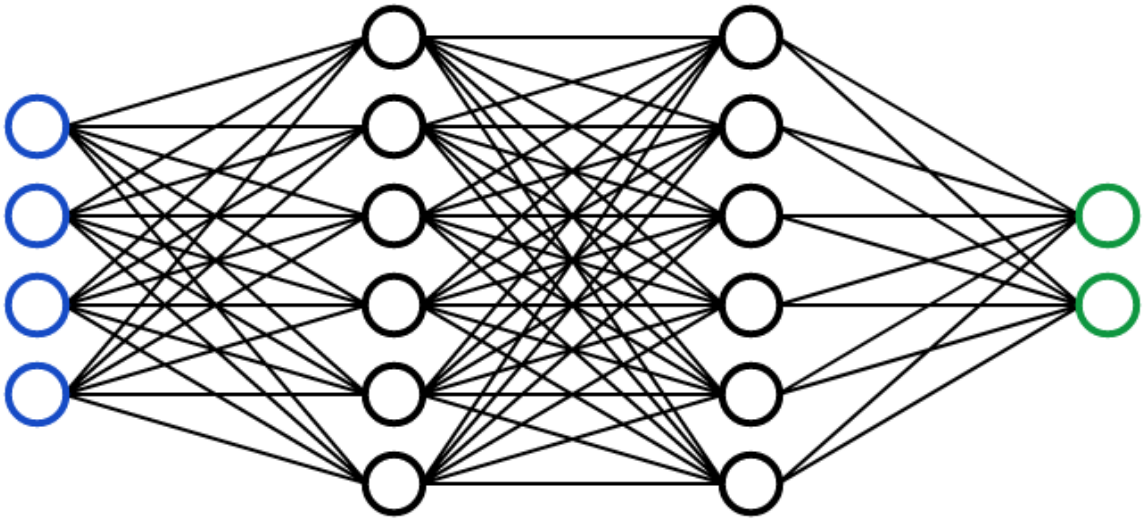
Building and Evaluating a Supervised Model for Fraud Detection

Henry Ajagbawa, Zach Cummings, Frank Wang, Vilhelm Sandberg, Sarah Russel, & Arjun Reddy



The University of Southern California

March 25, 2021

Table of Contents

Executive Summary

In this project, we were provided with a dataset consisting of 1 million rows and 10 fields, including a fraud label which could be used to train a supervised machine learning algorithm to recognize records that should be classified as fraudulent applications to a bank. We began by performing a preliminary analysis of the data, and then built a supervised model to predict whether or not a record should be classified as fraud.

First, we performed a descriptive analysis by building a data quality report to ensure the data was properly organized and ready for further analysis. For each variable we were provided, we examined its distribution and summary statistics. After gaining an understanding of the data and ensuring its integrity, we began constructing additional features that would serve as candidate variables for our predictive model. These engineered features were designed to illuminate typical signals of fraud in real life scenarios.

Once we'd engineered 365 additional candidate variables, we began the process of reducing dimensionality and identifying the few variables that best distinguished between fraudulent and non-fraudulent records. By first using a filter that combined the rank for the univariate KS and fraud detection rate scores for each variable, we were able to determine a subset of 80 variables that were good model input candidates. We further narrowed this list down to 30 using a logistic regression backwards selection wrapper.

Using our final 30 variables, we trained random forest, neural network, logistic regression and boosted tree models and adjusted various hyperparameter settings to fine tune each model. After model training, we evaluated the models' performance on the training, test, and out of time data and selected the model with the best performance on the out of time data, which most closely resembles how well the model will perform when applied in practice to incoming applications.

We are confident that our data cleaning, feature engineering, feature reduction, and model building processes could be applied to almost any credit card application database and result in a model that can helpfully classify potentially fraudulent applications before they get approved.

Description of Data

The following is our summary statistics table, which shows a statistical overview of each field we were originally provided with.

Field Name	Field Type	# of Records	% Populated	# Zeros	# Unique Values	Most Common Value
record	categorical	1,000,000	100	0	1000000	N/A
ssn	categorical	1,000,000	100	0	835819	999999999
firstname	categorical	1,000,000	100	0	78136	EAMSTRMT
lastname	categorical	1,000,000	100	0	177001	ERJSAXA
address	categorical	1,000,000	100	0	828774	123 MAIN ST
zip5	categorical	1,000,000	100	0	26370	68138
homephone	categorical	1,000,000	100	0	28244	9999999999
fraud_label	categorical	1,000,000	100	985607	2	0

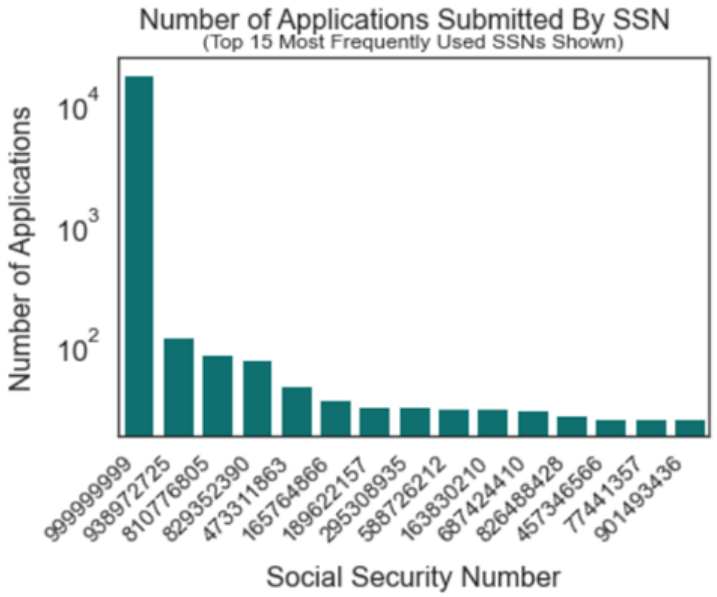


Figure Description: The most commonly used SSN is 999999999. Typically, this value is frivolous and is used as a placeholder for an actual SSN, or is deliberately used to commit fraud.

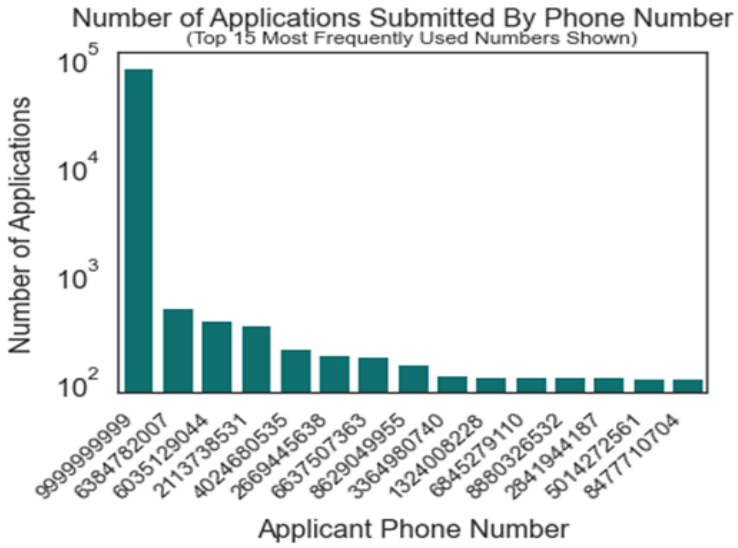


Figure Description: Above are the most frequently occurring phone numbers in data. The results are similar to the SSN distribution, in that 999999999 is the most frequently used number, and is used far more frequently than the next most frequently used phone numbers.

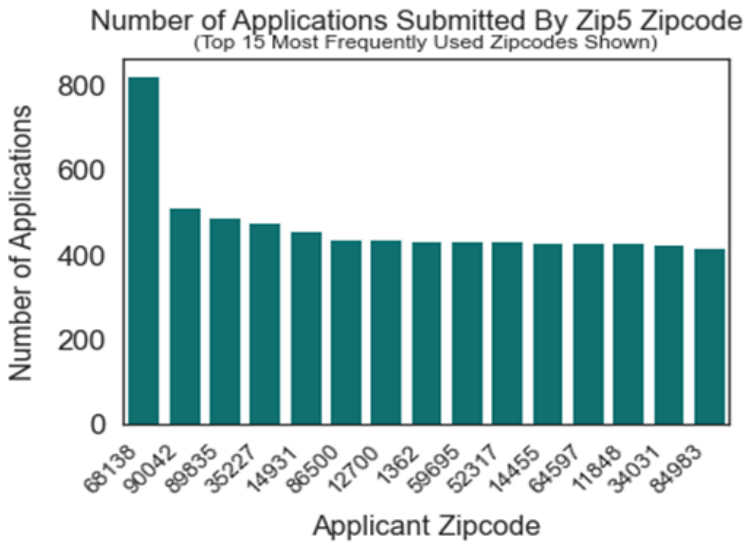


Figure Description: The most 10 frequently used zip codes in application data. Based on the distribution, the most frequently used zipcode is 68138, which is used far more than other zip codes occurring with high frequency.

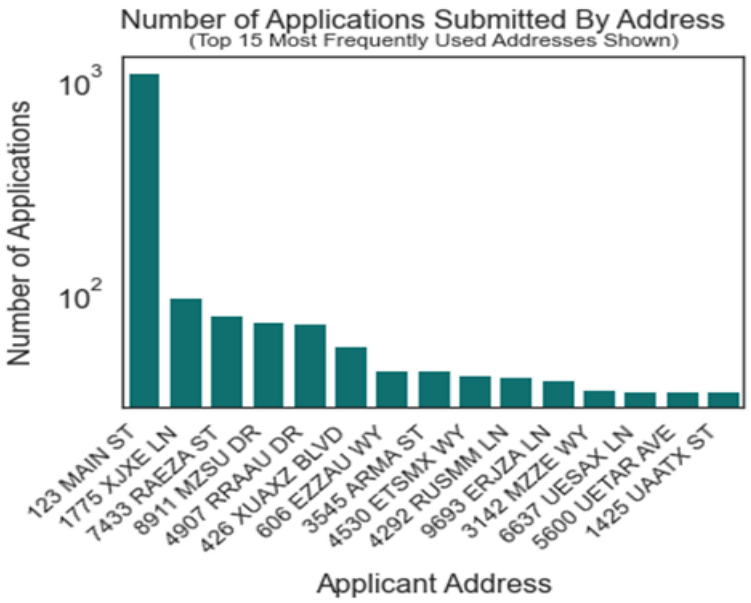


Figure Description: The distribution of addresses used. 123 Main Street is by far the most commonly used address. 123 Main Street is likely being used as a placeholder for an actual address, or as a commonly used address by fraudsters.

Data Cleaning

We performed data cleaning on 6 different variables in the dataset (the date column, the zip5 column, address column, date of birth column, and homephone column. The steps we used to clean each original field are detailed below.

Date

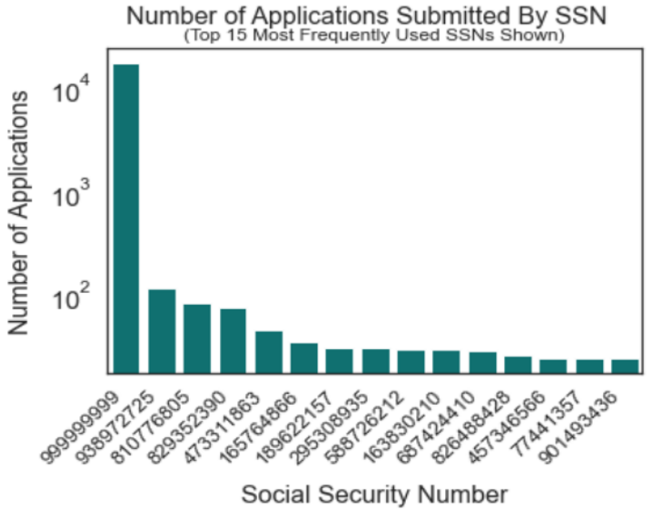
To fix up the date column we put in code to add dashes between the month, day, and year.

