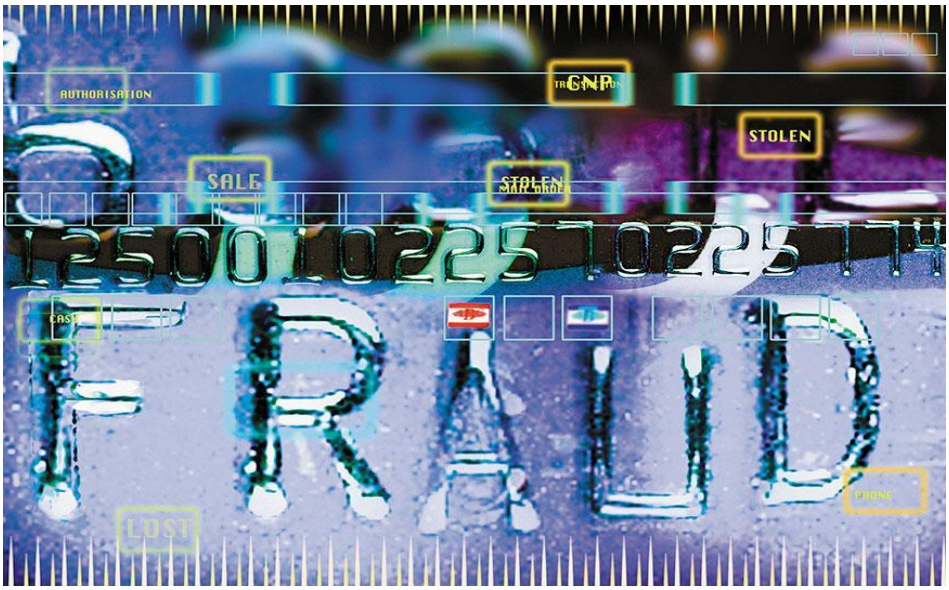
Credit Card Application Fraud Detection Report

Real Team 3

Sheng Ming, Baili Lu, Hao Qu,

Peipei Han, Shuhan Zhou, Manwen Hu

March 23, 2017



Contents

I. Executive Summary

The purpose of this project is to identify unusual items by giving a fraud score to each record in product applications using unsupervised learning models. The dataset was downloaded from the blackboard showing the application's detail information including date, applicant's name, SSN, address, date of birth and phone number of 100,000 records. Anomalous records per fraud model are more likely to be a signal of identity theft due to the limitation of the size of our data.

To identify potential fraud, time flow is critical and only the observations in the past can be utilized to build the model. Variables are built based on a time window. The appearance of frivolous values, which will be meaningless noise in the model, is checked and replaced with neutral values. The fix time window of 3 and 7 days together with all past period are applied in this project. Identifier variables which include both full name and date of birth, different combinations of counting for different time window and unique variables with the same combinations of other variables are created. By adding these additional variables in the original dataset, we tried to ferret out some hidden information and relationship among each variable.

Two methods have been applied to solve this problem: autoencoder and heuristic algorithm. To begin with, Z-scale is utilized to standardize the data and PCA is used to reduce the dimension. Result shows that 11 PC's can represent 93.75% of the data. The fraud score is calculated by training the Autoencoder model and run it on the entire database. Another way to get fraud score is to apply a heuristic function to PCA. The top 1% likelihood of fraud observations of both methods is compared, and it comes out that there's 58.1% of overlap for the two methods. The combination of autoencoder fraud score and heuristic fraud score is also used to check the potential records. After several calculations, the method of autoencoder is selected because the potential fraud records are more reasonably interpreted.

10 observations with highest fraud score are identified. The reasons of high scores can be the high frequency for appearance of certain personal information within different time windows selected and we will discuss in detail at the result section.

II. Data Distribution

2.1.File Description

The name of the data file is Application 100k. It is downloaded from an identity fraud detection institution and used to detect the fraudulent items in credit card application. There are 100,000 records and 9 fields in total. There are two date fields, three text fields and one continuous field. The time frame is from 01/01/2015 to 12/31/2015. It is worth mentioning that there is no null value in the dataset.

2.2.Summary Table

The column record is excluded from our analysis because it doesn't have any practical meaning. Below is the summarizing table that shows the unique values and populated percentage.

Field Name	date	ssn	first name	last name	address	zip5	dob	homephone
#Unique Value	365	96535	16576	36312	97563	16547	36816	22181
%Population	100%	100%	100%	100%	100%	100%	100%	100%

Figure 1 Summary Table

2.3.Variable Information

2.3.1. Record

Record is the index of each record, which is a unique identifier of each observation, ranging from 1 to 100000. And this variable does not provide any related information about possible fraud.

2.3.2. Date

Date is a categorical variable, formatted as yyyymmdd. The range is from 01/01/2015 to 12/31/2015. Figure 2 illustrates the dates throughout the year 2016, from which we can tell there is no discernable anomaly in the distribution.

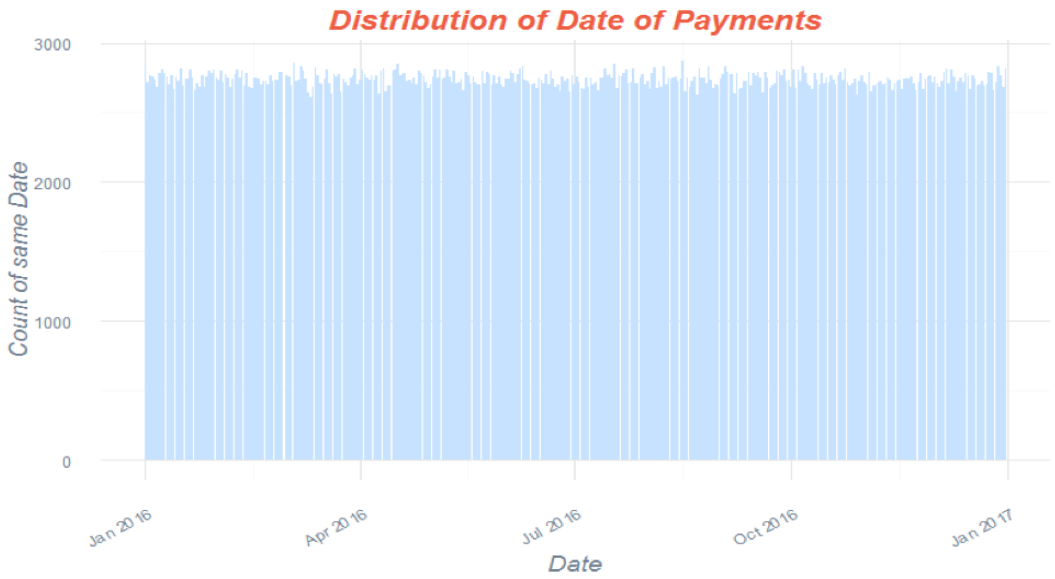


Figure 2 Distribution of Date of Payments

2.3.3. SSN

SSN is a categorical variable. There are 96535 unique values within this variable. The length is from 6 digits to 9 digits. From the chart, it is obvious that 737610282 is anomalous and frivolous in that it appears extremely frequently. To better visualize the SSN, frivolous value is removed and the Top 20 is selected.

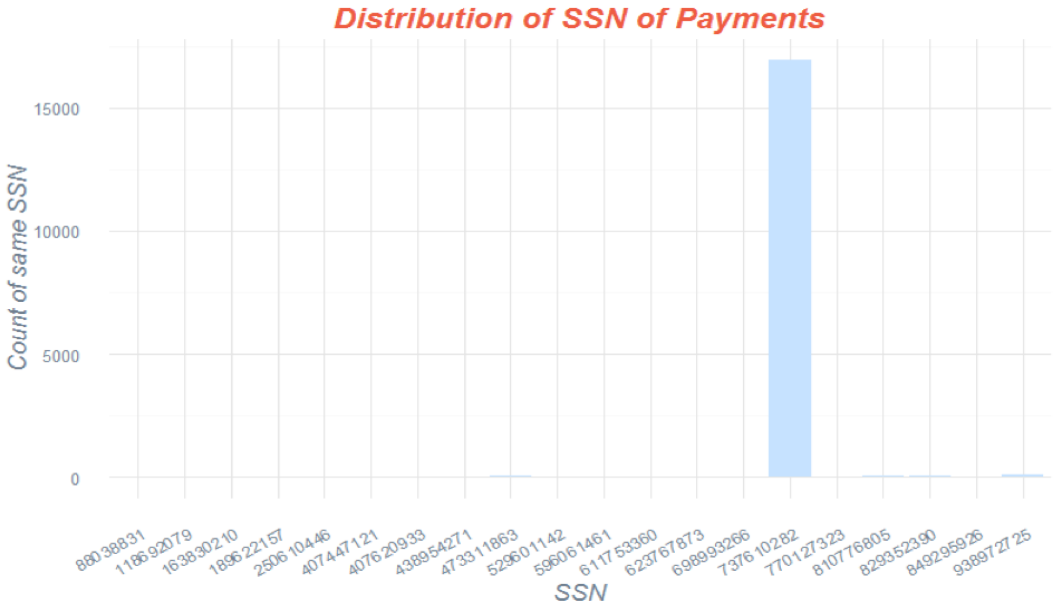


Figure 3 Distribution of SSN of Payments

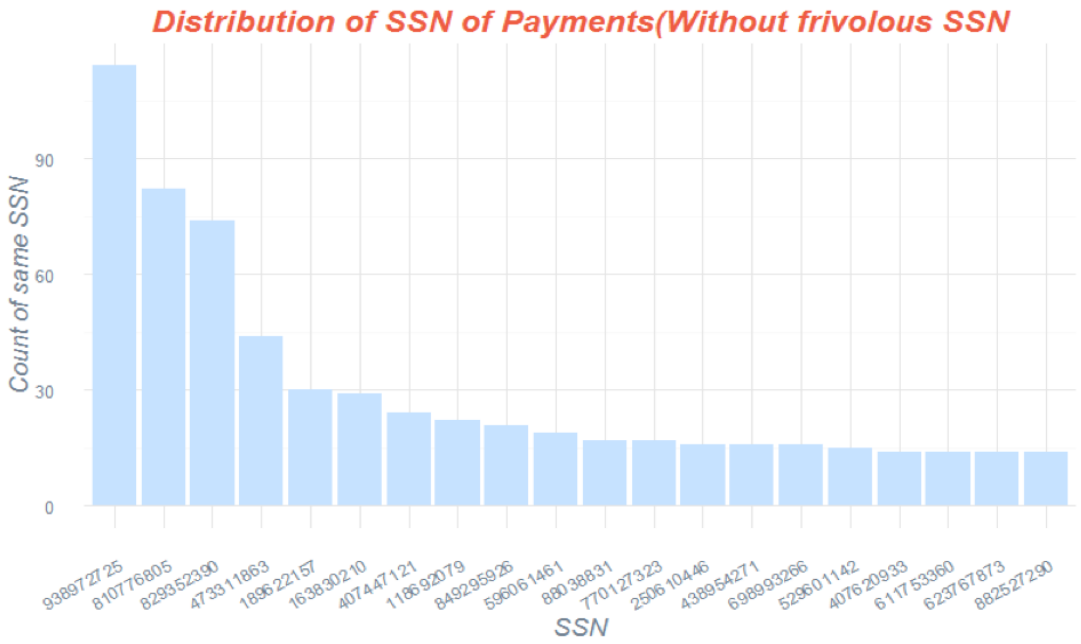


Figure 4 Distribution of SSN of Payments (Without Frivolous SSN)

2.3.4. First name

Firstname is a categorical variable. The most frequent name is EAMSTRMT, occurring 12,648 times. Figure 5 includes top 30 frequent first names.

