Home Credit Default Risk: Predicting which Clients can be Granted a Loan

PROJECT APPROACH:

The main aim of the project is prediction and classification of number of clients, who are eligible to receive a loan. Here I classified the people into two categories, one is people who are to be granted a loan and the other is the business who would not lend to give a loan. So, by this we can know the reliability and credibility of an applicant for the repayment of the loan.

The problem in particular strikes as difficult to address as this organization operates in an uncommon situation where, considering the poor to zero credit score, loans are to be issued to individuals that other banks would generally not consider. They are at a considerably higher financial risk because their market strategy relies on issuing loans to people that other banks would demonstrably deny. As a result, by mitigating at-risk clients while yielding income, the organization would save by using the dataset solutions to solve the problem at the best possible accuracy, comparatively better than they would earn without using the models.

Next, the data was cleaned and pre-processed in ways that made it suitable for use in various modeling approaches. For each possible modeling strategy, I decided to prepare each predictor. I went into an exploratory modeling process with the data usually prepared, where I tried to fit and tune at least two kinds of classification models using either all predictors or a subset of data that had near-zero variance and omitted strongly correlated predictors. This phase included using the predictive classification methods like K-means clustering algorithms, logistic regression analysis, or classification analysis such as decision trees to classify the population in no risk categories and at risk categories considering data of each individual such as low credit score, family income as variables of the model, credit card balance, previous applications.

I found predictive techniques that work with the data to identify the individuals that should be given the loan as compared to those that may pose a danger to the organization and should therefore be rejected.

Then there was time-consuming preprocessing performed to clean the dataset after testing and comparing a number of predictive model types. In accordance with the advantages and disadvantages of choosing each model in the sense of the project goals, the output outcomes of each model were then considered.

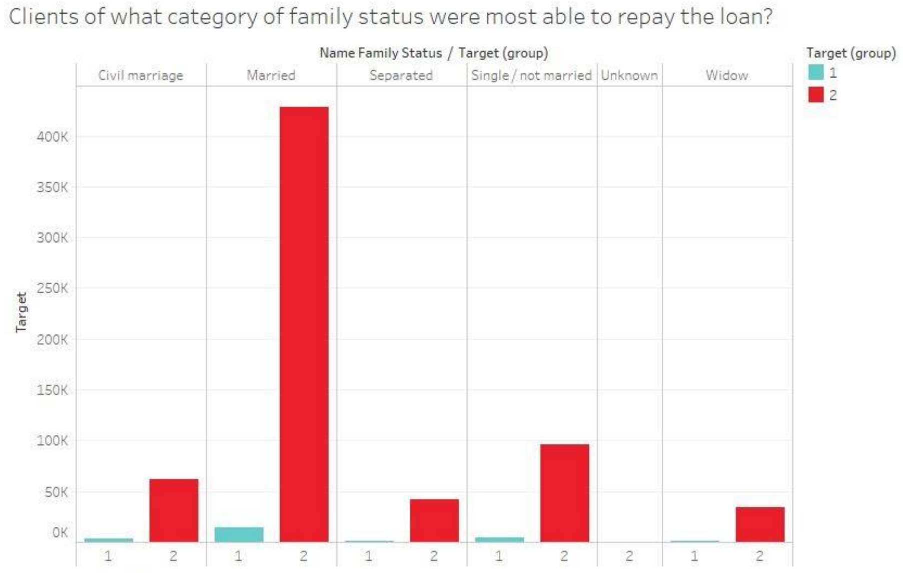
Dataset:

- The Home Credit Default Risk dataset is the dataset used for this research. (https:/www.kaggle.com/c/home-default-risk for credit). It was collected from the Home Credit Group, an international consumer finance company whose goal is to offer loans to individuals with a credit history that is very low to zero and without bank accounts.
- The dataset highlights the different customer information, including their financial background, from which we can find out about customers who might have trouble paying the loan and customers who have no problem paying the loan based on their history of data records, such as credit card balances, any prior home loan applications.
- About 350,000 customers and their information make up our dataset. From this dataset, we would predict the customers that will be able to repay the loan back, and those customers will not be able to repay a loan based on this dataset that was given to them. We estimated the number of customers who will be willing to pay back the loan based on the client's features in the dataset. This way, then the business would realize what kind of buyers, provided a loan, would be able to pay back.

Visualizing the Data:

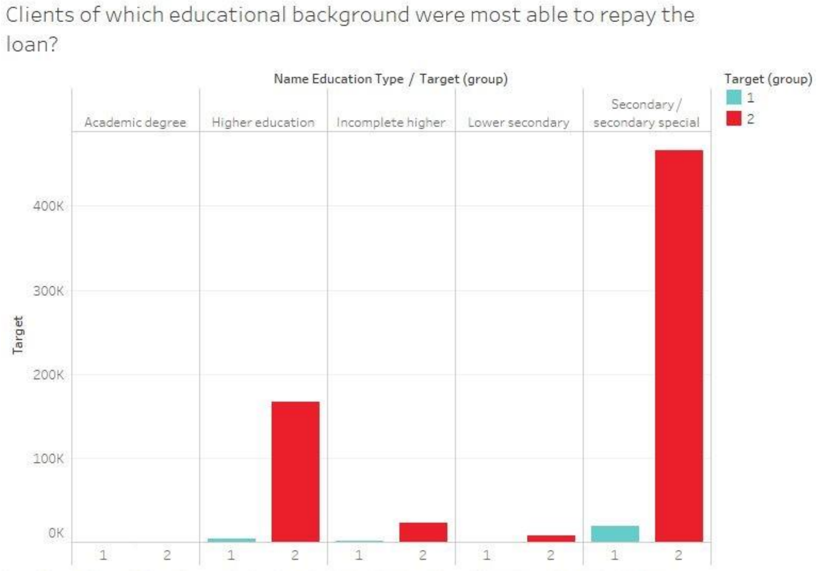
- To gain an understanding of the available data, the project was started with an exploratory data analysis process. This was accompanied by a concerted effort to clean and pre-process the information in ways that made it appropriate for use in a number of approaches to modeling.
- For visualizations of data items to seek connection with the "TARGET" data object, the whole dataset was used. In the visualizations, 1 represents the number of clients who were unable to repay the loan; 2 indicates the number of clients who were able to repay the loan without any problem.
- Here I performed some visualizations

Visual A: In this visualization I considered family status of the clients and predicted that married clients could back their loans, rather than the clients who were single or civil marriage, or if they are widow, or the status is unknown.

