3/25/2021

University of Southern California

Marshall school of business

Team Member:

Xue Gao,

Tianze Yang,

Yang Jiao,

Yunlong Wang,

Huiyi Zhang,

Xinzhu Zhang

DSO 562 Project 2

Identity Fraud in the Credit Card Dataset

Table of Contents

[Executive Summary](#_i0iuzmjh3vy) 3

[Description of Data](#_utjmrovrj8ub) 4

[Data Cleaning](#_xaw7tticpx0x) 6

[Candidate Variables](#_2ipyrpc2q5cq) 7

[Feature Selection Process](#_5ihaqqpd6fu5) 8

[Model Algorithms](#_ac8j6jza5lhz) 12

[Results](#_arwr1t3192l0) 18

[Conclusions](#_dfzosi4kgmq5) 21

[Appendix](#_389emsl7cnvy) 22

# Executive Summary

As technology advances rapidly evolve around methods of payments, also presented are potential exposures to various risks. As the most common type of identity theft, Credit card fraud is widely prevalent in the U.S. According to research, of the 1.5 billion credit cards issued in the United States, millions fall victim to this unscrupulous tactic each year.

In addition to performing unauthorized transactions using stolen or lost credit cards, it has been discovered that fraudsters often apply for a credit card in someone else’s name. Basic information such as legal name, date of birth, address, and social security numbers are rudimentary to this scheme; overtime, the criminals also adopted less conventional methods to extract relevant info from supporting documents to fly under the radar.

In this project, we aim to build a real-time fraud detection model to predict if a credit card applicant uses someone else’s in combination with made-up information to commit fraud.

The credit card application dataset we use contains one million rows of records each with ten columns, including date, ssn, first name, last name, address, zip5, dob, homephone, and fraud label.

In order to develop this predictive model, we went through several steps. At first, we did data cleaning and sorting to adjust the data type and replace frivolous values. Then, we did feature engineering by creating combination group variables and day since, velocity, and relative velocity variables for each combination group.

After generating candidate variables, we operated feature selection to select a number of best potential variables to train the models. By having a dataset split in training, testing, and out of time validation, we trained models by applying different machine learning algorithms and compared the performance by calculating average fraud detection rate at 3% to select the best model to be our final model.

As a result, we found that the Gradient Boosting Tree with Learning\_rate 0.01, n\_estimator 1100, and max\_depth of 5 has the highest average FDR at 3% in the test set. After that, we used that model to calculate the bins and cumulative goods, bads and built the tables in the results section.

# Description of Data

# Applications Data is a dataset containing records of 1,000,000 applications. It includes fields such as date of application, SSN, first and last name, address, zip code, date of birth, home phone number, and fraud label of each applicant.

File Name: applications data.csv

Data Source: An identity fraud prevention company

Time Period: Jan 1st 2016 – Dec 31st 2016

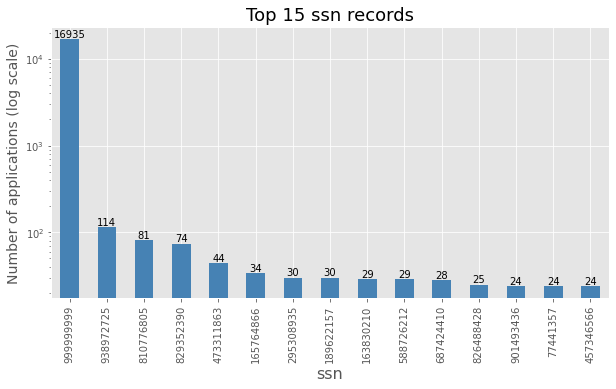
Number of Records: 1,000,000 records

Number of Fields: 9 variables in total: 7 categorical variables, 2 date variables

|  |  |  |  |
| --- | --- | --- | --- |
| Field Name | Data Type | %Populated | Unique number |
| date | Date variable | 100% | 365 |
| ssn | Categorical variable | 100% | 835819 |
| firstname | Categorical variable | 100% | 78136 |
| lastname | Categorical variable | 100% | 177001 |
| address | Categorical variable | 100% | 828774 |
| zip5 | Categorical variable | 100% | 26370 |
| dob | Date variable | 100% | 42673 |
| homephone | Categorical variable | 100% | 28244 |
| Fraud\_label | Categorical variable | 100% | 2 |

*date*: The date when the credit card application was filled. It ranges from Jan 1st 2016 – Dec 31st 2016.

*ssn*: The SSN used for that particular credit card application.



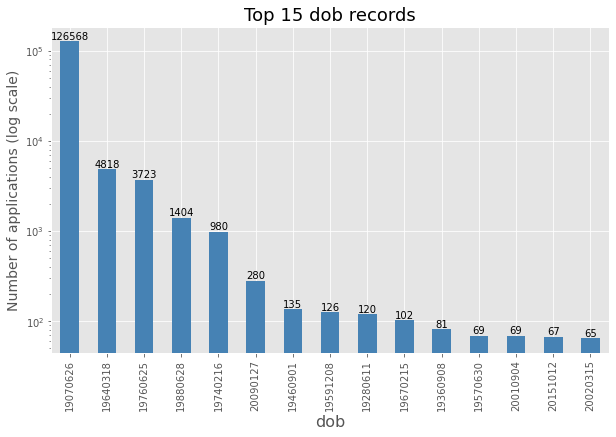
*firstname*: The first name used for that credit card application.

*lastname*: The last name used for that credit card application.

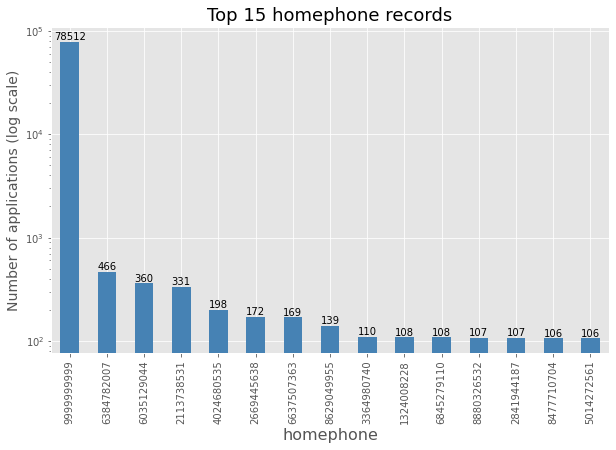
*address*: The address used for that credit card application, which includes street number and street name.

*zip5*: The 5-digit zip code used for that credit card application.

*dob*: The date of birth that the applicants used for application, the format is YYYY-MM-DD.



*homephone*: The phone number that the applicants used for credit card application.



*fraud\_lable*: Whether the record is considered fraud.

# Data Cleaning

The original dataset has a total of 9 fields. Each of the fields is 100% populated. There are no missing values in any of the fields.

The date field was originally in the type of int64. We firstly changed the data type to a string, added dashes in between to separate the year, month and day, and finally converted it to a datetime variable using the pandas to\_datetime function.

The zip5 field has some values in 4 digits and others in 5 digits. To unify the number of digits in each entry, we formatted the 4-digit ones by adding a 0 in front of each entry.

The most common value in the ssn field was 9999999999, which was clearly a frivolous value. To fix this problem, we replaced the frivolous value with a field that would not link -- the negative of the record number and then formatted the entries into 9 digits.

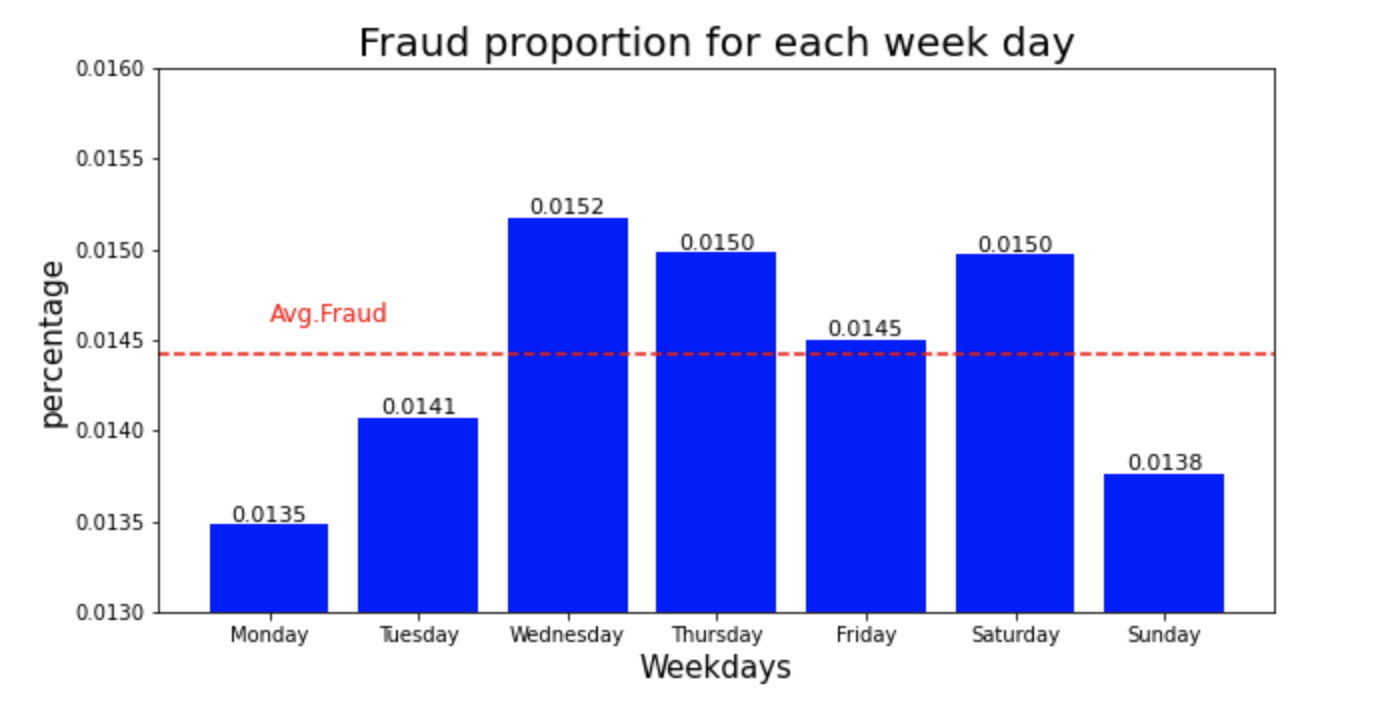
Similarly, the most common value in the homephone field was 9999999999. We replaced the frivolous value with the negative of the record number and formatted it into a 10-digit number.

