

**CAPSTONE PROJECT - FINAL REPORT**

**INTEREST RATE**

**PREDICTION**

****

**MENTOR SUBMITTED BY**

**Prof. Vidhya K Tejas Ayyar**

**Raj Panchal**

**Vaidehi Tare**

**Ruchika Gajeshwar**

**Lalit Kacha**

**PROBLEM DEFINITION**

Banks and lending firms offer various kinds of accounts and provide loans based on the requirements. Apart from it, there are other various activities like investments in market and different funds. Overall, the banking sector has a wide impact on the economy directly and indirectly.

There are many banks across the globe that are leveraging machine learning and AI in their daily routine and getting benefits out of it.

For example, top banks in the US like JPMorgan, Wells Fargo, Bank of America, City Bank and US banks are already using machine learning to provide various facilities to customers as well as for risk prevention and detection.

We know that banks have massive overheads, with thousands of employees to pay and hundreds of branches to maintain. To maintain profitability, banks must take large margins on the money that passes through them. Earning out of the difference in interests (what it pays to depositors and what it charges from borrowers) is the main source of revenue for any bank and has been the key element in the functioning of all traditional financial institutions.

We can use strategies which will permit these clubs maximise profits and minimize the risks involved in the sector.

**TECHNOLOGY USED**

Financial firms like banks and lending groups are generally free to determine the interest rate they will pay for deposits and charge for loans, but they must take the competition into account, as well as the market levels for numerous interest rates and policies.

There are many types of interest rates and loan products. When it comes to setting rates, certain loans, such as residential home mortgage loans, may not be based on the prime rate fixed by the firm but rather according to the policies set by the government.

These firms use an array of factors to set interest rates mainly to maximize profits for their shareholders. On the flip side, consumers and businesses seek the lowest rate possible. They start from client inputs, such as credit score, collateral provided, down payment and duration for the loan, employment status, assets owned and so on to calculate the optimum interest rates. **PROBLEM IMPORTANCE**



The data for the problem is an example of Peer-to-peer lending (or P2P lending) Club which is one of the most innovative financial products of recent times. It enables creditworthy borrowers lower their cost of loans and individual lenders/investors to lend directly to their peers and community thereby earning higher returns.

Lending clubs provide a virtual market place where borrowers and lenders can interact directly, without having to go through the traditional financial intermediaries like banks, who have become such behemoths in today’s time that they dictate all terms and conditions for both borrowers and lenders.

The project will use machine learning algorithms that leverage different determining factors of a loan applicant. Selection of significant factors will help develop a prediction algorithm which can estimate loan interest rates based on clients’ information. On one hand, knowing the factors will help consumers and borrowers to increase their credit worthiness and place themselves in a better position to negotiate for getting a lower interest rate. On the other hand, this will help lending companies to get an immediate fixed interest rate estimation based on client’s information. Here, our goal is to use a training dataset to predict the loan rate category (1 / 2 / 3) that will be assigned to each loan in our test set. We will use combination of the features in the dataset to make our loan rate category predictions.

**SUGGESTED SOLUTION**

So our problem is dealing with the identifying the customer according to their loan dispensing category in other words we can say that we have a to assign a label to the customer given various attributes. To overcome this problem we will be applying Supervise learning classification ML models on the given dataset. We'll be going through various steps for that as in the data set there our missing values so we'll be imputing them , we have to develop new features as well , compare the accuracy among different models and should come to conclusive model which will be predicting the best and giving a good accuracy.

***Week 1***

Exploratory Data Analysis (handling missing values, Outlier Treatment, Visualization)

***Week 2***

Feature Engineering, Statistical Analysis

and Feature Selection

***Week 3***

Creating a base model and evaluating it. Also, testing with other models

***Week 4***

Hyperparameter tuning to further increase the accuracy and build a better model

**DATASET DICTIONARY**

This dataset was provided by Kaggle for a Finance sector problem. The dataset had 164309 Rows and 14

columns in csv format. The data comprises of different features pertaining to various factors of every

customer applying for loan.

|  |  |
| --- | --- |
| **Variable** | **Definition** |
| Loan\_ID | Unique loan account id |
| Loan\_Amount\_Requested | Sanction loan amount applied for by the borrower's |
| Length\_Employed | Total tenure of employment in years |
| Home\_Owner | Ownership status of the current residing property. Values are Rent, Own, Mortgage, Other |
| Annual\_Income | Annual income of the borrower's as mentioned at the time of loan application |
| Income\_Verified | Indicates verification status of borrower's income. Values are income verified, not verified, or if the income source was verified |
| Purpose\_Of\_Loan | Indicated what is the utilization purpose of taking the loan |
| Debt\_To\_Income | A ratio calculated using the borrower’s total monthly debt payments on the total debt obligations, excluding mortgage and the requested loan, divided by the borrower’s self-reported monthly income |
| Inquiries\_Last\_6Mo | The number of inquiries by creditors on his Credit file during the past 6 months |
| Months\_Since\_Deliquency | The number of months passed since the borrower's last delinquency towards his EMI obligations |
| Number\_Open\_Accounts | The number of open credit lines in the borrower's credit file |
| Total\_Accounts | The total number of credit lines currently in the borrower's credit file (differs from above as in it also includes closed credit lines) |
| Gender | Gender of the borrower's |
| Interest\_Rate | Target Variable: Interest Rate category (1/2/3) of the loan application |

Out of all the columns available with us above, let us further see the categorization of these columns.

|  |  |
| --- | --- |
| **Numerical** | **Categorical** |
| Loan\_Amount\_Requested | Home\_Owner |
| Length\_Employed | Income\_Verified |
| Annual\_Income | Purpose\_Of\_Loan |
| Debt\_To\_Income | Gender |
| Inquiries\_Last\_6Mo | Interest\_Rate |
| Months\_Since\_Deliquency |  |
| Number\_Open\_Accounts |  |
| Total\_Accounts |  |