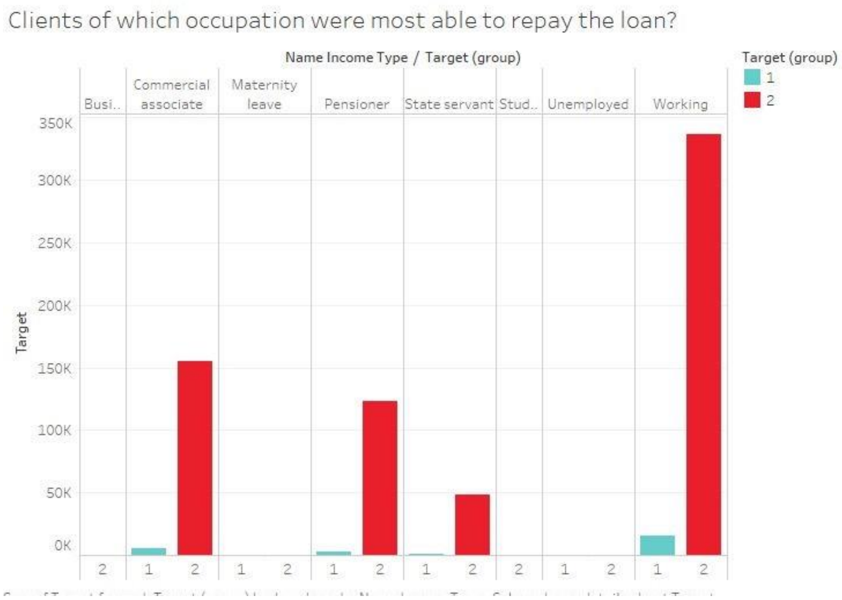


Sum of Target for each Target (group) broken down by Name Family Status. Colour shows details about Target (group).

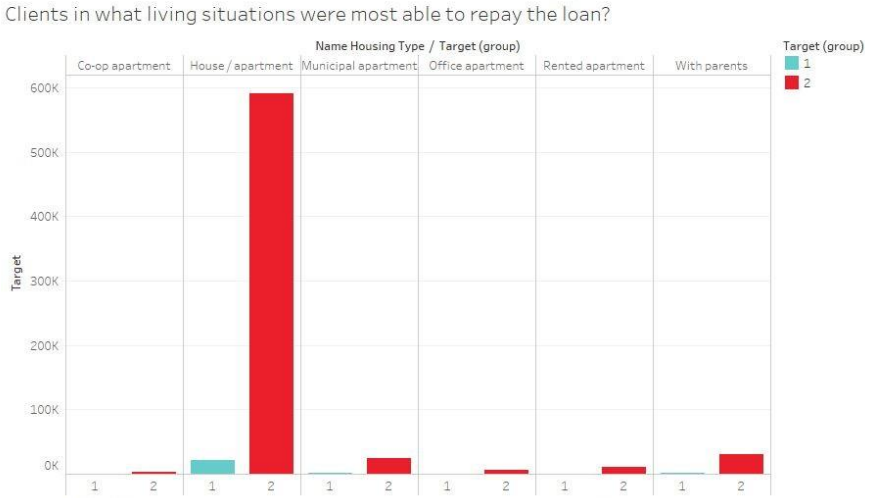
Visual B: In this visualization I considered type of education the clients had completed and predicted that clients who had completed the secondary education were granted most loans other than the people who had incomplete higher education and lower secondary education.



Visual C: In this visualization I considered the employment status of the clients and predicted that the working clients were granted most of the loans other than the clients who had business or commercial associates, ones on maternity leave, pensioners, state servants, students. People who are not employed would not get any loan.



Visual D: In this visualization I considered living status of the clients. The people who lived in house/apartment were granted loan, rather than the people who lived in co-op apartment, a municipal apartment, an office apartment or a rented apartment or who lived with parents.

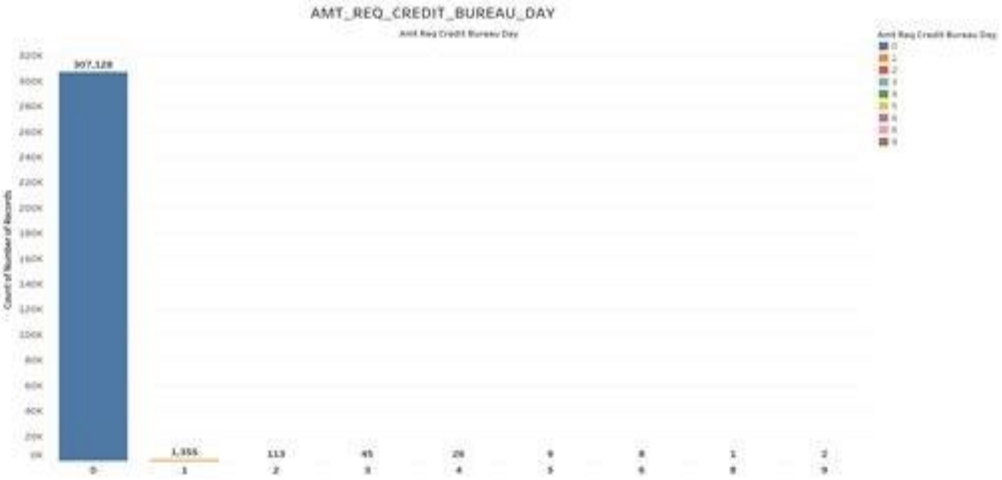


Preprocessing the Dataset:

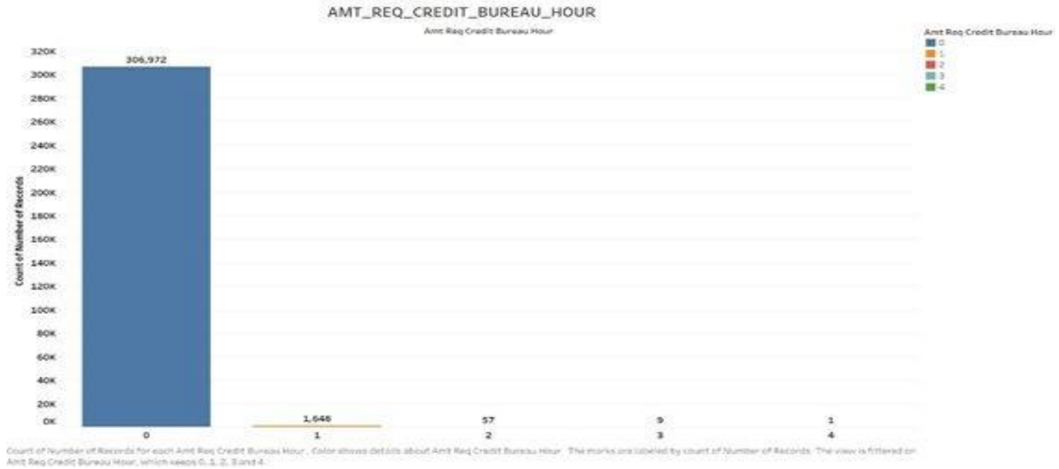
Step 1: EDA (Exploratory Data Analysis):

The dataset we had is already splitted into test and train datasets. So, now we combine both the datasets by using the outer joint.

Here we performed different EDA visualizations and based on the result the rest of the data is preprocessed in the following steps.

