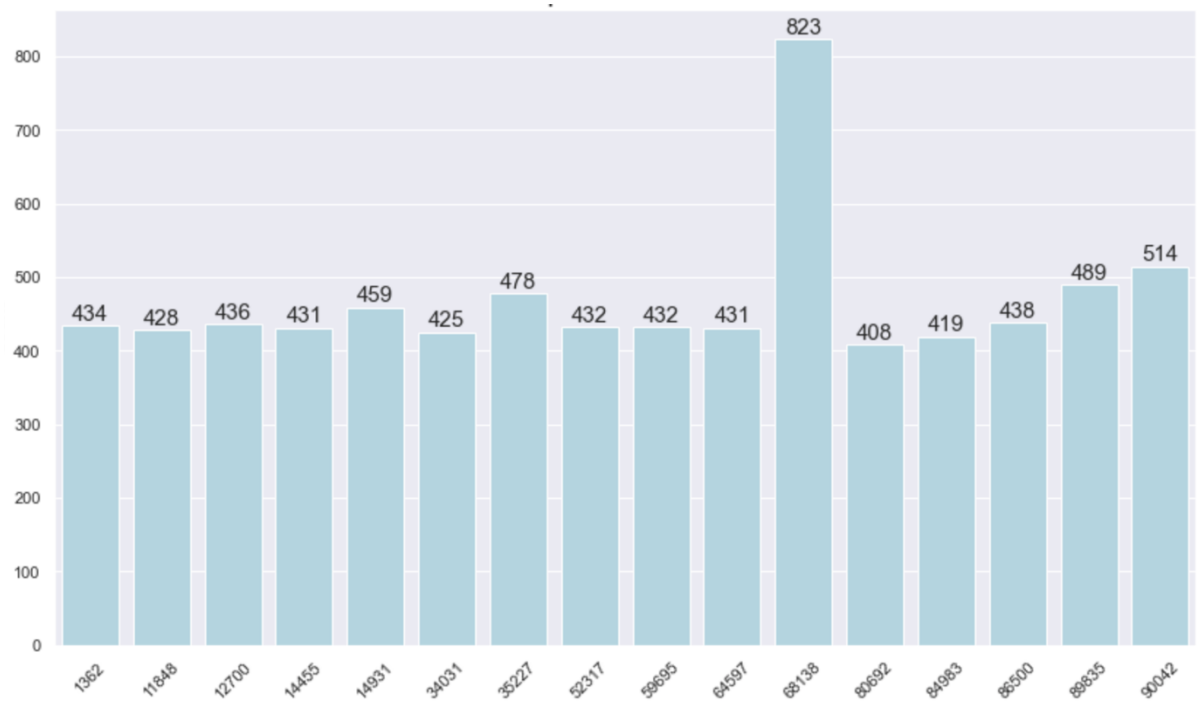
Field 5:
Name: LASTNAME
Description: The last name of the applicant.



