Application (Identity) Fraud Analysis

Team 2
Christina Acojedo
Mrinal Gupta
Athanasios Rompokos
Guyane Valian
Min Zhu

Project 2 Report
DSO562 | Spring 2020
March 15, 2020

Table of Contents

Executive Summary

The provided dataset contained application (identity) fraud cases. It was a supervised problem as the data included a column showing the application's fraud label (whether an application was fraudulent or not). It also contained several identifying data fields about the applicant such as SSN, address, phone number, etc. The dataset had 1,000,000 records and 10 data fields. We first described and visualized each of the 10 data fields and treated all frivolous values. Then we created 634 candidate variables and performed feature selection to reduce them to 30. Finally, we used a few different modeling algorithms (both linear and nonlinear) to predict fraudulent applications records.

Each of the 10 data fields in the dataset were analyzed and all data fields were found to be categorical variables. Several values were then calculated for each data field: the number of populated values, percent populated, number of unique values, the number of records equal to zero, and the most common value. We also provided a histogram or table for each data field to get a better feel for its distribution. Many of the variables had a right skewed distribution.

The dataset contained no missing values, but there were frivolous values in the ssn, address, homephone, and dob fields. To treat these frivolous values, we replaced each unimportant value with its record number. Since the record number was just a numbering system for the records in the dataset, its unique values helped to eliminate possible false relationships between our dependent and independent variables in our algorithms.

Once the dataset was cleaned, we attempted to create as many candidate variables as possible. We concatenated variables to create different variable combinations. Then for each combination, we calculated three more variable groups: the velocity candidate variables, the days-since candidate variables, and the relative velocity candidate variables.

After the variable creation process, feature selection was performed using filter and wrapper methods to select the best variables. For our filter methods, we used Kolmogorov-Smirnov (KS) and Fraud Detection Rate (FDR) at 3% to eliminate variables. For our wrapper method, we used recursive feature elimination with cross-validation. Once the final variables were chosen, the data was divided in three sections: training, testing, and out-of-time. The out-of-time section represented the last two months of data. In addition, the first two weeks of the data was omitted to get the most accurate results possible as there was little to no data prior to each datapoint.

To find the best model, the variables were tested in several different models. First, since we had a supervised binary classification problem, logistic regression was used as a base model to predict fraud. Logistic regression caught 52.6% of fraud at a 3% FDR. Next, nonlinear models such as random forest, boosted trees, and neural network were used to get better results than the base model. Parameter tuning was also performed on each model to get better results. Overall, our best model was random forest trees which caught 57.5% of the fraudulent records in the testing set and 54.8 % of the fraudulent records in the validation set at a 3% FDR after parameter tuning.

Our future work would include consulting with the subject matter experts further in the candidate variable and feature selection phases to highlight important model variables. Also, we would use more than one method during the wrapper and model building phases to further reduce model complexity. Lastly, we would run ensemble or stacking models for comparative purposes.

Description of Data

The analysis within this report is from a synthetic dataset originally created for academic organizations that were conducting research in collaboration with ID Analytics (https://www.idanalytics.com/). The dataset is of product application data (e.g. credit card or cell phone application data) that reflects the statistical qualities and characteristics of true application data. The distribution of the data fields and the linkage properties in the dataset are therefore representative of realistic US product application data.

The dataset contains a total of 1,000,000 application records and 10 data fields covering the 2016 calendar year (January 1, 2016 to December 31, 2016). It is important to note that although the 2016 calendar year was a leap year, there are no records for February 29, 2016. The lack of records for February 29, 2016 had no bearing on the analysis since the synthetic dataset was created with a typical calendar year in mind. Another important characteristic about the dataset is that each record is labeled (i.e. classified) with a binary value of either a 1 or 0 in the "fraud_label" data field. A record containing a label of 1 was considered a fraudulent application record, and a record with a label of 0 was considered a normal application record. There is a total of 14,393 records with a label of 1 and 985,607 records with a label of 0. The labeled records therefore enable the dataset to lend itself well to binary classification analysis. A summary table of all the data fields is provided in Figure 1.

Data Field	Num Records w/ a Value	Percent Populated	Num Unique Values	Most Common Value
record	1,000,000	100%	1,000,000	Not applicable
date	1,000,000	100%	365	20160816
ssn	1,000,000	100%	835,819	999999999
firstname	1,000,000	100%	78,136	EAMSTRMT
lastname	1,000,000	100%	177,001	ERJSAXA
address	1,000,000	100%	828,774	123 MAIN ST
zip5	1,000,000	100%	26,370	68138
dob	1,000,000	100%	42,673	19070626
homephone	1,000,000	100%	28,244	9999999999
fraud_label	1,000,000	100%	2	0

Figure 1. Data Field Summary Table.

Amongst the 10 data fields, we determined the most critical raw data fields in detecting potential fraud cases were those relating to the application record's "date" data field, "ssn" data field, "address" data field, and "dob" data field. Some of the relevant depictions amongst the critical data fields are provided below in the order in which they appear. For a full description of all the data fields in the dataset, please see Appendix A for the Data Quality Report (DQR).

Data Field: "date"

A data field containing the date of the application with a format of YYYYMMDD. There are 365 unique values for this data field. As previously discussed, despite the 2016 calendar year being a leap year, there are no records for February 29, 2016. Figure 2 below provides a quick overview of the top 10 days with the highest number of applications. Some of those days happen to coincide with US holidays or significant events. However, the variation is minor for these top 10 days when looking at the total number of applications.

Date	Number of Applications	Notes on US Holiday / Event
August 16, 2016	2,877	Back-to-School sales timeframe
March 4, 2016	2,861	
July 18, 2016	2,849	
January 1, 2016	2,848	New Year's Day
August 8, 2016	2,840	Back-to-School sales timeframe
December 28, 2016	2,832	Christmas and New Year's holidays
September 3, 2016	2,832	Labor Day Weekend
June 9, 2016	2,831	Around end of school year
October 6, 2016	2,831	
March 7, 2016	2,831	

Figure 2. Top 10 days with the Highest Number of Applications.

In addition to Figure 2, the graphs depicted below show the daily and weekly application trends over the 2016 calendar year. The graphs contain the normalized trend of daily and weekly applications where the applications are depicted based on their "fraud_label" data field values of "1" (red) or "0" (green).

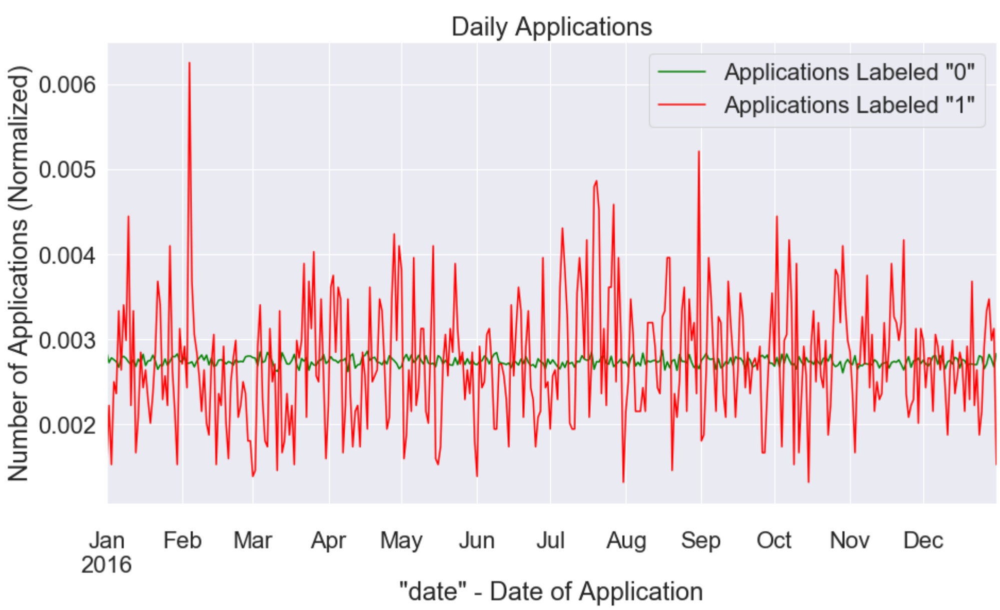


Figure 3: Normalized Trend of Daily Applications.

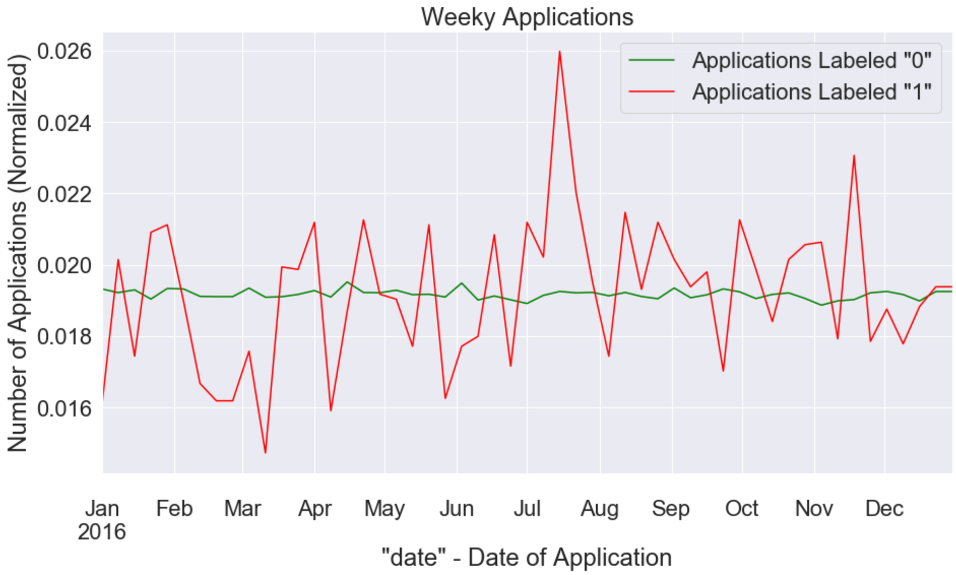


Figure 4: Normalized Trend of Weekly Applications.

Data Field: “ssn”

A 9-digit categorical data field containing the social security number (SSN) of the applicant. In absence of an SSN, a series of 9’s was entered as the applicant’s SSN. Also, for SSN entries that have less than 9 digits, those entries have a leading zero(s). The bar charts below show the top 20 “ssn” data field values. There are 16,935 records with a value of “999999999” in this data field. Since there are so many records with a “999999999” value, we show the first bar chart with the “999999999” value and a second bar chart without the “999999999” value.

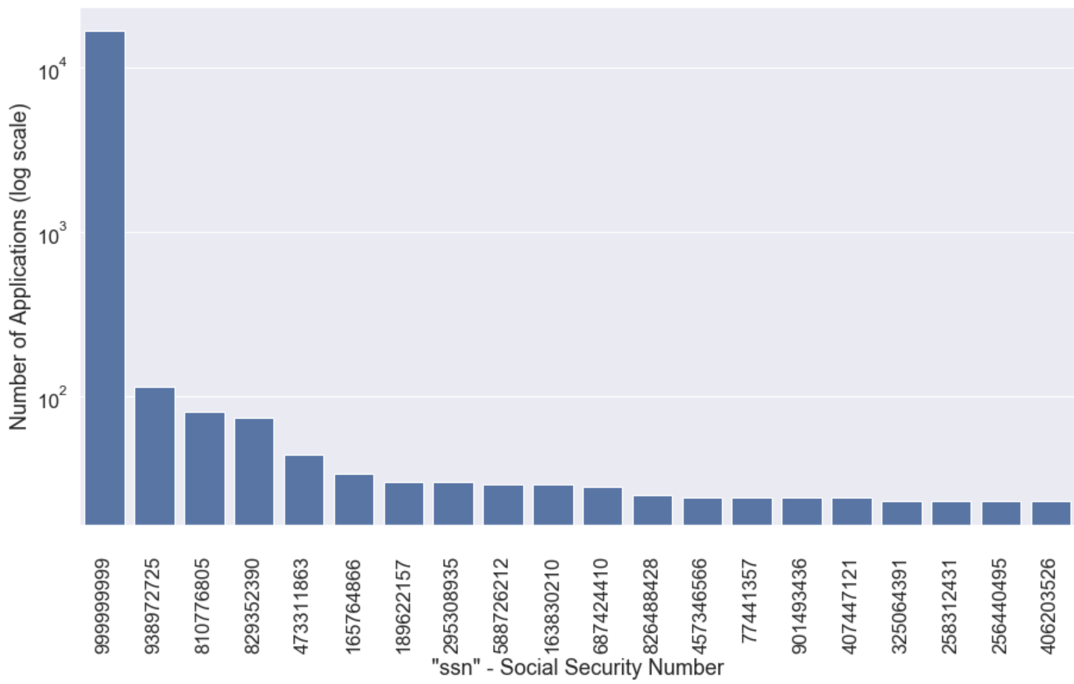


Figure 5: Top 20 SSN Frequencies Including Frivolous Value.

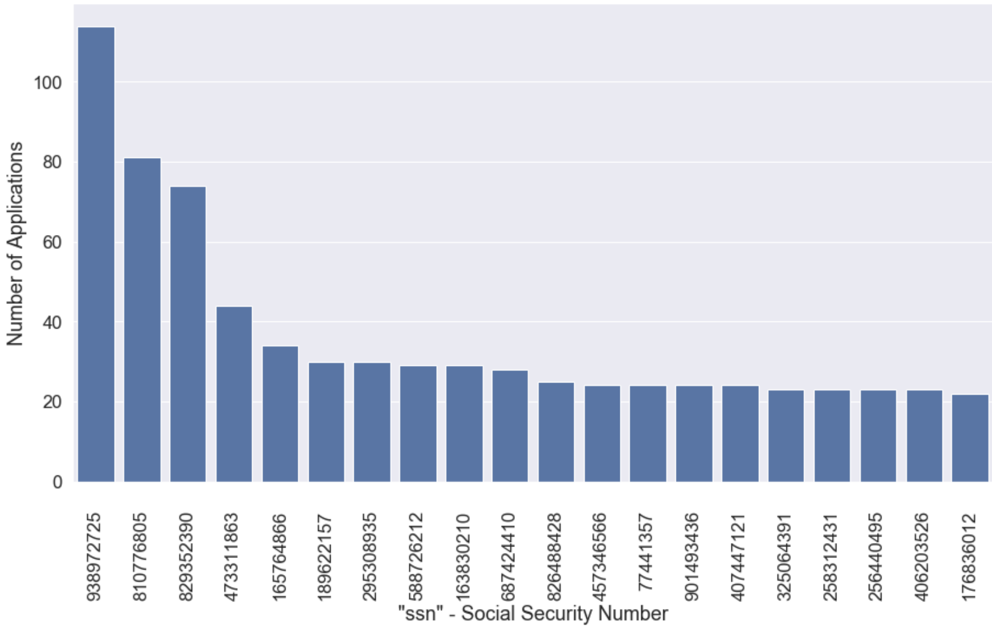


Figure 6: Top 20 SSN Frequencies Excluding Frivolous Value.

Data Field: "address"

A categorical data field containing the applicant's address. There are 1,079 records with an entry of "123 MAIN ST" as the address for an applicant. The bar charts below show the top 20 address values. Since there are so many records with "123 MAIN ST" as the address, we show the first bar chart with the "123 MAIN ST" address and the second bar chart without the "123 MAIN ST" address.

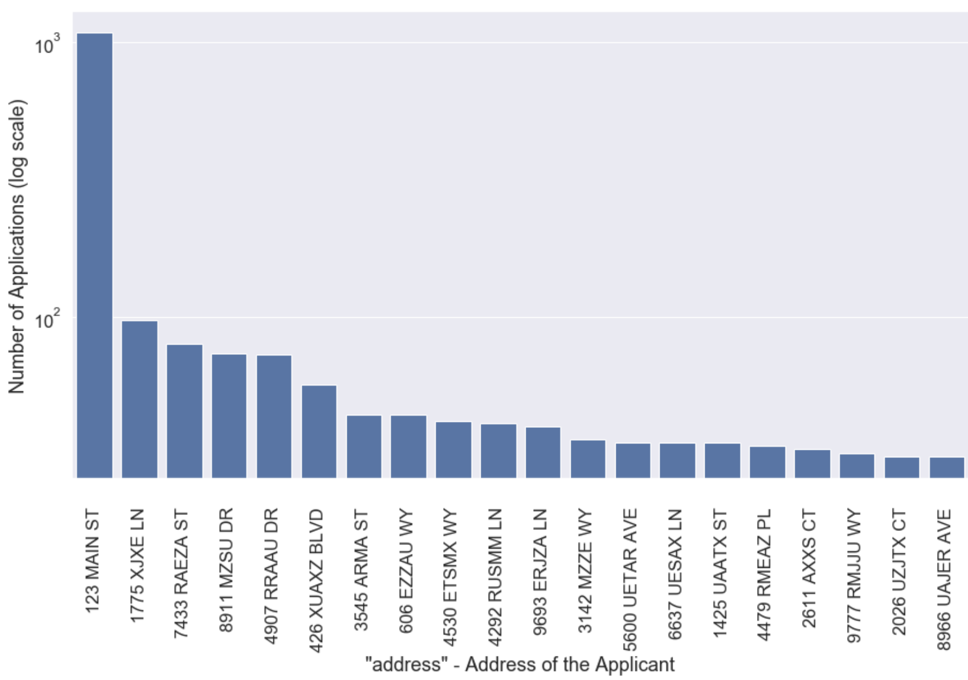


Figure 7: Top 20 Address Frequencies Including Frivolous Value.

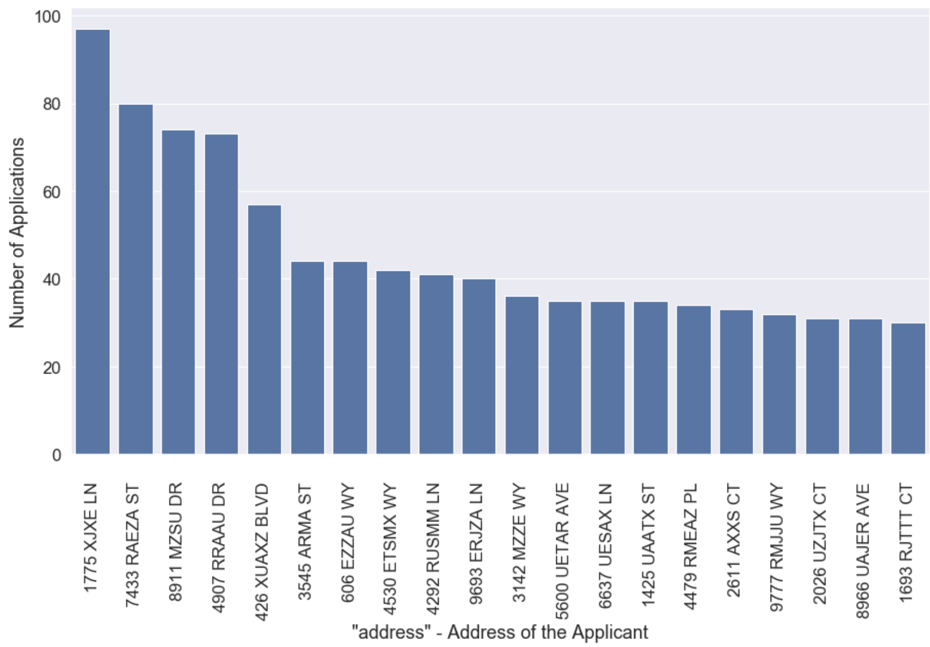


Figure 8: Top 20 Address Frequencies Excluding Frivolous Value.

Data Field: "dob"

An 8-digit categorical data field for the applicant's date of birth with a format of YYYYMMDD. There are 126,568 records with an entry of "19070626" (June 26, 1907) as the applicant's date of birth. The bar charts below show the top 20 "dob" data field values. Since there are so many records with "19070626" as the date of birth, we show the first bar chart with the "19070626" value and the second bar chart without the "19070626" value.

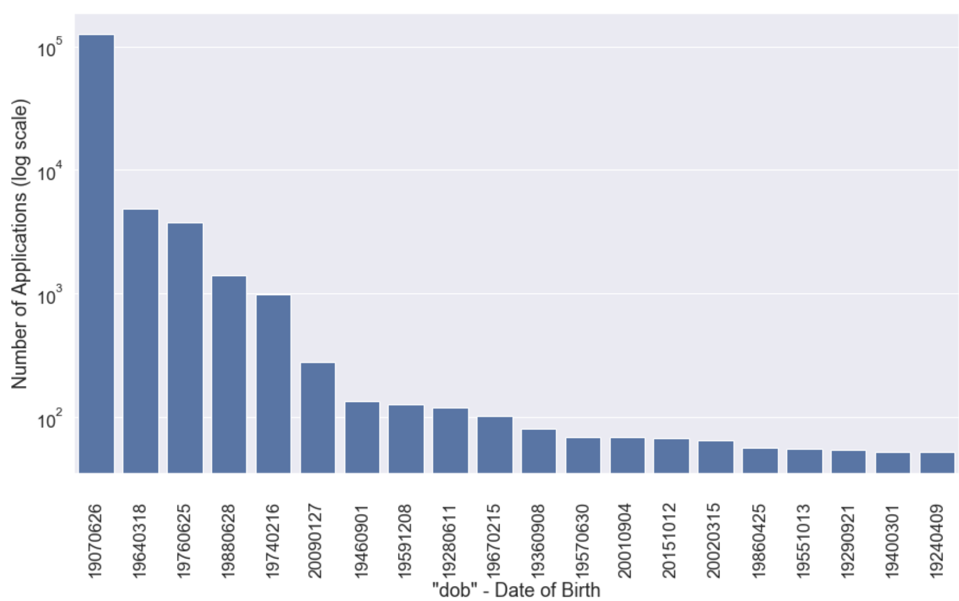


Figure 9: Top 20 Date of Birth Frequencies Including Frivolous Value.