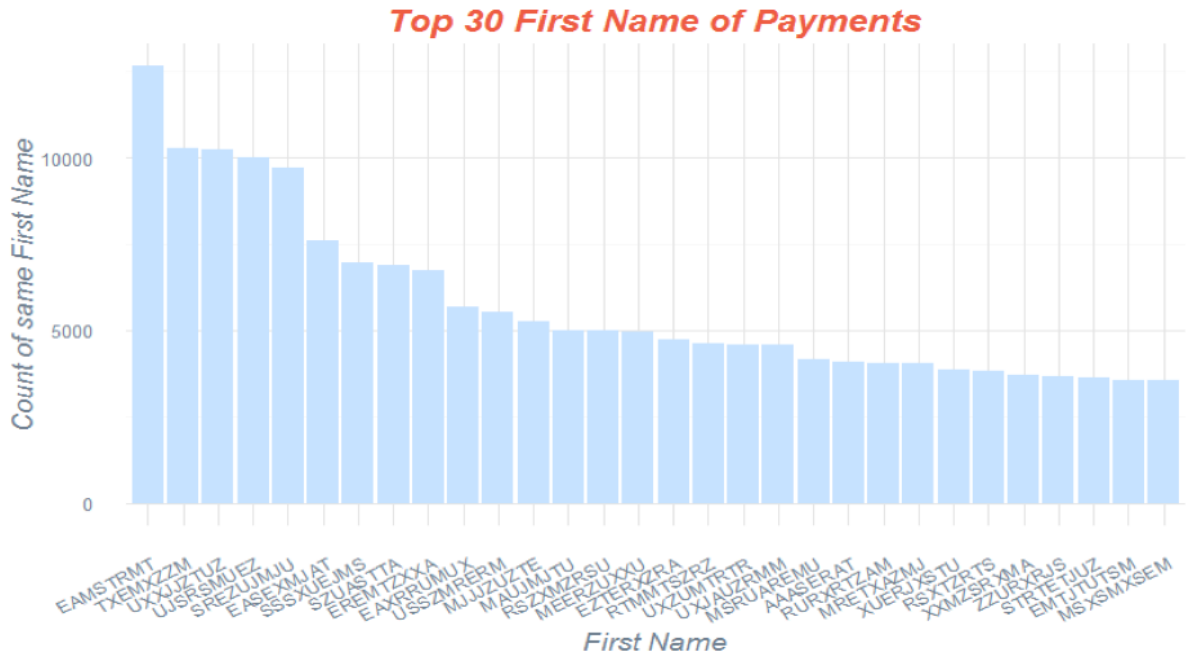


Figure 5 Top 30 First Name of Payments

2.3.5. Last name

Lastname is a categorical variable. The most frequent last name is ERJSAXA.

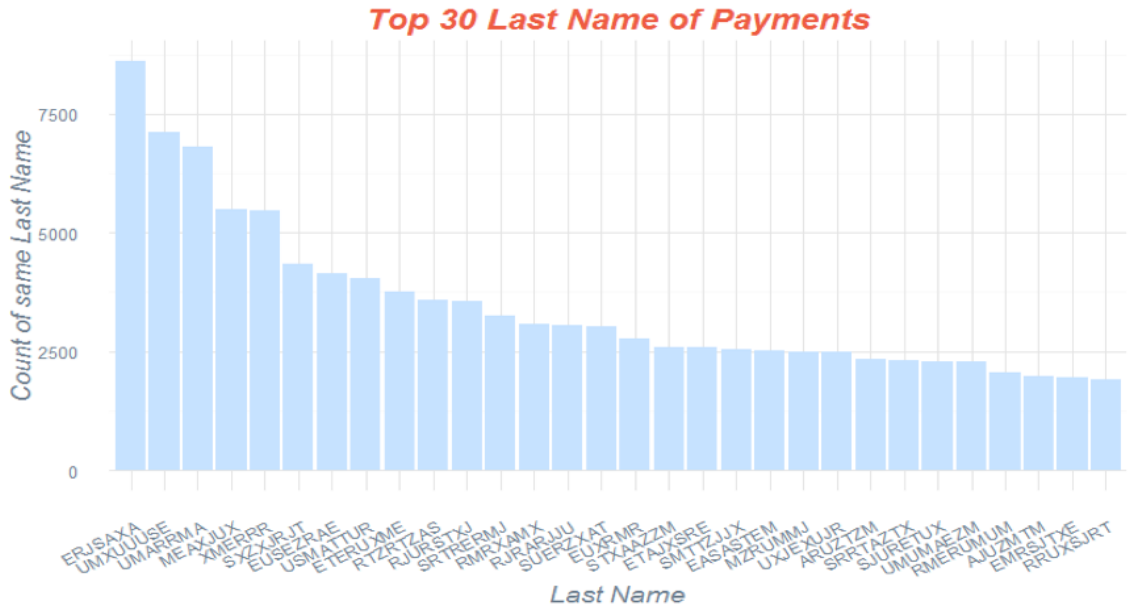


Figure 6 Top 30 Last Name of Payments

2.3.6. Address

Address is a categorical variable. The most frequent address is 2602 AJTJ AVE, which is possibly a frivolous value.

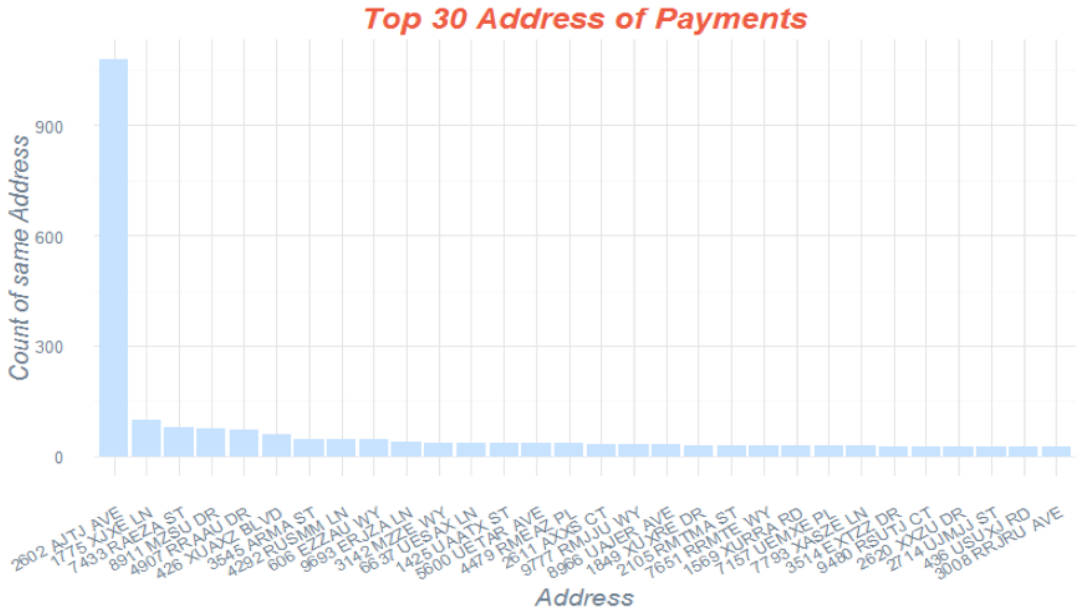


Figure 7 Top 30 Address of Payments

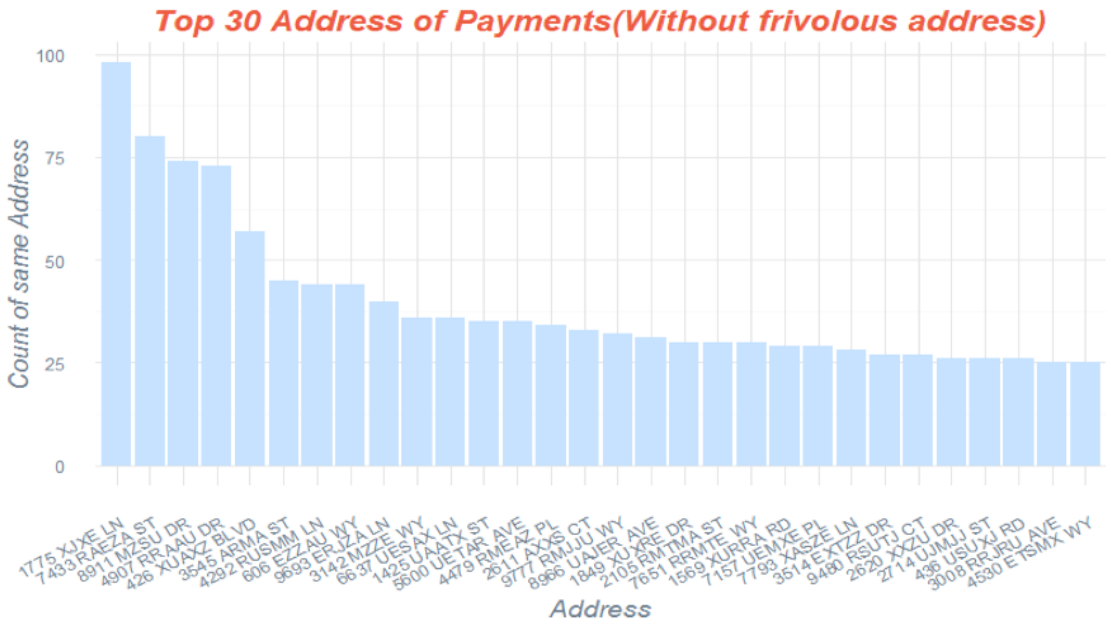


Figure 8 Top 30 Address of Payments (Without frivolous address)

2.3.7. Zip5

Zip5 is a categorical variable, which represents the zip code of that specific observation. Figure 9 summarizes the top 30 frequent values of zip5, and it can be noticed that the zip code of 68138 is the most frequent.

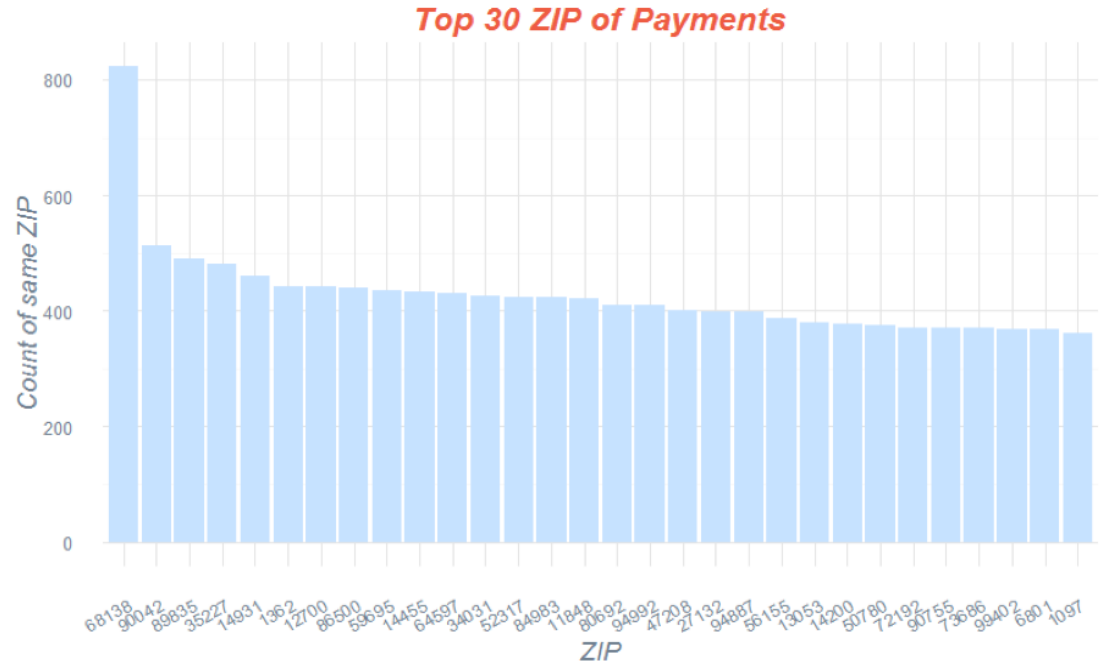


Figure 9 Top 30 ZIP code of Payments

2.3.8. Dob

Dob is a categorical variable. The most frequent date of birth is 06/26/1907. Based on the distribution plot on year of birth, 1907 can be a frivolous year because it has much more records than other years.