The address field had around a thousand frivolous values of “123 MAIN ST”. We created a value by concatenating the record number with the word “RECORD” and used it to replace each frivolous value under the address field.

The most common field value in the dob field was 19070626, which was an impossible number. Again, we replaced it with the negative of the record number and formatted it into an 8-digit number.

# Candidate Variables

First of all, we created a variable called “day of the week” to figure out the risk of each day in a week. From the risk table for day of week, it was obvious that Monday had the lowest fraud risk and Wednesday had the highest risk.



Then we linked the original fields and created new entities:

name = firstname + lastname

fulladdress = address + zip5

name\_dob = name + dob

name\_fulladdress = name + fulladdress

name\_homephone = name + homephone

fulladdress\_dob=fulladdress+dob

fulladdress\_homephone=fulladdress+homephone

dob\_homephone=dob+homephone

homephone\_name\_dob=homephone+name\_dob

Then we linked applicants’ SSN to all these entities described above. In total, we created 28 entities.

Next, we built ‘velocity’ variables for all these attributes, which means the number of records with the same attributes over the last 0, 1, 3, 7, 14, 30 days.

Another group ‘day since’ variables mean that how many days has passed since the last time the attributes appeared.

In the end, we built ‘relative velocity’ group variables, and they represented the proportion of the number of times we have seen that entity in the past days comes from the recent past.

|  |  |
| --- | --- |
| Variable groups | Variable name format |
| Velocity | attributes \_count\_xx |
| Day since | attibutes\_since |
| Relative velocity | attributes\_count\_yy\_by\_xx |

*xx: 0, 1, 3, 7, 14 days； yy: recent days*

# Feature Selection Process

During the feature selection stage, we calculated the Kolmogorov–Smirnov (KS) and Fraud Detection Rate (FDR) value for each variable and used backward stepwise methods to select 30 variables for our final models.

* Kolmogorov–Smirnov (KS)

Kolmogorov–Smirnov is the measurement of how well two distributions are separated. The larger the KS, the more separate the two distributions. We used KS to measure the differences between fraud records and non-fraud records for each variable created. Specifically, for each variable, we gathered a list of fields corresponding to fraud records and the other list of random numbers between 0 and 1. Then we applied stats.ks.2samp function to compute KS for all variables.

* Fraud Detection Rate (FDR)

Fraud Detection Rate is the percentage of all the fraud found at a score cutoff. 3% of FDR means the calculation of how many fraud records in the top 3% of all records. Specifically, we sort the data and compute the number of bad records in the top 3% of the records, then divided by the total number of fraud data.

Wrapped Method:

Backward step-by-step selection involves starting with all candidate variables, testing the deletion of each variable using the selection model to meet the criteria and repeating this process until no further variable can be eliminated. In the wrapped method, we use FDR score as the score in the function RFECV in order to have a better result. Below is a list of the 30 variables chosen by backward selection and are ones we used in our final models:

|  |  |  |  |
| --- | --- | --- | --- |
| Variable | Description | Variable | Description |
| address\_count\_0 | Number of same address seen in the past 0 days | fulladdress\_count\_0 | Number of same address plus zip code seen in the past 0 days |
| address\_count\_0\_by\_3 | Number of same address seen in the past 0 days divided by the same group in 3 days. | fulladdress\_count\_0\_by\_14 | Number of same full address and dob seen in the past 0 days divided by the same group in 14 days. |
| address\_count\_0\_by\_7 | Number of same address seen in the past 0 days divided by the same group in 7 days. | fulladdress\_count\_0\_by\_3 | Number of same full address and dob seen in the past 0 days divided by the same group in 3 days. |
| address\_count\_1 | Number of same address seen in the past 1 days | fulladdress\_count\_0\_by\_7 | Number of same full address and dob seen in the past 0 days divided by the same group in 7 days. |
| address\_count\_1\_by\_7 | Number of same address seen in the past 1 days divided by the same group in 7 days. | fulladdress\_count\_1 | Number of same address plus zip code seen in the past 1 days |
| address\_count\_3 | Number of same address seen in the past 3 days | fulladdress\_count\_3 | Number of same address plus zip code seen in the past 3 days |
| address\_count\_30 | Number of same address seen in the past 30 days | fulladdress\_count\_30 | Number of same address plus zip code seen in the past 30 days |
| homephone\_count\_3 | Number of same phone number seen in the past 3 days | fulladdress\_homephone\_count\_30 | Number of same address plus zip plus home phone number code seen in the past 30 days |
| name\_dob\_count\_0\_by\_14 | Number of same name and dob seen in the past 0 days divided by the same group in 14 days. | fulladdress\_homephone\_count\_7 | Number of same address plus zip plus home phone number code seen in the past 7 days |
| name\_dob\_count\_14 | Number of same address plus zip code seen in the past 30 days | ssn\_count\_0\_by\_30 | Number of same ssn in the past 0 days divided by the same group in 30 days. |
| name\_dob\_count\_30 | Number of same address plus zip code seen in the past 30 days | ssn\_count\_30 | Number of same ssn code seen in the past 30 days |
| name\_dob\_count\_7 | Number of same address plus zip code seen in the past 30 days | ssn\_count\_7 | Number of same ssn code seen in the past 7 days |
| ssn\_dob\_count\_0\_by\_14 | Number of same ssn and dob seen in the past 0 days divided by the same group in 14 days. | ssn\_dob\_count\_30 | Number of same SSN and date of birth seen in the past 30 days |
| ssn\_dob\_count\_0\_by\_30 | Number of same ssn and dob seen in the past 0 days divided by the same group in 30 days. | ssn\_dob\_count\_7 | Number of same SSN and date of birth seen in the past 7 days |
| ssn\_dob\_count\_14 | Number of same address plus zip code seen in the past 30 days | ssn\_firstname\_count\_14 | Number of same SSN and first name seen in the past 14 days |

# 

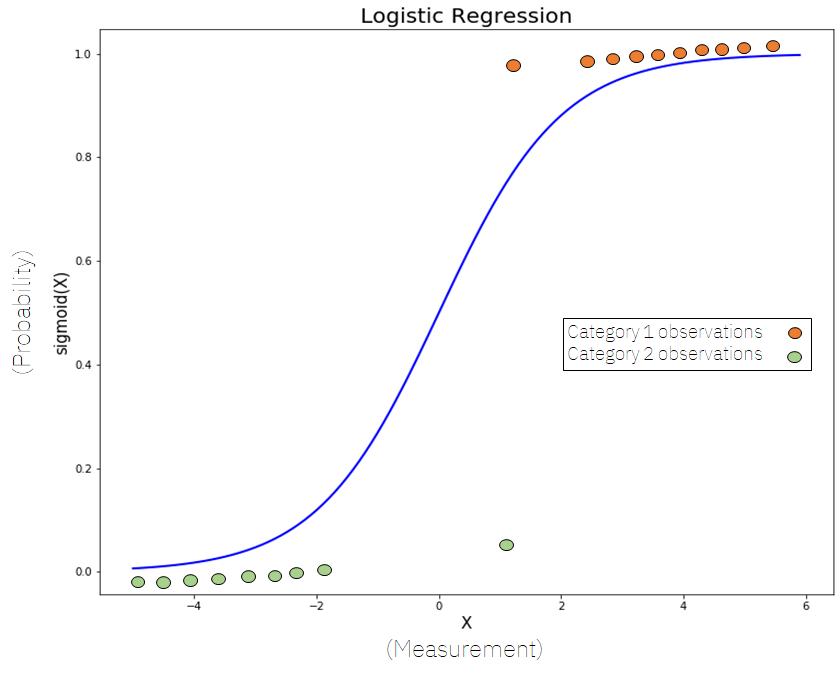
# Model Algorithms

After feature selection, we applied different machine learning algorithms to train models and calculated the average FDR at 3% for the train, test, and OOT data for each model. We applied 4 machine learning algorithms as below.

**Logistic Regression:**

Logistic regression analysis studies the relationship between a categorical dependent variable and a set of independent variables. Logistic regression can predict the probability of an outcome that has two values (i.e., 0 and 1).

In logistic regression, we don’t directly fit a straight line to our data like in linear regression. Instead, we fit a S-shaped curve, called Sigmoid, to our observations.

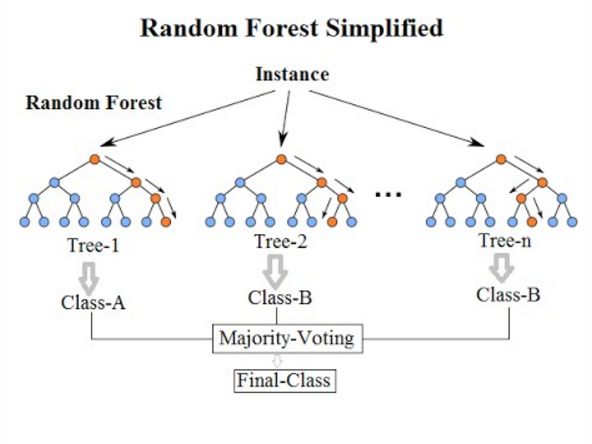


In the logistic regression the constant moves the curve left and right and the slope defines the steepness of the curve. By simple transformation, the logistic regression equation can be written as:

With the fraud label being the dependent variable and the 30 selected features being the independent variables, we trained the logistic regression model with parameter and . We then used the model to predict the probability of fraud on the train, test and oot dataset separately and get the best average FDR rate of 0.5445, 0.5364 and 0.5207 for each of the dataset.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Model | Parameter | | Average FDR(%) at 3% | | |
| Logistic Regression | Total variables | # of variables selected | Train | Test | OOT |
| 1 | 30 | 10 | 0.3577 | 0.3572 | 0.3206 |
| 2 | 30 | 20 | 0.5231 | 0.5213 | 0.5031 |
| 3 | 30 | 25 | 0.5436 | 0.5358 | 0.5194 |
| 4 | 30 | 30 | 0.5445 | 0.5364 | 0.5207 |

