**PREDICTIVE ANALYTICS**

**HOME LOAN APPROVAL PREDICTION**

**Submitted by**

**Nakka Snigdha (19MBMB05)**

**Santhosh M (19MBMB28)**

**Reddy Vishwaja (19MBMB33)**

**Sravanthi Putrevu (19MBMA10)**

**Perumalla Chandrakanth (19MBMA57)**

**Under the guidance of**

**Dr. Varsha Mamidi**



**SCHOOL OF MANAGEMENT STUDIES**

**UNIVERSITY OF HYDERABAD**

**List of Contents**

**Contents Page numbers**

Abstract 1

1. Introduction 1
   1. Machine Learning for Banking 1
   2. Managing Credit risk in Banking 1
2. Background 1-2
   1. Software used 1
   2. Dataset used 2
3. Methodology 2-16
   1. Hypothesis Generation 2-3
   2. Univariate Analysis 3-6
   3. Bivariate Analysis 6-8
   4. Data Pre-processing 9
   5. Feature Engineering 10-11
   6. Model Evaluation and Confusion Matrix 11-16
      1. Logistic Regression without Cross validation 12-13
      2. Logistic Regression with Cross validation 13-14
      3. K Nearest Neighbour 14-15
      4. Support Vector Machine 15
      5. Gaussian Naïve Bayes 16
4. Conclusion 17-18
5. Future Scope 18

References 19

**List of Figures**

Figure 1: Univariate Analysis of Target Variable 3

Figure 2: Univariate Analysis of Categorical Feature Variables 4

Figure 3: Univariate Analysis of Ordinal Feature Variables 4

Figure 4: Univariate Analysis of Numerical Feature Variable (ApplicantIncome) 5

Figure 5: Univariate Analysis of Numerical Feature Variables (Segregated by Education) 5

Figure 6: Univariate Analysis of Numerical Feature Variable (CoapplicantIncome) 5

Figure 7: Univariate Analysis of Numerical Feature Variable (LoanAmount) 6

Figure 8: Univariate Analysis of Numerical Feature Variable (Loan\_Amount\_Term) 6

Figure 9: Bivariate Analysis of **Categorical Independent Variable vs Target Variable 7**

Figure 10: Bivariate Analysis of **Numerical Independent Variable vs Target Variable 8**

Figure 11: Effect of log transformation 9

Figure 12: Distribution of Total\_Income 10

Figure 13: Distribution of Total\_Income (after log transformation) 10

Figure 14: Distribution of EMI 11

Figure 15: Distribution of Balance Income 11

Figure 16: Confusion Matrix of Logistic Regression without Cross validation 12

Figure 17: ROC Curve 14

Figure 18: Confusion Matrix of KNN 14

Figure 19: Confusion Matrix of SVM 15

Figure 20: Confusion Matrix of Gaussian NB 16

Figure 21: Comparison of Accuracies before Cross validation 17

Figure 22: Comparison of Accuracies after Cross validation 18

**List of Tables**

Table 1: Description of data attributes 2

Table 2: Classification Metrics of Logistic Regression without Cross validation 13

Table 3: K fold Cross validation Accuracy 13

Table 4: Classification Metrics of KNN 15

Table 5: Classification Metrics of SVM 15

Table 6: Classification Metrics of Gaussian NB 16

Table 7: Comparison of Accuracies before Cross validation 17

Table 8: Comparison of Accuracies after Cross validation 18

***Abstract:***

*With the enhancement in the banking sector, lots of people are applying for home loans, but the bank has its limited assets which it has to grant to a limited people, so finding out to whom the loan can be granted which will be a safer option for the bank is a typical process. This is done by mining the data of the previous records of the people to whom the loan was granted before and on the basis of these records/experiences the machine can be trained using the machine learning model which gives the most accurate result. If the loan approval process is automated, it can save many working hours and improve the speed of service to the customers. The increase in customer satisfaction and savings in operational costs are also significant. However, the benefits can only be reaped if the bank has a robust model to accurately predict which customer's home loan it should approve and which to reject, to minimize the risk of loan default. The main objective here is to predict whether assigning the loan to a particular person will be safe or not.*

**1. Introduction**

**1.1 Machine Learning for Banking**

Loan prediction is a widespread real-life problem that every retail bank faces in their lending operations. Dream Housing Finance company deals in all home loans. They have a presence across all urban, semi-urban and rural areas. Customers first apply for a home loan after that company validates the customer eligibility for a loan. The company wants to automate the loan eligibility process in real-time based on customer detail provided while filling the online application form. These details are Gender, Marital Status, Education, Number of Dependents, Income, Loan Amount, Credit History and others.

**1.2 Managing Credit risk in Banking**

Banks’ fundamental business model rely on financial intermediation by raising finance and lending (mortgage, real estate, consumer and companies’ loans), the latter is the primary source of credit risk composed from 2 main points loan approval and fraud. Credit scoring in retail portfolios reflects the default risk of a customer at the moment of the loan application; it helps to decide whether to accept or reject credit application based on three primary input data:

Customer information**:** Age, Gender, Marital status, Education, Employment, Geographical (Urban/Rural).