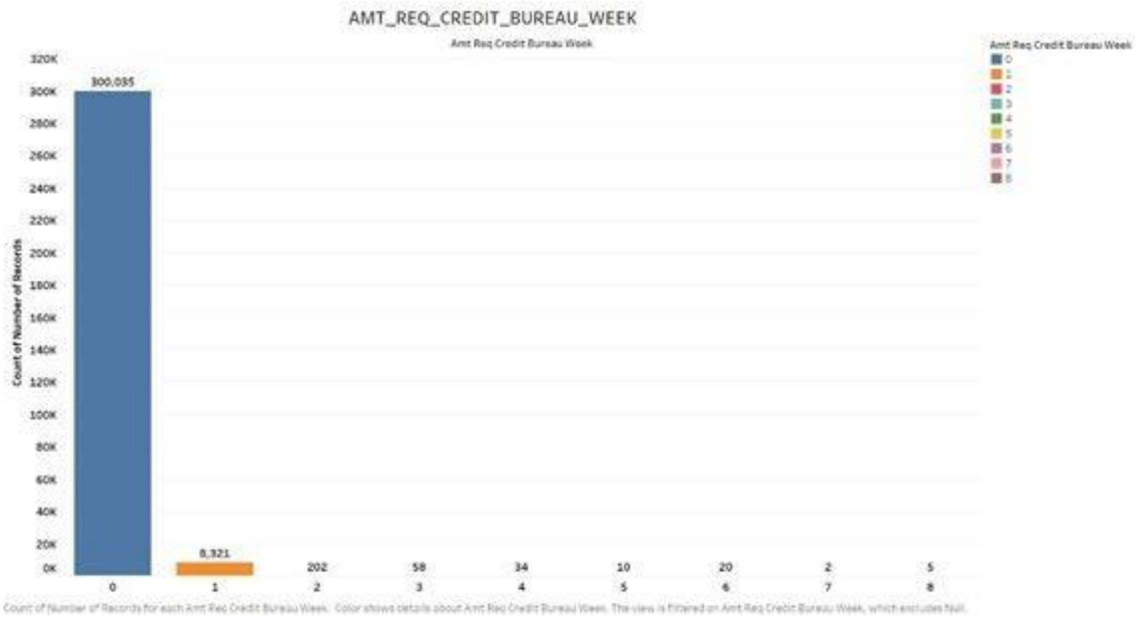


In the above visualization, I found that there were a lot of repeated small values in this column. Therefore, I substituted all the values into NaN that were greater than 2 and used the column as a categorical column.



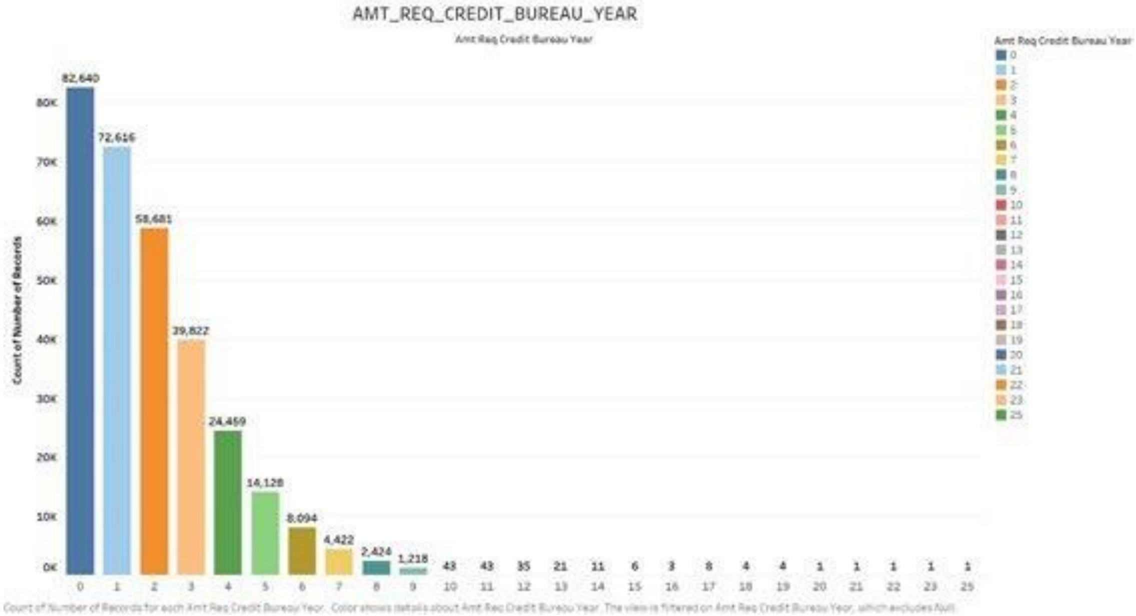
Count of Number of Records for each Amt Req Credit Bureau Hour. Color shows details about Amt Req Credit Bureau Hour. The marks are labeled by count of Number of Records. The view is filtered on Amt Req Credit Bureau Hour, which keeps 0, 1, 2, 3 and 4.

Same as the before visualization, I substituted all the values into NaN that were greater than 1, used it as a categorical column and imputed it with mode later



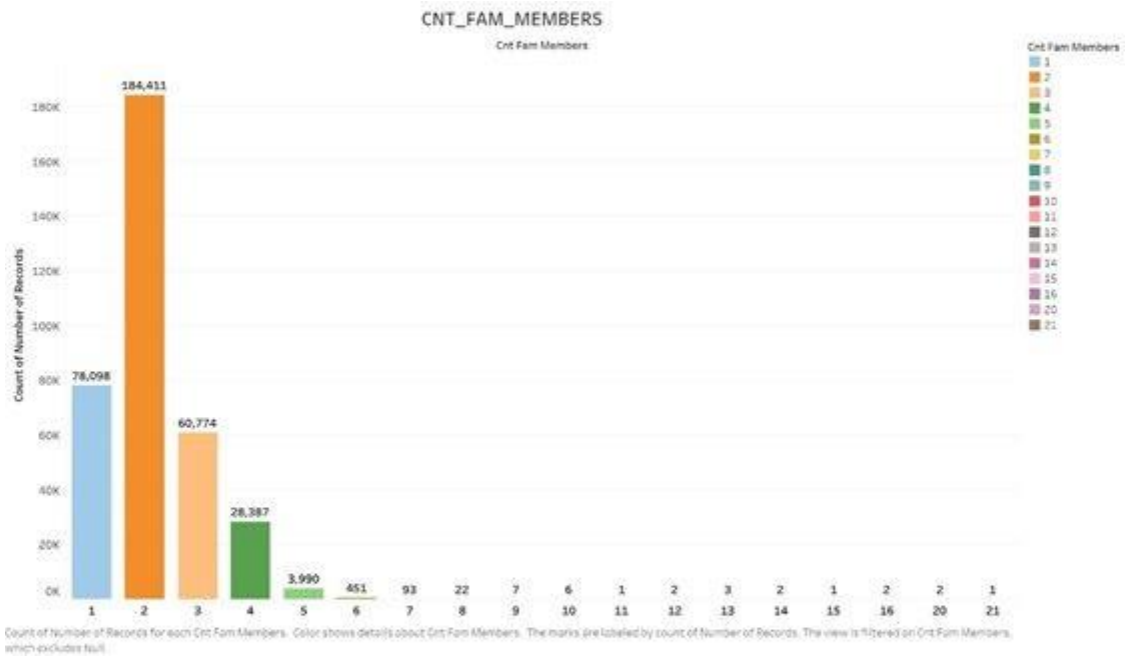
Count of Number of Records for each Amt Req Credit Bureau Week. Color shows details about Amt Req Credit Bureau Week. The view is filtered on Amt Req Credit Bureau Week, which excludes Null.

