Results and Discussions

Comparing machine learning approaches and methodologies to achieve the best prediction will be the main objective. We have selected Decision trees and Random forests as two of the fundamental Machine learning approaches ever known. Using them we will be generating loan default predictions to suggest the financial service providers’ chance to make a more accurate decision. This section includes details of the sample data, dataset structure, and their individual means. Apart from that different data cleaning and pre-processing steps and measures will be reasoned.

Data Sampling:

The process of collecting data from the entire population is a significantly difficult task and when the data is related to loans it is even harder to gather sufficient data to conduct any research.Manually recording the same would take years to compile and the individuals taking loans do not always share anything. Government organizations such as banks or insurance companies will never disclose any details. Even private organizations also do not share any with very few exceptions. While in this study, the data sample was obtained from the Lending Club organization 2015 database release which is the only available valid data from an actual company.

Data Understanding:

This phase can be considered as one of the most vital steps as in this phase, the understanding related to the structure of the data has to be developed which is very much substantial for model development. All this exploration of the data will improve the discovery process of the meaningful information and also helps in the identification of the anomalies in the data as well.

Dataset structure:

Dataset initially had around 111 attributes with each attribute having 42542 data.

For the development of the proposed system, the provided data is divided into two datasets each for the training purposeof the model and for the performance testing of themodel. Both of the datasets have 12 attributes related to the feature of individuals although the testing dataset has no

values in its class variable as they have to be predicted bythe developed model.Furthermore, the attributes are comprisedof five categorical, seven continuousincluding a class variable. Table No. 1 shows the description of each attribute of the dataset.

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Data Type** | **Description** |
| LOAN\_AMNT | Continuous | The listed amount of the loan applied for by the borrower. If at some point in time, the credit department reduces the loan amount, then it will be reflected in this value. |
| TERM | Categorical | The number of payments on the loan. Values are in months and can be either 36 or 60. |
| INT\_RATE | Continuous | Interest Rate on the loan |
| GRADE | Categorical | LC assigned loan grade |
| EMP\_LENGTH | Continuous | Employment length in years. Possible values are between 0 and 10 where 0 means less than one year and 10 means ten or more years. |
| HOME\_OWNERSHIP | Categorical | The homeownership status is provided by the borrower during registration or obtained from the credit report. Our values are: RENT, OWN, MORTGAGE, OTHER |
| ANNUAL\_INC | Continuous | The self-reported annual income provided by the borrower during registration. |
| VERIFICATION\_STATUS | Categorical | Indicates if income was verified by LC, not verified, or if the income source was verified |
| DTI | Continuous | A ratio calculated using the borrower’s total monthly debt payments on the total debt obligations, excluding mortgage and the requested LC loan, divided by the borrower’s self-reported monthly income. |
| OPEN\_ACC | Continuous | The number of open credit lines in the borrower's credit file. |
| TOTAL\_ACC | Continuous | The total number of credit lines currently in the borrower's credit file |
| LOAN\_STATUS | Categorical | Current status of the loan- Fully paid or default [Class Variable] |

Data Cleaning:

This step involves major tasks related to the modification of the data in the proposed study is the conversion of the data types, getting rid of unnecessary and unimportant data, and handling the missing values. As all the categorical attributes with character data types have to be converted into numeric for the purpose of the algorithm application. Moreover, all the missing values in the entire data have to be handled for improved and efficient modelling.

The first step in the modification task is related to the data structure of the categorical attributes. As all the values in the categorical attributes are in the character format which cannot be incorporated during the implementation of the algorithm. To cope with this issue all the values in the categorical attributes are converted to numeric as factors. Once the categorical attribute instances are changed from character to numeric, the next step is to convert the data types of the character attributes to numeric.

The next step in the modification process is related to the handling of the missing values in the dataset. Before the implementation of this task, Table 2 shows the number of missing values with respect to their attribute.

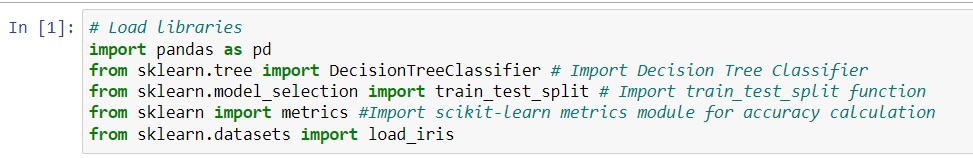
|  |  |
| --- | --- |
| **Attributes** | **Number Of Missing Values** |
| LOAN\_AMNT | 7 |
| TERM | 7 |
| INT\_RATE | 7 |
| GRADE | 1119 |
| EMP\_LENGTH | 7 |
| HOME\_OWNERSHIP | 7 |
| ANNUAL\_INC | 11 |
| VERIFICATION\_STATUS | 7 |
| DTI | 7 |
| OPEN\_ACC | 36 |
| TOTAL\_ACC | 36 |
| LOAN\_STATUS | 7 |

