

**FRAUD ANALYSIS: CREDIT CARD APPLICATION**

**Authors:**

**Gyan Prakash, Yufei Wang, Wei Tang, Alok Abhishek, Weichen Zhang, Pratyush Shankar, William Staudenmeier**

**FRAUD ANALYSIS: CREDIT CARD APPLICATION DATA**

**Table of Contents**

I. Executive Summary……………………………………………………………………2

II. Data Description……………………………………………………………………….3

III. Feature Selection……………………………………………………………………..14

IV. Algorithms……………………………………………………………………………..17

V. Results……………………………………………………………………………….…20

VI. Conclusions……………………………………………………………………………21

VII. Appendix……………………………………………………………………………....22

1. **Executive Summary**

Identity fraud is one of the fastest growing types of white-collar crime in the US. It can be devastating as it can not only affect one’s ability to secure credit but also compromise financial history, and in some cases, assets. When an identity is stolen, it usually takes a fair amount of time and money, not to mention a considerable amount of effort, to make sure the fraud is contained, and the victim doesn’t suffer any more adverse consequences.

This report provides an analysis of credit card application data for detecting fraud using supervised machine learning methods. We used tools like R, Python, SQL and Microsoft Excel to measure the goodness of our models. Such measures of goodness included, but were not limited to, False Detection Rate (FDR), False Positives, Kolmogorov-Smirnov test (K-S).

The original data set has 94,866 credit card application records. Each entry has 9 variables containing applicants’ personal information. An overall view of our analysis is given below.

* Building expert variables with different time windows
* Standardization and dimensionality reduction
* Application fraud algorithm
* Calculation of measures of goodness

We created a total of 114 variables. Out of 114, we chose 21 expert variables using subset selection methods to apply our supervised learning models. We built various fraud algorithms, such as Linear Regression, Support Vector Machine, Neural Network, Boosted Tree and Bootstrap Forest, and thereby, evaluated their respective fraud detection rates.

Bootstrap Forest model gave us the best results. The model was able to detect present 17.14% of the frauds at 10% (also called, FDR at 10%) in the out-of-time validation data set. Among 15,812 records, 4,061 incidents of fraud were detected.

1. **Data Description**

Credit card application dataset contains 94,866 entries corresponding to credit card applications. 20,164 of such entries (i.e., 21.26% of the applications) have been marked as fraud. Each record includes information such as name of applicant, address, ZIP code, date of birth, home phone number, date of application and Social Security Number (SSN), as report by the applicant. All variables were masked to conceal the identities of the applicants. The dataset was, however, representative of the applicant pool.

A detailed analysis of the dataset is given below. The Data Quality Report can be found in Appendix.

**II.I Data Cleaning and Transformation**

**II.I.I Correcting DOB**

The data only includes observations from 1900 to 1999. However, since the year is denoted by only two digits (YY), it is likely our analysis tools will plug in the wrong century. So, the year 1907 could be interpreted as 2007.

Our first step was to rectify this issue and create a uniform variable that is easy to interpret. Hence, we created a new variable called *newdob*, in which year was denoted using 4 digits (YYYY).

**II.I.II Adding Leading Zeros**

The observations for variables SSN, ZIP and home phone number had inconsistent lengths. This can be attributed to the fact that computers usually remove leading zeros, and this can make data analysis difficult, especially, when the variables are not quantitative.

To rectify this issue, we added leading zeros to observations corresponding to each of the three variables.

* New variable *newssn* has consistent length of 9 for all its observations
* New variable called *ZIP5* has consistent length of 5 for all its observations
* New variable called *homephone* has consistent length of 10 for all its observations

**II.I.III Handling Frivolous**

To eliminate the effects of frivolous values, we replaced the affected records in the expert variable with the mean of all the records.

**II.I.IV Description of Important Variables**

Here is the basic information about our dataset.

|  |  |
| --- | --- |
| Dataset | Personal Identifiable Information (PII) |
| Records | 94,866 |
| Columns | 10 categorical variables |
| Time period | 01/01/2016 – 12/31/2016 |
| Resource | Simulated by Professor Stephen Coggeshall |

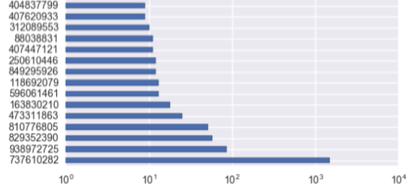
**Field Name: *ssn***

*ssn* is a categorical variable indicating the applicant’s self-reported SSN. This field is 100% populated with 86,711 unique values. An SSN shorter than 9 digits, indicates that there are leading zeros in the observations. For example, a value of ‘2503’ is actually ‘000002503’. It is obvious that the value ‘737610282’ is frivolous as its frequency is 173 times that of the second most frequently occurring SSN.

15 of the most frequently occurring *ssn* records have been listed below.

|  |  |
| --- | --- |
| ***ssn*** | **counts** |
| 737610282 | 1478 |
| 938972725 | 85 |
| 829352390 | 57 |
| 810776805 | 51 |
| 473311863 | 25 |
| 163830210 | 18 |
| 596061461 | 13 |
| 118692079 | 13 |
| 849295926 | 12 |
| 250610446 | 12 |
| 407447121 | 11 |
| 88038831 | 11 |
| 312089553 | 10 |
| 407620933 | 9 |
| 404837799 | 9 |

The distributions of the top 15 SSNs is as under.



**Field Name: *address***

*address* is a text variable that contains the reported addresses of applicants. The field is 100% populated with 88,167 unique values. The most frequent address ‘8911 MZSU DR 43516’ appeared 57 times. This address is likely to be a frivolous one.

15 of the most frequent addresses have been listed below:

|  |  |
| --- | --- |
| ***address*** | **counts** |
| 8911 MZSU DR 43516 | 57 |
| 2602 AJTJ AVE 68138 | 32 |
| 4292 RUSMM LN 77157 | 25 |
| 7651 RRMTE WY 52283 | 18 |
| 6981 RMR RD 11403 | 17 |
| 9570 UXJSS DR 11403 | 14 |
| 7136 RZUZM ST 82373 | 13 |
| 5336 RTEET WY 85658 | 13 |
| 2436 EXZJM AVE 00645 | 13 |
| 7544 XTMUA BLVD 29941 | 12 |
| 6637 UESAX LN 44371 | 12 |
| 3639 EERES CT 38406 | 12 |
| 3596 SRSU DR 34218 | 11 |
| 8 RZSMS ST 63240 | 11 |
| 8528 XJERR BLVD 95552 | 10 |

The distributions of the top 15 records is as under.



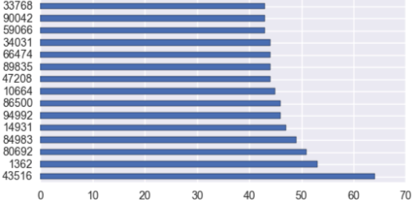
**Field Name: *zip5***

*zip5* is a categorical variable that contains ZIP codes of the applicants’ neighborhoods. The field is 100% populated with 15,855 unique values. The most frequent zip code, ‘43516’, appeared 64 times. This ZIP code is likely to be a frivolous one.

