

Product Application Identity Fraud Detection



**Yimeng Huang, Lingxiao Lyu, Haotian Wu
Jinze Xin, Yijia Xu, Ruoyan Zhou**

**DSO 562 Fraud Analytics
Project 2 Report
March 25, 2021**

Table of Contents

I. Executive Summary	2
II. Description of Data	3
III. Data Cleaning	7
IV. Candidate Variables	9
V. Feature Selection Process	16
VI. Model Algorithms	21
1. Logistic Regression (Base)	21
2. Decision Tree Classifier	22
3. Random Forest	24
4. Gradient Boosting	25
5. Neural Network	28
6. Stacked Ensemble Model	30
VII. Results	34
VIII. Conclusions	39
IX. Appendix	41
Appendix A. Data Quality Report (DQR)	41
Appendix B. Candidate Variables	48
Appendix C. Full Final Result Tables	60

I. Executive Summary

Fraud detection has been an unresolved problem for many financial institutions over decades and gained growing attention due to an increasing size of credit applications in the latest technology age. However, many institutions are still facing huge financial loss because of their poor ability to catch fraudulent applications. Thus, responding to a growing need for fraud detection, the project utilizes the latest data analytics methodologies, including basic classification and complex machine learning algorithms to better identify fraud applications.

The dataset we employed is a synthetic dataset that mimics real application data in the form of fields and field relationships occurrences, with a size of ten fields and one million unique records. As the first step, we created 871 candidate variables from the original nine entities with the fraud label aside. Next, we split our data into a modeling (training-testing) set and an out-of-time (OOT) validation set on 10/31/2016. Then we utilized two data filtering methods – univariate Kolmogorov-Smirnov (referred as KS) scores and Fraud Detection Rate (FDR) at top 3% calculation. By averaging the rankings from the two filters, we obtain the top 170 candidate variables (excluding the target variable “fraud_label” itself). After, we applied logistic regression and decision tree backward selection methods as data wrappers using the recursive feature elimination with cross validation selection (RFECV) to reduce dimensions of the dataset to only 30 features.

Thus, the final modelling data has a size of 794,996 lines of records and 30 input features, with “fraud_label” as the output variable. At the final modelling stage, we first passed the modelling data into Logistic Regression Model and Decision Tree Classification as two base models, each with different combinations of hyperparameters. Next, we implemented Random Forest, Gradient Boosting, and Neural Net, each with different combinations of hyperparameters. Finally, we also tried the Stacked Ensemble Model in hope of further improving the classification accuracy.

Using FDR as the performance measure, we compared all modeling results and selected Gradient Boosting Tree Model (GBM) as our final model. Next, we determined the optimal set of parameters by testing with various cases. Subsequently, we trained and tested the model using the complete modelling data and predicted results on OOT validation data. Our best model caught 56.9% of the fraudulent records in the testing set and 55.6% of the fraudulent records in the OOT validation set at a 3% FDR after parameter tuning.

II. Description of Data

1. Dataset High Level Description

Dataset Name: Application Data

Time Period: 2016-01-01 to 2016-12-31

Number of Fields: 10

Number of Records: 1,000,000

Dataset Purpose and Source:

The application data contains product (credit cards and cell phones) application data with personal identification information for the purpose of finding and labelling application/identity frauds through machine learning. It is a synthetic dataset built from studying the statistical properties of over a billion real U.S. applications over about ten years by an identity fraud prevention company. It was built so as to reproduce the important univariate and multivariate field distributions of real data.

A summary statistics table of all the data fields is shown below.

2. Summary Statistics Table

Field Name	Field Type	dtype	# Records	% Populated	# Unique Values	Most Common Field Value	% Most Common Field Value	Minimum Value	Maximum Value
record	categorical	int64	1000000	100	1000000	N/A	N/A	N/A	N/A
date	date	int64	1000000	100	365	20160816	0.29	20160101	20161231
ssn	categorical	int64	1000000	100	835819	999999999	1.69	N/A	N/A
firstname	categorical	object	1000000	100	78136	EAMSTRMT	1.27	N/A	N/A
lastname	categorical	object	1000000	100	177001	ERJSAXA	0.86	N/A	N/A
address	categorical	object	1000000	100	828774	123 MAIN ST	0.11	N/A	N/A
zip5	categorical	int64	1000000	100	26370	68138	0.08	N/A	N/A
dob	date	int64	1000000	100	42673	19070626	12.66	19000101	20161031
homephone	categorical	int64	1000000	100	28244	999999999	7.85	N/A	N/A
fraud_label	categorical	int64	1000000	100	2	0	98.56	N/A	N/A

* Note:

- Field Type: how we treat each field
- dtype: how Python reads each field without any transformation
- N/A: value not applicable for the nature of the field

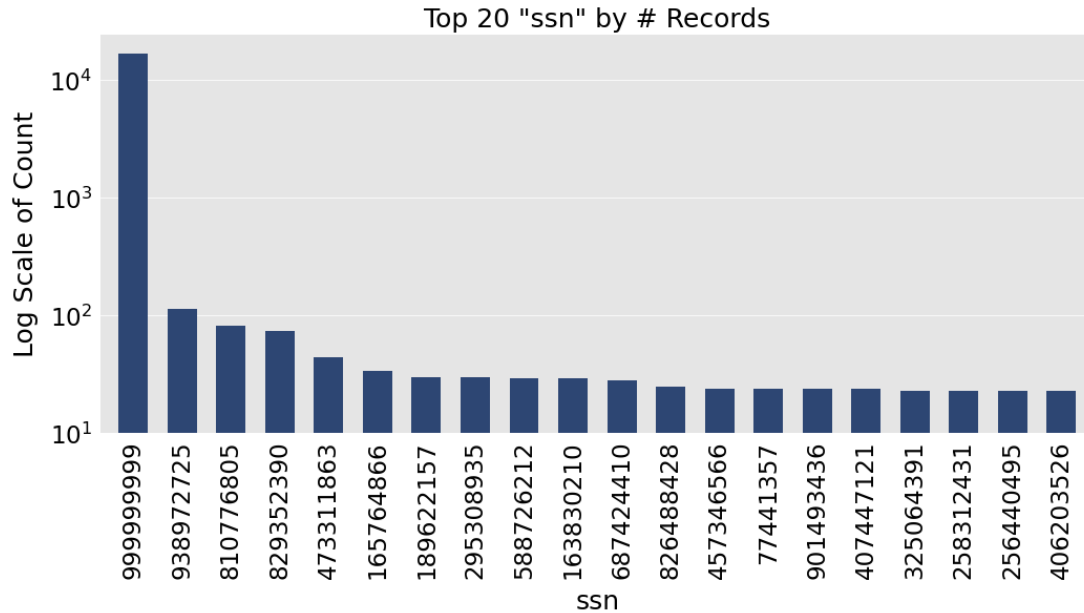
3. Important Field Distributions

Some data fields seem to have very unbalanced distributions or counts of records. Therefore, we would like to bring these fields and their distributions to your attention.

A. Data Field: "ssn"

Description: Social Security Number of the applicant.

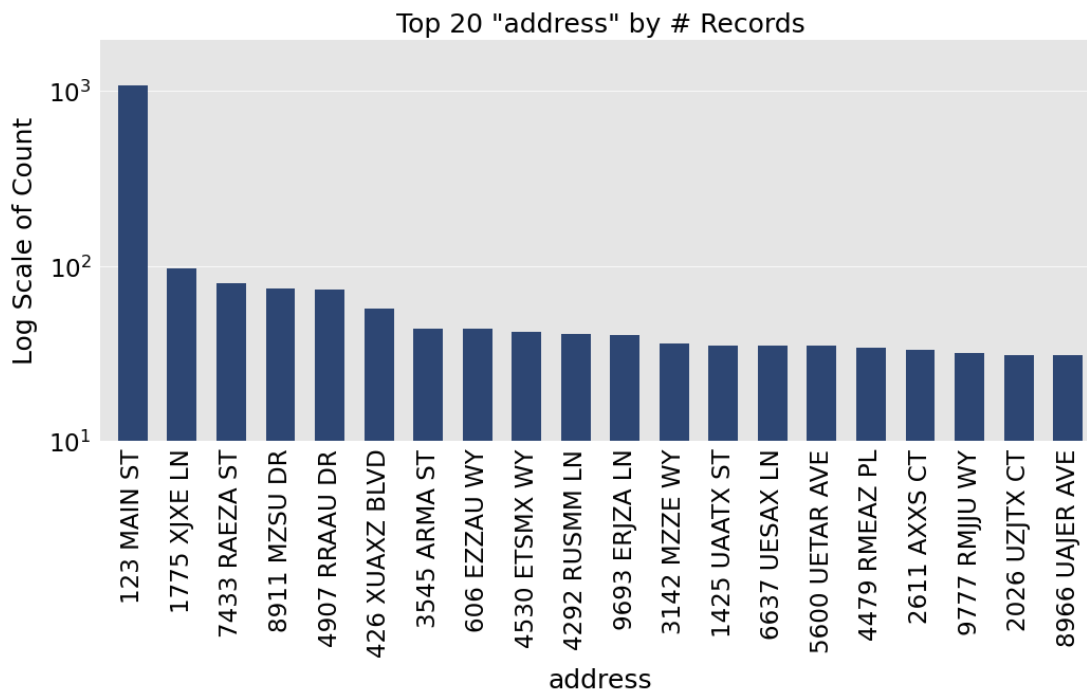
As shown in the figure below, the most frequent value of "ssn" is "999999999". It has 16,935 records, which is much more than the other "ssn" values. This value should be considered a frivolous value and will require special treatment during the data cleaning process.



B. Data Field: "address"

Description: Street address of the applicant.

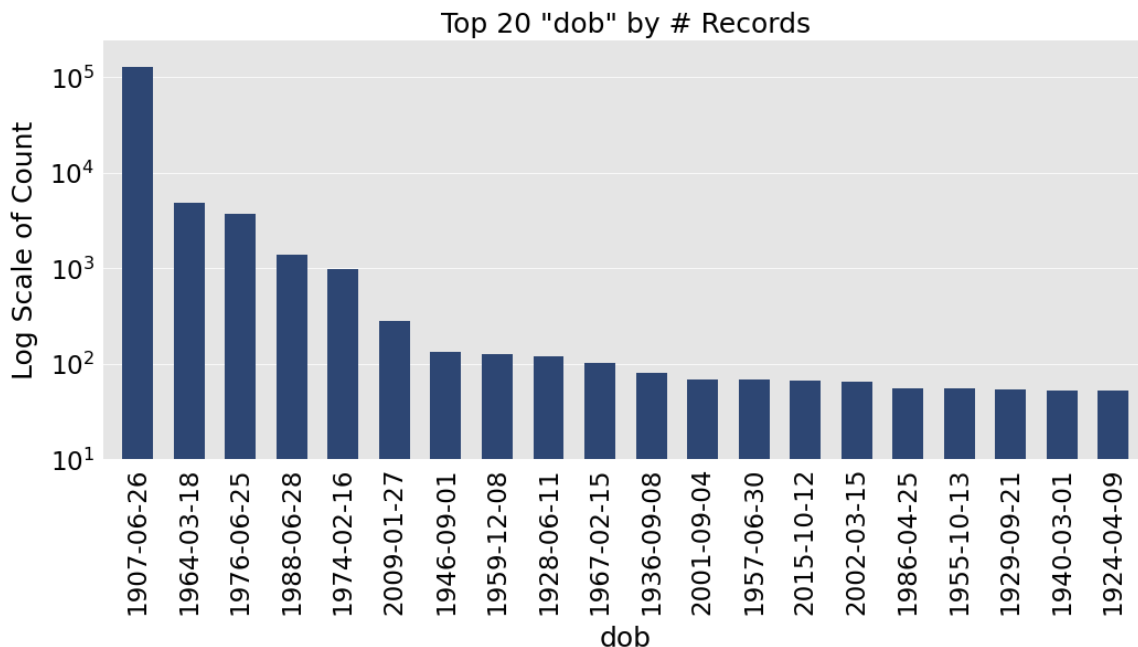
As shown in the figure below, the most frequent value of "address" is "123 MAIN ST". It has 1,079 records, which is much more than the other "address" values. This value is suggested to be considered a frivolous value and will require special treatment during data cleaning.



C. Data Field: "dob"

Description: Date of birth of the applicant, ranging from 1900-01-01 to 2016-10-31.

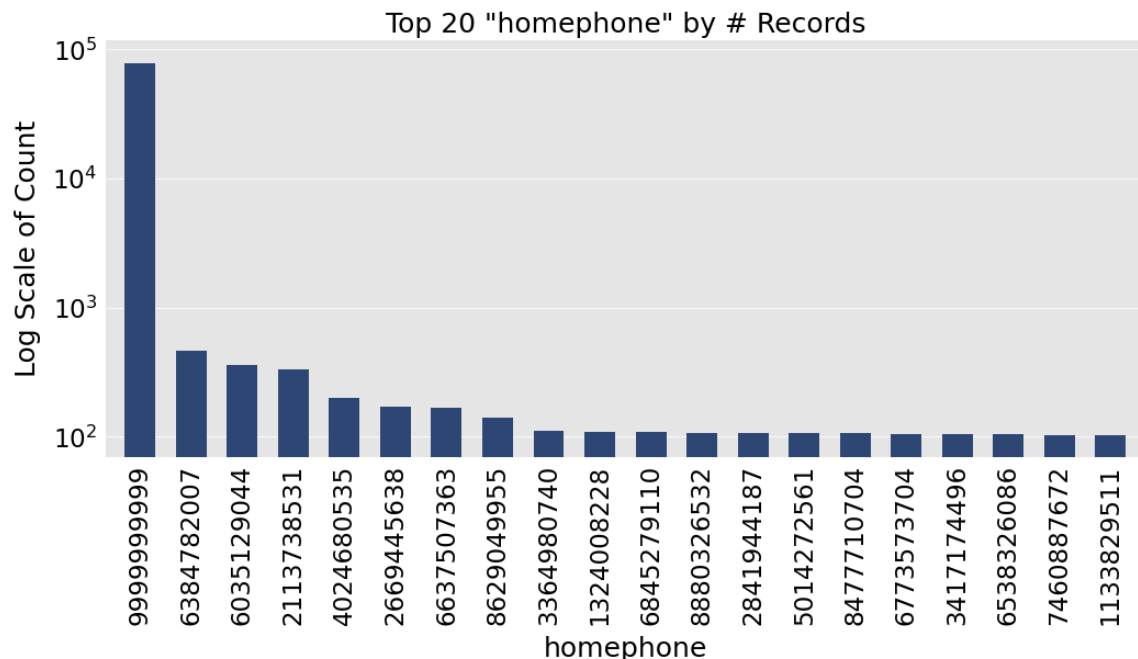
As shown in the figure below, the most frequent value of "dob" is "1907-06-26". It has 126,568 records, which is much more than the other "dob" values. This value is suggested to be considered a frivolous value and will require special treatment during the data cleaning process.



D. Data Field: "homephone"

Description: Home phone number of the applicant.

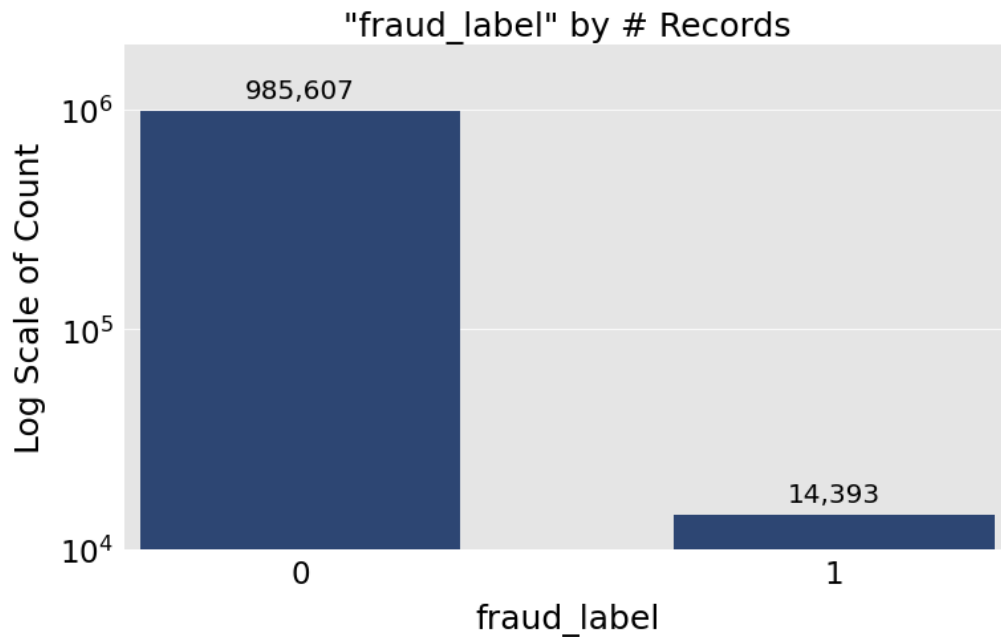
As shown in the figure below, the most frequent value of "homephone" is "9999999999". It has 78,512 records, which is much more than the other "homephone" values. This value should be considered a frivolous value and will require special treatment during the data cleaning process.



E. Data Field: "fraud_label"

Description: Whether the application is a fraud or not.

"fraud_label" is the target variable, a binary categorical field with "1" representing the application as a fraud. Only around 1.4% of the records – 14,393 applications – are identified as frauds while the majority (98.6%) of the records – 985,607 applications – are not fraudulent applications.



Other fields: Distributions and detailed information of the other fields can be found in Appendix A at the end of this report.

III. Data Cleaning

Data cleaning is an important step to prepare the data for variable creation, feature selection, and final modeling. This dataset is 100% populated; however, according to our Data Quality Report, certain unsuitable formats and frivolous field values did exist and would require special treatments. In this process, we did not transform the “record” and “fraud_label” fields.

1. Changing Field Formats

As shown in the summary statistics table above, some fields were read as integers by Python, but we would treat them as categorical or date values for our analyses later; these fields include “date”, “ssn”, “zip5”, “dob”, and “homephone”. Among these fields, “ssn”, “dob”, and “homephone” contained frivolous values, so we would change their formats in the next step.

First, we parsed the “date” field into the format of “YYYY-MM-DD.” For example, the value “20160101” was parsed as “2016-01-01”. In this way, it would be easier to read and analyze the “date” field with date-related attributes. For instance, we could identify the day of the week of a specific date value.

Next, we would want to ensure that all the “zip5” values actually contain five digits. Therefore, for those “zip5” values with fewer digits, we added “0” in the front. For example, the value “2765” was changed to “02765”.

2. Treating Frivolous Field Values

For this dataset, the frivolous values could be considered equivalent to missing values and had no specific meanings. While we decided that they were not risky, we knew our variable creation process later would involve linking between different fields and these frivolous values would create unnatural links and result in wrong variable values. These frivolous values would cause high numbers of links, which are risky. Therefore, we would want to change them in a way that results in low numbers of links. Consequently, we chose to replace them with unique values by using the “record” field, which is a unique identifier for each entry in the data.

A. Field “ssn”

The frivolous value for the field “ssn” is “999999999”, having much more counts than other values in the “ssn” field. We replaced each frivolous value with the negative of its corresponding record number to make it unique. We then wanted to ensure that all “ssn” values have nine digits, so for the ones with fewer digits, we added “0” in the front. For example, the frivolous value “999999999” with a record number of “11” was first changed to “-11” and then to “000000-11”.

B. Field “address”

The frivolous value for the field “address” is “123 MAIN ST”, having much more counts than other values in the “address” field. We replaced each frivolous value with its corresponding

record number with a text " RECORD". For example, if the initial "address" value was "123 MAIN ST" with a record number of "1248", we transformed it to "1248 RECORD".

C. Field "dob"

The frivolous value for the field "dob" is "19070626", having much more counts than other values in the "dob" field. We treated these values in the same way as treating the frivolous values of "ssn" above: replacing with the negative of the corresponding record number and adding "0" in the front to fill up eight digits. For example, the original "dob" value of "19070626" with a record number of "1" was transformed to "000000-1".

D. Field "homephone"

The frivolous value for the field "homephone" is "9999999999", having much more counts than other values in the "homephone" field. We treated these values in the same way as treating the frivolous values of "ssn": replacing with the negative of the corresponding record number and adding "0" in the front to fill up ten digits. For example, the original "homephone" value of "9999999999" with a record number of "18" was transformed to "0000000-18".

All the frivolous values in the original dataset were fixed for later analyses and modeling.

IV. Candidate Variables

In the original dataset, most of the fields are categorical in nature, which are not very useful as input variables when building models. Therefore, it is important for us to generate as many candidate variables as possible to be eventually used in our final models to identify fraud records. Consequently, we created candidate variables based on some common behavioral patterns of fraudsters, such as the frequency and periodicity of using fraudulent identities. As a result, we produced a total of 871 candidate variables, which can be broken down into the following categories:

- Risk Table Variable (1)
- Day Since Variables (22)
- Velocity Variables (132)
- Relative Velocity Variables (176)
- Number of Unique Contact Entity for Each Identity Entity (270)
- Number of Unique Identity Entity for Each Contact Entity (270)

Before creating our expert variables, we linked/concatenated some of the existing fields to serve as the base entities for our later creation process, as they could be more unique and informational to represent each applicant:

	Entity	Concatenation of Original Data Fields
1	name	firstname, lastname
2	fulladdress	address, zip5
3	name_dob	firstname, lastname, dob
4	name_fulladdress	firstname, lastname, address, zip5
5	name_homephone	firstname, lastname, homephone
6	fulladdress_dob	address, zip5, dob
7	fulladdress_homephone	address, zip5, homephone
8	dob_homephone	dob, homephone
9	homephone_name_dob	homephone, firstname, lastname, dob
10	ssn_firstname	ssn, firstname
11	ssn_lastname	ssn, lastname
12	ssn_address	ssn, address
13	ssn_zip5	ssn, zip5
14	ssn_dob	ssn, dob
15	ssn_homephone	ssn, homephone
16	ssn_name	ssn, firstname, lastname
17	ssn_fulladdress	ssn, address, zip5
18	ssn_name_dob	ssn, firstname, lastname, dob

In addition to these 18 entities we created, we added “ssn”, “address”, “dob”, and “homephone” from the original dataset to the base entity list. Therefore, we had a total of 22 entities for the expert variables creation process.

Note: Since this project involves a real-time fraud algorithm, we must treat the time flow correctly during the variable creation process. Therefore, when creating variables, we should only use data in the past for each record along with the record itself. We would also reserve the last two months of records – records after 10/31/2016 – as the out-of-time (OOT) sample for model validation and not use them for training and testing models. Consequently, for some of our variables created, we should not take into account records in the OOT sample. All the records before 11/1/2016 would be included in the training-testing set.

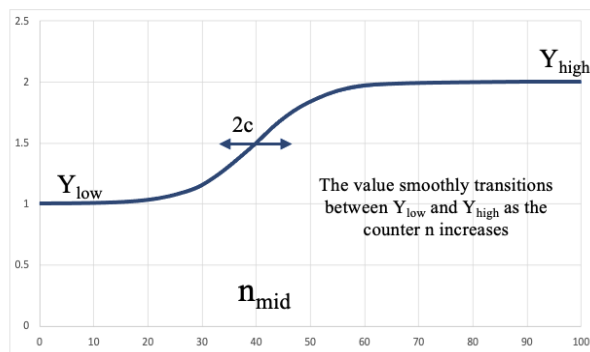
The procedure of creating variables of each category is described in detail below.

1. Risk Table Variable

Target encoding (a.k.a risk table) is a great way of encoding categorical variables because while it transforms categorical values into numerical values that directly encode the target variable, it would not cause dimensionality expansion – dramatically increase the number of variables. Before conducting target encoding, we first created a new column called “dow” – “day of week” – to prepare for target encoding. It was created by identifying the day of the week for each “date” field value. For each day of the week, we would calculate the proportion of fraud records. However, this calculation should not consider records in the OOT sample as mentioned before.

