**A DATA SCIENCE PROJECT ON LOAN STATUS PREDICTION**

**SUBMITTED BY-**

**ASHUTOSH MISHRA**

**DATA TRAINED EDUCATION**

Kindly find herewith attached my Github link to check the solution of this project- <https://github.com/ashu5436/Evaluation-Projects/blob/main/Loan%20Application%20Status%20Prediction%20(Evaluation%20Project-6).ipynb>

1. **PROBLEM STATEMENT**

Now-a-days most of the people are pounding the doors of financial institutes in-order to borrow the loans for their necessity but these financial institutes takes a lot of time while making their decisions as they need to perform the due diligence and verifications for the same. Hence the role of the financial institutes are really challenging because a single mistake from their end can lead to a big ‘NO’ to the customers those are hoping that their loan application would definitely get accepted. Also, since these institutes are getting n-numbers of cases on a daily basis as there is high demand for loans in the current market scenario so it would be getting tougher for them to analyze and decide the same. Hence Data science comes as a vital tool here to make it easy for these institutes to make their predictions on the basis of the Data Analytics, Visualizations and Machine Learning Algorithms.

**The problem statement of this project is to build a model that can predict whether the loan of the applicant will be approved or not on the basis of the details provided in the dataset.**

**2. Data Analysis**

Data Analysis is the process of collecting, cleaning, sorting, and processing raw data to extract relevant and valuable information to help businesses. With the help of data analysis we can find-out some important relations amongst the variables/attributes and which could play a vital role in decision making for businesses, entities or an individual.

This project consist of **614** rows and **13** variables/columns. Out of **13** variables there are **12** independent variables and **1** Target variable (Dependent).

|  |  |  |
| --- | --- | --- |
| FEATURES | | TARGET VARIABLE |
| Numerical Variables | Categorical Variables |
| Loan\_ID | Applicant\_Income | Loan\_Status |
| Gender | Coapplicant\_Income |
| Married | Loan\_Amount |
| Dependents | Loan\_Amount\_Term |
| Education | Credit\_History |
| Self\_Employed |  |
| Property\_Area |  |

|  |  |
| --- | --- |
| Attributes | Null Values |
| Loan\_ID | 0 |
| Gender | 13 |
| Married | 3 |
| Dependents | 15 |
| Education | 0 |
| Self\_Employed | 32 |
| Applicant Income | 0 |
| Coapplicant Income | 0 |
| Loan Amount | 22 |
| Loan\_Amount\_Term | 14 |
| Credit\_History | 50 |
| Property\_Area | 0 |
| Loan\_Status | 0 |
| **Total Null Values** | **149** |

|  |  |
| --- | --- |
| Attributes | Unique Values |
| Loan\_ID | 614 |
| Gender | 2 |
| Married | 2 |
| Dependents | 4 |
| Education | 2 |
| Self\_Employed | 2 |
| Applicant Income | 505 |
| Coapplicant Income | 287 |
| Loan Amount | 203 |
| Loan\_Amount\_Term | 10 |
| Credit\_History | 2 |
| Property\_Area | 3 |
| Loan\_Status | 2 |

From the **Descriptive Statistics** I observed that attributes- Loan Amount Term, Gender, Married, Credit History and Loan Status have more median than their respective mean, which indicates that data might **skewed** left hand side as well and also the interquartile difference for Applicant Income, Co-applicant Income, Loan Amount, Loan\_Amount\_Term are varying too much hence there might be possibility that **outliers** are present in the data set.