15 of the most frequently occurring values have been listed below:

|  |  |
| --- | --- |
| ***zip5*** | **counts** |
| 43516 | 64 |
| 1362 | 53 |
| 80692 | 51 |
| 84983 | 49 |
| 14931 | 47 |
| 94992 | 46 |
| 86500 | 46 |
| 10664 | 45 |
| 47208 | 44 |
| 89835 | 44 |
| 66474 | 44 |
| 34031 | 44 |
| 59066 | 43 |
| 90042 | 43 |
| 33768 | 43 |

The distributions of the top 15 most frequently occurring ZIP codes is as under.



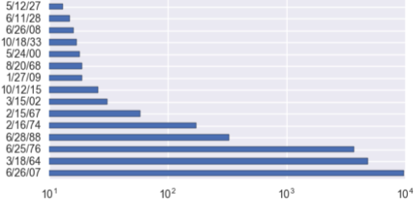
**Field Name: *dob***

*dob* is a date variable that contains applicants’ reported dates of birth. The field is 100% populated with 30,599 unique values. The most frequent date of birth ‘*6/26/07*’ appeared 9,681 times. This date of birth is likely a frivolous value.

15 of the most frequently occurring values have been listed below.

|  |  |
| --- | --- |
| ***dob*** | **counts** |
| 6/26/07 | 9681 |
| 3/18/64 | 4808 |
| 6/25/76 | 3698 |
| 6/28/88 | 330 |
| 2/16/74 | 173 |
| 2/15/67 | 59 |
| 3/15/02 | 31 |
| 10/12/15 | 26 |
| 8/20/68 | 19 |
| 1/27/09 | 19 |
| 5/24/00 | 18 |
| 10/18/33 | 17 |
| 6/26/08 | 16 |
| 6/11/28 | 15 |
| 5/12/27 | 13 |

The distributions of the 15 most frequent records are as under.



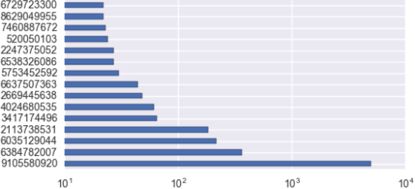
**Field Name: *homephone***

*homephone* is a categorical variable that contains applicants’ reported home phone numbers. The field is 100% populated with 22,181 unique values. The most frequent home phone number ‘*9105580920*’ appeared 7,735 times, which accounts for 7.74% of all records. This number is likely a frivolous value.

15 of the most frequently occurring values have been listed below.

|  |  |
| --- | --- |
| ***homephone*** | **counts** |
| 9105580920 | 4974 |
| 6384782007 | 364 |
| 6035129044 | 215 |
| 2113738531 | 184 |
| 3417174496 | 65 |
| 4024680535 | 61 |
| 2669445638 | 48 |
| 6637507363 | 44 |
| 5753452592 | 30 |
| 6538326086 | 27 |
| 2247375052 | 27 |
| 520050103 | 24 |
| 7460887672 | 23 |
| 8629049955 | 22 |
| 6729723300 | 22 |

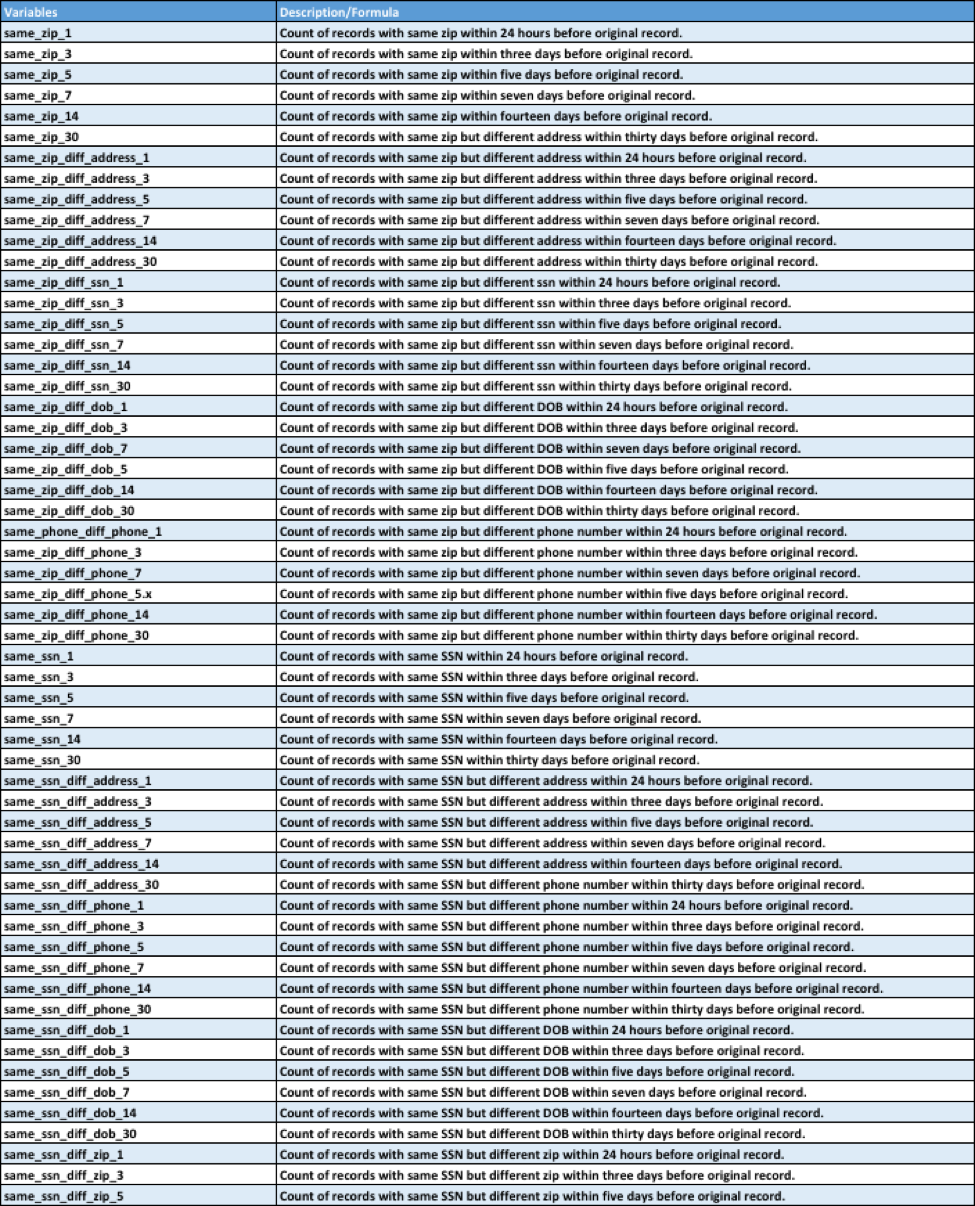
The distributions of the 15 most frequent records are as under.