For all the records in the training-testing set, we grouped the records by the “dow” (day of week) column and calculated the proportion of frauds within each group. Then, we applied the smoothing formula to assign the fraud proportion of each weekday to each record based on the corresponding day of week in a column called “dow_risk”. The smoothing formula was just to ensure that each “dow” group had enough records to be assigned with a value; if there were not enough records, then the group would be assigned with a default value – the overall proportion of fraud records in the training-testing set.

$$\text{Value} = Y_{\text{low}} + \frac{Y_{\text{high}} - Y_{\text{low}}}{1 + e^{-(n - n_{\text{mid}})/c}}$$

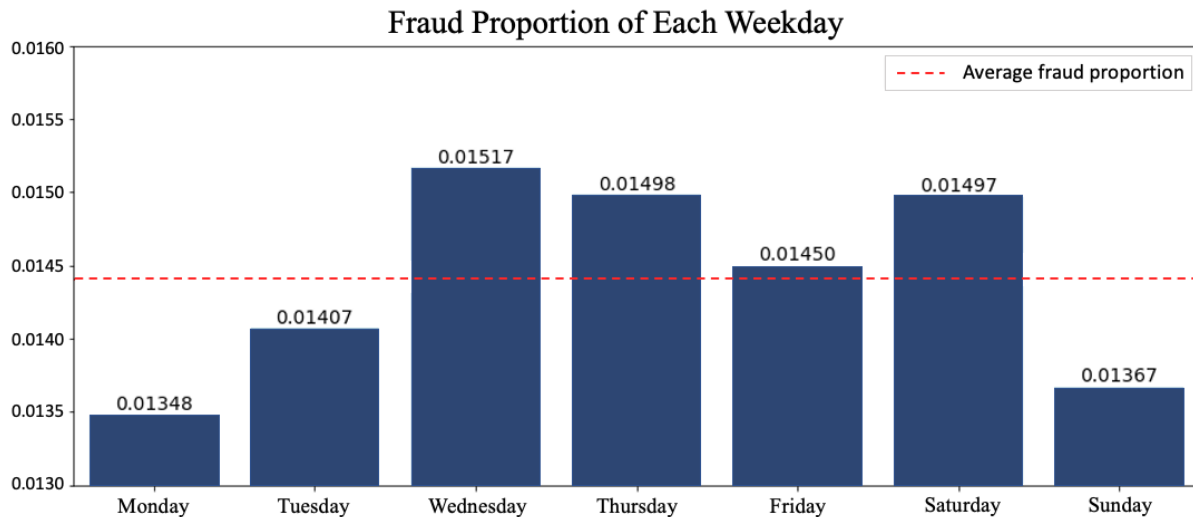


Smoothing counter n
Could be integer or continuous

In our case:

- Y_{low} is the average of “fraud_label” within the training-testing dataset.
- Y_{high} is the average of “fraud_label” for each weekday.
- $c = 4$
- $n_{mid} = 20$
- n = size of each group (e.g. Monday, Tuesday)

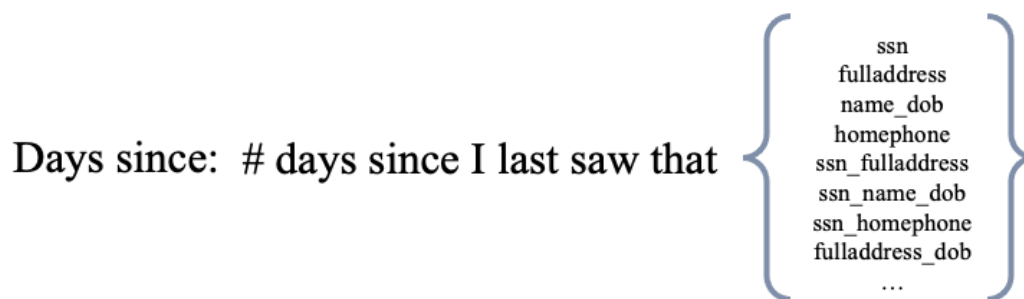
Below is a visualization of our risk table values:



2. Days-Since Variables

Days-since variables measure the number of days since the same entity value was last seen.

Below is a visual representation of some examples:



When a fraudster attempts to submit fraudulent applications, he or she may repeatedly use certain information of either him- or herself or the victim. Therefore, if a same entity value was seen very frequently, there would be a higher risk or probability of the applications being fraudulent, meaning that a lower value of a days-since variable indicates a higher risk.

We calculated the days-since variables by subtracting the date of the most recent record in the past with the same entity value from the date of each current record. If there was no calculated result returned, it means that we have never seen the same entity value before the current date, which indicates very low risk. Therefore, we would want to fill these null values with a

value that does not indicate high risks, which would be “365” in this case. This is because our data ranges over an entire year, and “365” suggests that we have not seen a record with the same entity value over the entire year. Here is a complete table of all the 22 days-since variables we created, one for each base entity:

Days-Since Variable Name		Days-Since Variable Name	
1	ssn_day_since	12	dob_homephone_day_since
2	address_day_since	13	homephone_name_dob_day_since
3	dob_day_since	14	ssn_firstname_day_since
4	homephone_day_since	15	ssn_lastname_day_since
5	name_day_since	16	ssn_address_day_since
6	fulladdress_day_since	17	ssn_zip5_day_since
7	name_dob_day_since	18	ssn_dob_day_since
8	name_fulladdress_day_since	19	ssn_homephone_day_since
9	name_homephone_day_since	20	ssn_name_day_since
10	fulladdress_dob_day_since	21	ssn_fulladdress_day_since
11	fulladdress_homephone_day_since	22	ssn_name_dob_day_since

3. Velocity Variables

Velocity variables measure the number of records with the same entity value seen over the last 0, 1, 3, 7, 14, 30 days. Below is a visual representation of the general equation:

$$\text{Velocity: \# records with the same } \left\{ \begin{array}{c} \text{ssn} \\ \text{fulladdress} \\ \text{name_dob} \\ \text{homephone} \\ \text{ssn_fulladdress} \\ \text{ssn_name_dob} \\ \text{ssn_homephone} \\ \text{fulladdress_dob} \\ \dots \end{array} \right\} \text{ over the last } \{0, 1, 3, 7, 14, 30\} \text{ days}$$

Similar to the days-since variables introduced above, this type of variables can also suggest risk associated with frequency, that is, the more frequently a same entity value was seen over a certain period, the riskier the application might be. Therefore, a higher value of a velocity variable may indicate a higher risk of fraud.

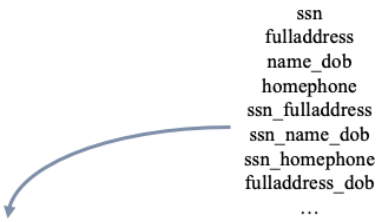
To calculate this type of variables, we examined each record with the records over the past n days (0, 1, 3, 7, 14, 30 days) with the same entity value and calculated their counts. Note that the number of records with the same entity over the last 0 days will give us the number of records with the same entity value seen within the same day. As we had 22 base entities and six timeframes, there were a total of $22 \times 6 = 132$ velocity variables. The table below gives some examples of the velocity variables. For a full list of them, please refer to Appendix B.

Velocity Variable Name		Velocity Variable Name	
1	ssn_count_0	6	ssn_count_30
2	ssn_count_1	7	address_count_0
3	ssn_count_3	8	address_count_1
4	ssn_count_7	9	address_count_3
5	ssn_count_14	10	address_count_7

4. Relative Velocity Variables

Relative Velocity variables measure the number of records with an entity value seen in a short time period (e.g. 1 day) over the number of records with the same entity value seen in a longer time period (e.g. 30 days). A higher number seen in a short period of time compared to its usual activity shows abnormality, indicating a higher risk of fraud. Therefore, for the relative velocity variables, the higher the ratio, the higher the probability of fraud. Below is a visual representation of the general equation:

Relative velocity:



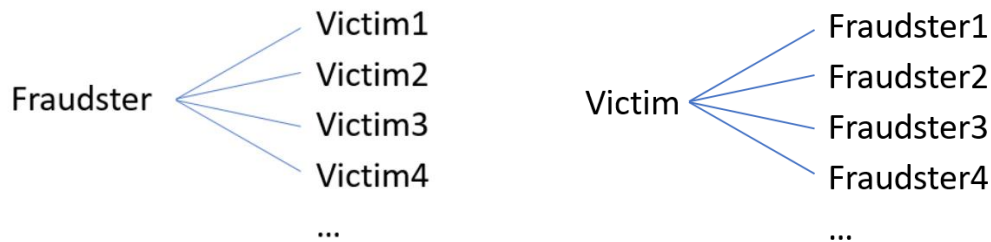
$$\frac{\text{\# apps with that *group* seen in the recent past}}{\text{\# apps with that *same group* seen in the past \{ 3, 7, 14, 30 \} days}}$$

In the numerator, we used 0- and 1-day velocity variable values as the “recent past” data. In the denominator, we used 3-, 7-, 14-, and 30-day velocity variable values. This gives us a total of $22 \times 2 \times 4 = 176$ relative velocity variables. Some examples of these variables are shown in the table below. For a full list of relative velocity variables, please refer to Appendix B.

Relative Velocity Variables		Relative Velocity Variables	
1	ssn_count_0_by_3	11	address_count_0_by_14
2	ssn_count_0_by_7	12	address_count_0_by_30
3	ssn_count_0_by_14	13	address_count_1_by_3
4	ssn_count_0_by_30	14	address_count_1_by_7
5	ssn_count_1_by_3	15	address_count_1_by_14
6	ssn_count_1_by_7	16	address_count_1_by_30
7	ssn_count_1_by_14	17	dob_count_0_by_3
8	ssn_count_1_by_30	18	dob_count_0_by_7
9	address_count_0_by_3	19	dob_count_0_by_14
10	address_count_0_by_7	20	dob_count_0_by_30

5. Number of Unique Entity 1 for Each Entity 2 over Past N Days

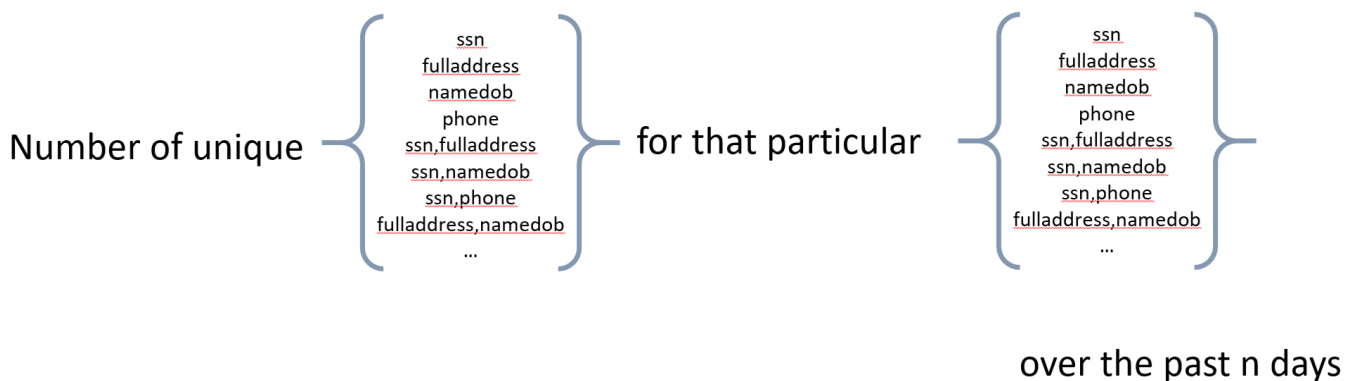
Normally, there are two potential patterns in fraudulent applications. First, an individual fraudster uses different victims' identities but his or her own contact information to fill in applications, so the same contact information may correspond to different identity information. On the opposite, a victim's information may be used by several different fraudsters, so the same identity information may correspond to different contact information. The pictures below illustrate these two patterns:



Therefore, following these patterns, we deliberately selected two groups of base entities from the 22 entities, one with identity-related information and the other with contact-related information:

- Identity group (9): "ssn", "dob", "name", "name_dob", "ssn_firstname", "ssn_lastname", "ssn_dob", "ssn_name", "ssn_name_dob"
- Contact group (5): "address", "homephone", "fulladdress", "fulladdress_homephone", "ssn_zip5"

The unique count variables measure the number of unique identity-related values used for each contact-related value or the number of unique contact-related values used for each identity-related value over the past n days (0, 1, 3, 7, 14, 30 days). A visual representation of the general equation is shown below:



For example, by counting the number of different "ssn" used for the same "address" over a period of time, we would think that the higher the count, the higher the probability that it could be a fraudulent activity. The **Entity 1** and **Entity 2** here can be replaced by any entity in each of the **identity group** and **contact group**. For each time period (0, 1, 3, 7, 14, 30 days), we conducted the unique count computation. We ended up with $2 \times (9 \times 5 \times 6) = 540$ unique

count variables. A snapshot of some of our unique count variables is shown below. For a full list of velocity variables, please refer to Appendix B.

Unique Count Variables		Unique Count Variables	
1	#_unique_address_for_ssn_0	11	#_unique_homephone_for_ssn_14
2	#_unique_address_for_ssn_1	12	#_unique_homephone_for_ssn_30
3	#_unique_address_for_ssn_3	13	#_unique_fulladdress_for_ssn_0
4	#_unique_address_for_ssn_7	14	#_unique_fulladdress_for_ssn_1
5	#_unique_address_for_ssn_14	15	#_unique_fulladdress_for_ssn_3
6	#_unique_address_for_ssn_30	16	#_unique_fulladdress_for_ssn_7
7	#_unique_homephone_for_ssn_0	17	#_unique_fulladdress_for_ssn_14
8	#_unique_homephone_for_ssn_1	18	#_unique_fulladdress_for_ssn_30
9	#_unique_homephone_for_ssn_3	19	#_unique_fulladdress_homephone_for_ssn_0
10	#_unique_homephone_for_ssn_7	20	#_unique_fulladdress_homephone_for_ssn_1

V. Feature Selection Process

Feature selection is the process of reducing the dimensionality of our data, more specifically, the number of input variables fed into the final models. It not only helps to reduce the complexity of models and the computational cost of modeling but also helps to improve model performance by removing less relevant variables and noise from the data. There are three main methods of feature selection:

- 1) Filter Method: The filter method evaluates each variable by itself based on its individual importance in predicting the target variable, which is measured by some simple univariate statistics of each variable. Some examples of such univariate statistics include correlation, Fisher score, information value, and etc.
- 2) Wrapper Method: The wrapper method wraps a simple model around the feature selection process and builds many models to evaluate different subsets, or combinations, of variables. A stepwise selection method is typically used to build these models with gradually increasing or decreasing numbers of variables, meaning adding or removing one variable at each step.
- 3) Embedded Method: The embedded method does feature selection as the models are built. It is when we add a regularization term to the objective function to minimize model complexity by reducing the number and/or size of model parameters as much as possible.

In this project, our feature selection process mainly includes the first two methods: filter and wrapper. Our goal was to select the top 30 variables from our 871 candidate variables.

1. Dataset Preparation

Before conducting the actual feature selection, we split all of our data into a modeling (training-testing) set and an out-of-time (OOT) set. The former contains application records from 1/15/2016 to 10/31/2016 and is used to build, train, and test our models. The latter contains the last two months of records and is used to validate our models at the end. Our feature selection process uses only the modeling set as we cannot build models with “future” data (the OOT set) and thus cannot select features based on their “future” values. The first two weeks of records were also removed from the modeling set since these records are too early to have all the candidate variables fully formed. As a result, our modeling set at this stage contained 794,996 records and 872 variables (871 candidate variables and one target variable).

2. Filter Method

In this step, we wanted to rank all variables based on their univariate Kolmogorov-Smirnov (KS) and Fraud Detection Rate (FDR) scores and select the top variables.

- The KS score is a simple but robust measure of the maximum separation between two cumulative distributions. Therefore, the KS score of each variable measures how well a variable can separate the non-fraud and fraud records, which is a good indicator of variable importance in predicting fraud labels. The higher the KS score, the better the

separation between the non-fraud and fraud records, and the better the variable is in predicting fraud labels in this case.

- The FDR score measures the percentage of all the frauds caught at a particular examination cutoff point for each variable. It is very commonly used in business applications as it explicitly shows how many frauds can be detected by examining a certain proportion of the population sorted by a particular variable. The higher the FDR score, the better the variable is in terms of detecting frauds in this context.

In order to build and run a sanity check on the filter, we first added a random variable to the modeling dataset, which assigns a random value between 0 and 1 to each record. If the filter runs correctly, we would expect the KS and FDR scores of this random variable to be very low. On the opposite, we should expect the KS and FDR scores of the target variable, “fraud_label”, itself to have the highest value 1. The modeling set at this stage contained 794,996 records and 873 variables.

For each variable, we computed its KS score and FDR at 3% of the population. The KS scores were directly computed using a Python function. The FDR scores were computed by first sorting the dataset by each variable from the highest to the lowest, then taking the top or bottom 3% of the records and counting the number of frauds detected, and finally dividing this number by the total number of frauds in the dataset. When we sorted all variables by either the KS or the FDR scores from the highest to the lowest, we found the “fraud_label” variable at the top of the list with a KS and FDR of 1 and the random variable towards the bottom of the list with very low scores as expected.

We then assigned a KS ranking number and an FDR ranking number to each variable based on their respective scores from the lowest to the highest and then took the average of these two ranking numbers. Consequently, each variable is assigned with an average ranking number that takes into account both the KS and the FDR scores. Finally, we sorted all variables by their average rankings from the highest to the lowest. The higher the average ranking number, the more important the variable is in predicting fraud labels in our context.

From this filter result, we deliberately selected the top 170 candidate variables (excluding the target variable itself) to be further evaluated using the wrapper method in the next step. We decided to select these 170 variables because as we were examining the filter result, many variables of the same kind were assigned with very close ranking numbers; therefore, it was up to the top 170 variables that included more varieties of variables.

3. Wrapper Method

In order to select our final top 30 variables from the 170 candidate variables selected from the filter, we built two wrappers using the recursive feature elimination with cross validation selection (RFECV). The two wrappers are logistic regression backward selection and decision tree backward selection. Before introducing the wrapper-based feature selection process, we would like to briefly explain some of the technical terms:

- RFECV is an algorithm designed to automatically select an optimal number of features based on cross-validated feature importance or scores. Cross validation, or CV, means that the model is trained and evaluated over the dataset several times with different splitting methods in order to return the final ranks of variables. While RFECV assigns a ranking number to each variable, we cannot trust the ranking order of variables who have the same ranking number. This was the reason why we chose to run two wrappers to select our final list of variables.
- Backward selection is a stepwise selection method that starts with building a single model using all variables and then removes the least important variable at each step to return the final subset of variables that gives the best performance. The RFECV here started with all 170 variables and cut one in each iteration and repeated this process for several times.
- Logistic regression and decision tree are the two models we chose to build the wrappers with. When constructing a wrapper, we typically use a fast and simple model. Therefore, we chose a linear model first (logistic regression) and then a simple nonlinear model (decision tree). Since this section focuses on the feature selection process, for explanations of these two model algorithms, please refer to the next section.

A. Wrapper 1 – Logistic Regression Backward Selection

Before feeding the data into the first wrapper, we first standardized all the 170 candidate variables so that they are on the same scale. We then ran the logistic regression wrapper with a CV of 2 over the standardized data. We did also add an L2 regularization term when building this wrapper, and we chose to score the variables by their Receiver Operating Characteristic Area Under Curve (ROC AUC) values, which is also a common measure of goodness for binary classification problems in addition to KS.

The wrapper returned a sorted list of all candidate variables with their ranking numbers. Among the 170 variables, there were 135 variables with a rank of 1. As mentioned before, we could not tell the actual importance ranking order of the 135 variables with the same ranking number, meaning that we were unable to determine which 30 variables were more important than the other 105 variables since they all had a rank of 1. Therefore, we decided to run a second wrapper on these variables to further narrow them down.

B. Wrapper 2 – Decision Tree Backward Selection

The decision tree wrapper was run over the 135 variables resulting from the first wrapper. When running the second wrapper, we used the original unstandardized data since a decision tree model is not sensitive to the feature scale. To be consistent, we also set the CV to be 2 and the scoring method to be ROC AUC for this wrapper. The resulting sorted list of candidate variables only contains four variables with the same rank of 1 this time. Therefore, we could easily determine the top 30 variables among all the variables from this ranking.

Please refer to the following list for our final 30 variables with their descriptions.

Final 30 Variables

RFECV Rank	Variable Name	Description
1	address_count_30	The number of records with the same address over the past 30 days
1	fulladdress_day_since	The number of days since the full address was last seen
1	homephone_count_3	The number of records with the same home phone number over the past 3 days
1	ssn_dob_day_since	The number of days since the SSN and date of birth combination was last seen
2	#_unique_name_for_fulladdress_30	The number of unique names seen for the same full address over the past 30 days
3	address_day_since	The number of days since the address was last seen
4	fulladdress_homephone_day_since	The number of days since the full address and home phone number combination was last seen
5	ssn_firstname_day_since	The number of days since the SSN and first name combination was last seen
6	homephone_count_7	The number of records with the same home phone number over the past 7 days
7	ssn_day_since	The number of days since the SSN was last seen
8	name_dob_day_since	The number of days since the name and date of birth combination was last seen
9	ssn_name_day_since	The number of days since the SSN and name combination was last seen
10	name_count_14	The number of records with the same name over the past 14 days
11	ssn_name_dob_count_30	The number of records with the same SSN, name, and date of birth combination over the past 30 days
12	ssn_lastname_day_since	The number of days since the SSN and last name combination was last seen
13	ssn_name_dob_day_since	The number of days since the SSN, name, and date of birth combination was last seen
14	name_count_7	The number of records with the same name over the past 7 days
15	ssn_dob_count_7	The number of records with the same SSN and date of birth combination over the past 7 days
16	address_count_1_by_14	The ratio of the number of records with the same address over the past 1 day to the number of records with the same address over the past 14 days
17	#_unique_dob_for_address_7	The number of unique dates of birth seen for the same address over the past 7 days

18	#_unique_name_for_address_14	The number of unique names seen for the same address over the past 14 days
19	ssn_count_30	The number of records with the same SSN over the past 30 days
20	#_unique_ssn_lastname_for_address_30	The number of unique SSN and last name combinations seen for the same address over the past 30 days
21	fulladdress_count_0_by_3	The ratio of the number of records with the same full address over the past 0 days to the number of records with the same full address over the past 3 days
22	address_count_0_by_30	The ratio of the number of records with the same address over the past 0 days to the number of records with the same address over the past 30 days
23	address_count_0_by_14	The ratio of the number of records with the same address over the past 0 days to the number of records with the same address over the past 14 days
24	ssn_count_7	The number of records with the same SSN over the past 7 days
25	#_unique_dob_for_address_3	The number of unique dates of birth seen for the same address over the past 3 days
26	name_dob_count_14	The number of records with the same name and date of birth combination over the past 14 days
27	#_unique_name_dob_for_address_30	The number of unique name and date of birth combinations seen for the same address over the past 30 days

VI. Model Algorithms

1. Logistic Regression (Base)

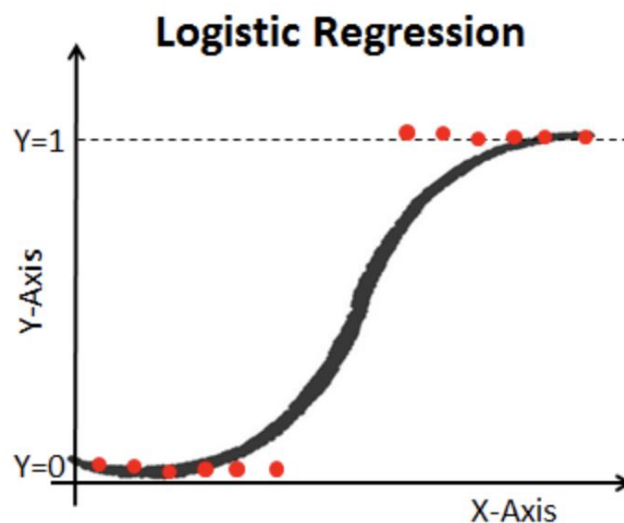
A. Logistic Regression Introduction

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

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

B. Logistic Regression Illustration

How logistic regression works is that it predicts the probability of a record belonging to one class or the other. Although it can be used for multivariable classification, it is a great tool for binary classification problems because it operates on probability. The output value in logistic regression is a numbered classification, but before the classification is given, the ACTUAL output is a numerical value of probability between 0 and 1. Based on the probability, a classification of 1 or 0 will be given. The algorithm essentially rounds the value according to a pre-set threshold to assign class labels, either 0 or 1.



Source: [Logistic Regression](#)

C. Logistic Regression Initial Results

For this project, we ran logistic regression as our baseline model since it is a simple linear model. We used a 5-fold cross validation and adjusted the number of variables used each time. A result table is shown below.

Logistic Regression					Average FDR @ 3%		
Iteration	KFold	Number of Variables	Class Weight	Penalty	TRAINING	TESTING	OOT
1	5	5	Balanced	L2	54.60%	54.50%	51.90%
2	5	10	Balanced	L2	54.90%	54.70%	52.60%
3	5	15	Balanced	L2	55.80%	55.70%	53.30%
4	5	20	Balanced	L2	55.80%	55.60%	53.50%
5	5	25	Balanced	L2	56.40%	56.30%	54.10%
6	5	30	Balanced	L2	56.40%	56.10%	54.00%

2. Decision Tree Classifier

We would like to introduce the decision tree classifier here because it is the base for several of our models that will be introduced next, but we did not build a simple decision tree model as our baseline model because we already chose logistic regression as our baseline model.

A. Decision Tree Introduction

Decision Tree is a non-parametric supervised learning method used for classification and regression. The Decision tree learns from data to approximate a sine curve with a set of if-then-else decision rules. The deeper the tree, the more complex the decision rules and the fitter the model.

The Decision tree builds classification or regression models in the form of a tree structure. It breaks down a data set into smaller and smaller subsets while incrementally developing an associated decision tree.

B. Decision Tree Components

The Decision tree has three main components – nodes, branches, leaf nodes. A decision node tests for the value of a certain attribute and has two or more branches. A branch or edge corresponds to the outcome of a test and connects to the next node of a leaf. A leaf node represents a classification or decision, it is a terminal node with no out-going branch.