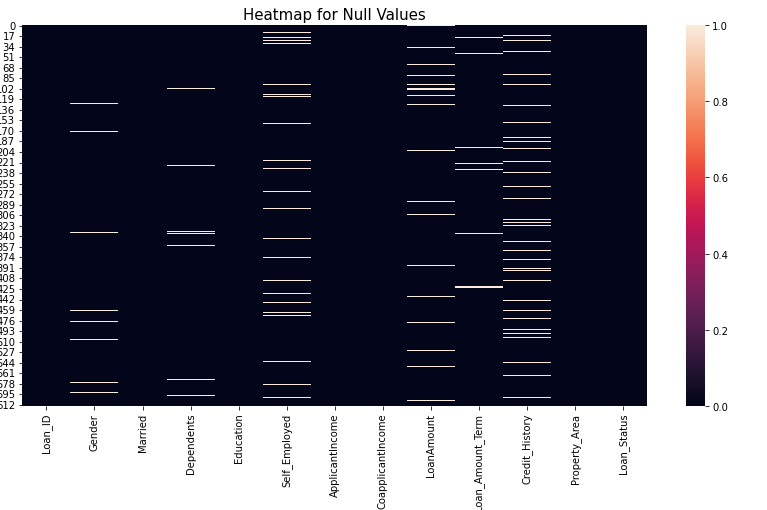
Also, the **Correlation** w.r.t. the target variable **Loan\_Status** is attached as below-

|  |  |
| --- | --- |
| Attributes | Correlation w.r.t. Loan\_Status |
| Coapplicant Income | -0.089189 |
| Education | -0.085884 |
| Loan Amount | -0.033214 |
| Loan\_Amount\_Term | -0.022549 |
| Applicant Income | -0.004710 |
| Self\_Employed | -0.003700 |
| Dependents | 0.010118 |
| Loan\_ID | 0.011773 |
| Gender | 0.017987 |
| Property\_Area | 0.032112 |
| Married | 0.091478 |
| Credit\_History | 0.540556 |
| Loan\_Status | 1.000000 |

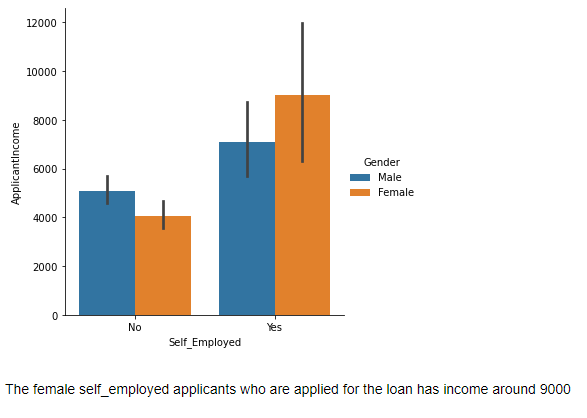
See, there are six attributes having negative values and it’s showing us that there is negative correlation w.r.t. **Loan\_Status** (which is our target variable). Hence will remove all the negative correlated columns as the columns are very less negative and some of the columns are almost zero correlated w.r.t. target variable. As we can see that variable **Credit History** is the mostimportant variable while predicting the loan status as it is **54%** positively correlated w.r.t. target variable which is highest amongst all other Feature variables.

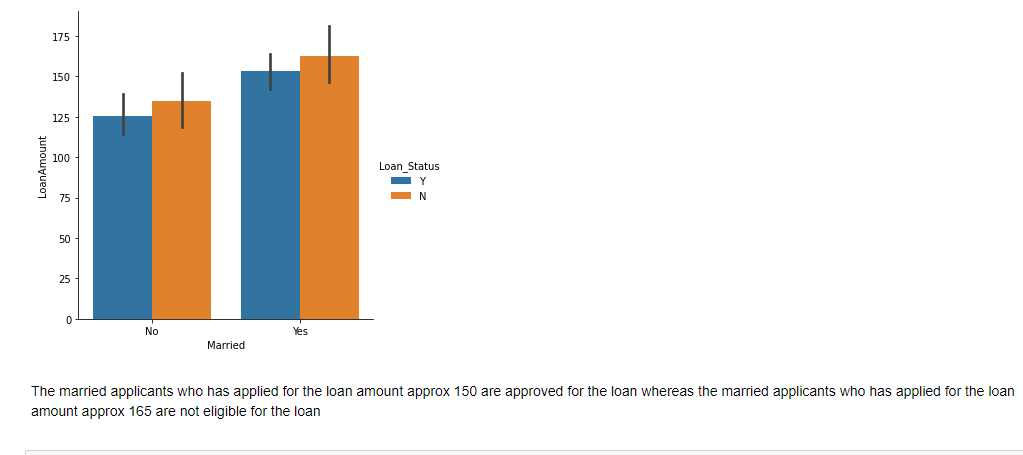
Negative correlation means if input is Positive then output would be Negative and vice-versa. Whereas, Positive correlation means if input is Positive then output would also be Positive and vice-versa.