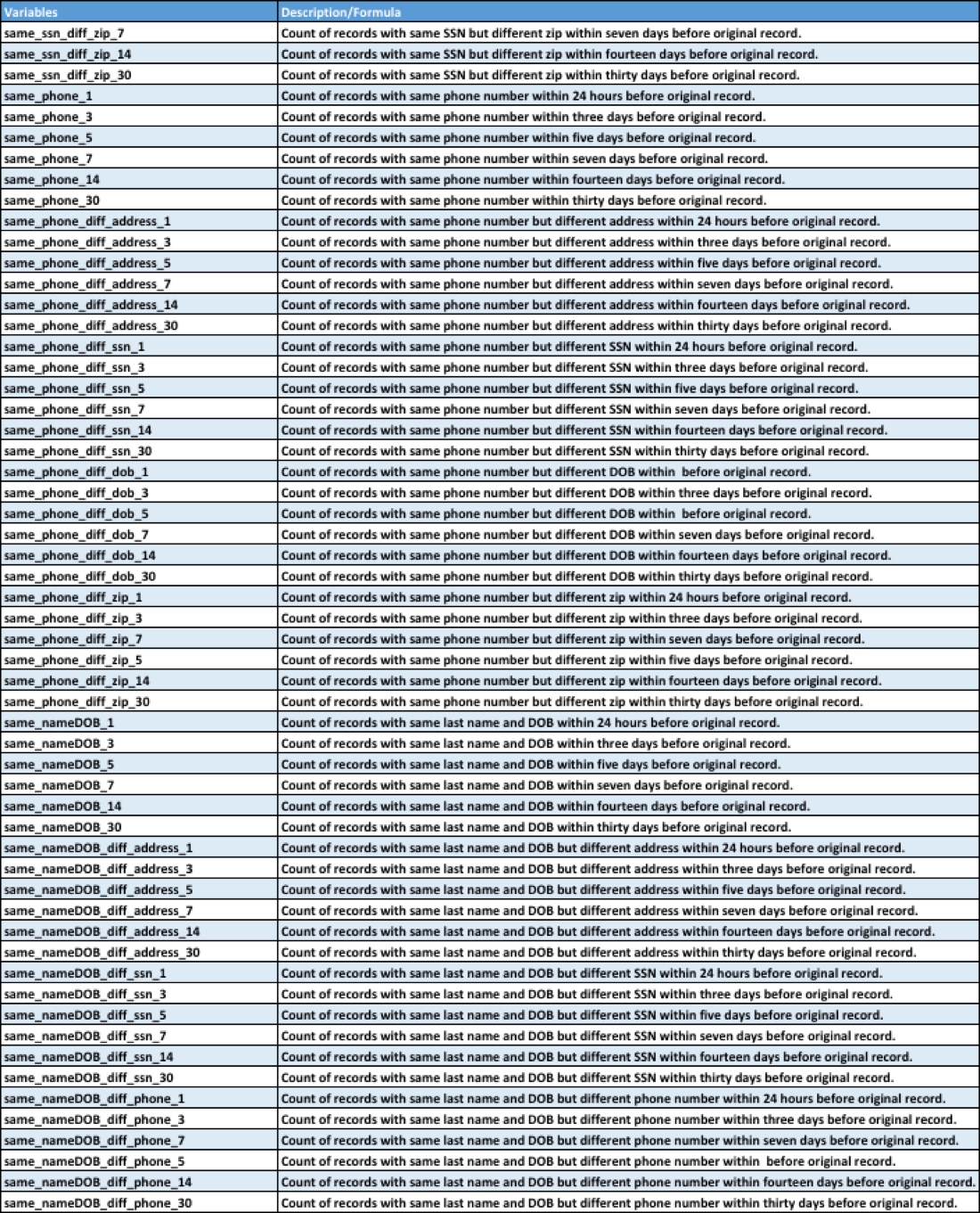


**II.II Variable Creation**

To make the analysis more in-depth and meaningful, we modified some of the existing variables before we built our expert variables. To create the expert variables, we chose several different time windows corresponding to 1, 3, 5, 7, 14 and 30 days. We chose unique categorical values to compare linkages across the time intervals corresponding to date of birth, SSN, home phone number, ZIP and last name. We ensured that the records took place before the original record by adding a conditional statement to our code specifying that the records being counted had to be less than the original record from which the linkages were drawn. The rationale was to train our models to identify fraudulent applications based on frequencies of certain fields over different time frames.

Using SQL Workbench, we created two identical databases from our original dataset. For every record in the first database, we counted the number of times an earlier record within the given time interval had the same categorical value. In this manner, we created 114 unique expert variables as shown in the table below.



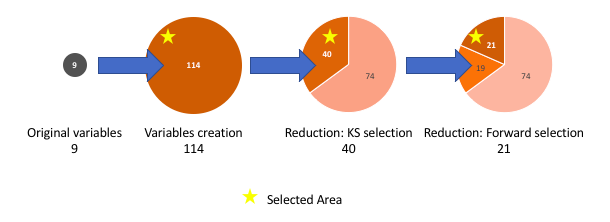


1. **Feature Selection**

Feature selection has multiplicative effects on the overall modeling process.

* **Reduces overfitting**: Less redundant data means less opportunity to make decisions based on noise
* **Improves accuracy**: Less misleading data means modeling accuracy improves
* **Reduces training time**: Less data means that algorithms train faster

We used filter feature selection (Kolmogorov–Smirnov test) and wrapper feature selection (forward selection) to identify the most valuable variables for our modeling. We selected 40 variables after Kolmogorov–Smirnov and from that, we selected 21 variables by forward selection. An illustration shown below summarizes the feature selection process.

****

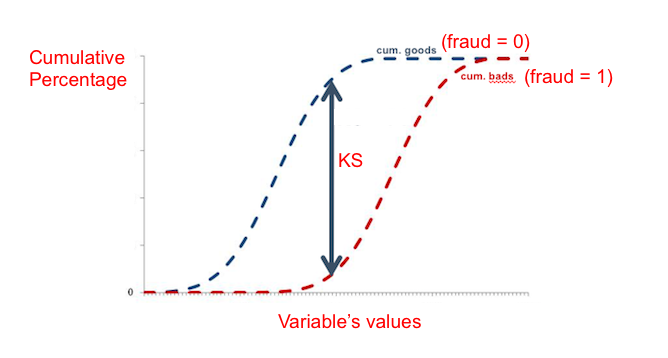
**III.I Filter Feature Selection**

Kolmogorov–Smirnov score can be used in filter feature selection to measure the ability of one variable to distinguish classes. In [statistics](https://en.wikipedia.org/wiki/Statistics), the Kolmogorov–Smirnov test (K–S test) is a [nonparametric test](https://en.wikipedia.org/wiki/Nonparametric_statistics) of the equality of continuous, one-dimensional [probability distributions](https://en.wikipedia.org/wiki/Probability_distribution) that can be used to compare a [sample](https://en.wikipedia.org/wiki/Random_sample) with a reference probability distribution (one-sample K–S test), or to compare two samples (two-sample K–S test).

