**Scope of Project**

As of 2019, in the last three years, the Indian banking system has lost Rs 1.76 lakh crore on account of non-performing loans defaulters. **Loans default causes huge loss for the banks, so there is a need to pay much attention on this issue and apply various methods to detect and predict default behavior of the customers, beforehand only.** The bank employees are not able to efficiently analyze or predict whether the customer can payback the amount or not that is, whether he is a good customer or a bad customer.

And, here comes the motivation for the project where we use machine learning predictive modeling to solve the problem. The aim of this project is to find the nature of the client applying for the loan and predict whether he is eligible to be granted loan or not. By comparing the new data points with the data points of thousands of prior customers, machine learning is able to predict the eligibility of the customer to acquire loan.

Advantages of using Machine learning predictive modeling as compared to the traditional way of loan approval are as follows.

From bank’s/lender’s point of view:

1. More accurate than manual approval because more attributes can be considered in ML modeling.
2. A great amount of reduction in time taken to approve the loans. So, more loans applications can be processed in the given time.
3. Reduction in the manpower as one model checks all the attributes itself.

From customers’/borrowers’ point of view:

1. The waiting period between application of loan and approval of loan will be reduced. Hence, loan granting speeds up.
2. More accurate prediction will lead to more good customers acquiring the loan.

**Dataset Description**

1. **Loan ID**

The unique ID for each loan application in the bank.

Variable Type: Categorical

Data Type: Integer

|  |  |
| --- | --- |
| **Count** | **90514** |
| **Unique** | **75774** |
| **Top** | **78b60929-4338-4d4f-aee5-588463e54300** |
| **Freq** | **2** |
| **NaN** | **0** |

1. **Customer ID**

The unique ID for each customer in the bank.

Variable Type: Categorical

Data Type: Integer

|  |  |
| --- | --- |
| **Count** | **90514** |
| **Unique** | **75774** |
| **Top** | **3178d924-03d2-45f7-90df-28034611d8b5** |
| **Freq** | **2** |
| **NaN** | **0** |

1. **Current Loan Amount**

For code purposes the name of this attribute is changedto **Loan\_amount**

The amount of loan asked for by the applicant.

Variable Type: Continuous

Data Type: Integer

|  |  |
| --- | --- |
| **Count** | **9.051400e+04** |
| **Mean** | **1.177071e+07** |
| **Std** | |  |  |  |  | | --- | --- | --- | --- | |  |  | **3.179645e+07** |  | |
| **Min** | **1.080200e+04** |
| **25%** | |  |  | | --- | --- | | **1.795200e+05** |  | |
| **50%** | **3.122680e+05** |
| **75%** | **5.248705e+05** |
| **Max** | **1.000000e+08** |
| **NaN** | **0** |

1. **Term**

The duration of time for which the loan is asked for.

Variable Type: Categorical

Data Type: String

|  |  |
| --- | --- |
| **Count** | **90514** |
| **Unique** | **2** |
| **Top** | **Short Term** |
| **Freq** | **65343** |
| **NaN** | **0** |

1. **Credit Score**

A feature denoting creditworthiness of the borrower.

Variable Type: Continuous

Data Type: Float

|  |  |
| --- | --- |
| **Count** | **73156.00000** |
| **Mean** | |  | | --- | | **1077.839384** | |
| **Std** | **1477.909807** |
| **Min** | |  |  | | --- | --- | |  |  |   **585.000000** |
| **25%** | **705.000000** |
| **50%** | **724.000000** |
| **75%** | **741.000000** |
| **Max** | **7510.000000** |
| **NaN** | **19154** |

1. **Annual Income**

The amount earned by the customer yearly in INR.

Variable Type: Continuous

Data Type: Float

|  |  |
| --- | --- |
| **Count** | **7.315600e+04** |
| **Mean** | **1.378678e+06** |
| **Std** | **1.099577e+06** |
| **Min** | **7.662700e+04** |
| **25%** | **8.493570e+05** |
| **50%** | **1.173763e+06** |
| **75%** | **1.650236e+06** |
| **Max** | **1.655574e+08** |
| **NaN** | **19154** |

1. **Years in current job**

For code purposes the name of this attribute is changedto **job**.

The number of years customer has been doing his/her current job.

Variable Type: Categorical

Data Type: Integer

|  |  |
| --- | --- |
| **Count** | **86702** |
| **Unique** | **11** |
| **Top** | **10+ years** |
| **Freq** | **28225** |
| **NaN** | **4222** |

1. **Home Ownership**

For code purposes the name of this attribute is changedto **property**.

The type of home the customer is living in.

Variable Type: Categorical

Data Type: String

|  |  |
| --- | --- |
| **Count** | **90514** |
| **Unique** | **4** |
| **Top** | **Home Mortgage** |
| **Freq** | **43866** |
| **NaN** | **0** |

1. **Purpose**

The reason stated by the customer for taking loan.

Variable Type: Categorical

Data Type: String

|  |  |
| --- | --- |
| **Count** | **90514** |
| **Unique** | **16** |
| **Top** | **Debt Consolidation** |
| **Freq** | **71076** |
| **NaN** | **0** |

1. **Monthly Debt**

Monthly debt can be any mortgages that customer has or is applying for, rent payments, car loans, student loans, any other loans and credit card debt, which the customer pays on monthly basis.

Variable Type: Continuous

Data Type: Float

|  |  |
| --- | --- |
| **Count** | **90514.000000** |
| **Mean** | **18458.633147** |
| **Std** | **12196.887627** |
| **Min** | **0.000000** |
| **25%** | **10193.500000** |
| **50%** | **16200.825000** |
| **75%** | **23979.900000** |
| **Max** | **435843.280000** |
| **NaN** | **0** |

1. **Years of Credit History**

For code purposes the name of this attribute is changedto **credit\_years.**

The length of time in years that credit accounts have been open and active.

Variable Type: Continuous

Data Type: Float

|  |  |
| --- | --- |
| **Count** | **90514.000000** |
| **Mean** | **18.195437** |
| **Std** | **7.002528** |
| **Min** | **3.600000** |
| **25%** | **13.500000** |
| **50%** | **16.900000** |
| **75%** | **21.700000** |
| **Max** | **70.500000** |

1. **Months since last delinquent**

For code purposes the name of this attribute is changedto **delinquency**

Duration in months for which he is falling off from paying loans.

Variable Type: Continuous

Data Type: String

|  |  |
| --- | --- |
| **Count** | **42428.000000** |
| **Mean** | **34.949562** |
| **Std** | **22.008267** |
| **Min** | **0.000000** |
| **25%** | **16.000000** |
| **50%** | **32.000000** |
| **75%** | **51.000000** |
| **Max** | **176.000000** |
| **NaN** | **53141** |

1. **Number of Open Accounts**

For code purposes the name of this attribute is changedto **open\_accounts**

The count of Bank accounts that the customer has.

Variable Type: Numerical (Discrete)

Data Type: Float

|  |  |
| --- | --- |
| **Counts** | **90514.000000** |
| **Mean** | **11.125837** |
| **Std** | **4.998590** |
| **Min** | **0.000000** |
| **25%** | **8.000000** |
| **50%** | **10.000000** |
| **75%** | **14.000000** |
| **Max** | **76.000000** |

1. **Number of Credit Problems**

For code purposes the name of this attribute is changedto **credit\_problems**.

The count of credit problems listed by customer.