Credit information: Total amount, Loan term.

Credit history: Payment history and delinquencies (payment delays).

**2. Background**

**2.1 Software used**

Python, Anaconda (JUPYTER Notebook)

The JupyterNotebook is an open-source web application that allows to create and share documents that contain live code, equations, visualizations and narrative text.

**2.2 Dataset used**

The training dataset is now used in the machine learning model, based on this dataset, the model is trained. Every new applicant detail filled at the time of application form acts as a test data set. After the operation of testing, the model predicts whether the new applicant is a fit case for approval of the loan or not.

|  |  |  |
| --- | --- | --- |
| Variable Description | | Datatype |
| Loan\_ID | Unique Loan ID | object |
| Gender | Male/ Female | object |
| Married | Applicant married (Y/N) | object |
| Dependents | Number of dependents | object |
| Education | Applicant Education (Graduate/Undergraduate) | object |
| Self\_Employed | Self-employed (Y/N) | object |
| ApplicantIncome | Applicant income | int64 |
| CoapplicantIncome | Co-applicant income | float64 |
| LoanAmount | Loan amount in thousands | float64 |
| Loan\_Amount\_Term | Term of the loan in months | float64 |
| Credit\_History | Credit history meets guidelines | float64 |
| Property\_Area | Urban/ Semi-Urban/ Rural | object |
| Loan\_Status | Loan approved (Y/N) | object |

**Table 1: Description of dataset attributes**

**3. Methodology**

This is a classification problem where we have to predict whether a loan will be approved or not. Specifically, it is a binary classification problem where we have to predict either one of the two classes given, i.e. approved (Y) or not approved (N). Another way to frame the problem is to predict whether the loan will likely to default or not if it is likely to default, then the loan would not be approved, and vice versa. The dataset we will be working on has 615 rows & 13 columns. The dependent variable or target variable is the Loan\_Status, while the rest are independent variable or features. We need to develop a model using the features to predict the target variable. We will need to pre-process the data, perform data cleaning & feature engineering and finally will be implementing models like K-NN, Logistic Regression, Naive Bayes and SVM to check the accuracy of each model.

**3.1 Hypothesis Generation**

Hypothesis Generation is the process of listing out all the possible factors that can affect the outcome, i.e. which of the features will have an impact on whether a loan will be approved or not. Some of the hypothesis may seem intuitive while others may not. Each of the following hypothesis is validated based on the dataset;

Education - Applicants with higher education level i.e. graduate level should have higher chances of loan approval.

Income: Applicants with higher income should have more chances of loan approval.

Loan amount: If the loan amount is less, the chances of loan approval should be high.

Loan term: Loans with shorter time period should have higher chances of approval.

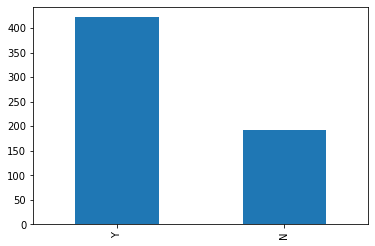
Previous credit history: Applicants who have repaid their previous debts should have higher chances of loan approval.

Monthly instalment amount: If the monthly instalment amount is low, the chances of loan approval should be high.

## 3.2 Univariate Analysis

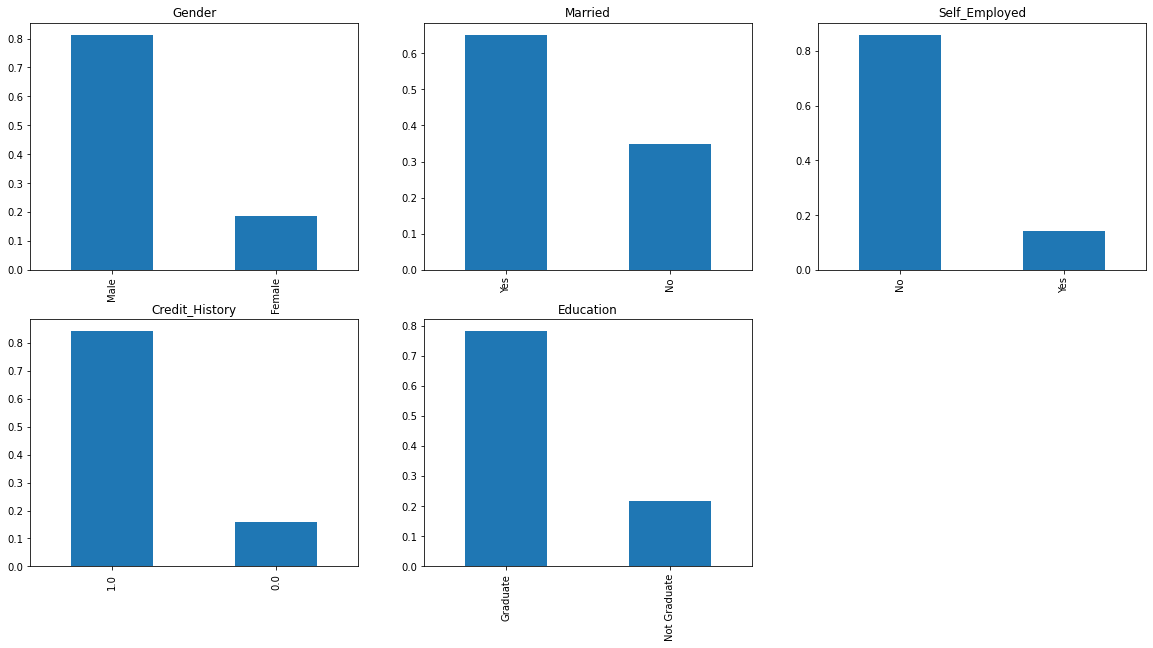
Univariate analysis is when we analyse each variable individually. For categorical features we can use frequency table or bar plots which will calculate the number of each category in a particular variable. For numerical features, a histogram or a box-plot can be used to look at the distribution of the variable. With a histogram, you can check the central tendency, variability, modality, and kurtosis of a distribution. Note that a histogram can’t show you if you have any outliers. This is why we also use box-plots.