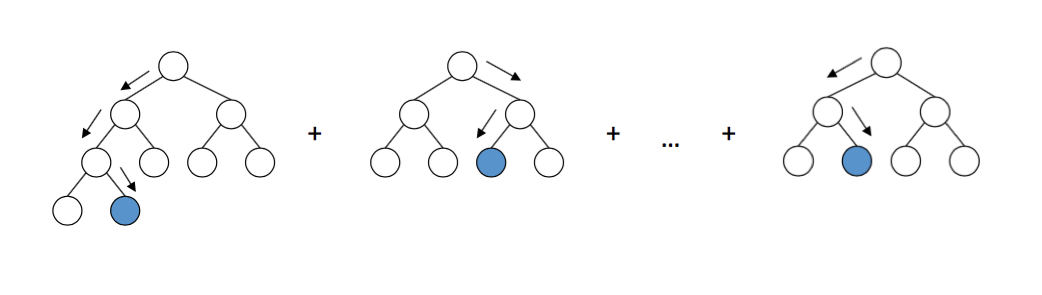
**Random Forests:**



Random forest is an integrated learning method based on classification and regression. It constructs a large number of decision trees during training and outputs the pattern of classes (classification) or average/average prediction of a single tree (regression). We used the randomForest package in the Python sklearn package. When we trained the random forest model, we tried generating 100, 200 and 300 decision trees with max\_depth of 60, 70, 80.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Random Forests | # of Vars | N\_estimator | Max\_depth | max\_features | Train | Test | OOT |
| 1 | 30 | 100 | 60 | 7 | 0.567 | 0.553 | 0.535 |
| 2 | 30 | 100 | 70 | 7 | 0.568 | 0.553 | 0.535 |
| 3 | 30 | 100 | 80 | 7 | 0.567 | 0.553 | 0.536 |
| 4 | 30 | 200 | 60 | 7 | 0.567 | 0.553 | 0.537 |
| 5 | 30 | 200 | 70 | 7 | 0.567 | 0.552 | 0.536 |
| 6 | 30 | 200 | 80 | 7 | 0.567 | 0.553 | 0.535 |
| 7 | 30 | 300 | 60 | 7 | 0.567 | 0.553 | 0.537 |
| 8 | 30 | 300 | 70 | 7 | 0.567 | 0.552 | 0.536 |
| 9 | 30 | 300 | 80 | 7 | 0.567 | 0.552 | 0.535 |

**Gradient Boosting Trees:**



Like other boosting methods, gradient boosting combines weak ‘learners’ into a single strong ‘learner’. The goal of this model is to predict values by minimizing the mean squared error. The model in supervised learning usually refers to the mathematical structure by which the prediction yi  is made from the input xi. Gradient boosting works by sequentially adding predictors to an ensemble, each one correcting its predecessor, i.e., each succeeding one attempts to fit the new predictor to the residual errors made by the previous one. In this project we use this model to be a classification model to determine whether it is fraud.

We use the GradientBoostingClassifier from the Scikit-learn package to predict the correct classification of fraud case and use the predict\_proba method to rank the predictions by their likelihood of being fraud before tallying the results and compare with the true count of fraud case in the test and oot set. As mentioned previously, this will be defined as the top 3% FDR rate and effectiveness of model selection will rely heavily on this metric.

In the quest for a best fitting model in this ensemble from Sci-kit Learn, 3 hyperparameters are adjusted to find the best possible outcome, namely,

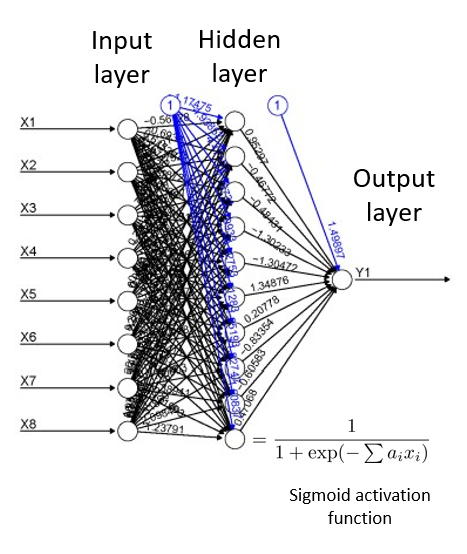
* max\_depth: the maximum length from a root to a node in a decision tree.
* n\_estimators: the number of trees in the ensemble.
* learning\_rate: a value to adjust the amount of information retained from each tree to compensate against overfitting.

We have tried 60 combinations of different hyperparameters. The max depths are 1,5,6, the numbers of trees are 700,800,1100,1400 and the learning rates goes from 0.01 to 0.1. We picked 5 models from the 60 models with the top 5 testing FDR.

In the following results, we found that Gradient Boosting Tree 1 with max\_depth = 5, N\_estimators = 1100, and leaning\_rate = 0.01 provided the most promising result where top 3%FDR equals 0.5592 and 0.5369 for the test and oot datasets, respectively.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Gradient Boosting Trees | Learning rate | n\_estimators | max\_depth | Training FDR | Testing FDR | OOT FDR |
| Gradient Boosting Tree 1 | 0.01 | 1100 | 5 | 0.5644 | 0.5592 | 0.5369 |
| Gradient Boosting Tree 2 | 0.01 | 800 | 6 | 0.5647 | 0.559 | 0.536 |
| Gradient Boosting Tree 3 | 0.01 | 800 | 5 | 0.564 | 0.5589 | 0.5369 |
| Gradient Boosting Tree 4 | 0.01 | 700 | 5 | 0.5637 | 0.5586 | 0.5369 |
| Gradient Boosting Tree 5 | 0.03 | 1400 | 1 | 0.5551 | 0.5523 | 0.5313 |

**Neural Net:**



Neural network is a machine learning algorithm that they make use of architecture that mimics how the neurons work in the brain. For example, a brain neuron receives an input and based on that input, fires off an output that is used by another neuron. The neural network simulates this behavior in learning about collecting the data and predicting outcomes.

In the above graph, we can see that a typical neural net consists of an input layer, hidden layers, and an output layer. The input layer is formed by all the independent variables. Each hidden layer is a set of nodes (neurons). Each neuron in the hidden layer receives weighted signals from all the nodes in the previous and transforms the linear combination of signals. The transform/activation function can be a logistic function(sigmoid) or something else. Finally, the output layer is the dependent variable.

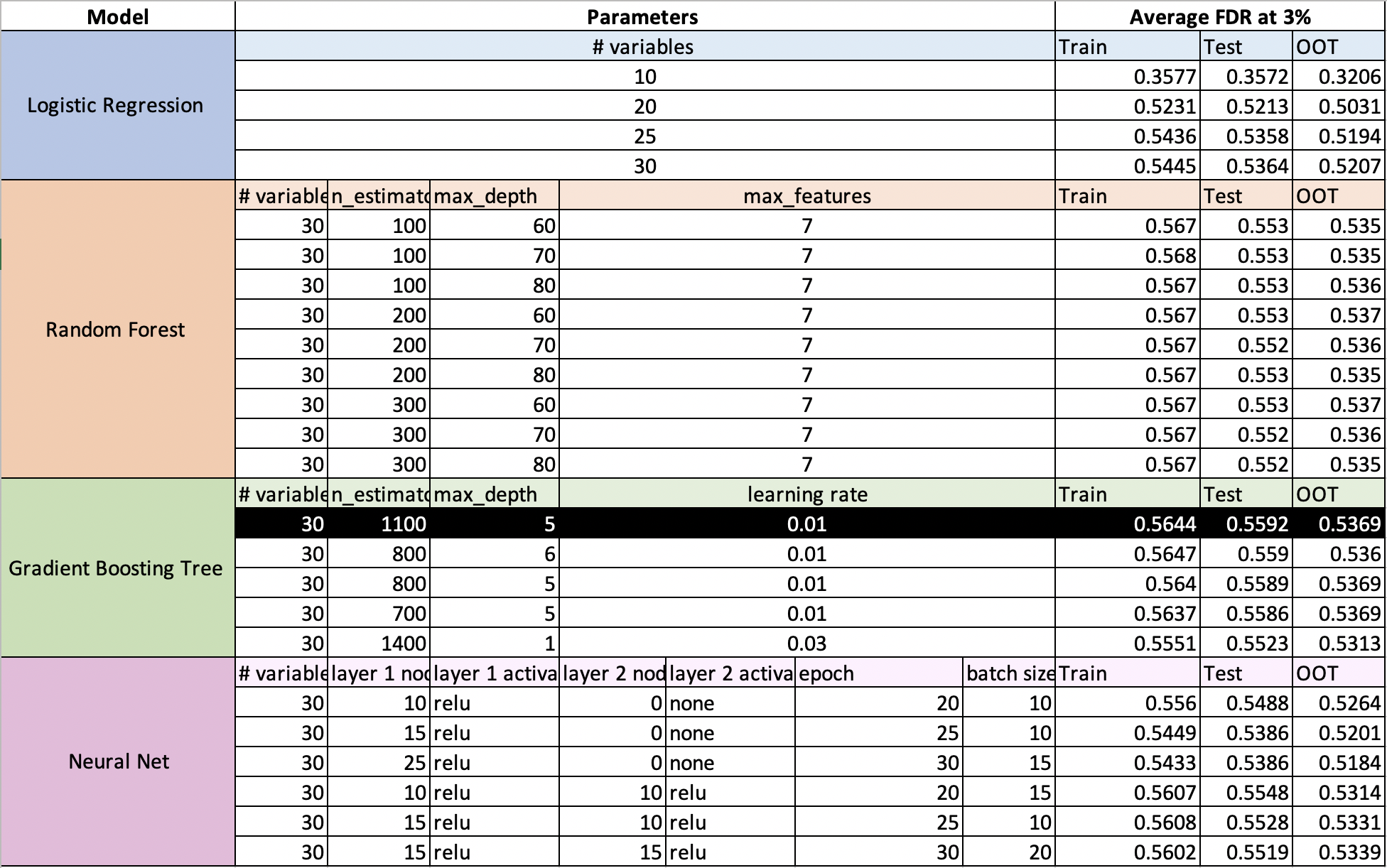
We trained six neural net models with different hyperparameters as in the following table and listed all average FDRs at 3% on training, testing, and oot data. According to each model’s performance, we didn’t observe overfitting so we could trust the results of these models. The best model is the highlighted Neural Net 6 with average FDR of 0.5339 on oot data.