Variable Type: Categorical

Data Type: Integer

|  |  |
| --- | --- |
| **Count** | **90514.000000** |
| **Mean** | **0.168482** |
| **Std** | **0.484199** |
| **Min** | **0.000000** |
| **25%** | **0.000000** |
| **50%** | **0.000000** |
| **75%** | **0.000000** |
| **Max** | **15.000000** |
| **NaN** | **0** |

1. **Current Credit Balance**

For code purposes the name of this attribute is changedto **credit\_bal**

The current balance that the customer holds in his bank account.

Variable Type: Continuous

Data Type: Float

|  |  |
| --- | --- |
| **Count** | **9.051400e+04** |
| **Mean** | **2.947552e+05** |
| **Std** | **3.777427e+05** |
| **Min** | **0.000000e+00** |
| **25%** | **1.125750e+05** |
| **50%** | **2.097695e+05** |
| **75%** | **3.681012e+05** |
| **Max** | **3.287897e+07** |
| **NaN** | **0** |

1. **Maximum Open Credit**

For code purposes the name of this attribute is changedto **MOC**

Open credit refers to accounts that customer can borrow from up to a maximum amount (like a credit card) but which must also be paid back in full each month.

Variable Type: Continuous

Data Type: Float

|  |  |
| --- | --- |
| **Count** | **9.051300e+04** |
| **Mean** | **7.651869e+05** |
| **Std** | **8.748301e+06** |
| **Min** | **0.000000e+00** |
| **25%** | **2.730420e+05** |
| **50%** | **4.676540e+05** |
| **75%** | **7.833980e+05** |
| **Max** | **1.539738e+09** |
| **NaN** | **2** |

1. **Bankruptcies**

Whether there have been any case of bankruptcy or not.

Variable Type: Numerical (Discrete)

Data Type: String

|  |  |
| --- | --- |
| **Count** | **90336.000000** |
| **Mean** | **0.117805** |
| **Std** | **0.351519** |
| **Min** | **0.000000** |
| **25%** | **0.000000** |
| **50%** | **0.000000** |
| **75%** | **0.000000** |
| **Max** | **7.000000** |
| **NaN** | **204** |

1. **Tax Liens**

Whether there is any claim against the asset of the customer due to unpaid taxes.

Variable Type: Categorical

Data Type: Float

|  |  |
| --- | --- |
| **Count** | **90504.000000** |
| **Mean** | **0.029093** |
| **Std** | **0.259285** |
| **Min** | **0.000000** |
| **25%** | **0.000000** |
| **50%** | **0.000000** |
| **75%** | **0.000000** |
| **Max** | **15.000000** |
| **NaN** | **10** |

1. **Loan Status**

Whether the loan was finally granted or not.

Variable Type: Categorical

Data Type: Integer

|  |  |
| --- | --- |
| **Count** | **90514** |
| **Unique** | **2** |
| **Top** | **Yes** |
| **Freq** | **70035** |
| **NaN** | **0** |

**Exploratory Analysis**

**Types of Data Visualization techniques used:**

1. **Histogram**

In statistics, a [histogram](https://en.wikipedia.org/wiki/Histogram) is representation of the distribution of numerical data, where the data are binned and the count for each bin is represented. More generally, a histogram is an aggregated bar chart, with several possible aggregation functions (e.g. sum, average, count...).

1. **Scatter Plot**

A scatter plot (aka scatter chart, scatter graph) used dots to represent value of two different numeric variables. The position of each dot on the horizontal and vertical axis indicates values for an individual data point. Scatter plot are used to observe relationships between variables.

1. **Box Plot**

Box Plot is the visual representation of the depicting groups of numerical data through their quartiles. Boxplot is also used for detect the outlier in data set. It captures the summary of the data efficiently with a simple box and whiskers and allows us to compare easily across groups. Boxplot summarizes a sample data using 25th, 50th and 75th percentiles. These percentiles are also known as the lower quartile, median and upper quartile.

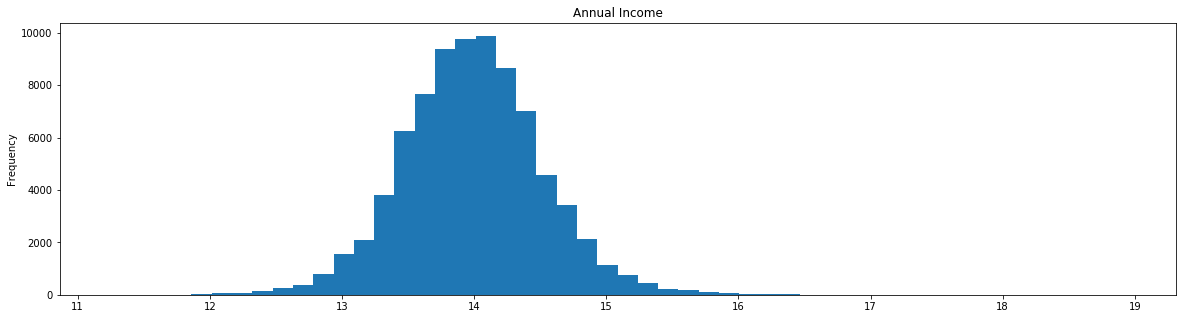
1. **Heatmap**

Heatmap is a way to show some sort of matrix plot. To use a heatmap the data should be in a matrix form. By matrix we mean that the index name and the column name must match in some way so that the data that we fill inside the cells are relevant.

1. **Count Plot**

A countplot basically counts the categories and returns a count of their occurrences. It is one of the simplest plots provided by the seaborn library.

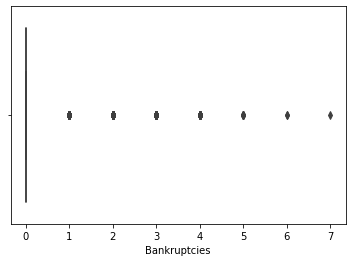
1. **Annual Income**

****

**Figure-1**

In the figure1, X- axis represents the annual income which shows the normal distribution of data. Here, the data is distributed in the range of 176627.0 to 165557400.0.

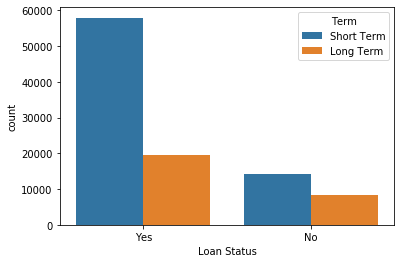
1. **Bankruptcies**

****

**Figure 2**

In the above figure 2, X- axis represents the outliers of the attribute Bankruptcies. Here, the data is spread from 0 to 7 in which outliers are ranging from 1 to 7.

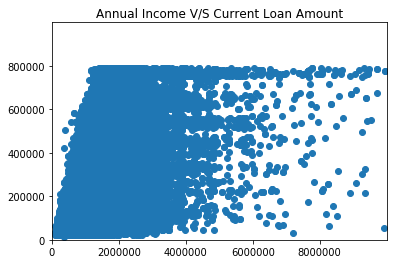
1. **Loan Status w.r.t Term**

****

**Figure 3**

In the above figure 3, the observation is that more no. of applicants with loans for Short Term are likely to get loan approval. Also, the no. of cases of loans for Short Term is higher.

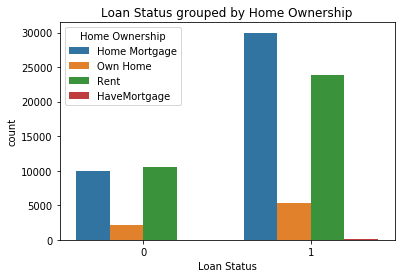
1. **Annual Income V/S Current Loan Amount**



**Figure 4**

The figure 4, shows that upto a certain limit i.e approx 1000000,Current Loan Amount varies proportionaly with Annual Income.Also,after the limit of 6000000 all other data are outliers.