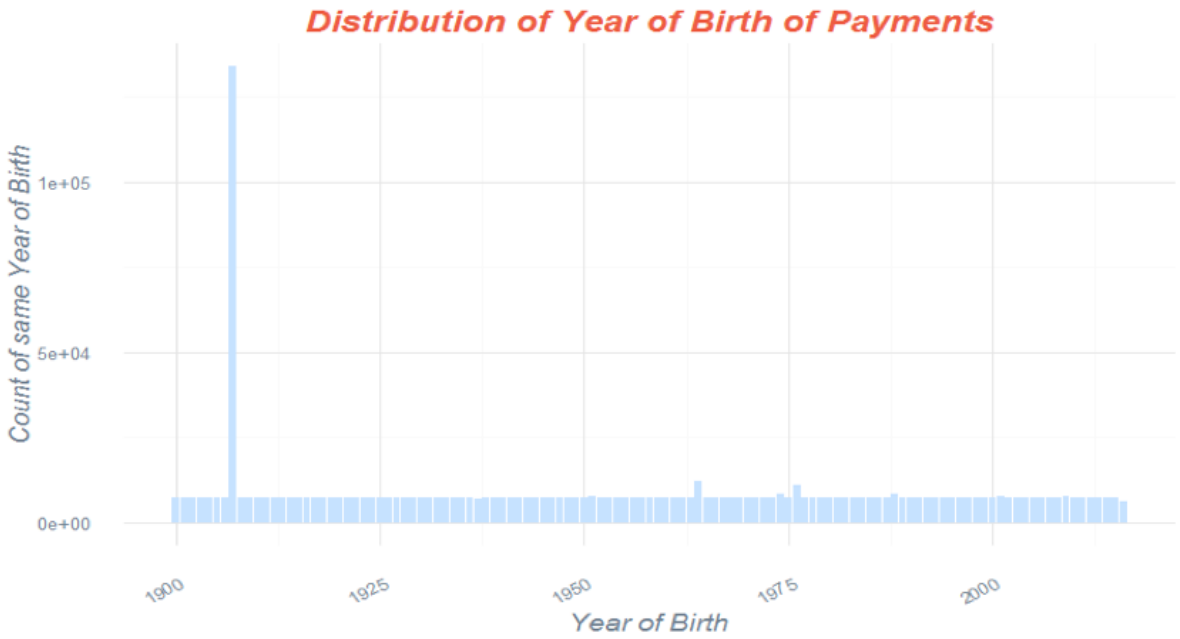


Figure 10 Distribution of Year of Birth of Payments

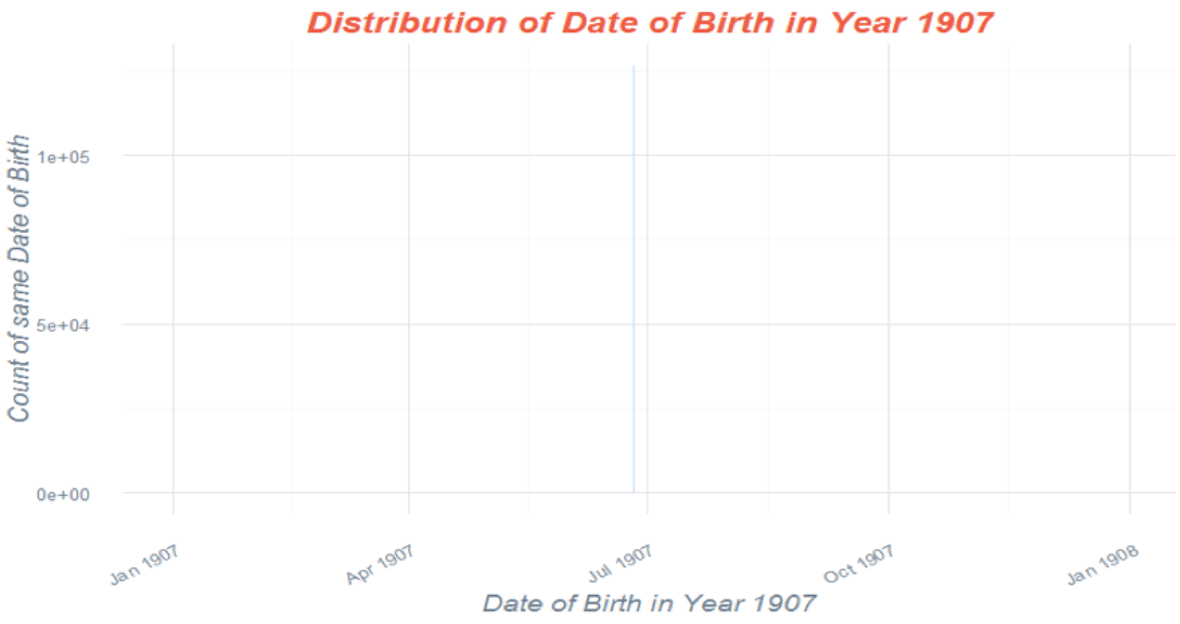


Figure 11 Distribution of Date of Birth in Year 1907

2.3.9. Homephone

Homephone is a categorical variable. Figure 12 shows the Top 20 frequent home phone, and the home phone number of 9105580920 is distinctively frequent, which is highly suspicious. Figure 13 shows the Top 20 home phone without the frivolous number.

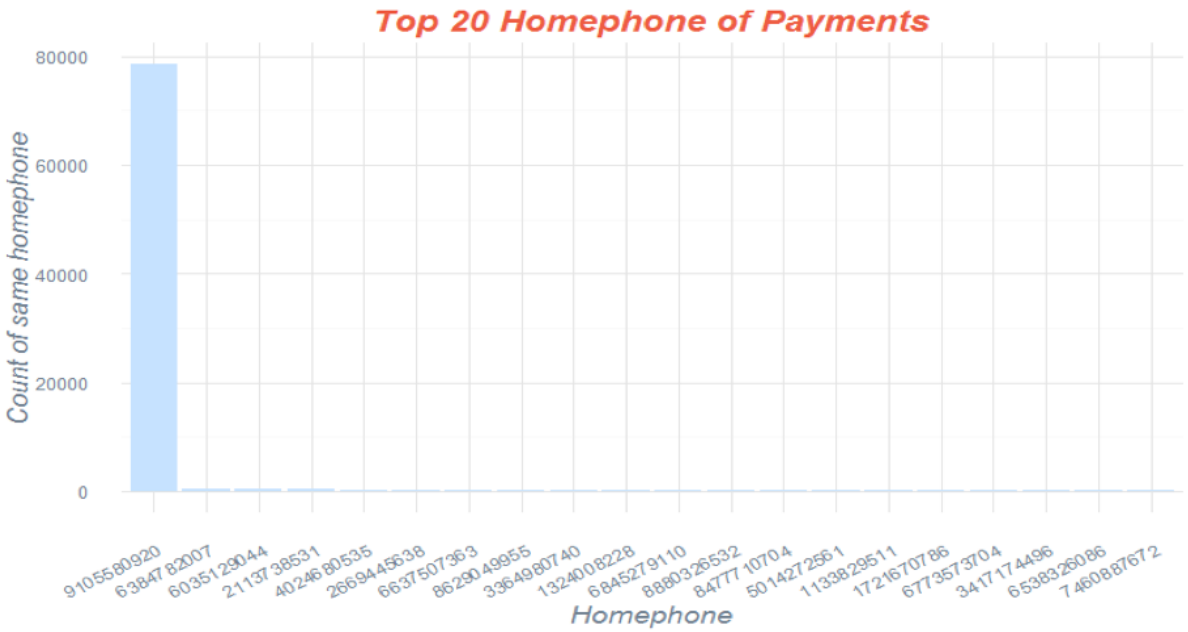


Figure 12 Top 20 Home phone of Payments

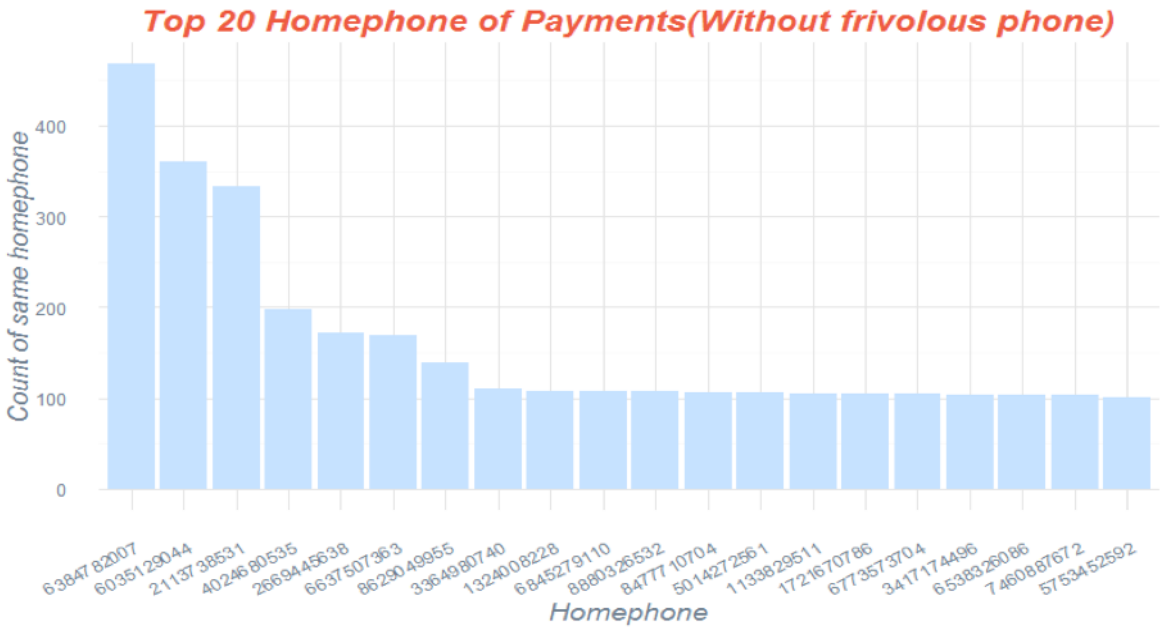


Figure 13 Top 20 Home phone of Payments (Without frivolous home phone)

III. Variables Manipulation

3.1. Preprocess original variables

Since the raw data quality is high (no missing values, accurate and structured), the step of preliminary data manipulation involves:

- Fill in the leading zeros for ssn, zip5 and homephone;
- Concatenate firstname and lastname as name;
- Concatenate name and dob as unique identifier (id);
- Concatenate address and zip5 as full address

3.2. Create time windows and frequency-related variables

One of the challenging parts in this project is generating real time system by deriving information and knowledge from the past records, which serves as a prerequisite of constructing variables and time windows.

After creating the daily counts breakdown for each variable (6 variables: name, dob, id (name&dob), ssn, address (address&zip5), homephone), we found it inefficient by using for-loops to do cumulative counts calculation. After a thorough investigation in searching for improvements, we worked with a package in R called data.table as an superior approach.

The following step is building some 'supplementary variables', which are the cumulative counts for the whole past time period within each date-value groups using built-in cumsum() function. Then, a copy for each table was created (one table for one variable individually) and applied with the 'rolling-join' technique. Detailed information of the process is described as below:

First, create a 'join_date' column in the copy of each table, which is equal to the date variable minus the length of the time window. Then, join the copy to the original table on the original date and this new 'join_date'. Next, we match the observation occurred exactly in the 'join_date' in each group to the original record, or, if no record found in that date, a **rolling forward** option is provided which means the method will find the **closest previous date** before

the 'join_date' and match that record. Then, we subtracted the current cumulative count with the n-day-ago cumulative count to get the cumulative count for the past n days. An example of the halfway result to calculate the past 7-day cumulative counts for each homephone is like this:

	homephone	date	daily_phone	cum_phone	join_date	i.date	i.daily_phone	i.cum_phone	phoneLag7Cum
1:	0000635392	2015-10-08	1	8	2015-10-16	2015-10-23	1	9	1
2:	0000635392	2015-10-23	1	9	2015-12-14	2015-12-21	1	10	1
3:	0000635392	2015-10-23	1	9	2015-12-18	2015-12-25	1	11	2
4:	0000635392	2015-10-23	1	9	2015-12-19	2015-12-26	1	12	3
5:	0001177773	2015-01-20	1	1	2015-01-13	2015-01-20	1	1	1

In order to obtain the past 7-day cumulative counts for homephone '0000635392' on 12/26/2015, we can simply subtract the cumulative count for '0000635392' by 12/26/2015, which is 12, with the cumulative count for '0000635392'' by 10/23/2015, which is 9. This is correct because 10/23/2015 is the **closest previous date** that '0000635392' appeared before 12/19/2015, which is 7 days before 12/26/2015. Therefore, the difference 3 is the right answer for the past 7-day cumulative count for '0000635392' on 12/26/2015.