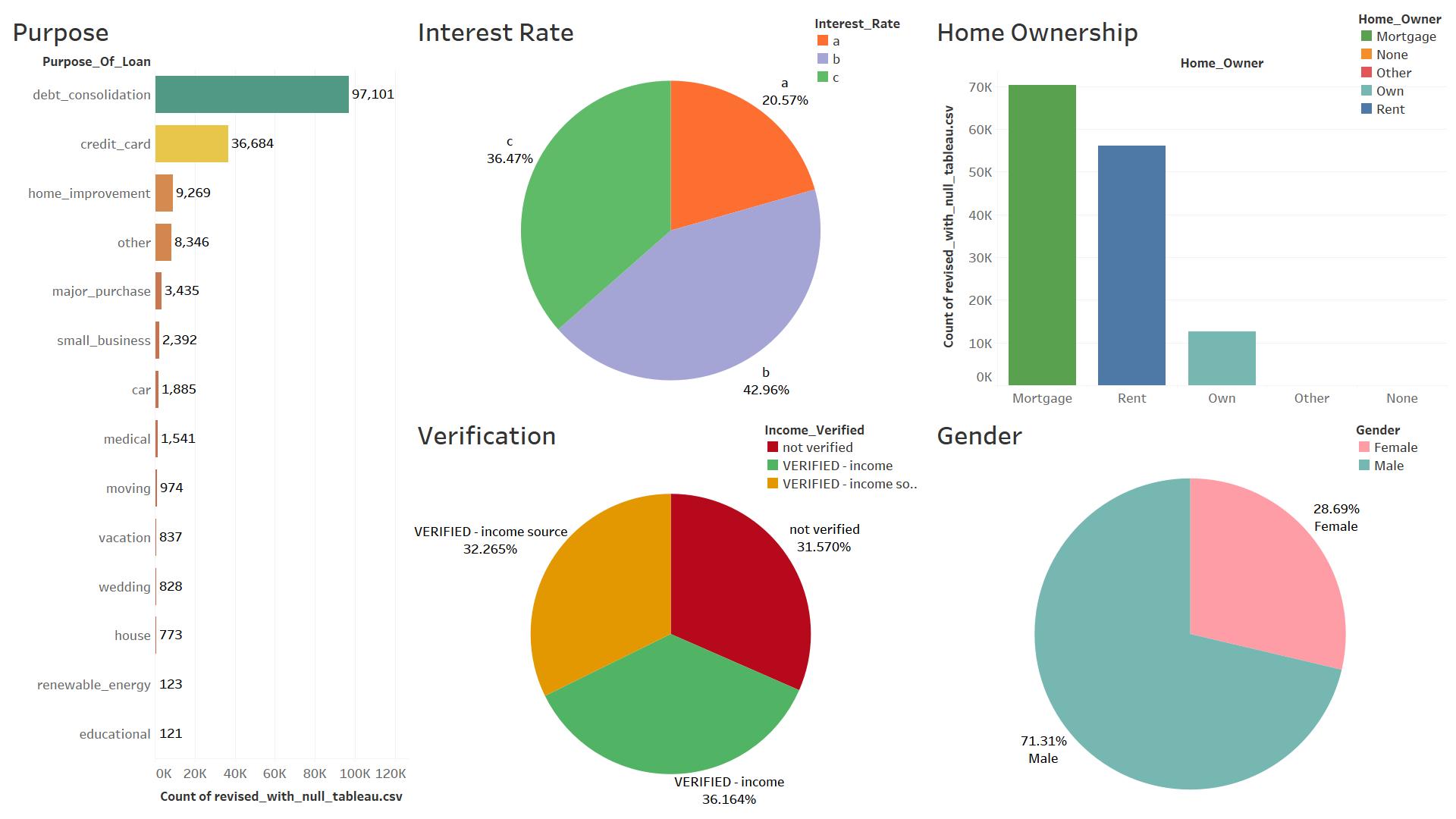
**TOTAL NUMERICAL COLUMNS = 8**

**TOTAL CATEGORICAL COLUMNS = 5**

Let us see the distribution of Missing Values across all the columns and the total percentage of Missing values in the dataset.

|  |  |
| --- | --- |
| **COLUMN NAME** | **COUNT OF MISSING VALUES** |
| Length\_Employed | 7371 |
| Home\_Owner | 25349 |
| Annual\_Income | 25102 |
| Months\_Since\_Deliquency | 88379 |

**VISUALISATIONS**



**The following conclusions can be derived from the above visualizations**

* *The maximum values in the* ***‘Loan Purpose’*** *column for which the loan amount is requested has been for debt consolidation**followed by credit card debts. However, the difference between the two is quite significant.*
* *The distribution of the Target column* ***‘Interest Rate’*** *is imbalanced, having interest rate 2, 3, 1 respectively in descending order.*
* *In the* ***‘Home Ownership’*** *column, it is observed that most loans are from clients having mortgage over homes, followed by rented and none. Only a small proportion of clients who have their own homes apply for loan.*
* *Data in* ***‘Verification’*** *column is almost* *balanced distribution of verified income, verified source and not verified categories.*
* We can also infer that maximum number of clients belong to the verified income category followed by verified income source and no verification with very less difference among the latter two.
* *It is observed that the gender ratio difference is maximum in class 1 rate of interest followed by class 2 and class 3 in the* ***‘Gender Distribution’*** *column.*

**DATA CLEANING**

While performing EDA the following discrepancies were found in the dataset:

1. Loan\_ID

This feature is a unique id column that does not provide any type of insight not would help the machine learning model in prediction.

1. Loan\_Amount\_Requested

This is a feature that should be numerical but is object because of the commas present between

the digits.

1. Length\_Employed

This feature has special characters and strings instead of numerical values.

**MISSING VALUE IMPUTATION**

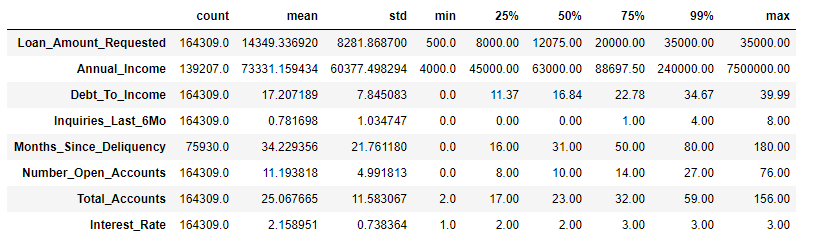
The dataset has a total of 8 % missing values. We cannot drop the missing values as it will result in data loss. We need to impute the missing values with suitable data.

We have used the fillna( ) function and imputed the missing values as explained below.