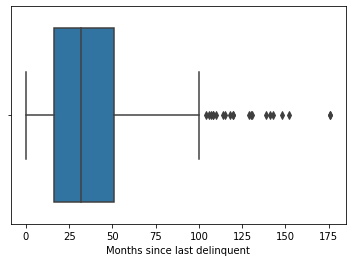
1. **Loan Status w.r.t Home Ownership**



**Figure 5**

The figure 5, shows people with mortgage are more likely to be granted loan than the ones with rented homes.

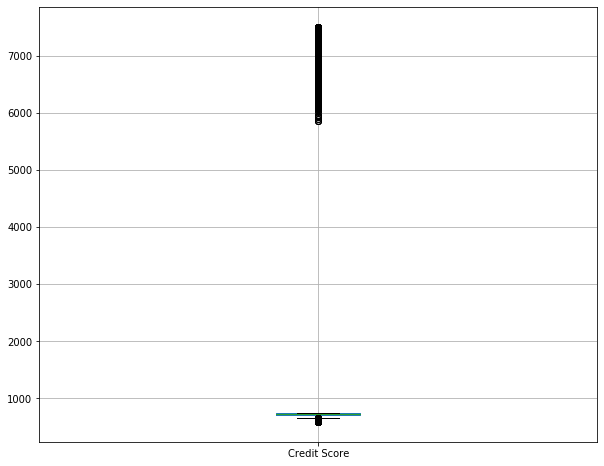
1. **Months since last delinquent**

****

**Figure 6**

In figure 6, X-axis represents the outliers spread in the range of approx 100 to 175.From this box plot we observed that the data is right skewd.

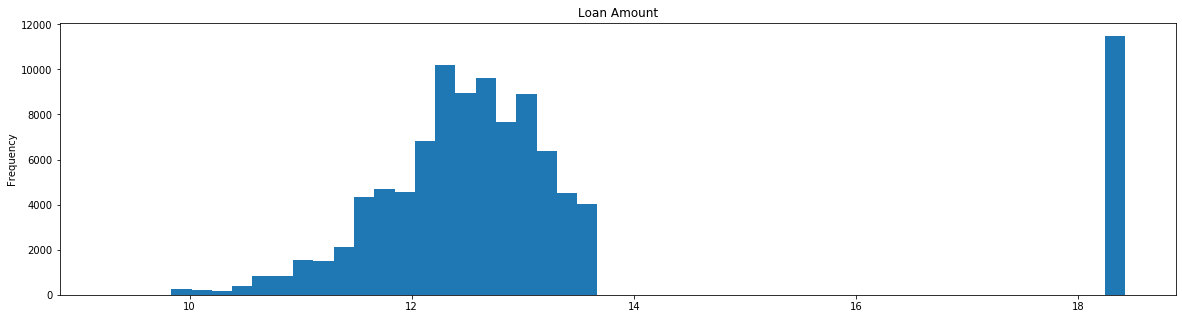
1. **Credit Score**

****

**Figure 7**

From this figure 7,the outliers appear to be in two different range,below the lower quartile and above the upper quartile.From this boxplot it is observed that the data is left skewed.

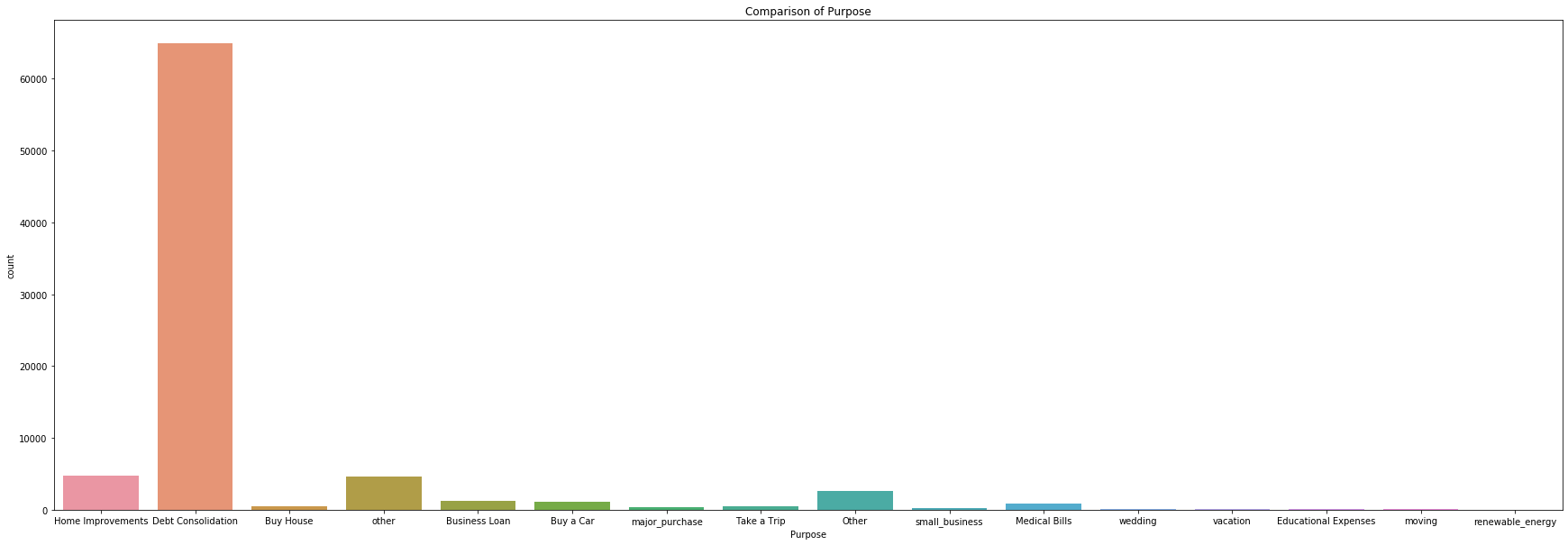
**8.Loan Amount**

****

**Figure 8**

From the figure 8, histogram shows the outliers are present in huge amount for Loan Amount above 10000000.

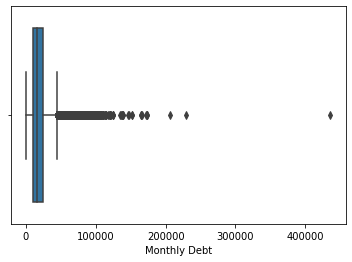
1. **Distinct Categories of Purpose Attribute**



**Figure 9**

From figure 9, it is evident that the category name Debt Consolidation is much more than the other categories. There are categories which are almost negligible as compared to other categories of the attribute.

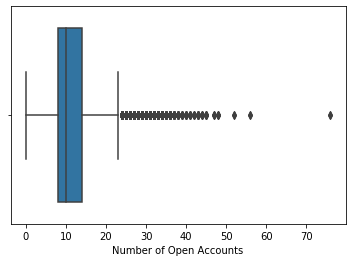
1. **Boxplot of Monthly Debt**

****

**Figure 10**

In the figure 10, all the outliers lie beyond the upper quartile. Data is normally distributed in the inter quartile range.