To examine the target variable, i.e., Loan\_Status, which is a categorical variable, its frequency table, percentage distribution and bar plot were analysed. It can be noted that the loan of 422 (around 69%) people out of 614 was approved. There is no imbalanced classes issue in this dataset, thus accuracy as an evaluation metric should be appropriate. On the other hand, if there are imbalanced or skewed classes, then we might need to use precision and recall as evaluation metrics.

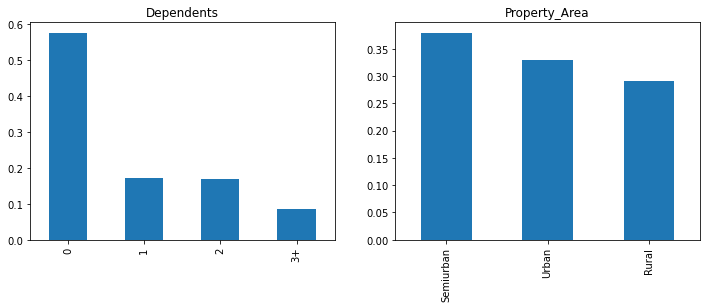


**Figure 1: Univariate Analysis of Target Variable**

There are five features that are categorical or binary (Gender, Married, Self\_Employed, Credit\_History, Education), whose bar graphs were analysed and It was inferred that; 80% applicants in the dataset are male, around 65% of the applicants in the dataset are married, around 15% applicants in the dataset are self-employed, around 85% applicants have credit history (repaid their debts) and around 80% of the applicants are graduates.

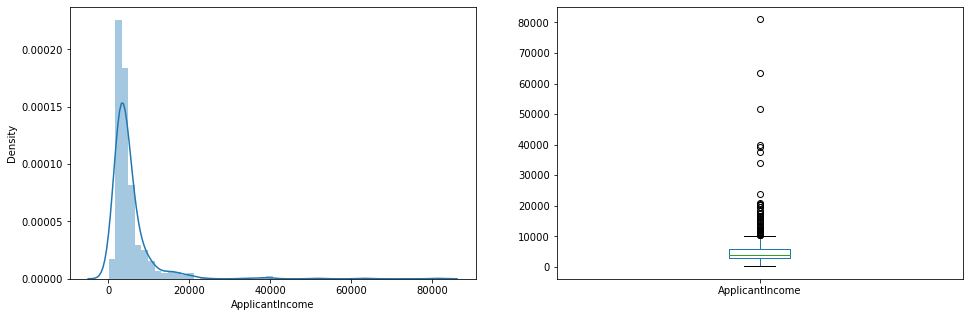
 **Figure 2: Univariate Analysis of Categorical Feature Variables**

Two features are ordinal variables having some order involved (Dependents, Property\_Area). The inferences made from their bar plots are, more than half of the applicants don’t have any dependents and most of the applicants are from Semiurban area.



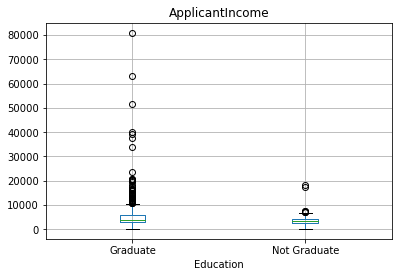
**Figure 3: Univariate Analysis of Ordinal Feature Variables**

Four features are Numerical (ApplicantIncome, CoapplicantIncome, LoanAmount, Loan\_Amount\_Term). From the Applicant income distribution, It can be inferred that most of the data in the distribution of applicant income is towards left which means it is not normally distributed. The distribution is right-skewed (positive skewness), which needs to be converted into a normally distribution.