We implemented Kolmogorov–Smirnov using Python in the following steps:

* For each variable, we calculated the cumulative percentage on every variable's class i.e., good (*fraud* = 0) and bad (*fraud* = 1) separately, so we get two cumulative distributions.
* For each variable, we used *stats.ks\_2samp()* function from *spicy* package in Python to calculate the K-S between distributions of good (*fraud* = 0) and bad (*fraud* = 1) records.
* Finally, we sorted variables by decreasing K-S and chose the first 40 variables for the next step.

An illustration of K-S is as under.

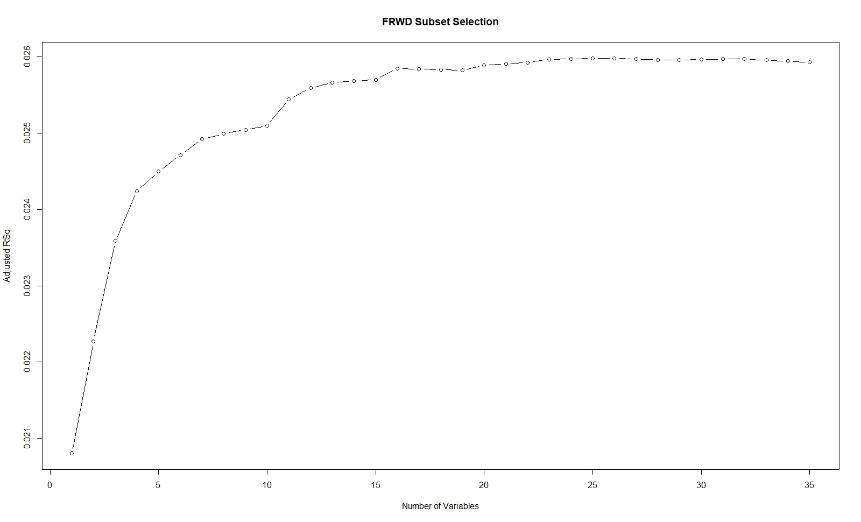


**III.II Wrapper Feature Selection**

Forward selection is one of the wrapper feature selection methods. It may be noted here that filter methods pick up the intrinsic properties of the features (i.e., the relevance of the features) measured via univariate statistics instead of cross-validation performance, whereas wrapper methods essentially optimize the classifier performance. Wrapper methods are computationally more expensive compared to filter methods due to the repeated learning steps and cross-validation.

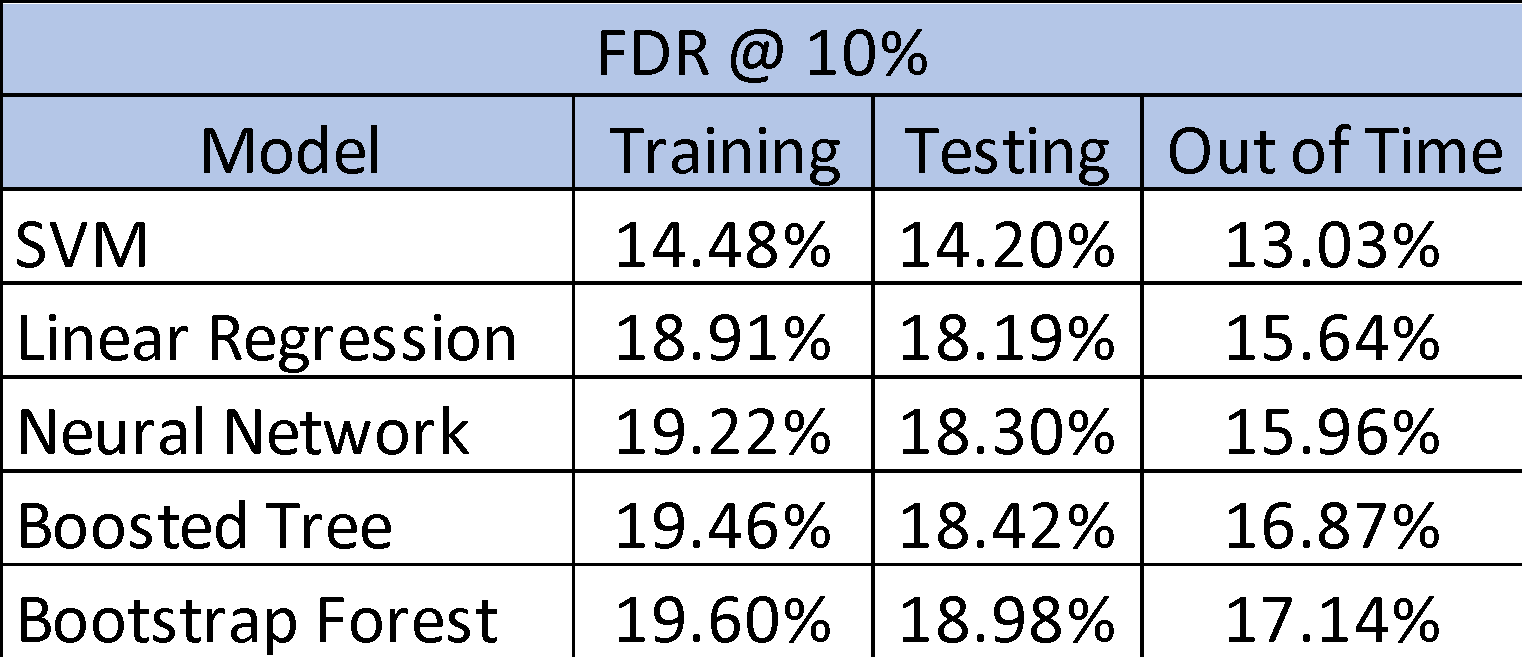
The simplest data-driven model building approach is called forward selection. In this approach, one adds variables to the model one at a time. At each step, each variable that is not already in the model is tested for inclusion in the model. And then, we use adjusted R2 to judge goodness of model with different number of variables. Based on forward selection, we found that model performance is optimal when the number of variables is 21. These variables were selected for model building.

A plot of number of variables versus adjusted R2 (forward subset selection) has been given below.



1. **Algorithms**

After creating expert variables, we did standardization and dimensionality reduction. Subsequently, we calculated the fraud detection rate at 10% for various supervised learning algorithms. The results have been summarized in the following table.