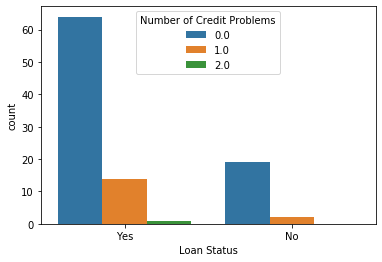
**11 . Number of open Accounts**

****

**Figure 11**

From figure 11, it is observed that data is spread mainly in the range 0 to 30 excepts some outliers lying till 80

1. **. Loan Status Vs. No. of credit problems**

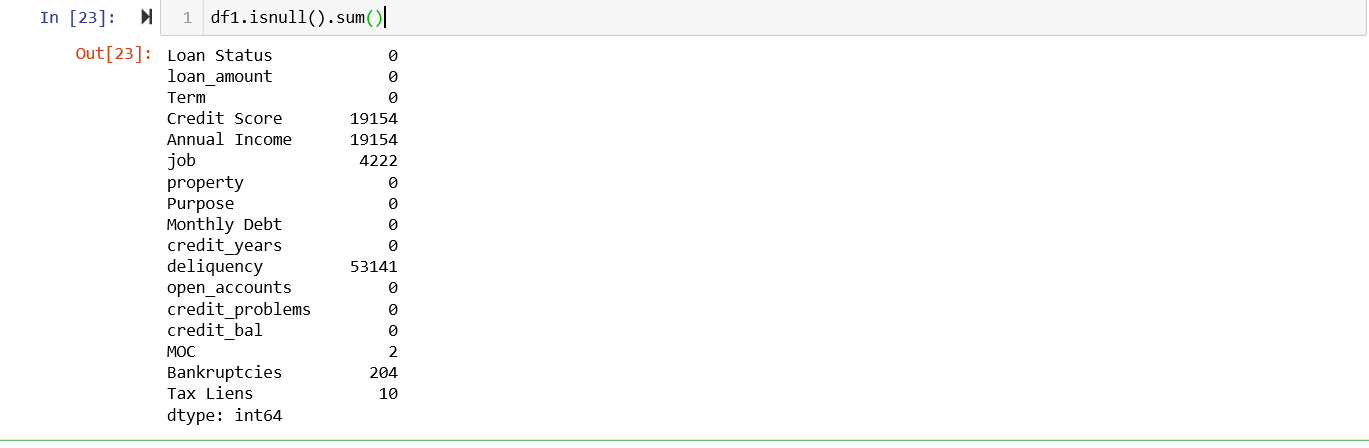
****

**Figure 12**

In the figure 12, count plot shows people with less no. of credit problems are more likely to be granted loan approval.

**Data Preparation**

1. **Finding NaN values**



1. **Handling NaN values**

Imputing NaN Values with statistical parameters:

1. Imputing with MEAN

Since the data distribution is sensitive to mean, replacing the NaN values with mean brings changes to the distribution of data.

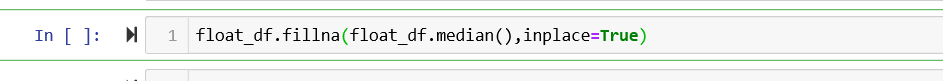
b) Imputing with MEDIAN

Most of the attributes where distribution of data is not normal, NaN is replaced with Median.

Replacing the NaN values of categorical variable with median.



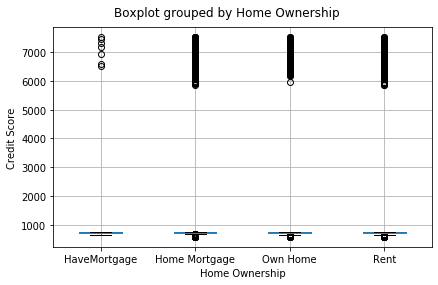
Replacing the NaN values of continuous variables with median.



c) Imputing with MODE

Replacing the continuous variables’ NaN values with mode changes the data for worse and for categorical data replacement of NaN with median showed better results.

1. **Finding Outliers**
2. Using Boxplot

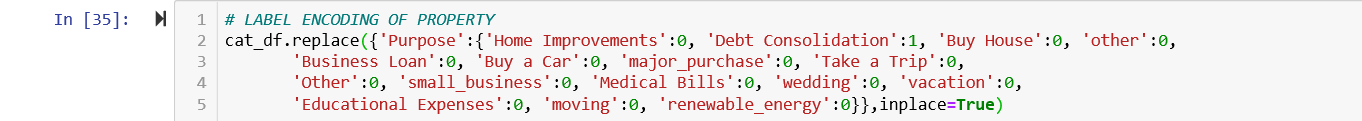


For the boxplot, the black circles represent the outliers whereas the green line is the median.

**Feature Extraction**

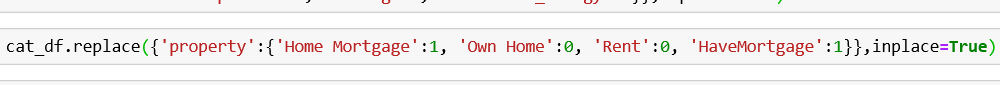
1. **Converting string values to numerical (Encoding)**
2. Label encoding the attribute “Purpose”

“Debt consolidation” category is assigned 1 and rest all other categories are assigned 0 for negligible no. of instances.

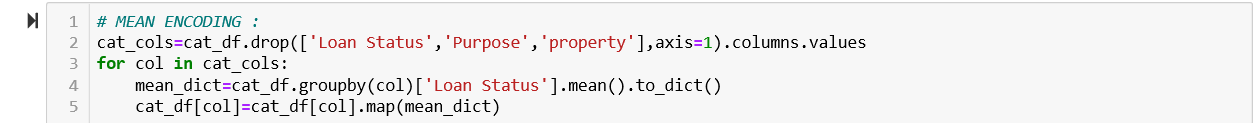
****

1. Label Encoding the attribute “property”

Assigning 1 to property category having Mortgage and rest categories are assigned 0.

****

1. Mean Encoding the categorical variables “property”,”purpose”,”Loan Status”

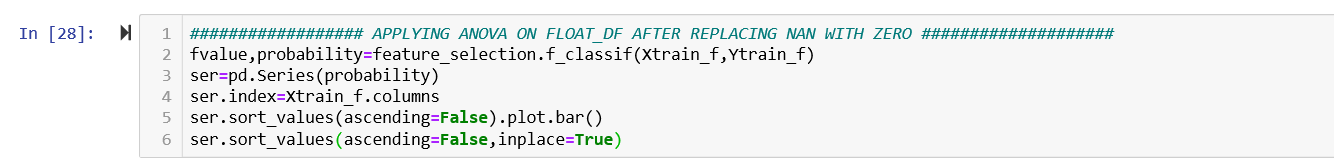
****

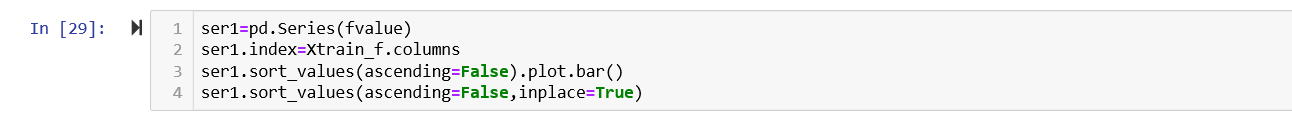
**Feature Selection**

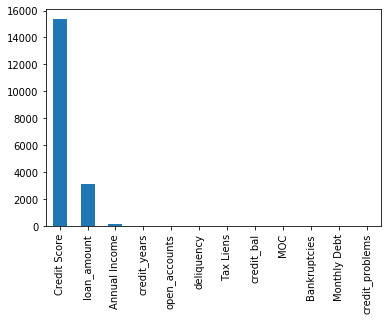
Since the dependent variable in the dataset is categorical,methods to find the impact of continuous and categorical variables on the dependent categorical variable are used.

1. **ANOVA(Analysis of Variance)**

For impact between Categorical and Continuous variable ANOVA is a statistical technique that assesses potential differences in a scale-level dependent variable by a nominal-level variable having 2 or more categories.

