Use of Machine Learning techniques for predicting whether a particular loan application will be approved or rejected

Building predictive models for credit decision

Shivam Bhatt

Nov 28, 2021.



Image Credits: Clix Capital

Introduction

Lending is the main source of income for most of the banks and financial institutions. The interest earned by lending are the profits for the banks and Financial Institutions. These lending comes out of the deposits made by the public to whom interest is paid for the deposits made. The difference between the lending and deposit rates are the profits that banks and Financial institutions get and thus it becomes absolutely necessary for them to accurately assess the credibility of the person to whom they lend.

The decision of granting loans to a particular applicant is a very tedious task and involves assessing various parameters like capacity, capital, conditions, character etc. Assessing all the parameters manually is a time consuming process. This is where Artificial Intelligence comes to the rescue. Using Machine learning algorithms can help us to develop predictive models that can help us based on available data whether a particular applicant should be granted a loan or not.

In this blog, we shall study about the use case of Machine Learning in the field of Credit Analysis. We shall develop a model based on various Machine Learning Algorithms to predict based on the historical available data whether a particular applicant should be granted a loan or not. We will use Python coding to build ML algorithm in Jupyter notebook. We will make use of various libraries like Pandas, Numpy, Seaborn, Matplotlib and sklearn available in Python for data visualization, analysis, cleaning and processing.

Table of Contents

1. Problem Definition
2. Data Analysis
3. EDA Concluding Remark
4. Pre-processing pipeline
5. Building Machine Learning Models
6. Concluding Remarks
7. Problem Definition

We have 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. This dataset includes details of applicants who have applied for loan. The dataset includes details like credit history, loan amount, their income, dependents etc. Let us understand each of the attributes.

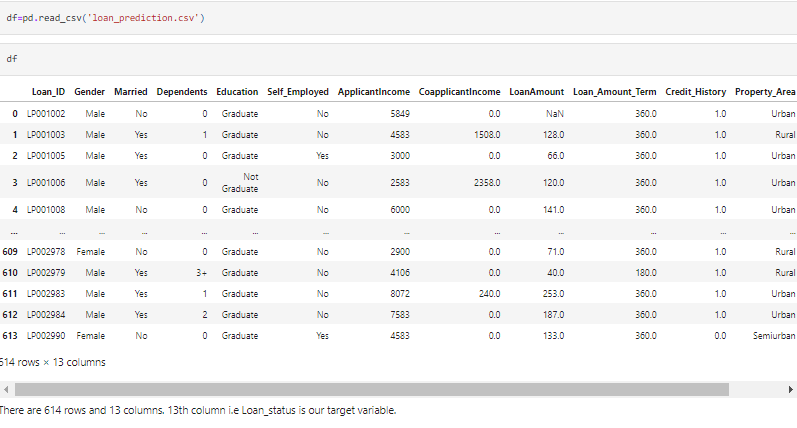
1. Loan\_ID — Loan ID for the Applicant applying for a loan
2. Gender — Gender of the Applicant
3. Married — Applicant’s marital status
4. Dependents — Number of dependents of the Applicant
5. Education — Applicant’s education status (Graduate/Under Graduate)
6. Self\_Employed — Applicant is self-employed or not
7. ApplicantIncome — Applicant’s Income
8. CoapplicantIncome — Co-applicant’s Income
9. LoanAmount — Loan Amount in thousands
10. Loan\_Amount\_Term — Term of the loan in months
11. Credit\_History — Applicant’s previous credit history meeting guidelines
12. Property\_Area — Urban, Semi-Urban, or Rural Areas
13. Loan\_Status — Loan Approval status (Target Variable)

Now that we have understood each attribute, let us start with building up a predictive model.

Step 1: Loading Data in Jupyter Notebook: The first step in building a model is loading the data in Jupyter notebook. Before loading data, we shall be importing some important libraries required in the project. Use following code to import libraries.