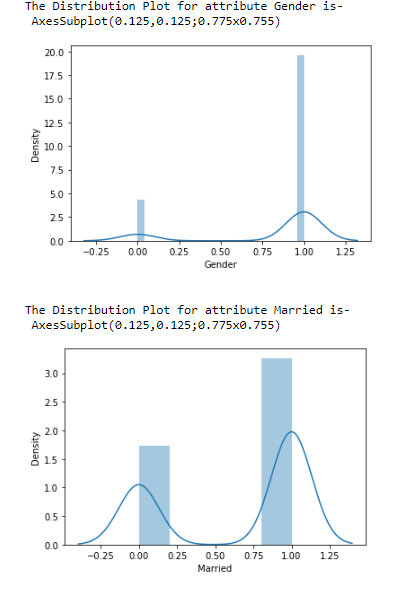
1. **Data Visualizations**

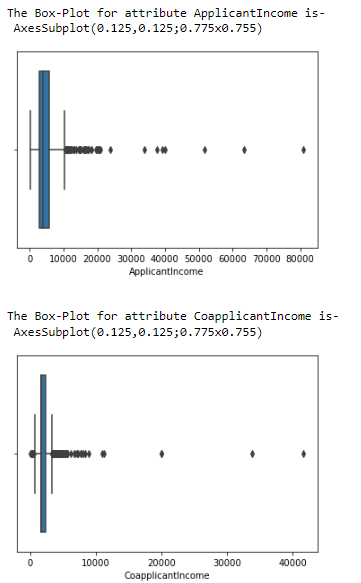
****

As, we can see that almost each attributes are containing some of the null values in the dataset and that we need to remove so that we can get our dataset free from noise.









As we can see that skewness and outliers are present in the dataset. Also, I’ve plotted different visualizations in the Jupyter (solution notebook) like Histogram, Count plot, Scatterplot, Line Plot, Categorical Plot, Violin Plot and Pair Plot to conclude the relations and important attributes amongst all. Here, in this project I’ll be writing down some of the important insights of the projects from data visualizations as below-

* Out of 614 applicant Approx. 500 are males.
* Approx. 400 applicants are married.
* Approx. 350 applicants don't have any dependent.
* Approx. 500 applicants are graduate.
* Out of 614 Approx. 100 Applicants are self-employed.
* Approx. 500 applicants has taken the loan amount tenure as 360 Months
* More than 400 applicants has at least taken one loan earlier
* Approx. 250 applicants belongs to semi-urban area.
* Approx. 450 applicants are approved for loan
* The female self-employed applicants who are applied for the loan has income around 9000
* The married applicants who has applied for the loan amount Approx. 150 are approved for the loan whereas the married applicants who has applied for the loan amount Approx. 165 are not eligible for the loan
* Most of the applicants who are graduate and applied for the loan amount Approx. 150 are approved for taking the loan where as if they apply above 160 they are not eligible for the loan
* Self-employed people who lives in urban area has the highest income of Approx. 8000
* Those Male and female both the applicants who has at least 1 credit history are eligible for the loan
* Applicant those are married and having three and above dependents in the family have income of more than 8000
* Applicants those are not married and having three and above independents in the family, their co-applicant income is approx. 6000
* Applicants who are not self-employed and have no dependents in the family has taken loan tenure as approx. 350Months
* Applicants those are graduate and living in urban areas has credit history as 1 and they are eligible for loan in most of the cases
* Graduate applicants having more than 3 dependents in the family has applied for the loan amount of approx. 225

##### Graduate Male professionals who are working in a company, being married and belongs to the semi-urban area having least dependents in the family has applied for the loan in high numbers and most of the cases are approved for the loan*.*

1. **EDA Concluding Remarks**

I’ve applied **Simple Imputer** method to replace all the null values present in different attributes with mean or mode depending on the type of variables. If the variables are of numeric data type then I’ve applied **mean** otherwise **mode** will take place.