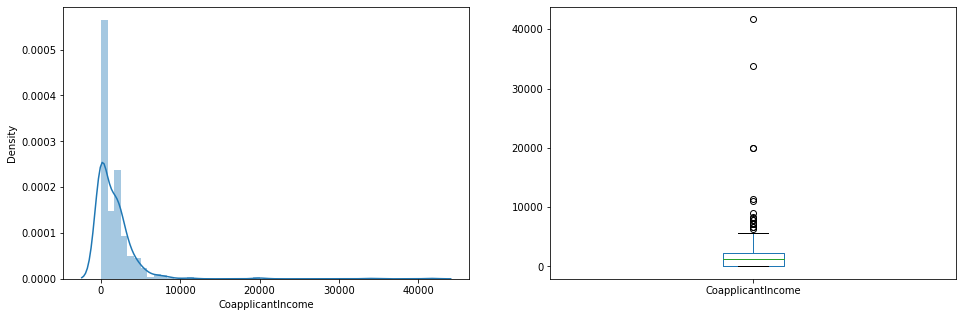
**Figure 4: Univariate Analysis of Numerical Feature Variable (ApplicantIncome)**

The boxplot confirms the presence of a lot of outliers/extreme values. This can be attributed to the income disparity in the society. Part of this can be driven by the fact that we are looking at people with different education levels. The following graph shows segregation by education.



**Figure 5: Univariate Analysis of Numerical Feature Variables (Segregated by Education)**

It was observed that there is a higher number of graduates with very high incomes, which is appearing to be the outliers. Secondly, the Co-applicant income distribution is a similar distribution as that of the applicant income. Majority of co-applicant’s income ranges from 0 to 5000 and a lot of outliers in the co-applicant income were observed too.

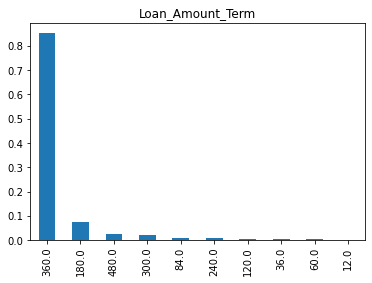
**Figure 6: Univariate Analysis of Numerical Feature Variable (CoapplicantIncome)**

The LoanAmount is however a fairly normal distribution (albeit still slightly right-skewed) but there are lot of outliers in this variable.



**Figure 7: Univariate Analysis of Numerical Feature Variable (LoanAmount)**

Lastly, the distribution of Loan\_Amount\_Term variable is observed to be a discrete variable and around 85% of the loans are 360 months term or 30 years period.



**Figure 8: Univariate Analysis of Numerical Feature Variable (Loan\_Amount\_Term)**

## 3.3 Bivariate Analysis

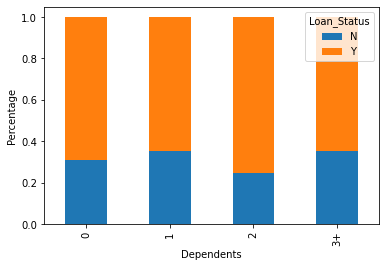
## After looking at every variable individually in univariate analysis, bivariate analysis can be used to test the hypotheses that we generated earlier by exploring them again with respect to the target variable.

## Firstly, a relation was found between target variable and categorical independent variables with the help of a stacked bar plot, which will give us the proportion of approved and unapproved loans.

## 

## 

## 



**Figure 9: Bivariate Analysis of Categorical Independent Variable vs Target Variable**

From the bar charts above, it can be inferred that;

* The proportion of male and female applicants is more or less same for both approved and unapproved loans
* The proportion of married applicants is higher for the approved loans
* The distribution of applicants with 1 or 3+ dependents is similar across both the categories of Loan\_Status
* There is nothing significant we can infer from Self\_Employed vs Loan\_Status plot.
* The proportion of loans getting approved for graduates is higher compared to non-graduates
* It seems people with credit history as 1 are more likely to get their loans approved
* The proportion of loans getting approved in semiurban area is higher as compared to that in rural or urban areas.

Secondly, by analysing numerical independent variables with respect to target variable and looking at the correlation between all the numerical variables excluding NA values using pearson correlation coefficient and using heat map to visualize the correlation, the most correlated values were observed to be; (ApplicantIncome - LoanAmount) with correlation coefficient of 0.57, (Credit\_History - Loan\_Status) with correlation coefficient of 0.56, LoanAmount is also correlated with CoapplicantIncome with correlation coefficient of 0.19.