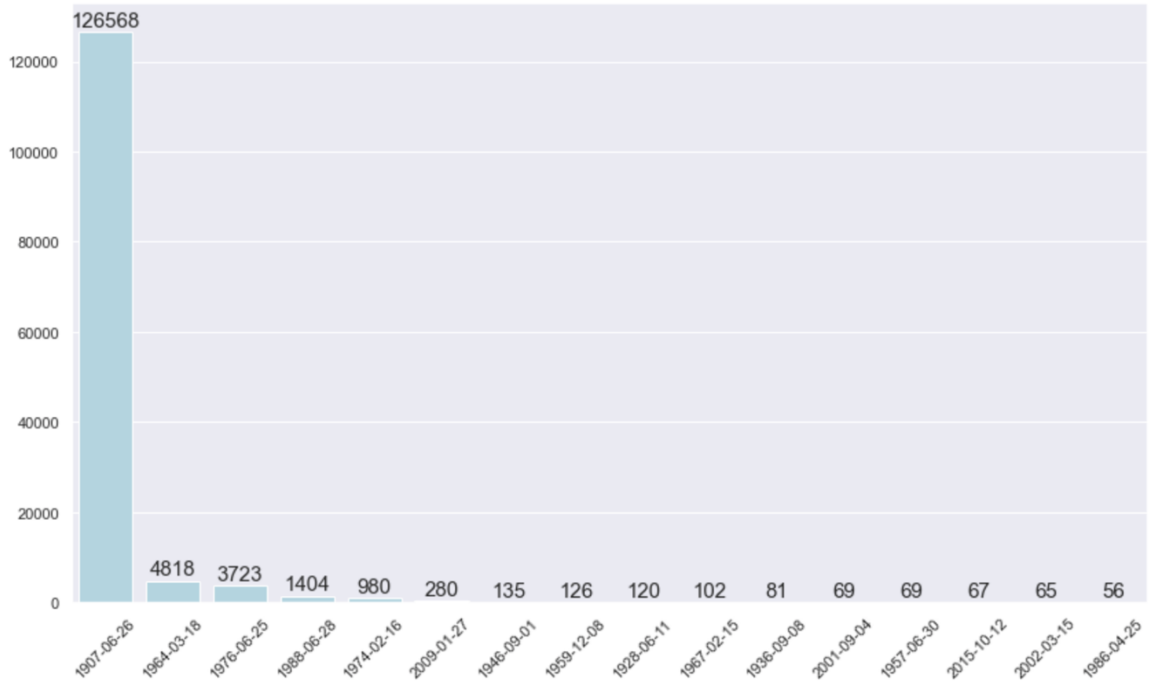
Field 6:
Name: ADDRESS
Description: The address of the applicant.



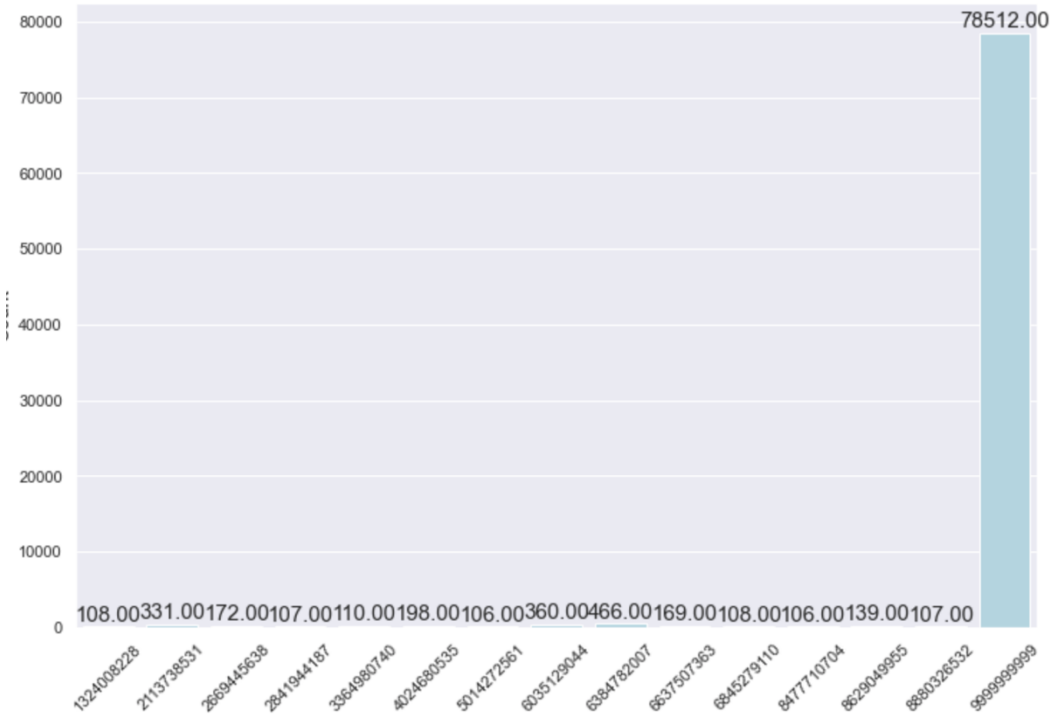
Field 7:
Name: ZIP5
Description: The zip code of the applicant.