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Neural Net | Nodes in layer 1 | Activation function layer 1 | Nodes in layer 2 | Activation function layer 2 | Activation function output layer | Optimizer | Epoch | Batch size | Training FDR | Testing FDR | OOT FDR |
| Neural net 1 | 10 | relu | None | None | sigmoid | adam | 20 | 10 | 0.556 | 0.5488 | 0.5264 |
| Neural net 2 | 15 | relu | None | None | sigmoid | adam | 25 | 10 | 0.5449 | 0.5386 | 0.5201 |
| Neural net 3 | 25 | relu | None | None | sigmoid | adam | 30 | 15 | 0.5433 | 0.5386 | 0.5184 |
| Neural net 4 | 10 | relu | 10 | relu | sigmoid | adam | 20 | 15 | 0.5607 | 0.5548 | 0.5314 |
| Neural net 5 | 15 | relu | 10 | relu | sigmoid | adam | 25 | 10 | 0.5608 | 0.5528 | 0.5331 |
| Neural net 6 | 15 | relu | 15 | relu | sigmoid | adam | 30 | 20 | 0.5602 | 0.5519 | 0.5339 |

# 

# Results

By training several models using different machine learning algorithms and adjusting hyperparameters for each particular machine learning algorithm, we compared the model performance based on average FDR at 3% on testing data and found that the best model is the gradient boosting tree as highlighted in the following graph. Of the attempted methods, a common peak value for the FDR at 3% is reached for the OOT dataset at around 53.5%. Since out of time data is considered unknown information in reality, we have little choice but to turn to the test set results. Random forest and Gradient Boosting Tree came in neck to neck and the latter won by a slight lead of around 0.5%. It is worth mentioning that the Gradient Boosting Tree is a more cultivated model in terms of number of trees and learning rate, making this choice both more time consuming and computationally demanding. Random Forest in comparison provided similar results with much less trees but deeper depth, demonstrating a diminishing marginal return on the complexity of model structure. We ultimately opted to use test set average FDR at 3% to be the guideline for the choice of model since algorithm efficiency and time limit is outside the scope of this project.

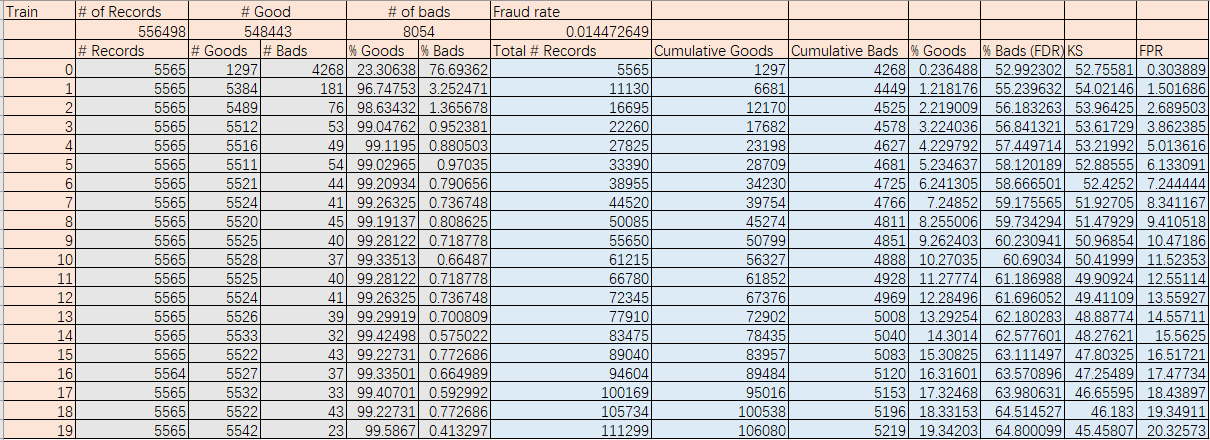


After selecting the first gradient boosting tree as our final model, we ran the model on training data again and reevaluated the model on both testing and oot data. We took a closer look at the structure of the predicted results on all three datasets by computing critical statistics by individual percentiles and then generated the tables below to examine how our final model performed on the three datasets when detecting fraudulent applications.

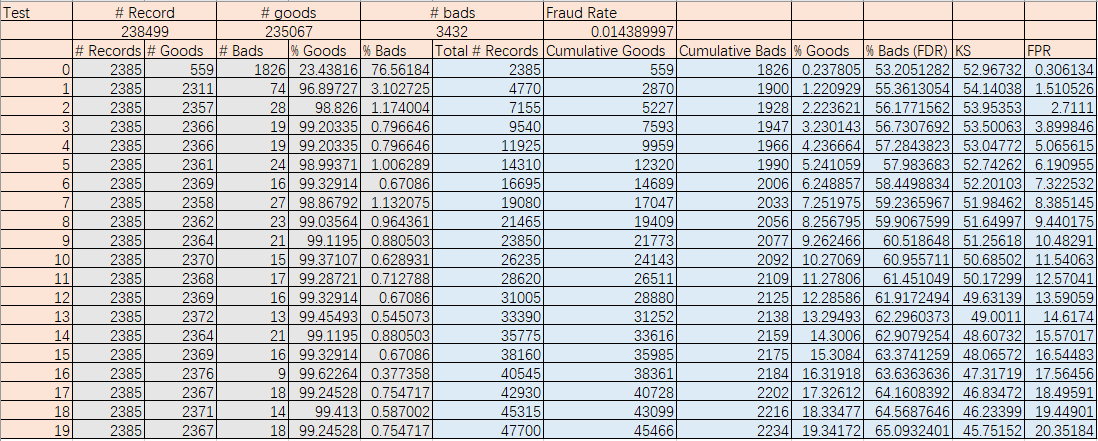
As a brief review, the oot data represented approximately 16.65% of the entire dataset, the remaining are split into training and testing by 75% and 25%, respectively. The underlying fraud rate is consistent for all three sets at around 1.43%, or 1 fraud case in about 70 applications.

More specifically, the first percentile of data ranked by our model of choice uniformly contained the largest percentage in true frauds identified, up to 76.7% in training, 76.56% in testing and 72.71% in OOT. This means that approximately 3 out of 4 applications that we deny arbitrarily without additional information in the first percentile is a true fraud case. This is an immediate choice for any financial institution where no other alternative is present. The gains in fraud detection percentage quickly diminishes in subsequent percentiles, however; before we penetrate through the first 4% of all applications, the true percentage of fraud in each additional percentile of data falls below 1%. As a comparison, the company now looks at a proposal much less tempting: give up 5% of all applications in the hope of catching up to 57.28% of all frauds, still good if the institution’s profit margin prevails but nowhere near as good if the model’s effectiveness holds through.

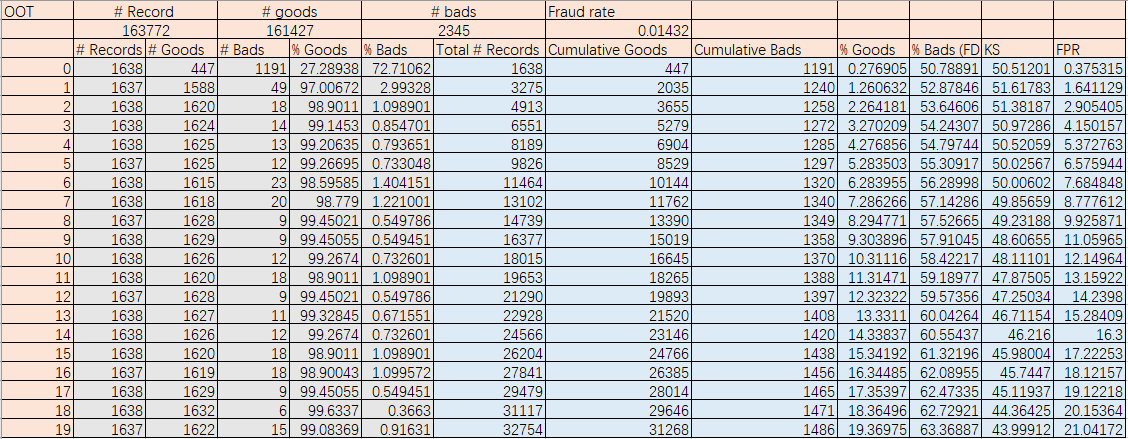
**Model Performance on Training**



**Model Performance on Testing**



**Model Performance on OOT**



# 

# Conclusions

The plethora of methods through which frauds are committed are unfathomable to recount; however, there are traces that remain to be picked up by the vigilant and learned. In this project, we attempted to predict the underlying fraudulent activity by observing none other than the information filled out on the applications. Via well-traversed methodologies developed by our seniors and predecessors, we were able to peek at the subtleties involuntarily revealed by the schemers, magnify their actions with tools based in both statistics and machine learning methods, and acutely label the potentially guilty. While the end result proves to be slightly better than a random toss of coin beyond a certain cutoff, these methods semented a handle on a problem otherwise extremely difficult to tackle. Moreover, we duly believe that with more experience on the subject matter and techniques, we could significantly improve the end results by creating better expert variables and implementing more appropriate data cleaning techniques and machine learning algorithms in a more polished manner. Also observed was that machine learning algorithms tend to perform better with datasets large in both volume and diversity and with models that churn through the numbers laboriously. These are invaluable experiences that we will be utilizing faithfully in the upcoming projects in the hope of a more streamlined process and more indicative results.

# Appendix

Data Quality Report

File Description:

Applications Data is a dataset containing records of 1,000,000 applications. It includes fields such as date of application, SSN, first and last name, address, zip code, date of birth, home phone number, and fraud label of each applicant.