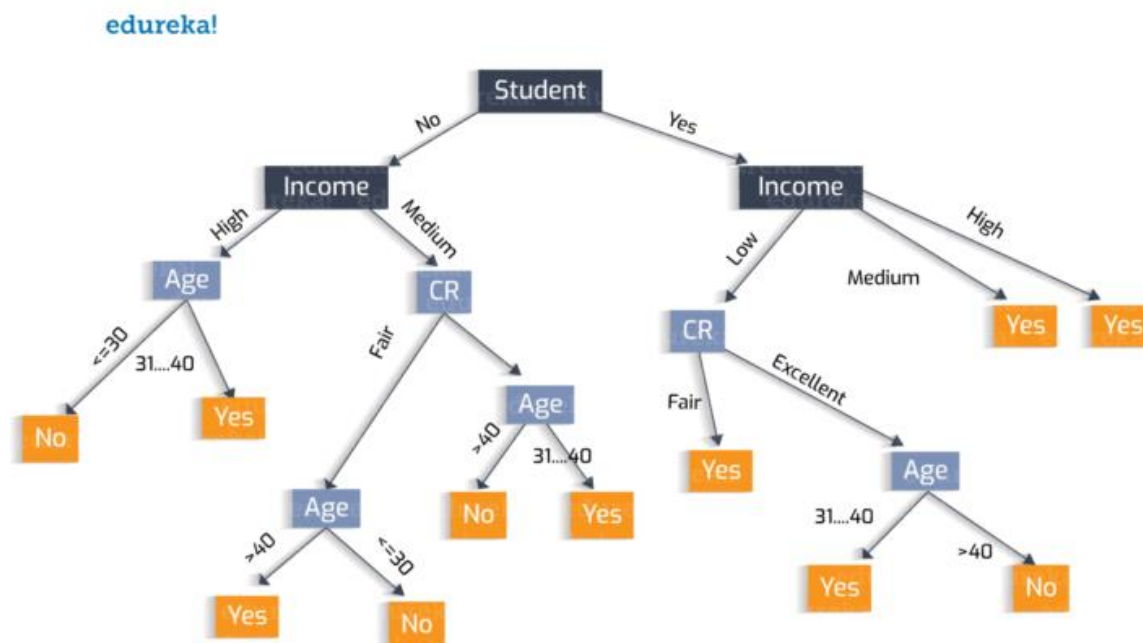
C. Decision Tree Illustration with Example

Consider the following data as an example of a decision tree.

rec	Age	Income	Student	Credit _rating	Buys_computer
r1	<=30	Hight	No	Fair	No
r2	<=30	Hight	No	Excellent	No
r3	31...40	Hight	No	Fair	Yes
r4	>40	Medium	No	Fair	Yes
r5	>40	Low	Yes	Fair	Yes
r6	>40	Low	Yes	Excellent	No
r7	31...40	Low	Yes	Excellent	Yes
r8	<=30	Medium	No	Fair	No
r9	<=30	Low	Yes	Fair	Yes
r10	>30	Medium	Yes	Fair	Yes
r11	<=30	Medium	Yes	Excellent	Yes
r12	31...40	Medium	No	Excellent	Yes
r13	31...40	High	Yes	Fair	Yes
r14	>40	Medium	No	Excellent	No

Source: [Sample Data Table](#)

There are a variety of decision trees that can be built from one sample dataset, each with different choices of variables and/or different splitting values for variables at each node. As shown below, using the attribute “student” as the initial test condition, one can create the following decision tree:



Source: [Sample Decision Tree](#)

D. Decision Tree Classifier

What we have seen above is an example of a classification tree, where the outcome was a categorical variable like “Yes” or “No”. Such a tree is built through a process known as binary recursive partitioning. This is an iterative process of splitting the data into partitions, and then splitting it up further on each of the branches.

3. Random Forest

A. Random Forest Introduction

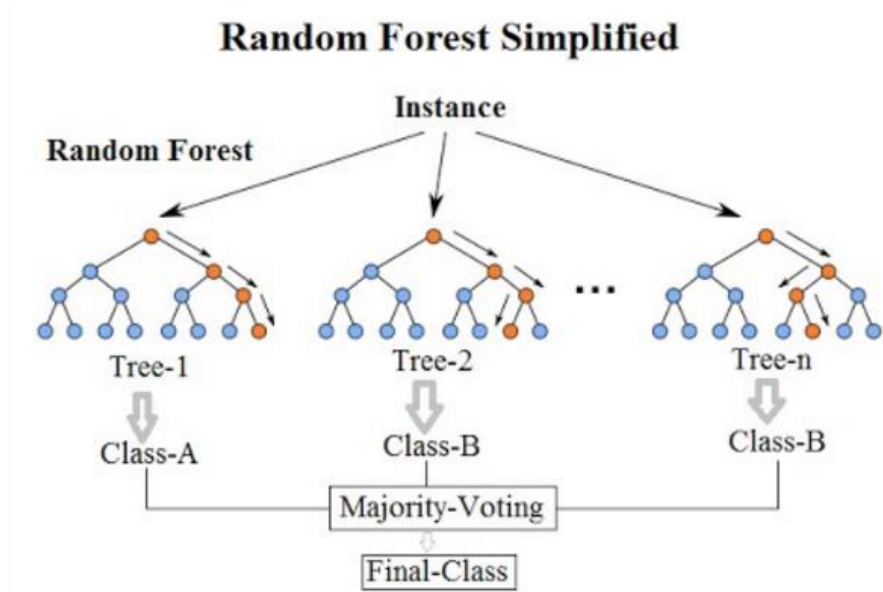
Random forest classifier creates a set of decision trees introduced above from randomly selected subsets of the training set. It then aggregates the votes from different decision trees to decide the final class of the test object.

B. Random Forest Important Components/Hyperparameters

- `n_estimators`: Number of trees in forest. Default is 10.
- `criterion`: “gini” or “entropy” same as decision tree classifier.
- `min_samples_split`: minimum number of working set size at node required to split. Default is 2.
- `min_samples_leaf`: Represents the minimum number of samples required to be at a leaf node.

C. Random Forest Illustration

Random Forests are a more robust option than a single decision tree. It constructs a multitude of decision trees when training the model and outputting the class that is the mode or mean predicted class of the individual trees. A random forest consists of a collection of trees on a random subset of features. Final predictions are the combined results of those trees. Random forests can handle complex data and are not prone to overfitting. They are interpretable by looking at feature importance, and can be adjusted to work well on highly imbalanced data. Their drawback is they're computationally complex. But Random Forest is very popular for fraud detection.



Source: [Random Forest Simplified](#)

D. Random Forest Initial Results

The table below shows the results of our random forest models built for this project.

Random Forest							Average FDR @ 3%		
Iteration	n_estimators	min_samples_split	min_samples_leaf	max_features	criterion	bootstrap	TRAINING	TESTING	OOT
1	200	15	1	sqrt	gini	TRUE	60.96%	55.94%	54.82%
2	15	2	3	5	gini	TRUE	60.90%	56.00%	53.90%
3	15	2	3	5	gini	TRUE	61.00%	56.10%	53.80%
4	15	2	3	10	gini	TRUE	61.10%	55.80%	53.50%
5	30	2	3	5	gini	TRUE	60.60%	56.30%	53.80%

4. Gradient Boosting

A. Gradient Boosting Introduction

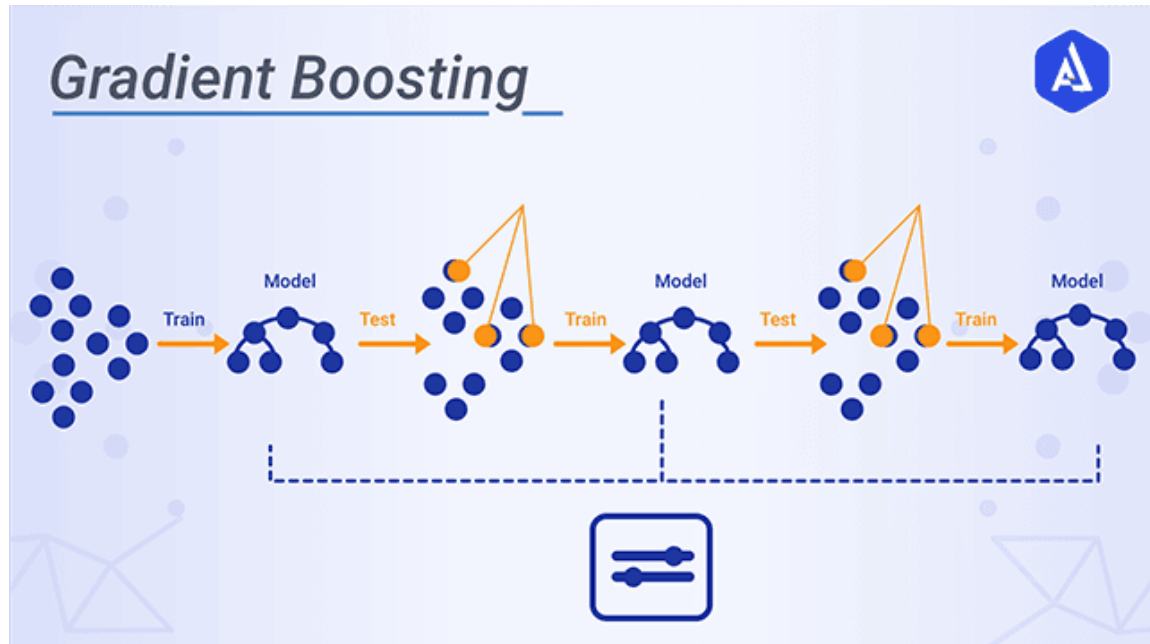
Gradient boosting is a machine learning technique for regression and classification problems, which produces a prediction model in the form of an ensemble of weak prediction models, typically decision trees.

B. Gradient Boosting Classifier

Gradient Boosting Classifier depends on a loss function that maps decisions to their associated costs. It builds the model in a stage-wise fashion using boosting methods, an ensemble meta-algorithm for primarily reducing bias, variance in supervised learning, and a family of machine learning algorithms that convert weak learners into strong ones.

C. Gradient Boosting Illustration

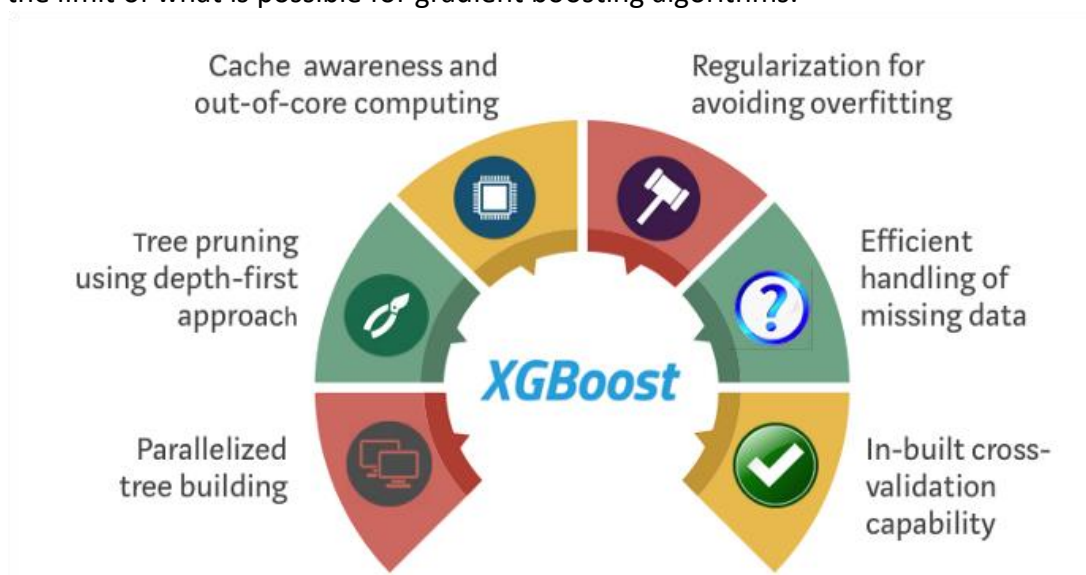
As shown in the image below, gradient boosting executes an iterative process building models and minimizing loss functions until some ideal threshold is reached. Then, all those weak models are combined to make a prediction.



Source: [Gradient Boosting Process](#)

D. XGBoost

XGBoost is a refined and customized version of a gradient boosting decision tree system, created with performance and speed in mind. XGBoost actually stands for “eXtreme Gradient Boosting”, and it refers to the fact that the algorithms and methods have been customized to push the limit of what is possible for gradient boosting algorithms.



Source: [How XGBoost optimizes standard GBM algorithm](#)

E. Gradient Boosting Important Components/Hyperparameters

- **n_estimators**: The number of boosting stages to perform. Gradient boosting is fairly robust to overfitting so a large number usually results in better performance.
- **learning_rate**: Learning rate shrinks the contribution of each tree by learning_rate. There is a trade-off between learning_rate and n_estimators.
- **max_depth**: The maximum depth of the individual regression estimators. The maximum depth limits the number of nodes in the tree.
- **max_feature**: The number of features to consider when looking for the best split.
- **min_samples_leaf**: The minimum number of samples required to be at a leaf node.
- **min_samples_split**: The minimum number of samples required to split an internal node.

F. Gradient Boosting Initial Results

The table below shows the results of our gradient boosting models built for this project.

Gradient Boosting Tree								Average FDR @ 3%		
Iteration	KFold	n_estimators	learning_rate	max_depth	max_feature	min_samples_leaf	min_samples_split	TRAINING	TESTING	OOT
1	5	200	0.01	3	5	30	500	55.17%	55.02%	52.31%
2	5	500	0.01	3	5	30	500	55.70%	55.59%	53.06%
3	5	1000	0.01	3	5	30	500	56.71%	56.36%	54.22%
4	5	1000	0.01	3	5	30	1500	56.61%	56.34%	54.12%
5	5	800	0.01	5	5	30	500	57.51%	57.13%	55.16%
6	5	800	0.01	5	5	30	1500	57.26%	56.75%	55.00%
7	5	1000	0.02	5	5	30	500	57.71%	57.31%	55.44%
8	5	1000	0.01	5	5	30	1500	57.42%	56.94%	55.15%
9	10	1000	0.05	5	5	30	500	57.93%	57.38%	55.61%
10	10	1000	0.05	5	5	30	1500	57.81%	57.41%	55.57%

G. XGBoosting Model Initial Results

The table below shows the results of our XGBoosting models built for this project.

XGBoosting						Average FDR @ 3%		
Iteration	KFold	learning_rate	n_estimators	max_depth	Objective	TRAINING	TESTING	OOT
1	10	0.02	1000	5	binary: logistic	58.08%	57.40%	55.54%
2	10	0.05	1000	5	binary: logistic	58.60%	57.33%	55.55%
3	10	0.02	1500	5	binary: logistic	58.26%	57.36%	55.57%
4	10	0.05	1500	5	binary: logistic	58.94%	57.27%	55.47%
5	10	0.02	1800	5	binary: logistic	58.39%	57.36%	55.58%
6	10	0.05	1800	5	binary: logistic	59.06%	57.17%	55.44%

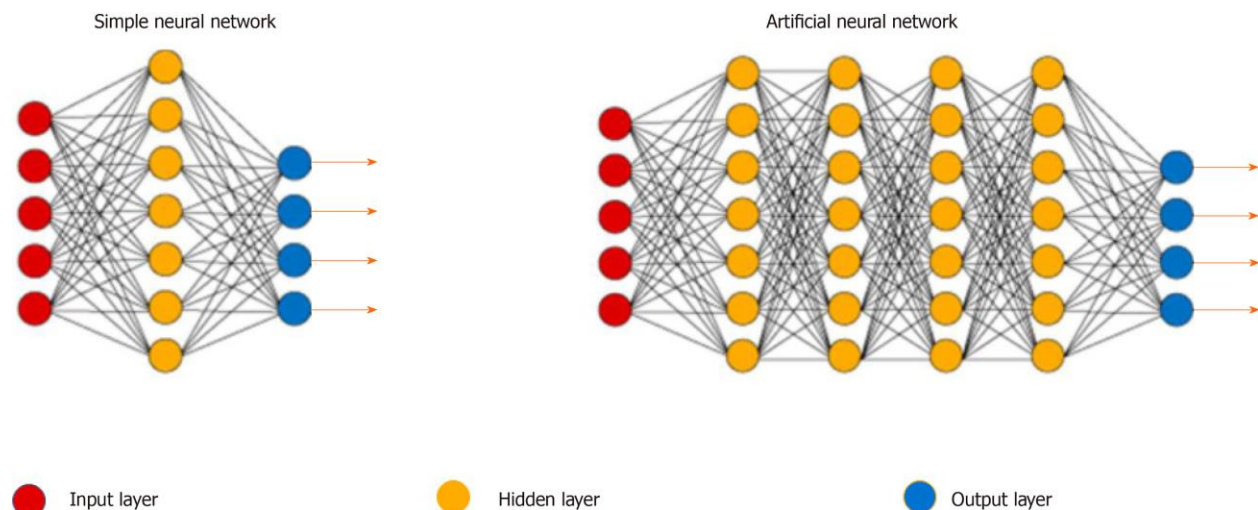
5. Neural Network

A. Neural Network Model Introduction

Neural Net is one of the machine learning algorithms that mimics the human brain, based on a collection of connected units or nodes called artificial neurons, which loosely model the neurons in a biological brain. The algorithm adjusts connections strength from nodes in one layer to nodes in next layers in a similar way to how human beings learn.

The structure of one Neural Net depends on the choice of nodes and layers. Neural Net can have one or multiple layers with one or more nodes at each layer. When adding more layers, Neural Network becomes more complicated and can conduct deep learning from huge, complex data.

How does the Neural Network algorithm work?



Source: [Neural Network Algorithm Examples](#)

As shown above, **at the input layer**, each record will be loaded into the model with random selected parameters. For every feature of one record, a randomly chosen weight will be passed to that inputted feature. Then, an error rate of the weight assigned for that feature will be computed. **At hidden layers**, each node in the hidden layer receives weighted signals from all the nodes in the previous layer and does a transform on this linear combination of signals. After passing all hidden layers, an activation function will transform the final values from the last hidden layer to fit the output, such as logistic or sigmoid functions. Even though weights are randomly selected at the beginning of building models, the algorithm will gradually improve by finding the best weight by “learning” from errors of previous batches of records with the only goal of minimizing the overall error rate.

B. PyTorch Advantages

There are a few neural network algorithms as machine learning methods, including Tensorflow, PyTorch, and MLPClassifier. As compared to other Neural Net, PyTorch neural net:

- Allows for dynamic computational graphs, meaning that network architectures can be changed during running time;
- And, it is easier to use and implement.

C. PyTorch Important Components/Hyperparameters

- Learning rate: the rate of change that the model responds the estimated error each time the model weights are updated
- Batch size: number of samples in each training set processed before the model is updated
- Epoch: number of cycles through the full training dataset
- layer_ n : layer structure for n^{th} layer

D. PyTorch Neural Net Model Initial Results

The table below shows the results of our PyTorch Neural Net models built for this project.

PyTorch Neural Net							Average FDR @ 3%		
Iteration	KFold	Learning Rate	Epochs	Batch Size	Layer_1	Layer_2	TRAINING	TESTING	OOT
1	5	0.0001	15	64	128	128	62.20%	60.19%	55.41%
2	5	0.0001	15	64	32	32	58.81%	57.71%	54.78%
3	5	0.0001	15	64	32	64	57.75%	57.50%	54.57%
4	5	0.0001	15	64	64	32	58.12%	57.45%	54.61%
5	5	0.0001	15	64	64	64	58.67%	58.16%	55.20%

E. Multilayer Perceptron (MLP) Classifier

MLP is another neural network algorithm in addition to PyTorch introduced above. It can be thought of as a deep artificial neural network. It is composed of more than one perceptron. They are composed of an input layer to receive the signal, an output layer that makes a decision or prediction about the input, and in between those two, an arbitrary number of hidden layers that are the true computational engine of the MLP.

F. MLPClassifier Important Components/Hyperparameters

- hidden_layer_size: The i^{th} element represents the number of neurons in the i^{th} hidden layer.
- activation: Activation function for the hidden layer.
- learning_rate: Learning rate schedule for weight updates.
- max_iter: Maximum number of iterations. The solver iterates until convergence (determined by "tol") or this number of iterations.

G. MLPClassifier Model Initial Results

The table below shows the results of our MLP models built for this project.

MLP Neural Network												Average FDR @ 3%		
Iteration	KFold	Layer	Nodes	Max_iter	Activation	Optimizer	Alpha	Learning_rate	Learning_rate_init	Momentum	Nesterovs_momentum	TRAINING	TESTING	OOT
1	10	1	5	100	relu	adam	0.0001	constant	0.001	N/A	N/A	57.04%	56.89%	54.83%
2	10	1	10	200	relu	adam	0.0001	constant	0.001	N/A	N/A	57.30%	57.08%	55.13%
3	10	1	10	200	logistic	adam	0.001	adaptive	0.001	N/A	N/A	57.53%	57.34%	55.37%
4	10	1	15	200	relu	sgd	0.0001	constant	0.001	0.9	TRUE	56.17%	55.89%	53.39%
5	10	1	15	150	logistic	sgd	0.001	adaptive	0.01	0.9	FALSE	56.94%	56.71%	54.39%
6	10	1	20	100	logistic	adam	0.001	constant	0.001	N/A	N/A	57.54%	57.25%	55.37%
7	10	2	5	200	relu	adam	0.0001	constant	0.001	N/A	N/A	57.26%	57.02%	55.03%
8	10	2	10	100	relu	sgd	0.001	adaptive	0.01	0.1	TRUE	56.74%	56.50%	54.18%
9	10	2	5	200	logistic	adam	0.0001	constant	0.001	N/A	N/A	57.25%	57.00%	54.97%
10	10	2	10	100	logistic	sgd	0.001	adaptive	0.01	0.9	TRUE	57.06%	56.87%	54.63%

6. Stacked Ensemble Model

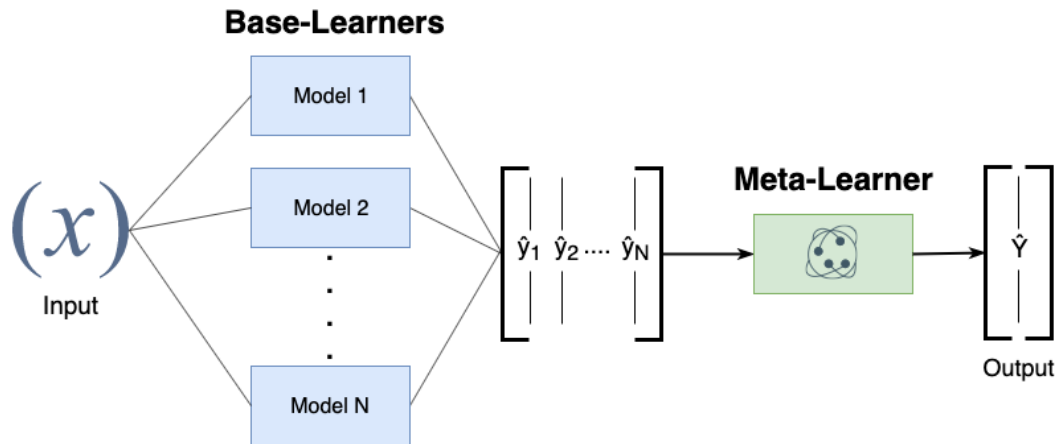
A. Ensemble Methods Introduction

Ensemble methods are commonly used to boost predictive accuracy by combining the predictions of multiple machine learning models. The process can be thought of as building successful human teams. Each team member makes a significant contribution and individual weaknesses and biases are offset by the strengths of other members.

B. Model Stacking

Model stacking is an efficient ensemble method in which the predictions, generated by using various machine learning algorithms, are used as inputs in a second-layer learning algorithm. This second-layer algorithm is trained to optimally combine the model predictions to form a new set of predictions.

As shown below, raw input data are first trained in different models as base-learners. Then, the output of those models are used as input for the meta learner model which produces the final output.



Source: [Sample Stacked Model Training Process](#)

C. Ensemble Stacking Model Initial Result

The table below shows the result of our attempted ensemble stacking model. We used the default hyperparameter settings for several models/estimators at level 0 and used the best hyperparameters resulted from the models built before for MLP, random forest, and gradient boosting tree. The resulting FDR, however, did not exceed the results from some of our individual models introduced before.

Stacking Model							Average FDR @ 3%		
Estimators (Level 0)					Final Estimator (Level 1)	Stack Method	TRAINING	TESTING	OOT
Logistic Regression	MLP Classifier		Random Forest Classifier		Gradient Boosting Tree	Predict Probability	58.54%	57.56%	55.36%
Default	Layer	1	n_estimators	200	n_estimators	1000			
	Nodes	10	min_samples	15	learning_rate	0.05			
	Max_iter	200	_split	1	max_depth	5			
	Activation	logistic	min_samples	1	max_feature	5			
	Optimizer	adam	_leaf	sqrt	min_samples	30			
	Alpha	0.001	max_features	gini	_leaf	500			
	Learning_rate	adaptive	criterion	TRUE	min_samples	_split			
	Learning_rate_init	0.001	bootstrap	TRUE	min_samples	_split			
KNeighbors Classifier	Decision Tree Classifier		Gaussian Naive Bayes						
Default	Default		Default						

Summary of Individual Model Results

Below is a summary table of all the individual model results, split onto two pages.