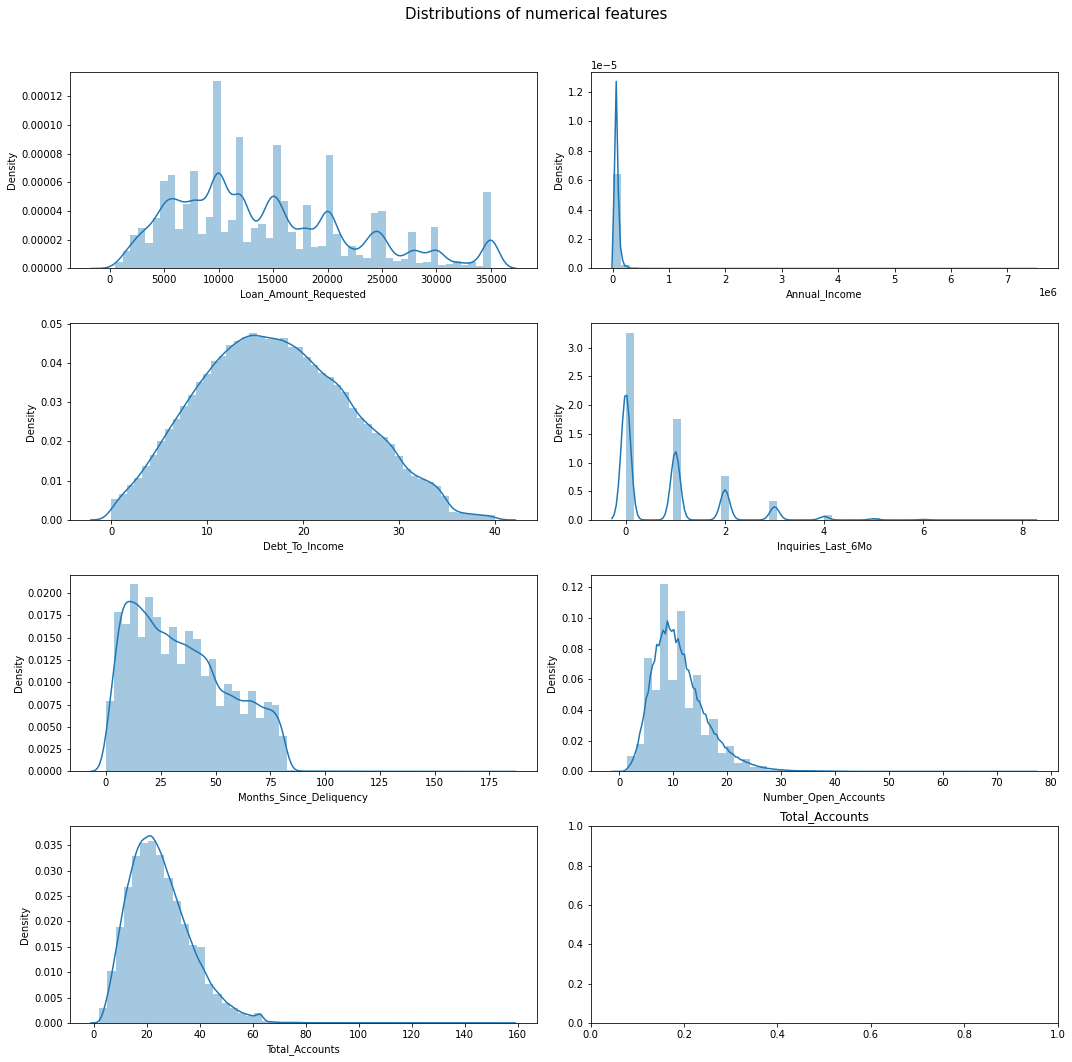
* **‘Month\_employed’** column has been filled with mode value since only 0.04% data was missing.
* **‘Home\_owner’** column has been filled with mode value of ‘Home\_owner’ column for each employment level to have more variation.
* **‘Annual\_Income’** has been filled using KNN Imputer with other numerical features only.
* We have not yet decided on **‘Delinquency’** column. May either drop or keep since it did not show any major impact in the boxplot.

**DATA DISTRIBUTION**

Using data.describe() to create a summary of the data to get a better understanding of the numerical features in the dataset.



**Using Distribution plot in python to visualise data distribution**



**PROPORTION PLOT**

****

###### **Inferences from the Distribution Plots**

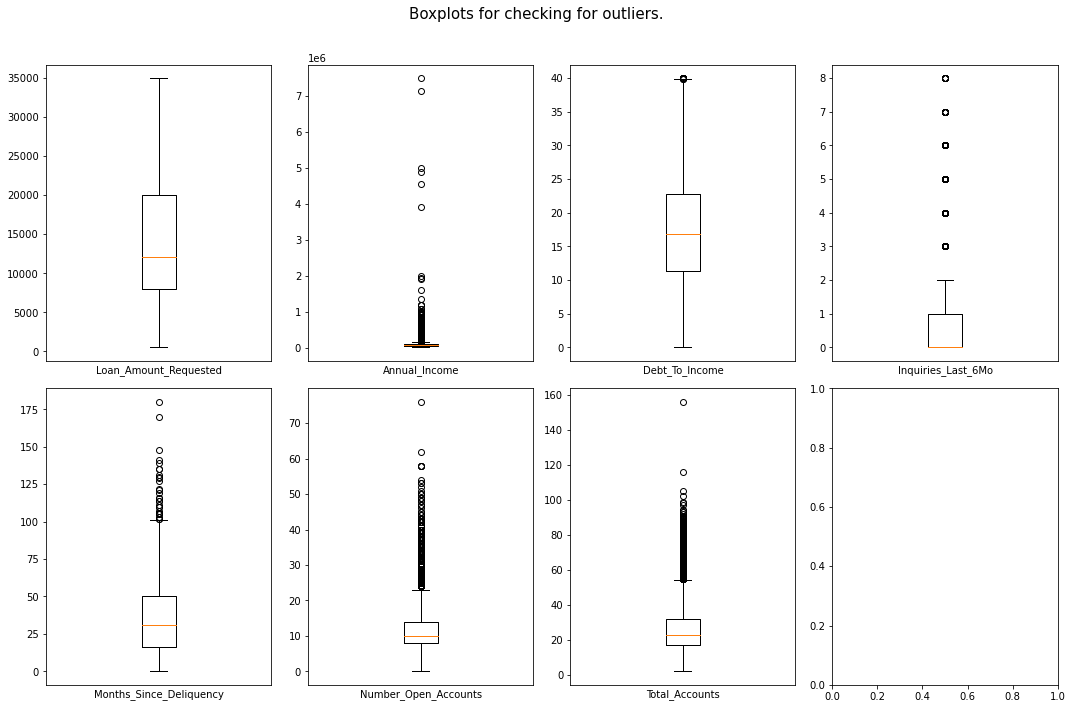
* **‘Annual\_Income’** has a high positive skewness. It contains outliers as can be seen in the boxplots.
* **‘Loan\_Amount’** requested has a descent spread and does not contain any outliers.
* **‘Months\_Since\_Delinquency’** also contains outliers as can be seen in its boxplot.
* Rest of the features have a positive skew and can be handled using any transformation techniques like Boxcox,Log or SquareRoot. These features also contain outliers.

###### **Inferences from the Proportion Plots**

* The target variable plot shows us that there are lesser applicants who have received a 1% interest rate as compared to 2-3%.
* Lot of loan applicants have been employed for over 10 years.
* People are mostly taking out loans to clear their credit card bills or to consolidate all their debts into one.
* Most applicants either live on rent or have a mortgage on their house.
* Most past approved applicants have had verified income sources. This feature should have only 2 categories as verified and not verified.
* Finally, the dataset is male dominated.