Based on the rationale above, the same methodology is applied to all 6 variables. In terms of the selection of lengths of time windows, we took 3-day and 7-day based on professor's advice. In sum, we built 21 frequency-related variables (daily count, past 3-day count and past 7-day count for each variable and the total).

3.3. Create variety-related variables and recency

Some unusual patterns could be revealed by observing the frequency-related variables, for example: The frequent occurrence for certain values in some days may be a potential implication of fraud. Taking it one step further, there are many more detailed insights derived such as how many different names appeared in the past 3 or 7 days for the same ssn (variety) and how long since the last occurrence of this id (recency).

As for the variety, the similar rolling-join method is applied as above on 4 major varieties:

- Number of different names for the same SSN
- Number of different SSNs for the same id
- Number of different address for the same id

- Number of different home phone numbers for the same id

In this section, 8 variety-related variables (past 3-day and 7-day variety respectively for name/ssn, ssn/id, address/id, phone/id) are created. A general belief from the analysis is, larger variety indicates higher risk of fraudulence, implying an intention of counterfeiting accounts.

When analyzing the recency (number of days since the last occurrence of the value), the focus is narrowed on the following 4 variables: name, id, ssn and address. Assumption here is the case that the smaller the recency is the higher probability there would be a fraud. If the recency is small like 0 or 1, which means that the same value occurred in the same day or consecutive days, this may be a dangerous signal of fraud. For normal applications, the corresponding value probably only occurred once so that the recency would be set at 999. By doing this, since most of the values of these four variables only appeared once in the dataset, the average recency for each variable is very close to 999, which means after applying z-scale, those 999 recency will be almost converted to 0 while the 0 or 1 recency will become very outstanding, which satisfies the original assumption.

3.4.Deal with the frivolous values

As is known from the beginning, there exists 4 frivolous values in the dataset:

DOB 19070626
Address 2602 AJIT AVE 68138
SSN 737610282
Homephone 9105580920

We used a straightforward approach to deal with frivolous values before joining sub-tables into original data: Replace each numeric value of records containing frivolous value with the mean value of that column calculated without the frivolous records. After doing the z-scaling later, all the numeric values in frivolous records would become zeros.

At last, join all the sub-tables into the original data, and the data manipulation section is completed.

3.5.Summary of Expert Variables

In total, 33 expert variables are created:

- 3 total counts (daily/3-day/7-day)
- 18 variable counts (daily/3-day/7-day)
- 8 variety counts (3-day/7-day: name/ssn, ssn/id, address/id, phone/id)
- 4 recency (name, ssn, address, id)

Although the number of expert variables is not large, the quality is ensured by eliminating some meaningless variables. And in this way it prevents reducing the effectiveness of pca. Another exciting thing to mention is, all the variables are constructed within 15 seconds, attributed to the powerful manipulation by utilizing data.table in R.

IV. Methods and Techniques

4.1.Standard normalization and PCA

As mentioned above, we totally have 44 variables to build model and calculate the fraud score. But high dimension could be problematic. It means high computation cost and leads to overfitting. There could also be high correlation among the variables.

Dimensionality reduction addresses these problems, while preserving most of the relevant information in the data needed to learn accurate, predictive models. The axes of the reduced subspace typically correspond to latent features that remove noise, abstract, compress and in general better describe the correlations and interactions among the original set of features - thus enabling learning algorithms to perform better.

Principal component analysis is the main linear technique for dimensionality reduction. It performs a linear mapping of the data to a lower-dimensional space in such a way that the variance of the data in the low-dimensional representation is maximized.

However, before implementing PCA, we should do $ZScale = \frac{x-\mu}{\sigma}$ standardization first to adjust values measured on different scales to a notionally common scale, as PCA is a variance maximizing exercise.

In practice, the covariance matrix of the data is constructed and the eigen vectors on this matrix are computed. The eigenvectors that correspond to the largest eigenvalues (the principal components) can now be used to reconstruct a large fraction of the variance of the original data. Moreover, the first few eigenvectors can often be interpreted in terms of the large-scale physical behavior of the system. The original space has been reduced to the space spanned by a few eigenvectors. One common criterion is to include all those PCs up to a predetermined total percent variance explained, such as 80%.

In practice, PCA can be implemented using *prcomp()* function in R. The variable standard deviations are stored in the attribute scale and scores are in the attribute x. After PCA, we selected 11 variables as the corresponding cumulative eigenvalue (variance) reaching 93.75%. Figure 14 shows that there is a decline at PC_{11} and behind PC_{11} there is less information contributed to dataset. Therefore, the first 11 variables are chosen to be the input of the fraud score algorithm to calculate fraud score. Figure 15 illustrates the variables which the most significant ones are for PC_1.

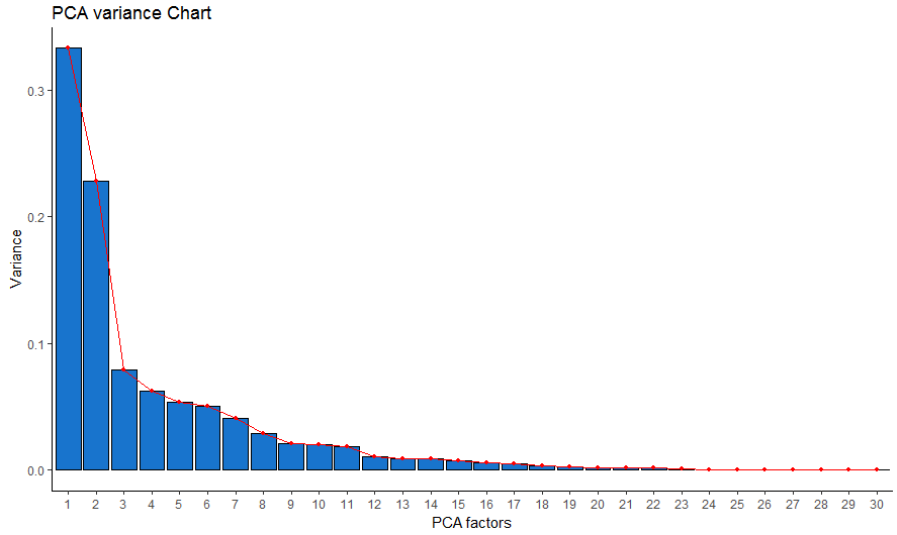


Figure 14 PCA Variance Chart

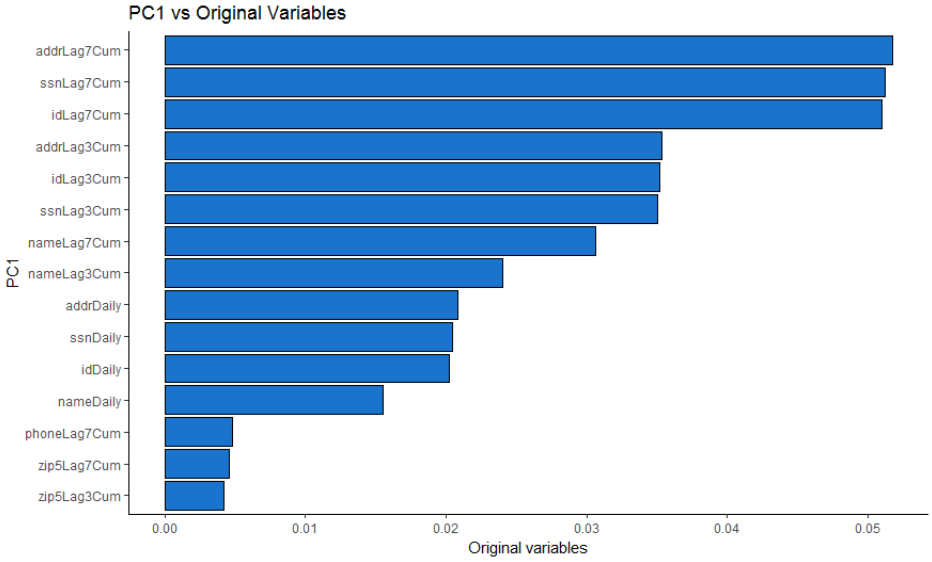


Figure 15 Original variables contributed to PC_1

4.2.Heuristic Modeling

To generate fraud score, we used Heuristic algorithm and Autoencoder individually. In terms of the Heuristic algorithm, we manipulated and modeled all the principal components in the following procedures:

1. Z-scale all the PCs.

   Although we have already z-scaled all the original and expert variables before we did PCA, we are now trying to investigate the deviation of each observation within each principal component. Since each principal component has different mean value and variance, a z-scale on all the PCs is a must before we do any calculation on the PCs.

2. Sum up the absolute values of all z-scaled PCs and take the square root.

   Since now we have the z-scaled PCs, one of the most straight-forward ways to measure the total deviation is to sum up all the absolute values of the z-scaled PCs. We choose to take the square root on the sum for a less skewed distribution and more comparable result to the Autoencoder score (will mention below). So we get:

   $$Score.Heuristic = (\sum^{i} |PC_i|)^{\frac{1}{3}}$$

3. Scale the Score. Heuristic to [0,1].

   To make our score more comparable, we scale the scores to [0,1] using (score-min)/(max-min).

The distribution of $Score.Heuristic$ is shown in Figure 16, which illustrates that the majority of the Heuristic fraud score are concentrated around 0.1 and there is an obvious skew and long tail after 0.5.

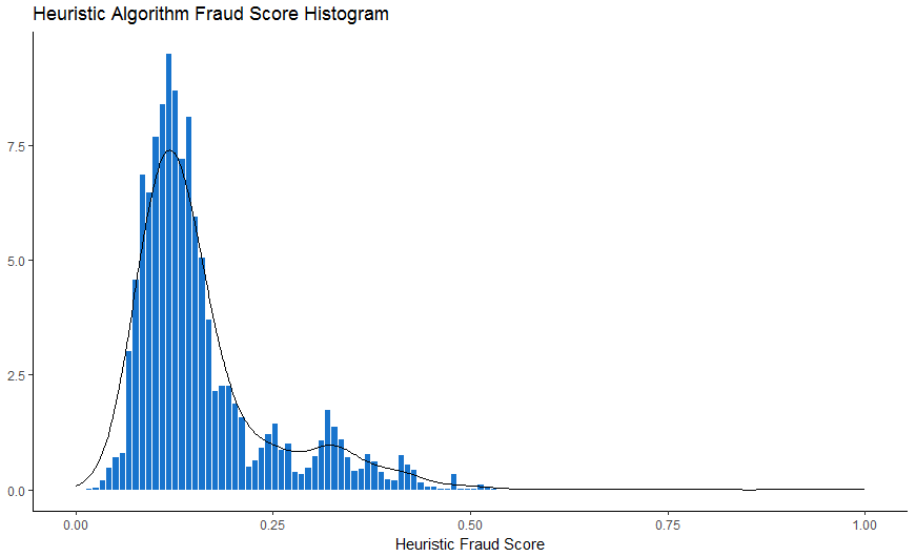


Figure 16 $Score.Heuristic$ **Histogram**

4.3.Autoencoder

An autoencoder neural network is an unsupervised learning algorithm that applies back-propagation, setting the target values to be equal to the inputs. The Autoencoder tries to learn a function $h_{w,b}(x) \approx x$. In other words, it is trying to learn an approximation to the identity function, to output $\hat{x}$ that is similar to x and the fraud score is given according to the reconstruction error. The general process is shown in Figure 17.