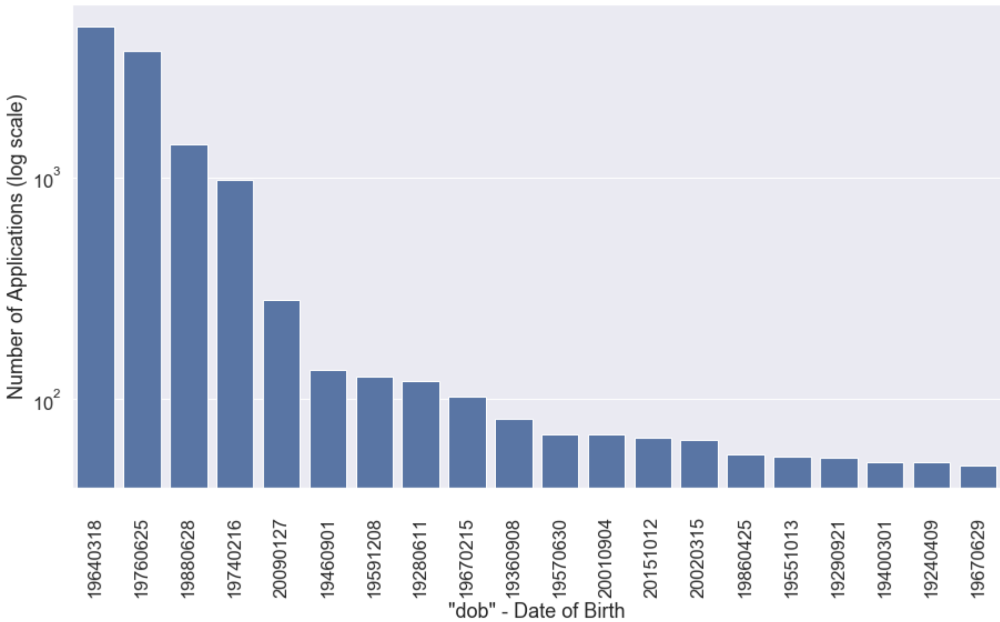


Figure 10: Top 20 Date of Birth Frequencies Excluding Frivolous Value.

Data Cleaning

All the data fields were 100% populated; however, we found that four of the data fields contained frivolous values. Figure 11 below shows the data fields, their frivolous values, and the number of application records with frivolous values in that data field.

Data Field	Frivolous Value	Number of Records
ssn	999999999	16,935
address	123 MAIN ST	1,079
dob	19070626	126,568
homephone	9999999999	78,512

Figure 11. Frivolous Values.

The use of frivolous values in application data is common due to varying requirements amongst companies, applicant omissions, possible privacy related issues, and a variety of other reasons. Before a full analysis can be conducted on a dataset containing frivolous values, those values must be cleaned to help avoid potential errors in the results and false conclusions. For this particular dataset, we opted to replace the frivolous values with a unique value.

There are different methods that can be used to clean frivolous values by replacing them with unique values. Entering a unique randomized number is one example of a method that can be used, but we opted to use the application's unique record number from the "record" data field to replace any frivolous values for that particular application record. For instance, if application record number 75 had a value of "999999999" in the "ssn" data field and a value of "123 MAIN ST" in the "address" data field, then the number 75 was used to replace the frivolous values in both the "ssn" and "address" data fields. This method was used for all frivolous values in the dataset until the dataset was completely free and clear of frivolous values.

More specifically, from a programming perspective, we used Python and the "where" function of the NumPy package to replace the frivolous values. The code is provided in the screenshot below and details on NumPy's "where" function can be found at https://docs.scipy.org/doc/numpy/reference/generated/numpy.where.html.

```
# Replace all frivolous values with the record number.
data['ssn'] = np.where(data['ssn'] == 999999999, data['record'], data['ssn'])
data['address'] = np.where(data['address'] == "123 MAIN ST", data['record'], data['address'])
data['dob'] = np.where(data['dob'] == 19070626, data['record'], data['dob'])
data['homephone'] = np.where(data['homephone'] == 9999999999, data['record'], data['homephone'])
```

Candidate Variable

In detecting potential fraud records from product application data, it is critical to develop additional candidate variables from the data fields in the original dataset that are related to the intended objective. Crafting candidate variables is a bit of an artform and necessitates not only an understanding of the data and the different modeling techniques, but domain knowledge of the subject area to include the business processes and customer or human behaviors surrounding that subject. For detecting potential fraud in the product application dataset, building candidate variables that relate to the periodicity and speed of product applications as they pertain to detecting individual fraud and victim identity fraud are of great importance. Therefore, we built a number of candidate variables based on the data fields in the original dataset as they applied to the concepts of the speed in which applications were seen (velocity variables), the number of days since the last time an application was seen (days since variables), and the speed in which applications were seen over a certain period of time in relation to what was considered normal (relative velocity).

Initial Variable Crafting

Before building numerical candidate variables that relate to periodicity and speed, we first needed to craft additional categorical variables that would serve as the basis for the periodicity and speed. This meant carefully thinking through logical combinations of the original data fields that fraudsters could possibly use in product applications.

From a programming perspective, we could have easily created every combination and concatenation of the original data fields through the use of tools and packages such as the "combinations()" function from Python's internal "itertools" module (reference https://docs.python.org/3.8/library/itertools.html#module-itertools), but this type of methodology can cause an unnecessary explosion in the volume of variables and data created for analysis. Such an approach could also result in resource-intensive computations that may necessitate server and/or cloud computing capabilities. We therefore opted to judiciously select rational and reasonable combinations of the original data fields and then manually create the necessary concatenations. This resulted in the development of 27 categorical variables as seen in Figure 12.

	Candidate Variable	Concatenation of Original Data Fields
1	**fullname**	firstname, lastname
2	**fullname-dob**	firstname, lastname, dob
3	**fullname-ssn**	firstname, lastname, ssn
4	**fullname-homephone**	firstname, lastname, homphone
5	**fullname-address**	firstname, lastname, address
6	**fullname-address-zip**	firstname, lastname, address, zip
7	**fullname-dob-homephone**	firstname, lastname, dob, homephone
8	**fullname-dob-zip**	firstname, lastname, dob, zip
9	**fullname-zip**	firstname, lastname, zip
10	**firstname-dob**	firstname, dob
11	**lastname-dob**	lastname, dob
12	**firstname-homephone**	firstname, homephone
13	**lastname-homephone**	lastname, homephone
14	**ssn-firstname**	ssn, firstname
15	**ssn-lastname**	ssn, lastname
16	**ssn-zip**	ssn, zip
17	**ssn-dob**	ssn, dob
18	**ssn-homephone**	ssn, homephone
19	**ssn-address**	ssn, address
20	**ssn-address-zip**	ssn, address, zip
21	**ssn-fullname-dob**	ssn, fullname, dob
22	**address-zip**	address, zip
23	**address-zip-fullname-dob**	address, zip, firstname, lastname, dob
24	**address-zip-homephone**	address, zip, homephone
25	**zip-homephone**	zip, homephone
26	**zip-dob**	zip, dob
27	**homephone-dob**	homephone, dob

Figure 12. Categorical Candidate Variables.

To perform these concatenations, we simply ensured each of the original data fields were converted to a string data type and then we manually concatenated the data fields by adding them together. Figure 13 below shows the equation for each categorical candidate variable, where "df_var" refers to the Pandas dataframe containing the original data fields and the new candidate variables.

```
# Make combinations with the name
df_var['fullname'] = df_var['firstname'] + df_var['lastname']
df_var['fullname-dob'] = df_var['fullname'] + df_var['dob']
df_var['fullname-ssn'] = df_var['fullname'] + df_var['ssn']
df_var['fullname-homephone'] = df_var['fullname'] + df_var['homephone']
df_var['fullname-address'] = df_var['fullname'] + df_var['address']
df_var['fullname-address-zip'] = df_var['fullname'] + df_var['address'] + df_var['zip5']
df_var['fullname-dob-homephone'] = df_var['fullname'] + df_var['dob'] + df_var['homephone']
df_var['fullname-dob-zip'] = df_var['fullname'] + df_var['dob'] + df_var['zip5']
df_var['fullname-zip'] = df_var['fullname'] + df_var['zip5']
df_var['firstname-dob'] = df_var['firstname'] + df_var['dob']
df_var['lastname-dob'] = df_var['lastname'] + df_var['dob']
df_var['firstname-homephone'] = df_var['firstname'] + df_var['homephone']
df_var['lastname-homephone'] = df_var['lastname'] + df_var['homephone']
```

```
# Make combinations with the ssn
df_var['ssn-firstname'] = df_var['ssn'] + df_var['firstname']
df_var['ssn-lastname'] = df_var['ssn'] + df_var['lastname']
df_var['ssn-zip'] = df_var['ssn'] + df_var['zip5']
df_var['ssn-dob'] = df_var['ssn'] + df_var['dob']
df_var['ssn-homephone'] = df_var['ssn'] + df_var['homephone']
df_var['ssn-address'] = df_var['ssn'] + df_var['address']
df_var['ssn-address-zip'] = df_var['ssn'] + df_var['address'] + df_var['zip5']
df_var['ssn-fullname-dob'] = df_var['ssn'] + df_var['fullname'] + df_var['dob']
```

```
# Make combinations of other data fields
df_var['address-zip'] = df_var['address'] + df_var['zip5']
df_var['address-zip-fullname-dob'] = df_var['address'] + df_var['zip5'] + df_var['fullname'] + df_var['dob']
df_var['address-zip-homephone'] = df_var['address'] + df_var['zip5'] + df_var['homephone']
df_var['zip-homephone'] = df_var['zip5'] + df_var['homephone']
df_var['zip-dob'] = df_var['zip5'] + df_var['dob']
df_var['homephone-dob'] = df_var['homephone'] + df_var['dob']
```

Figure 13. Categorical Candidate Variables.

With the creation of the 27 categorical candidate variables, we were then prepared to use them as the foundation for developing the numerical candidate variables for the velocity, days-since, and relative velocity variables.

Velocity Candidate Variables

The velocity variables are numeric in nature and capture the speed at which categorical data fields and candidate variables were seen in a dataset for a particular application record. Since fraudsters often operate in bursts of time, calculating the speed in which applications are seen is one way of finding these bursts of time as we seek to identify potentially fraudulent applications. A higher value calculated for a particular time period means there is increased risk and a stronger likelihood of fraud. The general equation for calculating the value for a velocity variable is as follows:

$$Velocity \;=\; \#\; of\; records \text{ with the same } data\; field \text{ over the last } X\; days$$

$$Velocity \;=\; \#\; of\; records \text{ with the same } variable \text{ over the last } X\; days$$

where X is an integer value from 0 to ∞ and the $\#\; of\; records$ is restricted to counting only the current record at hand and any records in the past (i.e. the $\#\; of\; records$ cannot include the count of records forward in time as it relates to time of day). It is important to note that a value of 0 for X represents the time period used for application records seen within the same day. From a more visual perspective, this equation can be viewed as follows:

$$\text{Velocity} = \text{ \# records with the same } \left\{ \begin{array}{c} \text{ssn} \\ \text{address} \\ \text{fullname-dob} \\ \text{fullname-address-zip} \\ \text{homephone} \\ \text{ssn-address} \\ \text{ssn-fullname-dob} \\ \text{ssn-homephone} \\ \ldots \end{array} \right\} \text{ over the last } \left\{ 0, 1, 3, 7, 14. . . \right\} \text{ days}$$

For instance, in calculating the velocity of the "ssn" data field over the last 1-day period, the value of the velocity variable for a particular record only counts the current application record and any previous application records in the dataset that came before the current application record. In Figure 14 below, we see that for the application records with an "ssn" value of "908225968" in the dataset, the velocity variable containing the number of records over a 1-day period is called "ssn_velocity1_date" and it contains the respective values for each application record. For record number 12233, the value of "ssn_velocity1_date" can only include the current record and those records from 2016-01-04 up to current record. Since there were 4 records on 2016-01-04 and 1 previous record on 2016-01-05 before record number 12233, the value for the velocity in "ssn_velocity1_date" is 6 for record number 12233.

record	date	ssn	ssn_velocity0_date	ssn_velocity1_date
921	2016-01-01	908225968	1	1
3334	2016-01-02	908225968	1	2
7200	2016-01-03	908225968	1	2
8361	2016-01-04	908225968	1	2
9498	2016-01-04	908225968	2	3
10120	2016-01-04	908225968	3	4
11027	2016-01-04	908225968	4	5
11953	2016-01-05	908225968	1	5
12233	2016-01-05	908225968	2	6
12406	2016-01-05	908225968	3	7

Figure 14. 0-Day and 1-Day Candidate Velocity Variables for SSN.

The equation and procedure described above was used for each of the 27 categorical candidate variables we created and for the “ssn”, “address”, “dob”, and “homephone” data fields. For the value of X, we used time periods of 0, 1, 3, 7, 14, 30, 90, and 180 days. After calculating all the velocity variables for the 31 data fields and categorical candidate variables, we ended up with 248 velocity candidate variables. A snapshot of some of the velocity candidate variables are provided in Figure 15 below. For the full list of the velocity candidate variables, please see Appendix B.

	Velocity Candidate Variable		Velocity Candidate Variable
1	**ssn_velocity0_date**	25	**homephone_velocity0_date**
2	**ssn_velocity1_date**	26	**homephone_velocity1_date**
3	**ssn_velocity3_date**	27	**homephone_velocity3_date**
4	**ssn_velocity7_date**	28	**homephone_velocity7_date**
5	**ssn_velocity14_date**	29	**homephone_velocity14_date**
6	**ssn_velocity30_date**	30	**homephone_velocity30_date**
7	**ssn_velocity90_date**	31	**homephone_velocity90_date**
8	**ssn_velocity180_date**	32	**homephone_velocity180_date**
9	**address_velocity0_date**	33	**fullname_velocity0_date**
10	**address_velocity1_date**	34	**fullname_velocity1_date**
11	**address_velocity3_date**	35	**fullname_velocity3_date**
12	**address_velocity7_date**	36	**fullname_velocity7_date**
13	**address_velocity14_date**	37	**fullname_velocity14_date**
14	**address_velocity30_date**	38	**fullname_velocity30_date**
15	**address_velocity90_date**	39	**fullname_velocity90_date**
16	**address_velocity180_date**	40	**fullname_velocity180_date**
17	**dob_velocity0_date**	41	**fullname-dob_velocity0_date**
18	**dob_velocity1_date**	42	**fullname-dob_velocity1_date**
19	**dob_velocity3_date**	43	**fullname-dob_velocity3_date**
20	**dob_velocity7_date**	44	**fullname-dob_velocity7_date**
21	**dob_velocity14_date**	45	**fullname-dob_velocity14_date**
22	**dob_velocity30_date**	46	**fullname-dob_velocity30_date**
23	**dob_velocity90_date**	47	**fullname-dob_velocity90_date**
24	**dob_velocity180_date**	48	**fullname-dob_velocity180_date**

Figure 15. Snapshot of Velocity Candidate Variables.

Days-Since Candidate Variables

The days-since candidate variables are numeric in nature and capture the number of days since the last time a categorical data field or candidate variable was seen for a particular application record. Since fraudsters often operate in bursts of time, calculating the number of days between a product application activity is another way of finding and characterizing these bursts of time as we seek to identify potentially fraudulent applications. Apart from the value of 0, a lower calculated value for this type of variable means there is increased risk and a higher likelihood of fraud. The general equation for calculating the value for a days-since variable is as follows:

$$days - since \ = \ \# \, of \ days \ since \ a \ data \ field \ was \ last \ seen$$

$$days - since \ = \ \# \, of \ days \ since \ a \ variable \ was \ last \ seen$$

From a more visual perspective, this equation can be viewed as follows:

Days since = # days since I last saw that

ssn
address
fullname-dob
fullname-address-zip
homephone
ssn-address
ssn-fullname-dob
ssn-homephone
…

As an example, let us revisit the application records with an "ssn" value of "908225968" in the dataset as seen in Figure 16 below. The application records for the "ssn" value of "908225968" all occurred during the first few days of January in 2016. On January 1, 2016, it was the very first day we saw the "ssn" value of "908225968" in the dataset. Thus, record number 921 has a days-since value of 0. However, since the "ssn" value of "908225968" was used over consecutive days from January 2, 2016 to January 5, 2016, each of the subsequent record numbers have a days-since value of 1 because there was only 1 day or less since the "ssn" value of "908225968" was last seen.

	record	date	ssn	ssn_daysSince
0	921	2016-01-01	908225968	0.0
1	3334	2016-01-02	908225968	1.0
2	7200	2016-01-03	908225968	1.0
3	8361	2016-01-04	908225968	1.0
4	9498	2016-01-04	908225968	1.0
5	10120	2016-01-04	908225968	1.0
6	11027	2016-01-04	908225968	1.0
7	11953	2016-01-05	908225968	1.0
8	12233	2016-01-05	908225968	1.0
9	12406	2016-01-05	908225968	1.0

Figure 16. Days-Since Candidate Variable for SSN.

The days-since candidate variables were resource intensive calculations. As such, we opted to calculate only a handful of days-since candidate variables for analysis. Figure 17 below shows all 14 of the days-since candidate variables we calculated on the dataset. Also, since these variables use a slightly different naming convention, Figure 17 provides the data field or categorical candidate variable that served as the basis for the days-since calculations. For a complete list of the days-since candidate variables, please see Appendix B.

	Days-Since Candidate Variable	Base Data Field or Variable
1	**ssn_daysSince**	ssn
2	**address_daysSince**	address-zip
3	**ndob_daysSince**	fullname-dob
4	**phone_daysSince**	homephone
5	**ssnaddress_daysSince**	ssn-address-zip
6	**ndobaddress_daysSince**	fullname-dob
7	**phoneaddress_daysSince**	address-zip-homephone
8	**ndobphone_daysSince**	fullname-dob-homephone
9	**addressphone_daysSince**	address-zip-homephone
10	**phonessn_daysSince**	ssn-homephone
11	**ndobaddress_daysSince**	address-zip-fullname-dob
12	**ssnndob_daysSince**	ssn-fullname-dob
13	**lastnamessn_daysSince**	ssn-lastname
14	**fnssn_daysSince**	ssn-firstname

Figure 17. Days-Since Candidate Variables.

Relative Velocity Candidate Variables

The relative velocity candidate variables are numeric in nature and capture the speed at which a categorical data field or candidate variable was seen for a particular application record over a certain period of time in relation to what was considered normal. This candidate variable is a comparative type of variable where we analyze how often we see a data field or variable during a short period of time (e.g. 0-day or 1-day time periods) in comparison to how often we see that data field or variable over a longer period of time (e.g. 7-day, 14-day, or longer time periods). We are therefore using this type of variable as an indication of fraud by determining if we are seeing more application records with a particular data field or variable within a short period of time versus how often we would normally see an application record with that data field or variable. A higher value for a relative velocity calculation correlates to an increased risk and a higher likelihood of fraudulent activity. The general equation for calculating the value for a relative velocity variable is as follows:

ssn
address
fullname-dob
fullname-address-zip
homephone
ssn-address
ssn-fullname-dob
ssn-homephone
...

$$\text{Relative velocity} = \frac{\text{\# apps with that } \textbf{\textit{group}} \text{ seen in the recent past}}{\text{\# apps with that } \textbf{\textit{same group}} \text{ seen in the past } \{1, 3, 7, 14, 30\} \text{ days}}$$

The equation above was used for each of the 27 categorical candidate variables we created and for the "ssn", "address", "dob", and "homephone" data fields. For the "recent past" values (numerator), we used the 0-day and 1-day velocity candidate variable values. For the "past days" values (denominator), we used the 3-day, 7-day, 14-day, 30-day, 90-day, and 180-day velocity candidate variable values. After calculating all of the relative velocity variables for the data fields and categorical candidate variables, we ended up with 372 relative velocity candidate variables. A snapshot of some of the relative velocity candidate variables are provided in Figure 18 below. For the full list of the relative velocity candidate variables, please see Appendix B.

	Relative Velocity Candidate Variable		Relative Velocity Candidate Variable
1	**ssn_0_dayvel_div_3_dayvel_relvelocity**	25	**dob_0_dayvel_div_3_dayvel_relvelocity**
2	**ssn_0_dayvel_div_7_dayvel_relvelocity**	26	**dob_0_dayvel_div_7_dayvel_relvelocity**
3	**ssn_0_dayvel_div_14_dayvel_relvelocity**	27	**dob_0_dayvel_div_14_dayvel_relvelocity**
4	**ssn_0_dayvel_div_30_dayvel_relvelocity**	28	**dob_0_dayvel_div_30_dayvel_relvelocity**
5	**ssn_0_dayvel_div_90_dayvel_relvelocity**	29	**dob_0_dayvel_div_90_dayvel_relvelocity**
6	**ssn_0_dayvel_div_180_dayvel_relvelocity**	30	**dob_0_dayvel_div_180_dayvel_relvelocity**
7	**ssn_1_dayvel_div_3_dayvel_relvelocity**	31	**dob_1_dayvel_div_3_dayvel_relvelocity**
8	**ssn_1_dayvel_div_7_dayvel_relvelocity**	32	**dob_1_dayvel_div_7_dayvel_relvelocity**
9	**ssn_1_dayvel_div_14_dayvel_relvelocity**	33	**dob_1_dayvel_div_14_dayvel_relvelocity**
10	**ssn_1_dayvel_div_30_dayvel_relvelocity**	34	**dob_1_dayvel_div_30_dayvel_relvelocity**
11	**ssn_1_dayvel_div_90_dayvel_relvelocity**	35	**dob_1_dayvel_div_90_dayvel_relvelocity**
12	**ssn_1_dayvel_div_180_dayvel_relvelocity**	36	**dob_1_dayvel_div_180_dayvel_relvelocity**
13	**address_0_dayvel_div_3_dayvel_relvelocity**	37	**homephone_0_dayvel_div_3_dayvel_relvelocity**
14	**address_0_dayvel_div_7_dayvel_relvelocity**	38	**homephone_0_dayvel_div_7_dayvel_relvelocity**
15	**address_0_dayvel_div_14_dayvel_relvelocity**	39	**homephone_0_dayvel_div_14_dayvel_relvelocity**
16	**address_0_dayvel_div_30_dayvel_relvelocity**	40	**homephone_0_dayvel_div_30_dayvel_relvelocity**
17	**address_0_dayvel_div_90_dayvel_relvelocity**	41	**homephone_0_dayvel_div_90_dayvel_relvelocity**
18	**address_0_dayvel_div_180_dayvel_relvelocity**	42	**homephone_0_dayvel_div_180_dayvel_relvelocity**
19	**address_1_dayvel_div_3_dayvel_relvelocity**	43	**homephone_1_dayvel_div_3_dayvel_relvelocity**
20	**address_1_dayvel_div_7_dayvel_relvelocity**	44	**homephone_1_dayvel_div_7_dayvel_relvelocity**
21	**address_1_dayvel_div_14_dayvel_relvelocity**	45	**homephone_1_dayvel_div_14_dayvel_relvelocity**
22	**address_1_dayvel_div_30_dayvel_relvelocity**	46	**homephone_1_dayvel_div_30_dayvel_relvelocity**
23	**address_1_dayvel_div_90_dayvel_relvelocity**	47	**homephone_1_dayvel_div_90_dayvel_relvelocity**
24	**address_1_dayvel_div_180_dayvel_relvelocity**	48	**homephone_1_dayvel_div_180_dayvel_relvelocity**

Figure 18. Snapshot of Relative Velocity Candidate Variables.