Zip5

To fix the zip code field, we noticed that zip codes that started with 0 were missing their first 0. For example the zip code 02739 was represented in this column as 2739. We created code to add in the first zero to any zip code record missing its leading zero.

Social Security Number

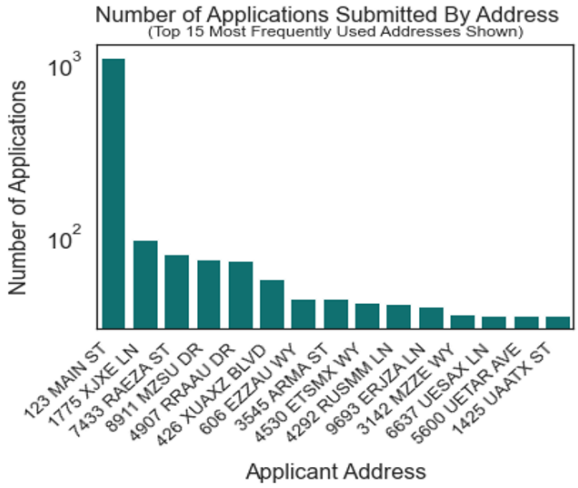
Looking at a histogram of the most used social security numbers below we see that there is a high frequency of the SSN 999-99-9999 being used. This is likely due to users typing in a false zipcode to get the application to pass, or due to a reporting error from the agency from which the data was collected.



Whatever the error might be, we chose to replace all instances of 999-99-9999 with something unique. For every ssn with the frivolous value, we replaced it with the negative of the record's record number, which we know will be unique and won't match with any other values when we perform grouping analysis in the feature engineering step.

Address

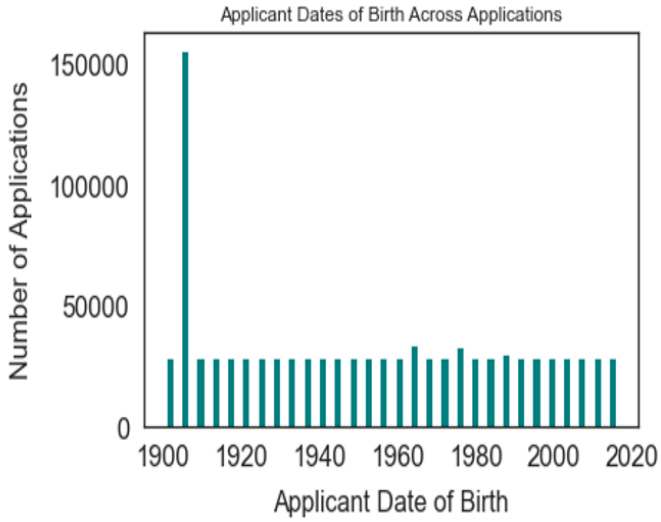
For address, we looked at the histogram below and saw a high number of credit card applications coming from 123 Main Street. We wanted to make sure this address was replaced with a unique value as well, since it was likely frivolous.



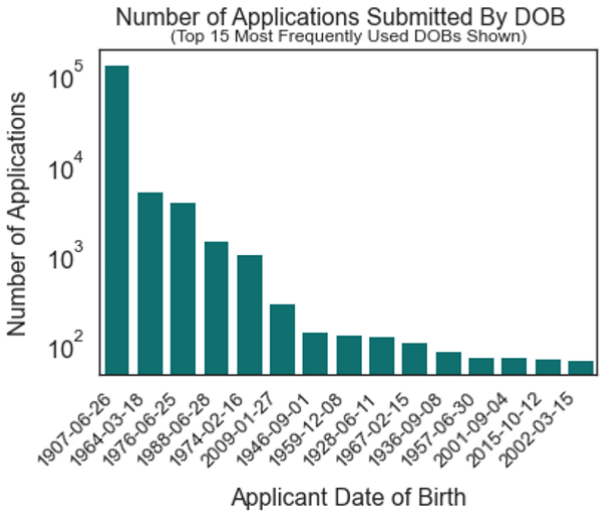
In order to replace 123 Main Street we added the record number to the end of any address with 123 Main Street to make it a unique address each time.

DOB

For Date of birth, when looking at the number of applications each year one can see a very large spike in the early 1900's.



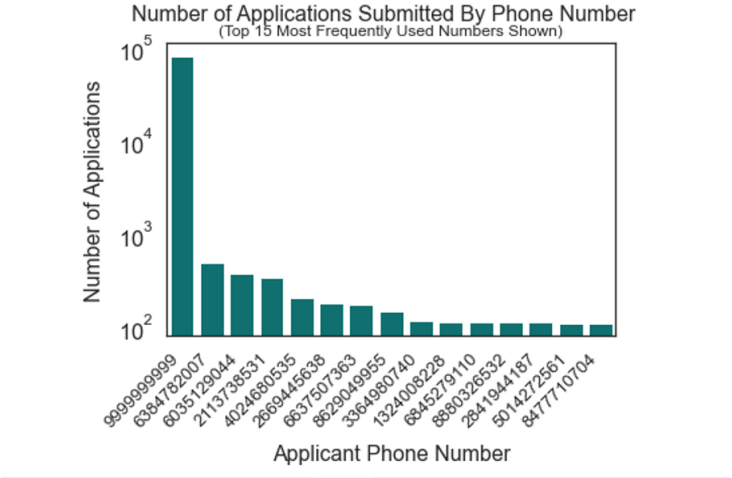
Upon inspecting further, we found that there are a very large number of applications with the birthday of 1907-06-26. This is likely another frivolous value so we needed to make sure that these date of births couldn't be linked in our feature engineering step.



In order to make the date of birth unique, we did the same thing we did with the social security number. We took the record of all date of births where it equaled the frivolous value and made it negative and subbed it in.

Homephone

When looking at the histogram below of the most common home phone values we saw 999-999-9999 occur frequently (similarly to the SSN field). This could have been a reporting issue or it could have been a frivolously entered value by customers looking to bypass mandatory phone number entry.



In order to fix the home phone value we did the same thing as with date of birth and ssn frivolous values where we substituted the value with the negative of the record number.

Creating Candidate Variables

The first step in creating variables for our model was to create additional entities that might be used for uniquely identifying someone. Some examples of these include someone's full name in conjunction with their date of birth or someone's full address along with their home phone number. We do this because most identifiers alone do not confirm the unique identity of an individual. For example, there may be multiple individuals with the same name or who have lived at the same address, but when we combine attributes, the chance of overlap lessens.

As one's social security number(ssn) is thought to be the pinnacle of unique identification, we also combined the ssn with some of the new as well as existing variables to create additional unique identifiers.

In this project, we focused on two scenarios of fraud. The first is one in which the fraudster has access to multiple core identifications(ssn, name and date of birth, etc) and the second is when one individual's core identities have been compromised and multiple fraudsters are using it. To capture the first type of fraud, we create variables that count how long it has been since we have last seen a certain non-core identification in our data. To account for the second type of fraud, we create variables that count how long it has been since we have last seen a core identification in the data. Next we create a variable that tracks how many days a certain variable appears over the preceding {0,1,3,7,14, and 30} days.. We do this as we suspect it may be more suspicious if we see repetition of a certain identification being used in a short period of time.

Next we created variables to measure relative velocity of the identifications. This was done by looking at how often a given identification was seen over the same day or over one day in the past versus how often it was seen over the past three days, week, two weeks, or 30 days. An example would be 'address_count_0_by_30' which would tell us how many times an address has been seen in the same day versus how many times that same address has been seen in the past 30 days. See the appendix for the full list of engineered features.

Feature Selection

In the previous section, we discussed the creation of myriad candidate variables. However, to use every field as a variable in our final model candidates would not only be computationally expensive, but would also introduce immense dimensionality, which in turn would increase sparsity, make all points outliers, and obscure important non-linear data patterns. To reduce dimensionality to a more manageable set of input variables, we employed a feature selection process consisting of two main mechanisms: a filter and a wrapper.

Filter

Since we began with 375 candidate variables, we first utilized a filter process to substantially reduce dimensionality in a fast and computationally inexpensive way. We decided to keep the top 80 variables as ranked by our filter. The exact number of variables to keep after the filtering process is arbitrary, but the goal is to err on the safe side; since the filters are simple measures of variable goodness, it's best to leave plenty of margin for the good variables to make it through the filter. The filter itself consisted of the average of two different scores: a Kolmogorov-Smirnov (KS) score and a Fraud Detection Rate at 3%.

When used for a binary classification problem, a KS score constructs two normalized distributions for each field. The distributions represent the density of records belonging to a given class for each value of the field in question. The score itself then measures how separate the two distributions are from each other. Said another way, a KS score is a simple way to measure how well a variable distinguishes between goods and bads, which is the key to a variable being helpful in a binary classification model.

The univariate fraud detection rate (FDR) at 3% indicates the percentage of fraudulent records that lie in the tails of a field's data distribution. To calculate univariate FDR at 3%, we take a subset of the data that includes only the field of interest and the fraud label for each record. We then sort the subset by the field of interest and take the top and bottom 3% of those records. Finally, we count how many frauds we can detect in those 3% subsets and divide the number detected by the total number of frauds in the dataset. We then take the maximum detection rate between the top and bottom of the distribution, and this maximum value represents the univariate FDR at 3% for the field of interest.

We used these two filter methods to assign each field its own KS and FDR score. Then, we assigned each field a rank corresponding to those scores, and took the average of the two ranks as the final score. For example, if field A had the fifth highest KS score and 9th highest FDR score, its average rank would be the total number of variables - 7 (in our specific case, 368). We then sorted all the fields by this final score in descending order and kept only the top 80 fields.

Wrapper

To further narrow down dimensionality, a more intelligent selection method is preferable, since more nuanced differences in importance to the model are harder to detect among the best variables. To distinguish which 30 candidate variables of the 80 remaining were best to use for our model, we employed a wrapper called Recursive Feature Elimination with Cross Validation (RFECV).

Wrappers are models "wrapped" around the feature selection process, and generally follow filters in the feature selection process. Our wrapper started by building a logistic regression model with all 80 variables, and then built 80 additional logistic regression models, each with one of the variables missing from the model. The variable missing from the model that experienced the *least* decay in performance across multiple cross validation folds from the original model in the step before was removed by the wrapper. The wrapper continued this process until it had selected 30 variables that were most important in predicting fraud.

With our 30 final variables selected, we subset our original training, testing, and out-of-time data to include only those 30 fields along with the fraud label for each record. Combined, the three datasets contain 31 fields with 1 million observations each.