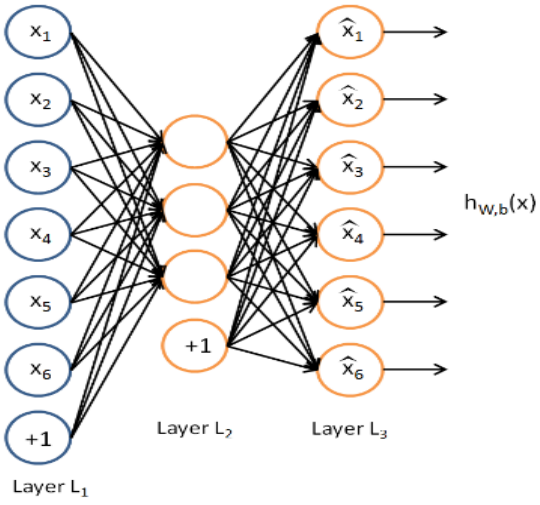


Figure 17 Autoencoder Algorithm

In practice, we used the h2o library and tuned the autoencoder model with different parameters such as the number of hidden layers, the number of neurons within each layer and the number of iterations. After a number of trials, we find one hidden layer with 4 neurons (which is the same as the number of our PCs) is an efficient neural network for our training and 30 iterations is good enough to make the result converge to an optimal value. Besides, we take the third root of the reconstruction error and scale the result to [0,1] to make the Autoencode score more comparable to the Heuristic score since we are about to combine them both linearly. And we get:

$$Score.Autoencoder = (Reconstruction.MSE)^{\frac{1}{3}}$$

The distribution of $Score.Autoencoder$ is shown in Figure 18, which shows a similar pattern compared with the distribution of Heuristic score.

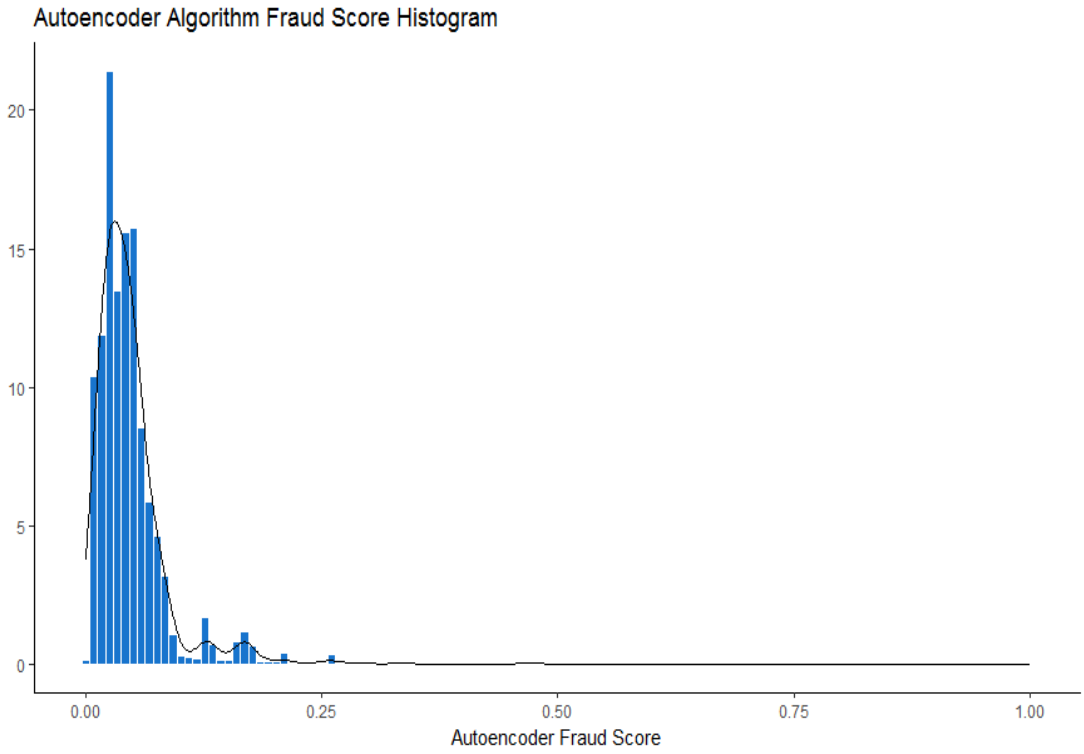


Figure 18 Score.AutoencoderHistogram

We sorted the records by Heuristic.Score and Autoencoder.Score respectively and selected the top 10,000 observations from each result. As it turns out, 5,810 of the records

appear in both two top 10,000 records!That's a remarkable result and a very good sign of the effectiveness of both the two models.

Since we have both the two scores now and they are about of the same scale and very similar distribution. We are planning to derive the final fraud score with a combination of the two scores. After all, we are doing an unsupervised learning modeling for the project. We should not bet all on one model.

4.4.Fraud Score Combination

Since the two scores have extremely similar distribution, we think the one of the most straight-forward ways to balance the two score is to make a simple linear combination. Since the Autoencoder algorithms is more sophisticated, we decided to give a 0.7 weight to Score.Autoencoder and 0.3 to Score.Heuristic, so we get:

$$Score.Combined = 0.7 * Score.Autoencoder + 0.3 * Score.Heuristic.$$

The result of combined fraud score is shown in Figure 19. The long tail indicates that amount of records have high fraud scores and we should get the original abnormal records and go deep for future research of the reasons that led to high fraud scores. Therefore, after the combination, we selected the top 1% highest fraud score records.

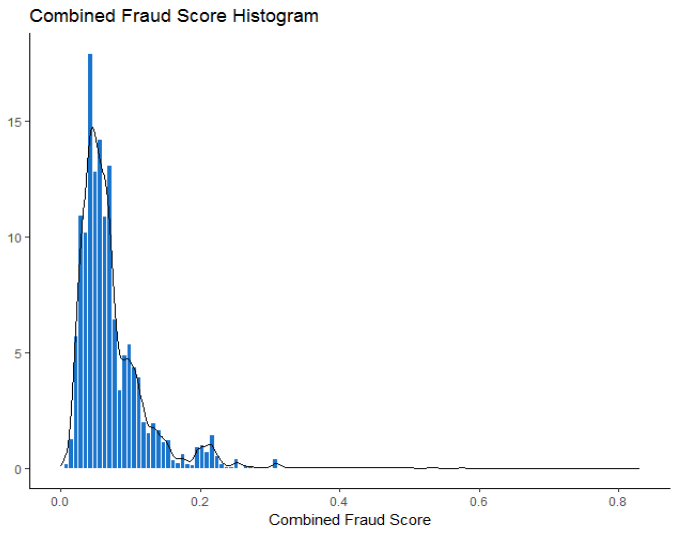


Figure 19 Combined Histogram

V. Result Analysis

As mentioned above, top 1% (10,000 rows) records were extracted from all the observations according to the combined fraud score. In this part, 10 very high fraud score records will be interpreted to show the reasons behind the high fraud scores.

Record	Score.Heur	Score.AE	Score.Combined	date	ssn	firstname	lastname	address	zip5	dob	homephone	identifier	idRecency
79349	0.9950779	0.3228233	0.5244997	10/16/2015	441432496	XUERJXSTU	TETSAJS	4702 UMAAS PI	33257	10/22/1955	4846336019	XUERJXSTU TETSAJS 19551022	1
2957	0.993427	0.322655	0.5238866	1/11/2015	451558906	MSRUAREMU	EZATSJSU	8626 XXSTE PL I	82999	11/20/1998	1033418292	MSRUAREMU EZATSJSU 19981120	2
17884	0.9903368	0.2908883	0.5007228	3/7/2015	0262311110	UZZJXEMZA	RZAUSZXJ	3677 RESRE CT	56407	11/8/2012	3848343172	UZZJXEMZA RZAUSZXJ 20121108	1
2640	0.9999101	0.2834259	0.4983712	1/10/2015	807202258	MSSZJXZXE	SZSXTTMA	1460 UEXMR D	49525	6/26/1907	9105580920	MSSZJXZXE SZSXTTMA 19070626	2
62428	1	0.2833661	0.4983562	8/16/2015	022066790	RSZXMZRSU	RRSJSAU	5407 SJZSE WY	47386	6/26/1907	9105580920	RSZXMZRSU RRSJSAU 19070626	1
53251	0.9891546	0.2839266	0.495495	7/13/2015	004178024	XSMZZJMZT	SZEZUUMX	7484 SSXUM LN	04910	6/26/1907	7633541686	XSMZZJMZT SZEZUUMX 19070626	1
87109	0.9855745	0.2848894	0.4950949	11/14/2015	997293490	SRTJJUTMA	EMMMTAAE	7790 AMMZ DR	01863	9/16/1902	0883904896	SRTJJUTMA EMMMTAAE 19020916	2
92096	0.9847312	0.283406	0.4938035	12/2/2015	975338076	ZZURXRJS	ZZZJUJ	4331 SAXAS ST	22643	11/19/1915	5945356325	ZZURXRJS ZZZJUJ 19151119	2
72092	0.9875835	0.2819535	0.4936425	9/20/2015	567707500	SZUASTTA	SUSJJZJJ	23 XAMSS LN 5	59634	5/24/1903	3574827637	SZUASTTA SUSJJZJJ 19030524	1
43904	0.9864642	0.282116	0.4934205	6/9/2015	958334381	SREZUJMJU	UAAMRSTJ	2925 EEXMM CT	40835	12/5/1925	09734164910	SREZUJMJU UAAMRSTJ 19251205	2
70099	0.9863433	0.2817977	0.4931614	9/13/2015	277740653	MJJJZUZTE	RMUTMTAX	2293 RZMTX CT	38937	11/5/1990	2475628833	MJJJZUZTE RMUTMTAX 19901105	1
78503	0.9839425	0.2825159	0.4929439	10/13/2015	586810301	RUJZUUJAZ	SXARMXSR	9298 USJJX DR	76296	10/28/1911	2294793337	RUJZUUJAZ SXARMXSR 19111028	2
34392	0.7685135	0.1689937	0.3488496	5/6/2015	521596592	EAXRRUMUX	STXAAZZM	1621 SUXM S	####	#######	9.106E+09	EAXRRUMUX STXAAZZM 1907	4

Record	countDaily	countLag7Cum	countLag3Cum	nameLag7Cum	nameLag3Cum	nameRecency	dobLag7Cum	dobLag3Cum	idLag7Cum	idLag3Cum	addrLag7id	addrLag3id	nameLag7ssn
79349	261	1829	792	2	2	1	2	2	2	2	0	0	0
2957	264	1981	789	2	2	2	2	2	2	2	0	0	0
17884	311	1886	822	2	2	1	2	2	2	2	1	1	1
2640	270	2001	835	2	2	2	1.085862	1.033818	2	2	1	1	1
62428	269	1952	842	2	2	1	1.085862	1.033818	2	2	1	1	1
53251	264	1872	793	2	2	1	1.085862	1.033818	2	2	1	1	1
87109	235	1900	850	2	2	2	2	2	2	2	1	1	1
92096	245	1916	805	2	2	2	2	2	2	2	1	1	1
72092	297	1991	827	2	2	1	2	2	2	2	1	1	1
43904	295	1923	808	2	2	2	2	2	2	2	1	1	1
70099	293	1926	816	2	2	1	2	2	2	2	1	1	1
78503	249	1844	796	2	2	2	2	2	2	2	1	1	1
34392	269	1839	784	3	2	2	1.085862	1.033818	2	1	2	1	1

Record	ssnLag7Cum	ssnLag3Cum	ssnRecency	addrLag7Cum	addrLag3Cum	addrRecency	phoneLag7Cum	phoneLag3Cum	ssnLag7id	ssnLag3id	nameLag3ssn	phoneLag7id	phoneLag3id
79349	2	2	1	2	2	1	2	2	0	0	0	0	0
2957	2	2	2	2	2	2	2	2	0	0	0	0	0
17884	2	2	1	2	2	1	2	2	1	1	1	1	1
2640	2	2	2	2	2	2	1.086719	1.033394	1	1	1	1	1
62428	2	2	1	2	2	1	1.086719	1.033394	1	1	1	1	1
53251	2	2	1	2	2	1	3	2	1	1	1	1	1
87109	2	2	2	2	2	2	2	2	1	1	1	1	1
92096	2	2	2	2	2	2	2	2	1	1	1	1	1
72092	2	2	1	2	2	1	2	2	1	1	1	1	1
43904	2	2	2	2	2	2	2	2	1	1	1	1	1
70099	2	2	1	2	2	1	2	2	1	1	1	1	1
78503	2	2	2	2	2	2	2	2	1	1	1	1	1
34392	1	1	999	1	1	999	1.086719	1.033394	2	1	1	2	1

The above ten records shows that observations with highest Heuristic scores also have highest Autoencoder scores. The high overlapped rate (58.1%) for the two scores in the top 10,000

records and high consistency among the top 10 provides strong evidence for the effectiveness and robustness of both the two models.