Once the null values are replaced with corresponding mean/mode I’ve transformed the categorical variables into their numeric category form with the help of **Label Encoder** method so that later we can normalize and scale the data points.

In the feature selection process I’ve remove all the less negative correlated columns (those are having correlation score almost zero) w.r.t. target variable and these variables are- **Coapplicant Income, Loan Amount, Loan\_Amount\_Term, Applicant Income.**

Also, in the data cleansing part since the threshold value I'm considering for the **outliers** is 10% but I'm getting percentage loss of 6% which is of course considerable for outliers removal since I've still 577 rows which I think an enough data and it will work well if I even consider the outlier removal, hence will remove the outliers.

I’ve applied **Square-Root** method to remove the **Skewness** from the dataset and then applied **Standard Scaler** to obtain the mean is equal to 1 and standard deviation is equal to 0 for the feature variables.

Principle Component Analysis is used to reduce the dimension of the feature variable and in order to do so, I’ve reduced the feature variables to three principle components. As, I’ve converted all the independents attributes in to three most useful components the machine learning model will now take less time to predict the target variable with great accuracy.

Since, the target variable is in the form of 1 and 0 hence will use the classification models but as the classes of the target variable are imbalanced in nature hence first will try to make it equal with the help of SMOTE technique then only we can train and test the classification algorithms.

1. **Pre-processing Pipeline**

**Data Profiling** is the process of examining, analyzing and reviewing data to collect descriptive statistics like min, max, count and sum, Collecting data types, length, info, dimensions etcetera.

**Data Cleansing**. The aim here is to find the easiest way to rectify quality issues, such as eliminating bad data, filling in missing data or otherwise ensuring the raw data is suitable for feature engineering.

**Data Transformation** is the process of converting raw data into a format or structure that would be more suitable for model building and also data discovery in general. It is an imperative step in feature engineering that facilitates discovering insights. I’ve used here Standard Scaler technique to normalize the data.

**Data Enrichment** is a critical first step in gleaning valuable insights that can benefit a company based on data collected through analytics or machine learning.

**Data Reduction.**Raw data sets often include redundant data that arise from characterizing phenomena in different ways or data that is not relevant to a particular ML, AI or analytics task. Data reduction uses techniques like principal component analysis to transform the raw data into a simpler form suitable for particular use cases.

**Data Validation**. At this stage, the data is split into two sets. The first set is used to train a machine learning or deep learning model. The second set is the testing data that is used to gauge the accuracy and robustness of the resulting model

1. **Libraries used in this project**

import numpy as np – For Numeric & Statistics Process

import pandas as pd- For Manipulating the data

import scipy.stats – Statistics Library

from scipy.stats import zscore,boxcox- Used for Outliers and Skewness Removal

import matplotlib.pyplot as plt- Data Visualization

import seaborn as sns- Data Visualization

import warnings

warnings.filterwarnings('ignore')

import statsmodels.api- Multicollinearity

from statsmodels.stats.outliers\_influence import variance\_inflation\_factor

from sklearn.impute import SimpleImputer- Replacing Nan with Mean/Median/Mode

from sklearn.preprocessing import LabelEncoder- Encoding Technique

from sklearn.preprocessing import StandardScaler- Normalize the dataset

sc=StandardScaler()

from sklearn.decomposition import PCA- Curse of dimensionality reduction

pca=PCA(n\_components=3

from imblearn.over\_sampling import SMOTE- Imbalancing the Target Classes

sm=SMOTE()