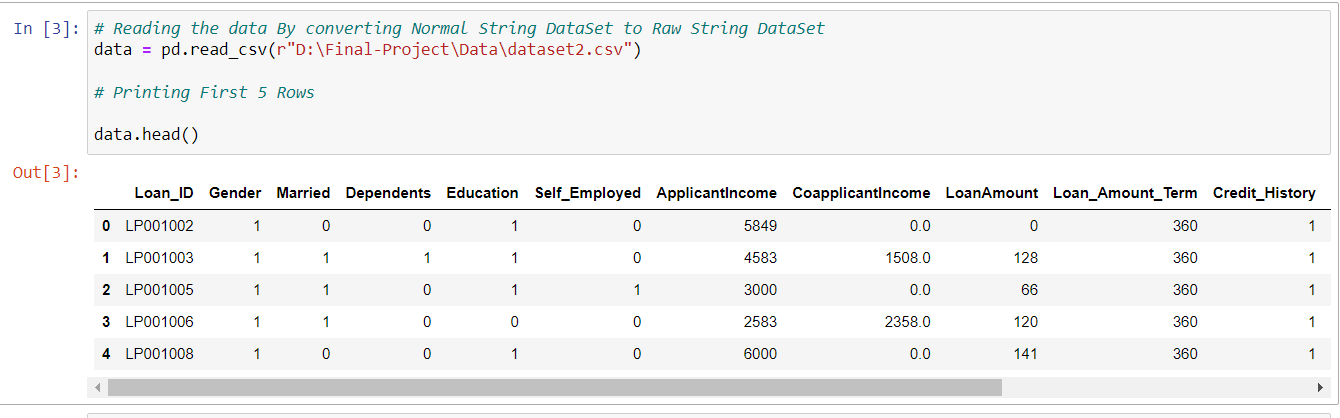
Various techniques are available in SAS to handle these missing values although, in this study, these missing values are handled by the “Median” method. As in this method, each missing value is replaced by the median of that particular attribute and this process continues until all the missing values are handled. In certain cases, we have omitted the entire row with missing values as it will not affect the dataset. (42512)

Models Involved:

**Decision Tree:**

Initially, we processed the dataset and put it up for the test with a decision tree classifier. Firstly, we import all the required packages and read them in the CSV file as shown below.





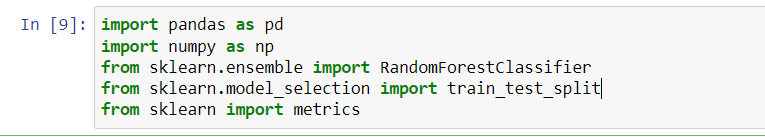
Now we will test the above data, in order to do that we split the data in an 80/20 ratio, where 80% of the data will be fed into the classifier to train it and 20% is kept aside to test its accuracy. The following snippets of code show the same.

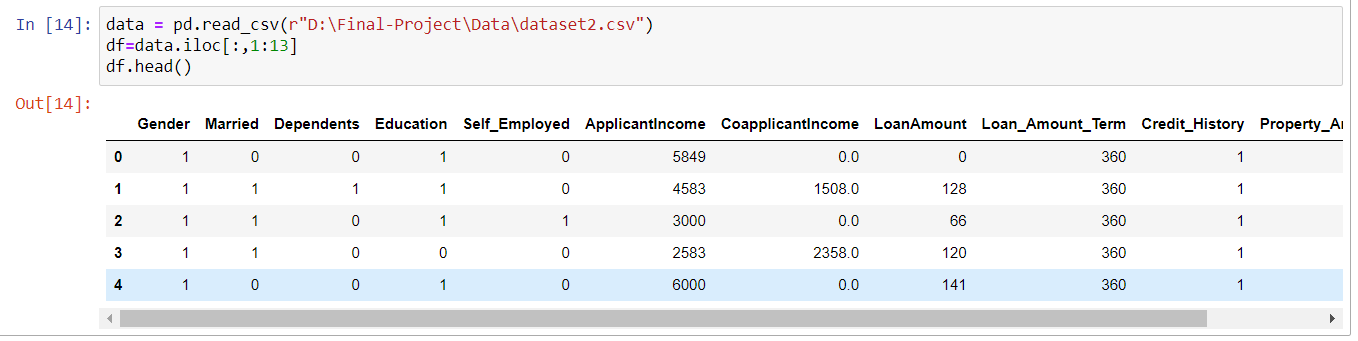


It is evident from the above image the decision tree classifier managed to get an accuracy rate of **74% (0.7411554384)** based on comparing actual test sets output values and predicted values.