**OUTLIERS AND TREATMENT**

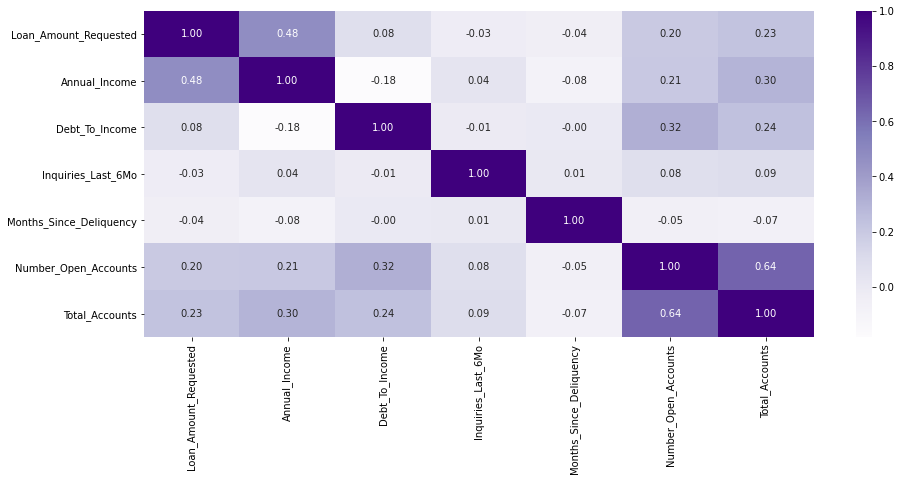
We use boxplots to visualise outliers present in the data.

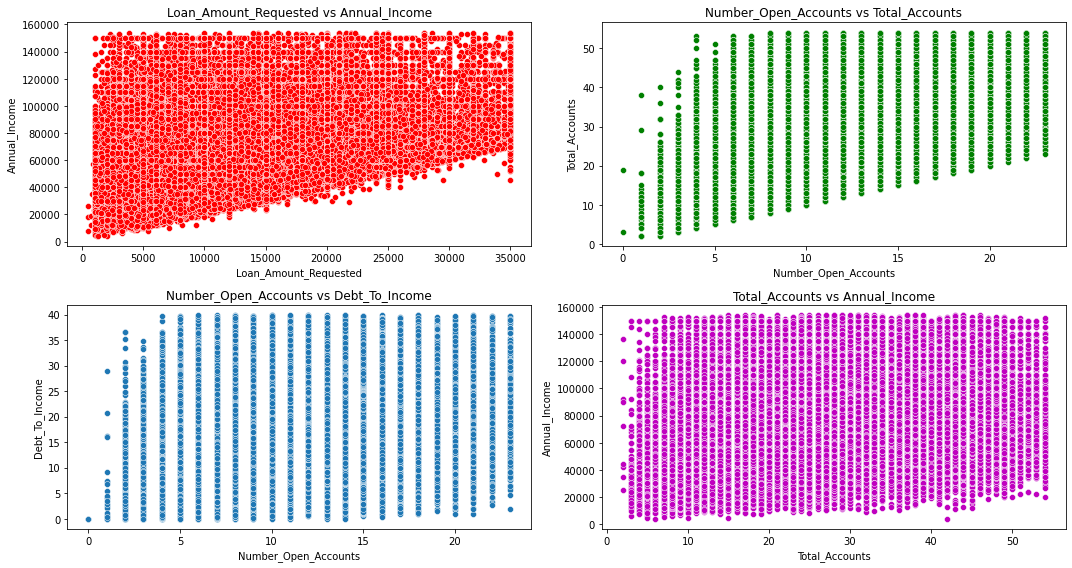


* Except loan\_amount\_requested all other features have outliers present as can be seen in the boxplot.
* Apparently the **‘Annual\_Income’** feature is having many outliers.
* We have treated the outliers using the IQR method.

**FEATURE CORRELATION**

We have used **Pearson correlation** to find correlation among features and plot them on a heatmap in Seaborn.

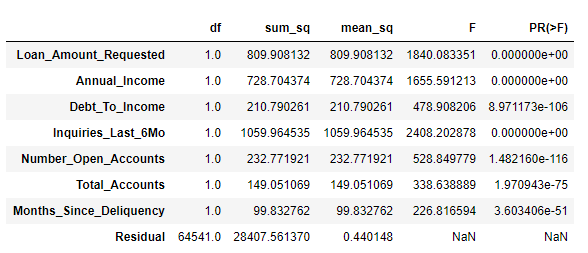




**STATISTICAL SUMMARY**

We have performed the following statistical tests in our analysis

* Anova Test (Numerical vs Categorical)
* Tukey HSD test (To check if mean difference is present)
* Chi Square Test (Categorical vs Categorical)

**Anova Test between numerical and target variable**

Since all the meaures are below 0.05, we can check for which category variables are seeing a difference in mean using the pairwise tukey hsd test.

**Tuckey HSD Test**

**Tukey HSD test for Loan\_Amount\_Requested is:**

Multiple Comparison of Means - Tukey HSD, FWER=0.05

=========================================================

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| group1 | group2 | meandiff | p-adj | lower | upper | reject |
| 1 | 2 | 189.3326 | 0.0019 | 58.8893 | 319.7759 | True |
| 1 | 3 | 2752.1392 | 0.001 | 2616.5296 | 2887.7489 | True |
| 2 | 3 | 2562.8067 | 0.001 | 2451.4188 | 2674.1945 | True |

**Tukey HSD test for Annual\_Income is:**

Multiple Comparison of Means - Tukey HSD, FWER=0.05

=========================================================

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| group1 | group2 | meandiff | p-adj | lower | upper | reject |
| 1 | 2 | -6706.2963 | 0.001 | -7164.3935 | -6248.1992 | True |
| 1 | 3 | -6654.8814 | 0.001 | -7131.1222 | -6178.6406 | True |
| 2 | 3 | 51.4149 | 0.9 | -339.7624 | 442.5923 | False |

**Tukey HSD test for Debt\_To\_Income is:**

Multiple Comparison of Means - Tukey HSD, FWER=0.05

=========================================================

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| group1 | group2 | meandiff | p-adj | lower | upper | reject |
| 1 | 2 | 2.0233 | 0.001 | 1.896 | 2.1507 | True |
| 1 | 3 | 3.3391 | 0.001 | 3.2067 | 3.4715 | True |
| 2 | 3 | 1.3157 | 0.001 | 1.207 | 1.4245 | True |

**Tukey HSD test for Inquiries\_Last\_6Mo is:**

Multiple Comparison of Means - Tukey HSD, FWER=0.05

=========================================================

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| group1 | group2 | meandiff | p-adj | lower | upper | reject |
| 1 | 2 | 0.1237 | 0.001 | 0.1122 | 0.1353 | True |
| 1 | 3 | 0.3205 | 0.001 | 0.3085 | 0.3325 | True |
| 2 | 3 | 0.1968 | 0.001 | 0.1869 | 0.2066 | True |

**Tukey HSD test for Number\_Open\_Accounts is:**

Multiple Comparison of Means - Tukey HSD, FWER=0.05

=========================================================

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| group1 | group2 | meandiff | p-adj | lower | upper | reject |
| 1 | 2 | 0.1237 | 0.001 | -0.2624 | -0.1221 | True |
| 1 | 3 | -0.0915 | 0.0092 | -0.1644 | -0.0185 | True |
| 2 | 3 | 0.1008 | 0.001 | 0.0409 | 0.1607 | True |