Feature Selection Process

Feature selection is the process in statistics that aims to reduce the number of features (dimensions) of input variables. Feature selection is performed in such a way so that the most statistically important features are selected from the original input. Also, features that are either highly correlated with others or not significant to perform an accurate prediction are ignored. Feature selection is pivotal in modern statistical learning as it often allows reduced computational costs and increased model performance. There are three main methods of feature selection:

1. **Filter Methods:** Filter methods use statistical measures to evaluate the relationship (correlation) of two distributions and measure the correlation between the distribution of each of the classes of each feature and the dependent variable. The features that are chosen are the ones with the highest correlation with the dependent variable.

2. **Wrapper Methods:** Wrapper methods utilize statistical models to evaluate the performance of each feature (or a subset of features) based on a performance metric (accuracy, AUC, f1 score, etc.). A common wrapper method is recursive feature elimination, in which a model recursively uses smaller and smaller sets of features until a desired number of features is reached.

3. **Embedded Methods:** Embedded methods perform feature elimination as the model is built. A common embedded method for feature selection is regularization, in which a norm is included in the loss function of a statistical model to penalize the number of features used.

Filter Methods

For our analysis, we performed feature selection using the Kolmogorov-Smirnov distance and the Fraud Detection Rate.

Kolmogorov–Smirnov (KS) Distance

The Kolmogorov-Smirnov (KS) distance metric measures the maximum distance between two distributions to determine how well the distributions are separated. A higher KS distance value correlates to a better separation between the two distributions and therefore a better feature in the context of feature selection. For this analysis, we used the KS distance metric by calculating the univariate KS value as a filtering method to aid in determining which features provide a better separation between the “fraud_label” values of 1 and 0. Meaning, for each numerical candidate variable, we generated the distribution of the two classes (1 and 0) based on the dependent variable ("fraud_label” data field). Subsequently, we measured the KS distance between the distributions of the two classes for each of the numerical candidate variables. More formally,

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

We then rank ordered the KS distance value in descending order for each of the numerical candidate variables and used this ranking to evaluate the importance of each variable.

Fraud Detection Rate (FDR) at 3%

The second metric we used for filtering the features was the Fraud Detection Rate (FDR). In general, the FDR is the percentage of all the frauds that are detected up to a particular cutoff point. In the context of this analysis for feature selection, we used a cutoff threshold of 3% and calculated the univariate FDR for each numerical candidate variable. The FDR at 3% was determined by first sorting the numerical candidate variables in descending order, and then computing the percentage of frauds in the top 3%. We then assigned a rank for each of the numerical candidate variables and used this ranking as a means to evaluate the importance of each variable.

Filter Method Aggregated Results

In our analysis, we obtained the univariate KS distance and the univariate FDR at 3% for each numerical candidate variable and used their average rank to serve as a final score on the importance of each feature. Subsequently, we kept the top 33% ranked numerical candidate variables, reducing the total number of features from 634 to 217.

Wrapper Methods

We used logistic regression with L2 penalty and 100 iterations as our model for the wrapper method for feature elimination. Logistic regression is a simple model to implement with a relatively low computational cost. We used recursive feature elimination with 3-fold cross validation and in each iteration, we cut one feature. We repeated this process twice. In the first pass, the number of features was decreased to 99 and in the second pass the number of features was decreased to the desired 30.

Final Variables	
address-zip-homephone_0_dayvel_div_180_dayvel_relvelocity	ssn-dob_0_dayvel_div_30_dayvel_relvelocity
address-zip-homephone_velocity30_date	ssn-dob_velocity14_date
address-zip-homephone_velocity7_date	ssn-dob_velocity30_date
address-zip_0_dayvel_div_7_dayvel_relvelocity	ssn-firstname_0_dayvel_div_180_dayvel_relvelocity
address-zip_velocity0_date	ssn-firstname_velocity30_date
address-zip_velocity30_date	ssn-firstname_velocity3_date
fullname-dob_velocity30_date	ssn-fullname-dob_0_dayvel_div_14_dayvel_relvelocity
fullname-ssn_0_dayvel_div_180_dayvel_relvelocity	ssn-fullname-dob_0_dayvel_div_180_dayvel_relvelocity
fullname-ssn_velocity14_date	ssn-fullname-dob_0_dayvel_div_30_dayvel_relvelocity
fullname-ssn_velocity30_date	ssn-fullname-dob_velocity14_date
fullname-ssn_velocity7_date	ssn-fullname-dob_velocity30_date
homephone_velocity1_date	ssn-lastname_0_dayvel_div_180_dayvel_relvelocity
ssn-dob_0_dayvel_div_14_dayvel_relvelocity	ssn-lastname_velocity14_date
ssn-dob_0_dayvel_div_180_dayvel_relvelocity	ssn-lastname_velocity30_date
ssn_velocity1_date	ssn_0_dayvel_div_30_dayvel_relvelocity

Figure 19: Final Variables for Algorithms

Model Algorithms

Logistic Regression

Logistic regression is one of the simplest and commonly used machine learning algorithms for binary classification problems. It is derived from a logit model which tries to predict the log of odds of a model as a linear combination of the predictor(s):

$$\log(O(Y=1|X=x))=\beta_0+\beta_1 X$$

With some rearrangement, you can get the formula for logistic regression:

$$p(X) = \frac{e^{\beta_0+\beta_1 X}}{1+e^{\beta_0+\beta_1 X}}.$$

Logistic regression follows a sigmoid function. It describes and estimates the relationship between one dependent binary variable and the independent variables. The sigmoid function gives an 'S' shaped curve that can take any real-valued number and map it into a value between 0 and 1 as seen in Figure 20 below. If the output of the sigmoid function is more than 0.5, we can classify the outcome as 1, and if it is less than 0.5, we can classify it as 0.

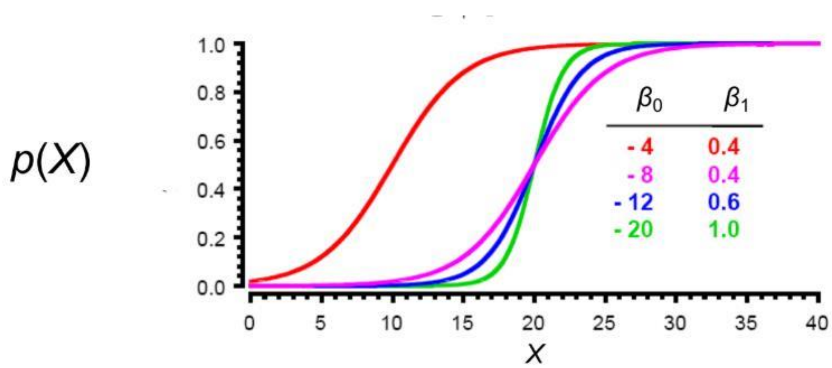


Figure 20: Sigmoid Function.

Parameters control the shape and location of the sigmoid curve, where β_0 controls the location of midpoint and β_1 controls the rise of the slope.

For our fraud analysis, we predicted the fraud label using logistic regression, ranked the records by probability in descending order and calculated the FDR at 3% for training, testing and OOT data separately. We first used 30 variables in total, and selected 10, 20, and 30 variables for building models. We also tried using 25 variables in total, and then selected 5, 15, and 25 variables for building models. Our logistic regression results are below:

Model	Parameter		Average FDR at 3%		
Logistic Regression	**Total Variables**	**Number of Variables Selected**	**TRAIN**	**TEST**	**OOT**
1	30	10	0.520346	0.507287	0.501258
2	30	20	0.547640	0.535296	0.518022
3	30	30	0.552899	0.543012	0.525985
4	25	5	0.463278	0.459560	0.452221
5	25	15	0.510079	0.499858	0.486589
6	25	25	0.549769	0.537582	0.530537

Figure 21: Logistic Regression Model Results.

Boosted Trees

In general, boosting is a method of improving prediction results by iteratively training a series of weak learners so that they may produce a strong learner. In applying the concept of boosting to decision trees, we get boosted trees. Boosted trees are a series of simple or constrained decision trees used as weak learners that are grown sequentially such that each subsequent tree in the series is grown using information about the misclassified results from the previous tree. In a very simplistic form, if we let $h(x)$ be a weak learner's output and we let w be the weight relative to the weak learner's accuracy, then the predicted output ($\hat{y}(x)$) for the t-th iteration is

$$\hat{y}(x) = \sum_t w_t h_t(x)$$

As the boosted tree algorithm progresses, it applies a weight (w_t) to each data record in the dataset relative to the accuracy of the output. The weights (w_t) for a record are therefore dependent on whether a record was misclassified with weights (w_t) increasing for misclassified records as it is subsequently misclassified throughout the algorithm so that each iterative decision tree in the series can place a heavier importance on that record to increase the likelihood of properly classifying the record. The overall idea is that by fitting a series of weak learners to the residuals (i.e. misclassified results), we slowly improve the model in areas where it does not perform well. The ultimate goal is to minimize the losses or the residual error in the objective function to improve prediction results and accuracy. Refer to Figure 22 below for a visualization of boosted trees.

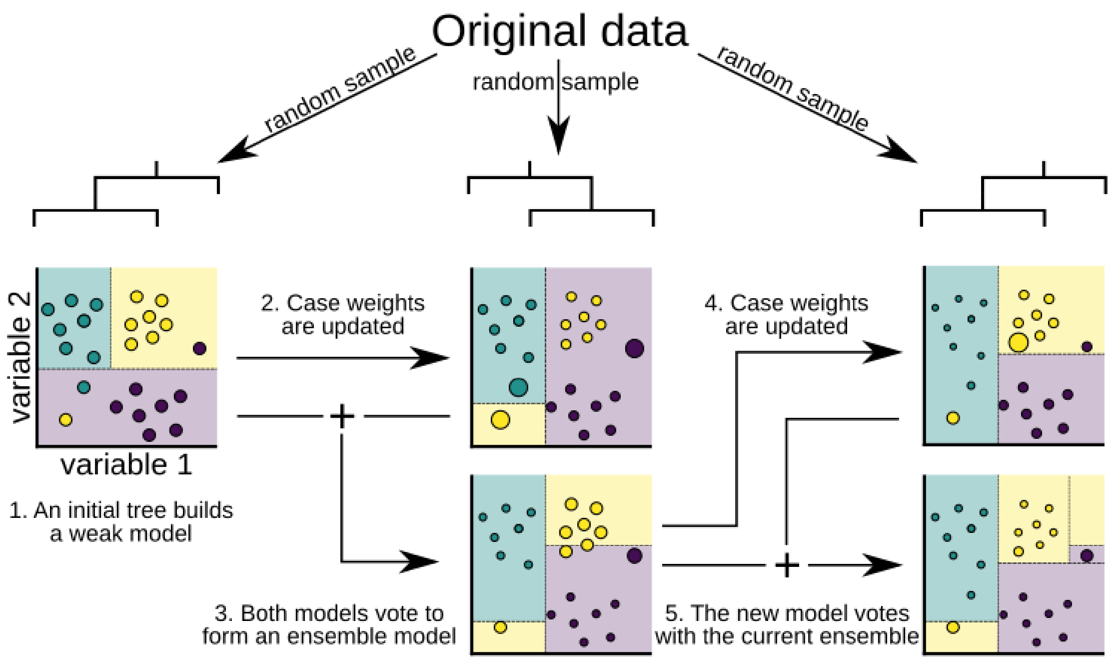


Figure 22: Boosted Trees Algorithm.

Boosted Trees Applied to the Dataset

There are a variety of boosted tree algorithms readily available for use within many of the popular machine learning libraries. For instance, Scikit-Learn and XGBoost are two popular machine learning libraries that contain boosted tree algorithms. For this dataset, we opted to use the XGBClassifier package from XGBoost's machine learning library for our boosted tree algorithm.

For any boosted tree algorithm, there are a few key hyperparameters that help with tuning the algorithm for a particular problem. Namely, the number of decision trees, the size of the decision tree, and the algorithm's learning rate. For the XGBClassifier, the number of decision trees corresponds to the "n_estimators" parameter, the size of the decision tree corresponds to the "max_depth" parameter, and the learning rate corresponds to the "eta" parameter. In selecting values for these hyperparameters, the information below provides some general guidelines when attempting to find optimal values:

- *Number of decision trees*: Too many trees can cause a model to overfit. However, it is generally good practice to increase the number of trees until there is little to no improvement in the model.
- *Decision tree depth*: A higher tree depth correlates to increased complexity. In a boosted tree algorithm, we are attempting to use a series of weak learners. Thus, shorter or less complex decisions trees are generally used in a boosted tree algorithm.
- *Learning rate*: The speed at which the algorithm learns from the updated weights at each iteration in the series can be controlled to help make the algorithm more robust to overfitting. Lowering the learning rate can shrink the weights at each step and therefore slow down the

speed at which the algorithm learns. This inherently means that more trees are often needed to tune the model and more time will be needed for the model to finish training. Common values for the learning rate are 0.01 and 0.001. However, when tuning this parameter, higher values may prove to be more optimal if the lower values show to result in little to no improvement.

In applying the XGBClassifier to the dataset, we used a very short list of values for each of the parameters above to conduct an initial analysis of the performance of the algorithm on the dataset. The results are shown in Figure 23 below.

Boosted Tree	Num of Variables	Num of Trees	Max Depth	Learning Rate	TRAIN FDR	TEST FDR	OOT FDR
1	30	600	4	0.01	0.57365	0.56179	0.54652
2	30	800	4	0.01	0.57352	0.56090	0.54819
3	30	1000	4	0.01	0.57389	0.56179	0.54819
4	30	600	5	0.01	0.57192	0.55971	0.54610
5	30	800	5	0.01	0.57389	0.56090	0.54831
6	30	1000	5	0.01	0.57438	0.56209	0.54803
7	30	600	4	0.001	0.55751	0.54634	0.53143
8	30	800	4	0.001	0.55899	0.54723	0.53269
9	30	1000	4	0.001	0.55899	0.54723	0.53269
10	30	600	5	0.001	0.55665	0.54367	0.53059
11	30	800	5	0.001	0.55849	0.54486	0.53185
12	30	1000	5	0.001	0.55825	0.54486	0.53185

Figure 23. Boosted Trees Results.

In addition to the results above, we attempted to use cross-validation and parameter tuning with Scikit-Learn’s GridSearchCV package to evaluate the XGBClassifier with additional values for each of the three main parameters seen in Figure 23 and with the “scale_pos_weight” parameter to help deal with the unbalanced labels (i.e. classes). Unfortunately, there was no major improvement in the accuracy of the model or the FDR scores. Given the lack of improvement and the results from our initial analysis on the other algorithms, we opted to forego detailed hyperparameter tuning for the XGBClassifier.

Neural Network

Neural networks are statistical models widely used today in many applications, including classification, image processing, and speech processing among others. The neural network is based on the perceptron algorithm, originally introduced in the 1950s and consists of multiple layers of perceptrons (or neurons). The neural network tries to imitate the function of a human brain, i.e, try to learn things, distinguish patterns and make decisions by training, in the same way that a human brain does.

A neural network consists of multiple layers of neurons:

- an input layer that is formed by the independent variables
- hidden layers
- the output layer which is formed by the dependent variable

Each of the neurons of each layer is connected to all neurons of the next layer, i.e, the output of a neuron on the i-th layer is the input of the neurons in the (i+1)-th layer. Each neuron has an optional threshold and an activation function, and each neuron connection has an associated weight that represents the significance of this connection. Each neuron receives as input a weighted signal and outputs the result of its activation function of that weighted signal. Subsequently, this output is multiplied by the weight of the neuron's output connection and is propagated to all the neurons in the next layer. The activation function serves to scale the input of the neuron and to provide a smooth, differentiable transition as input values change. Common activation functions are relu, sigmoid, tanh and are presented in Figure 24.

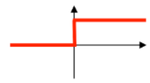
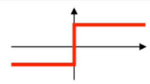
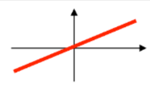
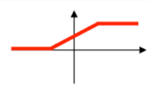
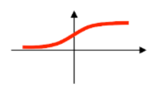
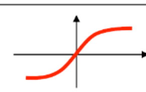
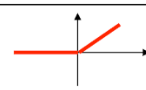
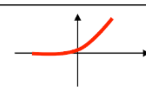
Activation function	Equation	Example	1D Graph
Unit step (Heaviside)	$\phi(z) = \begin{cases} 0, & z < 0, \\ 0.5, & z = 0, \\ 1, & z > 0, \end{cases}$	Perceptron variant	
Sign (Signum)	$\phi(z) = \begin{cases} -1, & z < 0, \\ 0, & z = 0, \\ 1, & z > 0, \end{cases}$	Perceptron variant	
Linear	$\phi(z) = z$	Adaline, linear regression	
Piece-wise linear	$\phi(z) = \begin{cases} 1, & z \geq \frac{1}{2}, \\ z + \frac{1}{2}, & -\frac{1}{2} < z < \frac{1}{2}, \\ 0, & z \leq -\frac{1}{2}, \end{cases}$	Support vector machine	
Logistic (sigmoid)	$\phi(z) = \frac{1}{1 + e^{-z}}$	Logistic regression, Multi-layer NN	
Hyperbolic tangent	$\phi(z) = \frac{e^{z} - e^{-z}}{e^{z} + e^{-z}}$	Multi-layer Neural Networks	
Rectifier, ReLU (Rectified Linear Unit)	$\phi(z) = max(0, z)$	Multi-layer Neural Networks	
Rectifier, softplus	$\phi(z) = \ln(1 + e^{z})$	Multi-layer Neural Networks	

Figure 24. Common Activation Functions.

In Figure 25, we can see a neural network architecture.

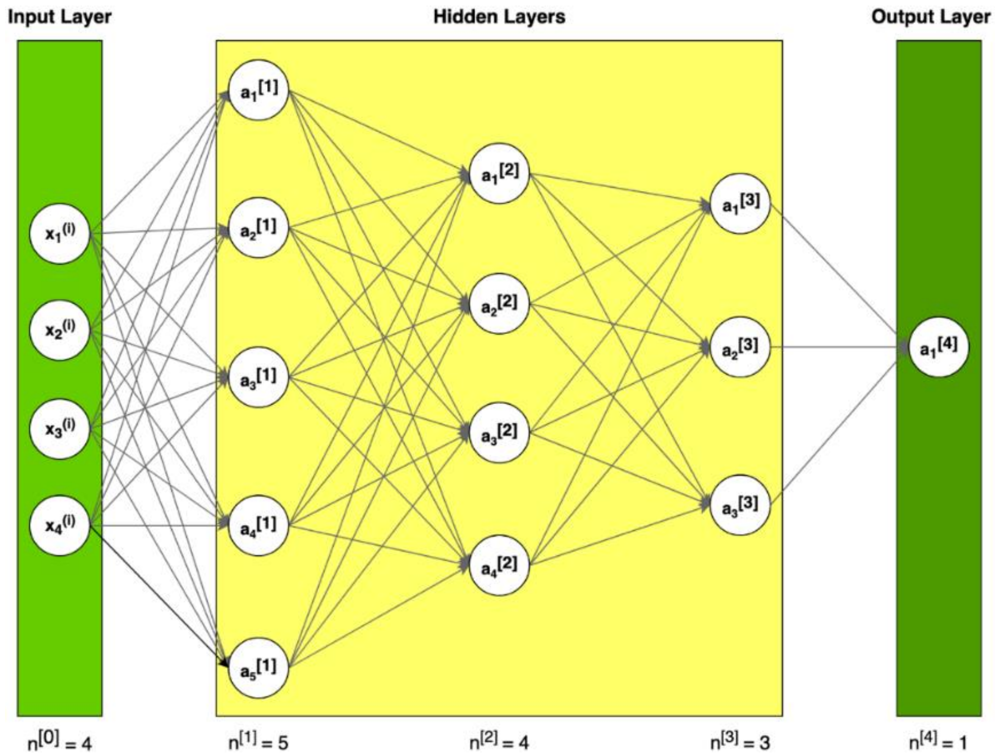


Figure 25. Neural Network Architecture

A neural network tries to predict the dependent variable (or label) of an input vector (independent variable). This is achieved by calculating the appropriate weights of the connections of the neurons in the network. Thus, the weights become a variable that we are trying to optimize over all available data in the input so that we have more accurate predictions as possible. For that matter, we define a loss function which is parameterized with the weights. More specifically, the loss function is defined as a convex function of the difference of the predicted and real dependent variables (estimation error) and our aim is to find the weights that produce the global minimum of the loss function, i.e, the weights that result to the minimum estimation error. Common loss functions used in neural networks are the mean square error (MSE), mean absolute error (MAE), binary cross-entropy etc. The optimal weights are found using the gradient descent algorithm (and its variations) in the loss function for each data point in the input. An iteration through all the data points of the input is called epoch. In Figure 26 we can see the loss function of a neural network over 30 epochs.