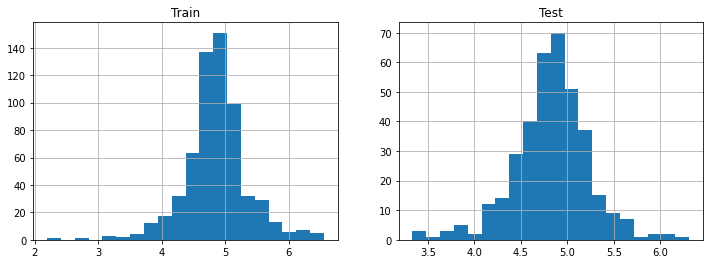
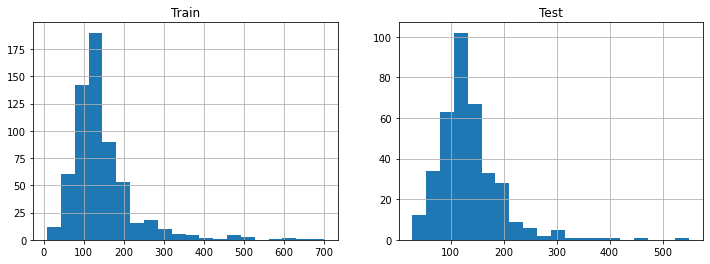
**Figure 10: Bivariate Analysis of Numerical Independent Variable vs Target Variable**

**3.4 Data Pre-processing**

Data pre-processing is a data mining technique that involves transforming raw data into an understandable format. Real-world data is often incomplete, inconsistent, and/or lacking in certain behaviours or trends, and is likely to contain many errors. Data pre-processing is a method of resolving such issues.

There are missing values in Gender, Married, Dependents, Self\_Employed, LoanAmount, Loan\_Amount\_Term and Credit\_History features. We can consider filling the missing values for numerical variables through imputation using mean or median and for categorical variables using mode. It was observed that in loan amount term variable, the value of 360 is repeating the most, so the missing values in this variable are replaced using the mode of the variable. For the LoanAmount variable median was used to impute the missing values as earlier it was observed that loan amount had outliers so the mean will not be the proper approach as it is highly affected by the presence of outliers. Similarly, to replace the missing values in Test set the mode[/median/mean](https://colab.research.google.com/drive/1rN0QM1aDg3iJNY0o8IZG51oVD8Pa8iP0) of the Training set was considered to avoid data leakage to the test set.

Due to the presence of outliers a bulk of the data in the loan amount is at the left and the right tail is longer (positively skewed). Log transformation was used to treat this, as it does not affect the smaller values much, but reduces the larger values and a distribution similar to normal distribution is obtained. Similar changes were done to the test file simultaneously and effect of extreme values has been significantly subsided.



**Figure 11: Effect of log transformation**

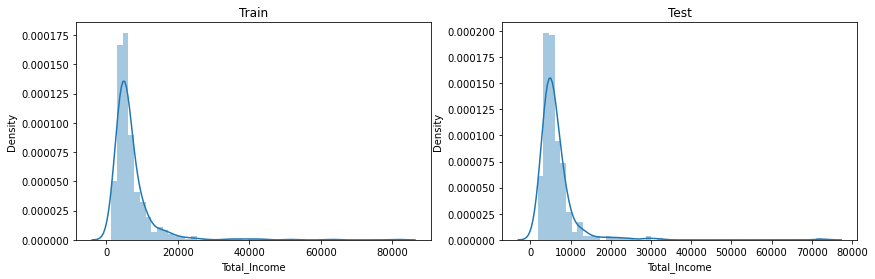
**3.5 Feature Engineering**

Based on the domain knowledge, the following three new features that might affect the target variable were created;

Total Income - As discussed during bivariate analysis Applicant Income and Coapplicant Income were combined, because if the total income is high, chances of loan approval might also be high.

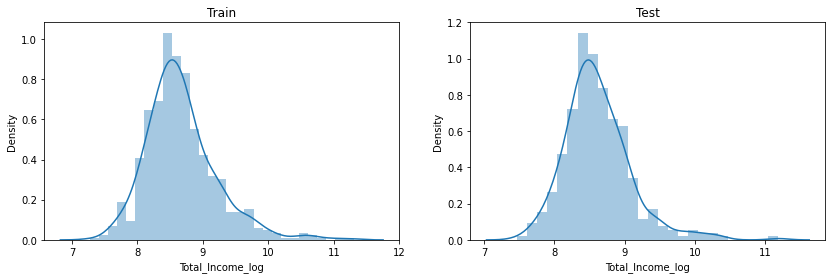
Equated Monthly Installment - EMI is the monthly amount to be paid by the applicant to repay the loan, calculated by taking the ratio of loan amount with respect to loan amount term. Idea behind making this variable is that people who have high EMIs might find it difficult to pay back the loan.

Balance Income - This is the income left after the EMI has been paid. Idea behind creating this variable is that if this value is high, the chances are high that a person will repay the loan, hence increasing the chances of loan approval.