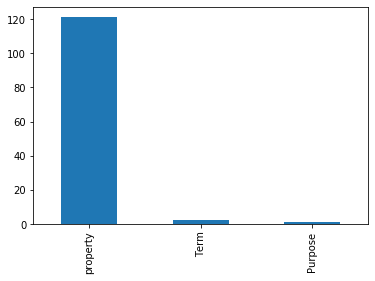
****

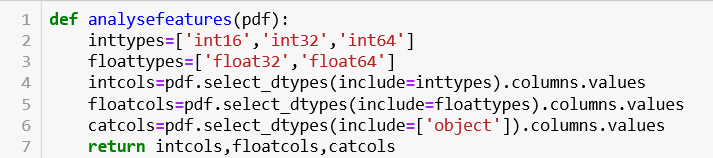
****



1. **CHI2**

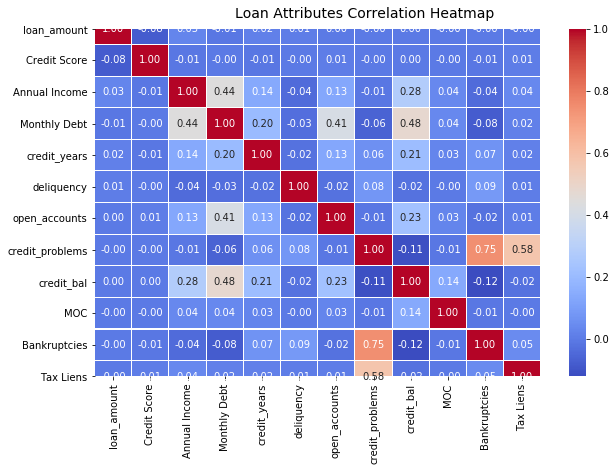
A chi-square test is used in statistics to test the independence of two events. Given the data of two variables, we can get observed count O and expected count E. Chi-Square measures how expected count E and observed count O deviates each other.

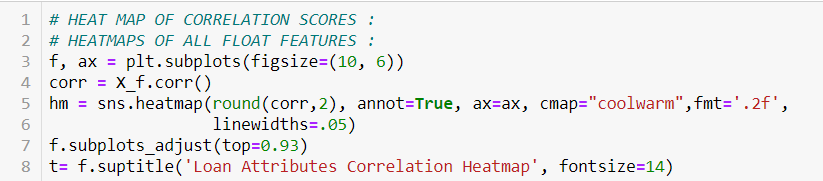
****

****

1. **Correlation**

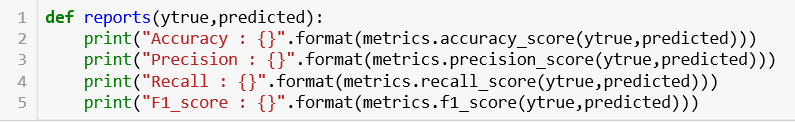
Correlation is a statistical technique that can show whether and how strongly pairs of variables are related.





**Model Building**

**Function made to assess the models using metrics:**

****

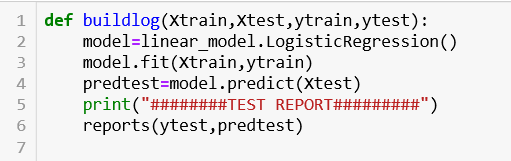
1. **Logistic Regression**

Logistic Regression is used when the dependent variable (target) is categorical. The data set used in this project has dependent variable of categorical type.



Figure (i)

Figure (i) shows the sigmoid function used in logistic regression .It gives value between 0 and 1.Minimum value is constant i.e 0 and maximum value is 1.

****

1. **Decision Tree**

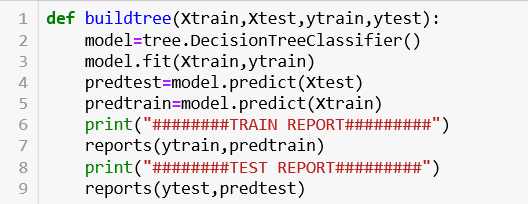
A decision tree is used to visually and explicitly represent decisions and decision making. As the name goes, it uses a tree-like model of decisions.

**Advantages of Decision Tree**

* Simple to understand, interpret, visualize.
* Decision trees implicitly perform variable screening or feature selection.
* Canhandle both numerical and categorical data. Can also handle multi-output problems.
* Decision trees require relatively little effort from users for data preparation.
* Nonlinear relationships between parameters do not affect tree performance.

**Disadvantages of Decision Trees**

* Decision-tree learners can create over-complex trees that do not generalize the data well. This is called over fitting.
* Decision trees can be unstable because small variations in the data might result in a completely different tree being generated*.* This is called [variance](https://medium.com/towards-data-science/balancing-bias-and-variance-to-control-errors-in-machine-learning-16ced95724db), which needs to be lowered by methods like bagging and [boosting](https://towardsdatascience.com/boosting-the-accuracy-of-your-machine-learning-models-f878d6a2d185).
* Greedy algorithms cannot guarantee to return the globally optimal decision tree. This can be mitigated by training multiple trees, where the features and samples are randomly sampled with replacement.
* Decision tree learners create[biased](https://medium.com/towards-data-science/balancing-bias-and-variance-to-control-errors-in-machine-learning-16ced95724db) trees if some classes dominate. It is therefore recommended to balance the data set prior to fitting with the decision tree.

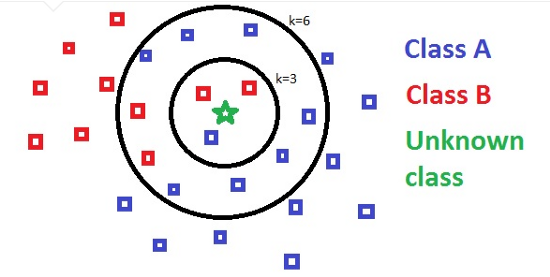
****

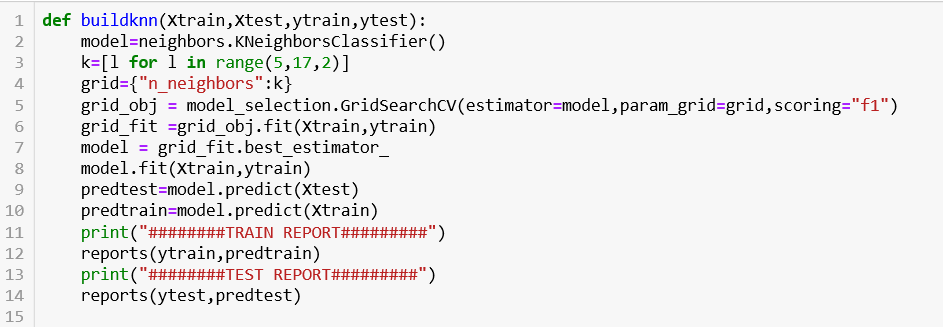
1. **KNN**

KNN (K-Nearest Neighbor) is a simple supervised classification algorithm we can use to assign a class to new data point. It can be used for regression as well, KNN does not make any assumptions on the data distribution, and hence it is non-parametric. It keeps all the training data to make future predictions by computing the similarity between an input sample and each training instance.

**KNN can be summarized as below:**

1. Computes the distance between the new data point with every training example.
2. For computing the distance measures such as Euclidean distance, Hamming distance or Manhattan distance will be used.
3. Model picks K entries in the database which are closest to the new data point.
4. Then it does the majority vote i.e. the most common class/label among those K entries will be the class of the new data point.