For the record 79349, in the last 7 days, identifier “XUERJXSTU TETSAJS 19551022” has appeared twice as well as two times in the last three days. The situation for the ssn is the same. The cumulative count of social security number 441432496 in the last seven and three days are both two.

For record 53251, home phone number 7633541686 has shown three times in seven days. Meanwhile, social security number 004178024 and address “7484 SSXUM LN 49101” also cumulatively appeared twice in the last three days. High frequency indicates a highly possible chance of fraud.

For record 34392, the same identifier summit application with two different social security numbers in the last seven days. The same identifier also used two different addresses and two phones during that week for credit card application. All these signals show a high potential for fraud.

All in all, for high fraud score applications, the personal information especially ssn, home phone number and address basically have existed at least twice in the last three or seven days. In practical, fraud could be a case by case problem. High fraud score could be the result of various reasons.

APPENDIX: DQR ON PAYMENTS DATA

VARIABLES SUMMARY

The file "Applications100k" contains 100,000 records with 9 fields (2 date fields, 3 text fields, and 3 categorical fields, and 1 continuous field). The dataset is sourced from an identity fraud detection institution, and below is a summary of the dataset.
File Name: Application
Time: Jan 1 – Dec 31, 2015
Fields Description:

- 2 date fields: date, dob(date of birth)
- 3 text fields: firstname, lastname, address
- 3 categorical fields: ssn, zip5, homephone
- 1 continuous field: record

Variable	Description	Percent Populated	Min	Max
Record	A unique identifier	100%	1	1000000
Date	Time period throughout 2016	100%	20160101	20161231
SSN	Social Security Number	100%	36	999999884
Firstname	First name	100%	-	-
Lastname	Last Name	100%	-	-
Address	Location	100%	-	-
ZIP5	ZIP code	100%	2	99999
DOB	Date of birth	100%	19000101	20161031
Homephone	Home phone number	100%	593799	9999317760

1. RECORD

- Description: The sequence number of each observation in the dataset, which is a unique identifier of each observation, ranging from 1-100,000.

2.DATE

- Description: the date format is yyyymmdd, describing the exact date of transaction in 2015.
- Distinct Value: 365
- Percent Populated: 100%

The plot above illustrates the dates throughout the year 2016, from which we can tell there is no discernable anomaly in the distribution.

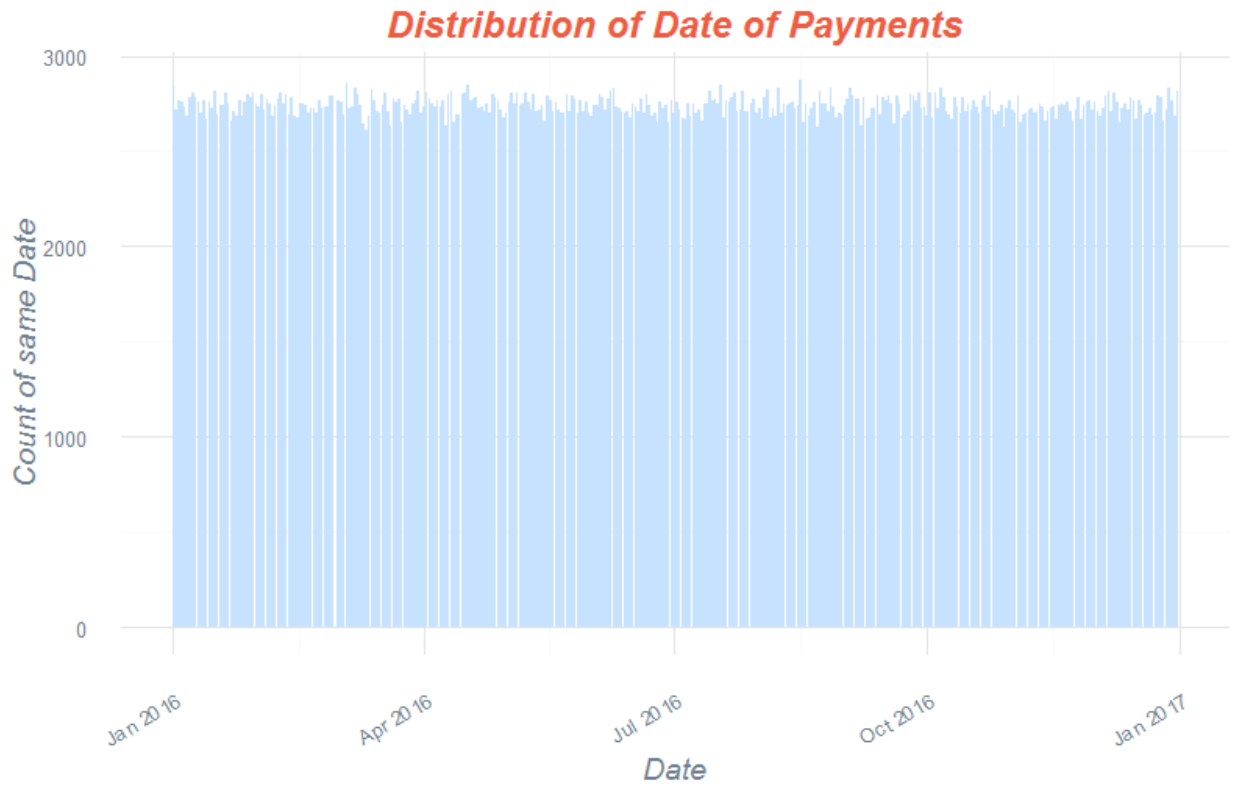


Figure 1 Distribution of Date of Payments

3.SSN

- Description: The social security number of each applicant.
- Distinct Value: 96535
- Percent Populated: 100%

The distribution of SSN is shown in the figure 2, the graph indicates there is an anomaly 737610282, which appears extreme frequently. However, after further investigation, it is proved to be a frivolous value. The optimal way to deal with is neutralize or ignore in the next phase.

The counts of other SSNs are too small to be observed in the presence of this frivolous SSN. To have a better understanding of other SSNs, I remove the frivolous SSN and pick the Top 20 SSN which show up most frequently apart from the frivolous one in figure 3.

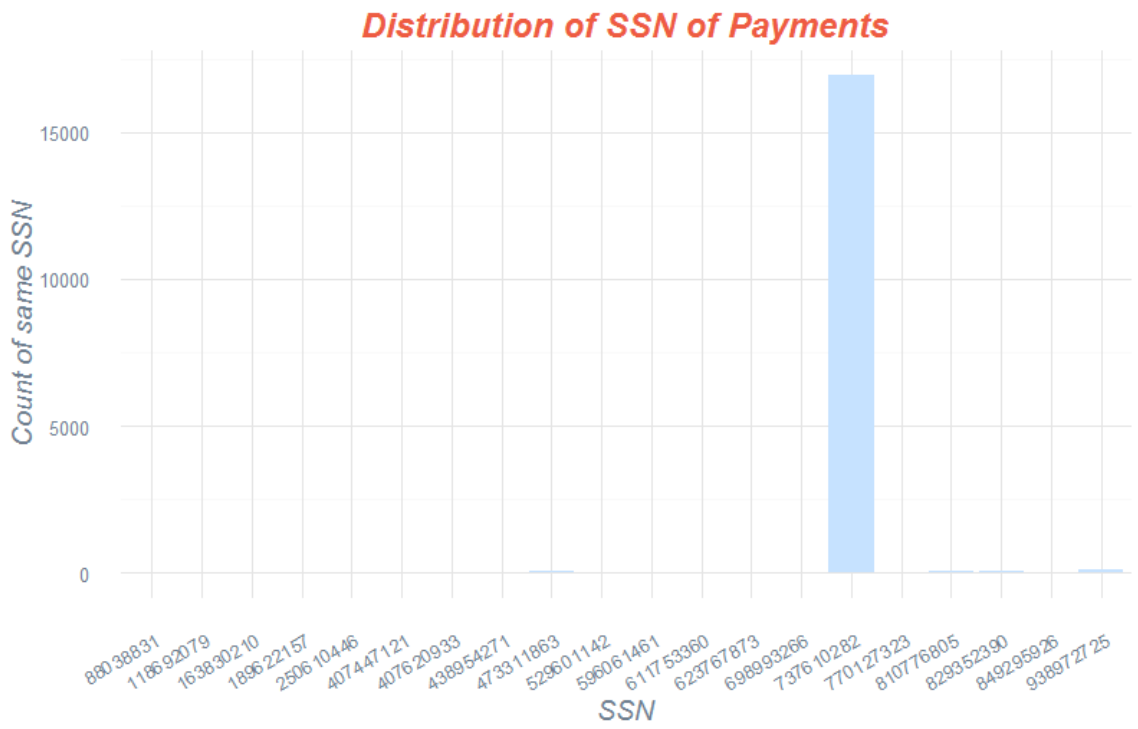


Figure 2 Distribution of SSN of Payments

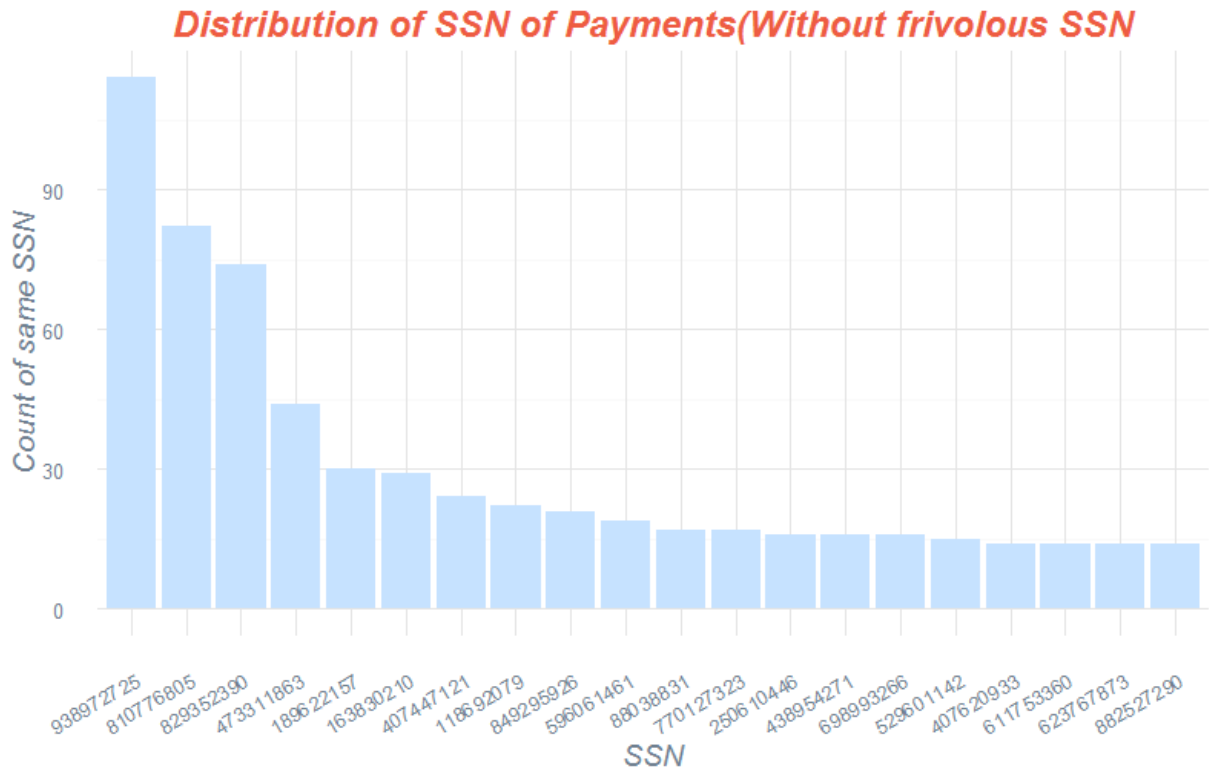


Figure 3 Distribution of SSN of Payments (Without Frivolous SSN)

4.FIRSTNAME

- Description: The first name of the applicant
- Distinct Value: 16576
- Percent Populated: 100%

The distribution of the Top 30 frequent first name is shown in the figure 4. The most frequent first name is EAMSTRMT, and 12,648 observations have it as their first name. Although there are many first name repetitively appears more than hundreds of times, we should be discreet on the issue on reporting frivolous value by looking at the dataset by combining last name, ssn.