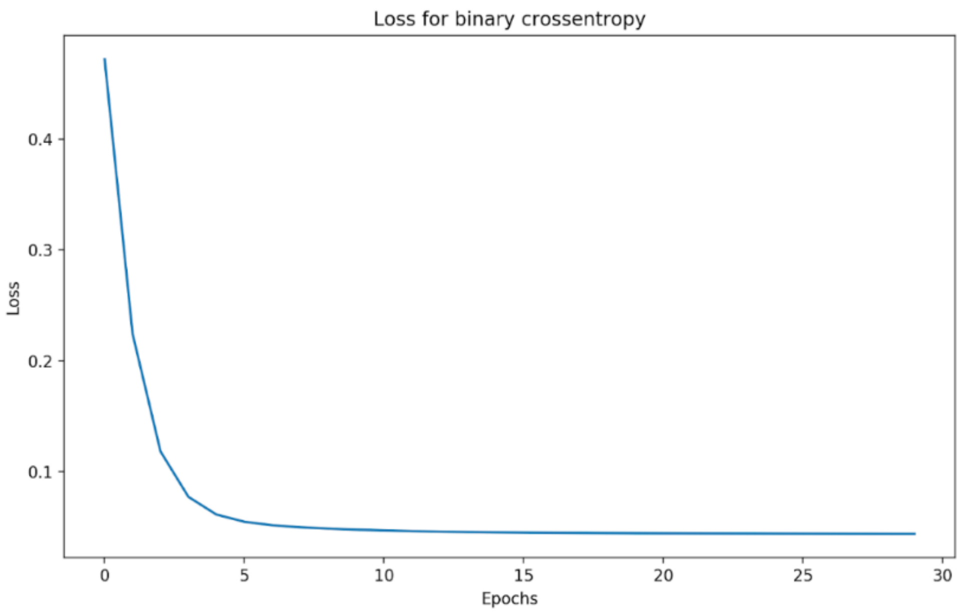


Figure 26. The Loss for Binary Cross-Entropy is Decreased as the Epochs Increase.

A neural network has a significant number of hyperparameters. To obtain the best results possible, we performed hyperparameter tuning with exhaustive search over a grid of parameters. The parameters tuned were:

1. *Batch size*: The number of samples that are used in one training pass of the network
2. *Number of epochs*: Number of times that the network will be trained with the total number of samples
3. *Optimization algorithm*: The algorithm used to train the network and minimize the loss function
4. *Learning rate and momentum of the optimization algorithm:* Parameters that affect the accuracy and convergence time of the optimization algorithm
5. *Neuron activation function*: The activation function used in each neuron in each layer
6. *Number of neurons in the hidden layer*: The number of neurons used in the hidden layer
7. *Number of hidden layers in the network*

We used a greedy approach to find the optimal setup for all the above parameters since the number of models to train increases exponentially with the number of parameters. More specifically, we used grid search to tune up to two parameters and subsequently used the best combination of these two parameters to tune the next one. The confusion matrix, accuracy score, and receiver operator characteristic (ROC) of the chosen model for the validation (OOT) set are below.

True Label	Predicted: Normal	Predicted: Fraud
Fraud	511	1,232
Normal	163,596	1,154

Figure 27. Confusion Matrix.

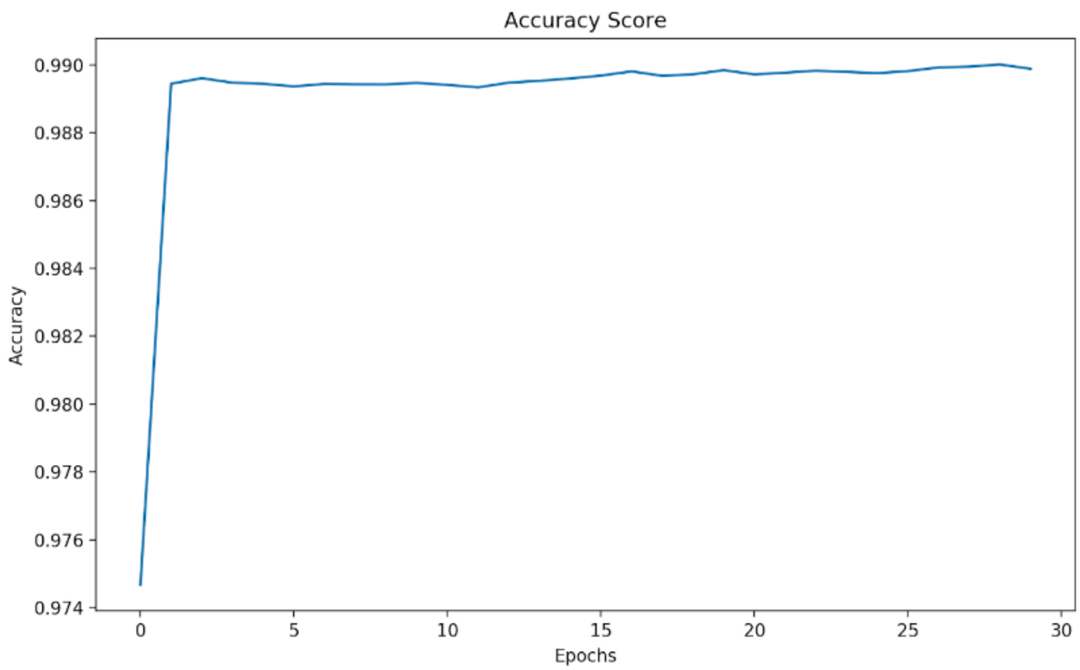


Figure 28. Accuracy Score.

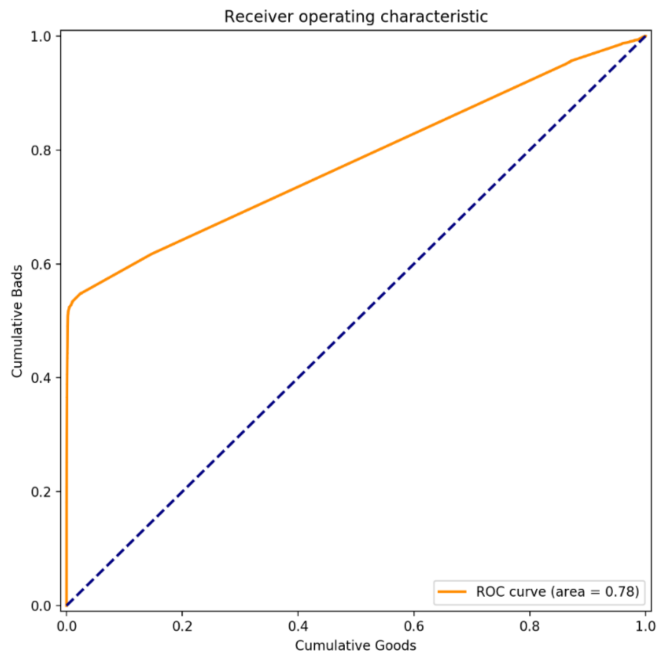


Figure 29. ROC Curve.

Finally, the results of the fraud detection with a neural network are presented in the next figure. We used different configurations of hidden layers and number of neurons and 3-fold cross-validation for more accurate results.

Model	Parameters			FDR at 3%		
Neural Network	Layer	Node	Epochs	TRAIN	TEST	OOT
1	1	15	30	57.03181	56.70786	54.77787
2	1	45	30	56.73536	56.29597	54.56832
3	1	75	50	56.82213	56.36952	54.65214
4	2	(45,20)	30	57.01374	56.66372	54.73596
5	2	(25,10)	50	57.10051	56.78141	54.77787
6	2	(25,10)	30	57.06435	56.76670	54.65214
7	3	(45,20,10)	30	57.05351	56.73728	54.77787
8	3	(20,45,10)	30	56.6667	56.42836	54.35876
9	4	(15,20,10,5)	50	57.02820	56.76670	54.77363
10	4	(25,15,10,5)	30	56.84382	56.34010	54.56832

Figure 30. Neural Network Results.

We notice from the above results that the train, test and validation (OOT) set FDRs do not change significantly with different configurations of neural networks and they are at 57%, 56%, and 54% respectively.

Random Forest

Random Forest is a statistical algorithm which is a slight improvement from a bagging algorithm. In a bagging algorithm, number of decision trees are made by taking bootstrapped samples of rows from the dataset involving all the possible number of predictors. However, this algorithm still contains the problem of higher variance from the decision trees. This is because, all the decision trees contain the most powerful predictor among all which induces high correlation between the predictions of the individual predictions of decision trees. As a result, the problem of high variance in the result persists. To overcome this problem, random forest does not involve all the predictors in making the decision trees. It randomly selects the number of predictors and subsequently makes the decision trees. This helps in making a sequence of uncorrelated trees where the most powerful predictor is not always present in every decision tree. This helps in reducing the variance and helps in getting rid of overfitting. Traditionally, we choose the number of predictors, m $=\sqrt{p}$, where p is the total number of predictors in the data. Figure 31 depicts the same thing where we can see that in the decision trees all the features are present whereas in the random forest, two decision trees are present with a subset of features selected from all the possible features present.

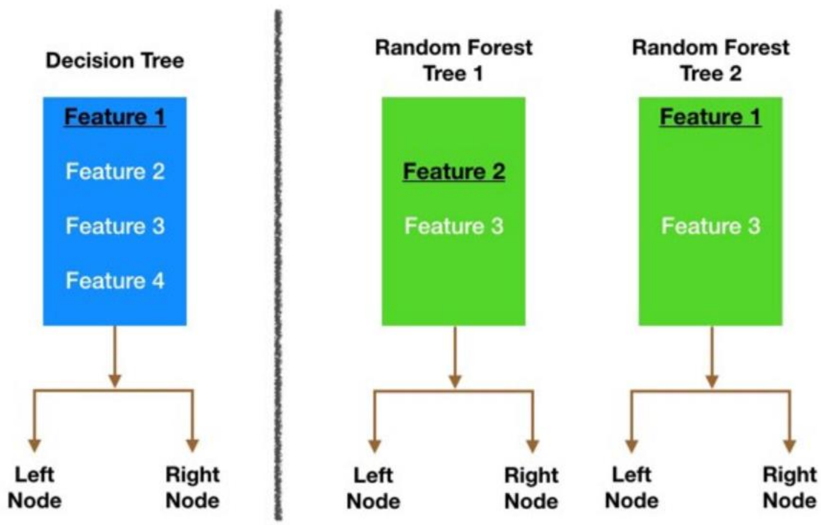


Figure 31. Decision Trees vs. Random Forest Trees

Random forest includes various hyperparameters which we can tune in order to improve our accuracy measure. Following is a short description of all the hyperparameters:

1) *Number of Estimators*: The number of trees that the algorithm will create, where the most common class predicted by each individual tree is the final prediction of the algorithm. Too many estimators in the model tends to create the problem of overfitting. The number should be chosen such that there is little or less improvement in the testing accuracy.
2) *Max_features*: The maximum number of features that the algorithm can choose to make each decision tree. Most common number of features to be selected are given by $\mathrm{m} = \sqrt{p}$, where p is the total number of predictors.
3) *Max_depth*: Represents the depth of each tree in the forest. Higher depth results in higher splits in each tree which may lead to overfitting.

4) *Min_samples_leaf*: Represents the minimum number of samples required to be at a leaf node.
5) *Min_samples_split*: Represents the minimum number of samples required to split an internal node.

Random Forest	No. of Variables	No. of Trees	No. of Features	Depth	TRAIN FDR	TEST FDR	OOT FDR
1	30	500	6	60	56.393	57.58	54.69
2	30	500	6	70	56.341	57.549	54.568
3	30	500	6	80	56.48	57.466	54.652
4	30	400	7	80	56.358	57.549	54.654
5	30	500	7	80	56.38	57.49	54.736
6	30	800	7	80	56.4	57.494	54.81
7	30	600	7	60	56.36	57.55	54.86
8	30	500	8	80	56.33	57.6	54.73
9	30	800	9	80	56.36	57.44	54.86
10	30	500	10	80	56.42	57.54	54.65
11	30	800	8	80	56.396	57.63	54.69
12	30	800	11	80	56.38	57.439	54.779

Figure 32: Random Forest Results.

Results

Figure 33 below shows the results from all the models that were tested to predict fraud.

Model	Parameter				Average FDR(%) at 3%		
Logistic Regression	Total Variables	# of Variables Selected			TRAIN	TEST	OOT
1	30	10			0.520346	0.507287	0.501258
2	30	20			0.547640	0.535296	0.518022
3	30	30			0.552899	0.543012	0.525985
4	25	5			0.463278	0.459560	0.452221
5	25	15			0.510079	0.499858	0.486589
6	25	25			0.549769	0.537582	0.530537
Boosted Tree	# of Variables	# of Trees	Max Depth	Learning Rate	TRAIN	TEST	OOT
1	30	600	4	0.01	0.57365	0.56179	0.54652
2	30	800	4	0.01	0.57352	0.5609	0.54819
3	30	1000	4	0.01	0.57389	0.56179	0.54819
4	30	600	5	0.01	0.57192	0.55971	0.5461
5	30	800	5	0.01	0.57389	0.5609	0.54861
6	30	1000	5	0.01	0.57438	0.56209	0.54903
7	30	600	4	0.001	0.55751	0.54634	0.53143
8	30	800	4	0.001	0.55899	0.54723	0.53269
9	30	1000	4	0.001	0.55899	0.54723	0.53269
10	30	600	5	0.001	0.55665	0.54367	0.53059
11	30	800	5	0.001	0.55849	0.54486	0.53185
10	30	1000	5	0.001	0.55825	0.54486	0.53185
Nueral Network	Layer	Node		Epoch	TRAIN	TEST	OOT
1	1	15		30	0.570318	0.567079	0.547779
2	1	45		30	0.567354	0.562960	0.545683
3	1	75		50	0.568221	0.563695	0.546521
4	2	(45,20)		30	0.570137	0.566637	0.547360
5	2	(25,10)		50	0.571005	0.567814	0.547779
6	2	(25,10)		30	0.570644	0.567667	0.546521
7	3	(45,20,10)		30	0.570535	0.567373	0.547779
8	3	(20,45,10)		30	0.566667	0.564284	0.543588
9	4	(15,20,10,5)		50	0.570282	0.567667	0.547736
10	4	(25,15,10,5)		30	0.568438	0.563401	0.545683
Random Forest	# of Variables	# of Trees	# of Features	Depth	TRAIN	TEST	OOT
1	30	500	6	60	0.563930	0.575800	0.546900
2	30	500	6	70	0.563410	0.575490	0.545680
3	30	500	6	80	0.564800	0.574660	0.546520
4	30	400	7	80	0.563580	0.575490	0.546540
5	30	500	7	80	0.563800	0.574900	0.547360
6	30	800	7	80	0.564000	0.574940	0.548100
7	30	600	7	60	0.563600	0.575500	0.548600
8	30	500	8	80	0.563300	0.576000	0.547300
9	30	800	9	80	0.563600	0.574400	0.548600
10	30	500	10	80	0.564200	0.575400	0.546500
11	30	800	8	80	0.563960	0.576300	0.546900
12	30	800	11	80	0.563800	0.574390	0.547790

Figure 33: Results of All Models

Final Model – Random Forest

After our initial results from training models with logistic regression, boosted trees (XGBoost), a random forest, and a neural network, we determined that our random forest model performed the best. The random forest model consistently outperformed the other models given its fraud detection rates for both the testing and out-of-time validation datasets (always above 57.4% for testing and always above 54.6% for OOT).

Hyperparameter Selection

In order to determine the optimum selection of hyperparameters, we used the GridSearchCV technique to determine the maximum accuracy for a set of hyperparameters. The following snippet shows the grid of the parameters that were fit one by one with every possible combination:

```
start = pd.datetime.now()
# Create the parameter grid based on the results of random search
param_grid = {
    'bootstrap': [True],
    'max_depth': [80, 90, 100, 110],
    'max_features': [6, 7,8,9],
    'n_estimators': [300,400,500,600,700,800]
}
# Create a based model
rf = RandomForestClassifier()
# Instantiate the grid search model
grid_search = GridSearchCV(estimator = rf, param_grid = param_grid,
                          cv = 2, n_jobs = -1, verbose = 2,scoring='neg_mean_squared_error')
print(pd.datetime.now()-start)
```

The best set was selected with the maximum neg_mean_squared_error. The best set contains:

N_estimators = 500, max_features = 7, max_depth = 80

After getting the best parameters, we ran our model and calculated the FDR at 3% which came out to be 54.73 %. However, we tried other sets of parameters which were closer to the best set and got a better FDR at 3% which was 54.86%. The following is the set of parameters which resulted in higher FDR:

N_estimators = 600, max_features = 7, max_depth = 60

The following is a list of results from various combinations of hyperparameters that we achieved:

Random Forest	No. of Variables	No. of Trees	No. of Features	Depth	TRAIN FDR	TEST FDR	OOT FDR
1	30	500	6	60	56.393	57.58	54.69
2	30	500	6	70	56.341	57.549	54.568
3	30	500	6	80	56.48	57.466	54.652
4	30	400	7	80	56.358	57.549	54.654
5	30	500	7	80	56.38	57.49	54.736
6	30	800	7	80	56.4	57.494	54.81
7	30	600	7	60	56.36	57.55	54.86
8	30	500	8	80	56.33	57.6	54.73
9	30	800	8	80	56.396	57.63	54.69
10	30	800	9	80	56.36	57.44	54.86
11	30	500	10	80	56.42	57.54	54.65
12	30	800	11	80	56.38	57.439	54.779

Figure 34. Random Forest Results for Various Hyperparameters

The following results are the in-depth analysis of the final Random Forest model for Training, testing, and Out-of time datasets:

Training	**# Records**	**# Goods**	**# Bads**	**Fraud Rate**								
	583454	575063	8391	0.0146								
Bin %	# Records	# Goods	# Bads	% Goods	% Bads	Total # Records	Cumulative Goods	Cumulative Bads	% Goods	% Bads (FDR)	KS	FPR
1	5835	1359	4476	23.29%	76.71%	5835	1359	4476	0.24%	53.34%	0.53	0.30
2	5835	5652	183	96.86%	3.14%	11670	7011	4659	1.22%	55.52%	0.54	1.50
3	5835	5759	76	98.70%	1.30%	17505	12770	4735	2.22%	56.43%	0.54	2.70
4	5835	5787	48	99.18%	0.82%	23340	18557	4783	3.23%	57.00%	0.54	3.88
5	5835	5795	40	99.31%	0.69%	29175	24352	4823	4.23%	57.48%	0.53	5.05
6	5835	5794	41	99.30%	0.70%	35010	30146	4864	5.24%	57.97%	0.53	6.20
7	5835	5801	34	99.42%	0.58%	40845	35947	4898	6.25%	58.37%	0.52	7.34
8	5835	5782	53	99.09%	0.91%	46680	41729	4951	7.26%	59.00%	0.52	8.43
9	5835	5795	40	99.31%	0.69%	52515	47524	4991	8.26%	59.48%	0.51	9.52
10	5835	5796	39	99.33%	0.67%	58350	53320	5030	9.27%	59.95%	0.51	10.60
11	5835	5780	55	99.06%	0.94%	64185	59100	5085	10.28%	60.60%	0.50	11.62
12	5835	5787	48	99.18%	0.82%	70020	64887	5133	11.28%	61.17%	0.50	12.64
13	5835	5783	52	99.11%	0.89%	75855	70670	5185	12.29%	61.79%	0.50	13.63
14	5835	5788	47	99.19%	0.81%	81690	76458	5232	13.30%	62.35%	0.49	14.61
15	5835	5791	44	99.25%	0.75%	87525	82249	5276	14.30%	62.88%	0.49	15.59
16	5835	5791	44	99.25%	0.75%	93360	88040	5320	15.31%	63.40%	0.48	16.55
17	5835	5791	44	99.25%	0.75%	99195	93831	5364	16.32%	63.93%	0.48	17.49
18	5835	5791	44	99.25%	0.75%	105030	99622	5408	17.32%	64.45%	0.47	18.42
19	5835	5799	36	99.38%	0.62%	110865	105421	5444	18.33%	64.88%	0.47	19.36
20	5835	5797	38	99.35%	0.65%	116700	111218	5482	19.34%	65.33%	0.46	20.29

Figure 35. Random Forest – Best Model Training Set Results

Testing	# Records	# Goods	# Bads	Fraud Rate								
	250053	246437	3616	0.0147								
Bin %	# Records	# Goods	# Bads	% Goods	% Bads	Total # Records	Cumulative Goods	Cumulative Bads	% Goods	% Bads (FDR)	KS	FPR
1	2501	564	1937	22.55%	77.45%	2501	564	1937	0.34%	53.57%	0.53	0.29
2	2501	2398	103	95.88%	4.12%	5002	2962	2040	1.80%	56.42%	0.55	1.45
3	2501	2461	40	98.40%	1.60%	7503	5423	2080	3.30%	57.52%	0.54	2.61
4	2501	2471	30	98.80%	1.20%	10004	7894	2110	4.81%	58.35%	0.54	3.74
5	2501	2482	19	99.24%	0.76%	12505	10376	2129	6.32%	58.88%	0.53	4.87
6	2501	2481	20	99.20%	0.80%	15006	12857	2149	7.83%	59.43%	0.52	5.98
7	2501	2485	16	99.36%	0.64%	17507	15342	2165	9.35%	59.87%	0.51	7.09
8	2501	2480	21	99.16%	0.84%	20008	17822	2186	10.86%	60.45%	0.50	8.15
9	2501	2476	25	99.00%	1.00%	22509	20298	2211	12.37%	61.14%	0.49	9.18
10	2501	2480	21	99.16%	0.84%	25010	22778	2232	13.88%	61.73%	0.48	10.21
11	2501	2490	11	99.56%	0.44%	27511	25268	2243	15.40%	62.03%	0.47	11.27
12	2501	2483	18	99.28%	0.72%	30012	27751	2261	16.91%	62.53%	0.46	12.27
13	2501	2487	14	99.44%	0.56%	32513	30238	2275	18.43%	62.91%	0.44	13.29
14	2501	2488	13	99.48%	0.52%	35014	32726	2288	19.94%	63.27%	0.43	14.30
15	2501	2479	22	99.12%	0.88%	37515	35205	2310	21.45%	63.88%	0.42	15.24
16	2501	2487	14	99.44%	0.56%	40016	37692	2324	22.97%	64.27%	0.41	16.22
17	2501	2487	14	99.44%	0.56%	42517	40179	2338	24.48%	64.66%	0.40	17.19
18	2501	2486	15	99.40%	0.60%	45018	42665	2353	26.00%	65.07%	0.39	18.13
19	2501	2489	12	99.52%	0.48%	47519	45154	2365	27.51%	65.40%	0.38	19.09
20	2501	2486	15	99.40%	0.60%	50020	47640	2380	29.03%	65.82%	0.37	20.02

Figure 36. Random Forest – Best Model Testing Set Results

OOT	# Records	# Goods	# Bads	Fraud Rate									
	166493	164107	2386	0.0145									
Bin %	# Records	# Goods	# Bads	% Goods	% Bads	Total # Records	Cumulative Goods	Cumulative Bads	% Goods	% Bads (FDR)	KS	FPR
1	1665	435	1230	26.13%	73.87%	1665	435	1230	0.27%	51.55%	0.51	0.35
2	1665	1614	51	96.94%	3.06%	3330	6087	1281	3.71%	53.69%	0.50	4.75
3	1665	1637	28	98.32%	1.68%	4995	7724	1309	4.71%	54.86%	0.50	5.90
4	1665	1648	17	98.98%	1.02%	6660	9372	1326	5.71%	55.57%	0.50	7.07
5	1665	1653	12	99.28%	0.72%	8325	11025	1338	6.72%	56.08%	0.49	8.24
6	1665	1651	14	99.16%	0.84%	9990	12676	1352	7.72%	56.66%	0.49	9.38
7	1665	1655	10	99.40%	0.60%	11655	14331	1362	8.73%	57.08%	0.48	10.52
8	1665	1642	23	98.62%	1.38%	13320	15973	1385	9.73%	58.05%	0.48	11.53
9	1665	1651	14	99.16%	0.84%	14985	17624	1399	10.74%	58.63%	0.48	12.60
10	1665	1654	11	99.34%	0.66%	16650	19278	1410	11.75%	59.09%	0.47	13.67
11	1665	1651	14	99.16%	0.84%	18315	20929	1424	12.75%	59.68%	0.47	14.70
12	1665	1652	13	99.22%	0.78%	19980	22581	1437	13.76%	60.23%	0.46	15.71
13	1665	1654	11	99.34%	0.66%	21645	24235	1448	14.77%	60.69%	0.46	16.74
14	1665	1646	19	98.86%	1.14%	23310	25881	1467	15.77%	61.48%	0.46	17.64
15	1665	1656	9	99.46%	0.54%	24975	27537	1476	16.78%	61.86%	0.45	18.66
16	1665	1653	12	99.28%	0.72%	26640	29190	1488	17.79%	62.36%	0.45	19.62
17	1665	1651	14	99.16%	0.84%	28305	30841	1502	18.79%	62.95%	0.44	20.53
18	1665	1654	11	99.34%	0.66%	29970	32495	1513	19.80%	63.41%	0.44	21.48
19	1665	1659	6	99.64%	0.36%	31635	34154	1519	20.81%	63.66%	0.43	22.48
20	1665	1655	10	99.40%	0.60%	33300	35809	1529	21.82%	64.08%	0.42	23.42

Figure 37. Random Forest – Best Model OOT Validation Set Results

Conclusion

A comprehensive analysis of application (identity) fraud cases was performed. First, the data was cleaned and all frivolous variables were updated to match their record numbers. Then, over 600 candidate variables were created and feature selection was performed (filter and wrapper methods) to pick the best variables. The variables were then used in several models: logistic regression, boosted trees, random forest, and neural network. Our best model to predict fraud was random forest which resulted in a 57% FDR at 3% for the testing dataset and a 54% FDR at 3% for the OOT dataset.