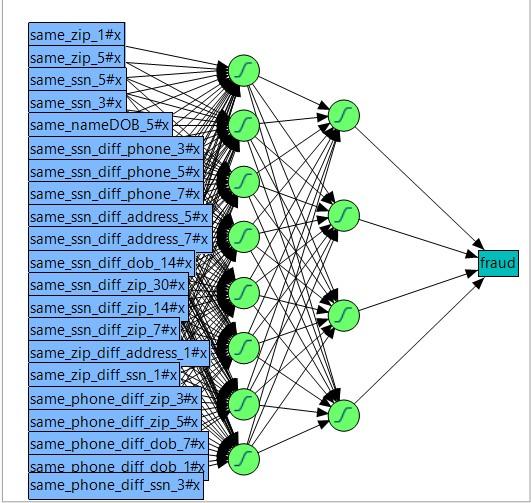


**IV.I Description of Algorithms**

**i. Support Vector Machine:** It’s a supervised learning model with associated learning algorithms that analyze data used for classification and regression analysis. Given a set of training examples, each marked as belonging to one or the other of two categories, an SVM training algorithm builds a model that assigns new examples to one category or the other, making it a non-probabilistic binary linear classifier.

**ii. Linear Regression:** Linear regression is a statistical modeling technique used to describe a continuous response variable as a linear function of one or more predictor variables.

**iii. Neural Network:** It is a technique in which while learning, one of the input patterns is given to the net's input layer. This pattern is propagated through the net (independent of its structure) to the net's output layer. The output layer generates an output pattern which is then compared to the target pattern. A depiction of our neural network model has been shown below.



**iv. Boosted Tree:** It is a technique for regression and classification problems, which produces a prediction model in the form of an ensemble of weak prediction models, typically decision trees.

**v. Bootstrap Forest:** It is an ensemble learning method for classification, regression and other tasks, that operate by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification), or mean prediction (regression) of the individual trees.

**IV.II Measures of Goodness for Fraud Models**

**Kolmogorov–Smirnov Test (K-S test):** Itis a nonparametric test of the equality of continuous, one-dimensional probability distributions that can be used to compare a sample with a reference probability distribution (one-sample K–S test), or to compare two samples (two-sample K–S test).

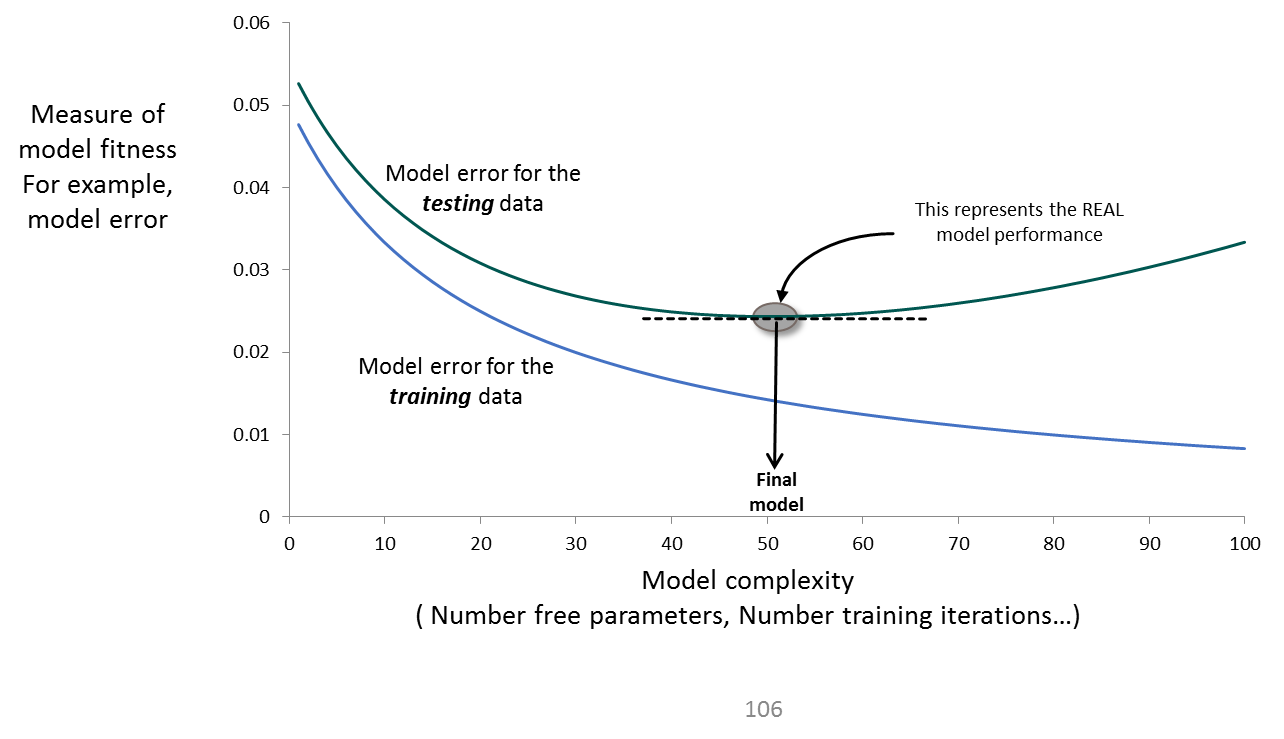
**Fraud Detection Rate (FDR):** It is measure of goodness wherein we evaluate what percentage of all the frauds that are caught at a cutoff location. For instance, an FDR of 50% at 10% means the model catches 50% of all the frauds in 10% of the population.

**False Positive:** A false positive error, or in short, a false positive, commonly called a "false alarm", is a result that indicates a given condition exists when it does not.