**All the libraries of classifications**

from sklearn.linear\_model import LogisticRegression

from sklearn.model\_selection import train\_test\_split

from sklearn.metrics import accuracy\_score,classification\_report,confusion\_matrix

from sklearn.naive\_bayes import GaussianNB

from sklearn.neighbors import KNeighborsClassifier

from sklearn.tree import DecisionTreeClassifier

from sklearn.ensemble import RandomForestClassifier,AdaBoostClassifier,GradientBoostingClassifier

from sklearn.metrics import roc\_curve,roc\_auc\_score

from sklearn.model\_selection import cross\_val\_score

from sklearn.model\_selection import GridSearchCV

1. **Building Machine Learning Models**

##### I’ve imported the libraries for the Logistic Regression model and I’ve set 80% of the data for the training and 20% for the testing. At random state 49 I'm getting almost equal training and testing accuracy score and equal F1-Score too, which indicates that model is performing well. Also, I tried the cross validation method too, to check whether the testing accuracy of the model is same for both the method or not.

|  |  |  |
| --- | --- | --- |
| Model | Method | Testing Accuracy Score |
| Logistic Regression | Train Test Split | 61% |
| Logistic Regression | Cross Validation | 68% |

Let’s find out **Confusion Matrix**, **Accuracy Score** and **F1-Score** for the logistic Regression model.

**Confusion Matrix** is a table that is used to define the performance of a classification algorithm. It is a useful machine learning method which allows us to measure Recall, Precision, Accuracy, and AUC-ROC curve. Confusion matrices are widely used because they give us a better idea of a model's performance than classification accuracy does. For example, in classification accuracy, there is no information about the number of misclassified instances.

**Precision basically stats as** what proportion of positive identifications was actually correct. In the simplest terms, Precision is the ratio between the True Positives and all the Positives. For our problem statement, that would be the measure of applicants that we correctly identify having eligible for loan out of all the applicants actually having it.

**Recall basically stats as** what proportion of actual positives was identified correctly. The recall is the measure of our model correctly identifying True Positives. Thus, for all the applicants who actually are eligible for loans, recall tells us how many we correctly identified as having a loan status as Yes.

The **F1-Score** combines the precision and recall of a classifier into a single metric by taking their harmonic mean. It is primarily used to compare the performance of two classifiers. Suppose that classifier A has a higher recall, and classifier B has higher precision. In this case, the F1-scores for both the classifiers can be used to determine which one produces better results.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Actual Outcome | Confusion Matrix for Logistics Regression Model | Predicted Outcomes | | Total Outcomes |
| **0** (Loan Not Approved)  Predicted Negative | **1** (Loan Approved)  Predicted Positive |
| **0** (Loan Not Approved)  Actual Negative | 52  TN | 38  FP | 90 |
| **1** (Loan Approved)  Actual Positive | 24  FN | 46  TP | 70 |
| Total Outcomes | | 76 | 84 | 160 |

**Calculating Positive Precision, Recall and F1-Score-**

**Positive Precision**=TP/ (TP+FP) = 46/ (46+38) = 0.55

So, what is the Precision for our Logistic Regression model! **Yes**, it is **0.55** or in other words; when it predicts that an applicant is eligible for loan, it is correct around 55% of the time

**Positive Recall or Sensitivity** =TP/ (TP+FN) =46/ (46+24) = 0.66

For our model, Recall = 0.66. Recall also gives a measure of how accurately our model is able to identify the relevant data. We refer to it as Sensitivity or True Positive Rate. **F1\_Score**=2\*Precision\*Recall/ (Precision + Recall) =2\*0.55\*0.66/ (0.55+0.66) = 0.60

**Testing Accuracy** of the Logistics Regression Model= TN+TP/ (TN+TP+FP+FN) = 0.61 = 61%

**Calculating Negative Precision, Recall and F1-Score-**

**Negative Precision**=TN/ (TN+FN) =52/52+24= 0.68

**Negative Recall or Negative** **Sensitivity** =TN/ (TN+FP) =52/52+38= 0.58

**F1\_Score**=2\*0.68\*0.58/ (0.68+0.58) = 0.63

A confusion matrix helps us gain an insight into how correct our predictions were and how they hold up against the actual values. From our train and test data, we already know that our test data consisted of 160 data points. That is the 4th row and 4th column value at the end. We also notice that there are some actual and predicted values. The actual values are the number of data points that were originally categorized into 0 or 1. The predicted values are the number of data points our Logistics Regression model predicted as 0 or 1.