In the above visualization, I found very less number of values that are greater than 2, therefore I removed all of them and replaced them with Nan and later imputed it with mode.

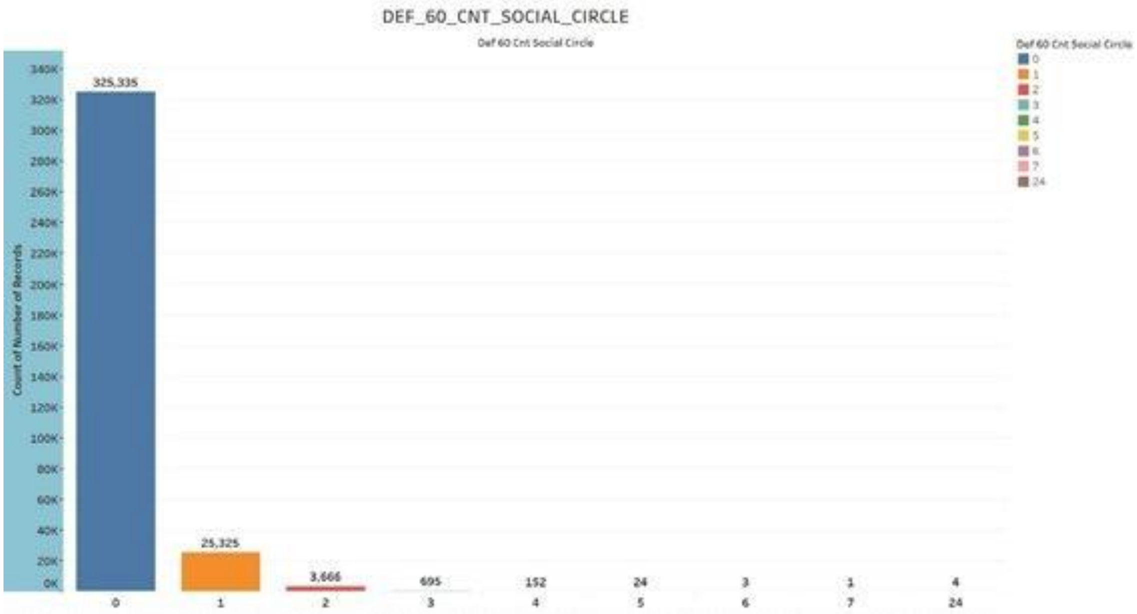


Count of Number of Records for each Amt Req Credit Bureau Year. Color shows details about Amt Req Credit Bureau Year. The view is filtered on Amt Req Credit Bureau Year, which excludes Null.

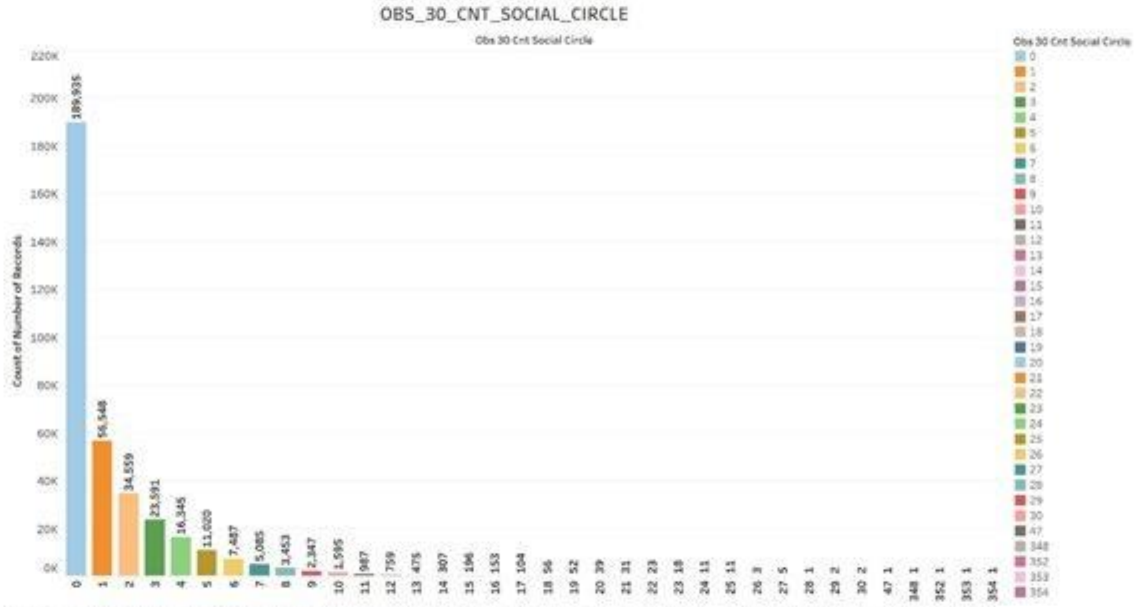
In the above visualization there are many values which are in very less amount in the column and the numbers from 0 to 9 are repeated several times. So, I removed all the values which are greater than 10 and later imputed it with model



In the above visualization, we can observe that the numbers before 5 are most repetitive than the other number which are after 5 are very less in number. So, I replaced all the numbers with Nan which are after 5 and later imputed with mode.

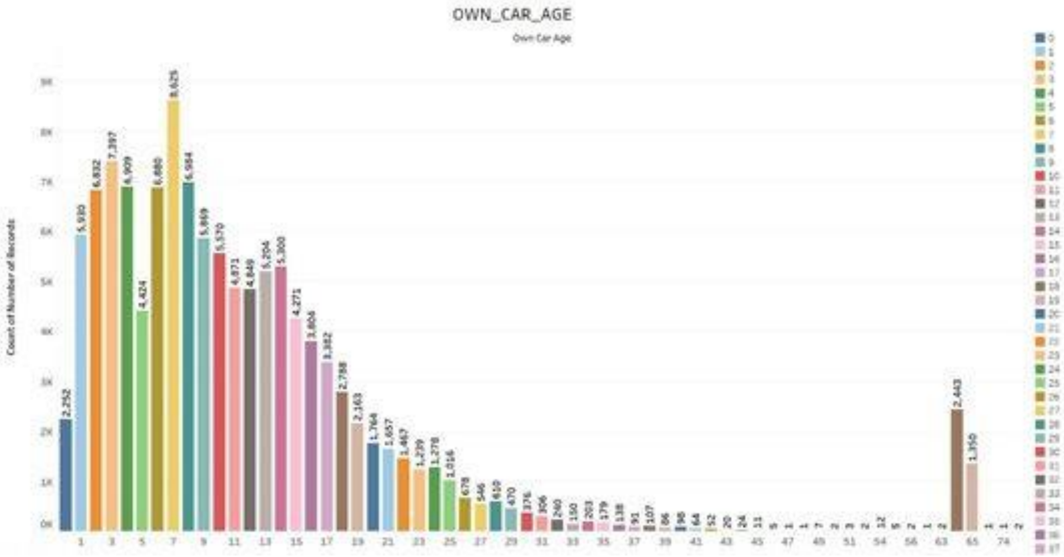


We can see in this visualization that numbers smaller than 4 have been repeated several times, so we replaced NaN with all numbers greater than 4, then later we replaced it with column mode.