File Name: applications data.csv

Number of Records: 1,000,000 records

Number of Fields: 9 variables in total: 7 categorical variables, 2 date variables

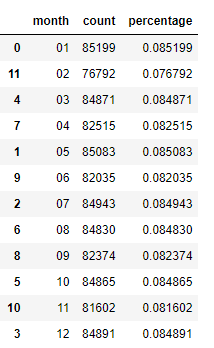
|  |  |  |  |
| --- | --- | --- | --- |
| Field Name | Data Type | % Populated | Unique number |
| date | Date variable | 100% | 365 |
| ssn | Categorical variable | 100% | 835819 |
| firstname | Categorical variable | 100% | 78136 |
| lastname | Categorical variable | 100% | 177001 |
| address | Categorical variable | 100% | 828774 |
| zip5 | Categorical variable | 100% | 26370 |
| dob | Date variable | 100% | 42673 |
| homephone | Categorical variable | 100% | 28244 |
| fraud\_label | Categorical variable | 100% | 2 |

Time of Records: Jan 1st 2016 – Dec 31st 2016

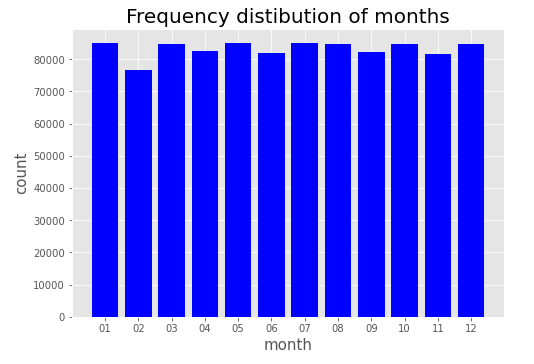
Field 1

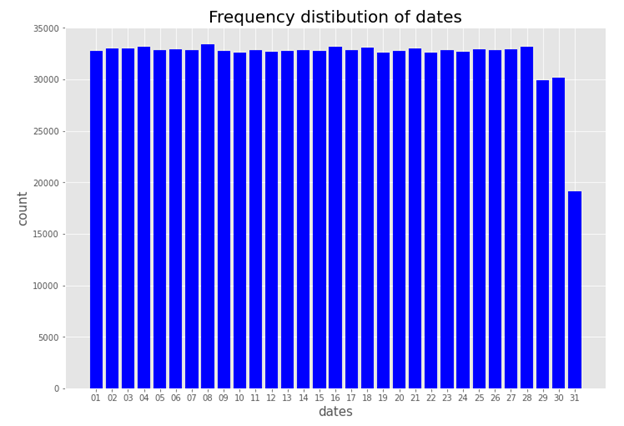
Field Name: date

Description:“date” is a date variable, including the application dates.



Plot the distribution of the months and dates as below, the overall frequency is consistent.



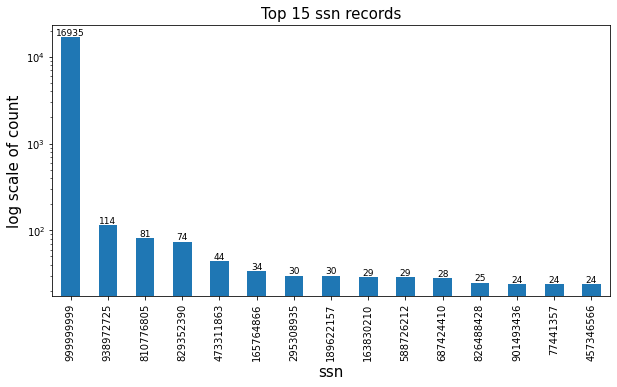
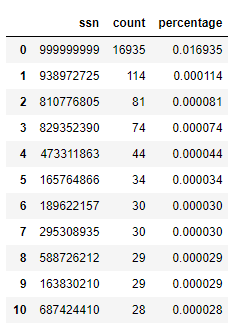


Field 2

Field Name: ssn

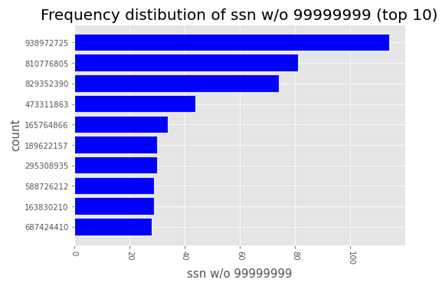
Description:“ssn” is a categorical variable, indicating the applicant’s ssn number.

The top 10 most frequent records are listed below:



Since the number of 999999999 takes most of the count and it means the record is frivolous, we plot the graph which exclude the record of 99999999.

The distribution of the top 10 records (excluding the most frequent record “999999999”):

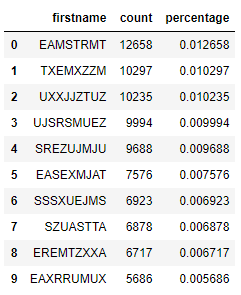


Field 3

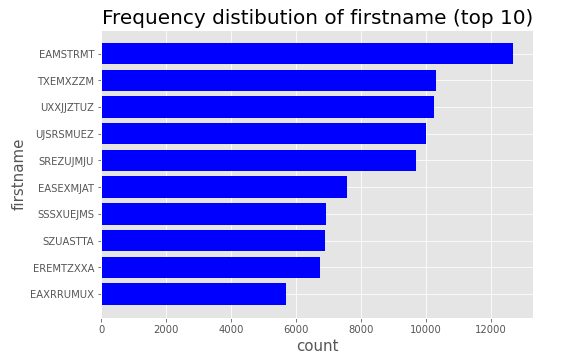
Field Name: firstname

Description:

“firstname” is a categorical variable, indicating the applicant’s first name.



The top 10 most frequent records are listed below:

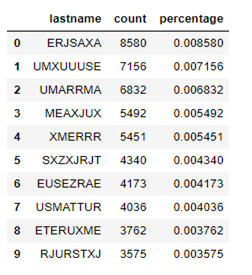


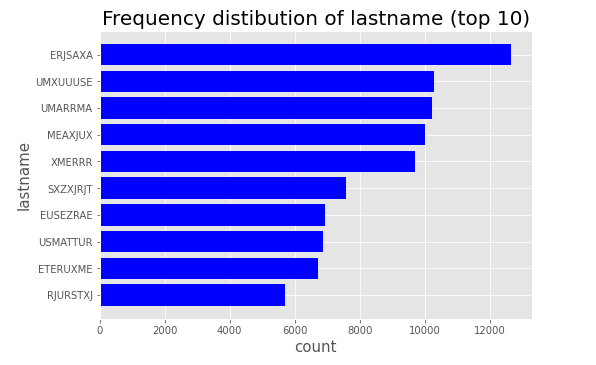
Field 4

Field Name: lastname

Description: “lastname” is a categorical variable, indicating the applicant’s last name.

The top 10 most frequent records are listed below:



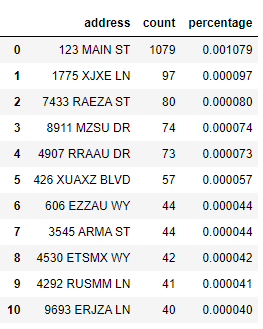


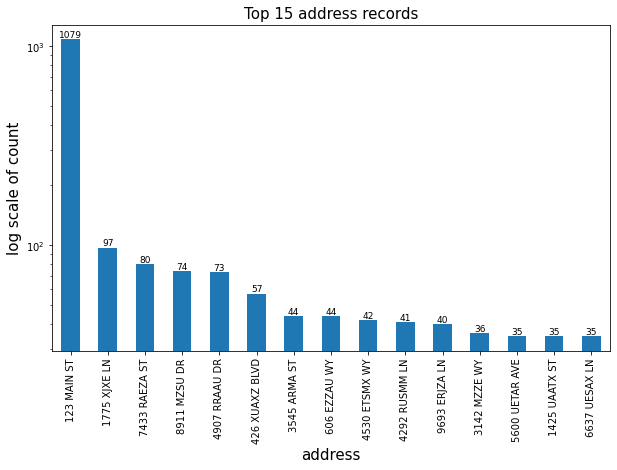
Field 5

Field Name: address

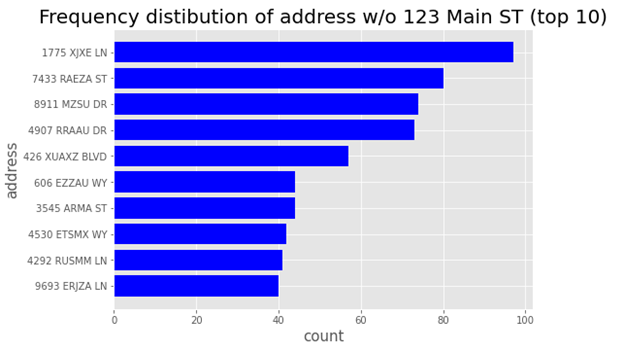
Description: “address” is a categorical variable that contains inputs from applicants of their address.

The top 10 most frequent records are listed below:





The distribution of the top 10 records (excluding the most frequent record “123 MAIN ST”):

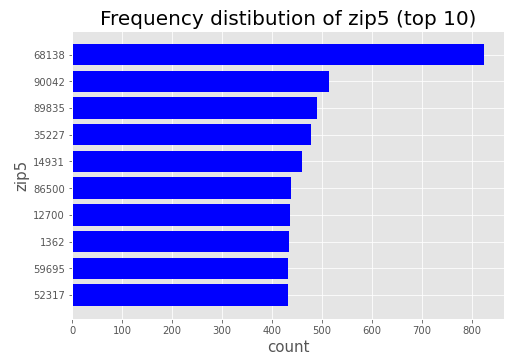


Field 6

Field Name: zip5

Description: “zip5” is a categorical variable that contains inputs from applicants of their zip codes.



The distribution of the top 10 records:

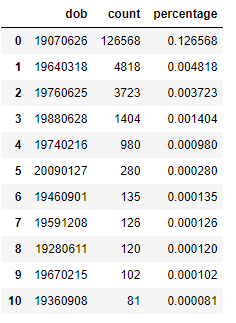
Field 7

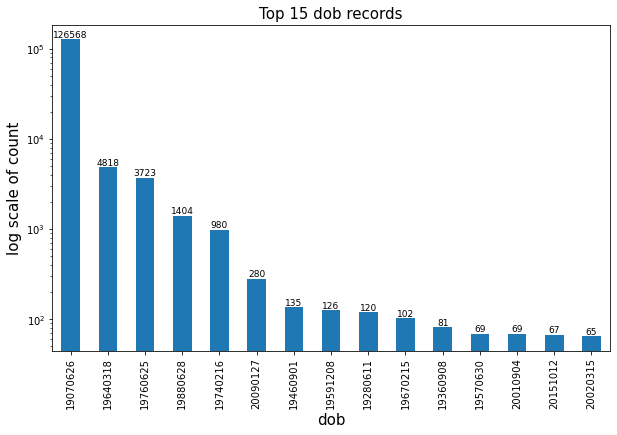
Field Name: dob

Description:

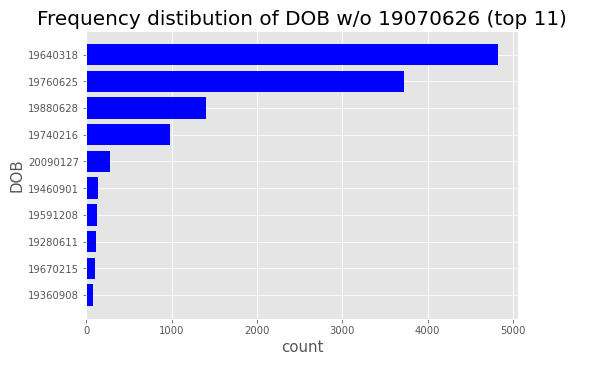
“dob” is a date variable that contains inputs from applicants of their date of birth.