****

1. **Naïve Bayes**

Naive Bayes utilizes the most fundamental probability knowledge and makes a naive assumption that all features are independent. Despite the simplicity (some may say oversimplification), Naive Bayes gives a decent performance in many applications.

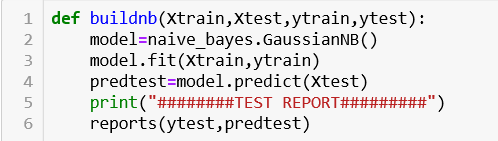
**Strength and Weakness**

**Strength**

* 1. Even though the naive assumption is rarely true, the algorithm performs surprisingly well in many cases
  2. Handles high dimensional data well. Easy to parallelize and handles big data well
  3. Performs better than more complicated models when the data set is small

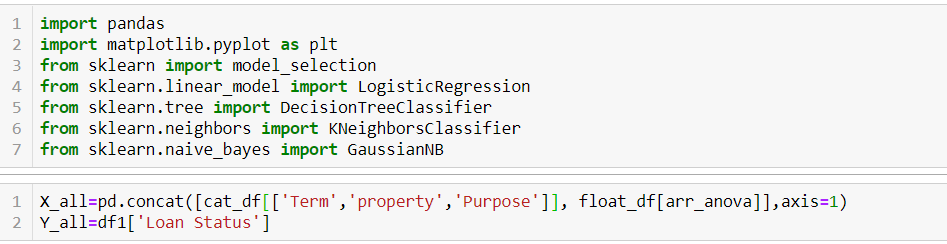
**Weakness**

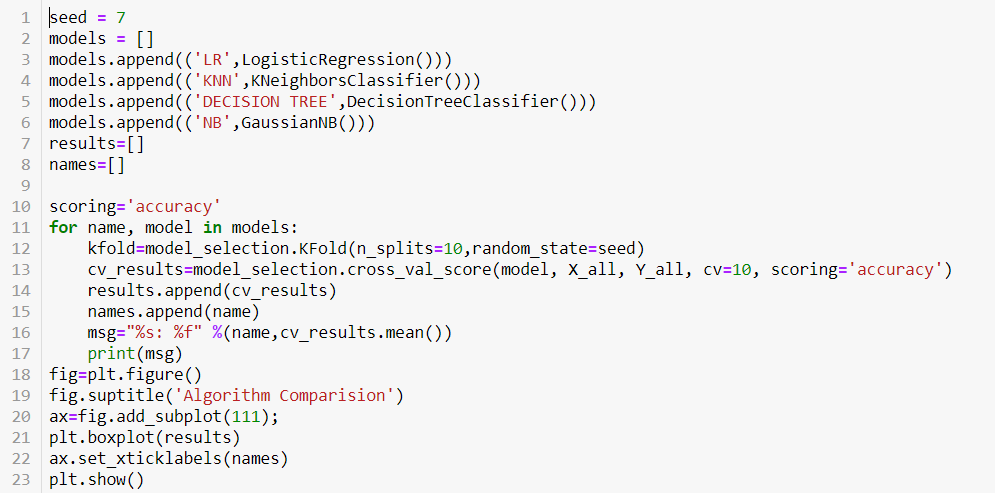
1. The estimated probability is often inaccurate because of the naive assumption. Not ideal for regression use or probability estimation
2. When data is abundant, other more complicated models tend to outperform Naive Bayes

****

1. **K-fold Cross Validation Technique**

When only a limited amount of data is available, to achieve an unbiased estimate of the model performance we use k-fold cross-validation. In the k-fold cross-validation, we divide the data into k subsets of equal size. We build models times, each time leaving out one of the subsets from training and use it as the test set. If k equals the sample size, this is a “leave-one-out” method.

****

****

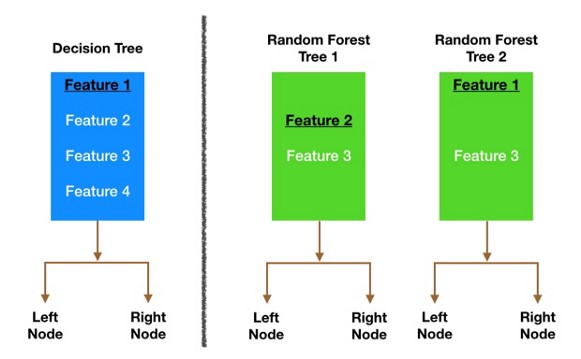
1. **Ensemble Learning**

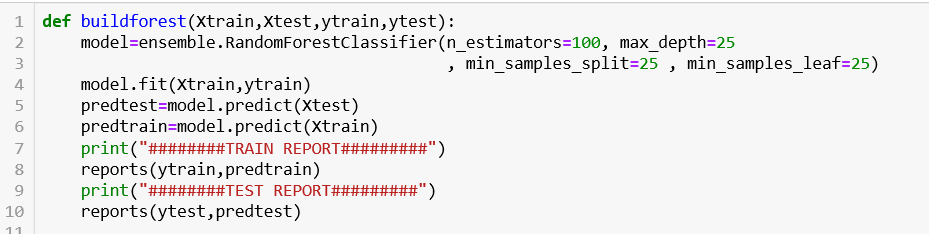
The three most popular methods for combining the predictions from different models are:

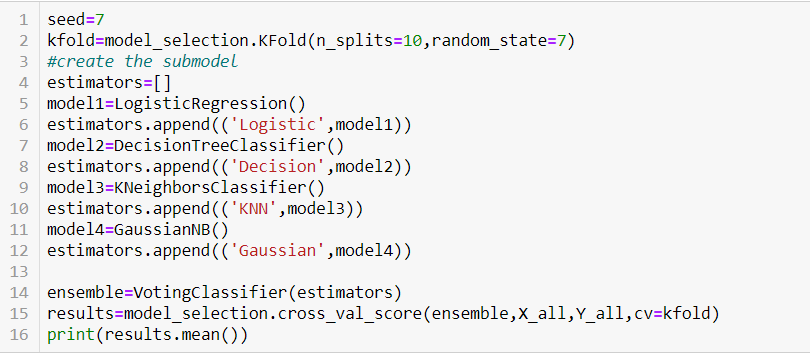
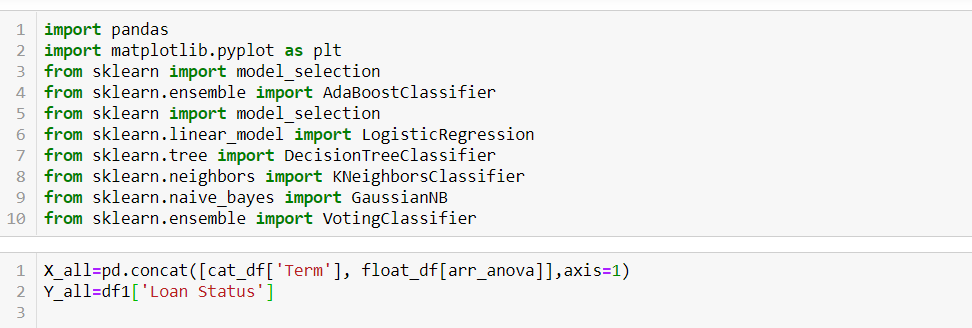
* 1. **Bagging**. Building multiple models (typically of the same type) from different subsamples of the training dataset.
  2. **Boosting**. Building multiple models (typically of the same type) each of which learns to fix the prediction errors of a prior model in the chain.
  3. **Voting**. Building multiple models (typically of differing types) and simple statistics (like calculating the mean) are used to combine predictions.
  4. **Random Forest**

The random forest is a classification algorithm consisting of many decisions trees. It uses bagging and feature randomness when building each individual tree to try to create an uncorrelated forest of trees whose prediction by committee is more accurate than that of any individual tree.What do we need in order for our random forest to make accurate class predictions?