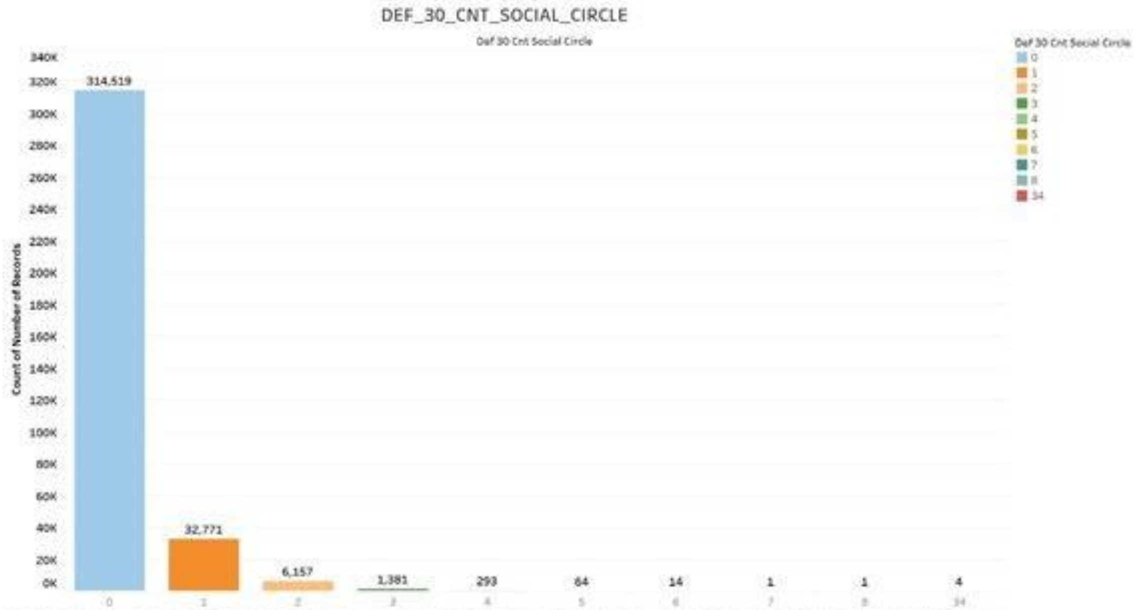


Count of Number of Records for each Obs 30 Cnt Social Circle. Color shows details about Obs 30 Cnt Social Circle. The marks are labeled by count of Number of Records. The view is filtered on Obs 30 Cnt Social Circle, which excludes Null.

We can see in this visual that the numbers below 17 have been replicated a lot of times and the numbers have a low count after that. Therefore, all the numbers that were greater than 17 have been deleted, replaced with NaN and then imputed with the mode to run the models.

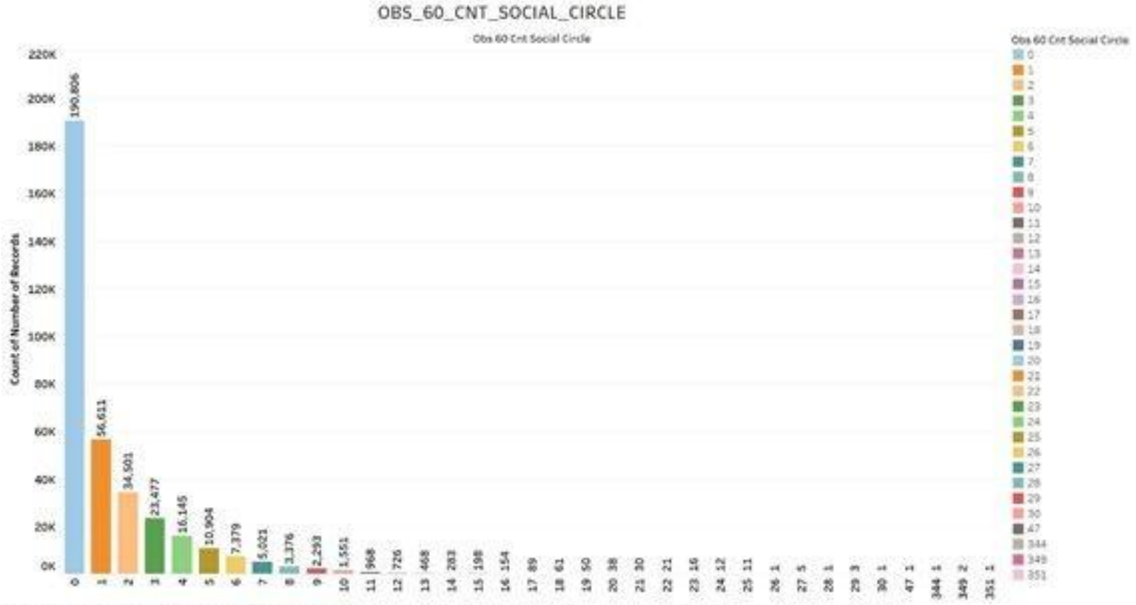


We can see in this visualization that numbers smaller than 60 have been replicated several times, so we replaced all numbers larger than 60 with NaN, then later we replaced them with column mode. Around 63 and 65, there were two outliers.



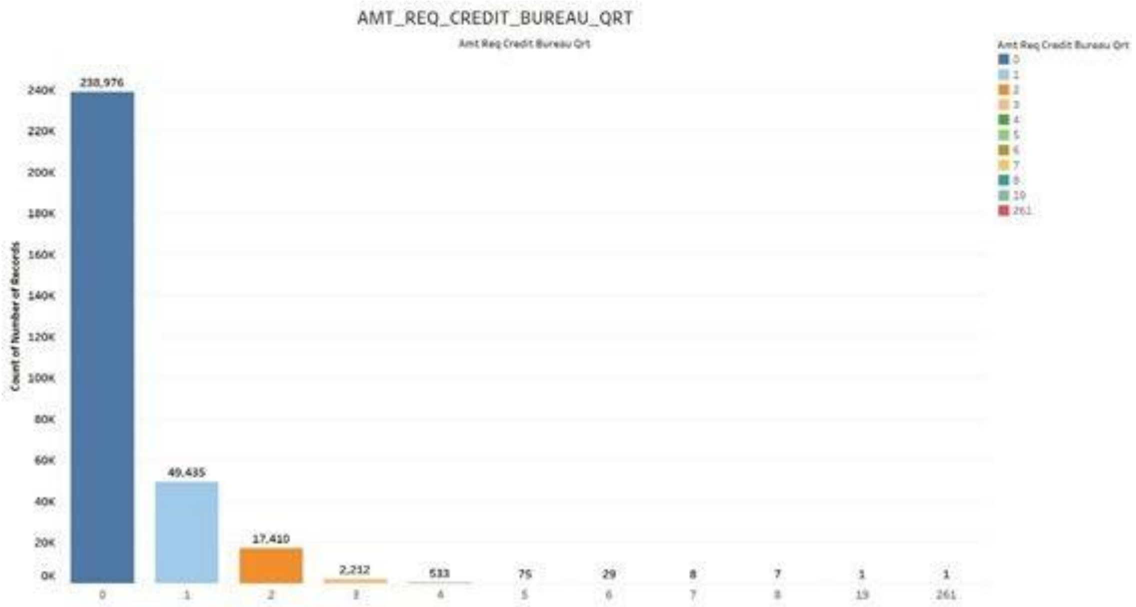
Count of Number of Records for each Def 30 Cnt Social Circle. Color shows details about Def 30 Cnt Social Circle. Details are shown for Def 30 Cnt Social Circle. The view is filtered on Def 30 Cnt Social Circle, which excludes Null.