**Tukey HSD test for Total\_Accounts is:**

Multiple Comparison of Means - Tukey HSD, FWER=0.05

=========================================================

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| group1 | group2 | meandiff | p-adj | lower | upper | reject |
| 1 | 2 | -1.4818 | 0.001 | -1.6493 | -1.3143 | True |
| 1 | 3 | -1.7551 | 0.001 | -1.9293 | -1.581 | True |
| 2 | 3 | -0.2733 | 0.001 | -0.4164 | -0.1303 | True |

**Tukey HSD test for Months\_Since\_Deliquency is:**

Multiple Comparison of Means - Tukey HSD, FWER=0.05

=========================================================

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| group1 | group2 | meandiff | p-adj | lower | upper | reject |
| 1 | 2 | -2.3008 | 0.001 | -2.889 | -1.7126 | True |
| 1 | 3 | -3.1654 | 0.001 | -3.7672 | - 2.5635 | True |
| 2 | 3 | -0.8646 | 0.001 | -1.3045 | - 0.4246 | True |

**CHI Square Test**



###### **There is no difference in gender within the 3 target categories.**

### **Inferences from Anova Test**

**H0:** There is no difference in means between interest rate groups and a specific feature.

**HA:** There is a difference in means between interest rate groups and a specific feature.

* All the numerical columns are having differences in mean for each of the target categories.
* This can be confirmed by the pairwise Tukey HSD test as well.

**Inferences from Chi Square Test**

**H0:** Assumes that there is no association between the two variables.

**HA:** Assumes that there is an association between the two variables.

* Gender is the only categorical column that is not seeing any difference within each target category.

**BASE MODEL**

* Here we have used Logistic Regression algorithm with ‘multinomial’ argument under the multiclass parameter as we have more than two classes in the target.
* The Train Test Split ratio used is as follows - 80:20
* We perform a train test split using model selection technique in sklearn. Also we will check whether our train and test sets represent the total population properly by performing ttest independent between test, train and overall data.
* **Inference**

**H0 -** Sample Mean = Population Mean

**HA -** Sample Mean != Population Mean

* Here samples are X\_test, y\_test, X\_train, y\_train. Since the P values for all the columns are coming greater than 0.05 therefore, we have ‘failed to reject’ the null hypothesis which means that all the samples are proper representation of the population.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Target Class | Precision | Recall | F1-score | Support |
|  |  |  |  |  |
| 1 | 0.21 | 0.00 | 0.00 | 6735 |
| 2 | 0.45 | 0.68 | 0.54 | 14149 |
| 3 | 0.48 | 0.45 | 0.47 | 11978 |
|  |  |  |  |  |
| Accuracy |  |  | 0.46 | 32862 |
| Macro average | 0.30 | 0.38 | 0.34 | 32862 |
| Weighted average | 0.41 | 0.46 | 0.40 | 32862 |

**Overall Accuracy of Base Model = 46 %**

**NEXT STEPS**

**Since our model accuracy is not up to our expectations we will go ahead with other steps. These steps include:**

* **Step 1**

Feature Selection using RFE and SFS

* **Step 2**

Implementation of few other algorithms. The Algorithms that we have considered are:

1. Logistic Regression
2. Decision Tree Classifier
3. Naïve Bayes Classifier
4. Random Forest Classifier
5. Extra Trees Classifier
6. Gradient Boosting Classifier
7. AdaBoost Classifier
8. XGB Classifier
9. CatBoost Classifier

* **Step 3**

Hyper Parameter Tuning

1. ***Logistic Regression***



**Definition:**

Logistic regression is a classification algorithm used to assign observations to a discrete set of classes. It is a Machine Learning algorithm which is used for the classification problems, it is a predictive analysis algorithm and based on the concept of probability.Some of the examples of classification problems are Email spam or not spam, Online transactions Fraud or not Fraud, Tumour Malignant or Benign. Logistic regression transforms its output using the logistic sigmoid function to return a probability value.

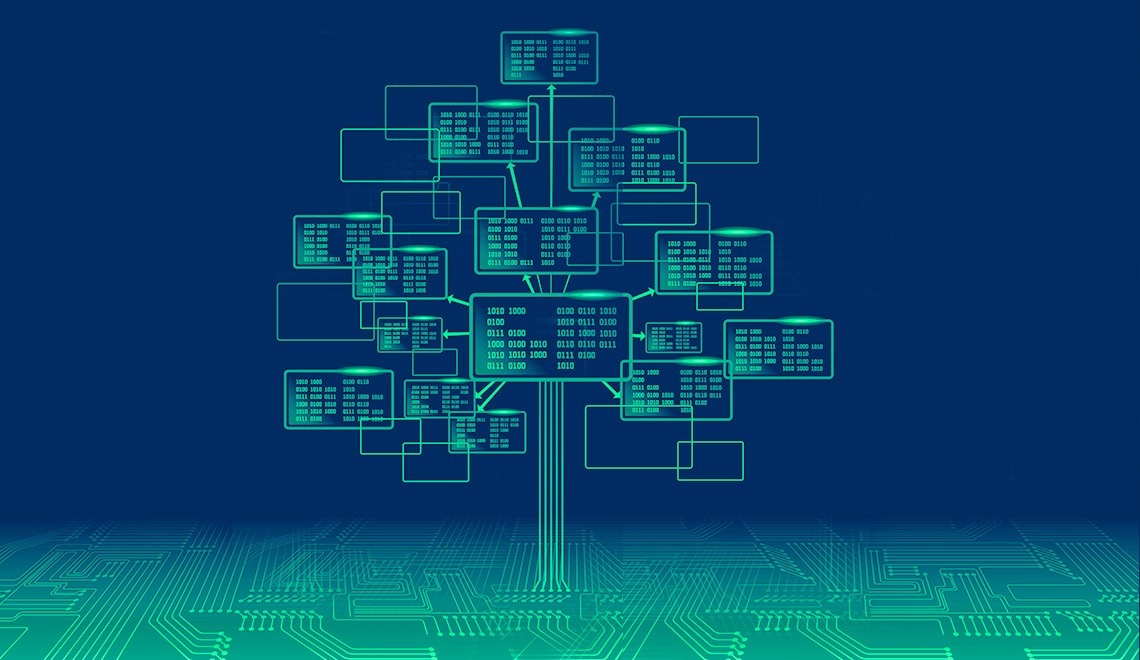
**Output:**

We have applied ***Feature Selection techniques with Logistic Regression model***.