	Model		Parameters						Average FDR @ 3%			
Logistic Regression	Iteration	KFold	Number of Variables				Class Weight	Penalty	TRAINING	TESTING	OOT	
	1	5	5				Balanced	L2	54.60%	54.50%	51.90%	
	2	5	10				Balanced	L2	54.90%	54.70%	52.60%	
	3	5	15				Balanced	L2	55.80%	55.70%	53.30%	
	4	5	20				Balanced	L2	55.80%	55.60%	53.50%	
	5	5	25				Balanced	L2	56.40%	56.30%	54.10%	
	6	5	30				Balanced	L2	56.40%	56.10%	54.00%	
Random Forest	Iteration		n_estimators	min_samples_split	min_samples_leaf	max_features	criterion	bootstrap	TRAINING	TESING	OOT	
	1		200	15	1	sqrt	gini	TRUE	60.96%	55.94%	54.82%	
	2		15	2	3	5	gini	TRUE	60.90%	56.00%	53.90%	
	3		15	2	3	5	gini	TRUE	61.00%	56.10%	53.80%	
	4		15	2	3	10	gini	TRUE	61.10%	55.80%	53.50%	
	5		30	2	3	5	gini	TRUE	60.60%	56.30%	53.80%	
Gradient Boosting Tree	Iteration	KFold	n_estimators	learning_rate	max_depth	max_feature	min_samples_leaf	min_samples_split	TRAINING	TESTING	OOT	
	1	5	200	0.01	3	5	30	500	55.17%	55.02%	52.31%	
	2	5	500	0.01	3	5	30	500	55.70%	55.59%	53.06%	
	3	5	1000	0.01	3	5	30	500	56.71%	56.36%	54.22%	
	4	5	1000	0.01	3	5	30	1500	56.61%	56.34%	54.12%	
	5	5	800	0.01	5	5	30	500	57.51%	57.13%	55.16%	
	6	5	800	0.01	5	5	30	1500	57.26%	56.75%	55.00%	
	7	5	1000	0.02	5	5	30	500	57.71%	57.31%	55.44%	
	8	5	1000	0.01	5	5	30	1500	57.42%	56.94%	55.15%	
	9	10	1000	0.05	5	5	30	500	57.93%	57.38%	55.61%	
10	10	1000	0.05	5	5	30	1500	57.81%	57.41%	55.57%		
XGBoost	Iteration	KFold	n_estimators		learning_rate		max_depth		Objective	TRAINING	TESTING	OOT
	1	10	1000		0.02		5		binary: logistic	58.08%	57.40%	55.54%
	2	10	1000		0.05		5		binary: logistic	58.60%	57.33%	55.55%
	3	10	1500		0.02		5		binary: logistic	58.26%	57.36%	55.57%
	4	10	1500		0.05		5		binary: logistic	58.94%	57.27%	55.47%
	5	10	1800		0.02		5		binary: logistic	58.39%	57.36%	55.58%
	6	10	1800		0.05		5		binary: logistic	59.06%	57.17%	55.44%

PyTorch Neural Net	Iteration	KFold	Learning Rate		Epochs		Batch Size		Layer_1		Layer_2		TRAINING	TESTING	OOT
	1	5	0.0001		15		64		128		128		62.20%	60.19%	55.41%
	2	5	0.0001		15		64		32		32		58.81%	57.71%	54.78%
	3	5	0.0001		15		64		32		64		57.75%	57.50%	54.57%
	4	5	0.0001		15		64		64		32		58.12%	57.45%	54.61%
	5	5	0.0001		15		64		64		64		58.67%	58.16%	55.20%
MLP Neural Net	Iteration	KFold	Layer	Nodes	Max_iter	Activation	Optimizer	Alpha	Learning_rate	Learning_rate_init	Momentum	Nesterovs_momentum	TRAINING	TESTING	OOT
	1	10	1	5	100	relu	adam	0.0001	constant	0.001	N/A	N/A	57.04%	56.89%	54.83%
	2	10	1	10	200	relu	adam	0.0001	constant	0.001	N/A	N/A	57.30%	57.08%	55.13%
	3	10	1	10	200	logistic	adam	0.001	adaptive	0.001	N/A	N/A	57.53%	57.34%	55.37%
	4	10	1	15	200	relu	sgd	0.0001	constant	0.001	0.9	TRUE	56.17%	55.89%	53.39%
	5	10	1	15	150	logistic	sgd	0.001	adaptive	0.01	0.9	FALSE	56.94%	56.71%	54.39%
	6	10	1	20	100	logistic	adam	0.001	constant	0.001	N/A	N/A	57.54%	57.25%	55.37%
	7	10	2	5	200	relu	adam	0.0001	constant	0.001	N/A	N/A	57.26%	57.02%	55.03%
	8	10	2	10	100	relu	sgd	0.001	adaptive	0.01	0.1	TRUE	56.74%	56.50%	54.18%
	9	10	2	5	200	logistic	adam	0.0001	constant	0.001	N/A	N/A	57.25%	57.00%	54.97%
	10	10	2	10	100	logistic	sgd	0.001	adaptive	0.01	0.9	TRUE	57.06%	56.87%	54.63%
Stacking Model	Iteration	Estimators (Level 0)								Final Estimator (Level 1)		Stack Method	TRAINING	TESTING	OOT
	1	Logistic Regression	MLP Classifier		Random Forest Classifier		KNeighbors Classifier	Decision Tree Classifier	Gaussian Naive Bayes	Gradient Boosting Tree		Predict Probability	58.54%	57.56%	55.36%
		Default	Parameters in MLP Iteration 3		Parameter in Random Forest Iteration 1		Default	Default	Default	Parameters in GBM Iteration 9					

VII. Results

1. Final Model Selection

Through comparison of initial results from training models using logistic regression, random forest, gradient boosting model (both GBM and XGBoost), neural net (MLP and PyTorch), and stacked ensemble model, we determined that the gradient boosting model performed the best.

Quantitatively, the **gradient boosting model** outperforms all other models and has an average fraud detection rate at 3 % above 57.5% on training data, 57% on testing dataset and 55.4% on OOT validation data.

Iteration	n_estimators	learning_rate	max_depth	max_feature	min_samples_leaf	min_samples_split	TRAINING	TESTING	OOT
44	800	0.05	5	5	30	500	57.91%	57.29%	55.49%
46	1000	0.05	5	5	30	500	58.01%	57.35%	55.47%
47	1000	0.05	5	5	30	1500	57.85%	57.28%	55.47%
42	500	0.05	5	5	30	500	57.78%	57.25%	55.45%
30	1000	0.02	5	5	30	500	57.71%	57.31%	55.44%
28	800	0.02	5	5	30	500	57.73%	57.23%	55.39%
29	800	0.02	5	5	30	1500	57.60%	57.18%	55.39%
27	500	0.02	5	5	30	1500	57.49%	56.98%	55.37%
31	1000	0.02	5	5	30	1500	57.64%	57.20%	55.37%
43	500	0.05	5	5	30	1500	57.70%	57.19%	55.37%

2. Final Hyperparameter Selections

After finalizing the model of choice, we proceeded to determine the optimal selection of hyperparameters using exhaustive case testing with KFold and having average fraud detection rate as the evaluation method. The results are shown in the following table:

Iteration	KFold	n_estimators	learning_rate	max_depth	max_feature	min_samples_leaf	min_samples_split	TRAINING	TESTING	OOT
0	10	1000	0.02	5	5	30	500	57.71%	57.34%	55.47%
1	10	1000	0.02	5	5	30	1500	57.68%	57.32%	55.54%
2	10	1500	0.02	5	5	30	500	57.79%	57.33%	55.50%
3	10	1500	0.02	5	5	30	1500	57.71%	57.30%	55.49%
4	10	1800	0.02	5	5	30	500	57.84%	57.41%	55.52%
5	10	1800	0.02	5	5	30	1500	57.75%	57.35%	55.47%
6	10	1000	0.05	5	5	30	500	57.93%	57.38%	55.61%
7	10	1000	0.05	5	5	30	1500	57.81%	57.41%	55.57%
8	10	1500	0.05	5	5	30	500	58.14%	57.32%	55.52%
9	10	1500	0.05	5	5	30	1500	57.94%	57.37%	55.49%
10	10	1800	0.05	5	5	30	500	58.23%	57.39%	55.56%
11	10	1800	0.05	5	5	30	1500	58.00%	57.39%	55.52%

From the final case testing, the best set of parameters has higher fraud detection rate on all train, test and validation data than results obtained in the previous tuning. Therefore, we decided to use the following combination of hyperparameters as our final choice:

n_estimators	learning_rate	max_depth	max_feature	min_samples_leaf	min_samples_split
1000	0.05	5	5	30	500

In the end, we resampled the whole modeling dataset into 75% training and 25% testing and returned the final result. Please refer to the following three pages for the top 20% of the final result tables. For the full final result tables, please refer to Appendix C.

Training Result (Top 20%)

Training	# Records	# Goods	# Bads	Fraud Rate								
	596247	587635	8612	0.014443679								
Bin Statistics						Cumulative Statistics						
Population Bin %	# Records	# Goods	# Bads	% Goods	% Bads	Total # Records	Cumulativ e Goods	Cumulativ e Bads	Cumulativ e % Goods	Cumulativ e % Bads (FDR)	KS Score	FPR
1	5963	1274	4689	21.37%	78.63%	5963	1274	4689	0.22%	54.45%	54.23	0.27
2	5963	5713	250	95.81%	4.19%	11926	6987	4939	1.19%	57.35%	56.16	1.41
3	5963	5886	77	98.71%	1.29%	17889	12873	5016	2.19%	58.24%	56.05	2.57
4	5963	5902	61	98.98%	1.02%	23852	18775	5077	3.20%	58.95%	55.75	3.7
5	5963	5898	65	98.91%	1.09%	29815	24673	5142	4.20%	59.71%	55.51	4.8
6	5963	5917	46	99.23%	0.77%	35778	30590	5188	5.21%	60.24%	55.03	5.9
7	5963	5907	56	99.06%	0.94%	41741	36497	5244	6.21%	60.89%	54.68	6.96
8	5963	5922	41	99.31%	0.69%	47704	42419	5285	7.22%	61.37%	54.15	8.03
9	5963	5913	50	99.16%	0.84%	53667	48332	5335	8.22%	61.95%	53.73	9.06
10	5963	5919	44	99.26%	0.74%	59630	54251	5379	9.23%	62.46%	53.23	10.09
11	5963	5919	44	99.26%	0.74%	65593	60170	5423	10.24%	62.97%	52.73	11.1
12	5963	5916	47	99.21%	0.79%	71556	66086	5470	11.25%	63.52%	52.27	12.08
13	5963	5933	30	99.50%	0.50%	77519	72019	5500	12.26%	63.86%	51.6	13.09
14	5963	5927	36	99.40%	0.60%	83482	77946	5536	13.26%	64.28%	51.02	14.08
15	5963	5926	37	99.38%	0.62%	89445	83872	5573	14.27%	64.71%	50.44	15.05
16	5963	5920	43	99.28%	0.72%	95408	89792	5616	15.28%	65.21%	49.93	15.99
17	5963	5920	43	99.28%	0.72%	101371	95712	5659	16.29%	65.71%	49.42	16.91
18	5963	5930	33	99.45%	0.55%	107334	101642	5692	17.30%	66.09%	48.79	17.86
19	5963	5924	39	99.35%	0.65%	113297	107566	5731	18.30%	66.55%	48.25	18.77
20	5963	5925	38	99.36%	0.64%	119260	113491	5769	19.31%	66.99%	47.68	19.67

Testing Result (Top 20%)

Testing	# Records	# Goods	# Bads	Fraud Rate								
	198749	195875	2874	0.01446045								
Bin Statistics						Cumulative Statistics						
Population Bin %	# Records	# Goods	# Bads	% Goods	% Bads	Total # Records	Cumulative Goods	Cumulative Bads	Cumulative % Goods	Cumulative % Bads (FDR)	KS Score	FPR
1	1988	462	1526	23.24%	76.76%	1988	462	1526	0.24%	53.10%	52.86	0.3
2	1988	1900	88	95.57%	4.43%	3976	2362	1614	1.21%	56.16%	54.95	1.46
3	1988	1967	21	98.94%	1.06%	5964	4329	1635	2.21%	56.89%	54.68	2.65
4	1988	1975	13	99.35%	0.65%	7952	6304	1648	3.22%	57.34%	54.12	3.83
5	1988	1968	20	98.99%	1.01%	9940	8272	1668	4.22%	58.04%	53.82	4.96
6	1988	1974	14	99.30%	0.70%	11928	10246	1682	5.23%	58.52%	53.29	6.09
7	1988	1975	13	99.35%	0.65%	13916	12221	1695	6.24%	58.98%	52.74	7.21
8	1988	1973	15	99.25%	0.75%	15904	14194	1710	7.25%	59.50%	52.25	8.3
9	1988	1966	22	98.89%	1.11%	17892	16160	1732	8.25%	60.26%	52.01	9.33
10	1988	1966	22	98.89%	1.11%	19880	18126	1754	9.25%	61.03%	51.78	10.33
11	1988	1968	20	98.99%	1.01%	21868	20094	1774	10.26%	61.73%	51.47	11.33
12	1988	1981	7	99.65%	0.35%	23856	22075	1781	11.27%	61.97%	50.7	12.39
13	1988	1970	18	99.09%	0.91%	25844	24045	1799	12.28%	62.60%	50.32	13.37
14	1988	1972	16	99.20%	0.80%	27832	26017	1815	13.28%	63.15%	49.87	14.33
15	1988	1976	12	99.40%	0.60%	29820	27993	1827	14.29%	63.57%	49.28	15.32
16	1988	1969	19	99.04%	0.96%	31808	29962	1846	15.30%	64.23%	48.93	16.23
17	1988	1972	16	99.20%	0.80%	33796	31934	1862	16.30%	64.79%	48.49	17.15
18	1988	1976	12	99.40%	0.60%	35784	33910	1874	17.31%	65.21%	47.9	18.09
19	1988	1972	16	99.20%	0.80%	37772	35882	1890	18.32%	65.76%	47.44	18.99
20	1988	1966	22	98.89%	1.11%	39760	37848	1912	19.32%	66.53%	47.21	19.79

OOT Result (Top 20%)

OOT	# Records	# Goods	# Bads	Fraud Rate								
	166493	164107	2386	0.014330933								
Bin Statistics						Cumulative Statistics						
Population Bin %	# Records	# Goods	# Bads	% Goods	% Bads	Total # Records	Cumulative Goods	Cumulative Bads	Cumulative % Goods	Cumulative % Bads (FDR)	KS Score	FPR
1	1665	434	1231	26.07%	73.93%	1665	434	1231	0.26%	51.59%	51.33	0.35
2	1665	1589	76	95.44%	4.56%	3330	2023	1307	1.23%	54.78%	53.55	1.55
3	1665	1646	19	98.86%	1.14%	4995	3669	1326	2.24%	55.57%	53.33	2.77
4	1665	1650	15	99.10%	0.90%	6660	5319	1341	3.24%	56.20%	52.96	3.97
5	1665	1654	11	99.34%	0.66%	8325	6973	1352	4.25%	56.66%	52.41	5.16
6	1665	1660	5	99.70%	0.30%	9990	8633	1357	5.26%	56.87%	51.61	6.36
7	1665	1652	13	99.22%	0.78%	11655	10285	1370	6.27%	57.42%	51.15	7.51
8	1665	1645	20	98.80%	1.20%	13320	11930	1390	7.27%	58.26%	50.99	8.58
9	1665	1658	7	99.58%	0.42%	14985	13588	1397	8.28%	58.55%	50.27	9.73
10	1665	1647	18	98.92%	1.08%	16650	15235	1415	9.28%	59.30%	50.02	10.77
11	1665	1652	13	99.22%	0.78%	18315	16887	1428	10.29%	59.85%	49.56	11.83
12	1665	1656	9	99.46%	0.54%	19980	18543	1437	11.30%	60.23%	48.93	12.9
13	1665	1651	14	99.16%	0.84%	21645	20194	1451	12.31%	60.81%	48.5	13.92
14	1665	1656	9	99.46%	0.54%	23310	21850	1460	13.31%	61.19%	47.88	14.97
15	1665	1656	9	99.46%	0.54%	24975	23506	1469	14.32%	61.57%	47.25	16
16	1665	1648	17	98.98%	1.02%	26640	25154	1486	15.33%	62.28%	46.95	16.93
17	1665	1653	12	99.28%	0.72%	28305	26807	1498	16.34%	62.78%	46.44	17.9
18	1665	1652	13	99.22%	0.78%	29970	28459	1511	17.34%	63.33%	45.99	18.83
19	1665	1652	13	99.22%	0.78%	31635	30111	1524	18.35%	63.87%	45.52	19.76
20	1665	1651	14	99.16%	0.84%	33300	31762	1538	19.35%	64.46%	45.11	20.65

VIII. Conclusions

Throughout the project, we have conducted comprehensive data analysis and modeling on the dataset to predict fraudulent applications. First, we performed data cleaning on the original application dataset. Specifically, we fixed the format of the “date” and “zip5” fields and the frivolous values in the “ssn”, “address”, “dob”, and “homephone” fields. After that, over 800 candidate variables were created using available field combinations. We then applied the feature selection process by ranking candidate variables based on their KS scores and FDR at 3%. According to the average ranking scores, we kept the top 170 variables for the next feature selection step. We used two backward selection wrappers with logistic regression and decision tree models to finalize our top 30 variables to predict frauds. With the best 30 variables, we built several machine learning models including Logistic Regression, Random Forest, Gradient boosting, XGBoosting, Neural Network models, and also attempted the stacked ensemble model. After tuning hyperparameters and calculating the best FDR for each model, our best model turned out to be the gradient boosting model, with the FDR at 3% for the testing set to be 56.9% and for the OOT set to be 55.6%.

If given more time and resources, our team would like to conduct further research and investigations in the following areas:

- 1) We would consult industry and domain experts regarding our process of creating and selecting features. In terms of variable creation, we were wondering if there were additional and nonredundant base entities we could create with the original data fields, so we could build more candidate variables from them. For the candidate variables, we would try to increase varieties based on domain experts’ knowledge and experience, since we only had five different types of variables for this project. As for feature selection, since our process mainly relied on statistics calculations and model results, we were wondering if introducing more human judgements into the process would improve our model performance. For example, we could potentially add expert factors to boost or reduce the effects of certain variables based on domain experts’ knowledge and experience.
- 2) The RFECV used in our feature selection process was suggested to be a popularly used but poor algorithm in terms of how it evaluates and ranks variables. It measures each variable’s individual contribution or importance to the models rather than the model performance; therefore, it does not rank the variables as expected or removes correlations. Therefore, we would explore the “SequentialFeaturSelector” (SFS) algorithm as our new wrapper to run the feature selection in future works.
- 3) We may also add an additional step in the feature selection process using the embedded method. Even though we added a regularization term when training some of our models, we did not actually reduce the number of variables based on those models. Therefore, we may deliberately add this step before building our models to further narrow down the list of the final variables.
- 4) For this project, we manually ran k-fold cross validation with manually selected combinations of hyperparameters for each model built, and for each iteration, we

calculated the FDR. Therefore, we may try to use “GridSearchCV” to automatically tune the hyperparameters by defining and passing in our own scoring function of calculating FDR. In this way, the algorithm can directly try different combinations of hyperparameters exhaustively and find the best one.

- 5) To improve our final model, we tried both model stacking and the SMOTE oversampling method, but their results seemed to be worse than our current final model. Therefore, we would like to further investigate the reasons and potentially resolve the issue and improve upon the current model.

IX. Appendix

Appendix A. Data Quality Report (DQR)

A. Description

The dataset “applications data.csv” contains product (credit cards and cell phones) application data with personal identification information for the purpose of finding and labelling application/identity fraud. It is a synthetic dataset built from studying the statistical properties of over a billion real U.S. applications over about ten years by an identity fraud prevention company. It was built so as to reproduce the important univariate and multivariate field distributions of real data. This dataset covers application data from 2016-01-01 to 2016-12-31 and has 10 fields and 1,000,000 records.

B. Summary Statistics Table

Field Name	Field Type	dtype	# Records	% Populated	# Unique Values	Most Common Field Value	% Most Common Field Value	Minimum Value	Maximum Value
record	categorical	int64	1000000	100	1000000	N/A	N/A	N/A	N/A
date	date	int64	1000000	100	365	20160816	0.29	20160101	20161231
ssn	categorical	int64	1000000	100	835819	999999999	1.69	N/A	N/A
firstname	categorical	object	1000000	100	78136	EAMSTRMT	1.27	N/A	N/A
lastname	categorical	object	1000000	100	177001	ERJSAXA	0.86	N/A	N/A
address	categorical	object	1000000	100	828774	123 MAIN ST	0.11	N/A	N/A
zip5	categorical	int64	1000000	100	26370	68138	0.08	N/A	N/A
dob	date	int64	1000000	100	42673	19070626	12.66	19000101	20161031
homephone	categorical	int64	1000000	100	28244	999999999	7.85	N/A	N/A
fraud_label	categorical	int64	1000000	100	2	0	98.56	N/A	N/A

* Note:

- Field Type: how we treat each field
- dtype: how Python reads each field without any transformation
- N/A: value not applicable for the nature of the field

C. Field Descriptions & Visualizations

Field 1:

Name: “record”

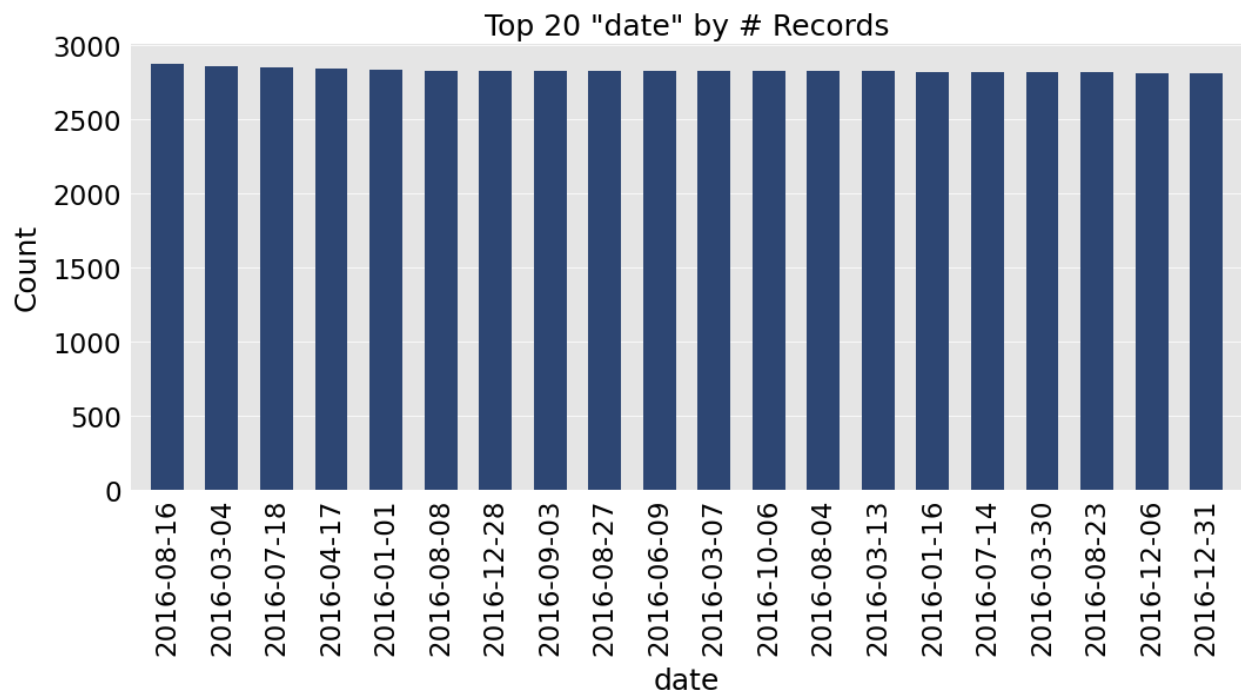
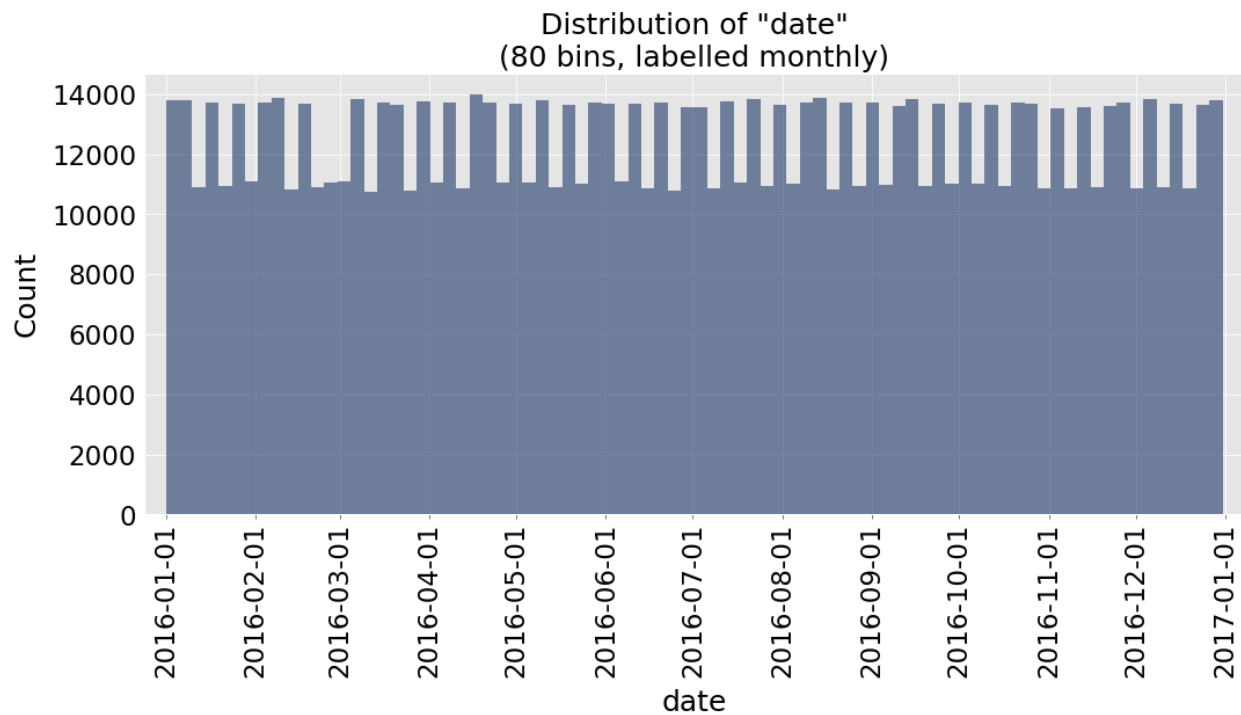
Description: Unique identifier of each entry in the data.