Ranking the Final Variables in Order of Importance

The logistic regression wrapper returned the 30 variables listed below. The order the variables are listed in corresponds to that variable's relative importance in distinguishing between fraudulent and non-fraudulent records. This rank order of importance was determined using a decision tree model run on the 30 variables provided by the wrapper. Decision trees are excellent for ranking variable importance, because the model can simply rank variables by how high up on the tree they are. In our case, the first cut was made on the name_dob_count_30 dimension, and the last cut was made on the ssn_dob_count_30 variable. These dimensions are consequently ranked first and last in importance among the 30 fed initially into the model.

name_dob_count_30	The number of times a record's date of birth has been seen in the last 30 days
fulladdress_count_7	Number of times the record's full address (which combines the street address and the zipcode) has been seen in the last week
fulladdress_count_0_by_30	Number of times the record's full address (which combines the street address and the zipcode) has been seen in the same day relative to the number of times it's been seen in the last 30 days

homephone_count_3	The number of times a record's home phone number has been seen in the last 3 days
zip5_count_1	The number of times a record's zip code has been seen in the same day and the day before
homephone_count_7	The number of times a record's home phone number has been seen in the last week
fulladdress_count_1_by_7	Number of times the record's full address (which combines the street address and the zipcode) has been seen in the same day and day before relative to the number of times it's been seen in the last week
ssn_count_7	The number of times a record's social security number has been seen in the last week
fulladdress_count_1	Number of times the record's full address (which combines the street address and the zipcode) has been seen in the same day and the day before
address_count_0	Number of times the record's address has been seen in the same day
fulladdress_homephone_count_30	The number of times a record's combination of full address and home phone number has been seen in the last 30 days
fulladdress_homephone_count_14	The number of times a record's combination of full address and home phone number has been seen in the last 14 days
ssn_firstname_count_7	The number of times a record's combination of social security number and first name has been seen in the last week
fulladdress_homephone_count_3	The number of times a record's combination of full address and home phone number has been seen in the last 3 days
fulladdress_homephone_count_7	The number of times a record's combination of full address and home phone number has been seen in the last week
address_count_7	Number of times the record's address has been seen in the past week
ssn_name_dob_count_30	The number of times a record's combination of social security number, full name, and date of birth has been seen in the last 30 days
fulladdress_count_3	Number of times the record's full address (which combines the street address and the zipcode) has been seen in the last 3 days
address_count_0_by_3	Number of times the record's full address (which combines the street address and the zipcode) has been seen in the same day relative to the number of times its been seen in the last 3 days

ssn_firstname_count_30	The number of times a record's combination of social security number and first name has been seen in the last 30 days
fulladdress_homephone_count_0_by_30	The number of times a record's combination of full address and home phone number has been seen in the same day relative to the number of times that combination has been recorded in the last 30 days
fulladdress_count_0_by_3	Number of times the record's full address (which combines the street address and the zipcode) has been seen in the same day relative to the number of times it's been seen in the last 3 days
address_count_0_by_7	Number of times the record's address has been seen in the same day relative to the number of times it's been seen in the last week
ssn_name_dob_count_14	The number of times a record's combination of social security number, full name, and date of birth has been seen in the last 14 days
name_dob_count_0_by_30	The number of times a record's combination of full name and date of birth has been seen in the same day relative to the number of times it's been seen in the last 30 days
ssn_count_14	The number of times a record's social security number has been seen in the last 14 days
name_dob_count_14	The number of times a record's combination of full name and date of birth has been seen in the last 14 days
name_dob_count_0_by_14	The number of times a record's combination of full name and date of birth has been seen in the same day relative to the number of times it's been seen in the last 14 days
ssn_lastname_count_7	The number of times a record's combination of social security number and last name has been seen in the last week
ssn_dob_count_30	The number of times a record's combination of social security number and date of birth has been seen in the last 30 days

Modeling

In practice, it's always best to start the modeling process with a simple linear model. All subsequent non-linear models' performances can then be compared to the original linear model baseline performance. We started by building a logistic regression model with all 30 variables and default hyperparameter settings to obtain this baseline model.

Logistic Regression

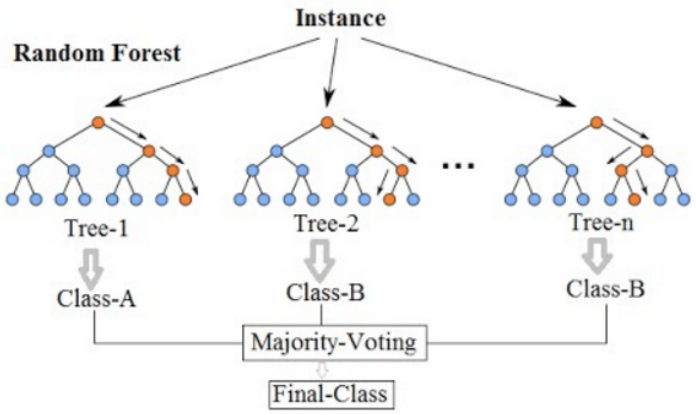
Logistic regression is the most commonly used linear classification model. It works by maximizing a likelihood function to estimate the coefficient parameters for each input variable in a sigmoid function which takes values between 0 and 1. Specifically, the logistic regression equation sets P(Y=1 | X= x) equal to e raised to the power of each input value multiplied by each coefficient value divided by 1 + e raised to the same power. By seeing all the training record input values, logistic regression can estimate the coefficients that best allow the function to output high probabilities for actual class 1 records.

Next, we ran some non-linear models in an attempt to improve model performance on the test and out-of-time (OOT) data. Following are descriptions of the algorithms we trained.

Random Forest

A single decision tree models a data surface in discrete steps, and works by testing candidate cut points for each dimension in the model. At each candidate cutpoint, the impurity across the cut is stored, and so the first cut the tree makes is on the dimension that reduced impurity the most with its best candidate cutpoint. Then the tree continues to carve the remaining variable space, repeating the aforementioned process separately in each resulting data space after a cut is made.

A random forest in turn is an ensemble of strong decision trees. A tree's strength is primarily determined by its depth, and therefore its complexity. A strong tree can have many layers and partition the data space in complex and highly-fitted ways. A single strong tree is prone to overfitting the data by making cuts that are too specifically tailored to the training set, and therefore won't perform well in general on test data. Random forests both reduce the variability present in a single tree and ameliorate overfitting issues by, in classification problems, gathering "votes" from each tree in the ensemble for each test record. Each tree outputs which class it thinks is most likely for each test record, and the majority class vote from all trees gets assigned to that record.

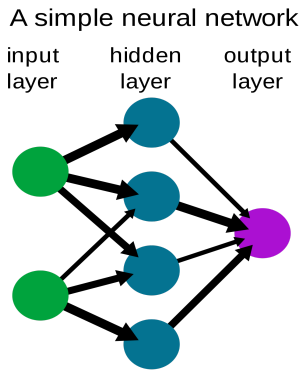


Some of the hyperparameters we adjusted when training random forest models include the number of trees in the ensemble, the maximum depth of each tree (fairly deep for random forests), the minimum number of samples required on either side of a partition for that split to occur in a given tree, and the criterion for evaluating the best split (either gini or entropy).

Neural Network

A neural network (neural net for short) is an algorithm designed to mimic the brain. Usually, neural nets consist of an input layer, a number of hidden layers greater than or equal to one, and an output layer, which in the case of a classification problem is the model's best guess at the record's class. Nodes in the hidden layer are analogous to neurons, and are sent signals from previous layers in the network that can be compared to impulses traveling along axons and synapses.

Each node performs a transformation or "activation" function on the linear combination of signals it receives from the previous layer, and outputs another signal to the next layer. The input signals are weighted, and the neural net trains by back-propagating errors and re-weighting the nodes. A training epoch constitutes one pass through the entire dataset. Eventually, each node will settle into a local optimum weight, and the model reaches a stable state.

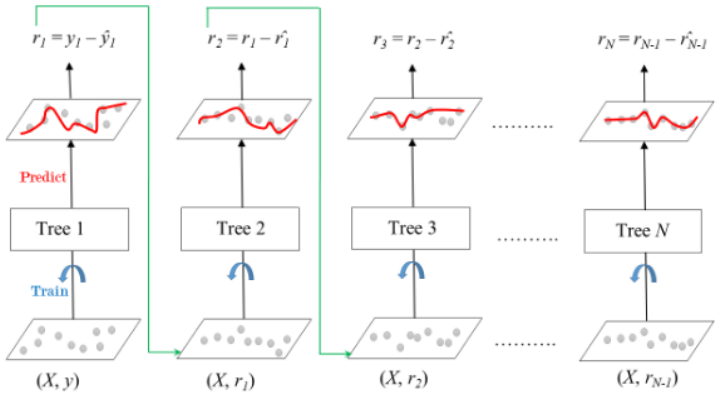


Some hyperparameters we adjusted when training neural nets include the number of hidden layers, the activation function type, the learning rate (which attempts to guide nodes into global optima over the course of training), the number of nodes per hidden layer, and the maximum number of training epochs.

Boosted Tree: Gradient Boosted Machine

Boosted trees, like random forest algorithms, use an ensemble of decision trees to formulate a final classification decision for each record. However, unlike random forests, boosted trees use far more trees, each with far less depth. Additionally, boosted trees hone in on errors made in previous trees and target those errors in the next tree. Because each tree is a weak learner, the algorithm is highly robust against overfitting.

Gradient boosting is a specific type of boosted tree algorithm that uses a gradient descent procedure to increasingly minimize the loss function with each new tree. Some hyperparameters that we tweaked for our GBM model were the number of trees used, the max depth of each tree, the loss function used, and the learning rate.



Model Algorithms

The following table shows the different models with various hyperparameters used. The green cell corresponds to the top performed value of hyperparameter for each model.

Model	Parameter				FDR at 3%		
Logistic Regression	Total Variables				TRAIN	TEST	OOT
1	30				0.557	0.576	0.536
2	5				0.348	0.366	0.321
3	10				0.349	0.366	0.323
4	15				0.375	0.395	0.345
5	20				0.515	0.532	0.504
Random Forest	Estimators	Max depth	criterion		TRAIN	TEST	OOT
1	50	100	gini		0.579	0.590	0.552
2	100	100	gini		0.579	0.588	0.552
3	150	100	gini		0.579	0.589	0.552
4	100	100	entropy		0.579	0.587	0.553
5	50	500	gini		0.579	0.587	0.552
6	50	250	gini		0.521	0.531	0.499
7	50	100	entropy		0.521	0.531	0.501
Neural Networks	layers	Max iterations	learning rate	activation	TRAIN	TEST	OOT
1	5	20	constant	relu	0.565	0.581	0.539
2	10	30	constant	relu	0.562	0.580	0.536
3	15	40	constant	relu	0.568	0.585	0.545
4	10	40	adaptive	relu	0.570	0.585	0.550
5	5	50	constant	tanh	0.566	0.583	0.539
Boosted Trees	Estimators	Max depth	loss	learning rate	TRAIN	TEST	OOT
1	600	3	deviance	0.1	0.5760	0.5905	0.5536
2	800	4	deviance	0.1	0.5761	0.5911	0.5532
3	600	5	exponential	0.1	0.5786	0.5880	0.5524
4	600	5	deviance	0.01	0.5687	0.5687	0.5440
5	1200	6	deviance	0.01	0.5580	0.5580	0.5377

Results

We selected our best performing model, a gradient boosted machine, which obtained a fraud detection rate at 3% of 0.537 on the OOT data. The model ran with the following hyperparameters:

- Estimators = 800
- Max depth = 4
- Loss function = deviance
- Learning rate = 0.1 and score all our records and sort the records by this score. We then bin the sorted distribution up into 1% population bins.

The results below show the top 30 population bin statistics and cumulative statistics for our best performing model. We've included one table for each of Training results, Testing results, and OOT results.

In each table, one can observe a high fraud detection rate in the top population bins, meaning that our model is doing a good and concise job of pushing the fraudulent records to the top.