The actual values are:

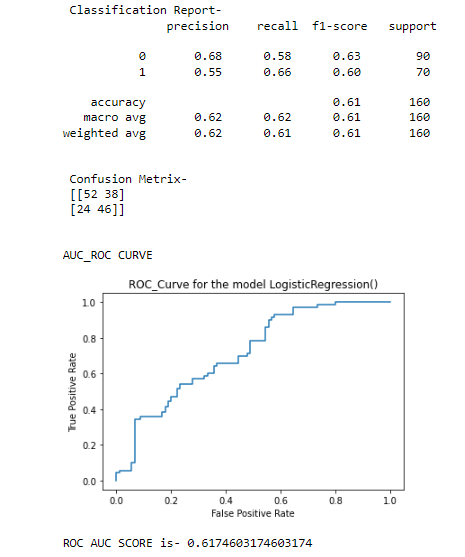
* The Applicants who’s loan is not approved = 90
* The Applicants who’s loan is approved = 70

The predicted values are:

* Number of applicants who were predicted as loan not get approved = 76
* Number of applicants who were predicted as loan got approved = 84

All the values we obtain above have a term. Let’s go over them one by one:

* The cases in which the applicants are not applicable for loans and our model also predicted as not applicable for loan it is called the **True Negatives.** For our matrix, True Negatives = 52.
* The cases in which the applicants are applicable for loans and our model also predicted as having it are called the **True Positives.** For our matrix, True Positives = 46
* However, there are some cases where the applicants are not applicable for loans, but our model has predicted that they do. This kind of error is the **Type I Error** and we call the values as **False Positives.** For our matrix, False Positives = 38
* Similarly, there are some cases where the applicants are eligible for loans, but our model has predicted that he/she doesn’t. This kind of error is the **Type II Error** and we call the values as **False Negatives.**  For our matrix, False Negatives = 24

**

**AUC - ROC** **Curve** is a performance measurement for the classification problems at various threshold settings. ROC is a probability curve and AUC represents the degree or measure of separability. It tells how much the model is capable of distinguishing between classes. Higher the AUC, the better the model is at predicting 0 classes as 0 and 1 classes as 1. By analogy, the Higher the AUC, the better the model is at distinguishing between applicants with the loan status as Yes and No.

The ROC curve is plotted with **TPR** against the **FPR** where TPR is on the **y-axis** and FPR is on the **x-axis**. The TPR defines how many correct positive results occur among all positive samples available during the test. FPR, on the other hand, defines how many incorrect positive results occur among all negative samples available during the test. In the Logistic Regression model the AUC ROC Score is 0.62 which indicates that out of 100, 62 times model is predicting the applicants are applicable for the loan.

At each K-Fold I'm getting the same CV score which means model is working well at each stage of random state. Now, I’ll be using **Hyper Parameter Tuning** technique to find out the best parameters for each of the model and then try to compare all the models w.r.t. their Testing accuracy, Training accuracy, CV Score, ROC AUC Score& F1-Scroe in a tabular form.

I’ve tested out the **Seven Models** out of which only **two models** are performing almost equal and rest are giving the testing accuracy lesser as compare to Decision Tree classifier and Random Forest Classifier. Also the CV score, F1-score and AUC ROC Score of other models are less than these top two models.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| S.N. | Model Name | Training Accuracy (%) | Testing Accuracy (%) | CV Score (%) | F1-Score (%) | ROC-AUC-Score (%) |
| **1** | **Logistic**  **Regression** | 69 | 61 | 61 | 61 | 61 |
| **2** | **Gaussian NB** | 69 | 61 | 63 | 61 | 63 |
| **3** | **K Neighbors**  **Classifier** | 82 | 73 | 69 | 73 | 73 |
| **4** | **Decision Tree Classifier** | 100 | 76 | 63 | 76 | 76 |
| **5** | **Random**  **Forest**  **Classifier** | 100 | 73 | 68 | 73 | 73 |
| **6** | **Ada Boost**  **Classifier** | 73 | 63 | 63 | 63 | 63 |
| **7** | **Gradient**  **Boosting**  **Classifier** | 86 | 69 | 65 | 69 | 69 |