1. We need features that have at least some predictive power**.** After all, if we put garbage in then we will get garbage out.
2. Decision trees of the forest and more importantly their predictions need to be uncorrelated (or at least have low correlations with each other). While the algorithm itself via feature randomness tries to engineer these low correlations for us, the features we select and the hyper-parameters we choose will impact the ultimate correlations as well.

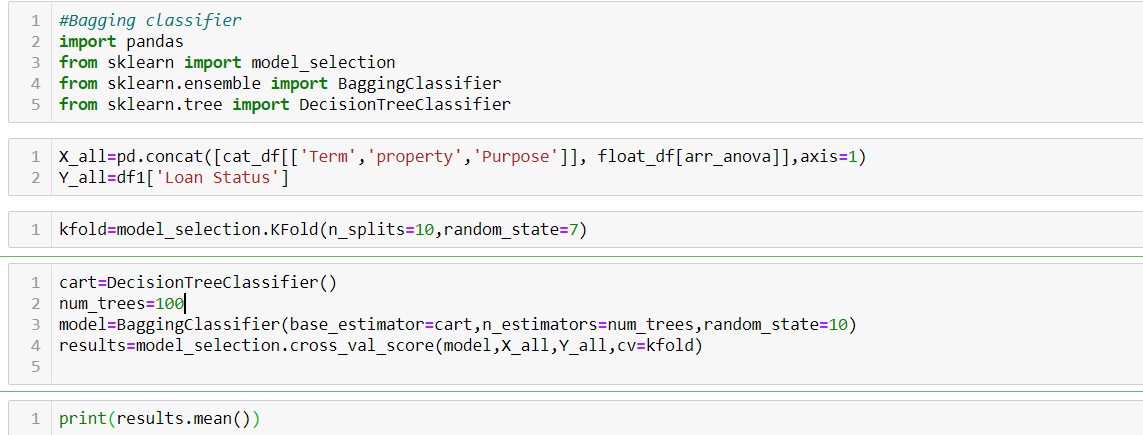


****



**6.1 Bagging**

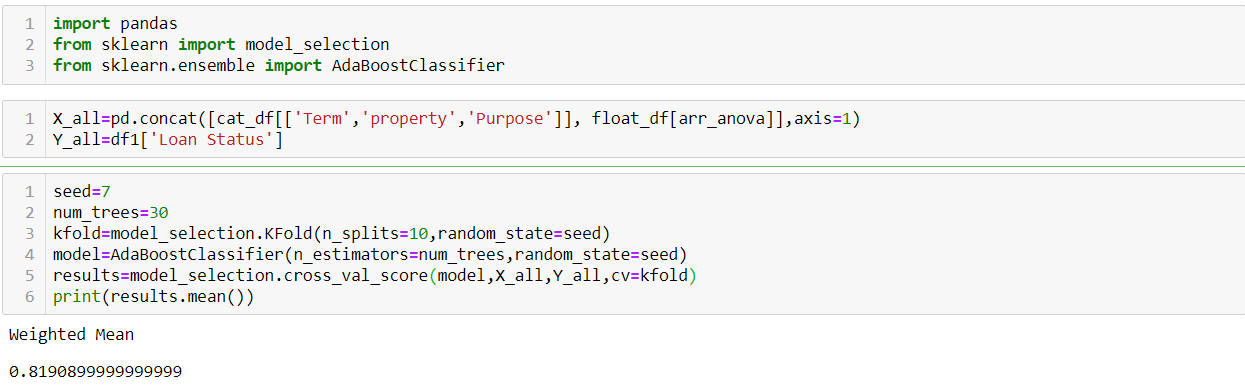
The idea of bagging is simple: we want to fit several independent models and “average” their predictions in order to obtain a model with a lower variance. However, we can’t, in practice, fit fully independent models because it would require too much data. So, we rely on the good “approximate properties” of bootstrap samples (representativity and independence) to fit models that are almost independent.



**6.2 Boosting**

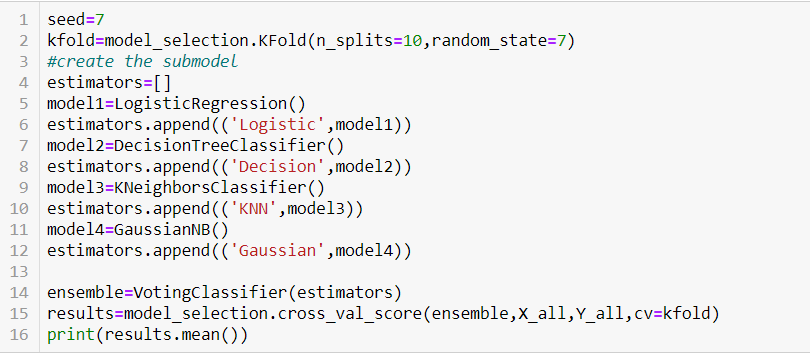
Boosting is a sequential process, where each subsequent model attempts to correct the errors of the previous model. The succeeding models are dependent on the previous model. Let’s understand the way boosting works in the below steps.

1. A subset is created from the original dataset.
2. Initially, all data points are given equal weights.
3. A base model is created on this subset.
4. This model is used to make predictions on the whole dataset.

****

**6.3 Voting**

A Voting Classifier is a machine learning model that trains on an ensemble of numerous models and predicts an output (class) based on their highest probability of chosen class as the output.  
It simply aggregates the findings of each classifier passed into Voting Classifier and predicts the output class based on the highest majority of voting. The idea is instead of creating separate dedicated models and finding the accuracy for each them, we create a single model which trains by these models and predicts output based on their combined majority of voting for each output class.



**Model Evaluation**

**Metrics to judge Prediction Models:**

**Precision**

[Precision](http://en.wikipedia.org/wiki/Information_retrieval#Precision) is the number of True Positives divided by the number of True Positives and False Positives. Put another way, it is the number of positive predictions divided by the total number of positive class values predicted. It is also called the [Positive Predictive Value](http://en.wikipedia.org/wiki/Positive_predictive_value) (PPV).

**Precision = True Positives / (True Positives + False Positives)**

**Recall**

[Recall](http://en.wikipedia.org/wiki/Information_retrieval#Recall) is the number of True Positives divided by the number of True Positives and the number of False Negatives. Put another way it is the number of positive predictions divided by the number of positive class values in the test data. It is also called Sensitivity or the True Positive Rate.

**Recall = True Positives / (True Positives + False Negatives)**

**Accuracy**

It is calculated using the following:

Accuracy = (TN + TP)/(TP + TN + FP + FN)

The accuracy tells that overall how often the model is making a correct prediction.

**F1 Score**

The [F1 Score](http://en.wikipedia.org/wiki/F1_score) is the **2\*((precision\*recall)/ (precision + recall))**. It is also called the F Score or the F Measure. Put another way, the F1 score conveys the balance between the precision and the recall.

**Confusion Matrix**

A clean and unambiguous way to present the prediction results of a classifier is to use a [confusion matrix](https://machinelearningmastery.com/confusion-matrix-machine-learning/) (also called a [contingency table](http://en.wikipedia.org/wiki/Contingency_table)).



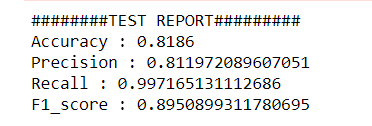
* **The training set** is a subset of the dataset to build predictive models.
* **Test set** or unseen examples is a subset of the dataset to assess the likely future performance of a model. If a model fit to the training set much better than it fits the test set, over fitting is probably the cause.