Training Set Results

Training	# Records		# Goods		# Bads		Fraud Rate					
	583454		575049		8405		0.014616146					
	Bin Statistics					Cumulative Statistics						
Population Bin %	records	goods	Bads	% Goods	% Bads	Total # Records	Cumulative Goods	Cumulative Bads	%Goods	%Bads (FDR)	KS	FPR
1	5834	1361	4473	23.33%	76.67%	5834	1361	4473	0.24%	53.56%	53.33	0.30
2	5834	5610	224	96.16%	3.84%	11668	6971	4697	1.21%	56.24%	55.03	1.48
3	5834	5746	88	98.49%	1.51%	17502	12717	4785	2.21%	57.30%	55.09	2.66
4	5834	5767	67	98.85%	1.15%	23336	18484	4852	3.21%	58.10%	54.89	3.81
5	5834	5779	55	99.06%	0.94%	29170	24263	4907	4.22%	58.76%	54.54	4.94
6	5834	5779	55	99.06%	0.94%	35004	30042	4962	5.22%	59.42%	54.19	6.05
7	5834	5782	52	99.11%	0.89%	40838	35824	5014	6.23%	60.04%	53.81	7.14
8	5834	5793	41	99.30%	0.70%	46672	41617	5055	7.24%	60.53%	53.29	8.23
9	5834	5781	53	99.09%	0.91%	52506	47398	5108	8.24%	61.17%	52.92	9.28
10	5834	5788	46	99.21%	0.79%	58340	53186	5154	9.25%	61.72%	52.47	10.32
11	5834	5793	41	99.30%	0.70%	64174	58979	5195	10.26%	62.21%	51.95	11.35
12	5834	5786	48	99.18%	0.82%	70008	64765	5243	11.26%	62.78%	51.52	12.35
13	5834	5786	48	99.18%	0.82%	75842	70551	5291	12.27%	63.36%	51.09	13.33
14	5834	5800	34	99.42%	0.58%	81676	76351	5325	13.28%	63.76%	50.49	14.34
15	5834	5792	42	99.28%	0.72%	87510	82143	5367	14.28%	64.27%	49.98	15.31
16	5834	5807	27	99.54%	0.46%	93344	87950	5394	15.29%	64.59%	49.30	16.31
17	5834	5785	49	99.16%	0.84%	99178	93735	5443	16.30%	65.18%	48.88	17.22
18	5834	5801	33	99.43%	0.57%	105012	99536	5476	17.31%	65.57%	48.26	18.18
19	5834	5805	29	99.50%	0.50%	110846	105341	5505	18.32%	65.92%	47.60	19.14
20	5834	5792	42	99.28%	0.72%	116680	111133	5547	19.33%	66.42%	47.10	20.03
21	5834	5799	35	99.40%	0.60%	122514	116932	5582	20.33%	66.84%	46.51	20.95
22	5834	5799	35	99.40%	0.60%	128348	122731	5617	21.34%	67.26%	45.92	21.85
23	5834	5801	33	99.43%	0.57%	134182	128532	5650	22.35%	67.66%	45.31	22.75
24	5834	5799	35	99.40%	0.60%	140016	134331	5685	23.36%	68.08%	44.72	23.63
25	5834	5796	38	99.35%	0.65%	145850	140127	5723	24.37%	68.53%	44.16	24.48
26	5834	5804	30	99.49%	0.51%	151684	145931	5753	25.38%	68.89%	43.51	25.37
27	5834	5806	28	99.52%	0.48%	157518	151737	5781	26.39%	69.23%	42.84	26.25
28	5834	5801	33	99.43%	0.57%	163352	157538	5814	27.40%	69.62%	42.22	27.10
29	5834	5797	37	99.37%	0.63%	169186	163335	5851	28.40%	70.06%	41.66	27.92
30	5834	5788	46	99.21%	0.79%	175020	169123	5897	29.41%	70.61%	41.20	28.68

Test Set Results

Testing	# Records	# Goods	# Bads	Fraud Rate
	250053	246451	3602	0.014615481

	Bin Statistics					Cumulative Statistics						
Population Bin %	records	goods	Bads	% Goods	% Bads	Total # Records	Cumulative Goods	Cumulative Bads	%Goods	%Bads (FDR)	KS	FPR
1	2500	585	1915	23.40%	76.60%	2500	585	1915	0.24%	53.96%	53.72	0.31
2	2500	2379	121	95.16%	4.84%	5000	2964	2036	1.20%	57.37%	56.17	1.46
3	2500	2460	40	98.40%	1.60%	7500	5424	2076	2.20%	58.50%	56.29	2.61
4	2500	2469	31	98.76%	1.24%	10000	7893	2107	3.20%	59.37%	56.17	3.75
5	2500	2474	26	98.96%	1.04%	12500	10367	2133	4.21%	60.10%	55.89	4.86
6	2500	2478	22	99.12%	0.88%	15000	12845	2155	5.21%	60.72%	55.51	5.96
7	2500	2476	24	99.04%	0.96%	17500	15321	2179	6.22%	61.40%	55.18	7.03
8	2500	2477	23	99.08%	0.92%	20000	17798	2202	7.22%	62.05%	54.82	8.08
9	2500	2482	18	99.28%	0.72%	22500	20280	2220	8.23%	62.55%	54.32	9.14
10	2500	2484	16	99.36%	0.64%	25000	22764	2236	9.24%	63.00%	53.77	10.18
11	2500	2482	18	99.28%	0.72%	27500	25246	2254	10.24%	63.51%	53.27	11.2
12	2500	2488	12	99.52%	0.48%	30000	27734	2266	11.25%	63.85%	52.6	12.24
13	2500	2483	17	99.32%	0.68%	32500	30217	2283	12.26%	64.33%	52.07	13.24
14	2500	2478	22	99.12%	0.88%	35000	32695	2305	13.27%	64.95%	51.68	14.18
15	2500	2478	22	99.12%	0.88%	37500	35173	2327	14.27%	65.57%	51.3	15.12
16	2500	2481	19	99.24%	0.76%	40000	37654	2346	15.28%	66.10%	50.82	16.05
17	2500	2477	23	99.08%	0.92%	42500	40131	2369	16.28%	66.75%	50.47	16.94
18	2500	2485	15	99.40%	0.60%	45000	42616	2384	17.29%	67.17%	49.88	17.88
19	2500	2483	17	99.32%	0.68%	47500	45099	2401	18.30%	67.65%	49.35	18.78
20	2500	2487	13	99.48%	0.52%	50000	47586	2414	19.31%	68.02%	48.71	19.71
21	2500	2481	19	99.24%	0.76%	52500	50067	2433	20.32%	68.55%	48.24	20.58
22	2500	2488	12	99.52%	0.48%	55000	52555	2445	21.32%	68.89%	47.57	21.49
23	2500	2491	9	99.64%	0.36%	57500	55046	2454	22.34%	69.15%	46.81	22.43
24	2500	2484	16	99.36%	0.64%	60000	57530	2470	23.34%	69.60%	46.25	23.29
25	2500	2480	20	99.20%	0.80%	62500	60010	2490	24.35%	70.16%	45.81	24.1
26	2500	2490	10	99.60%	0.40%	65000	62500	2500	25.36%	70.44%	45.08	25
27	2500	2484	16	99.36%	0.64%	67500	64984	2516	26.37%	70.89%	44.53	25.83
28	2500	2486	14	99.44%	0.56%	70000	67470	2530	27.38%	71.29%	43.91	26.67
29	2500	2479	21	99.16%	0.84%	72500	69949	2551	28.38%	71.88%	43.5	27.42
30	2500	2485	15	99.40%	0.60%	75000	72434	2566	29.39%	72.30%	42.91	28.23

Out of Time Set Results

OOT	# Records	# Goods	# Bads	Fraud Rate
	166493	164107	2386	0.014539294

	Bin Statistics					Cumulative Statistics						
Population Bin %	records	goods	Bads	% Goods	% Bads	Total # Records	Cumulative Goods	Cumulative Bads	%Goods	%Bads (FDR)	KS	FPR
1	1664	514	1150	30.89%	69.11%	1664	514	1150	0.31%	50.00%	49.69	0.45
2	1664	1608	56	96.63%	3.37%	3328	2122	1206	1.29%	52.43%	51.14	1.76
3	1664	1635	29	98.26%	1.74%	4992	3757	1235	2.29%	53.70%	51.41	3.04
4	1664	1650	14	99.16%	0.84%	6656	5407	1249	3.29%	54.30%	51.01	4.33
5	1664	1652	12	99.28%	0.72%	8320	7059	1261	4.30%	54.83%	50.52	5.60
6	1664	1651	13	99.22%	0.78%	9984	8710	1274	5.31%	55.39%	50.08	6.84
7	1664	1651	13	99.22%	0.78%	11648	10361	1287	6.31%	55.96%	49.64	8.05
8	1664	1648	16	99.04%	0.96%	13312	12009	1303	7.32%	56.65%	49.33	9.22
9	1664	1648	16	99.04%	0.96%	14976	13657	1319	8.32%	57.35%	49.03	10.35
10	1664	1654	10	99.40%	0.60%	16640	15311	1329	9.33%	57.78%	48.45	11.52
11	1664	1641	23	98.62%	1.38%	18304	16952	1352	10.33%	58.78%	48.45	12.54
12	1664	1657	7	99.58%	0.42%	19968	18609	1359	11.34%	59.09%	47.75	13.69
13	1664	1656	8	99.52%	0.48%	21632	20265	1367	12.35%	59.43%	47.09	14.82
14	1664	1648	16	99.04%	0.96%	23296	21913	1383	13.35%	60.13%	46.78	15.84
15	1664	1653	11	99.34%	0.66%	24960	23566	1394	14.36%	60.61%	46.25	16.91
16	1664	1651	13	99.22%	0.78%	26624	25217	1407	15.37%	61.17%	45.81	17.92
17	1664	1658	6	99.64%	0.36%	28288	26875	1413	16.38%	61.43%	45.06	19.02
18	1664	1655	9	99.46%	0.54%	29952	28530	1422	17.39%	61.83%	44.44	20.06
19	1664	1653	11	99.34%	0.66%	31616	30183	1433	18.39%	62.30%	43.91	21.06
20	1664	1650	14	99.16%	0.84%	33280	31833	1447	19.40%	62.91%	43.51	22.00
21	1664	1653	11	99.34%	0.66%	34944	33486	1458	20.41%	63.39%	42.99	22.97
22	1664	1650	14	99.16%	0.84%	36608	35136	1472	21.41%	64.00%	42.59	23.87
23	1664	1648	16	99.04%	0.96%	38272	36784	1488	22.42%	64.70%	42.28	24.72
24	1664	1653	11	99.34%	0.66%	39936	38437	1499	23.42%	65.17%	41.75	25.64
25	1664	1655	9	99.46%	0.54%	41600	40092	1508	24.43%	65.57%	41.13	26.59
26	1664	1658	6	99.64%	0.36%	43264	41750	1514	25.44%	65.83%	40.38	27.58
27	1664	1659	5	99.70%	0.30%	44928	43409	1519	26.45%	66.04%	39.59	28.58
28	1664	1655	9	99.46%	0.54%	46592	45064	1528	27.46%	66.43%	38.97	29.49
29	1664	1660	4	99.76%	0.24%	48256	46724	1532	28.47%	66.61%	38.14	30.50
30	1664	1659	5	99.70%	0.30%	49920	48383	1537	29.48%	66.83%	37.34	31.48

Conclusion

In this project we aimed to predict fraud in Credit Card Applications. As our data was labelled, this was a supervised learning task. Our approach encompassed most of the end-to-end data science framework, including data cleaning, feature engineering, feature selection, modelling and evaluation.