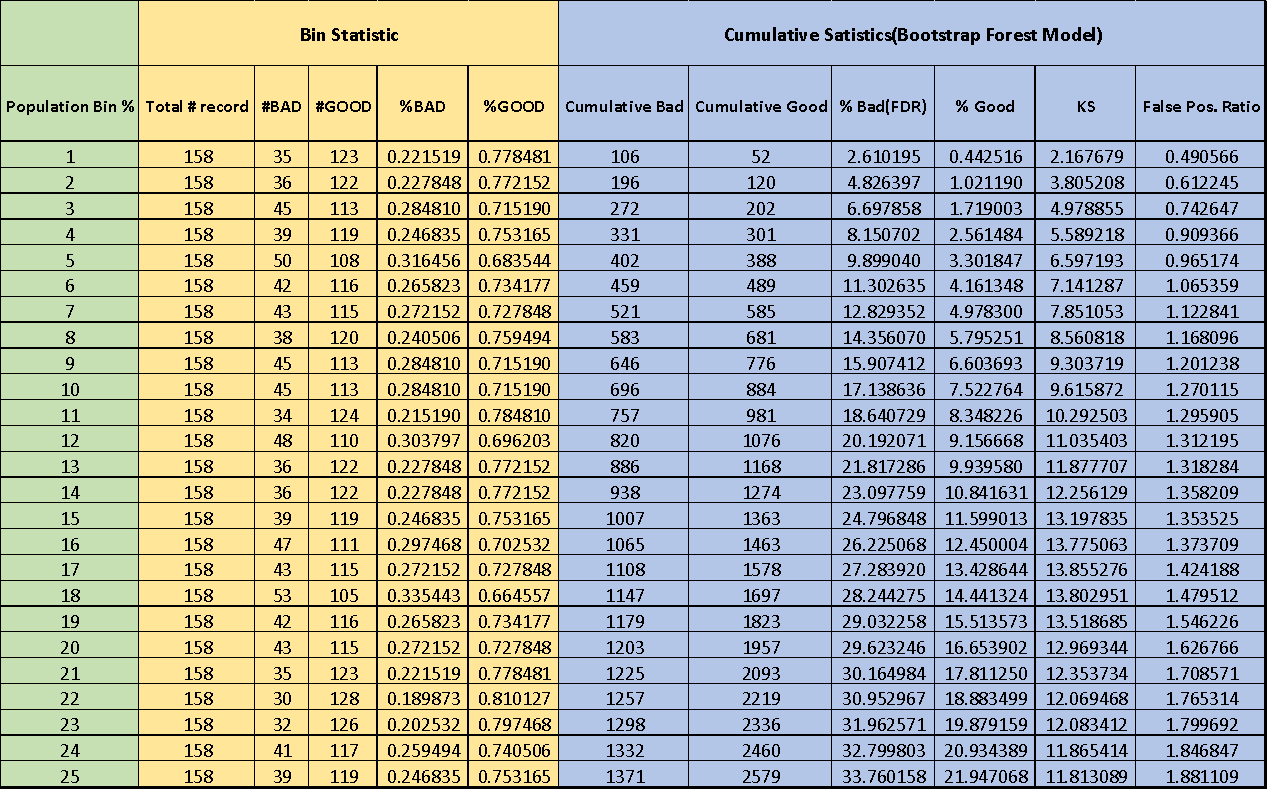
**IV.III Out of Time Model Validation**

We separated the data into multiple sets to ensure that our models are robust. We build the model on the training data (equivalent to 6 months’ worth of data), then evaluated it on the testing data (equivalent to 4 months’ worth of data). In addition to that, we also reserved a set of data (corresponding to applications during Nov-Dec, 2016) that was never used during training. In other words, it was a data set that our models had never seen before. Validation on this holdout sample is called out-of-time validation.

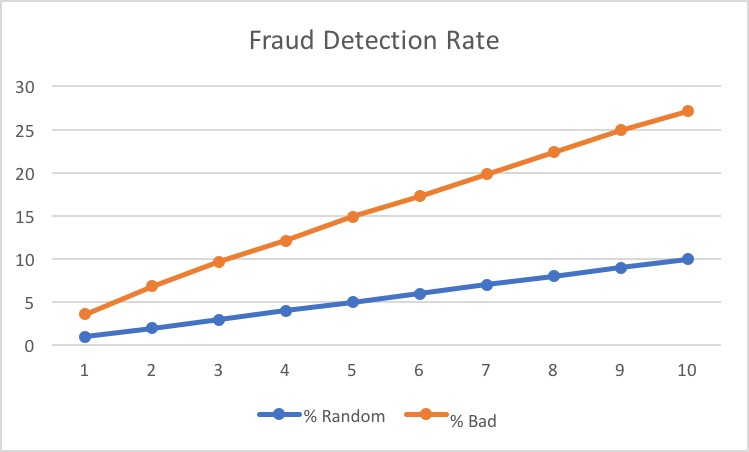
An illustration about optimal model performance has been given below. Out-of-time validation goes a step further and tests the robustness of the model once again.



1. **Results**

In this section we provide an overview of our results. We plot the score distributions as well as create a table of the top 25% bins.

Among 15,812 records in our out-of-time holdout sample, our best model, Bootstrap Forest, detected 11,751 good ones (*fraud* = 0) and 4,061 bad ones (*fraud* = 1).



1. **Conclusions**

In this project we started with exploratory analysis of the data which included descriptive analysis and visualization. Next, we cleaned and transformed the original credit card application dataset and created new variables so that we could create various supervised learning algorithms with the aim of detecting fraudulent applications.

After evaluating various algorithms, like Support Vector Machine, Neural Network, Boosted Tree, Linear Regression and Bootstrap Forest, we found that **Bootstrap Forest algorithm gave the best results with an FDR of 17.14% at 10%**, which is to say that the model catches 17.14% of all frauds in 10% of the population.

**VI.I Scope for Improvement**

There are several things that can be done to improve our models. Some have been listed below.

**i. Domain Expertise:** The final model can be improved by inputs from experts. We can think of creating better variables to better capture information and improve the accuracy of our models.

**ii. Augmenting Data:** If we can get more information about credit card applications it may be useful in improving the model further.

**iii. Comparison with Historic Data:** It’s a good idea to look at how fraud detection is applied in other areas and see what new ideas can be incorporated in our models. This might help in uncovering some new ideas and attributes which can further improve the model.

1. **Appendix**

**VII.I Data Quality Report**

**BASIC INFORMATION**

|  |  |
| --- | --- |
| **Dataset** | Personal Identifiable Information (PII) |
| **Records** | 94,866 |
| **Columns** | 10 categorical variables |
| **Time period** | 01/01/2016 – 12/31/2016 |
| **Resource** | Simulated by Professor Stephen Coggeshall |

**SUMMARY TABLE**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Type** | **Variables** | **# of Unique values** | **Count** | **Percentage populated** |
| Categorical | record | 94,899 | 94,866 | 100% |
| date | 365 | 94,866 | 100% |
| ssn | 86,771 | 94,866 | 100% |
| firstname | 14,626 | 94,866 | 100% |
| lastname | 31,513 | 94,866 | 100% |
| address | 88,167 | 94,866 | 100% |
| zip5 | 15,855 | 94,866 | 100% |
| dob | 30,599 | 94,866 | 100% |
| homephone | 20,762 | 94,866 | 100% |
| fraud | 2 | 94,866 | 100% |

**DATA ANALYSIS**

***record*** – order number of each record

Number of values: 94,866

Number of unique values: 94,899

Distribution: Starts from 1, increasing by 1 each time, with no repetition.