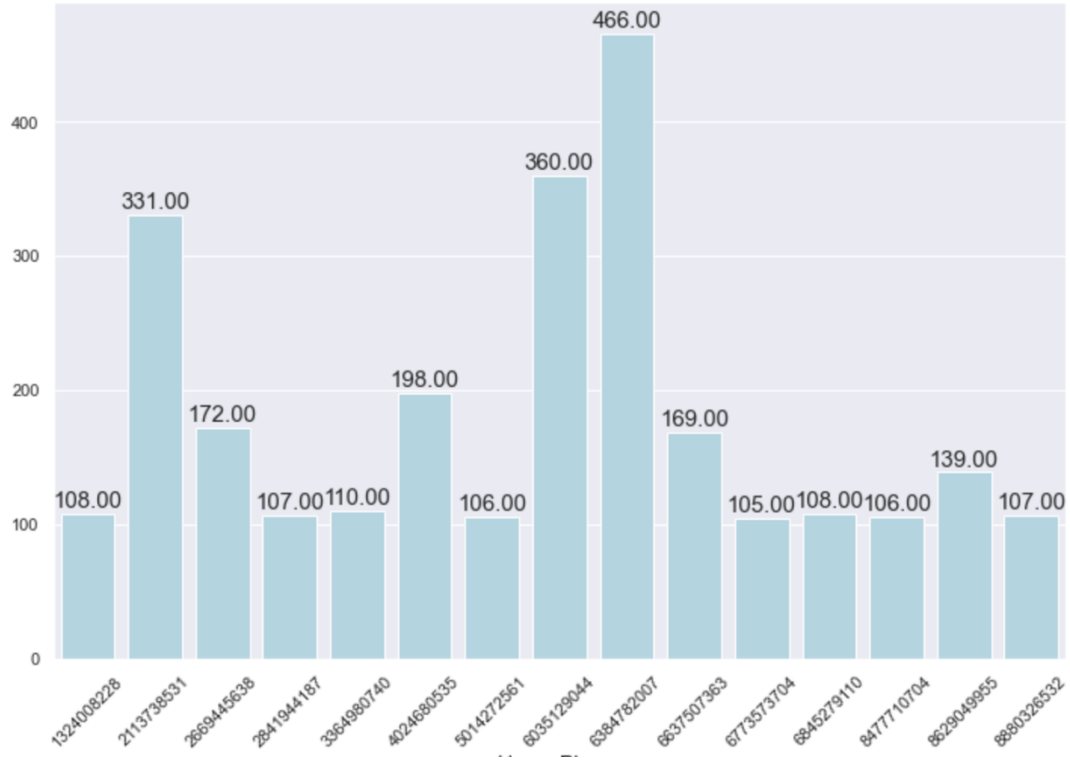
Field 8:
Name: DOB
Description: The date of birth of the applicant.



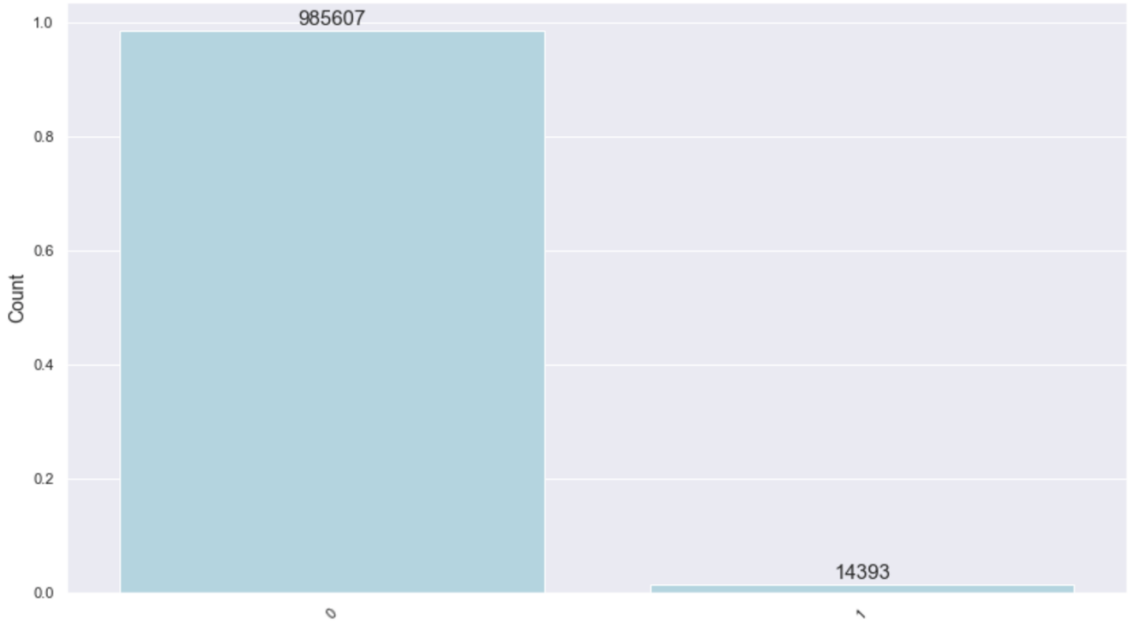
Field 9:
Name: HOMEPHONE
Description: The home phone number of the applicant.
Graph with 999999999:



Graph without 99999999:



Field 10:
Name: FRAUD_LABEL
Description: The fraud status of the applicant.
1: The applicant conducted fraud.
0: The applicant did not conduct fraud.



Appendix B Full list of All the Variables

Original Variables in the Dataset (11 Total)

record	date	ssn	firstname	lastname	address
zip5	dob	homephone	fraud_label		

Identity Variables (3 Total)

age	name	fulladdress

Identity Combination Variables (20 Total)

name_ssn	name_dob	name_fulladdress	name_homephone	ssn_dob	ssn_fulladdress
ssn_homephone	fulladdress_dob	fulladdress_homephone	dob_homephone	name_dob_homephone	name_dob_ssn
name_dob_fulladdress	name_homephone_ssn	name_homephone_fulladdress	name_ssn_fulladdress	ssn_dob_homephone	ssn_dob_fulladdress
ssn_homephone_fulladdress	dob_homephone_fulladdress				

dow (2 Total)

dow	dow_risk

Day Since (25 Total)

ssn_day_since	dob_day_since	homephone_day_since	name_day_since	fulladdress_day_since	name_ssn_day_since
name_dob_day_since	name_fulladdress_day_since	name_homephone_day_since	ssn_dob_day_since	ssn_fulladdress_day_since	ssn_homephone_day_since
fulladdress_dob_day_since	fulladdress_homephone_day_since	dob_homephone_day_since	name_dob_homephone_day_since	name_dob_ssn_day_since	name_dob_fulladdress_day_since
name_homepho	name_homepho	name_ssn_fullad	ssn_dob_homep	ssn_dob_fullad	ssn_homephone

ne_ssn_day_since	ne_fulladdress_day_since	dress_day_since	hone_day_since	dress_day_since	_fulladdress_day_since
dob_homephone_fulladdress_day_since					

Velocity (150 Total)

ssn_count_0	ssn_count_1	ssn_count_3	ssn_count_7	ssn_count_14	ssn_count_30
dob_count_0	dob_count_1	dob_count_3	dob_count_7	dob_count_14	dob_count_30
homephone_count_0	homephone_count_1	homephone_count_3	homephone_count_7	homephone_count_14	homephone_count_30
name_count_0	name_count_1	name_count_3	name_count_7	name_count_14	name_count_30
fulladdress_count_0	fulladdress_count_1	fulladdress_count_3	fulladdress_count_7	fulladdress_count_14	fulladdress_count_30
name_ssn_count_0	name_ssn_count_1	name_ssn_count_3	name_ssn_count_7	name_ssn_count_14	name_ssn_count_30
name_dob_count_0	name_dob_count_1	name_dob_count_3	name_dob_count_7	name_dob_count_14	name_dob_count_30
name_fulladdress_count_0	name_fulladdress_count_1	name_fulladdress_count_3	name_fulladdress_count_7	name_fulladdress_count_14	name_fulladdress_count_30
name_homephone_count_0	name_homephone_count_1	name_homephone_count_3	name_homephone_count_7	name_homephone_count_14	name_homephone_count_30
ssn_dob_count_0	ssn_dob_count_1	ssn_dob_count_3	ssn_dob_count_7	ssn_dob_count_14	ssn_dob_count_30
ssn_fulladdress_count_0	ssn_fulladdress_count_1	ssn_fulladdress_count_3	ssn_fulladdress_count_7	ssn_fulladdress_count_14	ssn_fulladdress_count_30
ssn_homephone_count_0	ssn_homephone_count_1	ssn_homephone_count_3	ssn_homephone_count_7	ssn_homephone_count_14	ssn_homephone_count_30
fulladdress_dob_count_0	fulladdress_dob_count_1	fulladdress_dob_count_3	fulladdress_dob_count_7	fulladdress_dob_count_14	fulladdress_dob_count_30
fulladdress_homephone_count_0	fulladdress_homephone_count_1	fulladdress_homephone_count_3	fulladdress_homephone_count_7	fulladdress_homephone_count_14	fulladdress_homephone_count_30
dob_homephone_count_0	dob_homephone_count_1	dob_homephone_count_3	dob_homephone_count_7	dob_homephone_count_14	dob_homephone_count_30
name_dob_hom	name_dob_hom	name_dob_hom	name_dob_hom	name_dob_hom	name_dob_hom