Field 2:

Name: “date”

Description: Date of the application, ranging from 2016-01-01 to 2016-12-31. The first graph shows the distribution of application dates over the entire range. A second graph of “date” is also included below to show the top 20 most frequent application dates. The most frequent

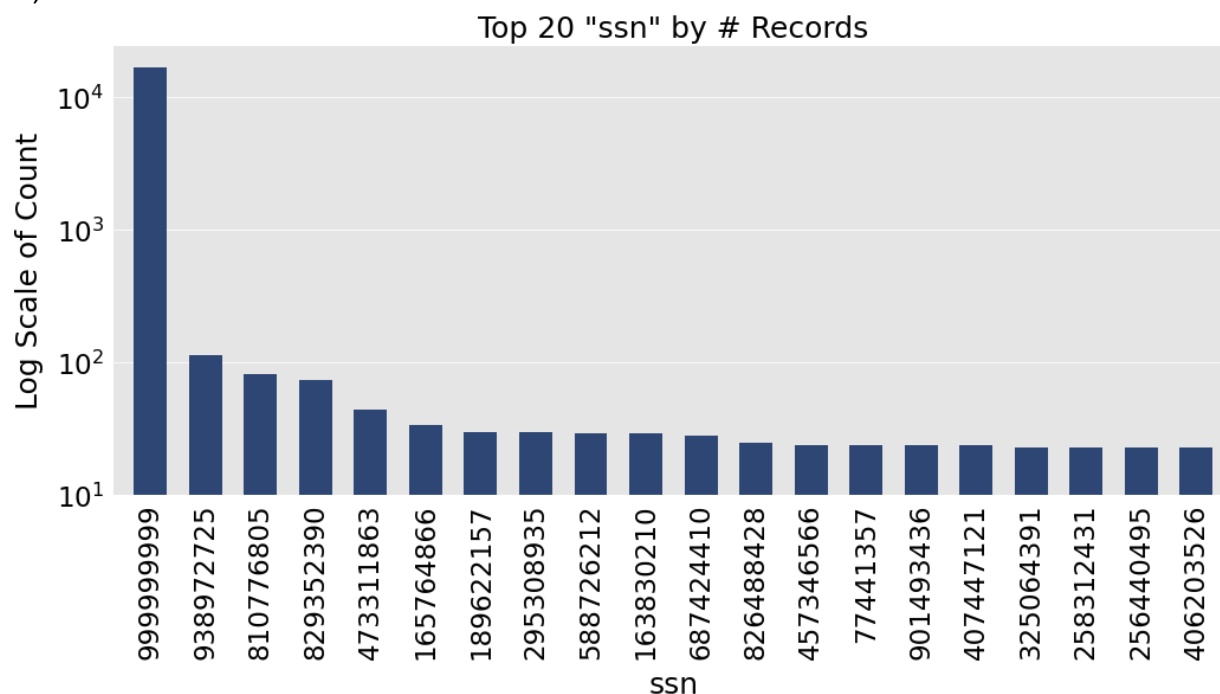
value "2016-08-16" has 2,877 records. The other 19 dates also have very close numbers of records, all above 2,800 records.



Field 3:

Name: "ssn"

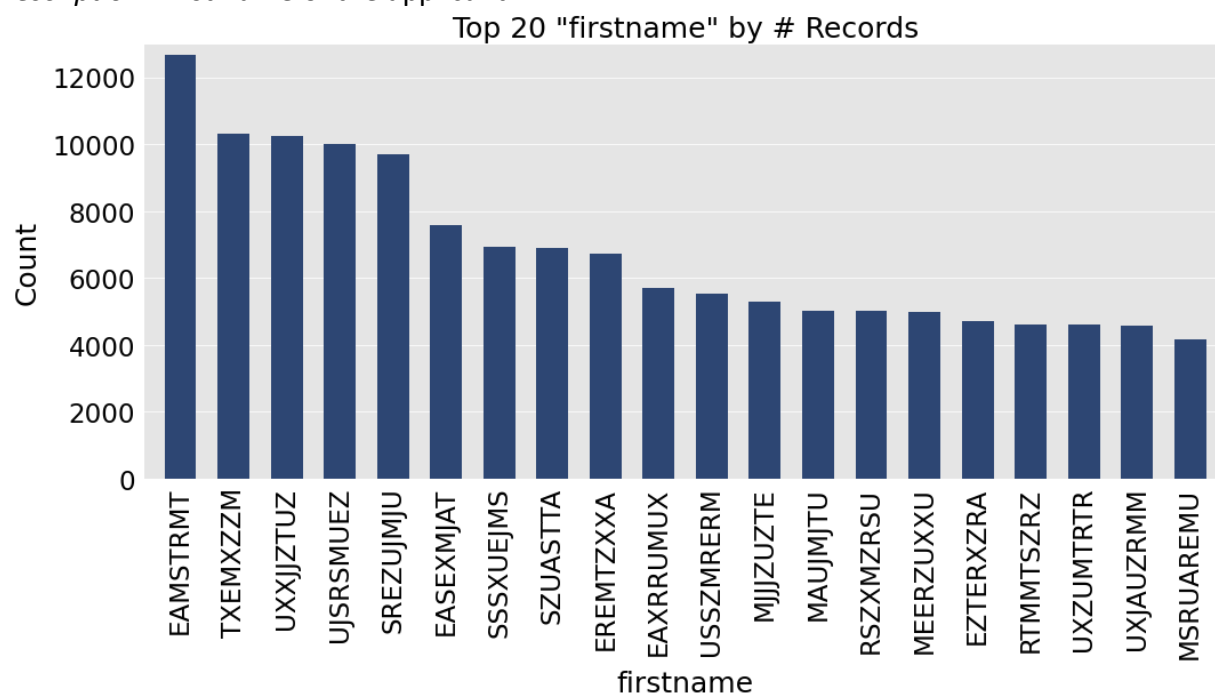
Description: Social Security Number of the applicant. The most frequent value "999999999" has 16,935 records.



Field 4:

Name: "firstname"

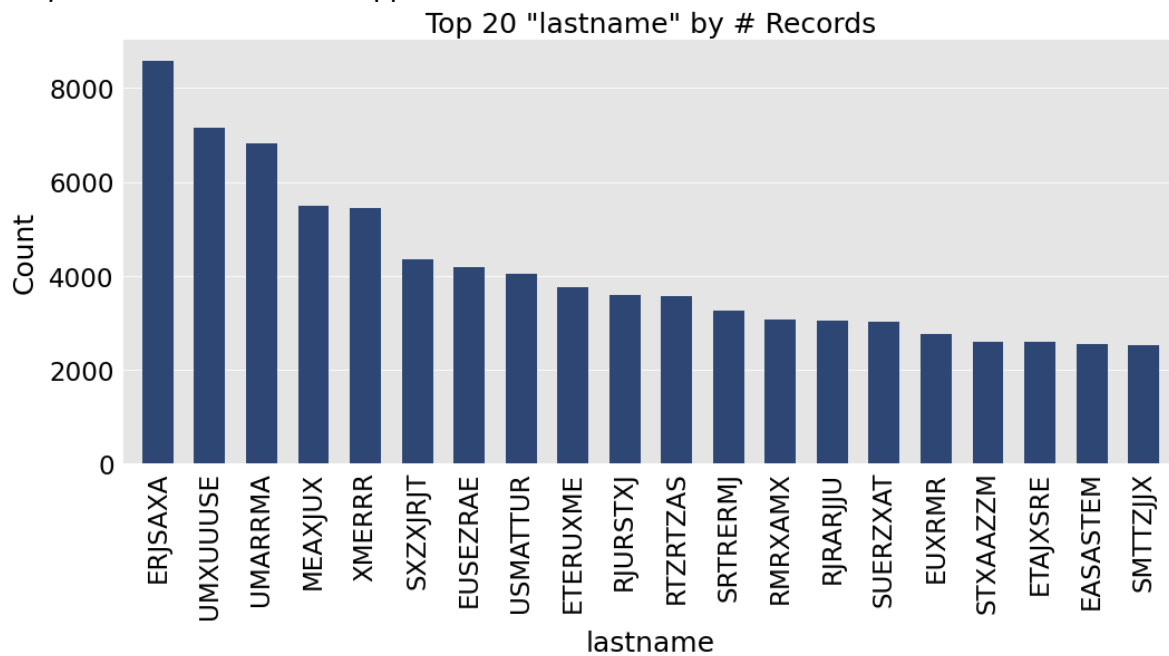
Description: First name of the applicant.



Field 5:

Name: "lastname"

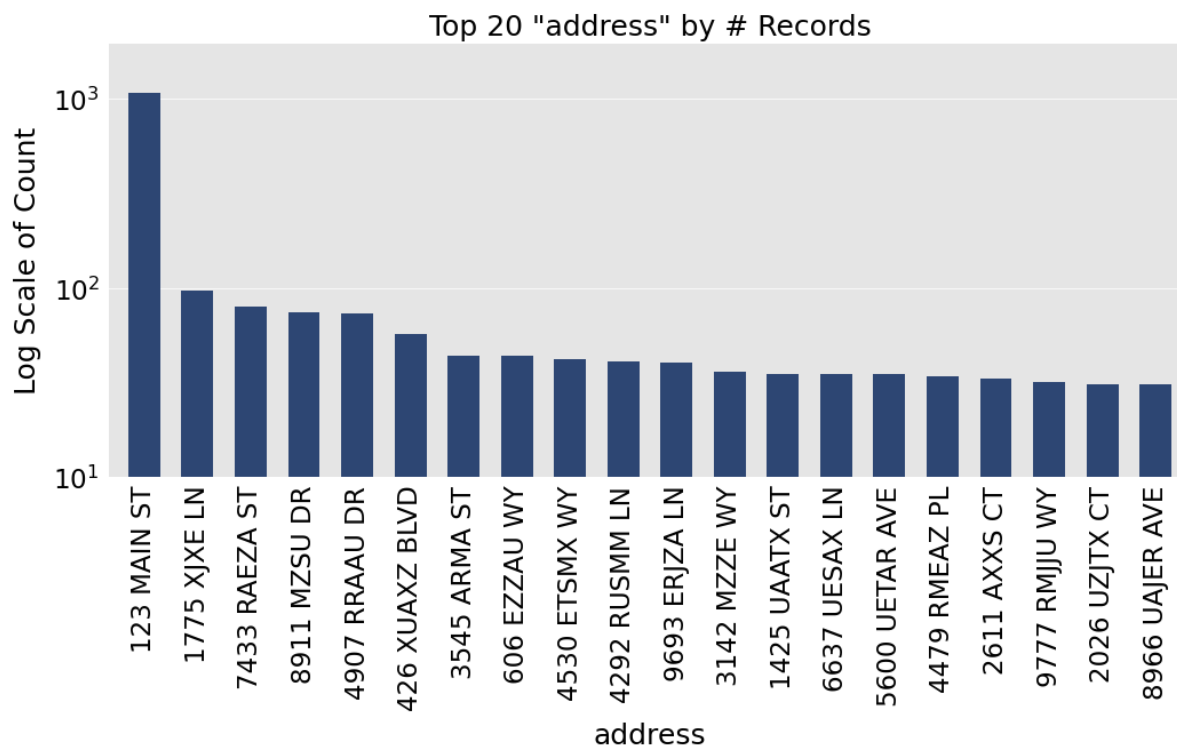
Description: Last name of the applicant.



Field 6:

Name: "address"

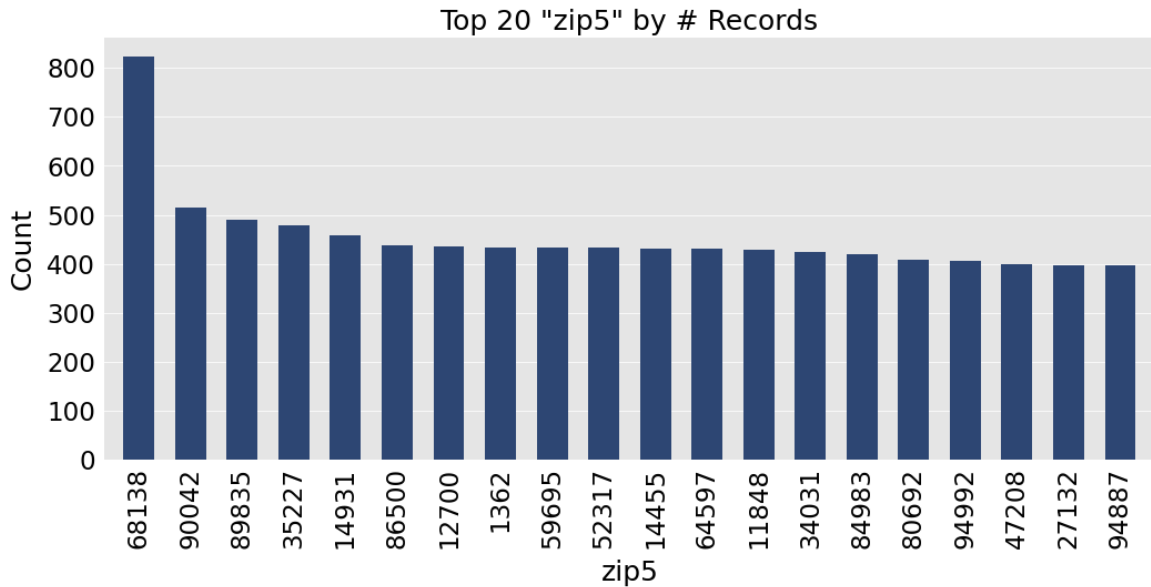
Description: Street address of the applicant. The most frequent value "123 MAIN ST" has 1,079 records.



Field 7:

Name: "zip5"

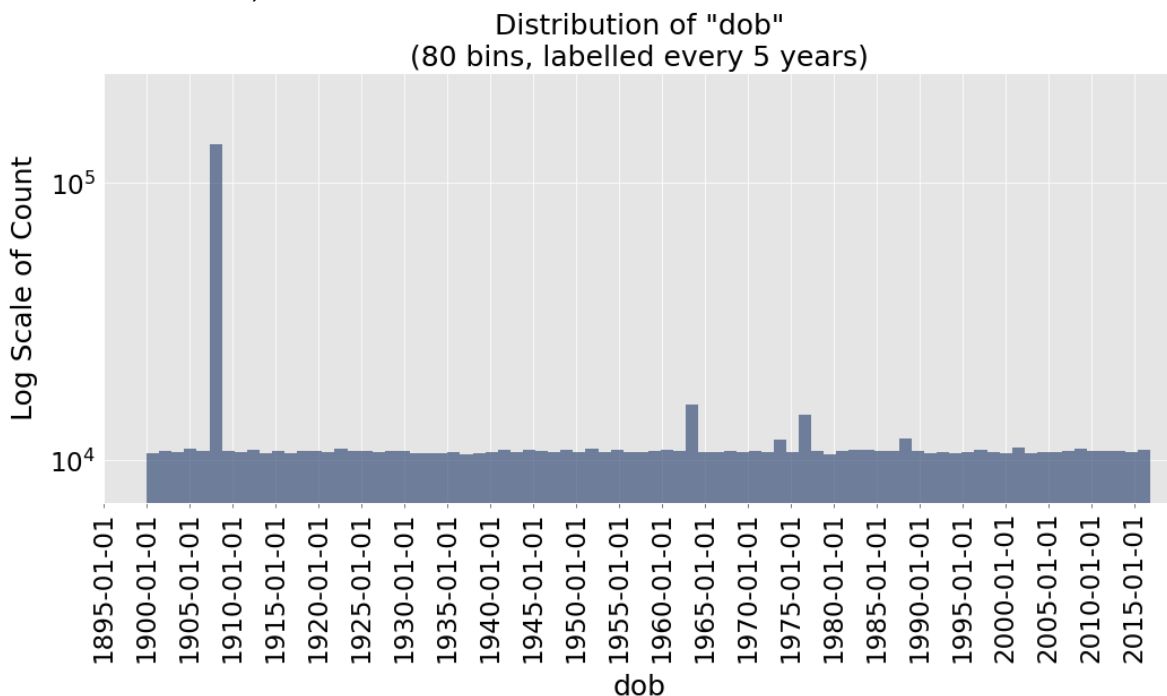
Description: 5-digit zip code of the applicant.

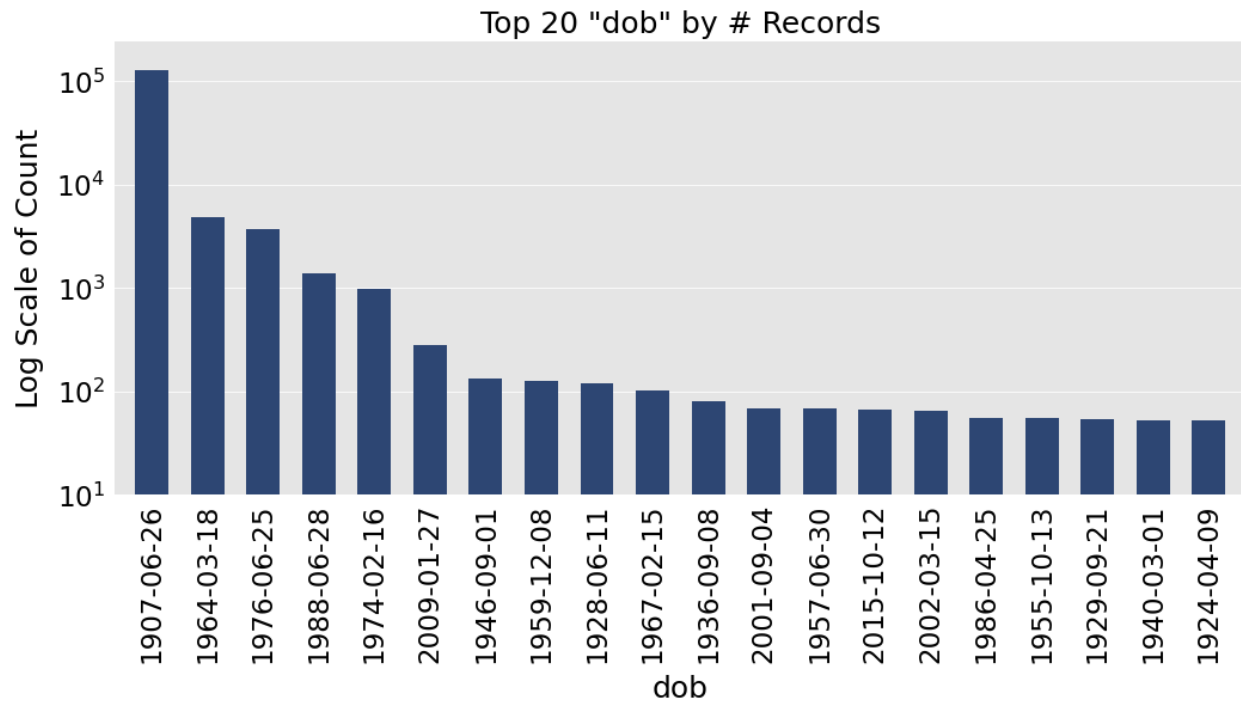


Field 8:

Name: "dob"

Description: Date of birth of the applicant, ranging from 1900-01-01 to 2016-10-31. The first graph shows the distribution of the birthdates over the entire range. A second graph of "dob" is also included below to show the top 20 most frequent dates of birth. The most frequent value "1907-06-26" has 126,568 records.

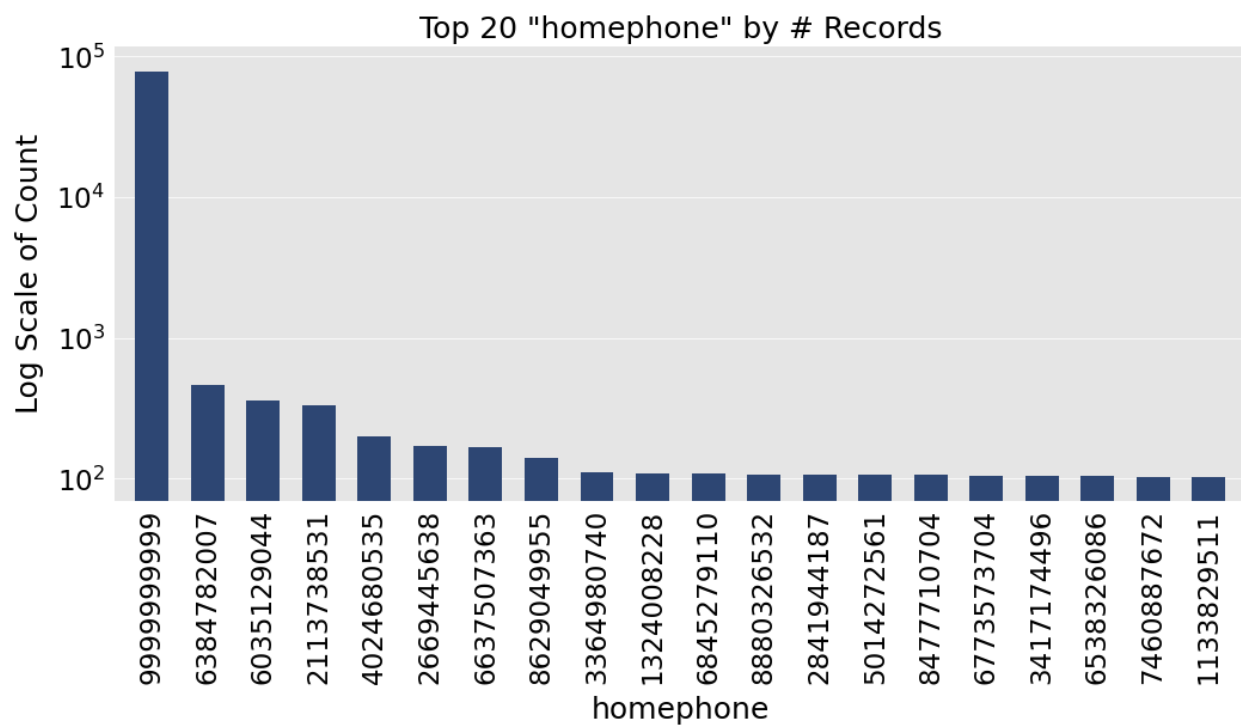




Field 9:

Name: "homephone"

Description: Home phone number of the applicant. The most frequent value "9999999999" has 78,512 records.



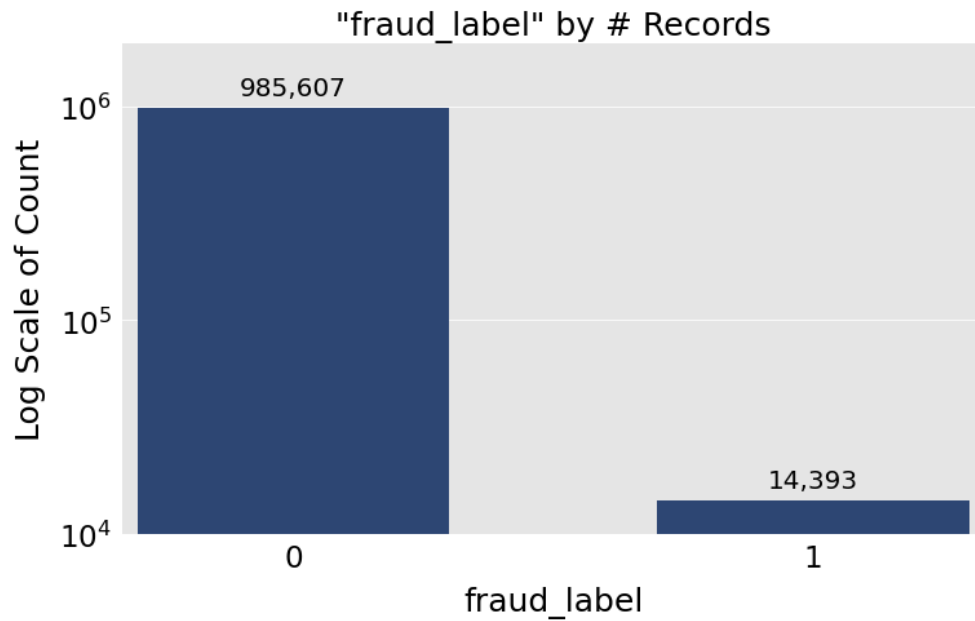
Field 10:

Name: "fraud_label"

Description: Whether the application was a fraud or not.