***date*** – date of generating records

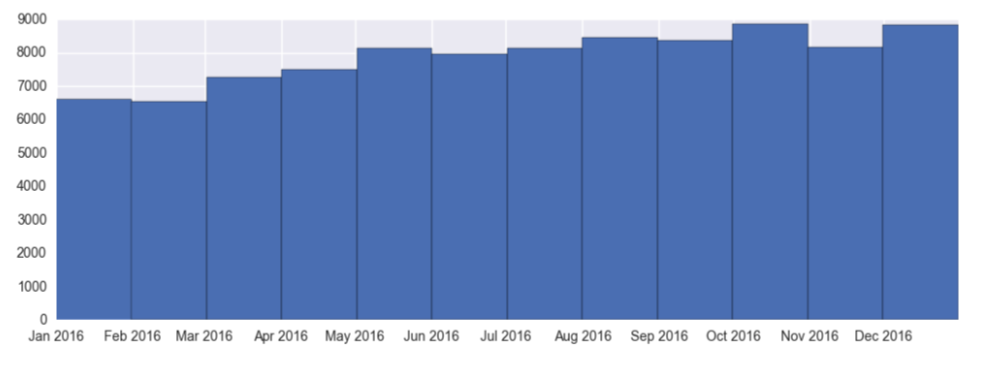
Number of values: 94,866

Number of unique values: 356

Top 15 values:

|  |  |  |
| --- | --- | --- |
| **rank** | **date** | **counts** |
| 1 | 06/09/16 | 329 |
| 2 | 12/29/16 | 328 |
| 3 | 11/19/16 | 325 |
| 4 | 09/18/16 | 324 |
| 5 | 10/18/16 | 324 |
| 6 | 10/02/16 | 320 |
| 7 | 12/10/16 | 320 |
| 8 | 12/08/16 | 320 |
| 9 | 10/07/16 | 320 |
| 10 | 12/30/16 | 319 |
| 11 | 08/27/16 | 315 |
| 12 | 12/31/16 | 307 |
| 13 | 09/25/16 | 306 |
| 14 | 10/21/16 | 305 |
| 15 | 09/15/16 | 305 |

Monthly distribution:



***ssn*** – social security number of each record

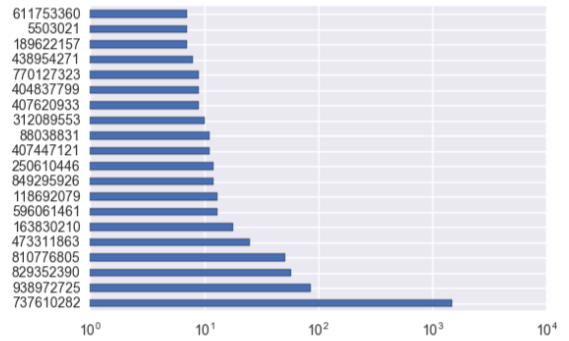
Number of values: 94,866

Number of unique values: 86,771

Top 15 values:

|  |  |  |
| --- | --- | --- |
| **rank** | **ssn** | **counts** |
| 1 | 737610282 | 1478 |
| 2 | 938972725 | 85 |
| 3 | 829352390 | 57 |
| 4 | 810776805 | 51 |
| 5 | 473311863 | 25 |
| 6 | 163830210 | 18 |
| 7 | 596061461 | 13 |
| 8 | 118692079 | 13 |
| 9 | 849295926 | 12 |
| 10 | 250610446 | 12 |
| 11 | 407447121 | 11 |
| 12 | 88038831 | 11 |
| 13 | 312089553 | 10 |
| 14 | 407620933 | 9 |
| 15 | 404837799 | 9 |

Log scale bar chart of top 15 values:



***firstname*** – first name of each record

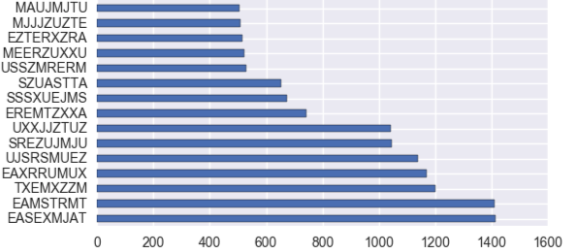
Number of values: 94,866

Number of unique values: 14,626

Top 15 values:

|  |  |  |
| --- | --- | --- |
| **rank** | **firstname** | **counts** |
| 1 | EASEXMJAT | 1414 |
| 2 | EAMSTRMT | 1411 |
| 3 | TXEMXZZM | 1200 |
| 4 | EAXRRUMUX | 1170 |
| 5 | UJSRSMUEZ | 1138 |
| 6 | SREZUJMJU | 1044 |
| 7 | UXXJJZTUZ | 1042 |
| 8 | EREMTZXXA | 742 |
| 9 | SSSXUEJMS | 675 |
| 10 | SZUASTTA | 653 |
| 11 | USSZMRERM | 529 |
| 12 | MEERZUXXU | 523 |
| 13 | EZTERXZRA | 516 |
| 14 | MJJJZUZTE | 510 |
| 15 | MAUJMJTU | 504 |

Bar chart of top 15 values:



***lastname*** - last name of each record

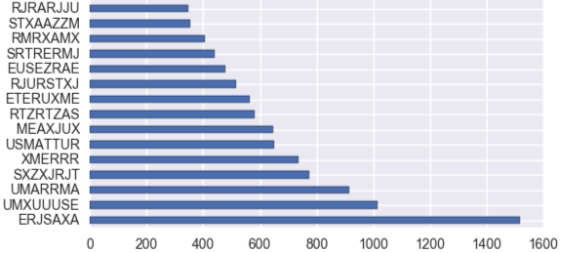
Number of values: 94,866

Number of unique values: 31,513

Top 15 values:

|  |  |  |
| --- | --- | --- |
| **rank** | **lastname** | **counts** |
| 1 | ERJSAXA | 1515 |
| 2 | UMXUUUSE | 1013 |
| 3 | UMARRMA | 913 |
| 4 | SXZXJRJT | 775 |
| 5 | XMERRR | 737 |
| 6 | USMATTUR | 649 |
| 7 | MEAXJUX | 645 |
| 8 | RTZRTZAS | 582 |
| 9 | ETERUXME | 562 |
| 10 | RJURSTXJ | 515 |
| 11 | EUSEZRAE | 476 |
| 12 | SRTRERMJ | 438 |
| 13 | RMRXAMX | 405 |
| 14 | STXAAZZM | 352 |
| 15 | RJRARJJU | 348 |

Bar chart of top 15 values:



***address*** - address of each record

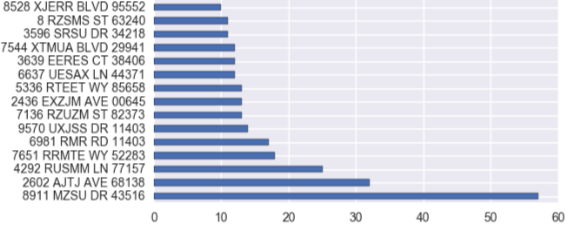
Number of values: 94,866

Number of unique values: 88,167

Top15 values:

|  |  |  |
| --- | --- | --- |
| **rank** | **address** | **counts** |
| 1 | 8911 MZSU DR 43516 | 57 |
| 2 | 2602 AJTJ AVE 68138 | 32 |
| 3 | 4292 RUSMM LN 77157 | 25 |
| 4 | 7651 RRMTE WY 52283 | 18 |
| 5 | 6981 RMR RD 11403 | 17 |
| 6 | 9570 UXJSS DR 11403 | 14 |
| 7 | 7136 RZUZM ST 82373 | 13 |
| 8 | 5336 RTEET WY 85658 | 13 |
| 9 | 2436 EXZJM AVE 00645 | 13 |
| 10 | 7544 XTMUA BLVD 29941 | 12 |
| 11 | 6637 UESAX LN 44371 | 12 |
| 12 | 3639 EERES CT 38406 | 12 |
| 13 | 3596 SRSU DR 34218 | 11 |
| 14 | 8 RZSMS ST 63240 | 11 |
| 15 | 8528 XJERR BLVD 95552 | 10 |