The top 10 most frequent records are listed below:

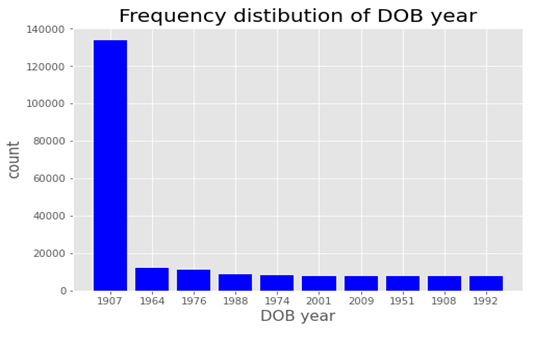




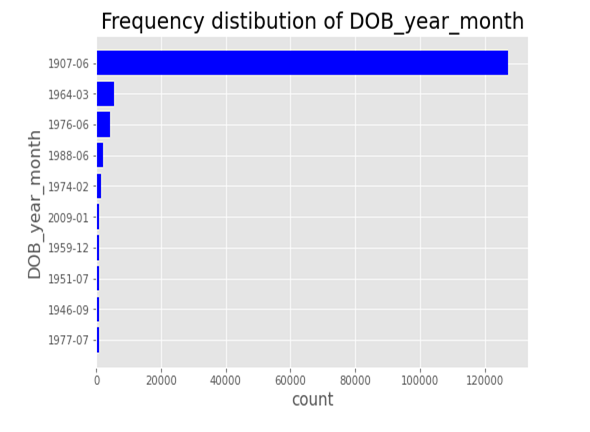
The distribution of the top 10 records (excluding the most frequent record “19070626”):



Also, the distribution of different DOB years is plotted below.



The distribution of different DOB years and months is plotted below

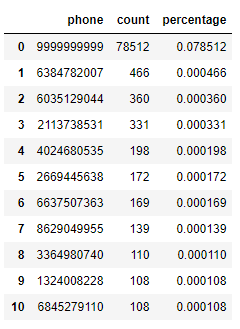


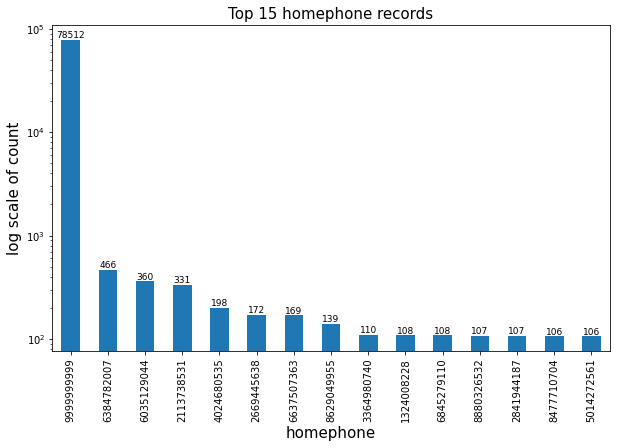
Field 8

Field Name: homephone

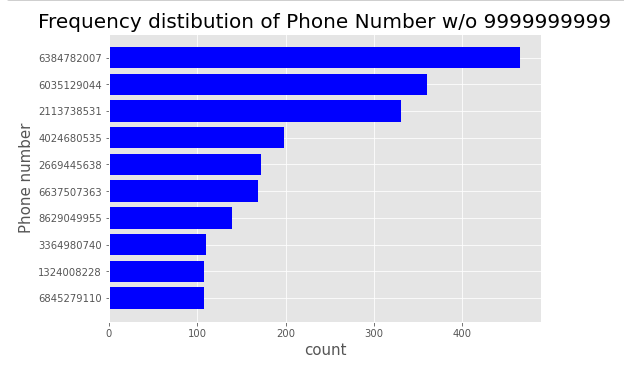
Description: “homephone” is a categorical variable that contains home phone number of each applicant.

The top 10 most frequent records are listed below:





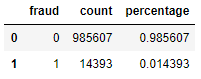
The distribution of the top 10 records (excluding the most frequent record “99999999999”):

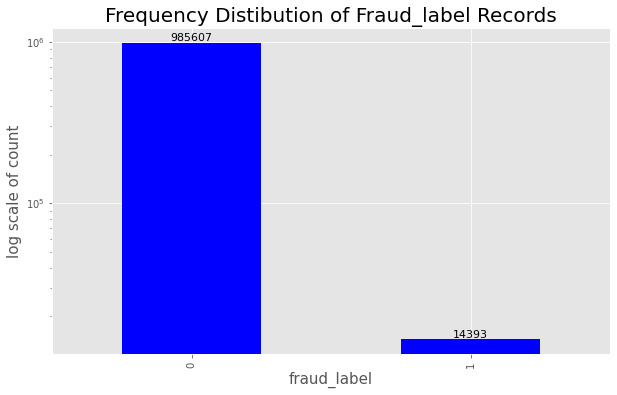


Field 9

Field Name: Fraud Label

Description:“fraud\_label” is a categorical variable whether the applicant is fraud.