****

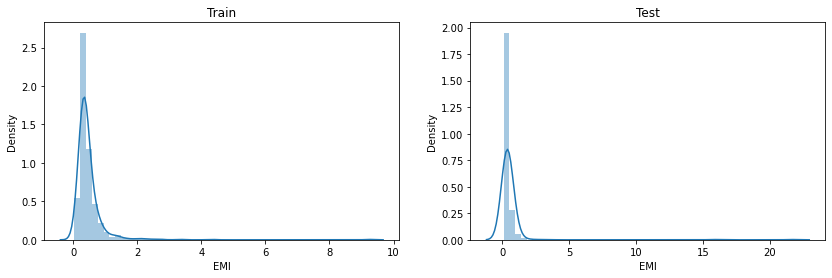
**Figure 12: Distribution of Total\_Income**

We can see it is shifted towards left, i.e., the distribution is right skewed. To make the distribution normal, log transformation is applied so that the distribution looks much closer to normal and effect of extreme values has been significantly subsided.

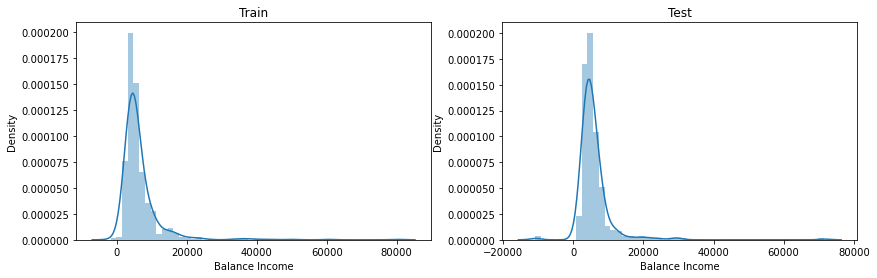
****

**Figure 13: Distribution of Total\_Income (after log transformation)**

Similar plots were created for EMI and Balance income andthe variables which were used to create these new features were droped. Reason for doing this is, the correlation between those old features and these new features will be very high and logistic regression assumes that the variables are not highly correlated. Also, noise can be removed from the dataset by removing "ApplicantIncome", "CoapplicantIncome", "LoanAmount", "Loan\_Amount\_Term" as they are already represented by "Total\_Income", "EMI" and "Balance Income".



**Figure 14: Distribution of EMI**



**Figure 15: Distribution of Balance Income**

**3.6 Model Evaluation and Confusion Matrix**

The process of model building is not complete without evaluation of model’s performance, as it helps to decide whether the predictions are accurate. When the results are plotted and compared with the actual values, and the distance between the predictions and actual values is calculated, less distance signifies higher accuracy of the predictions. Since this is a classification problem, the models (Logistic Regression, KNN, Naive Bayes and SVM) were evaluated using metrics; accuracy, precision, recall, specificity and ROC curve.

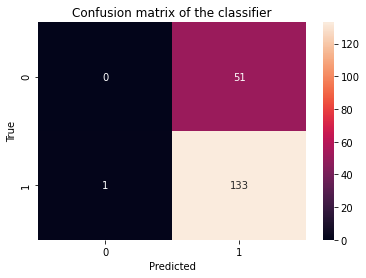
* + 1. **Logistic Regression without Cross validation**

Logistic Regression is a classification algorithm. It is used to predict a binary outcome (1 / 0, Yes / No, True / False) given a set of independent variables. It is an estimation of Logit function, which is a log of odds in favour of the event. This function creates a s-shaped curve with the probability estimate, which is very similar to the required step wise function.

While performing logistic regression, the variable Loan\_ID was dropped as it does not have any effect on the loan status and similar changes were made to the test dataset. The library scikit-learn (sklearn) was used for making different models as it is one of the most efficient tool which contains many inbuilt functions that can be used for modelling. Sklearn requires the target variable in a separate dataset, so, the target variable from the train dataset is dropped and saved in another dataset. Dummy variables are generated for the categorical variables using pandas get\_dummies function.

The model was then trained on training dataset to make predictions for the test dataset by dividing the train dataset into two parts; train and validation. The train part was used to train the model and predictions were made for the validation part. The train\_test\_split function from sklearn was used to divide the train dataset. The functions LogisticRegression and accuracy\_score was imported from sklearn to fit the logistic regression model.

The model prediction was about 72% accurate, i.e. the model identified 72% of the loan status correctly. On evaluating the model with the confusion matrix, an accuracy of 72%, precision of 72% and recall of 99% were observed.



**Figure 16: Confusion Matrix of Logistic Regression without Cross validation**

precision recall f1-score support

0 0.00 0.00 0.00 51

1 0.72 0.99 0.84 134

accuracy 0.72 185

macro avg 0.36 0.50 0.42 185

weighted avg 0.52 0.72 0.61 185

**Table 2: Classification Metrics of** **Logistic Regression without Cross validation**

* + 1. **Logistic Regression with Cross validation**

To check how robust the model is with unseen data Cross validation can be used. It is a technique which involves reserving a particular sample of a dataset on which model training was not done. Later, this sample is used to test the model before finalizing it. The common method for validation is k-fold cross validation, where stratification is the process used for rearranging the data so as to ensure that each fold is a good representative of the whole. It is generally a better approach when dealing with both bias and variance. If K=N, then it is called Leave one out cross validation, where N is the number of observations.