1. **Logistic Regression Model**

Though, decision tree is said to be the best of all learners this dataset is best tested by logistic regression model. It is a classifier model, working best because the dependent variable in the dataset is categorical.

//CODE//

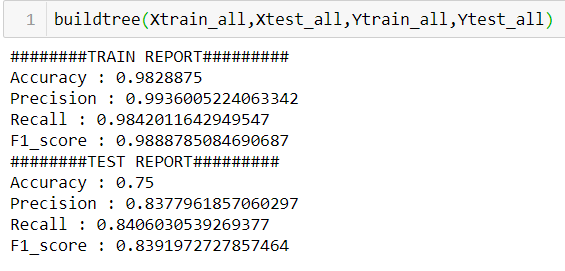
****

****

1. **Decision Tree**

Accuracy is best for Decision Tree model but f1\_score drops as recall and precision also increase.

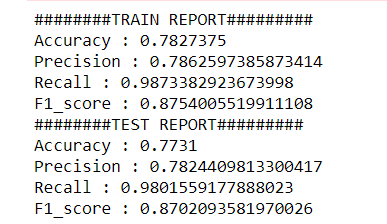
//CODE//

****

1. **KNN**

K-nearest neighbor gives the average performance of all the models applied to this dataset.

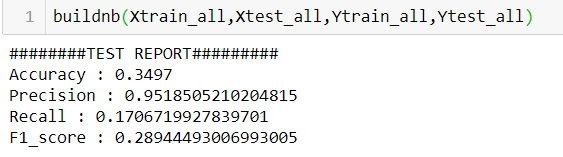
****

****

1. **Naive Bayes**

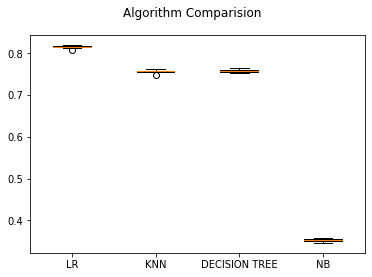
This model is not suitable for this dataset and gives [poor performance.

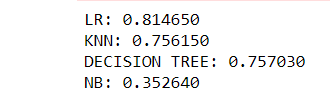
//CODE//

****

1. **K-Fold Cross Validation Technique**

For the dataset given, K-Fold Technique improves the performance for Naïve Bayes model and gives average result for the rest of models.

****

****

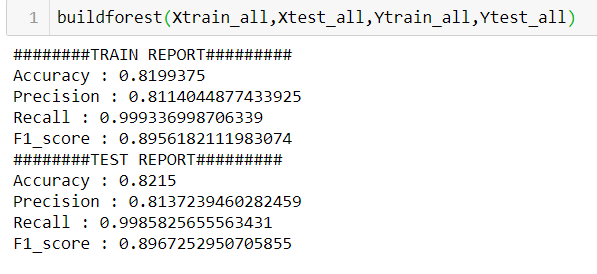
From boxplot above we see the highest performing model is logistic regression model and the poorest performance model is Naïve Bayes Model.

1. **Ensemble Learning**

**6.1 Random Forest**

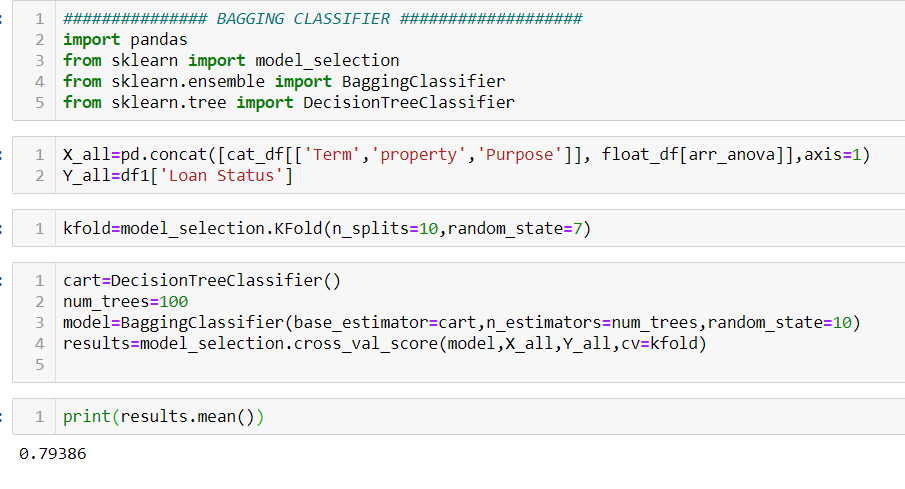
Random forest combines all sections of instances and features separately and repeatedly. For this dataset, random forest gives the best performance.

//CODE//

****

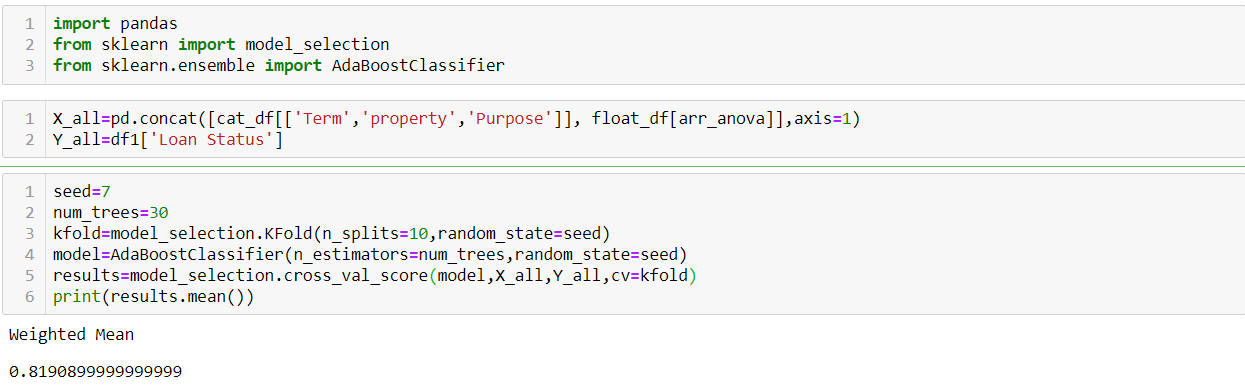
**6.2 Bagging**

This uses homogenous models and gives good performance as compared to simple split models but poor performance than random forest model.

****

**6.3 Boosting**

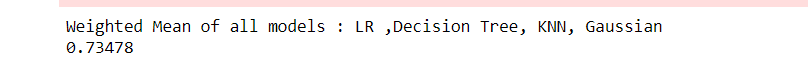
This improves the performance as compared to non boosting models.

****

**6.4 Voting**

Considering all the models, voting takes the weighted mean of all the four models and gives one single score.

In this case, Logistic Regression gives better performance alone but by taking weighted mean the performance is reduced, due to poor performance of Naïve Bayes.



**Proposing the model**

After the evaluation, the random forest model gives the best performance with f1\_score=89%.

**Certificate**

This is to certify that **Ruchika Singh** student of TMSL, Kolkata has successfully completed the project on Prediction of Loan Approval using Machine Learning under the guidance of **Kaushik Ghosh**.

**Signature (Teacher)**

**ACKNOWLEDGEMENT**

I would like to express my special thanks of gratitude to my teacher Kaushik Ghosh who gave me the golden opportunity to do this wonderful project on the topic Prediction of Loan Approval, which also helped me in doing a lot of Research and I came to know about so many new things I am really thankful to them.Secondly, I would also like to thank my friends who helped me a lot in finalizing this project within the limited time frame.