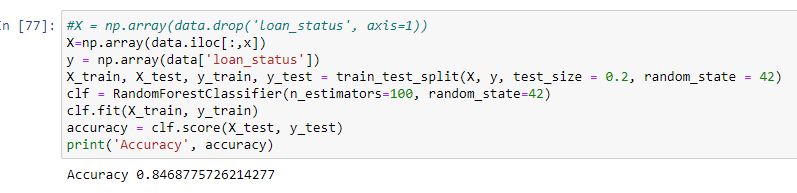
**Random Forest:**

Initially we have processed the dataset and put it up for the test with random forest classifier. Firstly, we import all the required packages and read in the CSV file as shown below.

****

****

Similarly, here also we split the data in an 80/20 ratio, where 80% of the data will be fed into the classifier to train it and 20% is kept aside to test its accuracy. The following snippets of code show the same.

****

Comparing actual test sets output values and predicted values Random Forest classifier managed to get an accuracy rate of **84% (0.8468775726).**

**Model Evaluation:**

Once the model implementation from all the proposed techniques and validation has been satisfied. Testing data was incorporated in each model and the developed model was employed for the prediction of the loan approval by employing the trained model with Decision Tree (DT), and Random Forest (RF).

For assessment parameters, a confusion matrix will be applied for the model performance evaluation purpose. As the confusion matrix is a reliable method to summarize the performance of the classifier. The confusion matrix presents the number of predictions with respect to their correctness as the name suggested it shows how confused the classifier was during the prediction process. The tabular form is used in the confusion matrix (see Fig. 2) for the purpose of the description of the classifier.

Table Confusion Matrix

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **True Class** | | | | |
| **Predicted Class** |  | **Positive** | **Negative** | **Measures** |
| **Positive** | True Positive (TP) | False Positive (FP) | Positive Predictive Value (PPV)= (TP/(TP+FP)) |
| **Negative** | False Negative (FN) | True Negative (TN) | Negative Predictive Value (NPV)= (TN/(TN+FN)) |
| **Measures** | Sensitivity  TP/(TP+FN) | Specificity  TN/(FP+TN) | Accuracy  (TP+TN)/(TP+FP+TN+FN) |

Table Confusion Matrix For Decision Tree Classifier

|  |  |  |  |
| --- | --- | --- | --- |
| **True Class** | | | |
| **Predicted Class** |  | **Positive** | **Negative** |
| **Positive** | 7486 | 1498 |
| **Negative** | 1253 | 391 |

Table Confusion Matrix For Random Forest

|  |  |  |  |
| --- | --- | --- | --- |
| **True Class** | | | |
| **Predicted Class** |  | **Positive** | **Negative** |
| **Positive** | 7180 | 31 |
| **Negative** | 1271 | 21 |

Another important assessment is the classification report which tells us about the precision, recall also known as sensitivity, f1-score. Precision denotes the percentage of correct outputs among all the returned outputs. Recall denotes the percentage of correct outputs among all the outputs that should be returned. F1-score is the harmonic mean of precision and recall. Classification report for both decision tree and random forest is shown below picture.