Here, Cross validation logistic model was made with stratified 5 folds and predictions were made for test dataset. In stratified k-fold, each fold contains roughly the same proportions of the different types of class labels. Previously, train\_test\_split was used to split the data, here, StratifiedKFold was used to split the data. The mean validation accuracy for this model turned out to be 0.72. The ROC curve was visualized for the same.

1 of kfold 5

accuracy\_score 0.7886178861788617

2 of kfold 5

accuracy\_score 0.6910569105691057

3 of kfold 5

accuracy\_score 0.6666666666666666

4 of kfold 5

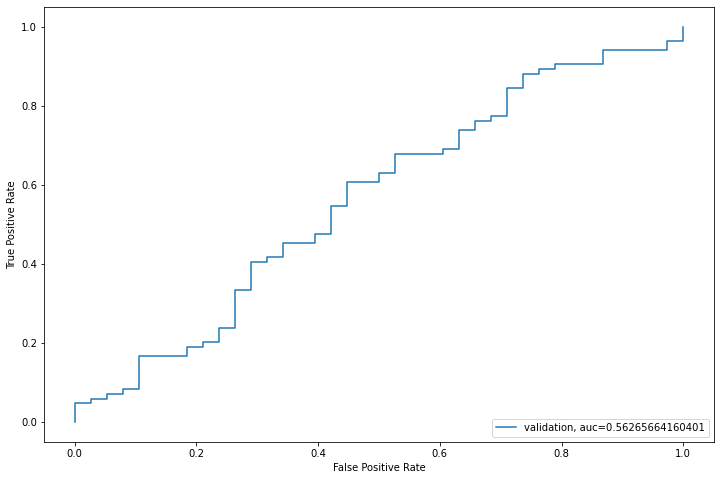
accuracy\_score 0.7804878048780488

5 of kfold 5

accuracy\_score 0.680327868852459

Mean validation accuracy: 0.7214314274290283

**Table 3: K-fold Cross validation Accuracy**

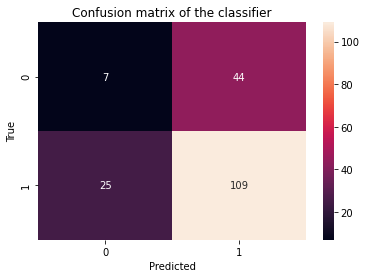


**Figure 17: ROC Curve**

Finally, the accuracy before validation for this Logistic Regression model was observed to be 66.9% before cross validation, and 75.28% after cross validation.

* + 1. **K Nearest Neighbour**

K-NN algorithm basically creates an imaginary boundary to classify the data. When new data points come in, the algorithm will try to predict that to the nearest of the boundary line. Therefore, larger k value means smother curves of separation resulting in less complex models. Whereas, smaller k value tends to overfit the data and resulting in complex models. The accuracy before validation for this KNN model was observed to be 72.03% and after the cross validation the accuracy was 60.14%.



**Figure 18: Confusion Matrix of KNN**

precision recall f1-score support

0 0.22 0.14 0.17 51

1 0.71 0.81 0.76 134

accuracy 0.63 185

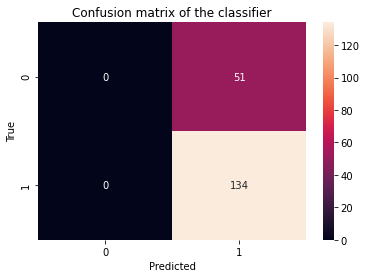
macro avg 0.47 0.48 0.46 185

weighted avg 0.58 0.63 0.60 185

**Table 4: Classification Metrics of** **KNN**

**3.6.4 Support Vector Machine**

Support vectors are data points that are closer to the hyperplane and influence the position and orientation of the hyperplane. Using these support vectors, we maximize the margin of the classifier. The accuracy before validation for this SVC model was 67.37 % and after the cross validation the accuracy was 66.66%.

****

**Figure 19: Confusion Matrix of SVM**

precision recall f1-score support

0 0.00 0.00 0.00 51

1 0.72 1.00 0.84 134

accuracy 0.72 185

macro avg 0.36 0.50 0.42 185

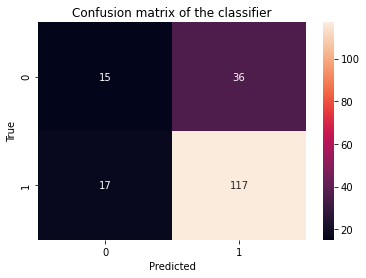
weighted avg 0.52 0.72 0.61 185

**Table 5: Classification Metrics of** **SVM**

**3.6.5 Gaussian Naive Bayes**

Naive Bayes classifier assumes that the effect of a particular feature in a class is independent of other features. Here, the loan applicant is classified desirable or not depending on his/her income, previous loan, location etc. Even if these features are interdependent, these features are still considered independently. This assumption simplifies computation, and that's why it is considered as naive. This assumption is called class conditional independence. Gaussian Naive Bayes is a variant of Naive Bayes that follows Gaussian normal distribution and supports continuous data.