ephone_count_0	ephone_count_1	ephone_count_3	ephone_count_7	ephone_count_14	ephone_count_30
name_dob_ssn_count_0	name_dob_ssn_count_1	name_dob_ssn_count_3	name_dob_ssn_count_7	name_dob_ssn_count_14	name_dob_ssn_count_30
name_dob_fulladdress_count_0	name_dob_fulladdress_count_1	name_dob_fulladdress_count_3	name_dob_fulladdress_count_7	name_dob_fulladdress_count_14	name_dob_fulladdress_count_30
name_homephone_ssn_count_0	name_homephone_ssn_count_1	name_homephone_ssn_count_3	name_homephone_ssn_count_7	name_homephone_ssn_count_14	name_homephone_ssn_count_30
name_homephone_fulladdress_count_0	name_homephone_fulladdress_count_1	name_homephone_fulladdress_count_3	name_homephone_fulladdress_count_7	name_homephone_fulladdress_count_14	name_homephone_fulladdress_count_30
name_ssn_fulladdress_count_0	name_ssn_fulladdress_count_1	name_ssn_fulladdress_count_3	name_ssn_fulladdress_count_7	name_ssn_fulladdress_count_14	name_ssn_fulladdress_count_30
ssn_dob_homephone_count_0	ssn_dob_homephone_count_1	ssn_dob_homephone_count_3	ssn_dob_homephone_count_7	ssn_dob_homephone_count_14	ssn_dob_homephone_count_30
ssn_dob_fulladdress_count_0	ssn_dob_fulladdress_count_1	ssn_dob_fulladdress_count_3	ssn_dob_fulladdress_count_7	ssn_dob_fulladdress_count_14	ssn_dob_fulladdress_count_30
ssn_homephone_fulladdress_count_0	ssn_homephone_fulladdress_count_1	ssn_homephone_fulladdress_count_3	ssn_homephone_fulladdress_count_7	ssn_homephone_fulladdress_count_14	ssn_homephone_fulladdress_count_30
dob_homephone_fulladdress_count_0	dob_homephone_fulladdress_count_1	dob_homephone_fulladdress_count_3	dob_homephone_fulladdress_count_7	dob_homephone_fulladdress_count_14	dob_homephone_fulladdress_count_30

Relative Velocity (200 Total)

ssn_count_0_by_3	ssn_count_0_by_7	ssn_count_0_by_14	ssn_count_0_by_30	ssn_count_1_by_3	ssn_count_1_by_7
ssn_count_1_by_14	ssn_count_1_by_30	dob_count_0_by_3	dob_count_0_by_7	dob_count_0_by_14	dob_count_0_by_30
dob_count_1_by_3	dob_count_1_by_7	dob_count_1_by_14	dob_count_1_by_30	homephone_count_0_by_3	homephone_count_0_by_7
homephone_count_0_by_14	homephone_count_0_by_30	homephone_count_1_by_3	homephone_count_1_by_7	homephone_count_1_by_14	homephone_count_1_by_30
name_count_0_by_3	name_count_0_by_7	name_count_0_by_14	name_count_0_by_30	name_count_1_by_3	name_count_1_by_7