The output is as follows -

|  |  |  |  |
| --- | --- | --- | --- |
| *Model* | *Data Type* | *F1 - Score* | *Accuracy* |
| Logistic Regression with SFS | **Non - Standardized** | **0.434845** | **0.495267** |
| Logistic Regression with SFS | **Standardized** | **0.441312** | **0.498155** |
| Logistic Regression with RFE | **Non - Standardized** | **0.477117** | **0.500011** |
| Logistic Regression with RFE | **Standardized** | **0.483583** | **0.504037** |

1. ***Decision Tree Classifier***



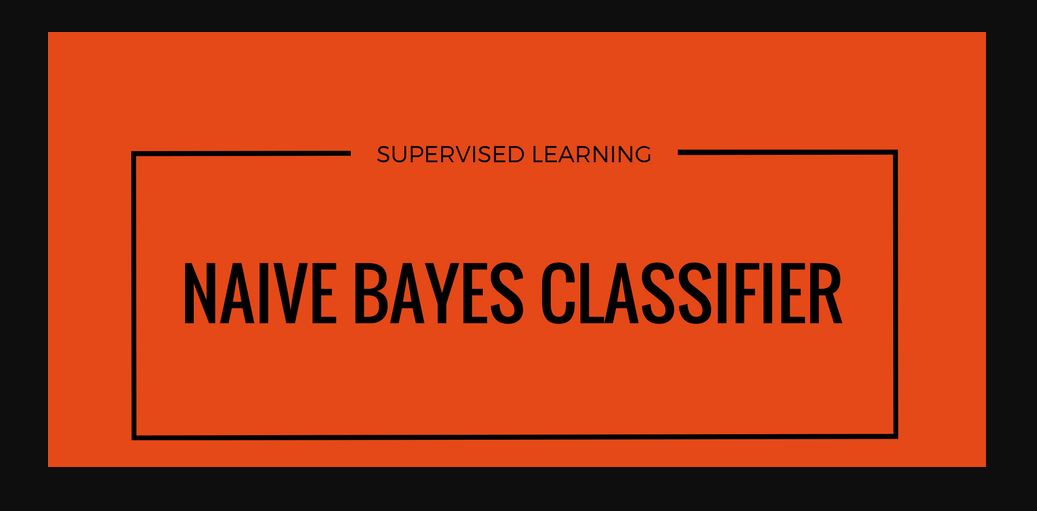
**Definition:**

Decision tree learning is one of the predictive modelling approaches used in [statistics](https://en.wikipedia.org/wiki/Statistics), [data mining](https://en.wikipedia.org/wiki/Data_mining) and [machine learning](https://en.wikipedia.org/wiki/Machine_learning). It uses a [decision tree](https://en.wikipedia.org/wiki/Decision_tree) (as a [predictive model](https://en.wikipedia.org/wiki/Predictive_modelling)) to go from observations about an item (represented in the branches) to conclusions about the item's target value (represented in the leaves). Tree models where the target variable can take a discrete set of values are called [classification](https://en.wikipedia.org/wiki/Classification) [trees](https://en.wikipedia.org/wiki/Decision_tree); in these tree structures, [leaves](https://en.wikipedia.org/wiki/Leaf_node) represent class labels and branches represent [conjunctions](https://en.wikipedia.org/wiki/Logical_conjunction) of features that lead to those class labels. Decision trees where the target variable can take continuous values (typically [real numbers](https://en.wikipedia.org/wiki/Real_numbers)) are called [regression](https://en.wikipedia.org/wiki/Regression_analysis) [trees](https://en.wikipedia.org/wiki/Decision_tree). Decision trees are among the most popular machine learning algorithms given their intelligibility and simplicity

**Output:**

|  |  |  |  |
| --- | --- | --- | --- |
| *Model* | *Data Type* | *F1 - Score* | *Accuracy* |
| Decision Tree Classifier | **Non - Standardized** | **0.416821** | **0.416335** |
| Decision Tree Classifier | **Standardized** | **0.416758** | **0.417171** |

1. ***Naïve Bayes Classifier***



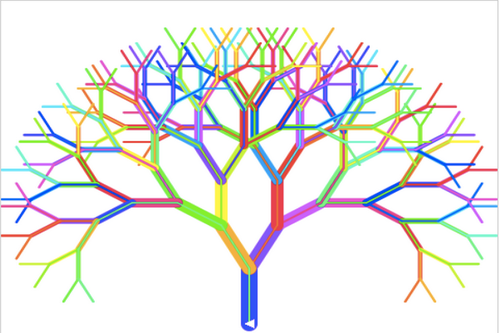
**Definition:**

Naive Bayes classifiers are a family of simple "[probabilistic classifiers](https://en.wikipedia.org/wiki/Probabilistic_classification)" based on applying [Bayes' theorem](https://en.wikipedia.org/wiki/Bayes%27_theorem) with strong (naïve) [independence](https://en.wikipedia.org/wiki/Statistical_independence) assumptions between the features. They are among the simplest [Bayesian network](https://en.wikipedia.org/wiki/Bayesian_network) models, but coupled with [kernel density estimation](https://en.wikipedia.org/wiki/Kernel_density_estimation), they can achieve higher accuracy levels. Naïve Bayes classifiers are highly scalable, requiring a number of parameters linear in the number of variables (features/predictors) in a learning problem. [Maximum-likelihood](https://en.wikipedia.org/wiki/Maximum-likelihood_estimation) training can be done by evaluating a [closed-form expression](https://en.wikipedia.org/wiki/Closed-form_expression), which takes [linear time](https://en.wikipedia.org/wiki/Linear_time), rather than by expensive [iterative approximation](https://en.wikipedia.org/wiki/Iterative_method) as used for many other types of classifiers.

**Output:**

|  |  |  |  |
| --- | --- | --- | --- |
| *Model* | *Data Type* | *F1 - Score* | *Accuracy* |
| Naïve Bayes Classifier | **Non - Standardized** | **0.458256** | **0.464866** |
| Naïve Bayes Classifier | **Standardized** | **NaN** | **NaN** |

1. ***Random Forest Classifier***



**Definition:**

Random forests or random decision forests are an [ensemble learning](https://en.wikipedia.org/wiki/Ensemble_learning) method for [classification](https://en.wikipedia.org/wiki/Statistical_classification), [regression](https://en.wikipedia.org/wiki/Regression_analysis) and other tasks that operates by constructing a multitude of [decision trees](https://en.wikipedia.org/wiki/Decision_tree_learning) at training time and outputting the class that is the [mode](https://en.wikipedia.org/wiki/Mode_(statistics)) of the classes (classification) or mean/average prediction (regression) of the individual trees. Random decision forests correct for decision trees' habit of [overfitting](https://en.wikipedia.org/wiki/Overfitting) to their [training set](https://en.wikipedia.org/wiki/Test_set). Random forests generally outperform [decision trees](https://en.wikipedia.org/wiki/Decision_tree_learning), but their accuracy is lower than gradient boosted trees. However, data characteristics can affect their performance.