Given additional time and computer resources, there are a few items that we would investigate further in future iterations of this process. First, we would consult subject matter experts regarding reasonable links in the variable building stage. In trying to create as many variable combinations as possible, we created over 600 variables. We did begin with two links provided by our expert, but it would have been helpful to have gotten information on additional important links. Second, this expert check would also hold true and be helpful during the variable selection process. The expert’s domain expertise would ensure we did not exclude important variables. Third, we would spend more time evaluating each of the models with additional parameters and additional values for each of the parameters. Fourth, in the future, we would compare and contrast L1 and L2 regularization methods to reduce model complexity during both the wrapper and model building stage in an attempt to get better results. Lastly, we would like to test ensemble and stacking models in an attempt to get better results.

Appendix A: Data Quality Report (DQR)

Data Overview

Description: The data analysis contained in this report is from a synthetic dataset originally created for academic organizations that were conducting research in collaboration with ID Analytics (https://www.idanalytics.com/). The dataset is of product application data (e.g. credit card or cell phone application data) that reflects the statistical qualities and characteristics of true application data. The distribution of the data fields and the linkage properties in the dataset are therefore representative of realistic US product application data. Lastly, the dataset lends itself well to binary classification analysis since each application record contains a binary label.

Data Source: Professor Stephen Coggeshall and ID Analytics.

Data Time Period: January 1, 2016 to December 31, 2016. Note that although the 2016 calendar year was a leap year, there are no records for February 29, 2016.

Number of Data Fields: 10

Number of Records: 1,000,000

Number of Records Labeled "0": 985,607

Number of Records Labeled "1": 14,393

Name of Data File: "application data.csv"

Size of Data File: 83 MB

Summary Table

The table below provides summary information for the 10 data fields in the dataset. The data fields are listed in the order in which they appear.

Data Field	Num Records w/ a Value	Percent Populated	Num Unique Values	Most Common Value
record	1,000,000	100%	1,000,000	Not applicable
date	1,000,000	100%	365	20160816
ssn	1,000,000	100%	835,819	999999999
firstname	1,000,000	100%	78,136	EAMSTRMT
lastname	1,000,000	100%	177,001	ERJSAXA
address	1,000,000	100%	828,774	123 MAIN ST
zip5	1,000,000	100%	26,370	68138
dob	1,000,000	100%	42,673	19070626
homephone	1,000,000	100%	28,244	9999999999
fraud_label	1,000,000	100%	2	0

Data Field Exploration

The subsections below provide additional detailed information about each data field within the dataset. The data fields are described in the order in which they appear.

Field 1: record

Description: A categorical data field containing an integer representing the unique application record number identifier from 1 to 1,000,000. All application records in the dataset contain a record number.

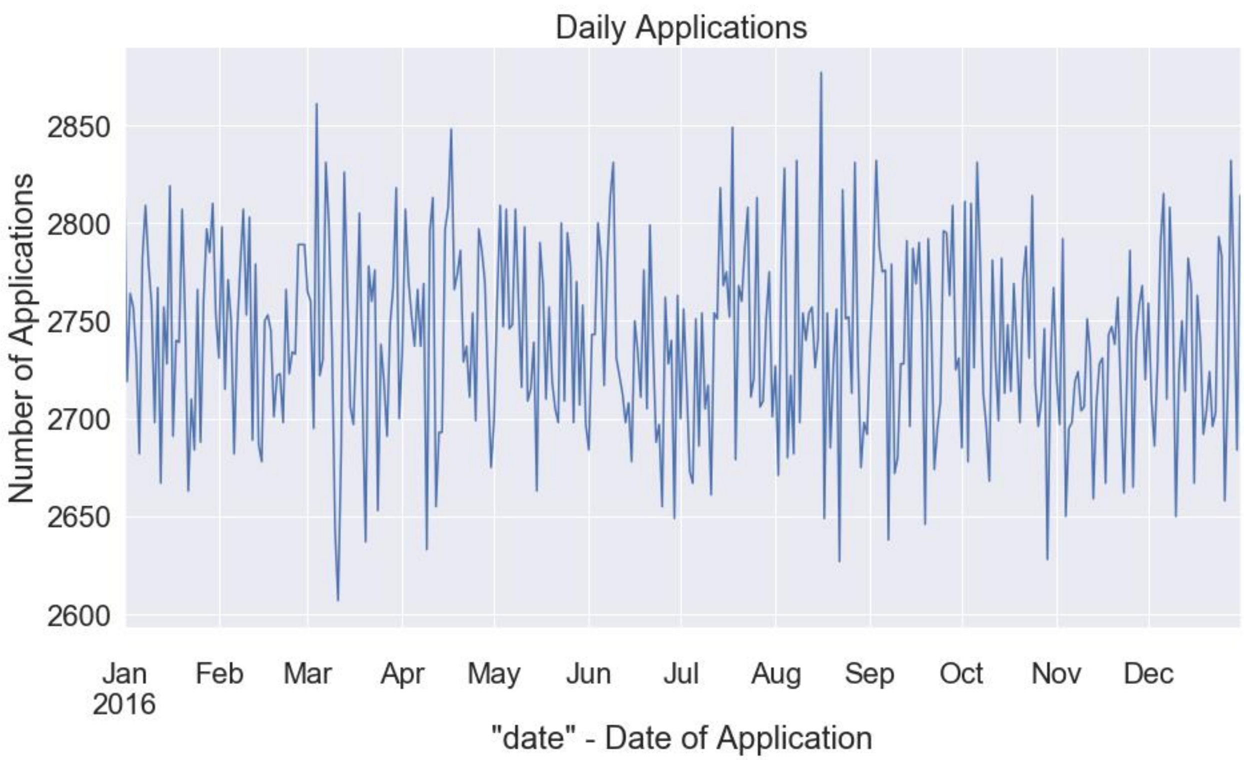
Field 2: date

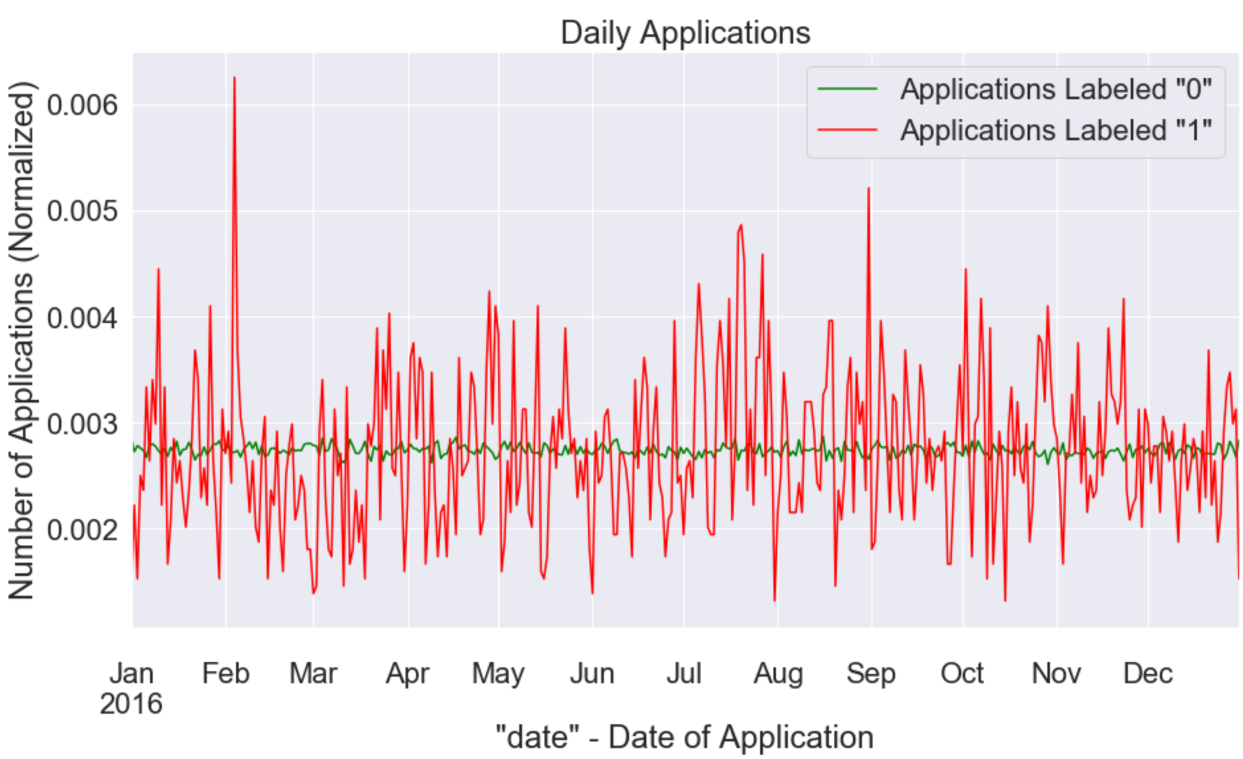
Description: A data field containing the date of the application with a format of YYYYMMDD. There are 365 unique values for this data field. Thus, despite the 2016 calendar year being a leap year, there are no records for February 29, 2016.

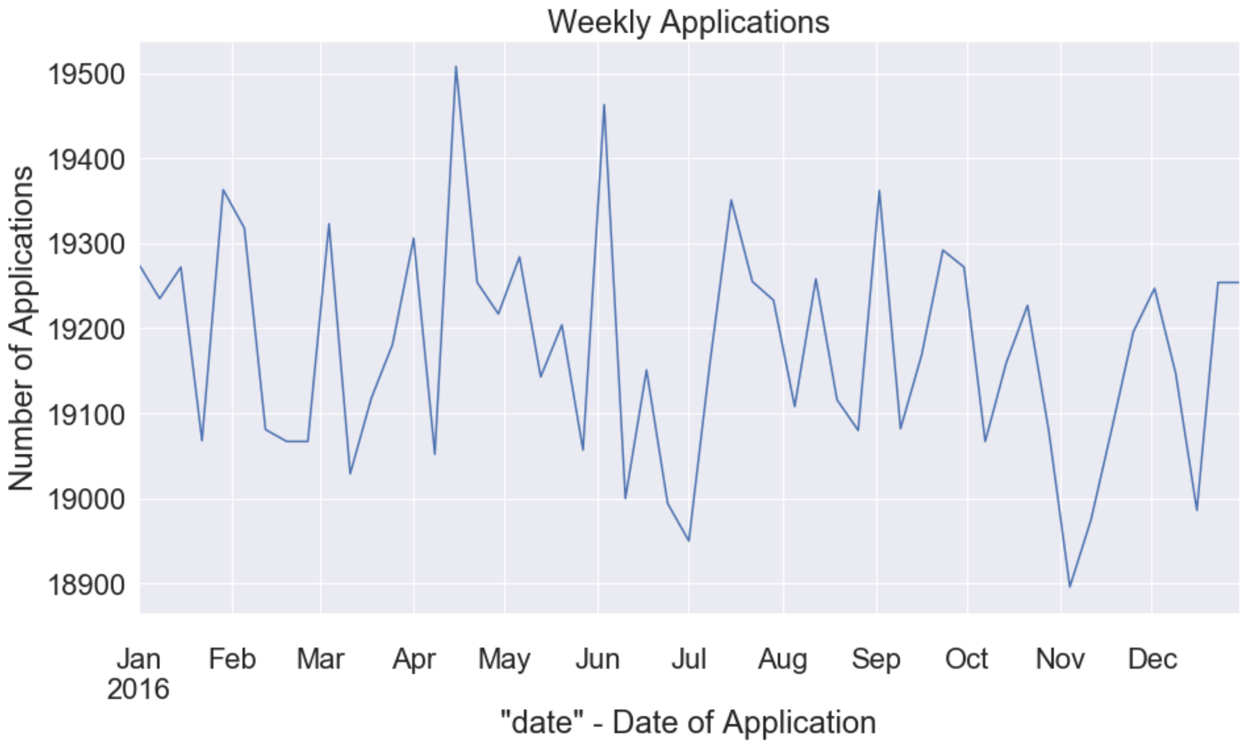
Before showing the daily and weekly application trends, the table below provides a quick overview of the top 10 days with the highest number of applications. Some of those days happen to coincide with US holidays or significant events. However, the variation is minor for these top 10 days when looking at the total number of applications.

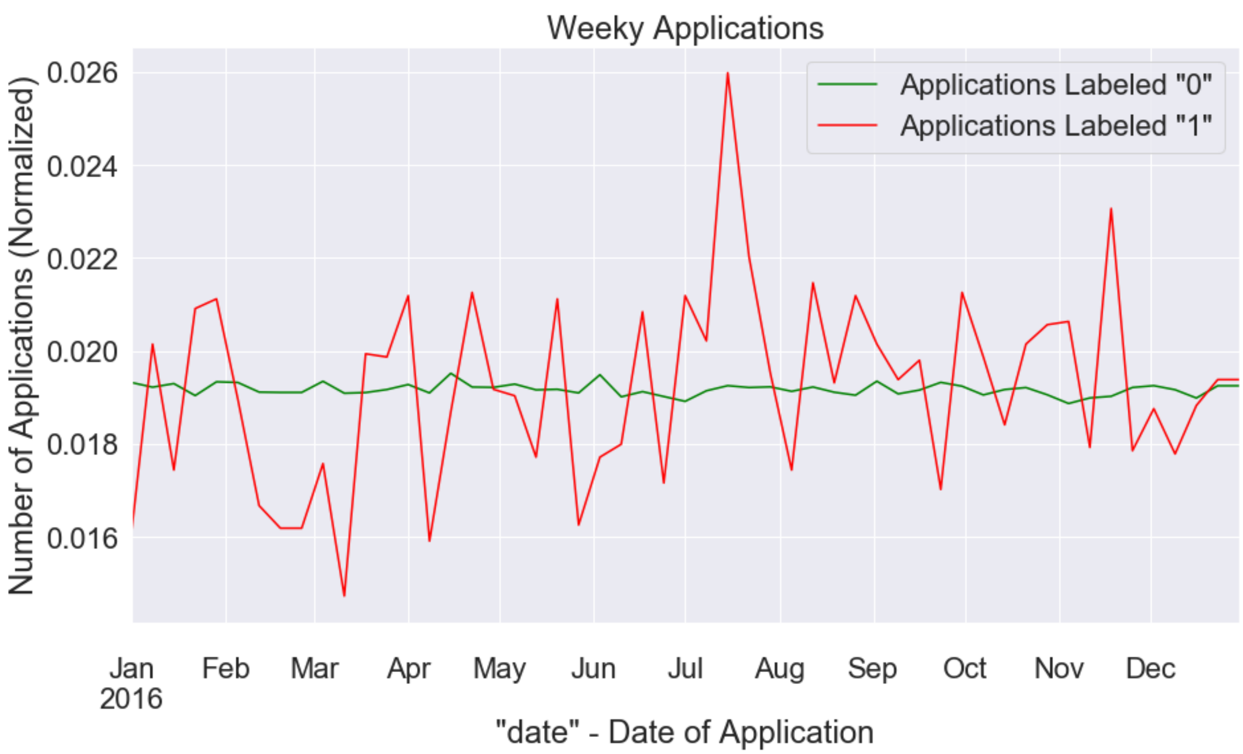
Date	Number of Applications	Notes on US Holiday / Event
August 16, 2016	2,877	Back-to-School sales timeframe
March 4, 2016	2,861	
July 18, 2016	2,849	
January 1, 2016	2,848	New Year’s Day
August 8, 2016	2,840	Back-to-School sales timeframe
December 28, 2016	2,832	Christmas and New Year’s holidays
September 3, 2016	2,832	Labor Day Weekend
June 9, 2016	2,831	Around end of school year
October 6, 2016	2,831	
March 7, 2016	2,831	

Finally, the graphs depicted below show the daily and weekly application trends over the 2016 calendar year. For both sets of graphs, the first graph contains the trend of applications as a whole (blue), while the second graph contains the normalized trend of applications where the applications are depicted based on their “fraud_label” data field values of “1” (red) or “0” (green).


Daily Applications
Number of Applications
2850
2800
2750
2700
2650
2600
Jan 2016
Feb
Mar
Apr
May
Jun
Jul
Aug
Sep
Oct
Nov
Dec
"date" - Date of Application


Daily Applications
Applications Labeled "0"
Applications Labeled "1"
Number of Applications (Normalized)
0.006
0.005
0.004
0.003
0.002
Jan 2016
Feb
Mar
Apr
May
Jun
Jul
Aug
Sep
Oct
Nov
Dec
"date" - Date of Application

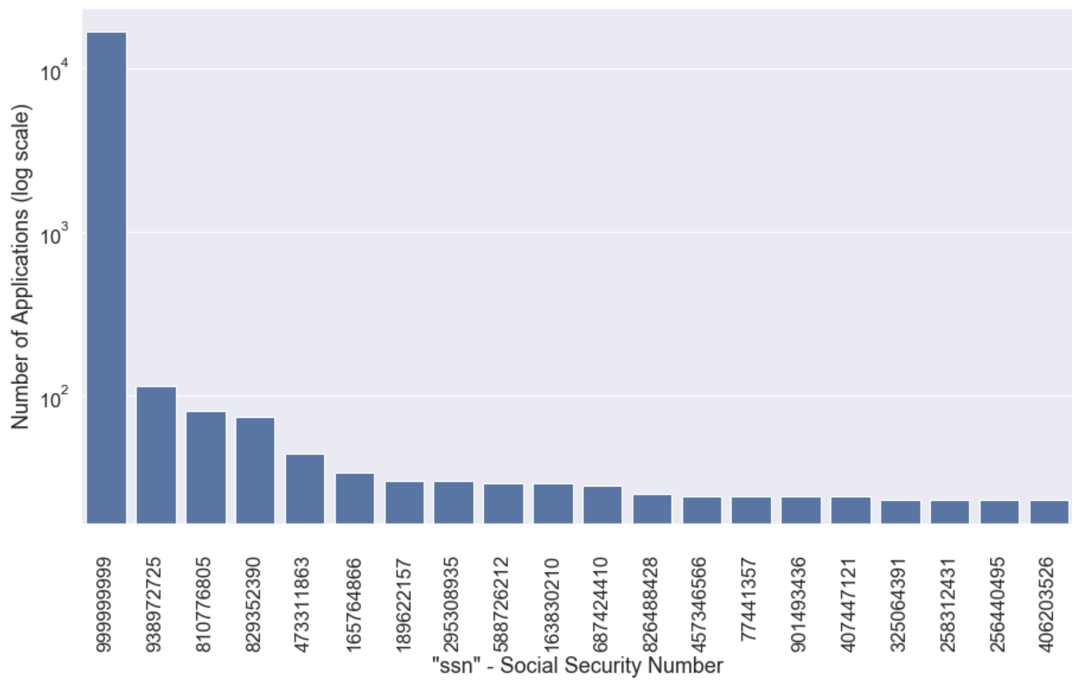

Weekly Applications
Number of Applications
19500
19400
19300
19200
19100
19000
18900
Jan
2016
Feb
Mar
Apr
May
Jun
Jul
Aug
Sep
Oct
Nov
Dec
"date" - Date of Application

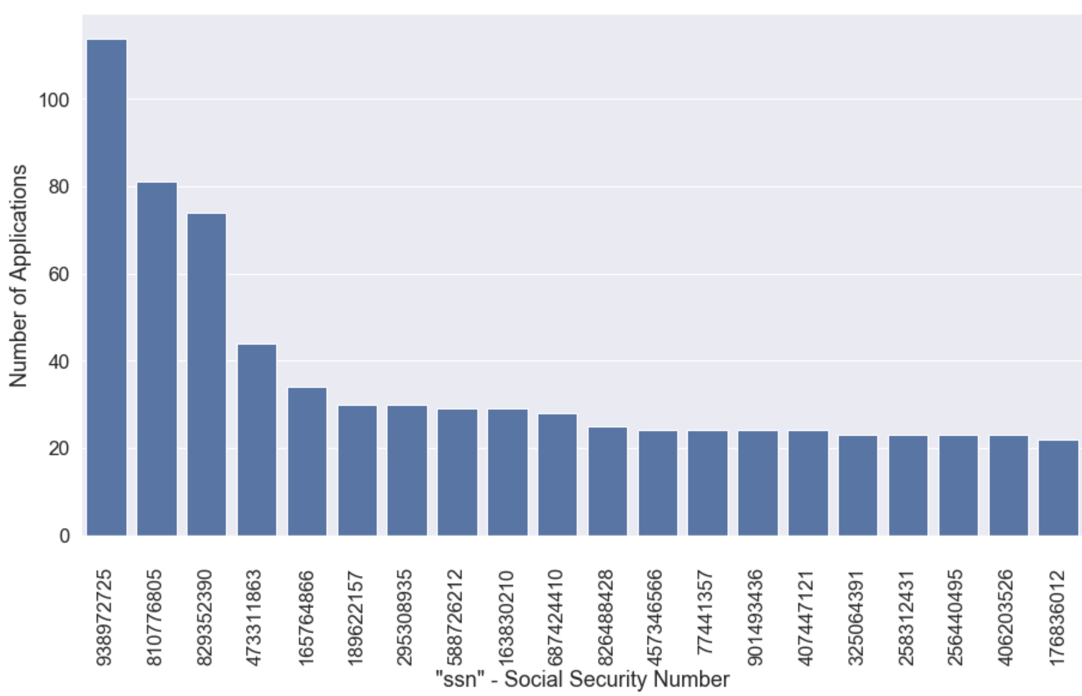

Weeky Applications
Number of Applications (Normalized)
0.026
0.024
0.022
0.020
0.018
0.016
Applications Labeled "0"
Applications Labeled "1"
Jan
2016
Feb
Mar
Apr
May
Jun
Jul
Aug
Sep
Oct
Nov
Dec
"date" - Date of Application

Field 3: ssn

Description: A 9-digit categorical data field containing the social security number (SSN) of the applicant. In absence of an SSN, a series of 9's was entered as the applicant's SSN. Also, for SSN entries that have less than 9 digits, those entries have a leading zero(s).