The values greater than 5 are replicated many times in this visualization and those less than 5 have very few numbers. Therefore, I substituted NaN for all the values greater than 5 and then imputed the NaN values with the mode.



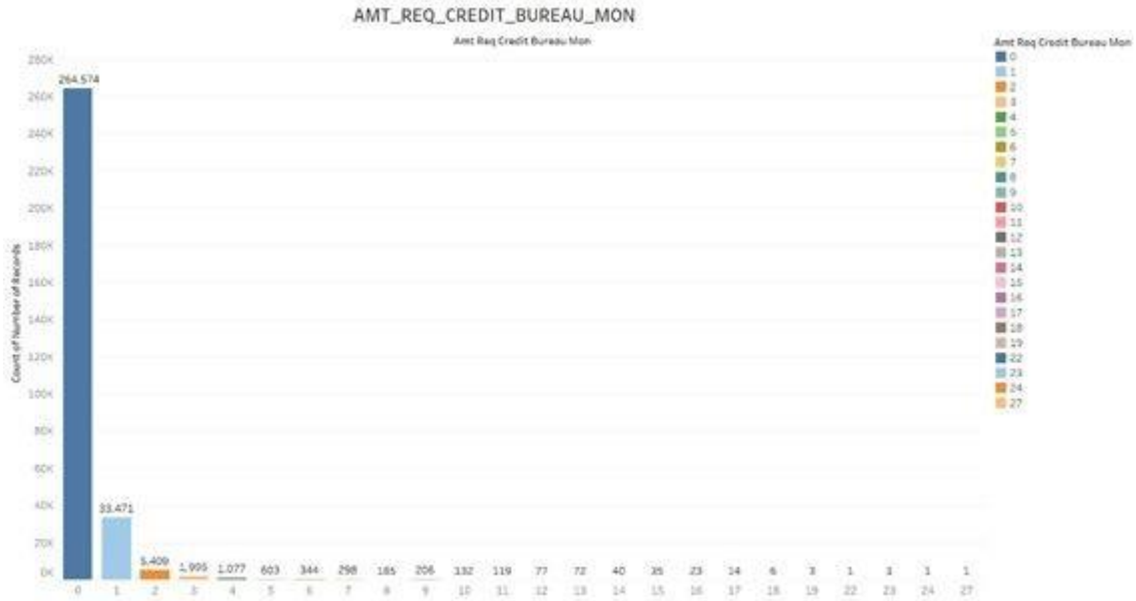
Count of Number of Records for each Obs 60 Cnt Social Circle. Color shows details about Obs 60 Cnt Social Circle. The view is filtered on Obs 60 Cnt Social Circle, which excludes Null.

The values greater than 16 are replicated several times in this visualization and those less than 16 have very few numbers. Therefore, I substituted NaN for all the values greater than 16 and then imputed the NaN values with the mode.



Count of Number of Records for each Amt Req Credit Bureau Qrt. Color shows details about Amt Req Credit Bureau Qrt. The view is filtered on Amt Req Credit Bureau Qrt, which excludes Null.

The values greater than 5 are replicated many times in this visualization and those less than 5 have very few numbers. Therefore, I substituted NaN for all the values greater than 5 and then imputed the NaN values and then imputed with the mode.



Count of Number of Records for each Amt Req Credit Bureau Mon. Color shows details about Amt Req Credit Bureau Mon. The view is filtered on Amt Req Credit Bureau Mon, which excludes Null.

We can see in this visual that the numbers below 5 have been replicated a lot of times and the numbers have a low count after that. Therefore, all the numbers that were greater than 5 have been deleted, replaced with NaN and then imputed with the mode to run the models.

Step 2: Dealing with the numerical columns

- I first picked all the columns of int64 and evaluated each column one by one. In a few columns like (DAYS-BIRTH, DAYS-EMPLOYED, DAYS-ID-PUBLISH), I found negative values. So, I converted these columns to positive and to years by dividing these columns by (-365).
- I also dealt with columns like observed social circle in the past 30 days, Amount required credit bureau hours spent etc of outliers with mode of the remaining values.
- Then after I started working on float 64 columns, here I did the same operations as before and to prevent the risk of skewing I also replaced few values based on their count.

Step 3: Dealing with the categorical columns

- I selected all the categorical columns, i.e. those of the object data type, and I checked and replaced the few 'XNA' gibberish/nonsensical values and the unattended values with NaN for each column.
- I applied the label encoder after completing the above for a few handpicked columns such as (NAME CONTRACT Form, CODE GENDER, FLAG OWN CAR AND FLAG OWN REALTY) with only two levels that translated the values to numbers in each of these columns.
- I applied hot encoder to all the categorical columns which have more than 2 levels of records. If an attribute has more than 2 categories and if label encoding is performed, then everything is encoded as integers and compared them based on their numerical value. One hot encoding for the variables having more than 2 categories.

Step 4: Dealing with missing values

- Now, in this section by printing the percentage of NaN values in the whole list of columns, we can remove those columns which has more then 45% of NaN values. These are the columns which has the high percentages of NA values.
- After that by replacing the NA values with either the mode or the mean depending on their form, such as using mode for categorical attributes and mean for numerical attributes, we addressed the remaining columns.
- In order to convert the data types of the remaining columns to numerical or categorical based on their precise nature from the above exploration, I then searched for column descriptions.

Step 5: Feature selection

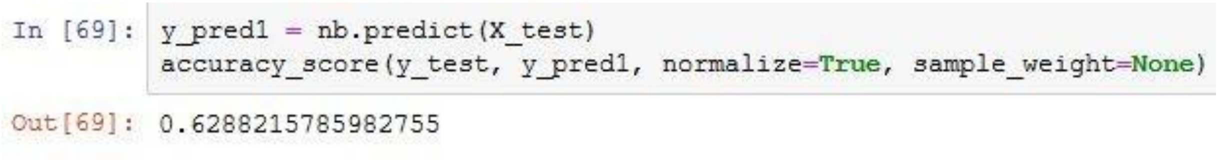
- One of the core aspects of the project was to find the best choice of features to maximize the model's accuracy.
- In order to improve the observations for the model before the feature selection, I added some new variables to the dataset based on the financial information that I had and generated some columns that could be relevant for our modeling. For example, by dividing the Amount Credit and the DAYS BIRTH, I created a variable called CREDIT AGE in the train collection.
- Then I ignored the NaN value rows of data to target columns.
- To get feature significance for both categorical and numerical columns separately, i used Kendall Density Methods. And then to run our model, I only used those attributes.
- To find the best model classification was performed on the training data.
- The difference in the ratios of target variables also caused the accuracy of low AUC values to decline. We therefore upsampled the training dataset for the final dataset again In the next step, we split the training data on the train and test to convert all train and test upsampled datasets using standard scalar and PCA.