Candidate Variables

|  |  |  |  |
| --- | --- | --- | --- |
| Velocity Candidate Variables | | | |
| 1 | ssn\_count\_0 | 85 | ssn\_lastname\_count\_0 |
| 2 | ssn\_count\_1 | 86 | ssn\_lastname\_count\_1 |
| 3 | ssn\_count\_3 | 87 | ssn\_lastname\_count\_3 |
| 4 | ssn\_count\_7 | 88 | ssn\_lastname\_count\_7 |
| 5 | ssn\_count\_14 | 89 | ssn\_lastname\_count\_14 |
| 6 | ssn\_count\_30 | 90 | ssn\_lastname\_count\_30 |
| 7 | address\_count\_0 | 91 | ssn\_address\_count\_0 |
| 8 | address\_count\_1 | 92 | ssn\_address\_count\_1 |
| 9 | address\_count\_3 | 93 | ssn\_address\_count\_3 |
| 10 | address\_count\_7 | 94 | ssn\_address\_count\_7 |
| 11 | address\_count\_14 | 95 | ssn\_address\_count\_14 |
| 12 | address\_count\_30 | 96 | ssn\_address\_count\_30 |
| 13 | dob\_count\_0 | 97 | ssn\_zip5\_count\_0 |
| 14 | dob\_count\_1 | 98 | ssn\_zip5\_count\_1 |
| 15 | dob\_count\_3 | 99 | ssn\_zip5\_count\_3 |
| 16 | dob\_count\_7 | 100 | ssn\_zip5\_count\_7 |
| 17 | dob\_count\_14 | 101 | ssn\_zip5\_count\_14 |
| 18 | dob\_count\_30 | 102 | ssn\_zip5\_count\_30 |
| 19 | homephone\_count\_0 | 103 | ssn\_dob\_count\_0 |
| 20 | homephone\_count\_1 | 104 | ssn\_dob\_count\_1 |
| 21 | homephone\_count\_3 | 105 | ssn\_dob\_count\_3 |
| 22 | homephone\_count\_7 | 106 | ssn\_dob\_count\_7 |
| 23 | homephone\_count\_14 | 107 | ssn\_dob\_count\_14 |
| 24 | homephone\_count\_30 | 108 | ssn\_dob\_count\_30 |
| 25 | name\_count\_0 | 109 | ssn\_homephone\_count\_0 |
| 26 | name\_count\_1 | 110 | ssn\_homephone\_count\_1 |
| 27 | name\_count\_3 | 111 | ssn\_homephone\_count\_3 |
| 28 | name\_count\_7 | 112 | ssn\_homephone\_count\_7 |
| 29 | name\_count\_14 | 113 | ssn\_homephone\_count\_14 |
| 30 | name\_count\_30 | 114 | ssn\_homephone\_count\_30 |
| 31 | fulladdress\_count\_0 | 115 | ssn\_name\_count\_0 |
| 32 | fulladdress\_count\_1 | 116 | ssn\_name\_count\_1 |
| 33 | fulladdress\_count\_3 | 117 | ssn\_name\_count\_3 |
| 34 | fulladdress\_count\_7 | 118 | ssn\_name\_count\_7 |
| 35 | fulladdress\_count\_14 | 119 | ssn\_name\_count\_14 |
| 36 | fulladdress\_count\_30 | 120 | ssn\_name\_count\_30 |
| 37 | name\_dob\_count\_0 | 121 | ssn\_fulladdress\_count\_0 |
| 38 | name\_dob\_count\_1 | 122 | ssn\_fulladdress\_count\_1 |
| 39 | name\_dob\_count\_3 | 123 | ssn\_fulladdress\_count\_3 |
| 40 | name\_dob\_count\_7 | 124 | ssn\_fulladdress\_count\_7 |
| 41 | name\_dob\_count\_14 | 125 | ssn\_fulladdress\_count\_14 |
| 42 | name\_dob\_count\_30 | 126 | ssn\_fulladdress\_count\_30 |
| 43 | name\_fulladdress\_count\_0 | 127 | ssn\_name\_dob\_count\_0 |
| 44 | name\_fulladdress\_count\_1 | 128 | ssn\_name\_dob\_count\_1 |
| 45 | name\_fulladdress\_count\_3 | 129 | ssn\_name\_dob\_count\_3 |
| 46 | name\_fulladdress\_count\_7 | 130 | ssn\_name\_dob\_count\_7 |
| 47 | name\_fulladdress\_count\_14 | 131 | ssn\_name\_dob\_count\_14 |
| 48 | name\_fulladdress\_count\_30 | 132 | ssn\_name\_dob\_count\_30 |
| 49 | name\_homephone\_count\_0 | 133 | ssn\_name\_fulladdress\_count\_0 |
| 50 | name\_homephone\_count\_1 | 134 | ssn\_name\_fulladdress\_count\_1 |
| 51 | name\_homephone\_count\_3 | 135 | ssn\_name\_fulladdress\_count\_3 |
| 52 | name\_homephone\_count\_7 | 136 | ssn\_name\_fulladdress\_count\_7 |
| 53 | name\_homephone\_count\_14 | 137 | ssn\_name\_fulladdress\_count\_14 |
| 54 | name\_homephone\_count\_30 | 138 | ssn\_name\_fulladdress\_count\_30 |
| 55 | fulladdress\_dob\_count\_0 | 139 | ssn\_name\_homephone\_count\_0 |
| 56 | fulladdress\_dob\_count\_1 | 140 | ssn\_name\_homephone\_count\_1 |
| 57 | fulladdress\_dob\_count\_3 | 141 | ssn\_name\_homephone\_count\_3 |
| 58 | fulladdress\_dob\_count\_7 | 142 | ssn\_name\_homephone\_count\_7 |
| 59 | fulladdress\_dob\_count\_14 | 143 | ssn\_name\_homephone\_count\_14 |
| 60 | fulladdress\_dob\_count\_30 | 144 | ssn\_name\_homephone\_count\_30 |
| 61 | fulladdress\_homephone\_count\_0 | 145 | ssn\_fulladdress\_dob\_count\_0 |
| 62 | fulladdress\_homephone\_count\_1 | 146 | ssn\_fulladdress\_dob\_count\_1 |
| 63 | fulladdress\_homephone\_count\_3 | 147 | ssn\_fulladdress\_dob\_count\_3 |
| 64 | fulladdress\_homephone\_count\_7 | 148 | ssn\_fulladdress\_dob\_count\_7 |
| 65 | fulladdress\_homephone\_count\_14 | 149 | ssn\_fulladdress\_dob\_count\_14 |
| 66 | fulladdress\_homephone\_count\_30 | 150 | ssn\_fulladdress\_dob\_count\_30 |
| 67 | dob\_homephone\_count\_0 | 151 | ssn\_fulladdress\_homephone\_count\_0 |
| 68 | dob\_homephone\_count\_1 | 152 | ssn\_fulladdress\_homephone\_count\_1 |
| 69 | dob\_homephone\_count\_3 | 153 | ssn\_fulladdress\_homephone\_count\_3 |
| 70 | dob\_homephone\_count\_7 | 154 | ssn\_fulladdress\_homephone\_count\_7 |
| 71 | dob\_homephone\_count\_14 | 155 | ssn\_fulladdress\_homephone\_count\_14 |
| 72 | dob\_homephone\_count\_30 | 156 | ssn\_fulladdress\_homephone\_count\_30 |
| 73 | homephone\_name\_dob\_count\_0 | 157 | ssn\_dob\_homephone\_count\_0 |
| 74 | homephone\_name\_dob\_count\_1 | 158 | ssn\_dob\_homephone\_count\_1 |
| 75 | homephone\_name\_dob\_count\_3 | 159 | ssn\_dob\_homephone\_count\_3 |
| 76 | homephone\_name\_dob\_count\_7 | 160 | ssn\_dob\_homephone\_count\_7 |
| 77 | homephone\_name\_dob\_count\_14 | 161 | ssn\_dob\_homephone\_count\_14 |
| 78 | homephone\_name\_dob\_count\_30 | 162 | ssn\_dob\_homephone\_count\_30 |
| 79 | ssn\_firstname\_count\_0 | 163 | ssn\_homephone\_name\_dob\_count\_0 |
| 80 | ssn\_firstname\_count\_1 | 164 | ssn\_homephone\_name\_dob\_count\_1 |
| 81 | ssn\_firstname\_count\_3 | 165 | ssn\_homephone\_name\_dob\_count\_3 |
| 82 | ssn\_firstname\_count\_7 | 166 | ssn\_homephone\_name\_dob\_count\_7 |
| 83 | ssn\_firstname\_count\_14 | 167 | ssn\_homephone\_name\_dob\_count\_14 |
| 84 | ssn\_firstname\_count\_30 | 168 | ssn\_homephone\_name\_dob\_count\_30 |
| Relative Velocity Candidate Variables | | | |
| 169 | ssn\_count\_0\_by\_3 | 276 | ssn\_firstname\_count\_0\_by\_30 |
| 170 | ssn\_count\_0\_by\_7 | 277 | ssn\_firstname\_count\_1\_by\_3 |
| 171 | ssn\_count\_0\_by\_14 | 278 | ssn\_firstname\_count\_1\_by\_7 |
| 172 | ssn\_count\_0\_by\_30 | 279 | ssn\_firstname\_count\_1\_by\_14 |
| 173 | ssn\_count\_1\_by\_3 | 280 | ssn\_firstname\_count\_1\_by\_30 |
| 174 | ssn\_count\_1\_by\_7 | 281 | ssn\_lastname\_count\_0\_by\_3 |
| 175 | ssn\_count\_1\_by\_14 | 282 | ssn\_lastname\_count\_0\_by\_7 |
| 176 | ssn\_count\_1\_by\_30 | 283 | ssn\_lastname\_count\_0\_by\_14 |
| 177 | address\_count\_0\_by\_3 | 284 | ssn\_lastname\_count\_0\_by\_30 |
| 178 | address\_count\_0\_by\_7 | 285 | ssn\_lastname\_count\_1\_by\_3 |
| 179 | address\_count\_0\_by\_14 | 286 | ssn\_lastname\_count\_1\_by\_7 |
| 180 | address\_count\_0\_by\_30 | 287 | ssn\_lastname\_count\_1\_by\_14 |
| 181 | address\_count\_1\_by\_3 | 288 | ssn\_lastname\_count\_1\_by\_30 |
| 182 | address\_count\_1\_by\_7 | 289 | ssn\_address\_count\_0\_by\_3 |
| 183 | address\_count\_1\_by\_14 | 290 | ssn\_address\_count\_0\_by\_7 |
| 184 | address\_count\_1\_by\_30 | 291 | ssn\_address\_count\_0\_by\_14 |
| 185 | dob\_count\_0\_by\_3 | 292 | ssn\_address\_count\_0\_by\_30 |
| 186 | dob\_count\_0\_by\_7 | 293 | ssn\_address\_count\_1\_by\_3 |
| 187 | dob\_count\_0\_by\_14 | 294 | ssn\_address\_count\_1\_by\_7 |
| 188 | dob\_count\_0\_by\_30 | 295 | ssn\_address\_count\_1\_by\_14 |
| 189 | dob\_count\_1\_by\_3 | 296 | ssn\_address\_count\_1\_by\_30 |
| 190 | dob\_count\_1\_by\_7 | 297 | ssn\_zip5\_count\_0\_by\_3 |
| 191 | dob\_count\_1\_by\_14 | 298 | ssn\_zip5\_count\_0\_by\_7 |
| 192 | dob\_count\_1\_by\_30 | 299 | ssn\_zip5\_count\_0\_by\_14 |
| 193 | homephone\_count\_0\_by\_3 | 300 | ssn\_zip5\_count\_0\_by\_30 |
| 194 | homephone\_count\_0\_by\_7 | 301 | ssn\_zip5\_count\_1\_by\_3 |
| 195 | homephone\_count\_0\_by\_14 | 302 | ssn\_zip5\_count\_1\_by\_7 |
| 196 | homephone\_count\_0\_by\_30 | 303 | ssn\_zip5\_count\_1\_by\_14 |
| 197 | homephone\_count\_1\_by\_3 | 304 | ssn\_zip5\_count\_1\_by\_30 |
| 198 | homephone\_count\_1\_by\_7 | 305 | ssn\_dob\_count\_0\_by\_3 |
| 199 | homephone\_count\_1\_by\_14 | 306 | ssn\_dob\_count\_0\_by\_7 |
| 200 | homephone\_count\_1\_by\_30 | 307 | ssn\_dob\_count\_0\_by\_14 |
| 201 | name\_count\_0\_by\_3 | 308 | ssn\_dob\_count\_0\_by\_30 |
| 202 | name\_count\_0\_by\_7 | 309 | ssn\_dob\_count\_1\_by\_3 |
| 203 | name\_count\_0\_by\_14 | 310 | ssn\_dob\_count\_1\_by\_7 |