We started off with exploratory data analysis in the form of a data quality report, providing insights such as summary statistics on every variable. Following this, we proceeded onto our feature engineering stage where we created 365 additional variables. We then scaled the variables to ensure they contributed equally to the models. For the feature selection, we applied a filter based on the univariate KS score and fraud detection rate scores to first reduce our candidate variables to 80 and then a logistic regression backward selection wrapper to finally reduce our variables to 30.

In the modelling stage, we implemented a range of algorithms, starting with a linear model: logistic regression as our baseline. We then implemented some non-linear models including Random Forests, Neural Networks and Gradient Boosted Trees to see which would perform the best or in other words to see which model best predicted product application fraud.

As stated above in the results section, our best model was a Gradient Boosted Tree (with a FDR at 3% of 0.537 on the Out of Time Data).

With more time and resources, we could have gone deeper into the hyperparameter tuning of our current models and explored a wider range of models to see if we could have further improved the model performance. Additionally, if we didn't have computing restraints, we could have supercharged our feature engineering stage, by potentially creating 1000s of additional variables, which likely would have led to greater model performance.

Appendix

DQR

Data Description

The data contained in the file 'applications data.csv' comprise a synthetic data set built from studying the statistical properties of more than a billion applications. The dataset is designed to reproduce the important univariate and multivariate field distributions of real credit application data and is sourced from an identity fraud prevention company. The data covers the full year of 2016 from the first day to the last and contains 1,000,000 (one million) records.

Data Summary

Date Fields Summary Table

Field Name	Field Type	# of Records	% Populated	# Zeros	Earliest Date	Latest Date	Most Common Date
date	datetime	1000000	100	0	1/1/2016	12/31/2016	8/16/2016
dob	datetime	1000000	100	0	1/1/1900	10/31/2016	6/26/1907

Categorical Fields Summary Table

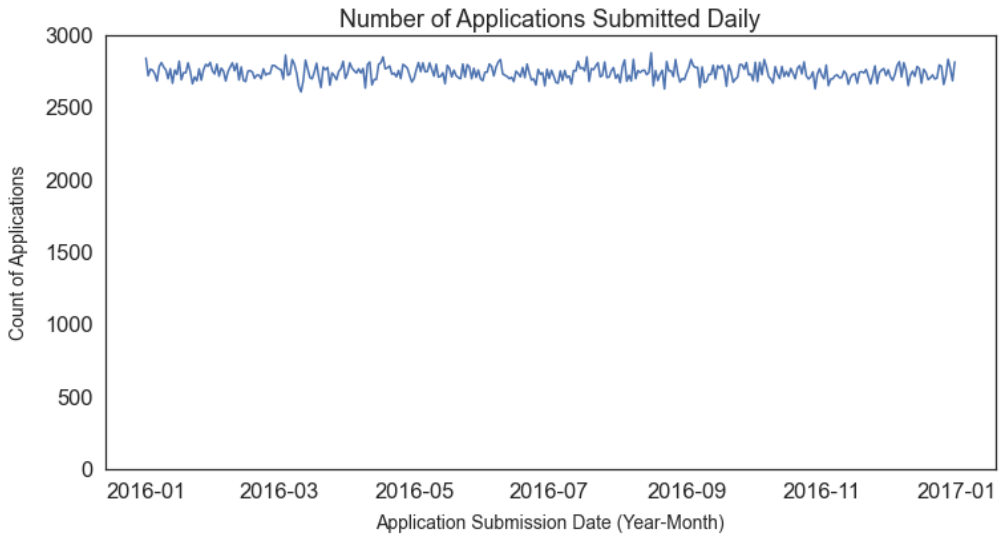
Field Name	Field Type	# of Records	% Populated	# Zeros	# Unique Values	Most Common Value
record	categorical	1,000,000	100	0	1000000	N/A
ssn	categorical	1,000,000	100	0	835819	999999999
firstname	categorical	1,000,000	100	0	78136	EAMSTRMT
lastname	categorical	1,000,000	100	0	177001	ERJSAXA
address	categorical	1,000,000	100	0	828774	123 MAIN ST
zip5	categorical	1,000,000	100	0	26370	68138
homephone	categorical	1,000,000	100	0	28244	9999999999
fraud_label	categorical	1,000,000	100	985607	2	0

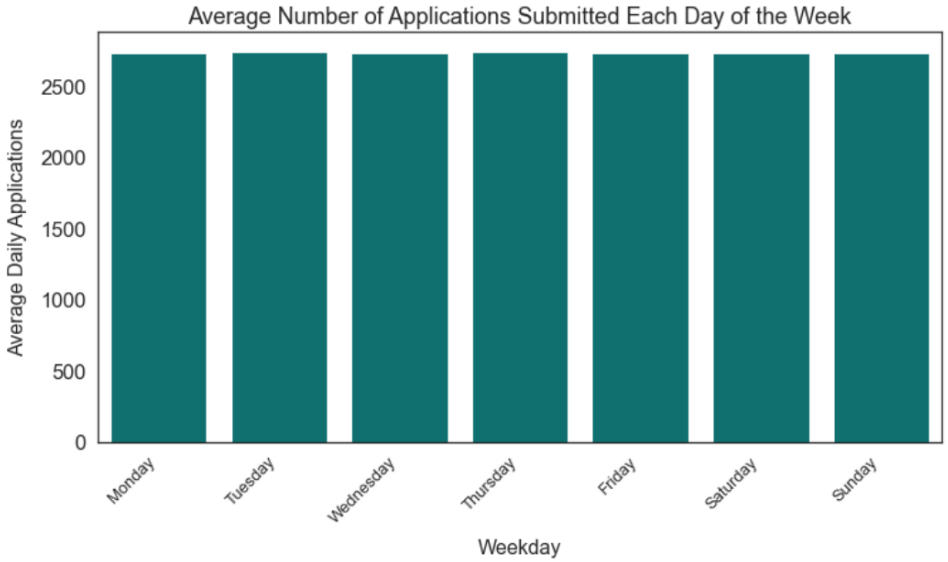
Field Descriptions and Distributions

Please note that in this report, the fields are listed in the same order in which they appear in the dataset, from left to right.

Field 1
Name: record
Description: An integer identifier for each row in the dataset. The record field is unique for each row and therefore does not merit a distribution inspection.

Field 2
Name: date
Description: The date (in year, month, day format) on which the application corresponding to the record was submitted. The date field spans the entire year of 2016.

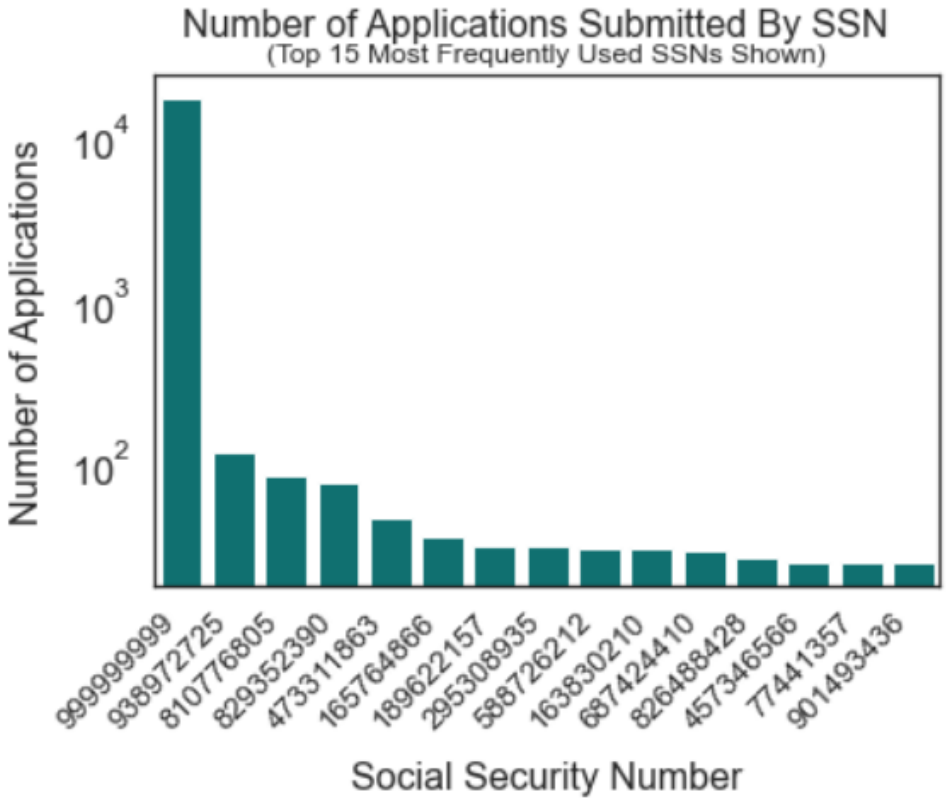




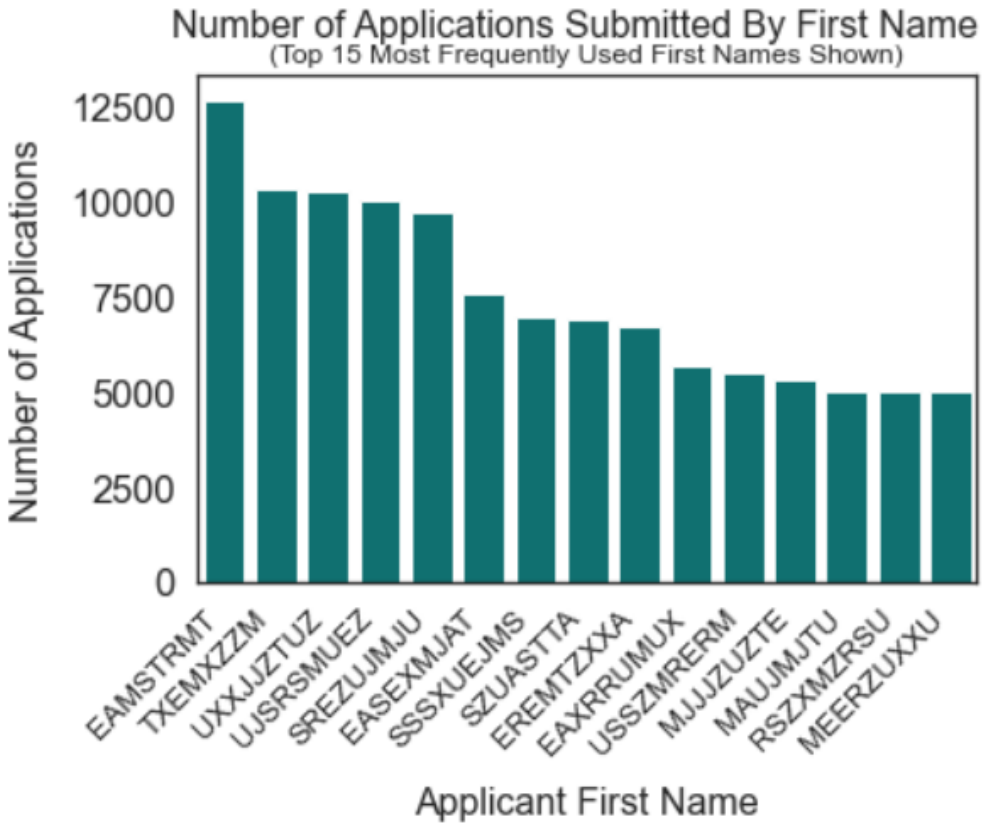
Field 3

Name: ssn

Description: The (usually) nine-digit social security number submitted by the applicant. Social security number should serve as a unique identifier for all American citizens, but this is not always the case since anyone can enter a fake SSN.