name_count_1_by_14	name_count_1_by_30	fulladdress_count_0_by_3	fulladdress_count_0_by_7	fulladdress_count_0_by_14	fulladdress_count_0_by_30
fulladdress_count_1_by_3	fulladdress_count_1_by_7	fulladdress_count_1_by_14	fulladdress_count_1_by_30	name_ssn_count_0_by_3	name_ssn_count_0_by_7
name_ssn_count_0_by_14	name_ssn_count_0_by_30	name_ssn_count_1_by_3	name_ssn_count_1_by_7	name_ssn_count_1_by_14	name_ssn_count_1_by_30
name_dob_count_0_by_3	name_dob_count_0_by_7	name_dob_count_0_by_14	name_dob_count_0_by_30	name_dob_count_1_by_3	name_dob_count_1_by_7
name_dob_count_1_by_14	name_dob_count_1_by_30	name_fulladdress_count_0_by_3	name_fulladdress_count_0_by_7	name_fulladdress_count_0_by_14	name_fulladdress_count_0_by_30
name_fulladdress_count_1_by_3	name_fulladdress_count_1_by_7	name_fulladdress_count_1_by_14	name_fulladdress_count_1_by_30	name_homephone_count_0_by_3	name_homephone_count_0_by_7
name_homephone_count_0_by_14	name_homephone_count_0_by_30	name_homephone_count_1_by_3	name_homephone_count_1_by_7	name_homephone_count_1_by_14	name_homephone_count_1_by_30
ssn_dob_count_0_by_3	ssn_dob_count_0_by_7	ssn_dob_count_0_by_14	ssn_dob_count_0_by_30	ssn_dob_count_1_by_3	ssn_dob_count_1_by_7
ssn_dob_count_1_by_14	ssn_dob_count_1_by_30	ssn_fulladdress_count_0_by_3	ssn_fulladdress_count_0_by_7	ssn_fulladdress_count_0_by_14	ssn_fulladdress_count_0_by_30
ssn_fulladdress_count_1_by_3	ssn_fulladdress_count_1_by_7	ssn_fulladdress_count_1_by_14	ssn_fulladdress_count_1_by_30	ssn_homephone_count_0_by_3	ssn_homephone_count_0_by_7
ssn_homephone_count_0_by_14	ssn_homephone_count_0_by_30	ssn_homephone_count_1_by_3	ssn_homephone_count_1_by_7	ssn_homephone_count_1_by_14	ssn_homephone_count_1_by_30
fulladdress_dob_count_0_by_3	fulladdress_dob_count_0_by_7	fulladdress_dob_count_0_by_14	fulladdress_dob_count_0_by_30	fulladdress_dob_count_1_by_3	fulladdress_dob_count_1_by_7
fulladdress_dob_count_1_by_14	fulladdress_dob_count_1_by_30	fulladdress_homephone_count_0_by_3	fulladdress_homephone_count_0_by_7	fulladdress_homephone_count_0_by_14	fulladdress_homephone_count_0_by_30
fulladdress_homephone_count_1_by_3	fulladdress_homephone_count_1_by_7	fulladdress_homephone_count_1_by_14	fulladdress_homephone_count_1_by_30	dob_homephone_count_0_by_3	dob_homephone_count_0_by_7
dob_homephone_count_0_by_14	dob_homephone_count_0_by_30	dob_homephone_count_1_by_3	dob_homephone_count_1_by_7	dob_homephone_count_1_by_14	dob_homephone_count_1_by_30
name_dob_homephone_count_0_by_3	name_dob_homephone_count_0_by_7	name_dob_homephone_count_0_by_14	name_dob_homephone_count_0_by_30	name_dob_homephone_count_1_by_3	name_dob_homephone_count_1_by_7
name_dob_hom	name_dob_hom	name_dob_ssn_	name_dob_ssn_	name_dob_ssn_	name_dob_ssn_