| 204 | name\_count\_0\_by\_30 | 311 | ssn\_dob\_count\_1\_by\_14 |
| 205 | name\_count\_1\_by\_3 | 312 | ssn\_dob\_count\_1\_by\_30 |
| 206 | name\_count\_1\_by\_7 | 313 | ssn\_homephone\_count\_0\_by\_3 |
| 207 | name\_count\_1\_by\_14 | 314 | ssn\_homephone\_count\_0\_by\_7 |
| 208 | name\_count\_1\_by\_30 | 315 | ssn\_homephone\_count\_0\_by\_14 |
| 209 | fulladdress\_count\_0\_by\_3 | 316 | ssn\_homephone\_count\_0\_by\_30 |
| 210 | fulladdress\_count\_0\_by\_7 | 317 | ssn\_homephone\_count\_1\_by\_3 |
| 211 | fulladdress\_count\_0\_by\_14 | 318 | ssn\_homephone\_count\_1\_by\_7 |
| 212 | fulladdress\_count\_0\_by\_30 | 319 | ssn\_homephone\_count\_1\_by\_14 |
| 213 | fulladdress\_count\_1\_by\_3 | 320 | ssn\_homephone\_count\_1\_by\_30 |
| 214 | fulladdress\_count\_1\_by\_7 | 321 | ssn\_name\_count\_0\_by\_3 |
| 215 | fulladdress\_count\_1\_by\_14 | 322 | ssn\_name\_count\_0\_by\_7 |
| 216 | fulladdress\_count\_1\_by\_30 | 323 | ssn\_name\_count\_0\_by\_14 |
| 217 | name\_dob\_count\_0\_by\_3 | 324 | ssn\_name\_count\_0\_by\_30 |
| 218 | name\_dob\_count\_0\_by\_7 | 325 | ssn\_name\_count\_1\_by\_3 |
| 219 | name\_dob\_count\_0\_by\_14 | 326 | ssn\_name\_count\_1\_by\_7 |
| 220 | name\_dob\_count\_0\_by\_30 | 327 | ssn\_name\_count\_1\_by\_14 |
| 221 | name\_dob\_count\_1\_by\_3 | 328 | ssn\_name\_count\_1\_by\_30 |
| 222 | name\_dob\_count\_1\_by\_7 | 329 | ssn\_fulladdress\_count\_0\_by\_3 |
| 223 | name\_dob\_count\_1\_by\_14 | 330 | ssn\_fulladdress\_count\_0\_by\_7 |
| 224 | name\_dob\_count\_1\_by\_30 | 331 | ssn\_fulladdress\_count\_0\_by\_14 |
| 225 | name\_fulladdress\_count\_0\_by\_3 | 332 | ssn\_fulladdress\_count\_0\_by\_30 |
| 226 | name\_fulladdress\_count\_0\_by\_7 | 333 | ssn\_fulladdress\_count\_1\_by\_3 |
| 227 | name\_fulladdress\_count\_0\_by\_14 | 334 | ssn\_fulladdress\_count\_1\_by\_7 |
| 228 | name\_fulladdress\_count\_0\_by\_30 | 335 | ssn\_fulladdress\_count\_1\_by\_14 |
| 229 | name\_fulladdress\_count\_1\_by\_3 | 336 | ssn\_fulladdress\_count\_1\_by\_30 |
| 230 | name\_fulladdress\_count\_1\_by\_7 | 337 | ssn\_name\_dob\_count\_0\_by\_3 |
| 231 | name\_fulladdress\_count\_1\_by\_14 | 338 | ssn\_name\_dob\_count\_0\_by\_7 |
| 232 | name\_fulladdress\_count\_1\_by\_30 | 339 | ssn\_name\_dob\_count\_0\_by\_14 |
| 233 | name\_homephone\_count\_0\_by\_3 | 340 | ssn\_name\_dob\_count\_0\_by\_30 |
| 234 | name\_homephone\_count\_0\_by\_7 | 341 | ssn\_name\_dob\_count\_1\_by\_3 |
| 235 | name\_homephone\_count\_0\_by\_14 | 342 | ssn\_name\_dob\_count\_1\_by\_7 |
| 236 | name\_homephone\_count\_0\_by\_30 | 343 | ssn\_name\_dob\_count\_1\_by\_14 |
| 237 | name\_homephone\_count\_1\_by\_3 | 344 | ssn\_name\_dob\_count\_1\_by\_30 |
| 238 | name\_homephone\_count\_1\_by\_7 | 345 | ssn\_name\_fulladdress\_count\_0\_by\_3 |
| 239 | name\_homephone\_count\_1\_by\_14 | 346 | ssn\_name\_fulladdress\_count\_0\_by\_7 |
| 240 | name\_homephone\_count\_1\_by\_30 | 347 | ssn\_name\_fulladdress\_count\_0\_by\_14 |
| 241 | fulladdress\_dob\_count\_0\_by\_3 | 348 | ssn\_name\_fulladdress\_count\_0\_by\_30 |
| 242 | fulladdress\_dob\_count\_0\_by\_7 | 349 | ssn\_name\_fulladdress\_count\_1\_by\_3 |
| 243 | fulladdress\_dob\_count\_0\_by\_14 | 350 | ssn\_name\_fulladdress\_count\_1\_by\_7 |
| 244 | fulladdress\_dob\_count\_0\_by\_30 | 351 | ssn\_name\_fulladdress\_count\_1\_by\_14 |
| 245 | fulladdress\_dob\_count\_1\_by\_3 | 352 | ssn\_name\_fulladdress\_count\_1\_by\_30 |
| 246 | fulladdress\_dob\_count\_1\_by\_7 | 353 | ssn\_name\_homephone\_count\_0\_by\_3 |
| 247 | fulladdress\_dob\_count\_1\_by\_14 | 354 | ssn\_name\_homephone\_count\_0\_by\_7 |
| 248 | fulladdress\_dob\_count\_1\_by\_30 | 355 | ssn\_name\_homephone\_count\_0\_by\_14 |
| 249 | fulladdress\_homephone\_count\_0\_by\_3 | 356 | ssn\_name\_homephone\_count\_0\_by\_30 |
| 250 | fulladdress\_homephone\_count\_0\_by\_7 | 357 | ssn\_name\_homephone\_count\_1\_by\_3 |
| 251 | fulladdress\_homephone\_count\_0\_by\_14 | 358 | ssn\_name\_homephone\_count\_1\_by\_7 |
| 252 | fulladdress\_homephone\_count\_0\_by\_30 | 359 | ssn\_name\_homephone\_count\_1\_by\_14 |
| 253 | fulladdress\_homephone\_count\_1\_by\_3 | 360 | ssn\_name\_homephone\_count\_1\_by\_30 |
| 254 | fulladdress\_homephone\_count\_1\_by\_7 | 361 | ssn\_fulladdress\_dob\_count\_0\_by\_3 |
| 255 | fulladdress\_homephone\_count\_1\_by\_14 | 362 | ssn\_fulladdress\_dob\_count\_0\_by\_7 |
| 256 | fulladdress\_homephone\_count\_1\_by\_30 | 363 | ssn\_fulladdress\_dob\_count\_0\_by\_14 |
| 257 | dob\_homephone\_count\_0\_by\_3 | 364 | ssn\_fulladdress\_dob\_count\_0\_by\_30 |
| 258 | dob\_homephone\_count\_0\_by\_7 | 365 | ssn\_fulladdress\_dob\_count\_1\_by\_3 |
| 259 | dob\_homephone\_count\_0\_by\_14 | 366 | ssn\_fulladdress\_dob\_count\_1\_by\_7 |
| 260 | dob\_homephone\_count\_0\_by\_30 | 367 | ssn\_fulladdress\_dob\_count\_1\_by\_14 |
| 261 | dob\_homephone\_count\_1\_by\_3 | 368 | ssn\_fulladdress\_dob\_count\_1\_by\_30 |
| 262 | dob\_homephone\_count\_1\_by\_7 | 369 | ssn\_fulladdress\_homephone\_count\_0\_by\_3 |
| 263 | dob\_homephone\_count\_1\_by\_14 | 370 | ssn\_fulladdress\_homephone\_count\_0\_by\_7 |
| 264 | dob\_homephone\_count\_1\_by\_30 | 371 | ssn\_fulladdress\_homephone\_count\_0\_by\_14 |
| 265 | homephone\_name\_dob\_count\_0\_by\_3 | 372 | ssn\_fulladdress\_homephone\_count\_0\_by\_30 |
| 266 | homephone\_name\_dob\_count\_0\_by\_7 | 373 | ssn\_fulladdress\_homephone\_count\_1\_by\_3 |
| 267 | homephone\_name\_dob\_count\_0\_by\_14 | 374 | ssn\_fulladdress\_homephone\_count\_1\_by\_7 |
| 268 | homephone\_name\_dob\_count\_0\_by\_30 | 375 | ssn\_fulladdress\_homephone\_count\_1\_by\_14 |
| 269 | homephone\_name\_dob\_count\_1\_by\_3 | 376 | ssn\_fulladdress\_homephone\_count\_1\_by\_30 |
| 270 | homephone\_name\_dob\_count\_1\_by\_7 | 377 | ssn\_dob\_homephone\_count\_0\_by\_3 |
| 271 | homephone\_name\_dob\_count\_1\_by\_14 | 378 | ssn\_dob\_homephone\_count\_0\_by\_7 |
| 272 | homephone\_name\_dob\_count\_1\_by\_30 | 379 | ssn\_dob\_homephone\_count\_0\_by\_14 |
| 273 | ssn\_firstname\_count\_0\_by\_3 | 380 | ssn\_dob\_homephone\_count\_0\_by\_30 |
| 274 | ssn\_firstname\_count\_0\_by\_7 | 381 | ssn\_dob\_homephone\_count\_1\_by\_3 |
| 275 | ssn\_firstname\_count\_0\_by\_14 | 382 | ssn\_dob\_homephone\_count\_1\_by\_7 |
| Day Since Candidate Variables | | | |
| 383 | ssn\_day\_since | 397 | ssn\_lastname\_day\_since |
| 384 | address\_day\_since | 398 | ssn\_address\_day\_since |
| 385 | dob\_day\_since | 399 | ssn\_zip5\_day\_since |
| 386 | homephone\_day\_since | 400 | ssn\_dob\_day\_since |
| 387 | name\_day\_since | 401 | ssn\_homephone\_day\_since |
| 388 | fulladdress\_day\_since | 402 | ssn\_name\_day\_since |
| 389 | name\_dob\_day\_since | 403 | ssn\_fulladdress\_day\_since |
| 390 | name\_fulladdress\_day\_since | 404 | ssn\_name\_dob\_day\_since |
| 391 | name\_homephone\_day\_since | 405 | ssn\_name\_fulladdress\_day\_since |
| 392 | fulladdress\_dob\_day\_since | 406 | ssn\_name\_homephone\_day\_since |
| 393 | fulladdress\_homephone\_day\_since | 407 | ssn\_fulladdress\_dob\_day\_since |
| 394 | dob\_homephone\_day\_since | 408 | ssn\_fulladdress\_homephone\_day\_since |
| 395 | homephone\_name\_dob\_day\_since | 409 | ssn\_dob\_homephone\_day\_since |
| 396 | ssn\_firstname\_day\_since | 410 | ssn\_homephone\_name\_dob\_day\_since |

Reference:

Team, V. (2021, February 25). Machine learning algorithms: What is a neural network? Retrieved March 25, 2021, from https://www.verypossible.com/insights/machine-learning-algorithms-what-is-a-neural-network

Ho, Tin Kam (1995). Random Decision Forests (PDF). Proceedings of the 3rd International Conference on Document Analysis and Recognition, Montreal, QC, 14–16 August 1995. pp. 278–282. Archived from the original (PDF) on 17 April 2016. Retrieved 5 June 2016.

https:web.archive.org/web/20160417030218/http://ect.bell-labs.com/who/tkh/publications/papers/odt.pdf

Brownlee, J. (2020, August 14). A gentle introduction to the gradient boosting algorithm for machine learning. Retrieved March 25, 2021, from https://machinelearningmastery.com/gentle-introduction-gradient-boosting-algorithm-machine-learning/

## Z\_ai (2020, February 8). Logistic Regression Explained. Retrieved March 23, 2021 from https://towardsdatascience.com/logistic-regression-explained-9ee73cede081