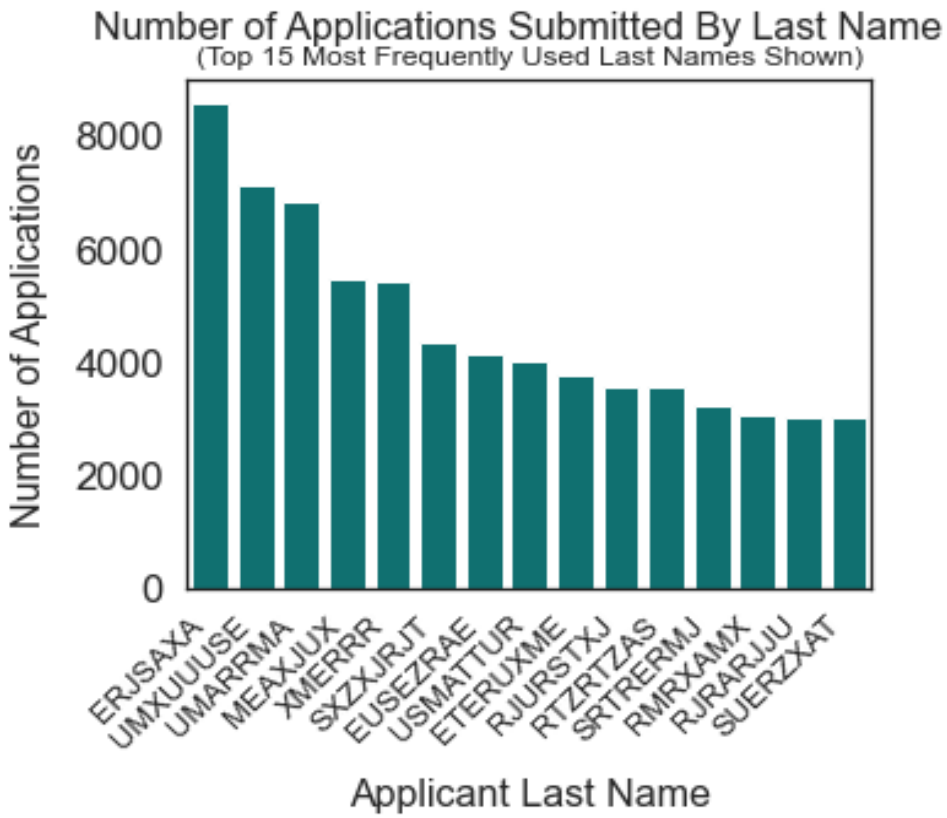
Field 4
Name: firstname
Description: The first name entered by the applicant.



Field 5

Name: lastname

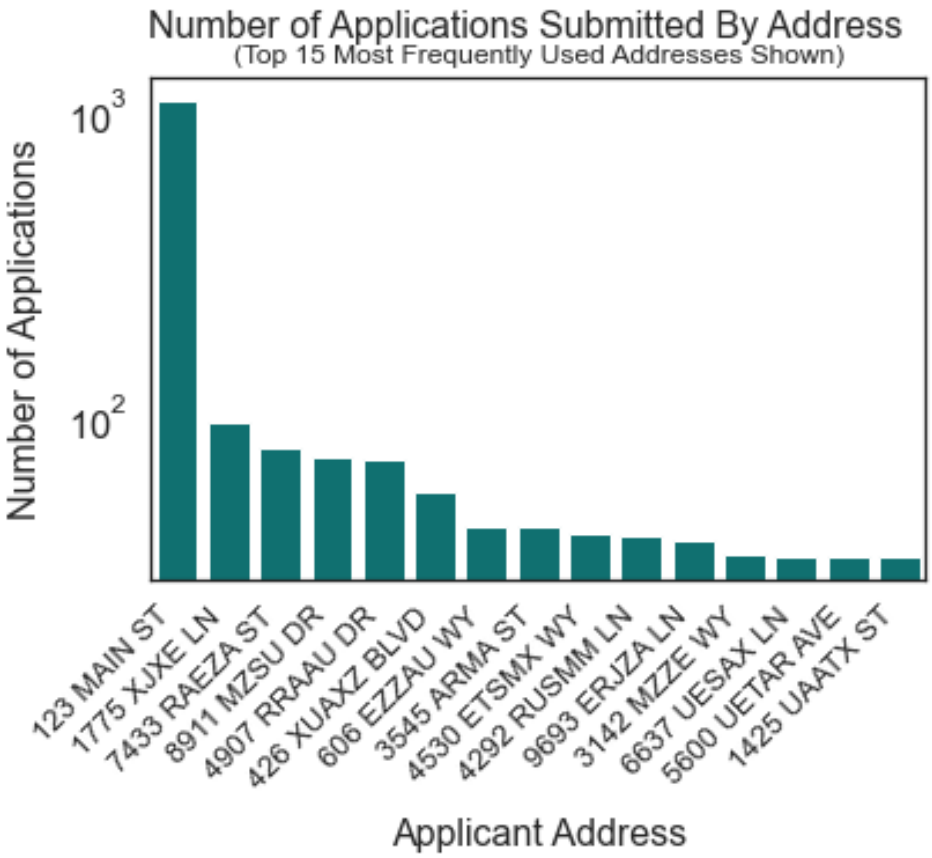
Description: The last name entered by the applicant.



Field 6

Name: address

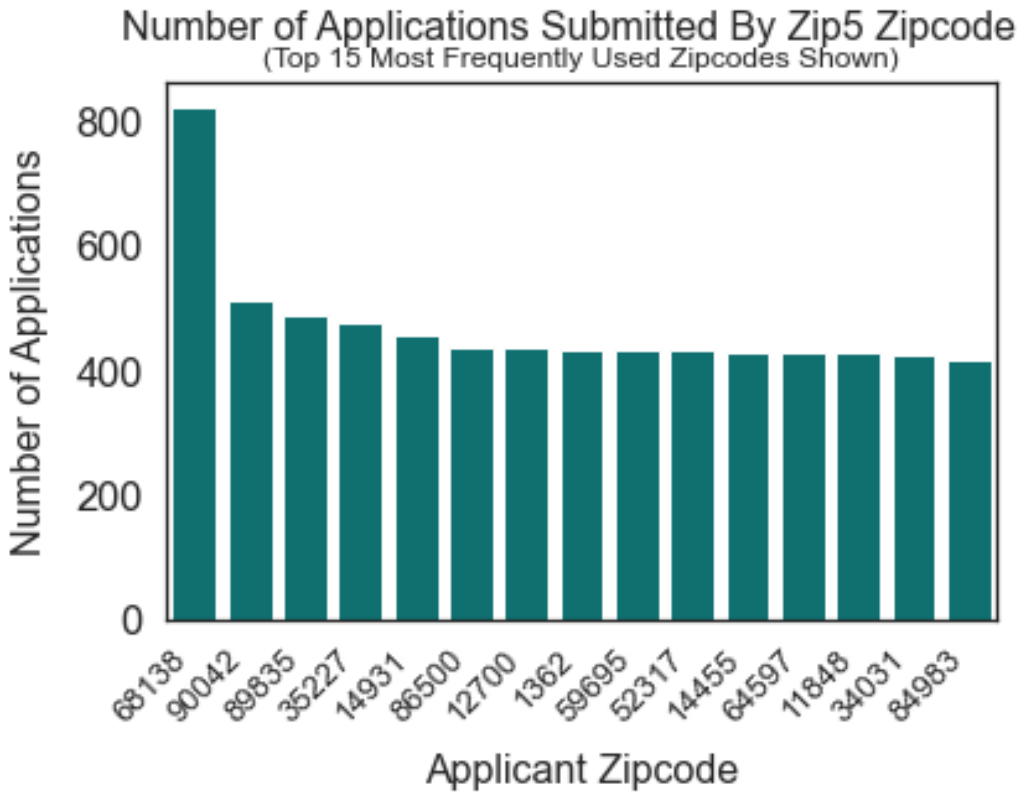
Description: The home street address (not including city, state, or zip code) submitted by the applicant.



Field 7

Name: zip5

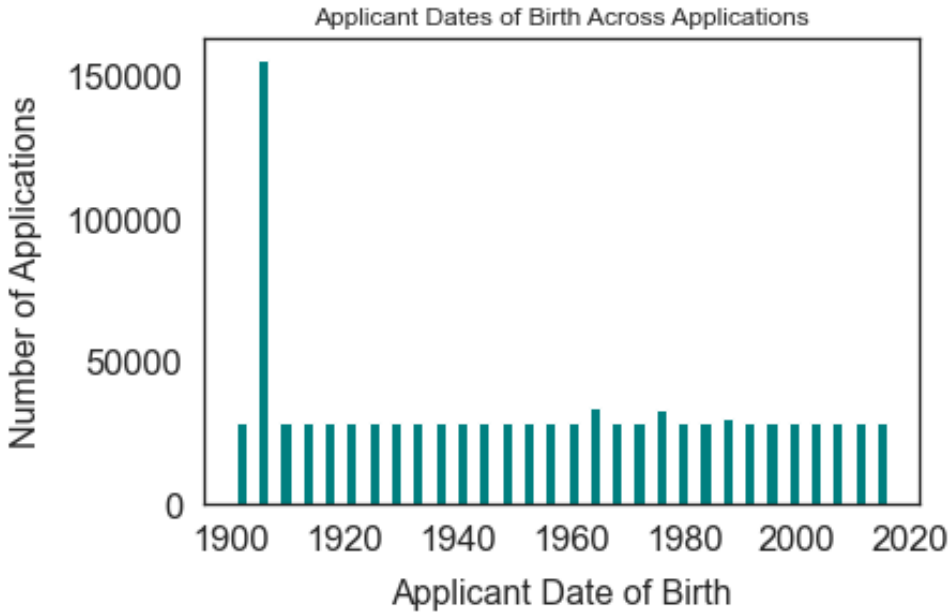
Description: The 5 digit zip code submitted by the applicant. Every street address in the United States has a corresponding 5-digit zip code.

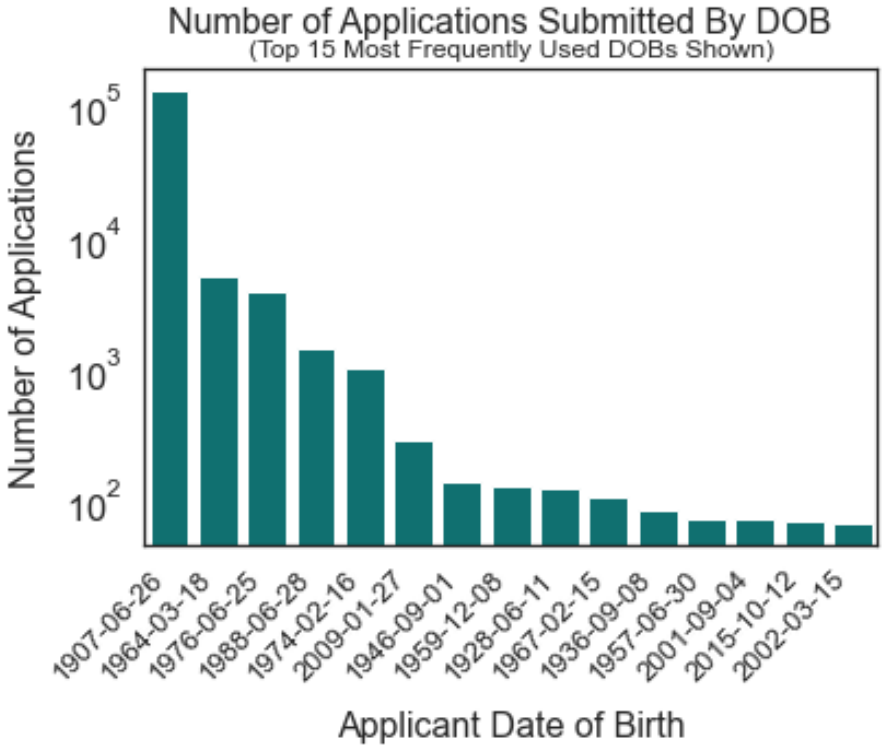


Field 8

Name: dob

Description: The date (in year, month, day format) on which the applicant claims they were born. Date of birth ranges from January 1st, 1900 to October 31, 2016.