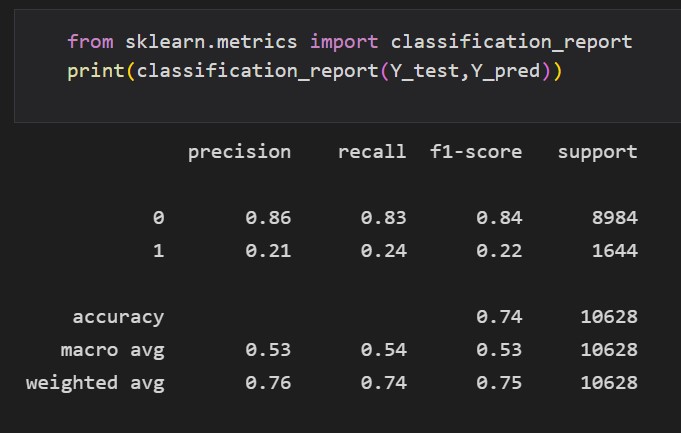


Figure Classification Report For Decision Tree

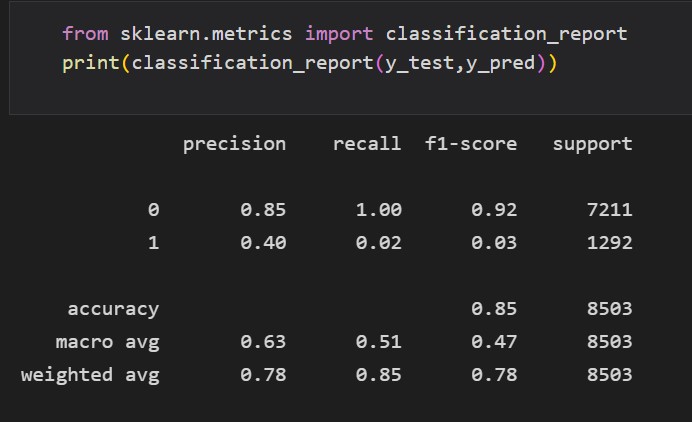


Figure Classification Report For Random Forest

From all the above evaluations and recorded results, it is evident that random forest has a greater accuracy in predicting the loan status over decision tree. In each and every respect random forest has a better chance of predicting whether the loan will be defaulted or not.

Future Scope:

So far, with the limited data we have gathered, we have utilized the same to compare two important Machine learning algorithms. Judging by all respect possible, we found out that Random Forest has greater accuracy in predicting Loan status which implies loads will be paid or will be defaulted. We think that this basic recommendation system can solve the purpose to an extent. As of now we only propose the idea and procedure as a future possibility. However, as the data we required is not available at present implementation will not be possible. But maybe in the future, it can be implemented.

Now the broader idea behind the proposed recommendation system is that the system will take specific details about the loan and return a value or an amount up to which the system suggests to the lender that up to that limit the borrower will be able to pay back the loaned amount. Above that limit that it would not be possible for the borrower to pay back, simply the borrower will default. The value that the system will return can be considered a threshold value or amount. So far the knowledge we gathered from bank and lending companies’ policies is that they provide the loan to the borrower based on the valuation of the stake, giving the borrower a percentage of the valuation predefined by the lender. However, they never judge whether the borrower can pay back the loaned amount. Even though the lenders know about the financial condition and other loans if there are any, still they lend the money. We suggest that lenders should judge the borrower’s payback capacity based on certain criteria which will be fed into the recommendation system to give the output.

Based on the above information we propose the following procedure:

1. Take the recommendation criteria as inputs
2. Get the LTV ratio(percentage) as per the selected loan type [equation 1]
3. Calculate the maximum possible loan amount that the lender able to lend
4. Calculate the maximum possible payback amount the borrower can return
5. Render the features from the calculated and input data and set them into the ML algorithm
6. Based on the suggested threshold amount offer the loan to the borrower

The recommendation criteria contain several information about the borrower.Firstly, the type or purpose of the loan. A loan can be of many types such as home loans, agricultural loans, business loans, etc. Based on the type,the LTV ratio is calculated. The second criterion is the type of account in the bank. It can either be a savings or a current account. Savings for personal usage and current for business purposes and each has different loan approval criteria. The next criterion is the income details or payslip etc. to get the income amount. Next are ITR files from which expenditures will be calculated. Lastly stake valuation amount. Now all of this sums up as a dataset for a borrower which will be used to make the suggestion. Bank may require other documents to process the loan. But for our proposed system this data is only relevant for the recommendation.

Now next we need the LTV ratio calculation which will be based on the bank’s predefined rates for different loan purposes. A loan-to-value (LTV) ratio in a loan is the percentage of the stake’s value that a bank or financial institution can lend to a property buyer.The formula used is as follows:

LTV Ratio (%) = (Amount Borrowed/Stake’s Value) x 100 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_(1)

Now using this LTV ratio, we will get the maximum possible amount that the bank will be able to lend the borrower.

Now, the important task is to collect the data from income details and ITR files to get the income and gross expenditure that the individual uses up in his/her dealings for day-to-day purposes for personal needs and other purposes such as policy premiums, and tax, subscription bills, etc. From here we calculate the amount left with the individual to pay the loan premiums resulting in calculating the maximum possible payback amount.

Now we first add the calculated and derived data into the dataset that we discussed at the very onset of the Experimentation which will be passed along with the initial data inputs and use the same to run with the machine learning algorithms. With this, the threshold amount is generated as the suggestion to the lender whether to lend or not lend the amount to the borrower.

We studied through different relevant work but it seems out future scope is something promising if right implemented.