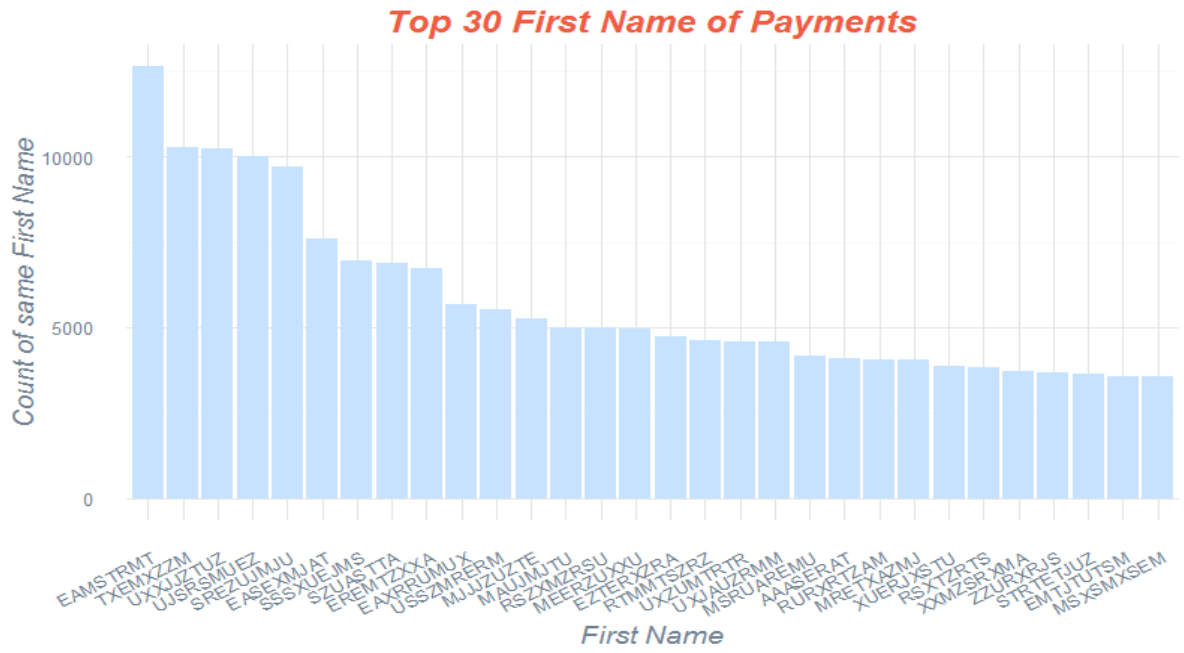


Figure 4 Top 30 First Name of Payments

5.LASTNAME

- Description: The last name of the applicant.
- Distinct Value: 36312
- Percent Populated: 100%

Figure 5 summarizes the top 30 frequent values of last name, although there are many last names repetitively appears more than hundreds of times, we should be discreet on the issue on reporting frivolous value, perhaps combine the field with other fields such as first name, dob.

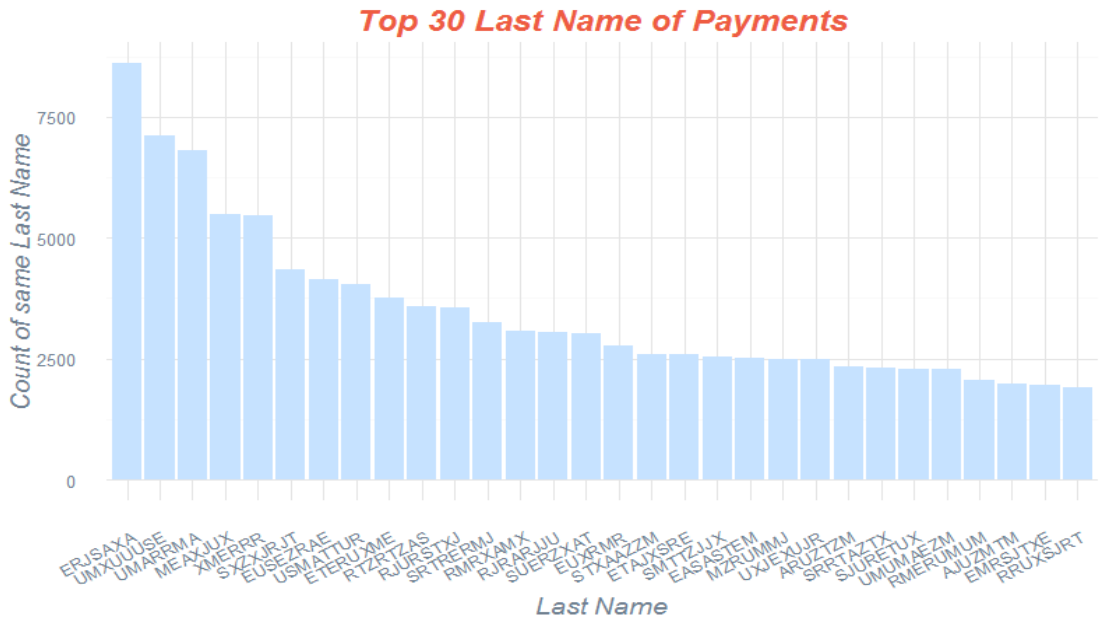


Figure 5 Top 30 Last Name of Payments

6. ADDRESS

- Description: The home address of each applicant.
- Distinct Value: 97563
- Percent Populated: 100%

The distribution of the Top 30 frequent address is shown in the figure 6. There is probably a frivolous address, 2602 AJTJ AVE. Figure 7 shows the top 30 address without frivolous one.

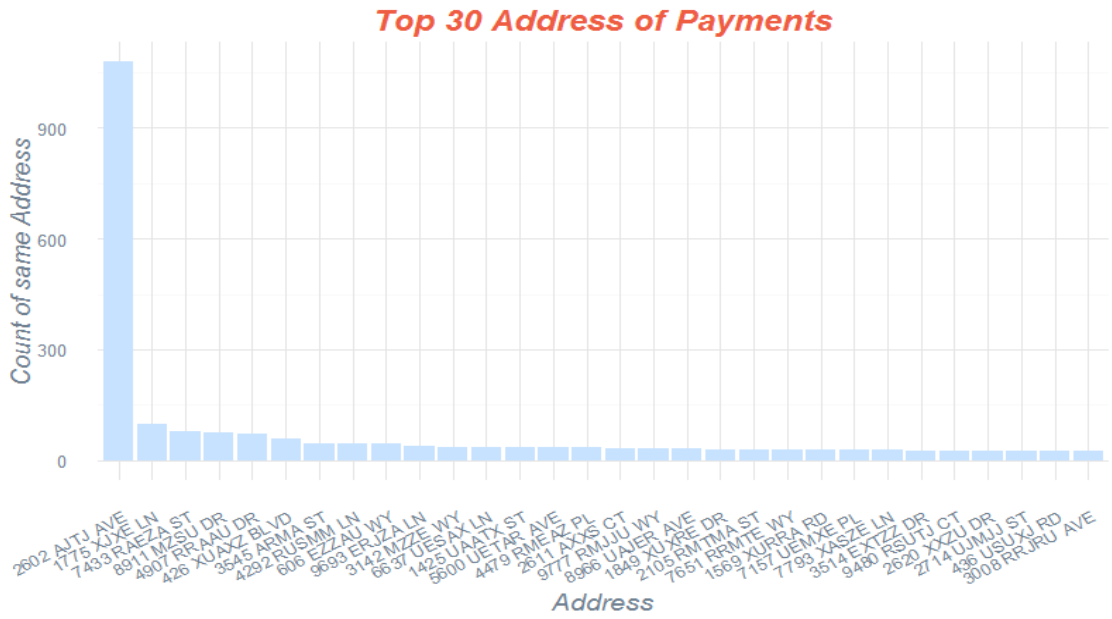


Figure 6 Top 30 Address of Payments

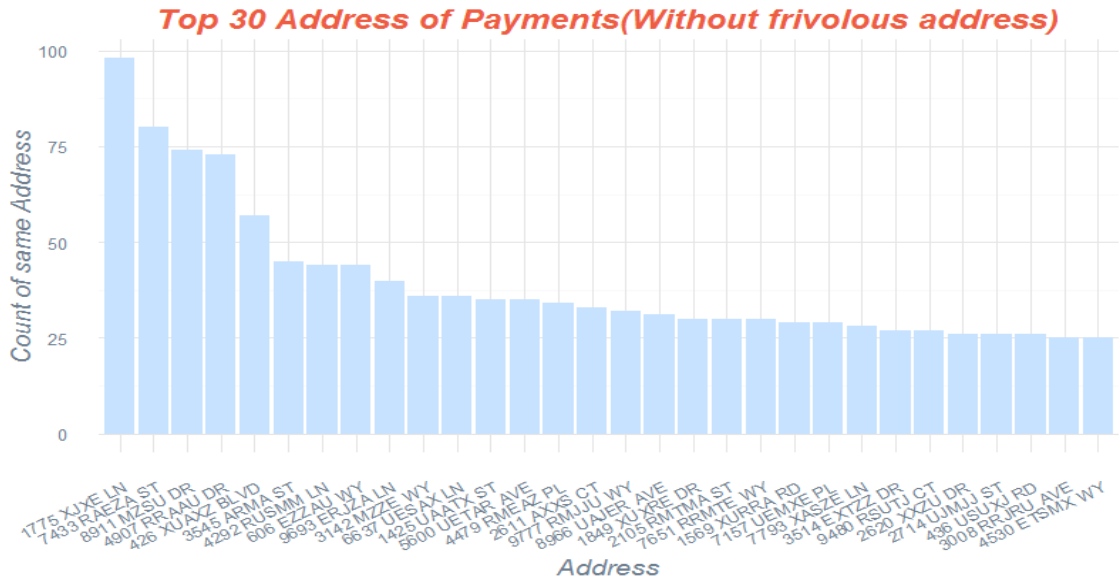


Figure 7 Top 30 Address of Payments (Without frivolous address)

7. ZIP5

- Description: The zip code of the residence address of each applicant.
- Distinct Value: 16547
- Percent Populated: 100%

Figure 8 summarizes the top 30 frequent values of zip5, and it can be noticed that the zip code of 68138 is the most frequent.