**Output:**

|  |  |  |  |
| --- | --- | --- | --- |
| *Model* | *Data Type* | *F1 - Score* | *Accuracy* |
| Random Forest Classifier | **Non - Standardized** | **0.484381** | **0.495330** |
| Random Forest Classifier | **Standardized** | **0.484153** | **0.495098** |

1. ***Extra Trees Classifier***



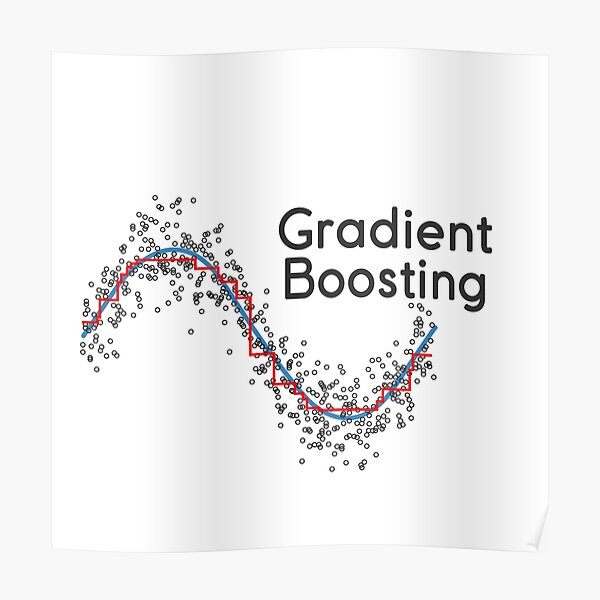
**Definition:**

Extra Trees Classifier is a type of ensemble learning technique which aggregates the results of multiple de-correlated decision trees collected in a “forest” to output its classification result. In concept, it is very similar to a Random Forest Classifier and only differs from it in the manner of construction of the decision trees in the forest.

**Output:**

|  |  |  |  |
| --- | --- | --- | --- |
| *Model* | *Data Type* | *F1 - Score* | *Accuracy* |
| Extra Trees Classifier | Non - Standardized | 0.469124 | 0.481915 |
| Extra Trees Classifier | Standardized | 0.469149 | 0.480039 |

1. ***Gradient Boosting Classifier***



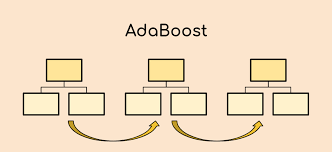
**Definition:**

Gradient Boosting is a [machine learning](https://en.wikipedia.org/wiki/Machine_learning) technique for [regression](https://en.wikipedia.org/wiki/Regression_(machine_learning)) and [classification](https://en.wikipedia.org/wiki/Classification_(machine_learning)) problems, which produces a prediction model in the form of an [ensemble](https://en.wikipedia.org/wiki/Ensemble_learning) of weak prediction models, typically [decision trees](https://en.wikipedia.org/wiki/Decision_tree_learning). When a decision tree is the weak learner, the resulting algorithm is called gradient boosted trees, which usually outperforms [random forest](https://en.wikipedia.org/wiki/Random_forest). It builds the model in a stage-wise fashion like other [boosting](https://en.wikipedia.org/wiki/Boosting_(machine_learning)) methods do, and it generalizes them by allowing optimization of an arbitrary [differentiable](https://en.wikipedia.org/wiki/Differentiable_function) [loss function](https://en.wikipedia.org/wiki/Loss_function).

**Output:**

|  |  |  |  |
| --- | --- | --- | --- |
| *Model* | *Data Type* | *F1 - Score* | *Accuracy* |
| Gradient Boosting Classifier | **Non - Standardized** | **0.497880** | **0.518331** |
| Gradient Boosting Classifier | **Standardized** | **0.497880** | **0.518331** |

1. ***AdaBoost Classifier***



**Definition:**

An AdaBoost Classifier is a meta-estimator that begins by fitting a classifier on the original dataset and then fits additional copies of the classifier on the same dataset but where the weights of incorrectly classified instances are adjusted such that subsequent classifiers focus more on difficult cases. The basic concept behind AdaBoost is to set the weights of classifiers and training the data sample in each iteration such that it ensures the accurate predictions of unusual observations. Any machine learning algorithm can be used as base classifier if it accepts weights on the training set.

**Output:**

|  |  |  |  |
| --- | --- | --- | --- |
| *Model* | *Data Type* | *F1 - Score* | *Accuracy* |
| AdaBoost Classifier | **Non - Standardized** | **0.495532** | **0.511764** |
| AdaBoost Classifier | **Standardized** | **0. 495532** | **0. 511764** |

1. ***XGB Classifier***



**Definition:**

XGBoost is an algorithm that has recently been dominating applied machine learning and Kaggle competitions for structured or tabular data. XGBoost is an implementation of gradient boosted decision trees designed for speed and performance. XGBoost is a software library that you can download and install on your machine, then access from a variety of interfaces. The library is laser focused on computational speed and model performance, as such there are few frills. Nevertheless, it does offer a number of advanced features. The implementation of the model supports the features of the scikit-learn and R implementations, with new additions like regularization.

**Output:**

|  |  |  |  |
| --- | --- | --- | --- |
| *Model* | *Data Type* | *F1 - Score* | *Accuracy* |
| XGB Classifier | **Non - Standardized** | **0.511055** | **0.523300** |
| XGB Classifier | **Standardized** | **0.511039** | **0.523285** |

1. ***CatBoost Classifier***



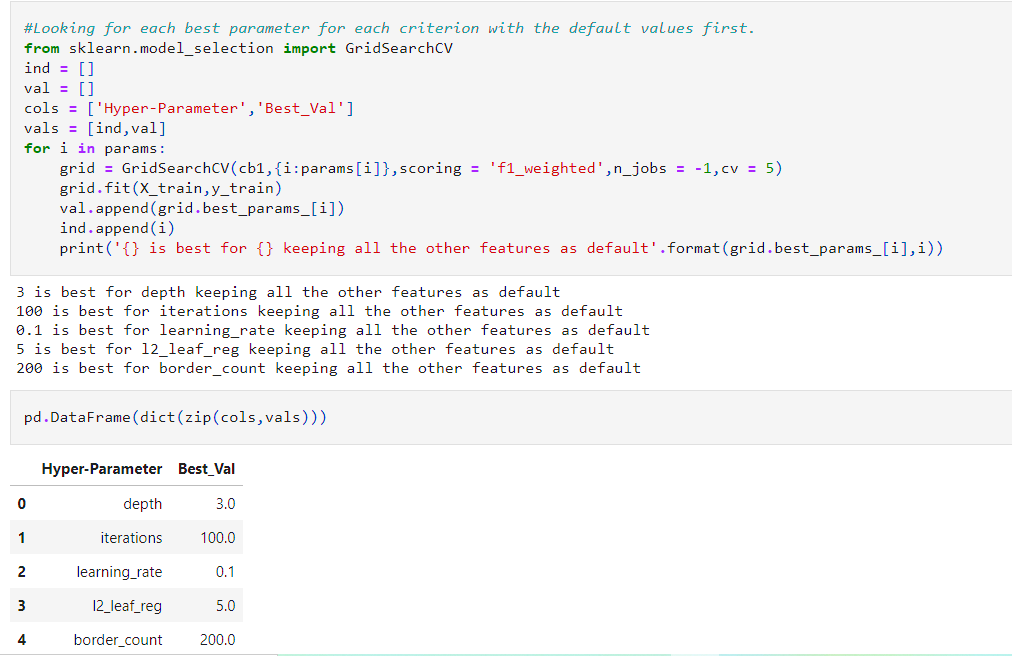
**Definition:**

CatBoost is an algorithm based on gradient boosting. It is a new machine learning technique developed by Yandex that out performs many existing boosting algorithms like XGBoost, Light GBM. The main difference between CatBoost and other algorithms is that CatBoost implements symmetric trees which helps in decreasing prediction time, which is extremely important for low latency environments.