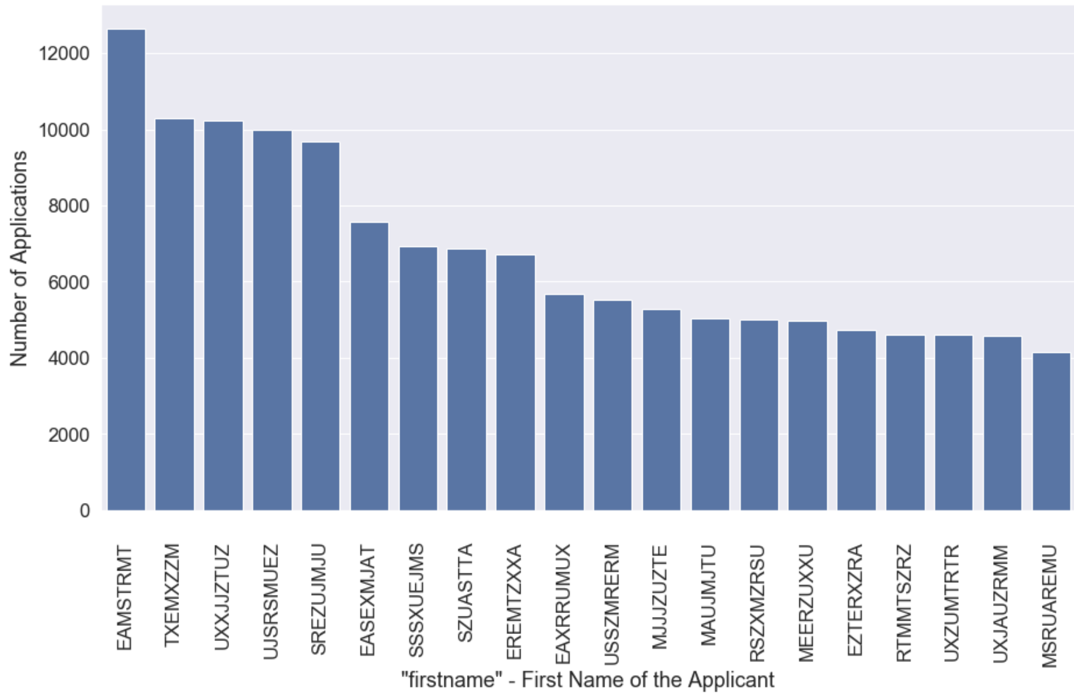
The bar charts below show the top 20 "ssn" data field values. There are 16,935 records with a value of "999999999" in this data field. Since there are so many records with a "999999999" value, we show the first bar chart with the "999999999" value and a second bar chart without the "999999999" value.





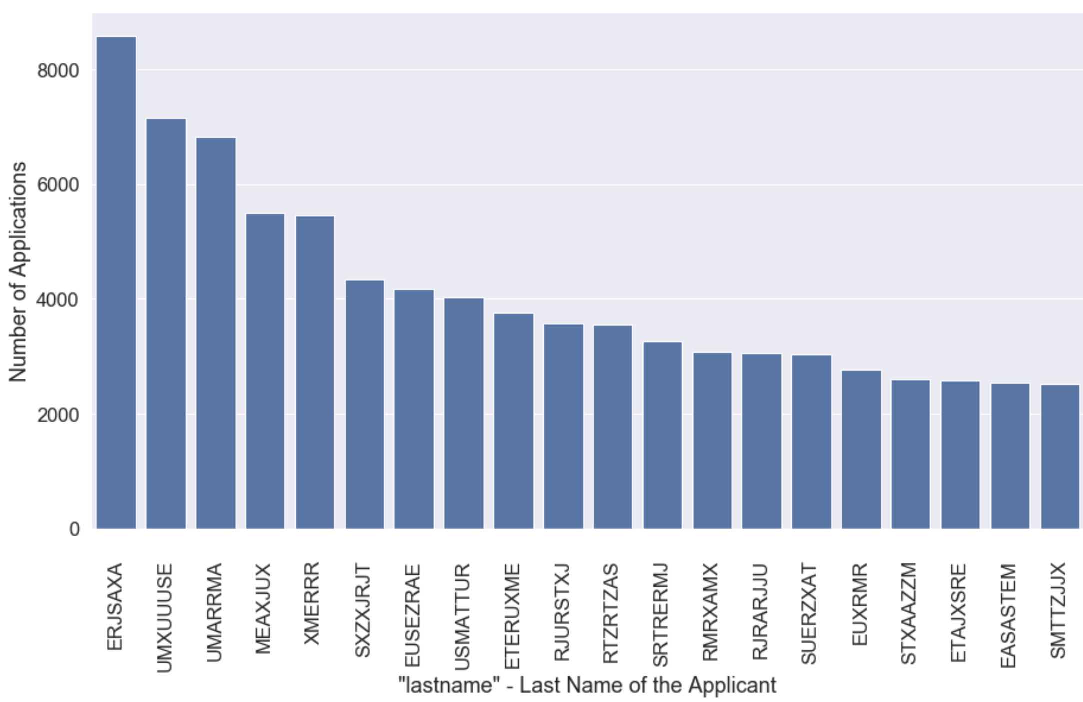
Field 4: firstname

Description: A categorical data field containing the first name of the applicant. The bar chart below provides the top 20 first names used for the applicant in the dataset.



Field 5: lastname

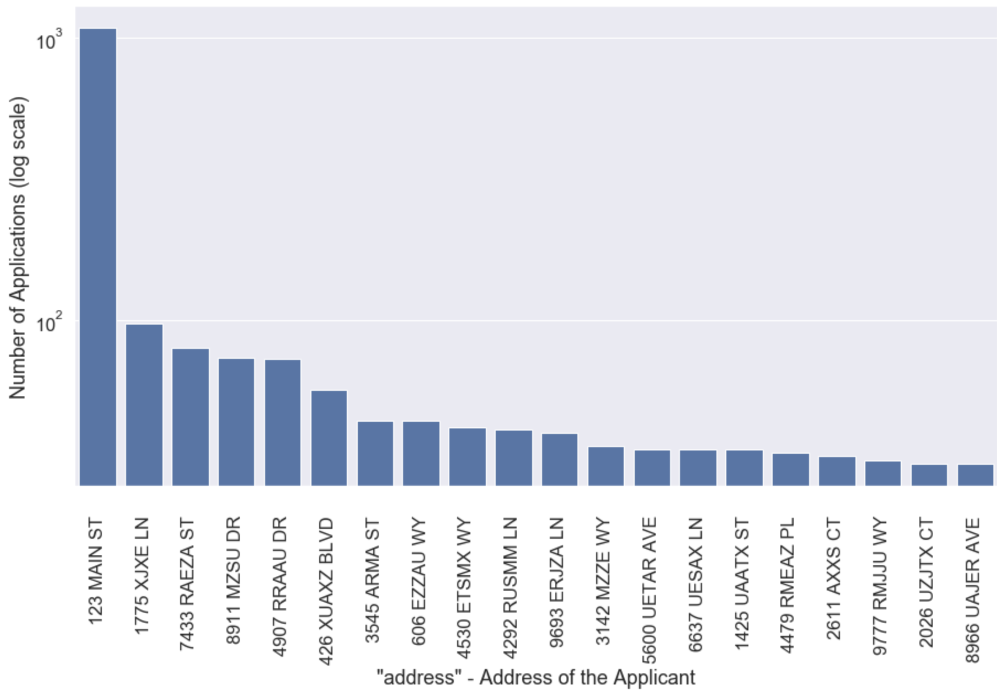
Description: A categorical data field containing the last name of the applicant. The bar chart below provides the top 20 last names used for the applicant in the dataset.

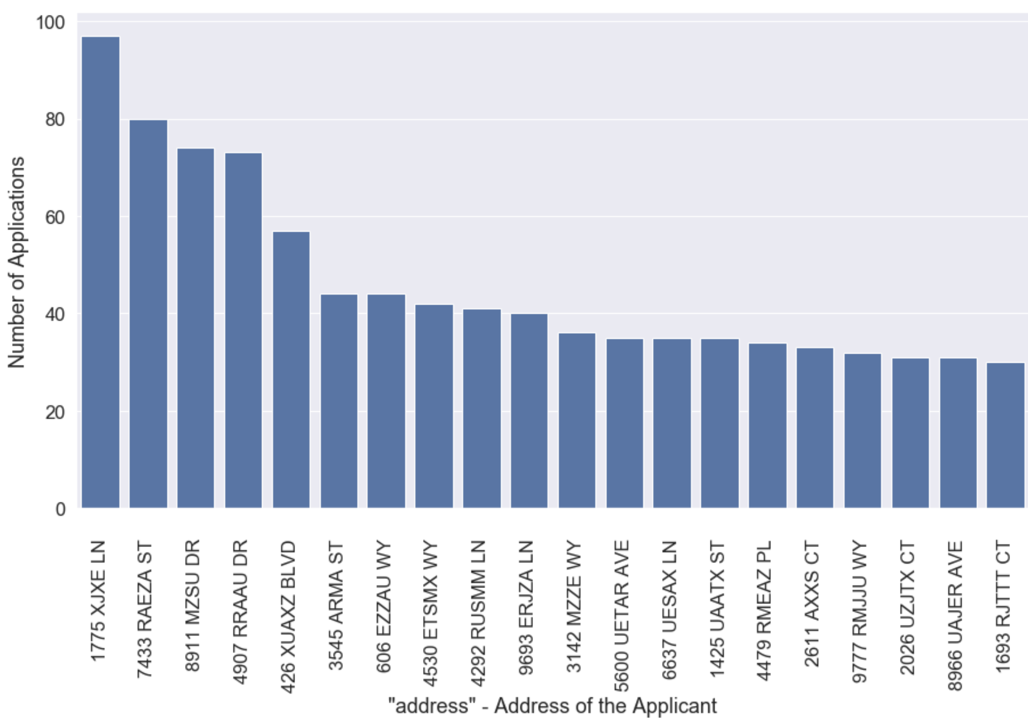


Field 6: address

Description: A categorical data field containing the applicant's address. There are 1,079 records with an entry of "123 MAIN ST" as the address for an applicant.

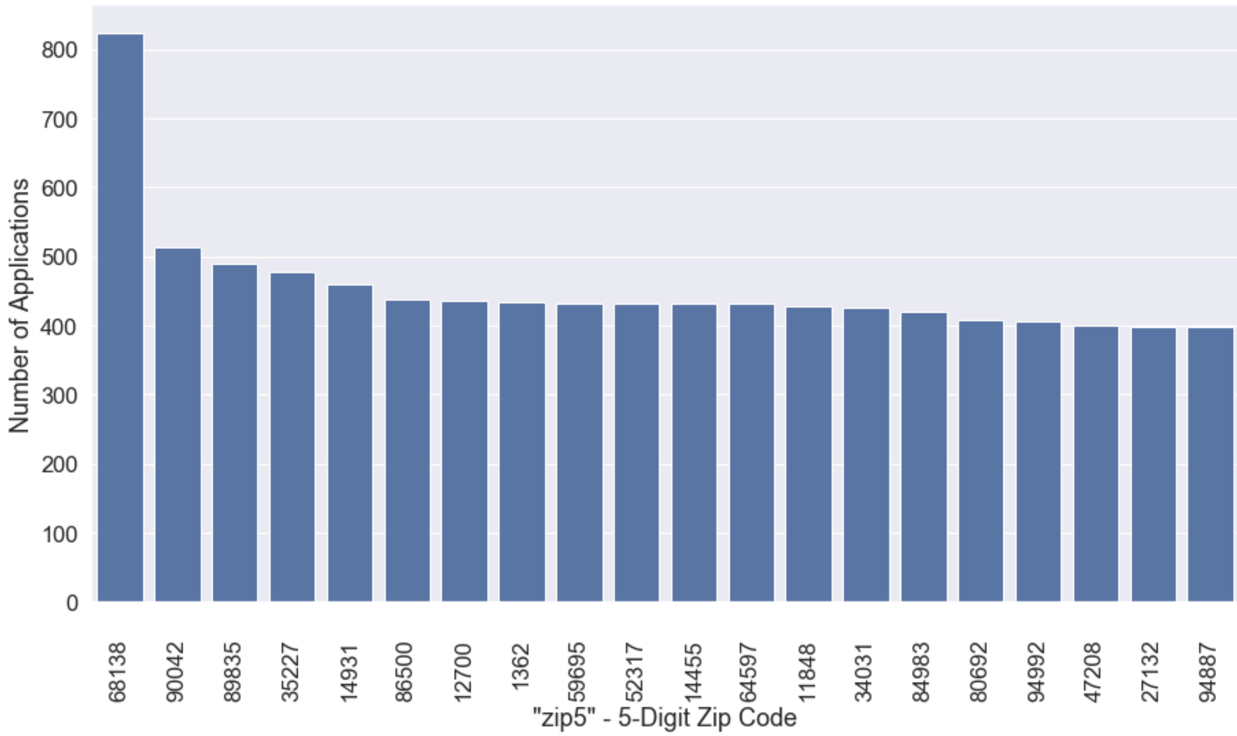
The bar charts below show the top 20 address values. Since there are so many records with "123 MAIN ST" as the address, we show the first bar chart with the "123 MAIN ST" address and the second bar chart without the "123 MAIN ST" address.





Field 7: zip5

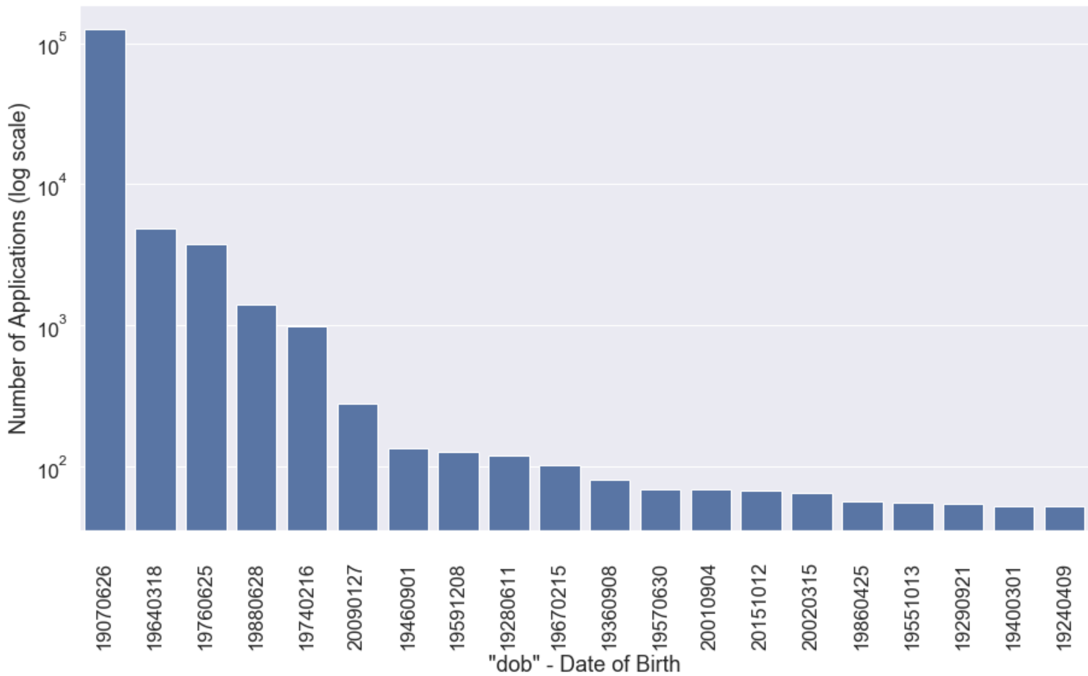
Description: A 5-digit categorical data field containing the zip code for the applicant's address. The bar chart below shows the top 20 zip codes used for the applicant's address. For those zip code entries that have less than 5 digits, those entries have a leading zero(s).

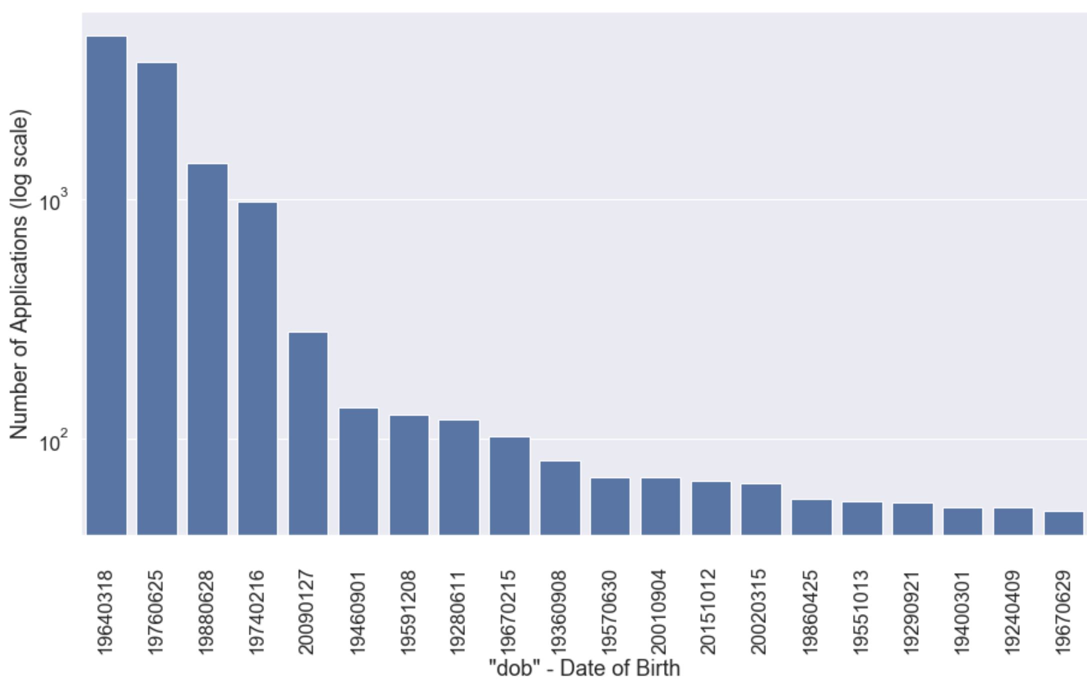


Field 8: dob

Description: An 8-digit categorical data field for the applicant's date of birth with a format of YYYYMMDD. There are 126,568 records with an entry of "19070626" (June 26, 1907) as the applicant's date of birth.

The bar charts below show the top 20 "dob" data field values. Since there are so many records with "19070626" as the date of birth, we show the first bar chart with the "19070626" value and the second bar chart without the "19070626" value.