"1" = Fraudulent application (14,393 records)

"0" = Non-fraudulent application (985,607 records)



Appendix B. Candidate Variables

1. Risk Table Variable (1)

dow_risk

2. Days-Since Variables (22)

Days-Since Variable Name		Days-Since Variable Name	
1	ssn_day_since	12	dob_homephone_day_since
2	address_day_since	13	homephone_name_dob_day_since
3	dob_day_since	14	ssn_firstname_day_since
4	homephone_day_since	15	ssn_lastname_day_since
5	name_day_since	16	ssn_address_day_since
6	fulladdress_day_since	17	ssn_zip5_day_since
7	name_dob_day_since	18	ssn_dob_day_since
8	name_fulladdress_day_since	19	ssn_homephone_day_since
9	name_homephone_day_since	20	ssn_name_day_since
10	fulladdress_dob_day_since	21	ssn_fulladdress_day_since
11	fulladdress_homephone_day_since	22	ssn_name_dob_day_since

3. Velocity Variables (132)

Velocity Variable Name		Velocity Variable Name	
1	ssn_count_0	67	dob_homephone_count_0
2	ssn_count_1	68	dob_homephone_count_1
3	ssn_count_3	69	dob_homephone_count_3
4	ssn_count_7	70	dob_homephone_count_7
5	ssn_count_14	71	dob_homephone_count_14
6	ssn_count_30	72	dob_homephone_count_30
7	address_count_0	73	homephone_name_dob_count_0
8	address_count_1	74	homephone_name_dob_count_1
9	address_count_3	75	homephone_name_dob_count_3
10	address_count_7	76	homephone_name_dob_count_7
11	address_count_14	77	homephone_name_dob_count_14
12	address_count_30	78	homephone_name_dob_count_30
13	dob_count_0	79	ssn_firstname_count_0
14	dob_count_1	80	ssn_firstname_count_1
15	dob_count_3	81	ssn_firstname_count_3
16	dob_count_7	82	ssn_firstname_count_7
17	dob_count_14	83	ssn_firstname_count_14
18	dob_count_30	84	ssn_firstname_count_30
19	homephone_count_0	85	ssn_lastname_count_0

20	homephone_count_1	86	ssn_lastname_count_1
21	homephone_count_3	87	ssn_lastname_count_3
22	homephone_count_7	88	ssn_lastname_count_7
23	homephone_count_14	89	ssn_lastname_count_14
24	homephone_count_30	90	ssn_lastname_count_30
25	name_count_0	91	ssn_address_count_0
26	name_count_1	92	ssn_address_count_1
27	name_count_3	93	ssn_address_count_3
28	name_count_7	94	ssn_address_count_7
29	name_count_14	95	ssn_address_count_14
30	name_count_30	96	ssn_address_count_30
31	fulladdress_count_0	97	ssn_zip5_count_0
32	fulladdress_count_1	98	ssn_zip5_count_1
33	fulladdress_count_3	99	ssn_zip5_count_3
34	fulladdress_count_7	100	ssn_zip5_count_7
35	fulladdress_count_14	101	ssn_zip5_count_14
36	fulladdress_count_30	102	ssn_zip5_count_30
37	name_dob_count_0	103	ssn_dob_count_0
38	name_dob_count_1	104	ssn_dob_count_1
39	name_dob_count_3	105	ssn_dob_count_3
40	name_dob_count_7	106	ssn_dob_count_7
41	name_dob_count_14	107	ssn_dob_count_14
42	name_dob_count_30	108	ssn_dob_count_30
43	name_fulladdress_count_0	109	ssn_homephone_count_0
44	name_fulladdress_count_1	110	ssn_homephone_count_1
45	name_fulladdress_count_3	111	ssn_homephone_count_3
46	name_fulladdress_count_7	112	ssn_homephone_count_7
47	name_fulladdress_count_14	113	ssn_homephone_count_14
48	name_fulladdress_count_30	114	ssn_homephone_count_30
49	name_homephone_count_0	115	ssn_name_count_0
50	name_homephone_count_1	116	ssn_name_count_1
51	name_homephone_count_3	117	ssn_name_count_3
52	name_homephone_count_7	118	ssn_name_count_7
53	name_homephone_count_14	119	ssn_name_count_14
54	name_homephone_count_30	120	ssn_name_count_30
55	fulladdress_dob_count_0	121	ssn_fulladdress_count_0
56	fulladdress_dob_count_1	122	ssn_fulladdress_count_1
57	fulladdress_dob_count_3	123	ssn_fulladdress_count_3
58	fulladdress_dob_count_7	124	ssn_fulladdress_count_7
59	fulladdress_dob_count_14	125	ssn_fulladdress_count_14
60	fulladdress_dob_count_30	126	ssn_fulladdress_count_30
61	fulladdress_homephone_count_0	127	ssn_name_dob_count_0
62	fulladdress_homephone_count_1	128	ssn_name_dob_count_1

63	fulladdress_homephone_count_3	129	ssn_name_dob_count_3
64	fulladdress_homephone_count_7	130	ssn_name_dob_count_7
65	fulladdress_homephone_count_14	131	ssn_name_dob_count_14
66	fulladdress_homephone_count_30	132	ssn_name_dob_count_30

4. Relative Velocity Variables (176)

Relative Velocity Variable Name		Relative Velocity Variable Name	
1	ssn_count_0_by_3	89	dob_homephone_count_0_by_3
2	ssn_count_0_by_7	90	dob_homephone_count_0_by_7
3	ssn_count_0_by_14	91	dob_homephone_count_0_by_14
4	ssn_count_0_by_30	92	dob_homephone_count_0_by_30
5	ssn_count_1_by_3	93	dob_homephone_count_1_by_3
6	ssn_count_1_by_7	94	dob_homephone_count_1_by_7
7	ssn_count_1_by_14	95	dob_homephone_count_1_by_14
8	ssn_count_1_by_30	96	dob_homephone_count_1_by_30
9	address_count_0_by_3	97	homephone_name_dob_count_0_by_3
10	address_count_0_by_7	98	homephone_name_dob_count_0_by_7
11	address_count_0_by_14	99	homephone_name_dob_count_0_by_14
12	address_count_0_by_30	100	homephone_name_dob_count_0_by_30
13	address_count_1_by_3	101	homephone_name_dob_count_1_by_3
14	address_count_1_by_7	102	homephone_name_dob_count_1_by_7
15	address_count_1_by_14	103	homephone_name_dob_count_1_by_14
16	address_count_1_by_30	104	homephone_name_dob_count_1_by_30
17	dob_count_0_by_3	105	ssn_firstname_count_0_by_3
18	dob_count_0_by_7	106	ssn_firstname_count_0_by_7
19	dob_count_0_by_14	107	ssn_firstname_count_0_by_14
20	dob_count_0_by_30	108	ssn_firstname_count_0_by_30
21	dob_count_1_by_3	109	ssn_firstname_count_1_by_3
22	dob_count_1_by_7	110	ssn_firstname_count_1_by_7
23	dob_count_1_by_14	111	ssn_firstname_count_1_by_14
24	dob_count_1_by_30	112	ssn_firstname_count_1_by_30
25	homephone_count_0_by_3	113	ssn_lastname_count_0_by_3
26	homephone_count_0_by_7	114	ssn_lastname_count_0_by_7
27	homephone_count_0_by_14	115	ssn_lastname_count_0_by_14
28	homephone_count_0_by_30	116	ssn_lastname_count_0_by_30
29	homephone_count_1_by_3	117	ssn_lastname_count_1_by_3
30	homephone_count_1_by_7	118	ssn_lastname_count_1_by_7
31	homephone_count_1_by_14	119	ssn_lastname_count_1_by_14
32	homephone_count_1_by_30	120	ssn_lastname_count_1_by_30
33	name_count_0_by_3	121	ssn_address_count_0_by_3
34	name_count_0_by_7	122	ssn_address_count_0_by_7
35	name_count_0_by_14	123	ssn_address_count_0_by_14

36	name_count_0_by_30	124	ssn_address_count_0_by_30
37	name_count_1_by_3	125	ssn_address_count_1_by_3
38	name_count_1_by_7	126	ssn_address_count_1_by_7
39	name_count_1_by_14	127	ssn_address_count_1_by_14
40	name_count_1_by_30	128	ssn_address_count_1_by_30
41	fulladdress_count_0_by_3	129	ssn_zip5_count_0_by_3
42	fulladdress_count_0_by_7	130	ssn_zip5_count_0_by_7
43	fulladdress_count_0_by_14	131	ssn_zip5_count_0_by_14
44	fulladdress_count_0_by_30	132	ssn_zip5_count_0_by_30
45	fulladdress_count_1_by_3	133	ssn_zip5_count_1_by_3
46	fulladdress_count_1_by_7	134	ssn_zip5_count_1_by_7
47	fulladdress_count_1_by_14	135	ssn_zip5_count_1_by_14
48	fulladdress_count_1_by_30	136	ssn_zip5_count_1_by_30
49	name_dob_count_0_by_3	137	ssn_dob_count_0_by_3
50	name_dob_count_0_by_7	138	ssn_dob_count_0_by_7
51	name_dob_count_0_by_14	139	ssn_dob_count_0_by_14
52	name_dob_count_0_by_30	140	ssn_dob_count_0_by_30
53	name_dob_count_1_by_3	141	ssn_dob_count_1_by_3
54	name_dob_count_1_by_7	142	ssn_dob_count_1_by_7
55	name_dob_count_1_by_14	143	ssn_dob_count_1_by_14
56	name_dob_count_1_by_30	144	ssn_dob_count_1_by_30
57	name_fulladdress_count_0_by_3	145	ssn_homephone_count_0_by_3
58	name_fulladdress_count_0_by_7	146	ssn_homephone_count_0_by_7
59	name_fulladdress_count_0_by_14	147	ssn_homephone_count_0_by_14
60	name_fulladdress_count_0_by_30	148	ssn_homephone_count_0_by_30
61	name_fulladdress_count_1_by_3	149	ssn_homephone_count_1_by_3
62	name_fulladdress_count_1_by_7	150	ssn_homephone_count_1_by_7
63	name_fulladdress_count_1_by_14	151	ssn_homephone_count_1_by_14
64	name_fulladdress_count_1_by_30	152	ssn_homephone_count_1_by_30
65	name_homephone_count_0_by_3	153	ssn_name_count_0_by_3
66	name_homephone_count_0_by_7	154	ssn_name_count_0_by_7
67	name_homephone_count_0_by_14	155	ssn_name_count_0_by_14
68	name_homephone_count_0_by_30	156	ssn_name_count_0_by_30
69	name_homephone_count_1_by_3	157	ssn_name_count_1_by_3
70	name_homephone_count_1_by_7	158	ssn_name_count_1_by_7
71	name_homephone_count_1_by_14	159	ssn_name_count_1_by_14
72	name_homephone_count_1_by_30	160	ssn_name_count_1_by_30
73	fulladdress_dob_count_0_by_3	161	ssn_fulladdress_count_0_by_3
74	fulladdress_dob_count_0_by_7	162	ssn_fulladdress_count_0_by_7
75	fulladdress_dob_count_0_by_14	163	ssn_fulladdress_count_0_by_14
76	fulladdress_dob_count_0_by_30	164	ssn_fulladdress_count_0_by_30
77	fulladdress_dob_count_1_by_3	165	ssn_fulladdress_count_1_by_3
78	fulladdress_dob_count_1_by_7	166	ssn_fulladdress_count_1_by_7

79	fulladdress_dob_count_1_by_14	167	ssn_fulladdress_count_1_by_14
80	fulladdress_dob_count_1_by_30	168	ssn_fulladdress_count_1_by_30
81	fulladdress_homephone_count_0_by_3	169	ssn_name_dob_count_0_by_3
82	fulladdress_homephone_count_0_by_7	170	ssn_name_dob_count_0_by_7
83	fulladdress_homephone_count_0_by_14	171	ssn_name_dob_count_0_by_14
84	fulladdress_homephone_count_0_by_30	172	ssn_name_dob_count_0_by_30
85	fulladdress_homephone_count_1_by_3	173	ssn_name_dob_count_1_by_3
86	fulladdress_homephone_count_1_by_7	174	ssn_name_dob_count_1_by_7
87	fulladdress_homephone_count_1_by_14	175	ssn_name_dob_count_1_by_14
88	fulladdress_homephone_count_1_by_30	176	ssn_name_dob_count_1_by_30

5. Unique Count Variables (540)

Unique Count Variable Name		Unique Count Variable Name	
1	#_unique_address_for_ssn_0	271	#_unique_ssn_for_address_0
2	#_unique_address_for_ssn_1	272	#_unique_ssn_for_address_1
3	#_unique_address_for_ssn_3	273	#_unique_ssn_for_address_3
4	#_unique_address_for_ssn_7	274	#_unique_ssn_for_address_7
5	#_unique_address_for_ssn_14	275	#_unique_ssn_for_address_14
6	#_unique_address_for_ssn_30	276	#_unique_ssn_for_address_30
7	#_unique_homephone_for_ssn_0	277	#_unique_dob_for_address_0
8	#_unique_homephone_for_ssn_1	278	#_unique_dob_for_address_1
9	#_unique_homephone_for_ssn_3	279	#_unique_dob_for_address_3
10	#_unique_homephone_for_ssn_7	280	#_unique_dob_for_address_7
11	#_unique_homephone_for_ssn_14	281	#_unique_dob_for_address_14
12	#_unique_homephone_for_ssn_30	282	#_unique_dob_for_address_30
13	#_unique_fulladdress_for_ssn_0	283	#_unique_name_for_address_0
14	#_unique_fulladdress_for_ssn_1	284	#_unique_name_for_address_1
15	#_unique_fulladdress_for_ssn_3	285	#_unique_name_for_address_3
16	#_unique_fulladdress_for_ssn_7	286	#_unique_name_for_address_7
17	#_unique_fulladdress_for_ssn_14	287	#_unique_name_for_address_14
18	#_unique_fulladdress_for_ssn_30	288	#_unique_name_for_address_30
19	#_unique_fulladdress_homephone_for_ssn_0	289	#_unique_name_dob_for_address_0
20	#_unique_fulladdress_homephone_for_ssn_1	290	#_unique_name_dob_for_address_1
21	#_unique_fulladdress_homephone_for_ssn_3	291	#_unique_name_dob_for_address_3
22	#_unique_fulladdress_homephone_for_ssn_7	292	#_unique_name_dob_for_address_7
23	#_unique_fulladdress_homephone_for_ssn_14	293	#_unique_name_dob_for_address_14
24	#_unique_fulladdress_homephone_for_ssn_30	294	#_unique_name_dob_for_address_30
25	#_unique_ssn_zip5_for_ssn_0	295	#_unique_ssn_firstname_for_address_0
26	#_unique_ssn_zip5_for_ssn_1	296	#_unique_ssn_firstname_for_address_1
27	#_unique_ssn_zip5_for_ssn_3	297	#_unique_ssn_firstname_for_address_3
28	#_unique_ssn_zip5_for_ssn_7	298	#_unique_ssn_firstname_for_address_7
29	#_unique_ssn_zip5_for_ssn_14	299	#_unique_ssn_firstname_for_address_14

30	#_unique_ssn_zip5_for_ssn_30	300	#_unique_ssn_firstname_for_address_30
31	#_unique_address_for_dob_0	301	#_unique_ssn_lastname_for_address_0
32	#_unique_address_for_dob_1	302	#_unique_ssn_lastname_for_address_1
33	#_unique_address_for_dob_3	303	#_unique_ssn_lastname_for_address_3
34	#_unique_address_for_dob_7	304	#_unique_ssn_lastname_for_address_7
35	#_unique_address_for_dob_14	305	#_unique_ssn_lastname_for_address_14
36	#_unique_address_for_dob_30	306	#_unique_ssn_lastname_for_address_30
37	#_unique_homephone_for_dob_0	307	#_unique_ssn_dob_for_address_0
38	#_unique_homephone_for_dob_1	308	#_unique_ssn_dob_for_address_1
39	#_unique_homephone_for_dob_3	309	#_unique_ssn_dob_for_address_3
40	#_unique_homephone_for_dob_7	310	#_unique_ssn_dob_for_address_7
41	#_unique_homephone_for_dob_14	311	#_unique_ssn_dob_for_address_14
42	#_unique_homephone_for_dob_30	312	#_unique_ssn_dob_for_address_30
43	#_unique_fulladdress_for_dob_0	313	#_unique_ssn_name_for_address_0
44	#_unique_fulladdress_for_dob_1	314	#_unique_ssn_name_for_address_1
45	#_unique_fulladdress_for_dob_3	315	#_unique_ssn_name_for_address_3
46	#_unique_fulladdress_for_dob_7	316	#_unique_ssn_name_for_address_7
47	#_unique_fulladdress_for_dob_14	317	#_unique_ssn_name_for_address_14
48	#_unique_fulladdress_for_dob_30	318	#_unique_ssn_name_for_address_30
49	#_unique_fulladdress_homephone_for_dob_0	319	#_unique_ssn_name_dob_for_address_0
50	#_unique_fulladdress_homephone_for_dob_1	320	#_unique_ssn_name_dob_for_address_1
51	#_unique_fulladdress_homephone_for_dob_3	321	#_unique_ssn_name_dob_for_address_3
52	#_unique_fulladdress_homephone_for_dob_7	322	#_unique_ssn_name_dob_for_address_7
53	#_unique_fulladdress_homephone_for_dob_14	323	#_unique_ssn_name_dob_for_address_14
54	#_unique_fulladdress_homephone_for_dob_30	324	#_unique_ssn_name_dob_for_address_30
55	#_unique_ssn_zip5_for_dob_0	325	#_unique_ssn_for_homephone_0
56	#_unique_ssn_zip5_for_dob_1	326	#_unique_ssn_for_homephone_1
57	#_unique_ssn_zip5_for_dob_3	327	#_unique_ssn_for_homephone_3
58	#_unique_ssn_zip5_for_dob_7	328	#_unique_ssn_for_homephone_7
59	#_unique_ssn_zip5_for_dob_14	329	#_unique_ssn_for_homephone_14
60	#_unique_ssn_zip5_for_dob_30	330	#_unique_ssn_for_homephone_30
61	#_unique_address_for_name_0	331	#_unique_dob_for_homephone_0
62	#_unique_address_for_name_1	332	#_unique_dob_for_homephone_1
63	#_unique_address_for_name_3	333	#_unique_dob_for_homephone_3
64	#_unique_address_for_name_7	334	#_unique_dob_for_homephone_7
65	#_unique_address_for_name_14	335	#_unique_dob_for_homephone_14
66	#_unique_address_for_name_30	336	#_unique_dob_for_homephone_30
67	#_unique_homephone_for_name_0	337	#_unique_name_for_homephone_0
68	#_unique_homephone_for_name_1	338	#_unique_name_for_homephone_1
69	#_unique_homephone_for_name_3	339	#_unique_name_for_homephone_3
70	#_unique_homephone_for_name_7	340	#_unique_name_for_homephone_7
71	#_unique_homephone_for_name_14	341	#_unique_name_for_homephone_14
72	#_unique_homephone_for_name_30	342	#_unique_name_for_homephone_30

73	#_unique_fulladdress_for_name_0	343	#_unique_name_dob_for_homephone_0
74	#_unique_fulladdress_for_name_1	344	#_unique_name_dob_for_homephone_1
75	#_unique_fulladdress_for_name_3	345	#_unique_name_dob_for_homephone_3
76	#_unique_fulladdress_for_name_7	346	#_unique_name_dob_for_homephone_7
77	#_unique_fulladdress_for_name_14	347	#_unique_name_dob_for_homephone_14
78	#_unique_fulladdress_for_name_30	348	#_unique_name_dob_for_homephone_30
79	#_unique_fulladdress_homephone_for_name_0	349	#_unique_ssn_firstname_for_homephone_0
80	#_unique_fulladdress_homephone_for_name_1	350	#_unique_ssn_firstname_for_homephone_1
81	#_unique_fulladdress_homephone_for_name_3	351	#_unique_ssn_firstname_for_homephone_3
82	#_unique_fulladdress_homephone_for_name_7	352	#_unique_ssn_firstname_for_homephone_7
83	#_unique_fulladdress_homephone_for_name_14	353	#_unique_ssn_firstname_for_homephone_14
84	#_unique_fulladdress_homephone_for_name_30	354	#_unique_ssn_firstname_for_homephone_30
85	#_unique_ssn_zip5_for_name_0	355	#_unique_ssn_lastname_for_homephone_0
86	#_unique_ssn_zip5_for_name_1	356	#_unique_ssn_lastname_for_homephone_1
87	#_unique_ssn_zip5_for_name_3	357	#_unique_ssn_lastname_for_homephone_3
88	#_unique_ssn_zip5_for_name_7	358	#_unique_ssn_lastname_for_homephone_7
89	#_unique_ssn_zip5_for_name_14	359	#_unique_ssn_lastname_for_homephone_14
90	#_unique_ssn_zip5_for_name_30	360	#_unique_ssn_lastname_for_homephone_30
91	#_unique_address_for_name_dob_0	361	#_unique_ssn_dob_for_homephone_0
92	#_unique_address_for_name_dob_1	362	#_unique_ssn_dob_for_homephone_1
93	#_unique_address_for_name_dob_3	363	#_unique_ssn_dob_for_homephone_3
94	#_unique_address_for_name_dob_7	364	#_unique_ssn_dob_for_homephone_7
95	#_unique_address_for_name_dob_14	365	#_unique_ssn_dob_for_homephone_14
96	#_unique_address_for_name_dob_30	366	#_unique_ssn_dob_for_homephone_30
97	#_unique_homephone_for_name_dob_0	367	#_unique_ssn_name_for_homephone_0
98	#_unique_homephone_for_name_dob_1	368	#_unique_ssn_name_for_homephone_1
99	#_unique_homephone_for_name_dob_3	369	#_unique_ssn_name_for_homephone_3
100	#_unique_homephone_for_name_dob_7	370	#_unique_ssn_name_for_homephone_7
101	#_unique_homephone_for_name_dob_14	371	#_unique_ssn_name_for_homephone_14
102	#_unique_homephone_for_name_dob_30	372	#_unique_ssn_name_for_homephone_30
103	#_unique_fulladdress_for_name_dob_0	373	#_unique_ssn_name_dob_for_homephone_0
104	#_unique_fulladdress_for_name_dob_1	374	#_unique_ssn_name_dob_for_homephone_1
105	#_unique_fulladdress_for_name_dob_3	375	#_unique_ssn_name_dob_for_homephone_3
106	#_unique_fulladdress_for_name_dob_7	376	#_unique_ssn_name_dob_for_homephone_7
107	#_unique_fulladdress_for_name_dob_14	377	#_unique_ssn_name_dob_for_homephone_14
108	#_unique_fulladdress_for_name_dob_30	378	#_unique_ssn_name_dob_for_homephone_30
109	#_unique_fulladdress_homephone_for_name_dob_0	379	#_unique_ssn_for_fulladdress_0
110	#_unique_fulladdress_homephone_for_name_dob_1	380	#_unique_ssn_for_fulladdress_1
111	#_unique_fulladdress_homephone_for_name_dob_3	381	#_unique_ssn_for_fulladdress_3
112	#_unique_fulladdress_homephone_for_name_dob_7	382	#_unique_ssn_for_fulladdress_7

113	#_unique_fulladdress_homephone_for_name_dob_14	383	#_unique_ssn_for_fulladdress_14
114	#_unique_fulladdress_homephone_for_name_dob_30	384	#_unique_ssn_for_fulladdress_30
115	#_unique_ssn_zip5_for_name_dob_0	385	#_unique_dob_for_fulladdress_0
116	#_unique_ssn_zip5_for_name_dob_1	386	#_unique_dob_for_fulladdress_1
117	#_unique_ssn_zip5_for_name_dob_3	387	#_unique_dob_for_fulladdress_3
118	#_unique_ssn_zip5_for_name_dob_7	388	#_unique_dob_for_fulladdress_7
119	#_unique_ssn_zip5_for_name_dob_14	389	#_unique_dob_for_fulladdress_14
120	#_unique_ssn_zip5_for_name_dob_30	390	#_unique_dob_for_fulladdress_30
121	#_unique_address_for_ssn_firstname_0	391	#_unique_name_for_fulladdress_0
122	#_unique_address_for_ssn_firstname_1	392	#_unique_name_for_fulladdress_1
123	#_unique_address_for_ssn_firstname_3	393	#_unique_name_for_fulladdress_3
124	#_unique_address_for_ssn_firstname_7	394	#_unique_name_for_fulladdress_7
125	#_unique_address_for_ssn_firstname_14	395	#_unique_name_for_fulladdress_14
126	#_unique_address_for_ssn_firstname_30	396	#_unique_name_for_fulladdress_30
127	#_unique_homephone_for_ssn_firstname_0	397	#_unique_name_dob_for_fulladdress_0
128	#_unique_homephone_for_ssn_firstname_1	398	#_unique_name_dob_for_fulladdress_1
129	#_unique_homephone_for_ssn_firstname_3	399	#_unique_name_dob_for_fulladdress_3
130	#_unique_homephone_for_ssn_firstname_7	400	#_unique_name_dob_for_fulladdress_7
131	#_unique_homephone_for_ssn_firstname_14	401	#_unique_name_dob_for_fulladdress_14
132	#_unique_homephone_for_ssn_firstname_30	402	#_unique_name_dob_for_fulladdress_30
133	#_unique_fulladdress_for_ssn_firstname_0	403	#_unique_ssn_firstname_for_fulladdress_0
134	#_unique_fulladdress_for_ssn_firstname_1	404	#_unique_ssn_firstname_for_fulladdress_1
135	#_unique_fulladdress_for_ssn_firstname_3	405	#_unique_ssn_firstname_for_fulladdress_3
136	#_unique_fulladdress_for_ssn_firstname_7	406	#_unique_ssn_firstname_for_fulladdress_7
137	#_unique_fulladdress_for_ssn_firstname_14	407	#_unique_ssn_firstname_for_fulladdress_14
138	#_unique_fulladdress_for_ssn_firstname_30	408	#_unique_ssn_firstname_for_fulladdress_30
139	#_unique_fulladdress_homephone_for_ssn_firstname_0	409	#_unique_ssn_lastname_for_fulladdress_0
140	#_unique_fulladdress_homephone_for_ssn_firstname_1	410	#_unique_ssn_lastname_for_fulladdress_1
141	#_unique_fulladdress_homephone_for_ssn_firstname_3	411	#_unique_ssn_lastname_for_fulladdress_3
142	#_unique_fulladdress_homephone_for_ssn_firstname_7	412	#_unique_ssn_lastname_for_fulladdress_7
143	#_unique_fulladdress_homephone_for_ssn_firstname_14	413	#_unique_ssn_lastname_for_fulladdress_14
144	#_unique_fulladdress_homephone_for_ssn_firstname_30	414	#_unique_ssn_lastname_for_fulladdress_30
145	#_unique_ssn_zip5_for_ssn_firstname_0	415	#_unique_ssn_dob_for_fulladdress_0
146	#_unique_ssn_zip5_for_ssn_firstname_1	416	#_unique_ssn_dob_for_fulladdress_1
147	#_unique_ssn_zip5_for_ssn_firstname_3	417	#_unique_ssn_dob_for_fulladdress_3
148	#_unique_ssn_zip5_for_ssn_firstname_7	418	#_unique_ssn_dob_for_fulladdress_7
149	#_unique_ssn_zip5_for_ssn_firstname_14	419	#_unique_ssn_dob_for_fulladdress_14
150	#_unique_ssn_zip5_for_ssn_firstname_30	420	#_unique_ssn_dob_for_fulladdress_30