MODEL SELECTION

- This is a problem of classification where the conclusions are derived from values observed. As the dataset was cleaned and preprocessed above the best classification algorithms were explored for different accuracies in the first step.
- At first, we used usability filtering techniques, with PCA fit and upsampled data to carry out the classification using Logistic Regression and Naive Bayes, resulting in 67 percent low accuracy rate.
- Secondly, we used filtered data directly without PCA fit and no upsampling to classify H2o generalized regression model to obtain increased accuracies.
- Elimination of 78 variables from 201 columns was caused by the data cleaning process and exploratory research performed before. This resulted in 123 columns, including the output Target variable.
- The final model with the highest precision was then used to estimate the probabilities of the target variable's missing values.

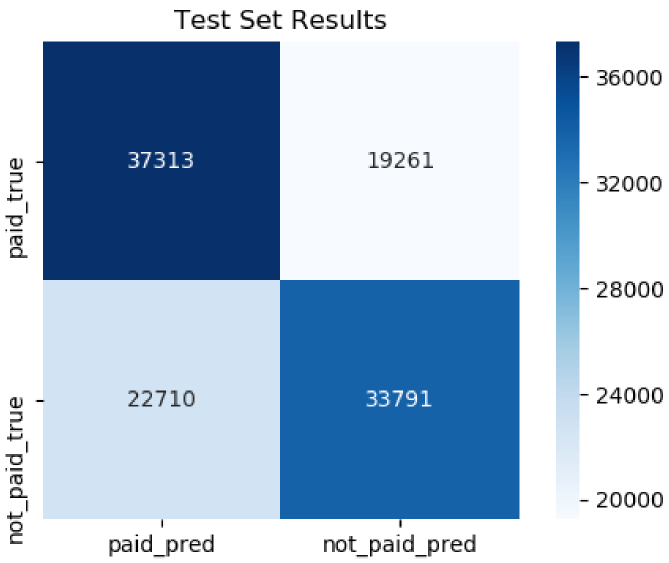
Gaussian Naive Bayes:

- I used the model fit on the data, which was divided into train and test, using the Gaussian Naive Bayes' classifier.
- The accuracy by this algorithm resulted in 62.88% accurate.

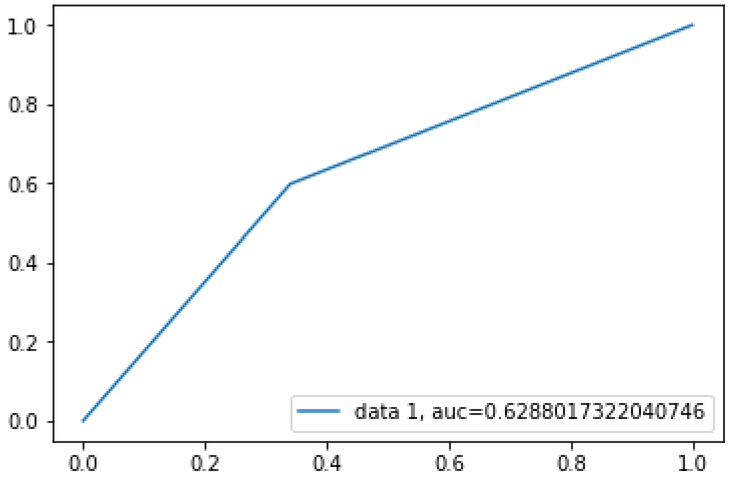


```
In [69]: y_pred1 = nb.predict(X_test)
         accuracy_score(y_test, y_pred1, normalize=True, sample_weight=None)

Out[69]: 0.6288215785982755
```

- The confusion matrix indicates that the number of clients who were correctly classified as those who could repay the loan back and the clients who were correctly classified as one who would not be able to repay the loan back, vs. the clients who would be able to repay the loan wrongly predicted and the clients who would not be able to repay the loan wrongly predicted.
- There were 37313 people who were classified correctly for being able to repay the loans and 33791 people who were correctly classified for not being able to repay the loan.

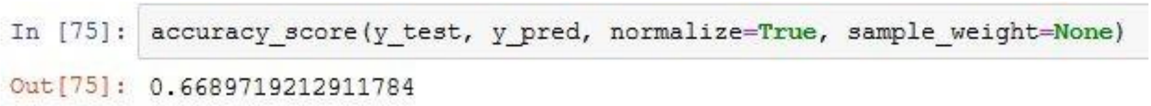


- In the Naive Bayes' Classifier, the AUC curve scored 0.6288 accuracy. This means that the degree to which the model can differentiate between the classes in order to properly classify them is 0.6288. The AUC curve results with Gaussian Naive Bayes' Algorithm.



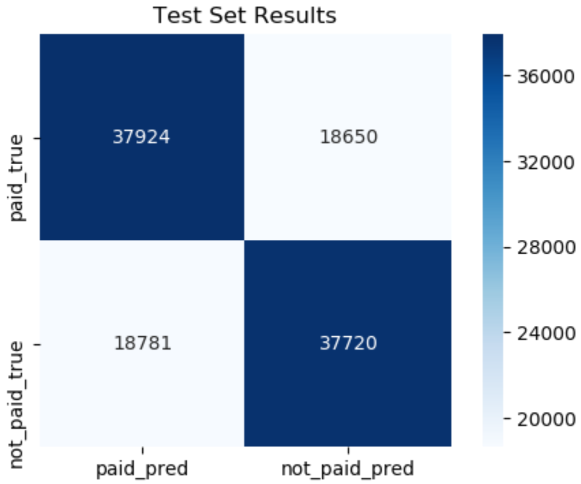
Logistic Regression:

- Logistic Regression is highly interpretable and has no parameters for tuning. So to divide into train and test, the upsampled and filtered data was fitted with PCA.
- By using the 'from sklearn.linear model import LogisticRegression' python library, the dataset was equipped with logistic regression in python.
- With the above, we got an accuracy of 66.89% in predicting the values.