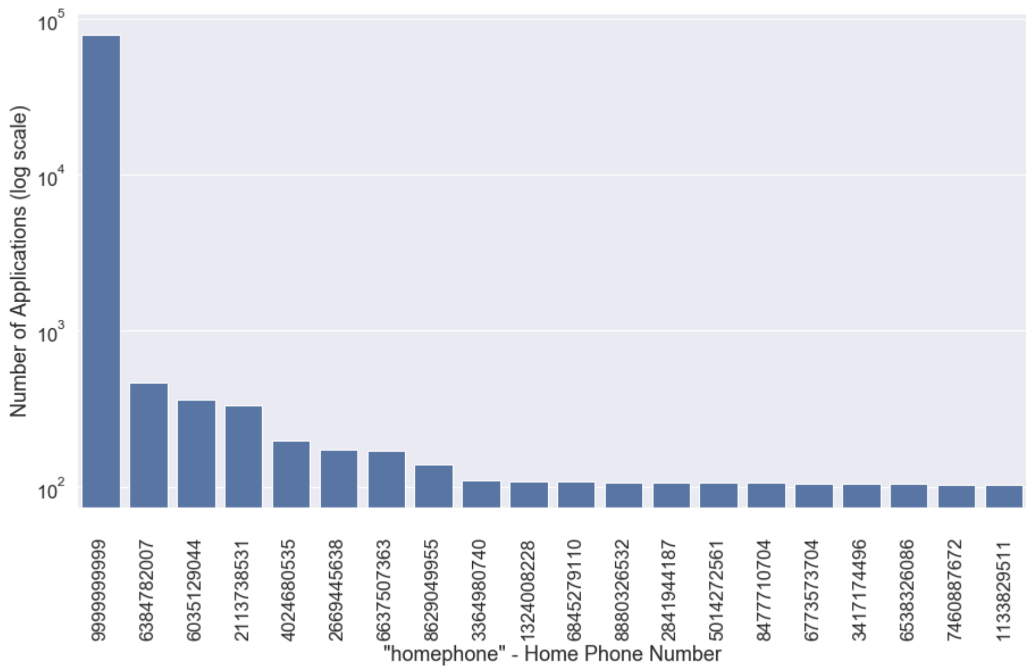


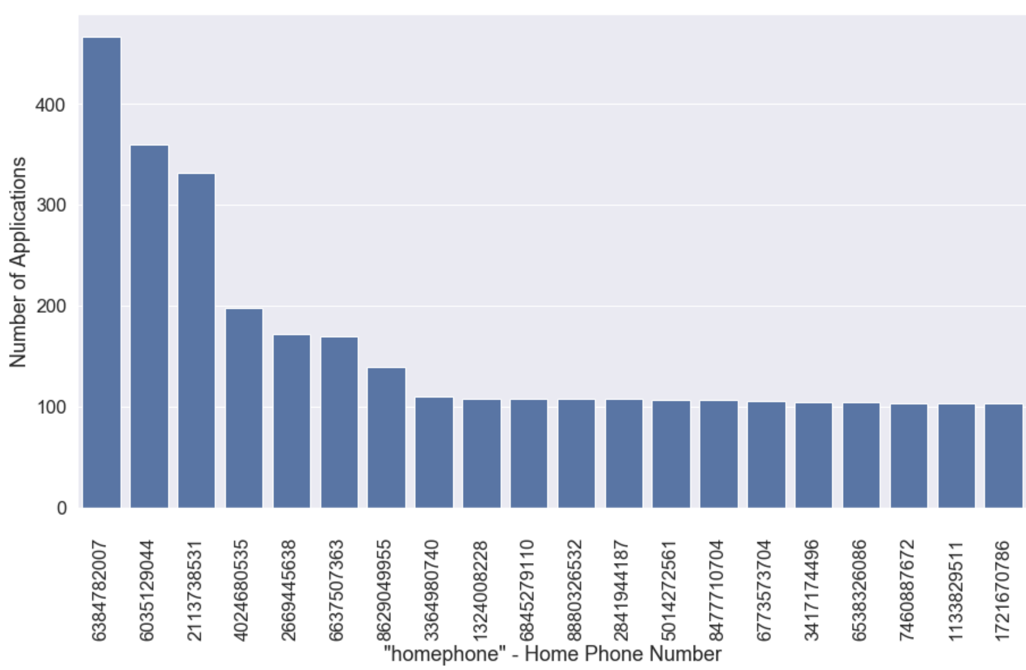


Field 9: homephone

Description: A 10-digit categorical data field containing the applicant's home phone number. In absence of a home phone number, a series of 9's was entered as the value. Also, for those entries that have less than 10 digits, those entries have a leading zero(s).

The bar charts below show the top 20 "homephone" data field values. There are 78,512 records with "9999999999" as the home phone number. Since there are so many records with "9999999999" as the home phone number, we show the first bar chart with the "9999999999" value and the second bar chart without the "9999999999" value.





Field 10: fraud_label

Description: A binary data field used to label the application record as either a zero or one. There is a total of 985,607 applications labeled as "0" and 14,393 applications labeled as "1" in the dataset.

Value of fraud_label	Number of Applications
0	985,607
1	14,393

Appendix B: Candidate Variables

Velocity Candidate Variables		Velocity Candidate Variables	
1	**ssn_velocity0_date**	125	**ssn-zip_velocity14_date**
2	**ssn_velocity1_date**	126	**ssn-zip_velocity30_date**
3	**ssn_velocity3_date**	127	**ssn-zip_velocity90_date**
4	**ssn_velocity7_date**	128	**ssn-zip_velocity180_date**
5	**ssn_velocity14_date**	129	**ssn-dob_velocity0_date**
6	**ssn_velocity30_date**	130	**ssn-dob_velocity1_date**
7	**ssn_velocity90_date**	131	**ssn-dob_velocity3_date**
8	**ssn_velocity180_date**	132	**ssn-dob_velocity7_date**
9	**address_velocity0_date**	133	**ssn-dob_velocity14_date**
10	**address_velocity1_date**	134	**ssn-dob_velocity30_date**
11	**address_velocity3_date**	135	**ssn-dob_velocity90_date**
12	**address_velocity7_date**	136	**ssn-dob_velocity180_date**
13	**address_velocity14_date**	137	**ssn-homephone_velocity0_date**
14	**address_velocity30_date**	138	**ssn-homephone_velocity1_date**
15	**address_velocity90_date**	139	**ssn-homephone_velocity3_date**
16	**address_velocity180_date**	140	**ssn-homephone_velocity7_date**
17	**dob_velocity0_date**	141	**ssn-homephone_velocity14_date**
18	**dob_velocity1_date**	142	**ssn-homephone_velocity30_date**
19	**dob_velocity3_date**	143	**ssn-homephone_velocity90_date**
20	**dob_velocity7_date**	144	**ssn-homephone_velocity180_date**
21	**dob_velocity14_date**	145	**ssn-address_velocity0_date**
22	**dob_velocity30_date**	146	**ssn-address_velocity1_date**
23	**dob_velocity90_date**	147	**ssn-address_velocity3_date**
24	**dob_velocity180_date**	148	**ssn-address_velocity7_date**
25	**homephone_velocity0_date**	149	**ssn-address_velocity14_date**
26	**homephone_velocity1_date**	150	**ssn-address_velocity30_date**
27	**homephone_velocity3_date**	151	**ssn-address_velocity90_date**
28	**homephone_velocity7_date**	152	**ssn-address_velocity180_date**
29	**homephone_velocity14_date**	153	**ssn-address-zip_velocity0_date**
30	**homephone_velocity30_date**	154	**ssn-address-zip_velocity1_date**
31	**homephone_velocity90_date**	155	**ssn-address-zip_velocity3_date**
32	**homephone_velocity180_date**	156	**ssn-address-zip_velocity7_date**
33	**fullname_velocity0_date**	157	**ssn-address-zip_velocity14_date**
34	**fullname_velocity1_date**	158	**ssn-address-zip_velocity30_date**
35	**fullname_velocity3_date**	159	**ssn-address-zip_velocity90_date**
36	**fullname_velocity7_date**	160	**ssn-address-zip_velocity180_date**
37	**fullname_velocity14_date**	161	**ssn-fullname-dob_velocity0_date**
38	**fullname_velocity30_date**	162	**ssn-fullname-dob_velocity1_date**
39	**fullname_velocity90_date**	163	**ssn-fullname-dob_velocity3_date**
40	**fullname_velocity180_date**	164	**ssn-fullname-dob_velocity7_date**

41	**fullname-dob_velocity0_date**	165	**ssn-fullname-dob_velocity14_date**
42	**fullname-dob_velocity1_date**	166	**ssn-fullname-dob_velocity30_date**
43	**fullname-dob_velocity3_date**	167	**ssn-fullname-dob_velocity90_date**
44	**fullname-dob_velocity7_date**	168	**ssn-fullname-dob_velocity180_date**
45	**fullname-dob_velocity14_date**	169	**address-zip_velocity0_date**
46	**fullname-dob_velocity30_date**	170	**address-zip_velocity1_date**
47	**fullname-dob_velocity90_date**	171	**address-zip_velocity3_date**
48	**fullname-dob_velocity180_date**	172	**address-zip_velocity7_date**
49	**fullname-ssn_velocity0_date**	173	**address-zip_velocity14_date**
50	**fullname-ssn_velocity1_date**	174	**address-zip_velocity30_date**
51	**fullname-ssn_velocity3_date**	175	**address-zip_velocity90_date**
52	**fullname-ssn_velocity7_date**	176	**address-zip_velocity180_date**
53	**fullname-ssn_velocity14_date**	177	**address-zip-fullname-dob_velocity0_date**
54	**fullname-ssn_velocity30_date**	178	**address-zip-fullname-dob_velocity1_date**
55	**fullname-ssn_velocity90_date**	179	**address-zip-fullname-dob_velocity3_date**
56	**fullname-ssn_velocity180_date**	180	**address-zip-fullname-dob_velocity7_date**
57	**fullname-homephone_velocity0_date**	181	**address-zip-fullname-dob_velocity14_date**
58	**fullname-homephone_velocity1_date**	182	**address-zip-fullname-dob_velocity30_date**
59	**fullname-homephone_velocity3_date**	183	**address-zip-fullname-dob_velocity90_date**
60	**fullname-homephone_velocity7_date**	184	**address-zip-fullname-dob_velocity180_date**
61	**fullname-homephone_velocity14_date**	185	**address-zip-homephone_velocity0_date**
62	**fullname-homephone_velocity30_date**	186	**address-zip-homephone_velocity1_date**
63	**fullname-homephone_velocity90_date**	187	**address-zip-homephone_velocity3_date**
64	**fullname-homephone_velocity180_date**	188	**address-zip-homephone_velocity7_date**
65	**fullname-address_velocity0_date**	189	**address-zip-homephone_velocity14_date**
66	**fullname-address_velocity1_date**	190	**address-zip-homephone_velocity30_date**
67	**fullname-address_velocity3_date**	191	**address-zip-homephone_velocity90_date**
68	**fullname-address_velocity7_date**	192	**address-zip-homephone_velocity180_date**
69	**fullname-address_velocity14_date**	193	**zip-homephone_velocity0_date**
70	**fullname-address_velocity30_date**	194	**zip-homephone_velocity1_date**
71	**fullname-address_velocity90_date**	195	**zip-homephone_velocity3_date**
72	**fullname-address_velocity180_date**	196	**zip-homephone_velocity7_date**
73	**fullname-address-zip_velocity0_date**	197	**zip-homephone_velocity14_date**
74	**fullname-address-zip_velocity1_date**	198	**zip-homephone_velocity30_date**
75	**fullname-address-zip_velocity3_date**	199	**zip-homephone_velocity90_date**
76	**fullname-address-zip_velocity7_date**	200	**zip-homephone_velocity180_date**
77	**fullname-address-zip_velocity14_date**	201	**zip-dob_velocity0_date**
78	**fullname-address-zip_velocity30_date**	202	**zip-dob_velocity1_date**
79	**fullname-address-zip_velocity90_date**	203	**zip-dob_velocity3_date**
80	**fullname-address-zip_velocity180_date**	204	**zip-dob_velocity7_date**
81	**fullname-dob-homephone_velocity0_date**	205	**zip-dob_velocity14_date**
82	**fullname-dob-homephone_velocity1_date**	206	**zip-dob_velocity30_date**
83	**fullname-dob-homephone_velocity3_date**	207	**zip-dob_velocity90_date**

84	**fullname-dob-homephone_velocity7_date**	208	**zip-dob_velocity180_date**
85	**fullname-dob-homephone_velocity14_date**	209	**homephone-dob_velocity0_date**
86	**fullname-dob-homephone_velocity30_date**	210	**homephone-dob_velocity1_date**
87	**fullname-dob-homephone_velocity90_date**	211	**homephone-dob_velocity3_date**
88	**fullname-dob-homephone_velocity180_date**	212	**homephone-dob_velocity7_date**
89	**fullname-dob-zip_velocity0_date**	213	**homephone-dob_velocity14_date**
90	**fullname-dob-zip_velocity1_date**	214	**homephone-dob_velocity30_date**
91	**fullname-dob-zip_velocity3_date**	215	**homephone-dob_velocity90_date**
92	**fullname-dob-zip_velocity7_date**	216	**homephone-dob_velocity180_date**
93	**fullname-dob-zip_velocity14_date**	217	**firstname-dob_velocity0_date**
94	**fullname-dob-zip_velocity30_date**	218	**firstname-dob_velocity1_date**
95	**fullname-dob-zip_velocity90_date**	219	**firstname-dob_velocity3_date**
96	**fullname-dob-zip_velocity180_date**	220	**firstname-dob_velocity7_date**
97	**fullname-zip_velocity0_date**	221	**firstname-dob_velocity14_date**
98	**fullname-zip_velocity1_date**	222	**firstname-dob_velocity30_date**
99	**fullname-zip_velocity3_date**	223	**firstname-dob_velocity90_date**
100	**fullname-zip_velocity7_date**	224	**firstname-dob_velocity180_date**
101	**fullname-zip_velocity14_date**	225	**lastname-dob_velocity0_date**
102	**fullname-zip_velocity30_date**	226	**lastname-dob_velocity1_date**
103	**fullname-zip_velocity90_date**	227	**lastname-dob_velocity3_date**
104	**fullname-zip_velocity180_date**	228	**lastname-dob_velocity7_date**
105	**ssn-firstname_velocity0_date**	229	**lastname-dob_velocity14_date**
106	**ssn-firstname_velocity1_date**	230	**lastname-dob_velocity30_date**
107	**ssn-firstname_velocity3_date**	231	**lastname-dob_velocity90_date**
108	**ssn-firstname_velocity7_date**	232	**lastname-dob_velocity180_date**
109	**ssn-firstname_velocity14_date**	233	**firstname-homephone_velocity0_date**
110	**ssn-firstname_velocity30_date**	234	**firstname-homephone_velocity1_date**
111	**ssn-firstname_velocity90_date**	235	**firstname-homephone_velocity3_date**
112	**ssn-firstname_velocity180_date**	236	**firstname-homephone_velocity7_date**
113	**ssn-lastname_velocity0_date**	237	**firstname-homephone_velocity14_date**
114	**ssn-lastname_velocity1_date**	238	**firstname-homephone_velocity30_date**
115	**ssn-lastname_velocity3_date**	239	**firstname-homephone_velocity90_date**
116	**ssn-lastname_velocity7_date**	240	**firstname-homephone_velocity180_date**
117	**ssn-lastname_velocity14_date**	241	**lastname-homephone_velocity0_date**
118	**ssn-lastname_velocity30_date**	242	**lastname-homephone_velocity1_date**
119	**ssn-lastname_velocity90_date**	243	**lastname-homephone_velocity3_date**
120	**ssn-lastname_velocity180_date**	244	**lastname-homephone_velocity7_date**
121	**ssn-zip_velocity0_date**	245	**lastname-homephone_velocity14_date**
122	**ssn-zip_velocity1_date**	246	**lastname-homephone_velocity30_date**
123	**ssn-zip_velocity3_date**	247	**lastname-homephone_velocity90_date**
124	**ssn-zip_velocity7_date**	248	**lastname-homephone_velocity180_date**

	Relative Velocity Candidate Variables		Relative Velocity Candidate Variables
1	ssn_0_dayvel_div_3_dayvel_relvelocity	187	ssn-zip_1_dayvel_div_3_dayvel_relvelocity
2	ssn_0_dayvel_div_7_dayvel_relvelocity	188	ssn-zip_1_dayvel_div_7_dayvel_relvelocity
3	ssn_0_dayvel_div_14_dayvel_relvelocity	189	ssn-zip_1_dayvel_div_14_dayvel_relvelocity
4	ssn_0_dayvel_div_30_dayvel_relvelocity	190	ssn-zip_1_dayvel_div_30_dayvel_relvelocity
5	ssn_0_dayvel_div_90_dayvel_relvelocity	191	ssn-zip_1_dayvel_div_90_dayvel_relvelocity
6	ssn_0_dayvel_div_180_dayvel_relvelocity	192	ssn-zip_1_dayvel_div_180_dayvel_relvelocity
7	ssn_1_dayvel_div_3_dayvel_relvelocity	193	ssn-dob_0_dayvel_div_3_dayvel_relvelocity
8	ssn_1_dayvel_div_7_dayvel_relvelocity	194	ssn-dob_0_dayvel_div_7_dayvel_relvelocity
9	ssn_1_dayvel_div_14_dayvel_relvelocity	195	ssn-dob_0_dayvel_div_14_dayvel_relvelocity
10	ssn_1_dayvel_div_30_dayvel_relvelocity	196	ssn-dob_0_dayvel_div_30_dayvel_relvelocity
11	ssn_1_dayvel_div_90_dayvel_relvelocity	197	ssn-dob_0_dayvel_div_90_dayvel_relvelocity
12	ssn_1_dayvel_div_180_dayvel_relvelocity	198	ssn-dob_0_dayvel_div_180_dayvel_relvelocity
13	address_0_dayvel_div_3_dayvel_relvelocity	199	ssn-dob_1_dayvel_div_3_dayvel_relvelocity
14	address_0_dayvel_div_7_dayvel_relvelocity	200	ssn-dob_1_dayvel_div_7_dayvel_relvelocity
15	address_0_dayvel_div_14_dayvel_relvelocity	201	ssn-dob_1_dayvel_div_14_dayvel_relvelocity
16	address_0_dayvel_div_30_dayvel_relvelocity	202	ssn-dob_1_dayvel_div_30_dayvel_relvelocity
17	address_0_dayvel_div_90_dayvel_relvelocity	203	ssn-dob_1_dayvel_div_90_dayvel_relvelocity
18	address_0_dayvel_div_180_dayvel_relvelocity	204	ssn-dob_1_dayvel_div_180_dayvel_relvelocity
19	address_1_dayvel_div_3_dayvel_relvelocity	205	ssn-homephone_0_dayvel_div_3_dayvel_relvelocity
20	address_1_dayvel_div_7_dayvel_relvelocity	206	ssn-homephone_0_dayvel_div_7_dayvel_relvelocity
21	address_1_dayvel_div_14_dayvel_relvelocity	207	ssn-homephone_0_dayvel_div_14_dayvel_relvelocity
22	address_1_dayvel_div_30_dayvel_relvelocity	208	ssn-homephone_0_dayvel_div_30_dayvel_relvelocity
23	address_1_dayvel_div_90_dayvel_relvelocity	209	ssn-homephone_0_dayvel_div_90_dayvel_relvelocity

24	address_1_dayvel_div_180_dayvel_relvelocity	210	ssn-homephone_0_dayvel_div_180_dayvel_relvelocity
25	dob_0_dayvel_div_3_dayvel_relvelocity	211	ssn-homephone_1_dayvel_div_3_dayvel_relvelocity
26	dob_0_dayvel_div_7_dayvel_relvelocity	212	ssn-homephone_1_dayvel_div_7_dayvel_relvelocity
27	dob_0_dayvel_div_14_dayvel_relvelocity	213	ssn-homephone_1_dayvel_div_14_dayvel_relvelocity
28	dob_0_dayvel_div_30_dayvel_relvelocity	214	ssn-homephone_1_dayvel_div_30_dayvel_relvelocity
29	dob_0_dayvel_div_90_dayvel_relvelocity	215	ssn-homephone_1_dayvel_div_90_dayvel_relvelocity
30	dob_0_dayvel_div_180_dayvel_relvelocity	216	ssn-homephone_1_dayvel_div_180_dayvel_relvelocity
31	dob_1_dayvel_div_3_dayvel_relvelocity	217	ssn-address_0_dayvel_div_3_dayvel_relvelocity
32	dob_1_dayvel_div_7_dayvel_relvelocity	218	ssn-address_0_dayvel_div_7_dayvel_relvelocity
33	dob_1_dayvel_div_14_dayvel_relvelocity	219	ssn-address_0_dayvel_div_14_dayvel_relvelocity
34	dob_1_dayvel_div_30_dayvel_relvelocity	220	ssn-address_0_dayvel_div_30_dayvel_relvelocity
35	dob_1_dayvel_div_90_dayvel_relvelocity	221	ssn-address_0_dayvel_div_90_dayvel_relvelocity
36	dob_1_dayvel_div_180_dayvel_relvelocity	222	ssn-address_0_dayvel_div_180_dayvel_relvelocity
37	homephone_0_dayvel_div_3_dayvel_relvelocity	223	ssn-address_1_dayvel_div_3_dayvel_relvelocity
38	homephone_0_dayvel_div_7_dayvel_relvelocity	224	ssn-address_1_dayvel_div_7_dayvel_relvelocity
39	homephone_0_dayvel_div_14_dayvel_relvelocity	225	ssn-address_1_dayvel_div_14_dayvel_relvelocity
40	homephone_0_dayvel_div_30_dayvel_relvelocity	226	ssn-address_1_dayvel_div_30_dayvel_relvelocity