**Output:**

|  |  |  |  |
| --- | --- | --- | --- |
| *Model* | *Data Type* | *F1 - Score* | *Accuracy* |
| CatBoost Classifier | **Non - Standardized** | **0.512051** | **0.524326** |
| CatBoost Classifier | **Standardized** | **0. 512051** | **0. 524326** |

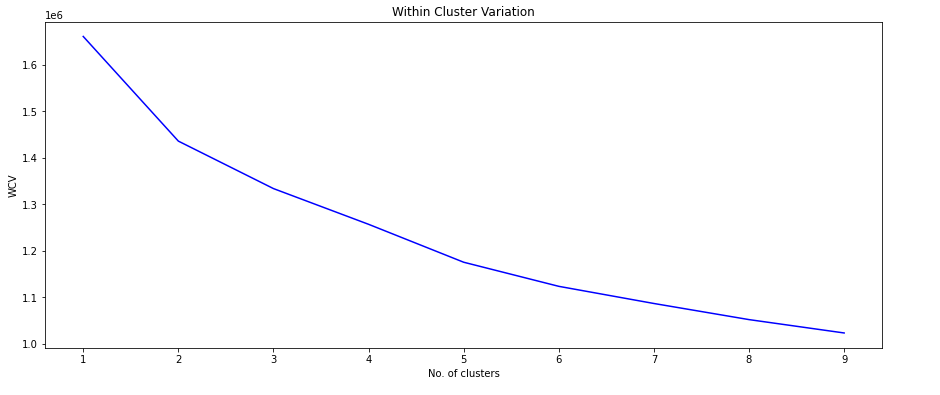
**HYPERPARAMETER TUNING**



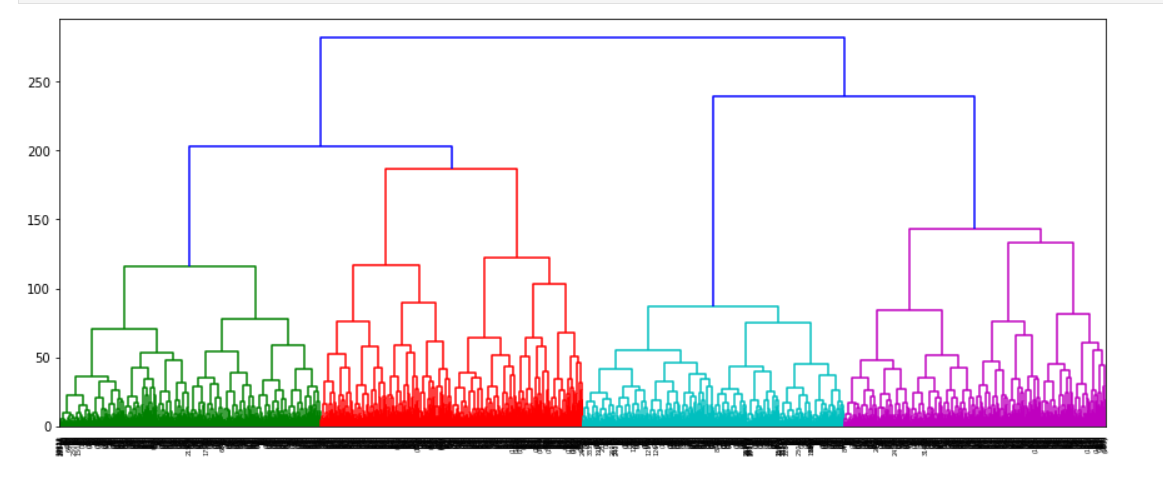
* We applied these best parameters to CatBoost Classifier.
* The original accuracy of CatBoost was 52%
* However, after implementing the CatBoost Classifier with tuned parameters, the accuracy of the model dropped to 51%

**CLUSTERING**

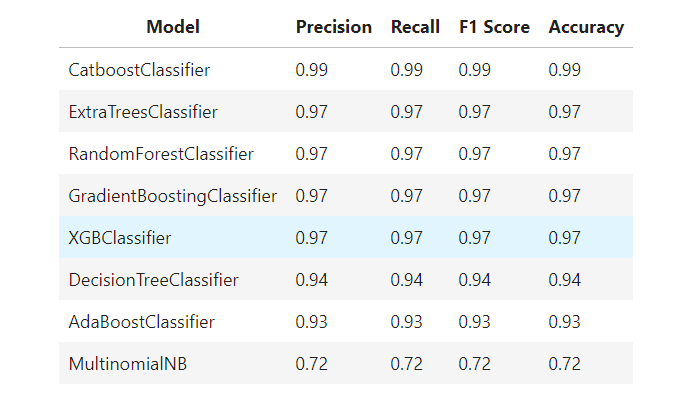
* Before applying any Clustering Algorithm, we first found the Optimal K-value. We found it using the ‘Within Cluster Variation’ Technique.



* The Optimal K-value that can be considered are 2, 3, 4. But as our dataset had 3 clusters in the beginning, we have considered the **Optimal K-value = 3**
* Further, we applied Agglomerative Clustering to the dataset.



* Lastly, we used these clusters as targets and applied Classification Algorithms on it.
* The following results were obtained -



* **After clustering and creating ML models on the new target, we can clearly see a massive improvement in model performance. It seems like there must have been some more parameters that have determined our original target variable that was not provided to us within our dataset.**

**BUSINESS APPLICATIONS**

**CONCLUSION**

* **CLASSIFICATION**

1. The following data transformation were performed
2. Imputing the months\_Since\_Deliquency with 360
3. Taking square root for loan\_amount\_requested to reduce positive skew.
4. Taking boxcox transformation of Annual\_Income
5. Taking square root for Number\_Open\_Accounts
6. Taking square root for Total\_Accounts
7. Since Months\_Since\_Deliquency has become bimodal we will be taking the log of Months\_Since\_Deliquency subtracted with its mean.
8. For Length\_Employed, we are converting the same into bins since we had one value as 10+years.
9. Base Model was implemented and accuracy obtained was 46%
10. Further, we applied Feature Selection Techniques like RFE and SFS.
11. Different algorithms were implemented.
12. Hyper parameter tuning was performed. However, accuracy dropped by 1% after Hyper parameter tuning.
13. SMOTE was applied for imbalanced data.
14. **Finally, the conclusion is that CatBoost Classifier gave the best accuracy at 60%**

* **CLUSTERING**

1. We first applied Within Cluster Variation and found Optimal K-Value = 3
2. Next, Agglomerative Clustering was applied.
3. PCA was also applied for Dimensionality Reduction.
4. Lastly, newly formed clusters were used as targets and Classification Algorithms were re-applied.
5. **Finally, the conclusion is that after clustering and creating ML models on the new target, we can clearly see a massive improvement in model performance. It seems like there must have been some more parameters that have determined our original target variable that was not provided to us within our dataset.**