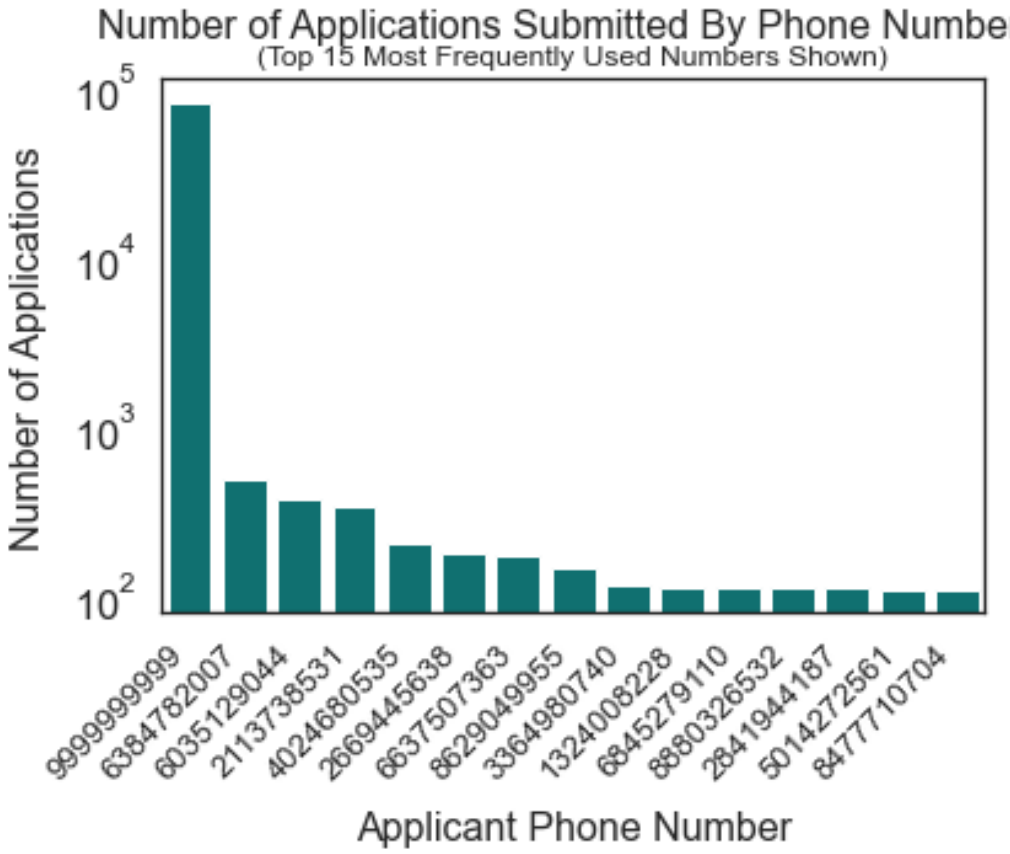




Field 9

Name: homephone

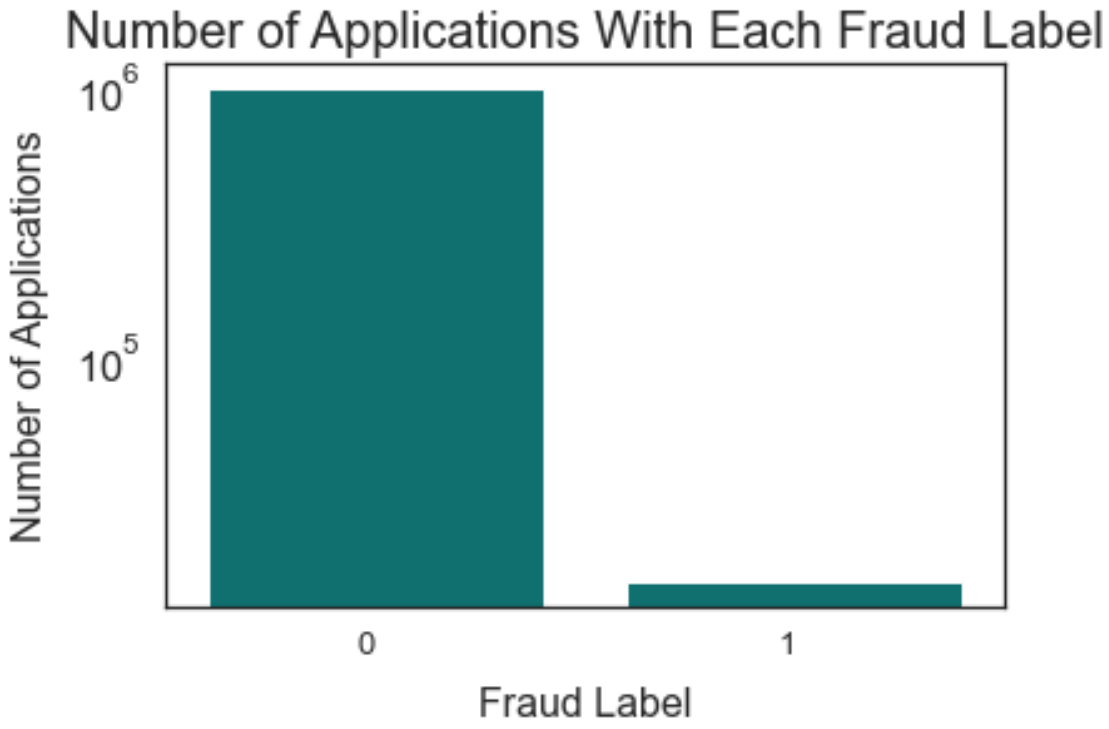
Description: The home phone number listed by the applicant. While U.S. phone numbers are typically nine or ten digits, the numbers submitted by applicants don't always adhere to this convention.

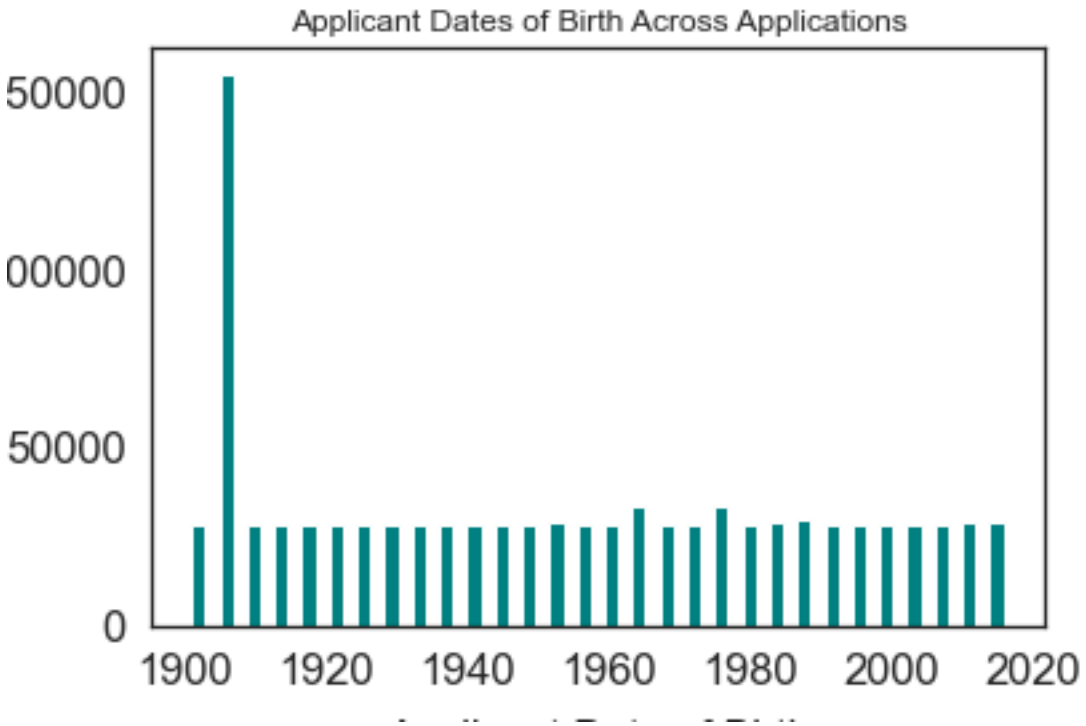


Field 10

Name: fraud_label

Description: A binary variable that can take values of 0, corresponding to no fraud, and 1, corresponding to fraud. Most of the values for this field take 0, meaning most of the records were labelled as non-fraudulent applications, which is typical for a dataset of this nature. The fraud_label field can be used as the target variable for a supervised model trained to detect fraudulent applications.





List of Candidate Variables

Following is a list of the variables we considered for our model. This list is all inclusive and is prior to any feature selection.

	Variable
1	record
2	date
3	ssn
4	firstname
5	lastname
6	address
7	zip5
8	dob
9	homephone

188	ssn_name_homephone_count_3
189	ssn_name_homephone_count_7
190	ssn_name_homephone_count_14
191	ssn_name_homephone_count_30
192	ssn_count_0_by_3
193	ssn_count_0_by_7
194	ssn_count_0_by_14
195	ssn_count_0_by_30
196	ssn_count_1_by_3

10	fraud_label
11	name
12	fulladdress
13	name_dob
14	name_fulladdress
15	name_homephone
16	fulladdress_dob
17	fulladdress_homephone
18	dob_homephone
19	homephone_name_dob
20	ssn_firstname
21	ssn_lastname
22	ssn_address
23	ssn_zip5
24	ssn_dob
25	ssn_homephone
26	ssn_name_dob
27	ssn_name_fulladdress
28	ssn_name_homephone
29	dow
30	dow_risk
31	ssn_day_since
32	ssn_count_0
33	ssn_count_1
34	ssn_count_3
35	ssn_count_7
36	ssn_count_14
37	ssn_count_30
38	address_day_since

197	ssn_count_1_by_7
198	ssn_count_1_by_14
199	ssn_count_1_by_30
200	address_count_0_by_3
201	address_count_0_by_7
202	address_count_0_by_14
203	address_count_0_by_30
204	address_count_1_by_3
205	address_count_1_by_7
206	address_count_1_by_14
207	address_count_1_by_30
208	zip5_count_0_by_3
209	zip5_count_0_by_7
210	zip5_count_0_by_14
211	zip5_count_0_by_30
212	zip5_count_1_by_3
213	zip5_count_1_by_7
214	zip5_count_1_by_14
215	zip5_count_1_by_30
216	dob_count_0_by_3
217	dob_count_0_by_7
218	dob_count_0_by_14
219	dob_count_0_by_30
220	dob_count_1_by_3
221	dob_count_1_by_7
222	dob_count_1_by_14
223	dob_count_1_by_30
224	homephone_count_0_by_3
225	homephone_count_0_by_7