151	#_unique_address_for_ssn_lastname_0	421	#_unique_ssn_name_for_fulladdress_0
152	#_unique_address_for_ssn_lastname_1	422	#_unique_ssn_name_for_fulladdress_1
153	#_unique_address_for_ssn_lastname_3	423	#_unique_ssn_name_for_fulladdress_3
154	#_unique_address_for_ssn_lastname_7	424	#_unique_ssn_name_for_fulladdress_7
155	#_unique_address_for_ssn_lastname_14	425	#_unique_ssn_name_for_fulladdress_14
156	#_unique_address_for_ssn_lastname_30	426	#_unique_ssn_name_for_fulladdress_30
157	#_unique_homephone_for_ssn_lastname_0	427	#_unique_ssn_name_dob_for_fulladdress_0
158	#_unique_homephone_for_ssn_lastname_1	428	#_unique_ssn_name_dob_for_fulladdress_1
159	#_unique_homephone_for_ssn_lastname_3	429	#_unique_ssn_name_dob_for_fulladdress_3
160	#_unique_homephone_for_ssn_lastname_7	430	#_unique_ssn_name_dob_for_fulladdress_7
161	#_unique_homephone_for_ssn_lastname_14	431	#_unique_ssn_name_dob_for_fulladdress_14
162	#_unique_homephone_for_ssn_lastname_30	432	#_unique_ssn_name_dob_for_fulladdress_30
163	#_unique_fulladdress_for_ssn_lastname_0	433	#_unique_ssn_for_fulladdress_homephone_0
164	#_unique_fulladdress_for_ssn_lastname_1	434	#_unique_ssn_for_fulladdress_homephone_1
165	#_unique_fulladdress_for_ssn_lastname_3	435	#_unique_ssn_for_fulladdress_homephone_3
166	#_unique_fulladdress_for_ssn_lastname_7	436	#_unique_ssn_for_fulladdress_homephone_7
167	#_unique_fulladdress_for_ssn_lastname_14	437	#_unique_ssn_for_fulladdress_homephone_14
168	#_unique_fulladdress_for_ssn_lastname_30	438	#_unique_ssn_for_fulladdress_homephone_30
169	#_unique_fulladdress_homephone_for_ssn_lastname_0	439	#_unique_dob_for_fulladdress_homephone_0
170	#_unique_fulladdress_homephone_for_ssn_lastname_1	440	#_unique_dob_for_fulladdress_homephone_1
171	#_unique_fulladdress_homephone_for_ssn_lastname_3	441	#_unique_dob_for_fulladdress_homephone_3
172	#_unique_fulladdress_homephone_for_ssn_lastname_7	442	#_unique_dob_for_fulladdress_homephone_7
173	#_unique_fulladdress_homephone_for_ssn_lastname_14	443	#_unique_dob_for_fulladdress_homephone_14
174	#_unique_fulladdress_homephone_for_ssn_lastname_30	444	#_unique_dob_for_fulladdress_homephone_30
175	#_unique_ssn_zip5_for_ssn_lastname_0	445	#_unique_name_for_fulladdress_homephone_0
176	#_unique_ssn_zip5_for_ssn_lastname_1	446	#_unique_name_for_fulladdress_homephone_1
177	#_unique_ssn_zip5_for_ssn_lastname_3	447	#_unique_name_for_fulladdress_homephone_3
178	#_unique_ssn_zip5_for_ssn_lastname_7	448	#_unique_name_for_fulladdress_homephone_7
179	#_unique_ssn_zip5_for_ssn_lastname_14	449	#_unique_name_for_fulladdress_homephone_14
180	#_unique_ssn_zip5_for_ssn_lastname_30	450	#_unique_name_for_fulladdress_homephone_30
181	#_unique_address_for_ssn_dob_0	451	#_unique_name_dob_for_fulladdress_homephone_0
182	#_unique_address_for_ssn_dob_1	452	#_unique_name_dob_for_fulladdress_homephone_1
183	#_unique_address_for_ssn_dob_3	453	#_unique_name_dob_for_fulladdress_homephone_3
184	#_unique_address_for_ssn_dob_7	454	#_unique_name_dob_for_fulladdress_homephone_7
185	#_unique_address_for_ssn_dob_14	455	#_unique_name_dob_for_fulladdress_homephone_14
186	#_unique_address_for_ssn_dob_30	456	#_unique_name_dob_for_fulladdress_

			homephone_30
187	#_unique_homephone_for_ssn_dob_0	457	#_unique_ssn_firstname_for_fulladdress_homephone_0
188	#_unique_homephone_for_ssn_dob_1	458	#_unique_ssn_firstname_for_fulladdress_homephone_1
189	#_unique_homephone_for_ssn_dob_3	459	#_unique_ssn_firstname_for_fulladdress_homephone_3
190	#_unique_homephone_for_ssn_dob_7	460	#_unique_ssn_firstname_for_fulladdress_homephone_7
191	#_unique_homephone_for_ssn_dob_14	461	#_unique_ssn_firstname_for_fulladdress_homephone_14
192	#_unique_homephone_for_ssn_dob_30	462	#_unique_ssn_firstname_for_fulladdress_homephone_30
193	#_unique_fulladdress_for_ssn_dob_0	463	#_unique_ssn_lastname_for_fulladdress_homephone_0
194	#_unique_fulladdress_for_ssn_dob_1	464	#_unique_ssn_lastname_for_fulladdress_homephone_1
195	#_unique_fulladdress_for_ssn_dob_3	465	#_unique_ssn_lastname_for_fulladdress_homephone_3
196	#_unique_fulladdress_for_ssn_dob_7	466	#_unique_ssn_lastname_for_fulladdress_homephone_7
197	#_unique_fulladdress_for_ssn_dob_14	467	#_unique_ssn_lastname_for_fulladdress_homephone_14
198	#_unique_fulladdress_for_ssn_dob_30	468	#_unique_ssn_lastname_for_fulladdress_homephone_30
199	#_unique_fulladdress_homephone_for_ssn_dob_0	469	#_unique_ssn_dob_for_fulladdress_homephone_0
200	#_unique_fulladdress_homephone_for_ssn_dob_1	470	#_unique_ssn_dob_for_fulladdress_homephone_1
201	#_unique_fulladdress_homephone_for_ssn_dob_3	471	#_unique_ssn_dob_for_fulladdress_homephone_3
202	#_unique_fulladdress_homephone_for_ssn_dob_7	472	#_unique_ssn_dob_for_fulladdress_homephone_7
203	#_unique_fulladdress_homephone_for_ssn_dob_14	473	#_unique_ssn_dob_for_fulladdress_homephone_14
204	#_unique_fulladdress_homephone_for_ssn_dob_30	474	#_unique_ssn_dob_for_fulladdress_homephone_30
205	#_unique_ssn_zip5_for_ssn_dob_0	475	#_unique_ssn_name_for_fulladdress_homephone_0
206	#_unique_ssn_zip5_for_ssn_dob_1	476	#_unique_ssn_name_for_fulladdress_homephone_1
207	#_unique_ssn_zip5_for_ssn_dob_3	477	#_unique_ssn_name_for_fulladdress_homephone_3
208	#_unique_ssn_zip5_for_ssn_dob_7	478	#_unique_ssn_name_for_fulladdress_homephone_7
209	#_unique_ssn_zip5_for_ssn_dob_14	479	#_unique_ssn_name_for_fulladdress_homephone_14
210	#_unique_ssn_zip5_for_ssn_dob_30	480	#_unique_ssn_name_for_fulladdress_homephone_30
211	#_unique_address_for_ssn_name_0	481	#_unique_ssn_name_dob_for_fulladdress_homephone_0

212	#_unique_address_for_ssn_name_1	482	#_unique_ssn_name_dob_for_fulladdress_homephone_1
213	#_unique_address_for_ssn_name_3	483	#_unique_ssn_name_dob_for_fulladdress_homephone_3
214	#_unique_address_for_ssn_name_7	484	#_unique_ssn_name_dob_for_fulladdress_homephone_7
215	#_unique_address_for_ssn_name_14	485	#_unique_ssn_name_dob_for_fulladdress_homephone_14
216	#_unique_address_for_ssn_name_30	486	#_unique_ssn_name_dob_for_fulladdress_homephone_30
217	#_unique_homephone_for_ssn_name_0	487	#_unique_ssn_for_ssn_zip5_0
218	#_unique_homephone_for_ssn_name_1	488	#_unique_ssn_for_ssn_zip5_1
219	#_unique_homephone_for_ssn_name_3	489	#_unique_ssn_for_ssn_zip5_3
220	#_unique_homephone_for_ssn_name_7	490	#_unique_ssn_for_ssn_zip5_7
221	#_unique_homephone_for_ssn_name_14	491	#_unique_ssn_for_ssn_zip5_14
222	#_unique_homephone_for_ssn_name_30	492	#_unique_ssn_for_ssn_zip5_30
223	#_unique_fulladdress_for_ssn_name_0	493	#_unique_dob_for_ssn_zip5_0
224	#_unique_fulladdress_for_ssn_name_1	494	#_unique_dob_for_ssn_zip5_1
225	#_unique_fulladdress_for_ssn_name_3	495	#_unique_dob_for_ssn_zip5_3
226	#_unique_fulladdress_for_ssn_name_7	496	#_unique_dob_for_ssn_zip5_7
227	#_unique_fulladdress_for_ssn_name_14	497	#_unique_dob_for_ssn_zip5_14
228	#_unique_fulladdress_for_ssn_name_30	498	#_unique_dob_for_ssn_zip5_30
229	#_unique_fulladdress_homephone_for_ssn_name_0	499	#_unique_name_for_ssn_zip5_0
230	#_unique_fulladdress_homephone_for_ssn_name_1	500	#_unique_name_for_ssn_zip5_1
231	#_unique_fulladdress_homephone_for_ssn_name_3	501	#_unique_name_for_ssn_zip5_3
232	#_unique_fulladdress_homephone_for_ssn_name_7	502	#_unique_name_for_ssn_zip5_7
233	#_unique_fulladdress_homephone_for_ssn_name_14	503	#_unique_name_for_ssn_zip5_14
234	#_unique_fulladdress_homephone_for_ssn_name_30	504	#_unique_name_for_ssn_zip5_30
235	#_unique_ssn_zip5_for_ssn_name_0	505	#_unique_name_dob_for_ssn_zip5_0
236	#_unique_ssn_zip5_for_ssn_name_1	506	#_unique_name_dob_for_ssn_zip5_1
237	#_unique_ssn_zip5_for_ssn_name_3	507	#_unique_name_dob_for_ssn_zip5_3
238	#_unique_ssn_zip5_for_ssn_name_7	508	#_unique_name_dob_for_ssn_zip5_7
239	#_unique_ssn_zip5_for_ssn_name_14	509	#_unique_name_dob_for_ssn_zip5_14
240	#_unique_ssn_zip5_for_ssn_name_30	510	#_unique_name_dob_for_ssn_zip5_30
241	#_unique_address_for_ssn_name_dob_0	511	#_unique_ssn_firstname_for_ssn_zip5_0
242	#_unique_address_for_ssn_name_dob_1	512	#_unique_ssn_firstname_for_ssn_zip5_1
243	#_unique_address_for_ssn_name_dob_3	513	#_unique_ssn_firstname_for_ssn_zip5_3
244	#_unique_address_for_ssn_name_dob_7	514	#_unique_ssn_firstname_for_ssn_zip5_7
245	#_unique_address_for_ssn_name_dob_14	515	#_unique_ssn_firstname_for_ssn_zip5_14
246	#_unique_address_for_ssn_name_dob_30	516	#_unique_ssn_firstname_for_ssn_zip5_30
247	#_unique_homephone_for_ssn_name_dob_0	517	#_unique_ssn_lastname_for_ssn_zip5_0

248	#_unique_homephone_for_ssn_name_dob_1	518	#_unique_ssn_lastname_for_ssn_zip5_1
249	#_unique_homephone_for_ssn_name_dob_3	519	#_unique_ssn_lastname_for_ssn_zip5_3
250	#_unique_homephone_for_ssn_name_dob_7	520	#_unique_ssn_lastname_for_ssn_zip5_7
251	#_unique_homephone_for_ssn_name_dob_14	521	#_unique_ssn_lastname_for_ssn_zip5_14
252	#_unique_homephone_for_ssn_name_dob_30	522	#_unique_ssn_lastname_for_ssn_zip5_30
253	#_unique_fulladdress_for_ssn_name_dob_0	523	#_unique_ssn_dob_for_ssn_zip5_0
254	#_unique_fulladdress_for_ssn_name_dob_1	524	#_unique_ssn_dob_for_ssn_zip5_1
255	#_unique_fulladdress_for_ssn_name_dob_3	525	#_unique_ssn_dob_for_ssn_zip5_3
256	#_unique_fulladdress_for_ssn_name_dob_7	526	#_unique_ssn_dob_for_ssn_zip5_7
257	#_unique_fulladdress_for_ssn_name_dob_14	527	#_unique_ssn_dob_for_ssn_zip5_14
258	#_unique_fulladdress_for_ssn_name_dob_30	528	#_unique_ssn_dob_for_ssn_zip5_30
259	#_unique_fulladdress_homephone_for_ssn_name_dob_0	529	#_unique_ssn_name_for_ssn_zip5_0
260	#_unique_fulladdress_homephone_for_ssn_name_dob_1	530	#_unique_ssn_name_for_ssn_zip5_1
261	#_unique_fulladdress_homephone_for_ssn_name_dob_3	531	#_unique_ssn_name_for_ssn_zip5_3
262	#_unique_fulladdress_homephone_for_ssn_name_dob_7	532	#_unique_ssn_name_for_ssn_zip5_7
263	#_unique_fulladdress_homephone_for_ssn_name_dob_14	533	#_unique_ssn_name_for_ssn_zip5_14
264	#_unique_fulladdress_homephone_for_ssn_name_dob_30	534	#_unique_ssn_name_for_ssn_zip5_30
265	#_unique_ssn_zip5_for_ssn_name_dob_0	535	#_unique_ssn_name_dob_for_ssn_zip5_0
266	#_unique_ssn_zip5_for_ssn_name_dob_1	536	#_unique_ssn_name_dob_for_ssn_zip5_1
267	#_unique_ssn_zip5_for_ssn_name_dob_3	537	#_unique_ssn_name_dob_for_ssn_zip5_3
268	#_unique_ssn_zip5_for_ssn_name_dob_7	538	#_unique_ssn_name_dob_for_ssn_zip5_7
269	#_unique_ssn_zip5_for_ssn_name_dob_14	539	#_unique_ssn_name_dob_for_ssn_zip5_14
270	#_unique_ssn_zip5_for_ssn_name_dob_30	540	#_unique_ssn_name_dob_for_ssn_zip5_30

Appendix C. Full Final Result Tables

1. Training Result

Training	# Records	# Goods	# Bads	Fraud Rate								
	596247	587635	8612	0.014443679								
Bin Statistics						Cumulative Statistics						
Population Bin %	# Records	# Goods	# Bads	% Goods	% Bads	Total # Records	Cumulative Goods	Cumulative Bads	Cumulative % Goods	Cumulative % Bads (FDR)	KS Score	FPR
1	5963	1274	4689	21.37%	78.63%	5963	1274	4689	0.22%	54.45%	54.23	0.27
2	5963	5713	250	95.81%	4.19%	11926	6987	4939	1.19%	57.35%	56.16	1.41
3	5963	5886	77	98.71%	1.29%	17889	12873	5016	2.19%	58.24%	56.05	2.57
4	5963	5902	61	98.98%	1.02%	23852	18775	5077	3.20%	58.95%	55.75	3.7
5	5963	5898	65	98.91%	1.09%	29815	24673	5142	4.20%	59.71%	55.51	4.8
6	5963	5917	46	99.23%	0.77%	35778	30590	5188	5.21%	60.24%	55.03	5.9
7	5963	5907	56	99.06%	0.94%	41741	36497	5244	6.21%	60.89%	54.68	6.96
8	5963	5922	41	99.31%	0.69%	47704	42419	5285	7.22%	61.37%	54.15	8.03
9	5963	5913	50	99.16%	0.84%	53667	48332	5335	8.22%	61.95%	53.73	9.06
10	5963	5919	44	99.26%	0.74%	59630	54251	5379	9.23%	62.46%	53.23	10.09
11	5963	5919	44	99.26%	0.74%	65593	60170	5423	10.24%	62.97%	52.73	11.1
12	5963	5916	47	99.21%	0.79%	71556	66086	5470	11.25%	63.52%	52.27	12.08
13	5963	5933	30	99.50%	0.50%	77519	72019	5500	12.26%	63.86%	51.6	13.09
14	5963	5927	36	99.40%	0.60%	83482	77946	5536	13.26%	64.28%	51.02	14.08
15	5963	5926	37	99.38%	0.62%	89445	83872	5573	14.27%	64.71%	50.44	15.05
16	5963	5920	43	99.28%	0.72%	95408	89792	5616	15.28%	65.21%	49.93	15.99
17	5963	5920	43	99.28%	0.72%	101371	95712	5659	16.29%	65.71%	49.42	16.91
18	5963	5930	33	99.45%	0.55%	107334	101642	5692	17.30%	66.09%	48.79	17.86
19	5963	5924	39	99.35%	0.65%	113297	107566	5731	18.30%	66.55%	48.25	18.77
20	5963	5925	38	99.36%	0.64%	119260	113491	5769	19.31%	66.99%	47.68	19.67

21	5963	5917	46	99.23%	0.77%	125223	119408	5815	20.32%	67.52%	47.2	20.53
22	5963	5920	43	99.28%	0.72%	131186	125328	5858	21.33%	68.02%	46.69	21.39
23	5963	5922	41	99.31%	0.69%	137149	131250	5899	22.34%	68.50%	46.16	22.25
24	5963	5921	42	99.30%	0.70%	143112	137171	5941	23.34%	68.99%	45.65	23.09
25	5963	5921	42	99.30%	0.70%	149075	143092	5983	24.35%	69.47%	45.12	23.92
26	5963	5913	50	99.16%	0.84%	155038	149005	6033	25.36%	70.05%	44.69	24.7
27	5963	5912	51	99.14%	0.86%	161001	154917	6084	26.36%	70.65%	44.29	25.46
28	5963	5931	32	99.46%	0.54%	166964	160848	6116	27.37%	71.02%	43.65	26.3
29	5963	5923	40	99.33%	0.67%	172927	166771	6156	28.38%	71.48%	43.1	27.09
30	5963	5924	39	99.35%	0.65%	178890	172695	6195	29.39%	71.93%	42.54	27.88
31	5963	5937	26	99.56%	0.44%	184853	178632	6221	30.40%	72.24%	41.84	28.71
32	5963	5930	33	99.45%	0.55%	190816	184562	6254	31.41%	72.62%	41.21	29.51
33	5963	5922	41	99.31%	0.69%	196779	190484	6295	32.42%	73.10%	40.68	30.26
34	5963	5931	32	99.46%	0.54%	202742	196415	6327	33.42%	73.47%	40.05	31.04
35	5963	5917	46	99.23%	0.77%	208705	202332	6373	34.43%	74.00%	39.57	31.75
36	5963	5933	30	99.50%	0.50%	214668	208265	6403	35.44%	74.35%	38.91	32.53
37	5963	5931	32	99.46%	0.54%	220631	214196	6435	36.45%	74.72%	38.27	33.29
38	5963	5918	45	99.25%	0.75%	226594	220114	6480	37.46%	75.24%	37.78	33.97
39	5963	5923	40	99.33%	0.67%	232557	226037	6520	38.47%	75.71%	37.24	34.67
40	5963	5923	40	99.33%	0.67%	238520	231960	6560	39.47%	76.17%	36.7	35.36
41	5963	5926	37	99.38%	0.62%	244483	237886	6597	40.48%	76.60%	36.12	36.06
42	5963	5924	39	99.35%	0.65%	250446	243810	6636	41.49%	77.06%	35.57	36.74
43	5963	5928	35	99.41%	0.59%	256409	249738	6671	42.50%	77.46%	34.96	37.44
44	5963	5919	44	99.26%	0.74%	262372	255657	6715	43.51%	77.97%	34.46	38.07
45	5963	5924	39	99.35%	0.65%	268335	261581	6754	44.51%	78.43%	33.92	38.73
46	5963	5918	45	99.25%	0.75%	274298	267499	6799	45.52%	78.95%	33.43	39.34
47	5963	5932	31	99.48%	0.52%	280261	273431	6830	46.53%	79.31%	32.78	40.03
48	5963	5917	46	99.23%	0.77%	286224	279348	6876	47.54%	79.84%	32.3	40.63
49	5963	5930	33	99.45%	0.55%	292187	285278	6909	48.55%	80.23%	31.68	41.29
50	5963	5931	32	99.46%	0.54%	298150	291209	6941	49.56%	80.60%	31.04	41.95

51	5963	5930	33	99.45%	0.55%	304113	297139	6974	50.57%	80.98%	30.41	42.61
52	5963	5931	32	99.46%	0.54%	310076	303070	7006	51.57%	81.35%	29.78	43.26
53	5963	5916	47	99.21%	0.79%	316039	308986	7053	52.58%	81.90%	29.32	43.81
54	5963	5914	49	99.18%	0.82%	322002	314900	7102	53.59%	82.47%	28.88	44.34
55	5963	5924	39	99.35%	0.65%	327965	320824	7141	54.60%	82.92%	28.32	44.93
56	5963	5925	38	99.36%	0.64%	333928	326749	7179	55.60%	83.36%	27.76	45.51
57	5963	5920	43	99.28%	0.72%	339891	332669	7222	56.61%	83.86%	27.25	46.06
58	5963	5924	39	99.35%	0.65%	345854	338593	7261	57.62%	84.31%	26.69	46.63
59	5963	5929	34	99.43%	0.57%	351817	344522	7295	58.63%	84.71%	26.08	47.23
60	5963	5927	36	99.40%	0.60%	357780	350449	7331	59.64%	85.13%	25.49	47.8
61	5963	5932	31	99.48%	0.52%	363743	356381	7362	60.65%	85.49%	24.84	48.41
62	5963	5932	31	99.48%	0.52%	369706	362313	7393	61.66%	85.85%	24.19	49.01
63	5963	5925	38	99.36%	0.64%	375669	368238	7431	62.66%	86.29%	23.63	49.55
64	5963	5932	31	99.48%	0.52%	381632	374170	7462	63.67%	86.65%	22.98	50.14
65	5963	5933	30	99.50%	0.50%	387595	380103	7492	64.68%	86.99%	22.31	50.73
66	5963	5928	35	99.41%	0.59%	393558	386031	7527	65.69%	87.40%	21.71	51.29
67	5963	5922	41	99.31%	0.69%	399521	391953	7568	66.70%	87.88%	21.18	51.79
68	5963	5914	49	99.18%	0.82%	405484	397867	7617	67.71%	88.45%	20.74	52.23
69	5963	5926	37	99.38%	0.62%	411447	403793	7654	68.71%	88.88%	20.17	52.76
70	5963	5918	45	99.25%	0.75%	417410	409711	7699	69.72%	89.40%	19.68	53.22
71	5963	5924	39	99.35%	0.65%	423373	415635	7738	70.73%	89.85%	19.12	53.71
72	5963	5930	33	99.45%	0.55%	429336	421565	7771	71.74%	90.23%	18.49	54.25
73	5963	5933	30	99.50%	0.50%	435299	427498	7801	72.75%	90.58%	17.83	54.8
74	5963	5927	36	99.40%	0.60%	441262	433425	7837	73.76%	91.00%	17.24	55.3
75	5963	5923	40	99.33%	0.67%	447225	439348	7877	74.77%	91.47%	16.7	55.78
76	5963	5931	32	99.46%	0.54%	453188	445279	7909	75.77%	91.84%	16.07	56.3
77	5963	5929	34	99.43%	0.57%	459151	451208	7943	76.78%	92.23%	15.45	56.81
78	5963	5924	39	99.35%	0.65%	465114	457132	7982	77.79%	92.68%	14.89	57.27
79	5963	5934	29	99.51%	0.49%	471077	463066	8011	78.80%	93.02%	14.22	57.8
80	5963	5929	34	99.43%	0.57%	477040	468995	8045	79.81%	93.42%	13.61	58.3