41	homephone_0_dayvel_div_90_dayvel_relvelocity	227	ssn-address_1_dayvel_div_90_dayvel_relvelocity
42	homephone_0_dayvel_div_180_dayvel_relvelocity	228	ssn-address_1_dayvel_div_180_dayvel_relvelocity
43	homephone_1_dayvel_div_3_dayvel_relvelocity	229	ssn-address-zip_0_dayvel_div_3_dayvel_relvelocity
44	homephone_1_dayvel_div_7_dayvel_relvelocity	230	ssn-address-zip_0_dayvel_div_7_dayvel_relvelocity
45	homephone_1_dayvel_div_14_dayvel_relvelocity	231	ssn-address-zip_0_dayvel_div_14_dayvel_relvelocity
46	homephone_1_dayvel_div_30_dayvel_relvelocity	232	ssn-address-zip_0_dayvel_div_30_dayvel_relvelocity
47	homephone_1_dayvel_div_90_dayvel_relvelocity	233	ssn-address-zip_0_dayvel_div_90_dayvel_relvelocity
48	homephone_1_dayvel_div_180_dayvel_relvelocity	234	ssn-address-zip_0_dayvel_div_180_dayvel_relvelocity
49	fullname_0_dayvel_div_3_dayvel_relvelocity	235	ssn-address-zip_1_dayvel_div_3_dayvel_relvelocity
50	fullname_0_dayvel_div_7_dayvel_relvelocity	236	ssn-address-zip_1_dayvel_div_7_dayvel_relvelocity
51	fullname_0_dayvel_div_14_dayvel_relvelocity	237	ssn-address-zip_1_dayvel_div_14_dayvel_relvelocity
52	fullname_0_dayvel_div_30_dayvel_relvelocity	238	ssn-address-zip_1_dayvel_div_30_dayvel_relvelocity
53	fullname_0_dayvel_div_90_dayvel_relvelocity	239	ssn-address-zip_1_dayvel_div_90_dayvel_relvelocity
54	fullname_0_dayvel_div_180_dayvel_relvelocity	240	ssn-address-zip_1_dayvel_div_180_dayvel_relvelocity
55	fullname_1_dayvel_div_3_dayvel_relvelocity	241	ssn-fullname-dob_0_dayvel_div_3_dayvel_relvelocity
56	fullname_1_dayvel_div_7_dayvel_relvelocity	242	ssn-fullname-dob_0_dayvel_div_7_dayvel_relvelocity
57	fullname_1_dayvel_div_14_dayvel_relvelocity	243	ssn-fullname-dob_0_dayvel_div_14_dayvel_relvelocity
58	fullname_1_dayvel_div_30_dayvel_relvelocity	244	ssn-fullname-dob_0_dayvel_div_30_dayvel_relvelocity
59	fullname_1_dayvel_div_90_dayvel_relvelocity	245	ssn-fullname-dob_0_dayvel_div_90_dayvel_relvelocity
60	fullname_1_dayvel_div_180_dayvel_relvelocity	246	ssn-fullname-dob_0_dayvel_div_180_dayvel_relvelocity
61	fullname-dob_0_dayvel_div_3_dayvel_relvelocity	247	ssn-fullname-dob_1_dayvel_div_3_dayvel_relvelocity
62	fullname-dob_0_dayvel_div_7_dayvel_relvelocity	248	ssn-fullname-dob_1_dayvel_div_7_dayvel_relvelocity

63	fullname-dob_0_dayvel_div_14_dayvel_relvelocity	249	ssn-fullname-dob_1_dayvel_div_14_dayvel_relvelocity
64	fullname-dob_0_dayvel_div_30_dayvel_relvelocity	250	ssn-fullname-dob_1_dayvel_div_30_dayvel_relvelocity
65	fullname-dob_0_dayvel_div_90_dayvel_relvelocity	251	ssn-fullname-dob_1_dayvel_div_90_dayvel_relvelocity
66	fullname-dob_0_dayvel_div_180_dayvel_relvelocity	252	ssn-fullname-dob_1_dayvel_div_180_dayvel_relvelocity
67	fullname-dob_1_dayvel_div_3_dayvel_relvelocity	253	address-zip_0_dayvel_div_3_dayvel_relvelocity
68	fullname-dob_1_dayvel_div_7_dayvel_relvelocity	254	address-zip_0_dayvel_div_7_dayvel_relvelocity
69	fullname-dob_1_dayvel_div_14_dayvel_relvelocity	255	address-zip_0_dayvel_div_14_dayvel_relvelocity
70	fullname-dob_1_dayvel_div_30_dayvel_relvelocity	256	address-zip_0_dayvel_div_30_dayvel_relvelocity
71	fullname-dob_1_dayvel_div_90_dayvel_relvelocity	257	address-zip_0_dayvel_div_90_dayvel_relvelocity
72	fullname-dob_1_dayvel_div_180_dayvel_relvelocity	258	address-zip_0_dayvel_div_180_dayvel_relvelocity
73	fullname-ssn_0_dayvel_div_3_dayvel_relvelocity	259	address-zip_1_dayvel_div_3_dayvel_relvelocity
74	fullname-ssn_0_dayvel_div_7_dayvel_relvelocity	260	address-zip_1_dayvel_div_7_dayvel_relvelocity
75	fullname-ssn_0_dayvel_div_14_dayvel_relvelocity	261	address-zip_1_dayvel_div_14_dayvel_relvelocity
76	fullname-ssn_0_dayvel_div_30_dayvel_relvelocity	262	address-zip_1_dayvel_div_30_dayvel_relvelocity
77	fullname-ssn_0_dayvel_div_90_dayvel_relvelocity	263	address-zip_1_dayvel_div_90_dayvel_relvelocity
78	fullname-ssn_0_dayvel_div_180_dayvel_relvelocity	264	address-zip_1_dayvel_div_180_dayvel_relvelocity
79	fullname-ssn_1_dayvel_div_3_dayvel_relvelocity	265	address-zip-fullname-dob_0_dayvel_div_3_dayvel_relvelocity
80	fullname-ssn_1_dayvel_div_7_dayvel_relvelocity	266	address-zip-fullname-dob_0_dayvel_div_7_dayvel_relvelocity
81	fullname-ssn_1_dayvel_div_14_dayvel_relvelocity	267	address-zip-fullname-dob_0_dayvel_div_14_dayvel_relvelocity
82	fullname-ssn_1_dayvel_div_30_dayvel_relvelocity	268	address-zip-fullname-dob_0_dayvel_div_30_dayvel_relvelocity
83	fullname-ssn_1_dayvel_div_90_dayvel_relvelocity	269	address-zip-fullname-dob_0_dayvel_div_90_dayvel_relvelocity
84	fullname-ssn_1_dayvel_div_180_dayvel_relvelocity	270	address-zip-fullname-dob_0_dayvel_div_180_dayvel_relvelocity

85	**fullname-homephone_0_dayvel_div_3_dayvel_relvelocity**	271	**address-zip-fullname-dob_1_dayvel_div_3_dayvel_relvelocity**
86	**fullname-homephone_0_dayvel_div_7_dayvel_relvelocity**	272	**address-zip-fullname-dob_1_dayvel_div_7_dayvel_relvelocity**
87	**fullname-homephone_0_dayvel_div_14_dayvel_relvelocity**	273	**address-zip-fullname-dob_1_dayvel_div_14_dayvel_relvelocity**
88	**fullname-homephone_0_dayvel_div_30_dayvel_relvelocity**	274	**address-zip-fullname-dob_1_dayvel_div_30_dayvel_relvelocity**
89	**fullname-homephone_0_dayvel_div_90_dayvel_relvelocity**	275	**address-zip-fullname-dob_1_dayvel_div_90_dayvel_relvelocity**
90	**fullname-homephone_0_dayvel_div_180_dayvel_relvelocity**	276	**address-zip-fullname-dob_1_dayvel_div_180_dayvel_relvelocity**
91	**fullname-homephone_1_dayvel_div_3_dayvel_relvelocity**	277	**address-zip-homephone_0_dayvel_div_3_dayvel_relvelocity**
92	**fullname-homephone_1_dayvel_div_7_dayvel_relvelocity**	278	**address-zip-homephone_0_dayvel_div_7_dayvel_relvelocity**
93	**fullname-homephone_1_dayvel_div_14_dayvel_relvelocity**	279	**address-zip-homephone_0_dayvel_div_14_dayvel_relvelocity**
94	**fullname-homephone_1_dayvel_div_30_dayvel_relvelocity**	280	**address-zip-homephone_0_dayvel_div_30_dayvel_relvelocity**
95	**fullname-homephone_1_dayvel_div_90_dayvel_relvelocity**	281	**address-zip-homephone_0_dayvel_div_90_dayvel_relvelocity**
96	**fullname-homephone_1_dayvel_div_180_dayvel_relvelocity**	282	**address-zip-homephone_0_dayvel_div_180_dayvel_relvelocity**
97	**fullname-address_0_dayvel_div_3_dayvel_relvelocity**	283	**address-zip-homephone_1_dayvel_div_3_dayvel_relvelocity**
98	**fullname-address_0_dayvel_div_7_dayvel_relvelocity**	284	**address-zip-homephone_1_dayvel_div_7_dayvel_relvelocity**
99	**fullname-address_0_dayvel_div_14_dayvel_relvelocity**	285	**address-zip-homephone_1_dayvel_div_14_dayvel_relvelocity**

100	fullname-address_0_dayvel_div_30_dayvel_relvelocity	286	address-zip-homephone_1_dayvel_div_30_dayvel_relvelocity
101	fullname-address_0_dayvel_div_90_dayvel_relvelocity	287	address-zip-homephone_1_dayvel_div_90_dayvel_relvelocity
102	fullname-address_0_dayvel_div_180_dayvel_relvelocity	288	address-zip-homephone_1_dayvel_div_180_dayvel_relvelocity
103	fullname-address_1_dayvel_div_3_dayvel_relvelocity	289	zip-homephone_0_dayvel_div_3_dayvel_relvelocity
104	fullname-address_1_dayvel_div_7_dayvel_relvelocity	290	zip-homephone_0_dayvel_div_7_dayvel_relvelocity
105	fullname-address_1_dayvel_div_14_dayvel_relvelocity	291	zip-homephone_0_dayvel_div_14_dayvel_relvelocity
106	fullname-address_1_dayvel_div_30_dayvel_relvelocity	292	zip-homephone_0_dayvel_div_30_dayvel_relvelocity
107	fullname-address_1_dayvel_div_90_dayvel_relvelocity	293	zip-homephone_0_dayvel_div_90_dayvel_relvelocity
108	fullname-address_1_dayvel_div_180_dayvel_relvelocity	294	zip-homephone_0_dayvel_div_180_dayvel_relvelocity
109	fullname-address-zip_0_dayvel_div_3_dayvel_relvelocity	295	zip-homephone_1_dayvel_div_3_dayvel_relvelocity
110	fullname-address-zip_0_dayvel_div_7_dayvel_relvelocity	296	zip-homephone_1_dayvel_div_7_dayvel_relvelocity
111	fullname-address-zip_0_dayvel_div_14_dayvel_relvelocity	297	zip-homephone_1_dayvel_div_14_dayvel_relvelocity
112	fullname-address-zip_0_dayvel_div_30_dayvel_relvelocity	298	zip-homephone_1_dayvel_div_30_dayvel_relvelocity
113	fullname-address-zip_0_dayvel_div_90_dayvel_relvelocity	299	zip-homephone_1_dayvel_div_90_dayvel_relvelocity
114	fullname-address-zip_0_dayvel_div_180_dayvel_relvelocity	300	zip-homephone_1_dayvel_div_180_dayvel_relvelocity
115	fullname-address-zip_1_dayvel_div_3_dayvel_relvelocity	301	zip-dob_0_dayvel_div_3_dayvel_relvelocity

116	**fullname-address-zip_1_dayvel_div_7_dayvel_relvelocity**	302	**zip-dob_0_dayvel_div_7_dayvel_relvelocity**
117	**fullname-address-zip_1_dayvel_div_14_dayvel_relvelocity**	303	**zip-dob_0_dayvel_div_14_dayvel_relvelocity**
118	**fullname-address-zip_1_dayvel_div_30_dayvel_relvelocity**	304	**zip-dob_0_dayvel_div_30_dayvel_relvelocity**
119	**fullname-address-zip_1_dayvel_div_90_dayvel_relvelocity**	305	**zip-dob_0_dayvel_div_90_dayvel_relvelocity**
120	**fullname-address-zip_1_dayvel_div_180_dayvel_relvelocity**	306	**zip-dob_0_dayvel_div_180_dayvel_relvelocity**
121	**fullname-dob-homephone_0_dayvel_div_3_dayvel_relvelocity**	307	**zip-dob_1_dayvel_div_3_dayvel_relvelocity**
122	**fullname-dob-homephone_0_dayvel_div_7_dayvel_relvelocity**	308	**zip-dob_1_dayvel_div_7_dayvel_relvelocity**
123	**fullname-dob-homephone_0_dayvel_div_14_dayvel_relvelocity**	309	**zip-dob_1_dayvel_div_14_dayvel_relvelocity**
124	**fullname-dob-homephone_0_dayvel_div_30_dayvel_relvelocity**	310	**zip-dob_1_dayvel_div_30_dayvel_relvelocity**
125	**fullname-dob-homephone_0_dayvel_div_90_dayvel_relvelocity**	311	**zip-dob_1_dayvel_div_90_dayvel_relvelocity**
126	**fullname-dob-homephone_0_dayvel_div_180_dayvel_relvelocity**	312	**zip-dob_1_dayvel_div_180_dayvel_relvelocity**
127	**fullname-dob-homephone_1_dayvel_div_3_dayvel_relvelocity**	313	**homephone-dob_0_dayvel_div_3_dayvel_relvelocity**
128	**fullname-dob-homephone_1_dayvel_div_7_dayvel_relvelocity**	314	**homephone-dob_0_dayvel_div_7_dayvel_relvelocity**
129	**fullname-dob-homephone_1_dayvel_div_14_dayvel_relvelocity**	315	**homephone-dob_0_dayvel_div_14_dayvel_relvelocity**
130	**fullname-dob-homephone_1_dayvel_div_30_dayvel_relvelocity**	316	**homephone-dob_0_dayvel_div_30_dayvel_relvelocity**
131	**fullname-dob-homephone_1_dayvel_div_90_dayvel_relvelocity**	317	**homephone-dob_0_dayvel_div_90_dayvel_relvelocity**
132	**fullname-dob-homephone_1_dayvel_div_180_dayvel_relvelocity**	318	**homephone-dob_0_dayvel_div_180_dayvel_relvelocity**

133	fullname-dob-zip_0_dayvel_div_3_dayvel_relvelocity	319	homephone-dob_1_dayvel_div_3_dayvel_relvelocity
134	fullname-dob-zip_0_dayvel_div_7_dayvel_relvelocity	320	homephone-dob_1_dayvel_div_7_dayvel_relvelocity
135	fullname-dob-zip_0_dayvel_div_14_dayvel_relvelocity	321	homephone-dob_1_dayvel_div_14_dayvel_relvelocity
136	fullname-dob-zip_0_dayvel_div_30_dayvel_relvelocity	322	homephone-dob_1_dayvel_div_30_dayvel_relvelocity
137	fullname-dob-zip_0_dayvel_div_90_dayvel_relvelocity	323	homephone-dob_1_dayvel_div_90_dayvel_relvelocity
138	fullname-dob-zip_0_dayvel_div_180_dayvel_relvelocity	324	homephone-dob_1_dayvel_div_180_dayvel_relvelocity
139	fullname-dob-zip_1_dayvel_div_3_dayvel_relvelocity	325	firstname-dob_0_dayvel_div_3_dayvel_relvelocity
140	fullname-dob-zip_1_dayvel_div_7_dayvel_relvelocity	326	firstname-dob_0_dayvel_div_7_dayvel_relvelocity
141	fullname-dob-zip_1_dayvel_div_14_dayvel_relvelocity	327	firstname-dob_0_dayvel_div_14_dayvel_relvelocity
142	fullname-dob-zip_1_dayvel_div_30_dayvel_relvelocity	328	firstname-dob_0_dayvel_div_30_dayvel_relvelocity
143	fullname-dob-zip_1_dayvel_div_90_dayvel_relvelocity	329	firstname-dob_0_dayvel_div_90_dayvel_relvelocity
144	fullname-dob-zip_1_dayvel_div_180_dayvel_relvelocity	330	firstname-dob_0_dayvel_div_180_dayvel_relvelocity
145	fullname-zip_0_dayvel_div_3_dayvel_relvelocity	331	firstname-dob_1_dayvel_div_3_dayvel_relvelocity
146	fullname-zip_0_dayvel_div_7_dayvel_relvelocity	332	firstname-dob_1_dayvel_div_7_dayvel_relvelocity
147	fullname-zip_0_dayvel_div_14_dayvel_relvelocity	333	firstname-dob_1_dayvel_div_14_dayvel_relvelocity
148	fullname-zip_0_dayvel_div_30_dayvel_relvelocity	334	firstname-dob_1_dayvel_div_30_dayvel_relvelocity
149	fullname-zip_0_dayvel_div_90_dayvel_relvelocity	335	firstname-dob_1_dayvel_div_90_dayvel_relvelocity
150	fullname-zip_0_dayvel_div_180_dayvel_relvelocity	336	firstname-dob_1_dayvel_div_180_dayvel_relvelocity
151	fullname-zip_1_dayvel_div_3_dayvel_relvelocity	337	lastname-dob_0_dayvel_div_3_dayvel_relvelocity
152	fullname-zip_1_dayvel_div_7_dayvel_relvelocity	338	lastname-dob_0_dayvel_div_7_dayvel_relvelocity
153	fullname-zip_1_dayvel_div_14_dayvel_relvelocity	339	lastname-dob_0_dayvel_div_14_dayvel_relvelocity
154	fullname-zip_1_dayvel_div_30_dayvel_relvelocity	340	lastname-dob_0_dayvel_div_30_dayvel_relvelocity
155	fullname-zip_1_dayvel_div_90_dayvel_relvelocity	341	lastname-dob_0_dayvel_div_90_dayvel_relvelocity

156	**fullname-zip_1_dayvel_div_180_dayvel_relvelocity**	342	**lastname-dob_0_dayvel_div_180_dayvel_relvelocity**
157	**ssn-firstname_0_dayvel_div_3_dayvel_relvelocity**	343	**lastname-dob_1_dayvel_div_3_dayvel_relvelocity**
158	**ssn-firstname_0_dayvel_div_7_dayvel_relvelocity**	344	**lastname-dob_1_dayvel_div_7_dayvel_relvelocity**
159	**ssn-firstname_0_dayvel_div_14_dayvel_relvelocity**	345	**lastname-dob_1_dayvel_div_14_dayvel_relvelocity**
160	**ssn-firstname_0_dayvel_div_30_dayvel_relvelocity**	346	**lastname-dob_1_dayvel_div_30_dayvel_relvelocity**
161	**ssn-firstname_0_dayvel_div_90_dayvel_relvelocity**	347	**lastname-dob_1_dayvel_div_90_dayvel_relvelocity**
162	**ssn-firstname_0_dayvel_div_180_dayvel_relvelocity**	348	**lastname-dob_1_dayvel_div_180_dayvel_relvelocity**
163	**ssn-firstname_1_dayvel_div_3_dayvel_relvelocity**	349	**firstname-homephone_0_dayvel_div_3_dayvel_relvelocity**
164	**ssn-firstname_1_dayvel_div_7_dayvel_relvelocity**	350	**firstname-homephone_0_dayvel_div_7_dayvel_relvelocity**
165	**ssn-firstname_1_dayvel_div_14_dayvel_relvelocity**	351	**firstname-homephone_0_dayvel_div_14_dayvel_relvelocity**
166	**ssn-firstname_1_dayvel_div_30_dayvel_relvelocity**	352	**firstname-homephone_0_dayvel_div_30_dayvel_relvelocity**
167	**ssn-firstname_1_dayvel_div_90_dayvel_relvelocity**	353	**firstname-homephone_0_dayvel_div_90_dayvel_relvelocity**
168	**ssn-firstname_1_dayvel_div_180_dayvel_relvelocity**	354	**firstname-homephone_0_dayvel_div_180_dayvel_relvelocity**
169	**ssn-lastname_0_dayvel_div_3_dayvel_relvelocity**	355	**firstname-homephone_1_dayvel_div_3_dayvel_relvelocity**
170	**ssn-lastname_0_dayvel_div_7_dayvel_relvelocity**	356	**firstname-homephone_1_dayvel_div_7_dayvel_relvelocity**
171	**ssn-lastname_0_dayvel_div_14_dayvel_relvelocity**	357	**firstname-homephone_1_dayvel_div_14_dayvel_relvelocity**

172	**ssn-lastname_0_dayvel_div_30_dayvel_relvelocity**	358	**firstname-homephone_1_dayvel_div_30_dayvel_relvelocity**
173	**ssn-lastname_0_dayvel_div_90_dayvel_relvelocity**	359	**firstname-homephone_1_dayvel_div_90_dayvel_relvelocity**
174	**ssn-lastname_0_dayvel_div_180_dayvel_relvelocity**	360	**firstname-homephone_1_dayvel_div_180_dayvel_relvelocity**
175	**ssn-lastname_1_dayvel_div_3_dayvel_relvelocity**	361	**lastname-homephone_0_dayvel_div_3_dayvel_relvelocity**
176	**ssn-lastname_1_dayvel_div_7_dayvel_relvelocity**	362	**lastname-homephone_0_dayvel_div_7_dayvel_relvelocity**
177	**ssn-lastname_1_dayvel_div_14_dayvel_relvelocity**	363	**lastname-homephone_0_dayvel_div_14_dayvel_relvelocity**
178	**ssn-lastname_1_dayvel_div_30_dayvel_relvelocity**	364	**lastname-homephone_0_dayvel_div_30_dayvel_relvelocity**
179	**ssn-lastname_1_dayvel_div_90_dayvel_relvelocity**	365	**lastname-homephone_0_dayvel_div_90_dayvel_relvelocity**
180	**ssn-lastname_1_dayvel_div_180_dayvel_relvelocity**	366	**lastname-homephone_0_dayvel_div_180_dayvel_relvelocity**
181	**ssn-zip_0_dayvel_div_3_dayvel_relvelocity**	367	**lastname-homephone_1_dayvel_div_3_dayvel_relvelocity**
182	**ssn-zip_0_dayvel_div_7_dayvel_relvelocity**	368	**lastname-homephone_1_dayvel_div_7_dayvel_relvelocity**
183	**ssn-zip_0_dayvel_div_14_dayvel_relvelocity**	369	**lastname-homephone_1_dayvel_div_14_dayvel_relvelocity**
184	**ssn-zip_0_dayvel_div_30_dayvel_relvelocity**	370	**lastname-homephone_1_dayvel_div_30_dayvel_relvelocity**
185	**ssn-zip_0_dayvel_div_90_dayvel_relvelocity**	371	**lastname-homephone_1_dayvel_div_90_dayvel_relvelocity**
186	**ssn-zip_0_dayvel_div_180_dayvel_relvelocity**	372	**lastname-homephone_1_dayvel_div_180_dayvel_relvelocity**

Day Since Candidate Variables	
1	**ssn_daysSince**
2	**address_daysSince**
3	**ndob_daysSince**
4	**phone_daysSince**
5	**ssnaddress_daysSince**
6	**ndobaddress_daysSince**
7	**phoneaddress_daysSince**
8	**ndobphone_daysSince**
9	**addressphone_daysSince**
10	**phonessn_daysSince**
11	**ndobaddress_daysSince**
12	**ssnndob_daysSince**
13	**lastnamessn_daysSince**
14	**fnssn_daysSince**