ephone_count_1_by_14	ephone_count_1_by_30	count_0_by_3	count_0_by_7	count_0_by_14	count_0_by_30
name_dob_ssn_count_1_by_3	name_dob_ssn_count_1_by_7	name_dob_ssn_count_1_by_14	name_dob_ssn_count_1_by_30	name_dob_fulladdress_count_0_by_3	name_dob_fulladdress_count_0_by_7
name_dob_fulladdress_count_0_by_14	name_dob_fulladdress_count_0_by_30	name_dob_fulladdress_count_1_by_3	name_dob_fulladdress_count_1_by_7	name_dob_fulladdress_count_1_by_14	name_dob_fulladdress_count_1_by_30
name_homephone_ssn_count_0_by_3	name_homephone_ssn_count_0_by_7	name_homephone_ssn_count_0_by_14	name_homephone_ssn_count_0_by_30	name_homephone_ssn_count_1_by_3	name_homephone_ssn_count_1_by_7
name_homephone_ssn_count_1_by_14	name_homephone_ssn_count_1_by_30	name_homephone_fulladdress_count_0_by_3	name_homephone_fulladdress_count_0_by_7	name_homephone_fulladdress_count_0_by_14	name_homephone_fulladdress_count_0_by_30
name_homephone_fulladdress_count_1_by_3	name_homephone_fulladdress_count_1_by_7	name_homephone_fulladdress_count_1_by_14	name_homephone_fulladdress_count_1_by_30	name_ssn_fulladdress_count_0_by_3	name_ssn_fulladdress_count_0_by_7
name_ssn_fulladdress_count_0_by_14	name_ssn_fulladdress_count_0_by_30	name_ssn_fulladdress_count_1_by_3	name_ssn_fulladdress_count_1_by_7	name_ssn_fulladdress_count_1_by_14	name_ssn_fulladdress_count_1_by_30
ssn_dob_homephone_count_0_by_3	ssn_dob_homephone_count_0_by_7	ssn_dob_homephone_count_0_by_14	ssn_dob_homephone_count_0_by_30	ssn_dob_homephone_count_1_by_3	ssn_dob_homephone_count_1_by_7
ssn_dob_homephone_count_1_by_14	ssn_dob_homephone_count_1_by_30	ssn_dob_fulladdress_count_0_by_3	ssn_dob_fulladdress_count_0_by_7	ssn_dob_fulladdress_count_0_by_14	ssn_dob_fulladdress_count_0_by_30
ssn_dob_fulladdress_count_1_by_3	ssn_dob_fulladdress_count_1_by_7	ssn_dob_fulladdress_count_1_by_14	ssn_dob_fulladdress_count_1_by_30	ssn_homephone_fulladdress_count_0_by_3	ssn_homephone_fulladdress_count_0_by_7
ssn_homephone_fulladdress_count_0_by_14	ssn_homephone_fulladdress_count_0_by_30	ssn_homephone_fulladdress_count_1_by_3	ssn_homephone_fulladdress_count_1_by_7	ssn_homephone_fulladdress_count_1_by_14	ssn_homephone_fulladdress_count_1_by_30
dob_homephone_fulladdress_count_0_by_3	dob_homephone_fulladdress_count_0_by_7	dob_homephone_fulladdress_count_0_by_14	dob_homephone_fulladdress_count_0_by_30	dob_homephone_fulladdress_count_1_by_3	dob_homephone_fulladdress_count_1_by_7
dob_homephone_fulladdress_count_1_by_14	dob_homephone_fulladdress_count_1_by_30				

Uniqueness (17 Total)

name_ssn_name_ssn_fulladdress_unique_0	ssn_fulladdress_name_ssn_fulladdress_unique_0	ssn_homephone_ssn_homephone_fulladdress_unique_0	name_homephone_name_homephone_ssn_unique_0	name_homephone_name_homephone_fulladdress_unique_0	ssn_name_homephone_ssn_unique_0
ssn_name_ssn_fulladdress_unique_0	ssn_ssn_homephone_fulladdress_unique_0	name_name_homephone_ssn_unique_0	name_name_homephone_fulladdress_unique_0	name_name_ssn_fulladdress_unique_0	fulladdress_name_homephone_fulladdress_unique_0
fulladdress_name_ssn_fulladdress_unique_0	fulladdress_ssn_homephone_fulladdress_unique_0	homephone_name_homephone_ssn_unique_0	homephone_name_homephone_fulladdress_unique_0	homephone_ssn_homephone_fulladdress_unique_0	