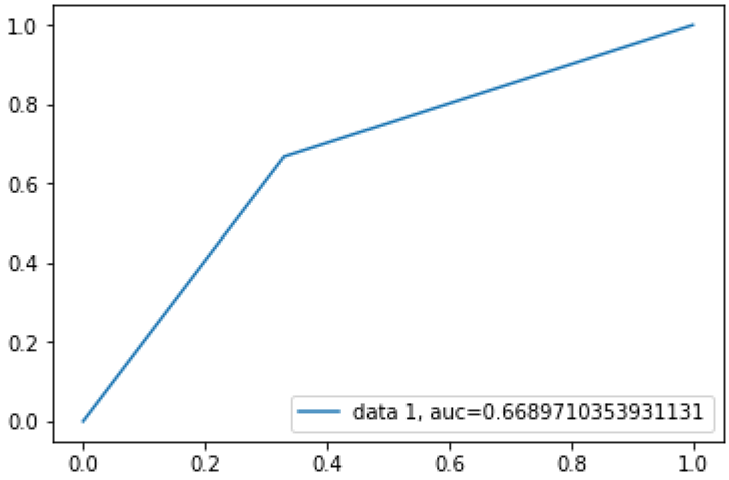


```
In [75]:  accuracy_score(y_test, y_pred, normalize=True, sample_weight=None)

Out[75]:  0.6689719212911784
```

- The confusion matrix of the logistic regression model is as shown above, where 37924 people were predicted correctly for being able to repay the loans and 37720 people were predicted correctly for "Not able to pay the loan".

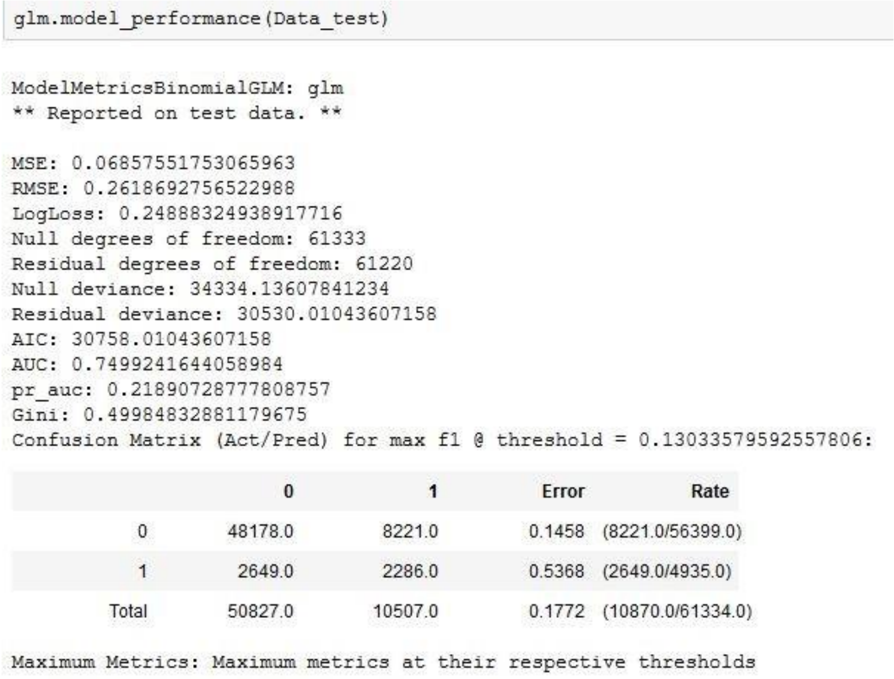


- The AUC curve has a score of 0.6689 highlights the goodness of the logistic regression classifier above.



H2O's Generalized Linear Model Estimator:

- The data was fitted to a maximum likelihood model using the Generalized Linear Estimator with H2o. For each of the predicted probabilities, this model results in probabilities that are more likely to occur.
- From the 'h2o.estimators.glm import H2OGeneralizedLinearEstimator' package, the original dataset that was only filtered without any PCA fit or upsampling was fitted to the H2o generalized linear model estimator in python.
- The results of the above models were then converted to the ranges of the listed models.
- With the above model, we have 82.86 accuracy.
- The above model's confusion matrix correctly estimated 48178 individuals able to pay the loan and 2286 individuals who were not able to pay the loan correctly.
- The AUC from the results for the above model was 0.7499.



```
glm.model_performance(Data_test)
```

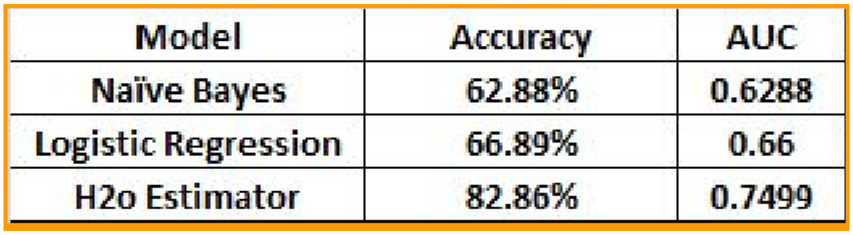
```
ModelMetricsBinomialGLM: glm
** Reported on test data. **

MSE: 0.06857551753065963
RMSE: 0.2618692756522988
LogLoss: 0.24888324938917716
Null degrees of freedom: 61333
Residual degrees of freedom: 61220
Null deviance: 34334.13607841234
Residual deviance: 30530.01043607158
AIC: 30758.01043607158
AUC: 0.7499241644058984
pr_auc: 0.21890728777808757
Gini: 0.49984832881179675
Confusion Matrix (Act/Pred) for max f1 @ threshold = 0.13033579592557806:
```

	0	1	Error	Rate
0	48178.0	8221.0	0.1458	(8221.0/56399.0)
1	2649.0	2286.0	0.5368	(2649.0/4935.0)
Total	50827.0	10507.0	0.1772	(10870.0/61334.0)

```
Maximum Metrics: Maximum metrics at their respective thresholds
```

Conclusion:

From the above results, it was obvious that the H2o generalized linear model estimator was the best classifier in the model list.



Model	Accuracy	AUC
Naïve Bayes	62.88%	0.6288
Logistic Regression	66.89%	0.66
H2o Estimator	82.86%	0.7499

References:

Home Credit Group. Home Credit Default Risk. Retrieved May 5, 2020, from

https://www.kaggle.com/c/home-credit-default-risk

Home Credit Group. (2019). Home Credit : Complete EDA + Feature Importance. Retrieved from

https://www.kaggle.com/codename007/home-credit-complete-eda-feature-importance#5-2

Home Credit Group.(2018). Home Credit Default Risk - Exploration + Baseline Model. Retrieved from

https://www.kaggle.com/shivamb/homecreditrisk-extensive-eda-baseline-0-772

Rahul Agarwal. July 27,2019. The 5 Feature Selection Algorithms every Data Scientist should know. Retrieved from

https://towardsdatascience.com/the-5-feature-selection-algorithms-every-data-scientist-need-to-know-3a6b566efd2

Home Credit Group. (2019). Home Credit : Complete EDA + Feature Importance. Retrieved from

https://www.kaggle.com/codename007/home-credit-complete-eda-feature-importance#5-2

Home Credit Group.(2018). Home Credit Default Risk - Exploration + Baseline Model. Retrieved from

https://www.kaggle.com/shivamb/homecreditrisk-extensive-eda-baseline-0-772

Rahul Agarwal. July 27,2019. The 5 Feature Selection Algorithms every Data Scientist should know. Retrieved from

https://towardsdatascience.com/the-5-feature-selection-algorithms-every-data-scientist-need-to-know-3a6b566efd2