81	5963	5929	34	99.43%	0.57%	483003	474924	8079	80.82%	93.81%	12.99	58.78
82	5963	5930	33	99.45%	0.55%	488966	480854	8112	81.83%	94.19%	12.36	59.28
83	5963	5922	41	99.31%	0.69%	494929	486776	8153	82.84%	94.67%	11.83	59.71
84	5963	5926	37	99.38%	0.62%	500892	492702	8190	83.84%	95.10%	11.26	60.16
85	5963	5919	44	99.26%	0.74%	506855	498621	8234	84.85%	95.61%	10.76	60.56
86	5963	5928	35	99.41%	0.59%	512818	504549	8269	85.86%	96.02%	10.16	61.02
87	5963	5920	43	99.28%	0.72%	518781	510469	8312	86.87%	96.52%	9.65	61.41
88	5963	5922	41	99.31%	0.69%	524744	516391	8353	87.88%	96.99%	9.11	61.82
89	5963	5930	33	99.45%	0.55%	530707	522321	8386	88.89%	97.38%	8.49	62.28
90	5963	5925	38	99.36%	0.64%	536670	528246	8424	89.89%	97.82%	7.93	62.71
91	5963	5932	31	99.48%	0.52%	542633	534178	8455	90.90%	98.18%	7.28	63.18
92	5963	5931	32	99.46%	0.54%	548596	540109	8487	91.91%	98.55%	6.64	63.64
93	5963	5935	28	99.53%	0.47%	554559	546044	8515	92.92%	98.87%	5.95	64.13
94	5963	5947	16	99.73%	0.27%	560522	551991	8531	93.93%	99.06%	5.13	64.7
95	5963	5940	23	99.61%	0.39%	566485	557931	8554	94.95%	99.33%	4.38	65.22
96	5963	5941	22	99.63%	0.37%	572448	563872	8576	95.96%	99.58%	3.62	65.75
97	5963	5954	9	99.85%	0.15%	578411	569826	8585	96.97%	99.69%	2.72	66.37
98	5963	5948	15	99.75%	0.25%	584374	575774	8600	97.98%	99.86%	1.88	66.95
99	5963	5956	7	99.88%	0.12%	590337	581730	8607	99.00%	99.94%	0.94	67.59
100	5910	5905	5	99.92%	0.08%	596247	587635	8612	100.00%	100.00%	0	68.23

2. Testing Result

Testing	# Records	# Goods	# Bads	Fraud Rate								
	198749	195875	2874	0.01446045								
Bin Statistics						Cumulative Statistics						
Population Bin %	# Records	# Goods	# Bads	% Goods	% Bads	Total # Records	Cumulative Goods	Cumulative Bads	Cumulative % Goods	Cumulative % Bads (FDR)	KS Score	FPR
1	1988	462	1526	23.24%	76.76%	1988	462	1526	0.24%	53.10%	52.86	0.3
2	1988	1900	88	95.57%	4.43%	3976	2362	1614	1.21%	56.16%	54.95	1.46
3	1988	1967	21	98.94%	1.06%	5964	4329	1635	2.21%	56.89%	54.68	2.65
4	1988	1975	13	99.35%	0.65%	7952	6304	1648	3.22%	57.34%	54.12	3.83
5	1988	1968	20	98.99%	1.01%	9940	8272	1668	4.22%	58.04%	53.82	4.96
6	1988	1974	14	99.30%	0.70%	11928	10246	1682	5.23%	58.52%	53.29	6.09
7	1988	1975	13	99.35%	0.65%	13916	12221	1695	6.24%	58.98%	52.74	7.21
8	1988	1973	15	99.25%	0.75%	15904	14194	1710	7.25%	59.50%	52.25	8.3
9	1988	1966	22	98.89%	1.11%	17892	16160	1732	8.25%	60.26%	52.01	9.33
10	1988	1966	22	98.89%	1.11%	19880	18126	1754	9.25%	61.03%	51.78	10.33
11	1988	1968	20	98.99%	1.01%	21868	20094	1774	10.26%	61.73%	51.47	11.33
12	1988	1981	7	99.65%	0.35%	23856	22075	1781	11.27%	61.97%	50.7	12.39
13	1988	1970	18	99.09%	0.91%	25844	24045	1799	12.28%	62.60%	50.32	13.37
14	1988	1972	16	99.20%	0.80%	27832	26017	1815	13.28%	63.15%	49.87	14.33
15	1988	1976	12	99.40%	0.60%	29820	27993	1827	14.29%	63.57%	49.28	15.32
16	1988	1969	19	99.04%	0.96%	31808	29962	1846	15.30%	64.23%	48.93	16.23
17	1988	1972	16	99.20%	0.80%	33796	31934	1862	16.30%	64.79%	48.49	17.15
18	1988	1976	12	99.40%	0.60%	35784	33910	1874	17.31%	65.21%	47.9	18.09
19	1988	1972	16	99.20%	0.80%	37772	35882	1890	18.32%	65.76%	47.44	18.99
20	1988	1966	22	98.89%	1.11%	39760	37848	1912	19.32%	66.53%	47.21	19.79

21	1988	1976	12	99.40%	0.60%	41748	39824	1924	20.33%	66.95%	46.62	20.7
22	1988	1976	12	99.40%	0.60%	43736	41800	1936	21.34%	67.36%	46.02	21.59
23	1988	1979	9	99.55%	0.45%	45724	43779	1945	22.35%	67.68%	45.33	22.51
24	1988	1974	14	99.30%	0.70%	47712	45753	1959	23.36%	68.16%	44.8	23.36
25	1988	1976	12	99.40%	0.60%	49700	47729	1971	24.37%	68.58%	44.21	24.22
26	1988	1981	7	99.65%	0.35%	51688	49710	1978	25.38%	68.82%	43.44	25.13
27	1988	1977	11	99.45%	0.55%	53676	51687	1989	26.39%	69.21%	42.82	25.99
28	1988	1973	15	99.25%	0.75%	55664	53660	2004	27.40%	69.73%	42.33	26.78
29	1988	1978	10	99.50%	0.50%	57652	55638	2014	28.40%	70.08%	41.68	27.63
30	1988	1972	16	99.20%	0.80%	59640	57610	2030	29.41%	70.63%	41.22	28.38
31	1988	1971	17	99.14%	0.86%	61628	59581	2047	30.42%	71.22%	40.8	29.11
32	1988	1976	12	99.40%	0.60%	63616	61557	2059	31.43%	71.64%	40.21	29.9
33	1988	1973	15	99.25%	0.75%	65604	63530	2074	32.43%	72.16%	39.73	30.63
34	1988	1977	11	99.45%	0.55%	67592	65507	2085	33.44%	72.55%	39.11	31.42
35	1988	1974	14	99.30%	0.70%	69580	67481	2099	34.45%	73.03%	38.58	32.15
36	1988	1974	14	99.30%	0.70%	71568	69455	2113	35.46%	73.52%	38.06	32.87
37	1988	1974	14	99.30%	0.70%	73556	71429	2127	36.47%	74.01%	37.54	33.58
38	1988	1975	13	99.35%	0.65%	75544	73404	2140	37.47%	74.46%	36.99	34.3
39	1988	1974	14	99.30%	0.70%	77532	75378	2154	38.48%	74.95%	36.47	34.99
40	1988	1975	13	99.35%	0.65%	79520	77353	2167	39.49%	75.40%	35.91	35.7
41	1988	1973	15	99.25%	0.75%	81508	79326	2182	40.50%	75.92%	35.42	36.35
42	1988	1973	15	99.25%	0.75%	83496	81299	2197	41.51%	76.44%	34.93	37
43	1988	1981	7	99.65%	0.35%	85484	83280	2204	42.52%	76.69%	34.17	37.79
44	1988	1973	15	99.25%	0.75%	87472	85253	2219	43.52%	77.21%	33.69	38.42
45	1988	1977	11	99.45%	0.55%	89460	87230	2230	44.53%	77.59%	33.06	39.12
46	1988	1976	12	99.40%	0.60%	91448	89206	2242	45.54%	78.01%	32.47	39.79
47	1988	1982	6	99.70%	0.30%	93436	91188	2248	46.55%	78.22%	31.67	40.56
48	1988	1977	11	99.45%	0.55%	95424	93165	2259	47.56%	78.60%	31.04	41.24
49	1988	1973	15	99.25%	0.75%	97412	95138	2274	48.57%	79.12%	30.55	41.84
50	1988	1970	18	99.09%	0.91%	99400	97108	2292	49.58%	79.75%	30.17	42.37

51	1988	1978	10	99.50%	0.50%	101388	99086	2302	50.59%	80.10%	29.51	43.04
52	1988	1977	11	99.45%	0.55%	103376	101063	2313	51.60%	80.48%	28.88	43.69
53	1988	1975	13	99.35%	0.65%	105364	103038	2326	52.60%	80.93%	28.33	44.3
54	1988	1981	7	99.65%	0.35%	107352	105019	2333	53.62%	81.18%	27.56	45.01
55	1988	1972	16	99.20%	0.80%	109340	106991	2349	54.62%	81.73%	27.11	45.55
56	1988	1975	13	99.35%	0.65%	111328	108966	2362	55.63%	82.19%	26.56	46.13
57	1988	1978	10	99.50%	0.50%	113316	110944	2372	56.64%	82.53%	25.89	46.77
58	1988	1977	11	99.45%	0.55%	115304	112921	2383	57.65%	82.92%	25.27	47.39
59	1988	1977	11	99.45%	0.55%	117292	114898	2394	58.66%	83.30%	24.64	47.99
60	1988	1978	10	99.50%	0.50%	119280	116876	2404	59.67%	83.65%	23.98	48.62
61	1988	1976	12	99.40%	0.60%	121268	118852	2416	60.68%	84.06%	23.38	49.19
62	1988	1975	13	99.35%	0.65%	123256	120827	2429	61.69%	84.52%	22.83	49.74
63	1988	1981	7	99.65%	0.35%	125244	122808	2436	62.70%	84.76%	22.06	50.41
64	1988	1978	10	99.50%	0.50%	127232	124786	2446	63.71%	85.11%	21.4	51.02
65	1988	1978	10	99.50%	0.50%	129220	126764	2456	64.72%	85.46%	20.74	51.61
66	1988	1978	10	99.50%	0.50%	131208	128742	2466	65.73%	85.80%	20.07	52.21
67	1988	1973	15	99.25%	0.75%	133196	130715	2481	66.73%	86.33%	19.6	52.69
68	1988	1977	11	99.45%	0.55%	135184	132692	2492	67.74%	86.71%	18.97	53.25
69	1988	1975	13	99.35%	0.65%	137172	134667	2505	68.75%	87.16%	18.41	53.76
70	1988	1980	8	99.60%	0.40%	139160	136647	2513	69.76%	87.44%	17.68	54.38
71	1988	1978	10	99.50%	0.50%	141148	138625	2523	70.77%	87.79%	17.02	54.94
72	1988	1977	11	99.45%	0.55%	143136	140602	2534	71.78%	88.17%	16.39	55.49
73	1988	1979	9	99.55%	0.45%	145124	142581	2543	72.79%	88.48%	15.69	56.07
74	1988	1971	17	99.14%	0.86%	147112	144552	2560	73.80%	89.07%	15.27	56.47
75	1988	1975	13	99.35%	0.65%	149100	146527	2573	74.81%	89.53%	14.72	56.95
76	1988	1977	11	99.45%	0.55%	151088	148504	2584	75.82%	89.91%	14.09	57.47
77	1988	1975	13	99.35%	0.65%	153076	150479	2597	76.82%	90.36%	13.54	57.94
78	1988	1971	17	99.14%	0.86%	155064	152450	2614	77.83%	90.95%	13.12	58.32
79	1988	1973	15	99.25%	0.75%	157052	154423	2629	78.84%	91.48%	12.64	58.74
80	1988	1973	15	99.25%	0.75%	159040	156396	2644	79.84%	92.00%	12.16	59.15

81	1988	1969	19	99.04%	0.96%	161028	158365	2663	80.85%	92.66%	11.81	59.47
82	1988	1975	13	99.35%	0.65%	163016	160340	2676	81.86%	93.11%	11.25	59.92
83	1988	1975	13	99.35%	0.65%	165004	162315	2689	82.87%	93.56%	10.69	60.36
84	1988	1974	14	99.30%	0.70%	166992	164289	2703	83.87%	94.05%	10.18	60.78
85	1988	1979	9	99.55%	0.45%	168980	166268	2712	84.88%	94.36%	9.48	61.31
86	1988	1978	10	99.50%	0.50%	170968	168246	2722	85.89%	94.71%	8.82	61.81
87	1988	1972	16	99.20%	0.80%	172956	170218	2738	86.90%	95.27%	8.37	62.17
88	1988	1976	12	99.40%	0.60%	174944	172194	2750	87.91%	95.69%	7.78	62.62
89	1988	1974	14	99.30%	0.70%	176932	174168	2764	88.92%	96.17%	7.25	63.01
90	1988	1975	13	99.35%	0.65%	178920	176143	2777	89.93%	96.62%	6.69	63.43
91	1988	1976	12	99.40%	0.60%	180908	178119	2789	90.94%	97.04%	6.1	63.86
92	1988	1978	10	99.50%	0.50%	182896	180097	2799	91.94%	97.39%	5.45	64.34
93	1988	1981	7	99.65%	0.35%	184884	182078	2806	92.96%	97.63%	4.67	64.89
94	1988	1978	10	99.50%	0.50%	186872	184056	2816	93.97%	97.98%	4.01	65.36
95	1988	1978	10	99.50%	0.50%	188860	186034	2826	94.98%	98.33%	3.35	65.83
96	1988	1982	6	99.70%	0.30%	190848	188016	2832	95.99%	98.54%	2.55	66.39
97	1988	1977	11	99.45%	0.55%	192836	189993	2843	97.00%	98.92%	1.92	66.83
98	1988	1984	4	99.80%	0.20%	194824	191977	2847	98.01%	99.06%	1.05	67.43
99	1988	1976	12	99.40%	0.60%	196812	193953	2859	99.02%	99.48%	0.46	67.84
100	1937	1922	15	99.23%	0.77%	198749	195875	2874	100.00%	100.00%	0	68.15

3. OOT Result

OOT	# Records	# Goods	# Bads	Fraud Rate								
	166493	164107	2386	0.014330933								
Bin Statistics						Cumulative Statistics						
Population Bin %	# Records	# Goods	# Bads	% Goods	% Bads	Total # Records	Cumulative Goods	Cumulative Bads	Cumulative % Goods	Cumulative % Bads (FDR)	KS Score	FPR
1	1665	434	1231	26.07%	73.93%	1665	434	1231	0.26%	51.59%	51.33	0.35
2	1665	1589	76	95.44%	4.56%	3330	2023	1307	1.23%	54.78%	53.55	1.55
3	1665	1646	19	98.86%	1.14%	4995	3669	1326	2.24%	55.57%	53.33	2.77
4	1665	1650	15	99.10%	0.90%	6660	5319	1341	3.24%	56.20%	52.96	3.97
5	1665	1654	11	99.34%	0.66%	8325	6973	1352	4.25%	56.66%	52.41	5.16
6	1665	1660	5	99.70%	0.30%	9990	8633	1357	5.26%	56.87%	51.61	6.36
7	1665	1652	13	99.22%	0.78%	11655	10285	1370	6.27%	57.42%	51.15	7.51
8	1665	1645	20	98.80%	1.20%	13320	11930	1390	7.27%	58.26%	50.99	8.58
9	1665	1658	7	99.58%	0.42%	14985	13588	1397	8.28%	58.55%	50.27	9.73
10	1665	1647	18	98.92%	1.08%	16650	15235	1415	9.28%	59.30%	50.02	10.77
11	1665	1652	13	99.22%	0.78%	18315	16887	1428	10.29%	59.85%	49.56	11.83
12	1665	1656	9	99.46%	0.54%	19980	18543	1437	11.30%	60.23%	48.93	12.9
13	1665	1651	14	99.16%	0.84%	21645	20194	1451	12.31%	60.81%	48.5	13.92
14	1665	1656	9	99.46%	0.54%	23310	21850	1460	13.31%	61.19%	47.88	14.97
15	1665	1656	9	99.46%	0.54%	24975	23506	1469	14.32%	61.57%	47.25	16
16	1665	1648	17	98.98%	1.02%	26640	25154	1486	15.33%	62.28%	46.95	16.93
17	1665	1653	12	99.28%	0.72%	28305	26807	1498	16.34%	62.78%	46.44	17.9
18	1665	1652	13	99.22%	0.78%	29970	28459	1511	17.34%	63.33%	45.99	18.83
19	1665	1652	13	99.22%	0.78%	31635	30111	1524	18.35%	63.87%	45.52	19.76
20	1665	1651	14	99.16%	0.84%	33300	31762	1538	19.35%	64.46%	45.11	20.65

21	1665	1657	8	99.52%	0.48%	34965	33419	1546	20.36%	64.79%	44.43	21.62
22	1665	1655	10	99.40%	0.60%	36630	35074	1556	21.37%	65.21%	43.84	22.54
23	1665	1653	12	99.28%	0.72%	38295	36727	1568	22.38%	65.72%	43.34	23.42
24	1665	1652	13	99.22%	0.78%	39960	38379	1581	23.39%	66.26%	42.87	24.28
25	1665	1656	9	99.46%	0.54%	41625	40035	1590	24.40%	66.64%	42.24	25.18
26	1665	1654	11	99.34%	0.66%	43290	41689	1601	25.40%	67.10%	41.7	26.04
27	1665	1655	10	99.40%	0.60%	44955	43344	1611	26.41%	67.52%	41.11	26.91
28	1665	1659	6	99.64%	0.36%	46620	45003	1617	27.42%	67.77%	40.35	27.83
29	1665	1658	7	99.58%	0.42%	48285	46661	1624	28.43%	68.06%	39.63	28.73
30	1665	1655	10	99.40%	0.60%	49950	48316	1634	29.44%	68.48%	39.04	29.57
31	1665	1656	9	99.46%	0.54%	51615	49972	1643	30.45%	68.86%	38.41	30.42
32	1665	1656	9	99.46%	0.54%	53280	51628	1652	31.46%	69.24%	37.78	31.25
33	1665	1656	9	99.46%	0.54%	54945	53284	1661	32.47%	69.61%	37.14	32.08
34	1665	1655	10	99.40%	0.60%	56610	54939	1671	33.48%	70.03%	36.55	32.88
35	1665	1652	13	99.22%	0.78%	58275	56591	1684	34.48%	70.58%	36.1	33.61
36	1665	1654	11	99.34%	0.66%	59940	58245	1695	35.49%	71.04%	35.55	34.36
37	1665	1654	11	99.34%	0.66%	61605	59899	1706	36.50%	71.50%	35	35.11
38	1665	1658	7	99.58%	0.42%	63270	61557	1713	37.51%	71.79%	34.28	35.94
39	1665	1652	13	99.22%	0.78%	64935	63209	1726	38.52%	72.34%	33.82	36.62
40	1665	1653	12	99.28%	0.72%	66600	64862	1738	39.52%	72.84%	33.32	37.32
41	1665	1654	11	99.34%	0.66%	68265	66516	1749	40.53%	73.30%	32.77	38.03
42	1665	1650	15	99.10%	0.90%	69930	68166	1764	41.54%	73.93%	32.39	38.64
43	1665	1653	12	99.28%	0.72%	71595	69819	1776	42.54%	74.43%	31.89	39.31
44	1665	1660	5	99.70%	0.30%	73260	71479	1781	43.56%	74.64%	31.08	40.13
45	1665	1652	13	99.22%	0.78%	74925	73131	1794	44.56%	75.19%	30.63	40.76
46	1665	1657	8	99.52%	0.48%	76590	74788	1802	45.57%	75.52%	29.95	41.5
47	1665	1652	13	99.22%	0.78%	78255	76440	1815	46.58%	76.07%	29.49	42.12
48	1665	1661	4	99.76%	0.24%	79920	78101	1819	47.59%	76.24%	28.65	42.94
49	1665	1656	9	99.46%	0.54%	81585	79757	1828	48.60%	76.61%	28.01	43.63
50	1665	1654	11	99.34%	0.66%	83250	81411	1839	49.61%	77.07%	27.46	44.27

51	1665	1649	16	99.04%	0.96%	84915	83060	1855	50.61%	77.75%	27.14	44.78
52	1665	1655	10	99.40%	0.60%	86580	84715	1865	51.62%	78.16%	26.54	45.42
53	1665	1655	10	99.40%	0.60%	88245	86370	1875	52.63%	78.58%	25.95	46.06
54	1665	1656	9	99.46%	0.54%	89910	88026	1884	53.64%	78.96%	25.32	46.72
55	1665	1649	16	99.04%	0.96%	91575	89675	1900	54.64%	79.63%	24.99	47.2
56	1665	1661	4	99.76%	0.24%	93240	91336	1904	55.66%	79.80%	24.14	47.97
57	1665	1649	16	99.04%	0.96%	94905	92985	1920	56.66%	80.47%	23.81	48.43
58	1665	1654	11	99.34%	0.66%	96570	94639	1931	57.67%	80.93%	23.26	49.01
59	1665	1656	9	99.46%	0.54%	98235	96295	1940	58.68%	81.31%	22.63	49.64
60	1665	1660	5	99.70%	0.30%	99900	97955	1945	59.69%	81.52%	21.83	50.36
61	1665	1647	18	98.92%	1.08%	101565	99602	1963	60.69%	82.27%	21.58	50.74
62	1665	1659	6	99.64%	0.36%	103230	101261	1969	61.70%	82.52%	20.82	51.43
63	1665	1653	12	99.28%	0.72%	104895	102914	1981	62.71%	83.03%	20.32	51.95
64	1665	1652	13	99.22%	0.78%	106560	104566	1994	63.72%	83.57%	19.85	52.44
65	1665	1651	14	99.16%	0.84%	108225	106217	2008	64.72%	84.16%	19.44	52.9
66	1665	1658	7	99.58%	0.42%	109890	107875	2015	65.73%	84.45%	18.72	53.54
67	1665	1653	12	99.28%	0.72%	111555	109528	2027	66.74%	84.95%	18.21	54.03
68	1665	1650	15	99.10%	0.90%	113220	111178	2042	67.75%	85.58%	17.83	54.45
69	1665	1654	11	99.34%	0.66%	114885	112832	2053	68.76%	86.04%	17.28	54.96
70	1665	1653	12	99.28%	0.72%	116550	114485	2065	69.76%	86.55%	16.79	55.44
71	1665	1653	12	99.28%	0.72%	118215	116138	2077	70.77%	87.05%	16.28	55.92
72	1665	1653	12	99.28%	0.72%	119880	117791	2089	71.78%	87.55%	15.77	56.39
73	1665	1652	13	99.22%	0.78%	121545	119443	2102	72.78%	88.10%	15.32	56.82
74	1665	1655	10	99.40%	0.60%	123210	121098	2112	73.79%	88.52%	14.73	57.34
75	1665	1646	19	98.86%	1.14%	124875	122744	2131	74.80%	89.31%	14.51	57.6
76	1665	1651	14	99.16%	0.84%	126540	124395	2145	75.80%	89.90%	14.1	57.99
77	1665	1653	12	99.28%	0.72%	128205	126048	2157	76.81%	90.40%	13.59	58.44
78	1665	1655	10	99.40%	0.60%	129870	127703	2167	77.82%	90.82%	13	58.93
79	1665	1654	11	99.34%	0.66%	131535	129357	2178	78.82%	91.28%	12.46	59.39
80	1665	1651	14	99.16%	0.84%	133200	131008	2192	79.83%	91.87%	12.04	59.77

81	1665	1656	9	99.46%	0.54%	134865	132664	2201	80.84%	92.25%	11.41	60.27
82	1665	1650	15	99.10%	0.90%	136530	134314	2216	81.85%	92.88%	11.03	60.61
83	1665	1652	13	99.22%	0.78%	138195	135966	2229	82.85%	93.42%	10.57	61
84	1665	1655	10	99.40%	0.60%	139860	137621	2239	83.86%	93.84%	9.98	61.47
85	1665	1650	15	99.10%	0.90%	141525	139271	2254	84.87%	94.47%	9.6	61.79
86	1665	1656	9	99.46%	0.54%	143190	140927	2263	85.88%	94.84%	8.96	62.27
87	1665	1655	10	99.40%	0.60%	144855	142582	2273	86.88%	95.26%	8.38	62.73
88	1665	1658	7	99.58%	0.42%	146520	144240	2280	87.89%	95.56%	7.67	63.26
89	1665	1654	11	99.34%	0.66%	148185	145894	2291	88.90%	96.02%	7.12	63.68
90	1665	1658	7	99.58%	0.42%	149850	147552	2298	89.91%	96.31%	6.4	64.21
91	1665	1658	7	99.58%	0.42%	151515	149210	2305	90.92%	96.61%	5.69	64.73
92	1665	1655	10	99.40%	0.60%	153180	150865	2315	91.93%	97.02%	5.09	65.17
93	1665	1657	8	99.52%	0.48%	154845	152522	2323	92.94%	97.36%	4.42	65.66
94	1665	1656	9	99.46%	0.54%	156510	154178	2332	93.95%	97.74%	3.79	66.11
95	1665	1654	11	99.34%	0.66%	158175	155832	2343	94.96%	98.20%	3.24	66.51
96	1665	1658	7	99.58%	0.42%	159840	157490	2350	95.97%	98.49%	2.52	67.02
97	1665	1658	7	99.58%	0.42%	161505	159148	2357	96.98%	98.78%	1.8	67.52
98	1665	1655	10	99.40%	0.60%	163170	160803	2367	97.99%	99.20%	1.21	67.94
99	1665	1653	12	99.28%	0.72%	164835	162456	2379	98.99%	99.71%	0.72	68.29
100	1658	1651	7	99.58%	0.42%	166493	164107	2386	100.00%	100.00%	0	68.78