Also, we can find out the classification report, accuracy score, AUC ROC Score and different insights of Random Forest Classifier and Decision Tree Classifier in the attached screenshot given as below-

##### *–*

##### 

##### Decision Tree Classifier model seems perfect as compare to other models as the training accuracy is almost 100% while testing accuracy and CV score is 76% which is good enough. Also the CV score and testing accuracy are same. It's also indicates that our model is performing excellent by each method either random\_state or K-Fold method. Also, The F1-score is 76% too it means that error are on lower side and ROC\_AUC\_SCORE is 0.76,which is greater than the threshold value of 0.6, which indicates that the machine probability is good while predicting 1 as 1 and 0 as 0.

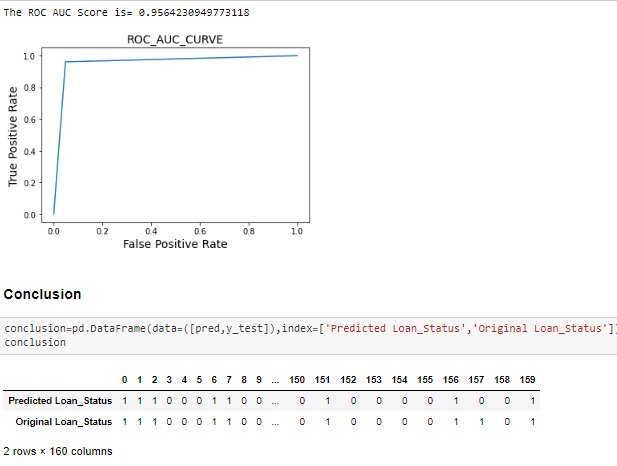
|  |  |  |
| --- | --- | --- |
| Model | Positive Precision | Positive Recall |
| Decision Tree Classifiers | 71 | 76 |
| Random Forest Classifier | 66 | 79 |

#### Decision Tree Classifier has a Precision as 71% which shows that it is correct 71% of the time while predicting the applicants those are eligible for loans whereas we are getting precision of only 66% from Random Forest Classifier model which is still good accuracy but it is less than our Decision tree classifier.

#### Decision Tree Classifier has a Recall as 76% it means that it correctly identifies 76% of all the approved loan cases and which is better than Random Forest classifier method.

|  |  |
| --- | --- |
| Algorithms | Best Parameters |
| K Neighbors Classifier | **algorithm='auto', weights='uniform'** |
| Decision Tree Classifier | **criterion='entropy', max\_features='auto', splitter='random'** |
| Random Forest Classifier | **class\_weight= 'balanced', criterion='entropy', max\_features='sqrt'** |
| Ada Boost Classifier | **algorithm= 'SAMME'** |
| Gradient Boosting Classifier | **criterion='squared\_error', loss= 'exponential', max\_features= 'sqrt'** |

**Concluding Remarks**

****

#### When I deployed our Decision Tree Classifier Model to the y\_test data what I found is that the testing accuracy of the model went to almost 96% which is excellent accuracy for predicting any target variable correctly. Also the ROC AUC Score is 0.96 which is greater than 0.6 of threshold value and it indicates that out of 100 times,96 times model is predicting the right classes i.e. 1 as 1 and 0 as 0 and this is still a great accuracy. As we can see in the conclusion portion we have got almost same value in predicted Loan\_Status as compare to original Loan\_Status. So we can say that Decision Tree Classifier model has great accuracy while predicting the Loan\_Status of the applicants.