The accuracy before validation for this Gaussian NB model was observed to be 79.49% and after the cross validation the accuracy was 59.42%.



**Figure 20: Confusion Matrix of Gaussian NB**

precision recall f1-score support

0 0.47 0.29 0.36 51

1 0.76 0.87 0.82 134

accuracy 0.71 185

macro avg 0.62 0.58 0.59 185

weighted avg 0.68 0.71 0.69 185

**Table 6: Classification Metrics of** **Gaussian NB**

**4. Conclusion**

Total Income, Balance Income and EMI are the most important factors for predicting the class of the loan applicant (whether the applicant would be ‘approved’ or ‘not approved’). In near future this module of prediction can be integrated with the module of automated processing system. The system was trained on old training dataset, in future software can be made such that new testing data can be used after certain time.

After trying and testing four different algorithms, the best accuracy was achieved by Logistic Regression (0.75), followed by SVC (0.67), KNN (0.6) and Gaussian NB (0.59). On the whole, logistic regression classifier provides the best result in terms of accuracy for the given dataset, without any feature engineering needed. Because of its simplicity and the fact that it can be implemented relatively easy and quick, Logistic Regression is often a good baseline that data scientists can use to measure the performance of other more complex algorithms.

The outputs of accuracies of all the four model before cross validation, and visualization of comparison of these accuracies is shown below;

Mean validation accuracies of four algorithms are listed below

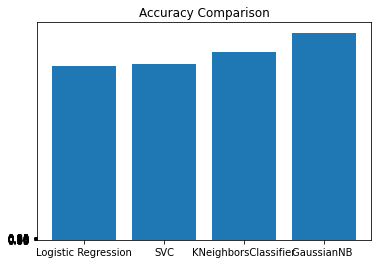
Logistic Regression : 66.9

SVC : 67.37

KNeighborsClassifier : 72.03

GaussianNB : 79.49

**Table 7: Comparison of Accuracies before Cross validation**

****

**Figure 21: Comparison of Accuracies before Cross validation**

The outputs of accuracies of all the four model after cross validation, and visualization of comparison of these accuracies is shown below;

kfold accuracies of four algorithms are listed below

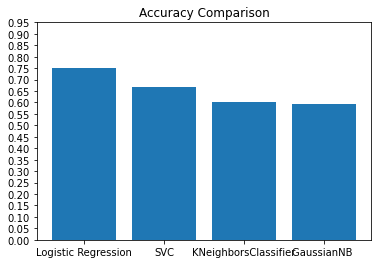
Average of the Logistic Regression = 0.75

Average of the SVC = 0.67

Average of the KNeighborsClassifier = 0.6

Average of the GaussianNB = 0.59

**Table 8: Comparison of Accuracies after Cross validation**

****

**Figure 22: Comparison of Accuracies after Cross validation**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Accuracy before Cross validation | Accuracy after Cross validation | Sensitivity | Specificity |
| Logistic Regression | 66.9 | 75.28 | 0.99 | 0 |
| SVM | 67.37 | 66.66 | 1.0 | 0 |
| KNN | 72.03 | 60.14 | 0.813 | 0.13 |
| Gaussian NB | 79.49 | 59.42 | 0.87 | 0.29 |

**Table 9: Comparison of Model Performance**

**5. Future Scope**

Time Series Analysis can be done using the Loan data of several years, for prediction of the approximate time, when the client can default. Future analysis can be done on predicting the approximate Interest rates that the loan applicant is expected to be charged as per his profile, if his loan is approved. This can be useful for loan applicants, since some banks approve loans, but give very high interest rates to the customer.

An app with proper UI can be built, which will take various inputs from the user like name, address, loan amount, loan duration, etc. and give a prediction of whether their loan application can be approved by the banks or not based on their inputs along with an approximate interest rate.

***References:***

***Machine Learning Repositories:***

*UCI & Kaggle*

***Websites:***

[*www.kdnuggets.com*](http://www.kdnuggets.com)

[*www.analyticsvidhya.com*](http://www.analyticsvidhya.com)

[*www.machinelearningmastery.com*](http://www.machinelearningmastery.com)

[*https://towardsdatascience.com/data-types-in-statistics-347e152e8bee*](https://towardsdatascience.com/data-types-in-statistics-347e152e8bee)

**INDIVIDUAL CONTRIBUTION:**