39	address_count_0
40	address_count_1
41	address_count_3
42	address_count_7
43	address_count_14
44	address_count_30
45	zip5_day_since
46	zip5_count_0
47	zip5_count_1
48	zip5_count_3
49	zip5_count_7
50	zip5_count_14
51	zip5_count_30
52	dob_day_since
53	dob_count_0
54	dob_count_1
55	dob_count_3
56	dob_count_7
57	dob_count_14
58	dob_count_30
59	homephone_day_since
60	homephone_count_0
61	homephone_count_1
62	homephone_count_3
63	homephone_count_7
64	homephone_count_14
65	homephone_count_30
66	name_day_since
67	name_count_0

226	homephone_count_0_by_14
227	homephone_count_0_by_30
228	homephone_count_1_by_3
229	homephone_count_1_by_7
230	homephone_count_1_by_14
231	homephone_count_1_by_30
232	name_count_0_by_3
233	name_count_0_by_7
234	name_count_0_by_14
235	name_count_0_by_30
236	name_count_1_by_3
237	name_count_1_by_7
238	name_count_1_by_14
239	name_count_1_by_30
240	fulladdress_count_0_by_3
241	fulladdress_count_0_by_7
242	fulladdress_count_0_by_14
243	fulladdress_count_0_by_30
244	fulladdress_count_1_by_3
245	fulladdress_count_1_by_7
246	fulladdress_count_1_by_14
247	fulladdress_count_1_by_30
248	name_dob_count_0_by_3
249	name_dob_count_0_by_7
250	name_dob_count_0_by_14
251	name_dob_count_0_by_30
252	name_dob_count_1_by_3
253	name_dob_count_1_by_7
254	name_dob_count_1_by_14

68	name_count_1
69	name_count_3
70	name_count_7
71	name_count_14
72	name_count_30
73	fulladdress_day_since
74	fulladdress_count_0
75	fulladdress_count_1
76	fulladdress_count_3
77	fulladdress_count_7
78	fulladdress_count_14
79	fulladdress_count_30
80	name_dob_day_since
81	name_dob_count_0
82	name_dob_count_1
83	name_dob_count_3
84	name_dob_count_7
85	name_dob_count_14
86	name_dob_count_30
87	name_fulladdress_day_since
88	name_fulladdress_count_0
89	name_fulladdress_count_1
90	name_fulladdress_count_3
91	name_fulladdress_count_7
92	name_fulladdress_count_14
93	name_fulladdress_count_30
94	name_homephone_day_since
95	name_homephone_count_0

255	name_dob_count_1_by_30
256	name_fulladdress_count_0_by_3
257	name_fulladdress_count_0_by_7
258	name_fulladdress_count_0_by_14
259	name_fulladdress_count_0_by_30
260	name_fulladdress_count_1_by_3
261	name_fulladdress_count_1_by_7
262	name_fulladdress_count_1_by_14
263	name_fulladdress_count_1_by_30
264	name_homephone_count_0_by_3
265	name_homephone_count_0_by_7
266	name_homephone_count_0_by_14
267	name_homephone_count_0_by_30
268	name_homephone_count_1_by_3
269	name_homephone_count_1_by_7
270	name_homephone_count_1_by_14
271	name_homephone_count_1_by_30
272	fulladdress_dob_count_0_by_3
273	fulladdress_dob_count_0_by_7
274	fulladdress_dob_count_0_by_14
275	fulladdress_dob_count_0_by_30
276	fulladdress_dob_count_1_by_3
277	fulladdress_dob_count_1_by_7
278	fulladdress_dob_count_1_by_14
279	fulladdress_dob_count_1_by_30
280	fulladdress_homephone_count_0_by_3
281	fulladdress_homephone_count_0_by_7
282	fulladdress_homephone_count_0_by_14

96	name_homephone_count_1
97	name_homephone_count_3
98	name_homephone_count_7
99	name_homephone_count_14
100	name_homephone_count_30
101	fulladdress_dob_day_since
102	fulladdress_dob_count_0
103	fulladdress_dob_count_1
104	fulladdress_dob_count_3
105	fulladdress_dob_count_7
106	fulladdress_dob_count_14
107	fulladdress_dob_count_30
108	fulladdress_homephone_day_since
109	fulladdress_homephone_count_0
110	fulladdress_homephone_count_1
111	fulladdress_homephone_count_3
112	fulladdress_homephone_count_7
113	fulladdress_homephone_count_14
114	fulladdress_homephone_count_30
115	dob_homephone_day_since
116	dob_homephone_count_0
117	dob_homephone_count_1

283	fulladdress_homephone_count_0_by_30
284	fulladdress_homephone_count_1_by_3
285	fulladdress_homephone_count_1_by_7
286	fulladdress_homephone_count_1_by_14
287	fulladdress_homephone_count_1_by_30
288	dob_homephone_count_0_by_3
289	dob_homephone_count_0_by_7
290	dob_homephone_count_0_by_14
291	dob_homephone_count_0_by_30
292	dob_homephone_count_1_by_3
293	dob_homephone_count_1_by_7
294	dob_homephone_count_1_by_14
295	dob_homephone_count_1_by_30
296	homephone_name_dob_count_0_by_3
297	homephone_name_dob_count_0_by_7
298	homephone_name_dob_count_0_by_14
299	homephone_name_dob_count_0_by_30
300	homephone_name_dob_count_1_by_3
301	homephone_name_dob_count_1_by_7
302	homephone_name_dob_count_1_by_14
303	homephone_name_dob_count_1_by_30
304	ssn_firstname_count_0_by_3

118	dob_homephone_count_3
119	dob_homephone_count_7
120	dob_homephone_count_14
121	dob_homephone_count_30
122	homephone_name_dob_day_since
123	homephone_name_dob_count_0
124	homephone_name_dob_count_1
125	homephone_name_dob_count_3
126	homephone_name_dob_count_7
127	homephone_name_dob_count_14
128	homephone_name_dob_count_30
129	ssn_firstname_day_since
130	ssn_firstname_count_0
131	ssn_firstname_count_1
132	ssn_firstname_count_3
133	ssn_firstname_count_7
134	ssn_firstname_count_14
135	ssn_firstname_count_30
136	ssn_lastname_day_since
137	ssn_lastname_count_0
138	ssn_lastname_count_1
139	ssn_lastname_count_3
140	ssn_lastname_count_7
141	ssn_lastname_count_14
142	ssn_lastname_count_30
143	ssn_address_day_since
144	ssn_address_count_0
145	ssn_address_count_1
146	ssn_address_count_3

305	ssn_firstname_count_0_by_7
306	ssn_firstname_count_0_by_14
307	ssn_firstname_count_0_by_30
308	ssn_firstname_count_1_by_3
309	ssn_firstname_count_1_by_7
310	ssn_firstname_count_1_by_14
311	ssn_firstname_count_1_by_30
312	ssn_lastname_count_0_by_3
313	ssn_lastname_count_0_by_7
314	ssn_lastname_count_0_by_14
315	ssn_lastname_count_0_by_30
316	ssn_lastname_count_1_by_3
317	ssn_lastname_count_1_by_7
318	ssn_lastname_count_1_by_14
319	ssn_lastname_count_1_by_30
320	ssn_address_count_0_by_3
321	ssn_address_count_0_by_7
322	ssn_address_count_0_by_14
323	ssn_address_count_0_by_30
324	ssn_address_count_1_by_3
325	ssn_address_count_1_by_7
326	ssn_address_count_1_by_14
327	ssn_address_count_1_by_30
328	ssn_zip5_count_0_by_3
329	ssn_zip5_count_0_by_7
330	ssn_zip5_count_0_by_14
331	ssn_zip5_count_0_by_30
332	ssn_zip5_count_1_by_3
333	ssn_zip5_count_1_by_7

147	ssn_address_count_7
148	ssn_address_count_14
149	ssn_address_count_30
150	ssn_zip5_day_since
151	ssn_zip5_count_0
152	ssn_zip5_count_1
153	ssn_zip5_count_3
154	ssn_zip5_count_7
155	ssn_zip5_count_14
156	ssn_zip5_count_30
157	ssn_dob_day_since
158	ssn_dob_count_0
159	ssn_dob_count_1
160	ssn_dob_count_3
161	ssn_dob_count_7
162	ssn_dob_count_14
163	ssn_dob_count_30
164	ssn_homephone_day_since
165	ssn_homephone_count_0
166	ssn_homephone_count_1
167	ssn_homephone_count_3
168	ssn_homephone_count_7
169	ssn_homephone_count_14
170	ssn_homephone_count_30
171	ssn_name_dob_day_since
172	ssn_name_dob_count_0
173	ssn_name_dob_count_1
174	ssn_name_dob_count_3

334	ssn_zip5_count_1_by_14
335	ssn_zip5_count_1_by_30
336	ssn_dob_count_0_by_3
337	ssn_dob_count_0_by_7
338	ssn_dob_count_0_by_14
339	ssn_dob_count_0_by_30
340	ssn_dob_count_1_by_3
341	ssn_dob_count_1_by_7
342	ssn_dob_count_1_by_14
343	ssn_dob_count_1_by_30
344	ssn_homephone_count_0_by_3
345	ssn_homephone_count_0_by_7
346	ssn_homephone_count_0_by_14
347	ssn_homephone_count_0_by_30
348	ssn_homephone_count_1_by_3
349	ssn_homephone_count_1_by_7
350	ssn_homephone_count_1_by_14
351	ssn_homephone_count_1_by_30
352	ssn_name_dob_count_0_by_3
353	ssn_name_dob_count_0_by_7
354	ssn_name_dob_count_0_by_14
355	ssn_name_dob_count_0_by_30
356	ssn_name_dob_count_1_by_3
357	ssn_name_dob_count_1_by_7
358	ssn_name_dob_count_1_by_14
359	ssn_name_dob_count_1_by_30
360	ssn_name_fulladdress_count_0_by_3
361	ssn_name_fulladdress_count_0_by_7

175	ssn_name_dob_count_7
176	ssn_name_dob_count_14
177	ssn_name_dob_count_30
178	ssn_name_fulladdress_day_since
179	ssn_name_fulladdress_count_0
180	ssn_name_fulladdress_count_1
181	ssn_name_fulladdress_count_3
182	ssn_name_fulladdress_count_7
183	ssn_name_fulladdress_count_14
184	ssn_name_fulladdress_count_30
185	ssn_name_homephone_day_since
186	ssn_name_homephone_count_0
187	ssn_name_homephone_count_1

362	ssn_name_fulladdress_count_0_by_14
363	ssn_name_fulladdress_count_0_by_30
364	ssn_name_fulladdress_count_1_by_3
365	ssn_name_fulladdress_count_1_by_7
366	ssn_name_fulladdress_count_1_by_14
367	ssn_name_fulladdress_count_1_by_30
368	ssn_name_homephone_count_0_by_3
369	ssn_name_homephone_count_0_by_7
370	ssn_name_homephone_count_0_by_14
371	ssn_name_homephone_count_0_by_30
372	ssn_name_homephone_count_1_by_3
373	ssn_name_homephone_count_1_by_7
374	ssn_name_homephone_count_1_by_14
375	ssn_name_homephone_count_1_by_30