Bar chart of TOP 15 values:



***zip5*** – zip code of each record

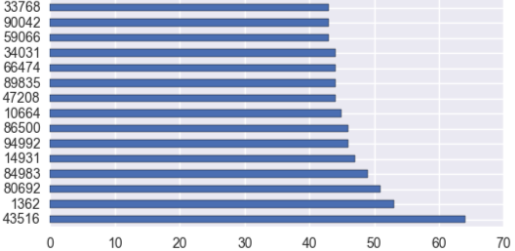
Number of values: 94,866

Number of unique values: 15,855

Top15 values:

|  |  |  |
| --- | --- | --- |
| **rank** | **zip5** | **counts** |
| 1 | 43516 | 64 |
| 2 | 1362 | 53 |
| 3 | 80692 | 51 |
| 4 | 84983 | 49 |
| 5 | 14931 | 47 |
| 6 | 94992 | 46 |
| 7 | 86500 | 46 |
| 8 | 10664 | 45 |
| 9 | 47208 | 44 |
| 10 | 89835 | 44 |
| 11 | 66474 | 44 |
| 12 | 34031 | 44 |
| 13 | 59066 | 43 |
| 14 | 90042 | 43 |
| 15 | 33768 | 43 |

Bar chart of TOP 15 values:



***dob*** – date of birth of each record

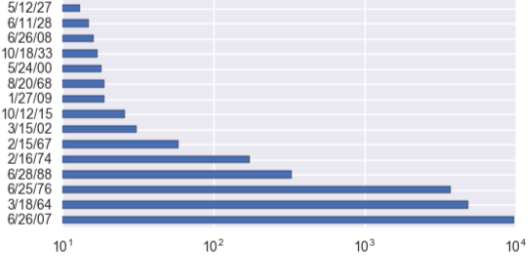
Number of values: 94,866

Number of unique values: 30,599

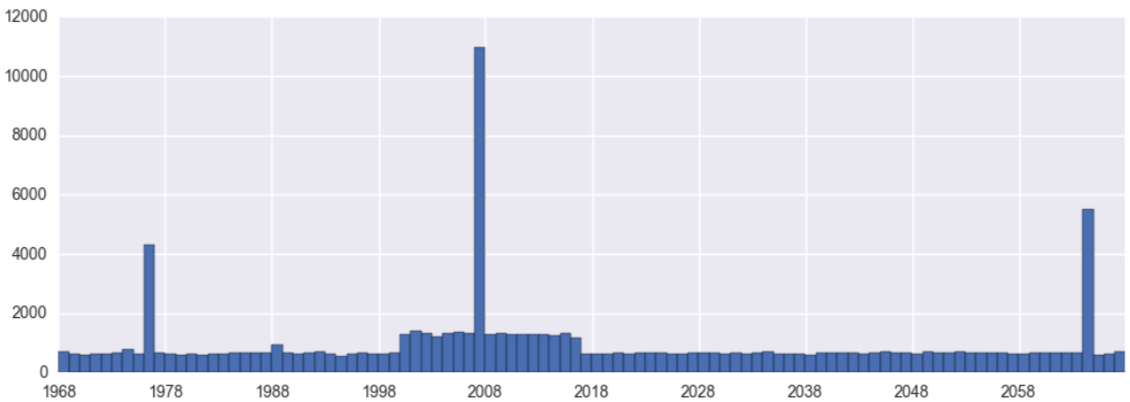
Top 15 values:

|  |  |  |
| --- | --- | --- |
| **rank** | **dob** | **counts** |
| 1 | 6/26/07 | 9681 |
| 2 | 3/18/64 | 4808 |
| 3 | 6/25/76 | 3698 |
| 4 | 6/28/88 | 330 |
| 5 | 2/16/74 | 173 |
| 6 | 2/15/67 | 59 |
| 7 | 3/15/02 | 31 |
| 8 | 10/12/15 | 26 |
| 9 | 8/20/68 | 19 |
| 10 | 1/27/09 | 19 |
| 11 | 5/24/00 | 18 |
| 12 | 10/18/33 | 17 |
| 13 | 6/26/08 | 16 |
| 14 | 6/11/28 | 15 |
| 15 | 5/12/27 | 13 |

Bar chart of top 15 values:



Yearly distribution:



***homephone*** – home phone number of each record

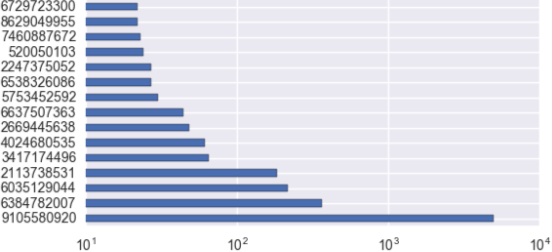
Number of values: 94,866

Number of unique values: 20,762

Top 15 values:

|  |  |  |
| --- | --- | --- |
| **rank** | **homephone** | **counts** |
| 1 | 9105580920 | 4974 |
| 2 | 6384782007 | 364 |
| 3 | 6035129044 | 215 |
| 4 | 2113738531 | 184 |
| 5 | 3417174496 | 65 |
| 6 | 4024680535 | 61 |
| 7 | 2669445638 | 48 |
| 8 | 6637507363 | 44 |
| 9 | 5753452592 | 30 |
| 10 | 6538326086 | 27 |
| 11 | 2247375052 | 27 |
| 12 | 520050103 | 24 |
| 13 | 7460887672 | 23 |
| 14 | 8629049955 | 22 |
| 15 | 6729723300 | 22 |

Bar chart of TOP 15 values:



***fraud*** – judgement of each record, whether it is fraud or not

Number of values: 94,866

Number of unique values: 2

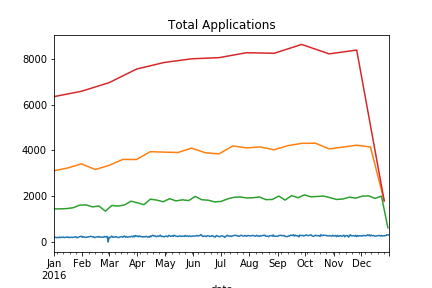
Values distribution:

|  |  |  |  |
| --- | --- | --- | --- |
| **Rank** | **fraud** | **counts** | **percentage** |
| 1 | 0 | 70702 | 78.7% |
| 2 | 1 | 20164 | 21.3% |

1 means fraud, so 21.3% of records are fraud.

**FURTHER ANALYSIS**

**Number of transactions monthly versus semimonthly versus weekly versus daily**



monthly

semimonthly

weekly

daily