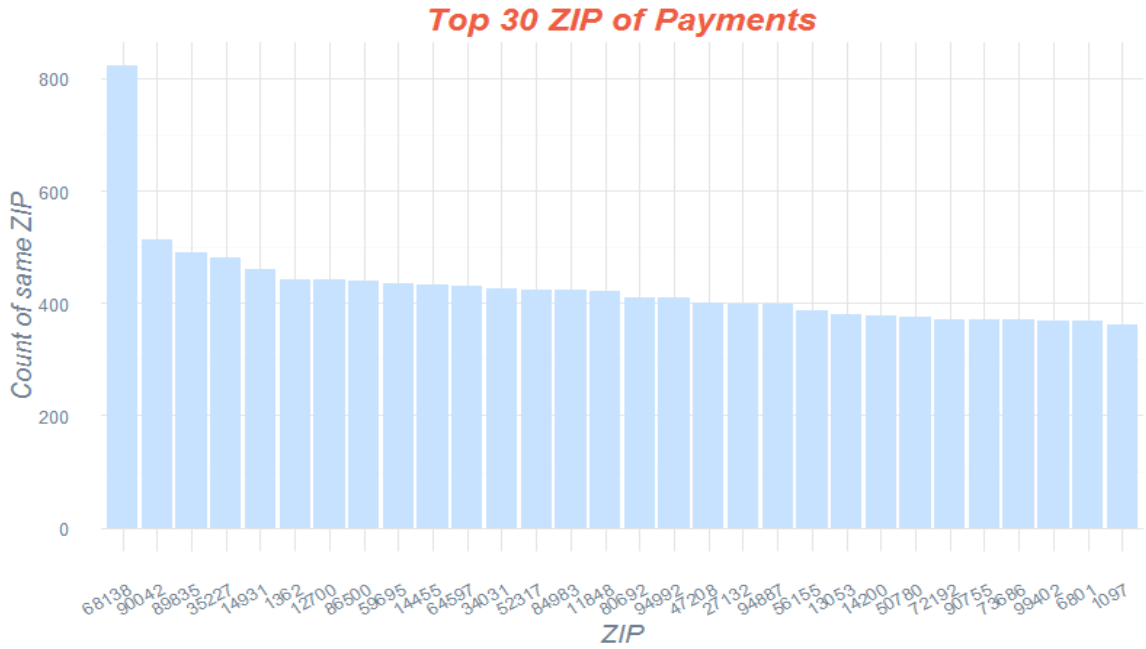


Figure 8 Top 30 ZIP code of Payments

8. DOB

- Description: The date of birth of each applicant.
- Distinct Value: 36816
- Percent Populated: 100%

Instead of plotting the distribution of dob, the distribution of Year of Birth might give us a primary understanding of the dob distribution. There is a frivolous year of birth which has more than 100,000 observations in the figure 9.

To get a further understanding of the frivolous year 1907, a distribution of dob in that specific year will help us figure out which dates in that year lead to the frivolous date of birth.

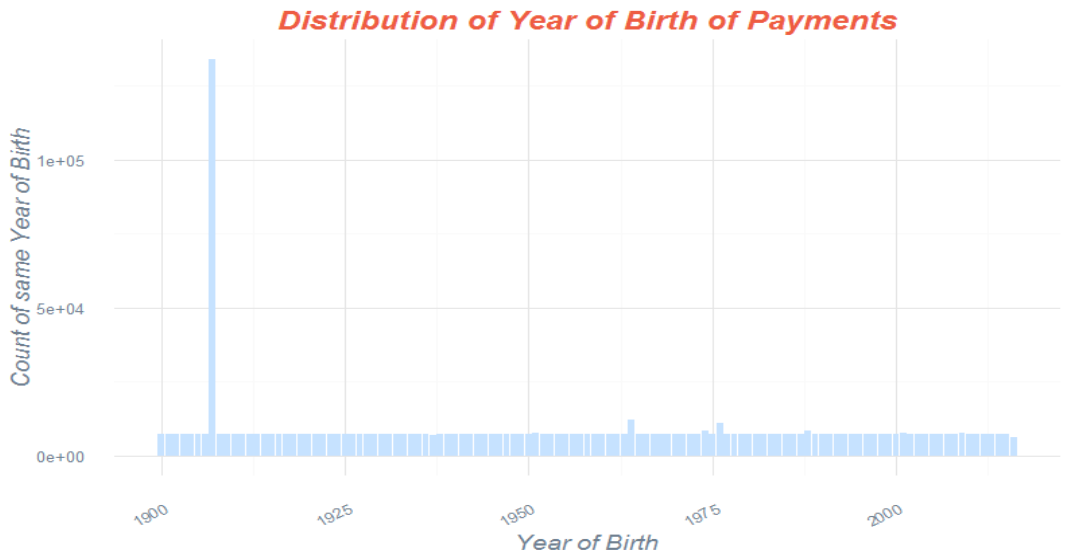


Figure 9 Distribution of Year of Birth of Payments

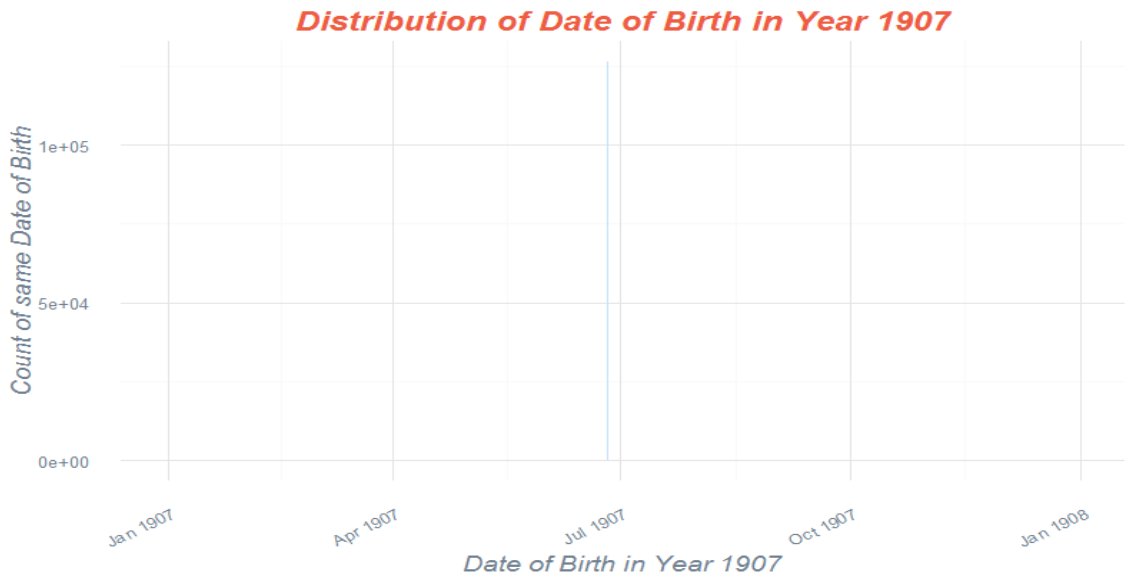


Figure 10 Distribution of Date of Birth in Year 1907

9. HOMEPHONE

- Description: The home phone number of each applicant.
- Distinct Value: 22181
- Percent Populated: 100%

Figure 11 shows the Top 20 frequent home phone, and the home phone number of 9105580920 is distinctively frequent, which is highly suspicious. Figure 12 shows the Top 20 home phone without the frivolous number.

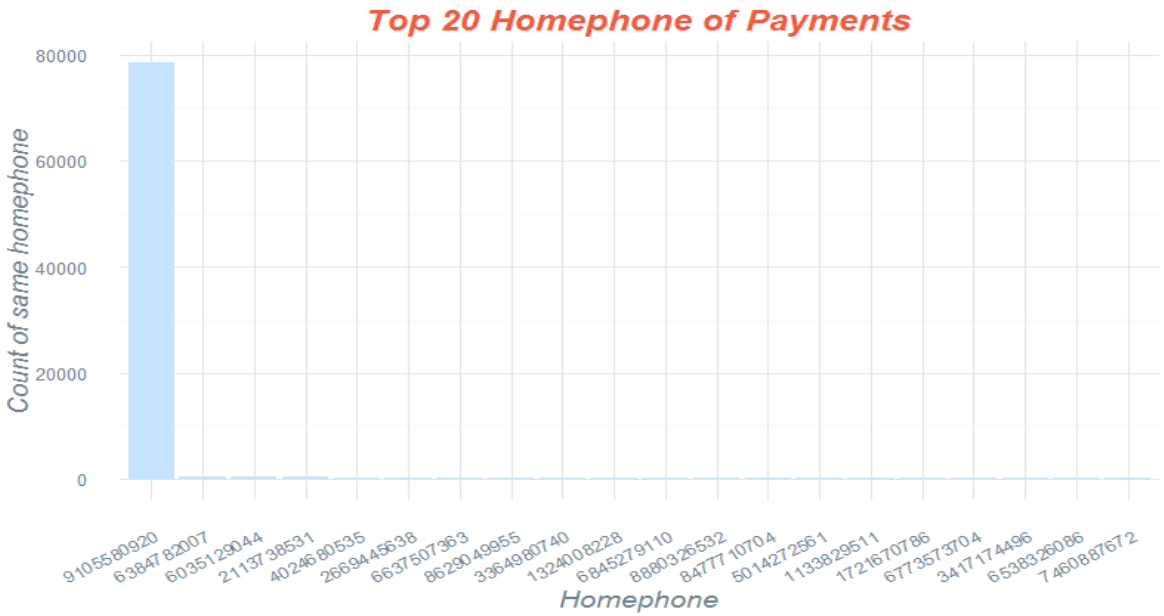


Figure 11 Top 20 Home phone of Payments

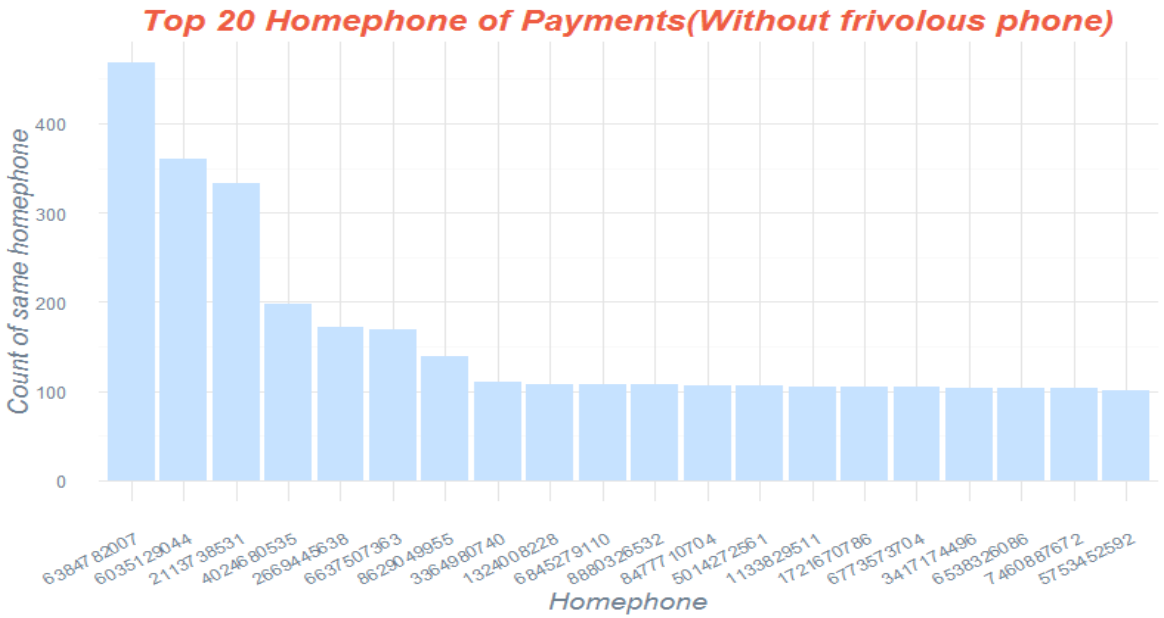


Figure 12 Top 20 Home phone of Payments (Without frivolous home phone)