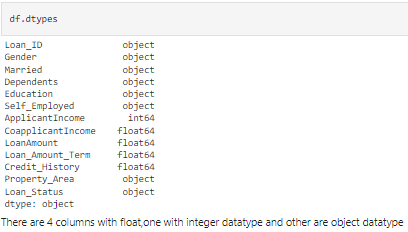


Once we have imported all important libraries, the next step is to load the dataset in Jupyter notebook using the following line of code.



1. Data Analysis

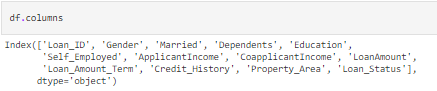
Once we have imported the dataset, the next step is to understand each attribute of the dataset and the data itself. The columns heads have been explained previously. Now let us understand attribute using various codes. As can be seen, there are 614 rows and 13 columns in the data set. We look datatype of each data.



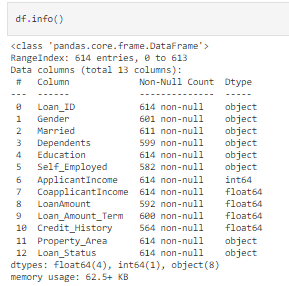
There are three formats of datatypes

1. Object: These represents variables that are of Categorical type
2. Integer: These are integers without any decimal places
3. Float: These are numerical data with decimal values.

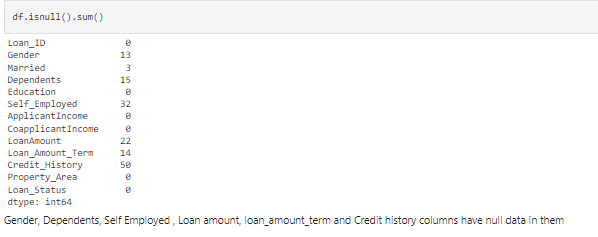
Now we look at the column headings.



We can see the headings of each columns. Now we will check how many non-null data are there in each cell.

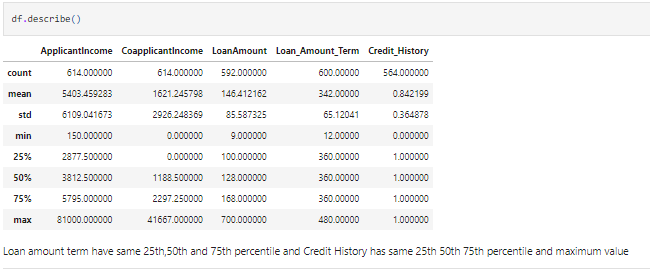


As can be seen, there are null objects in many columns. To find the count of same, we use following code



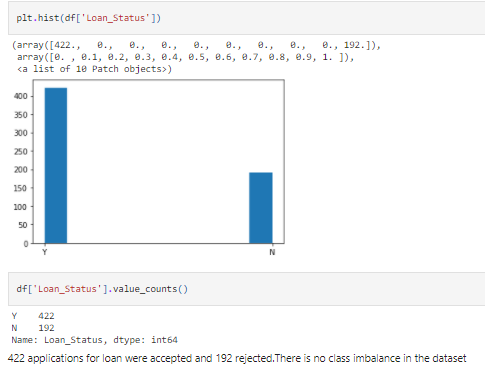
Once we have obtained number is null data, we will replace the same in pre-processing part.

We shall obtain statistical properties of numerical data using following code



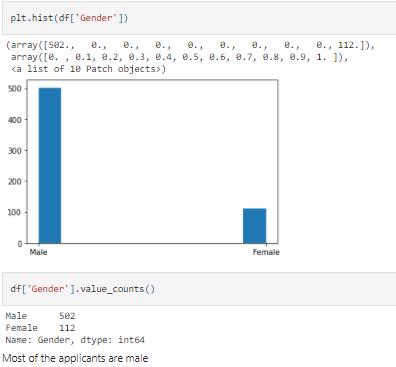
We get certain information which will be used in pre-processing step from this. Credit History is having same quartile values and so is the loan amount term.

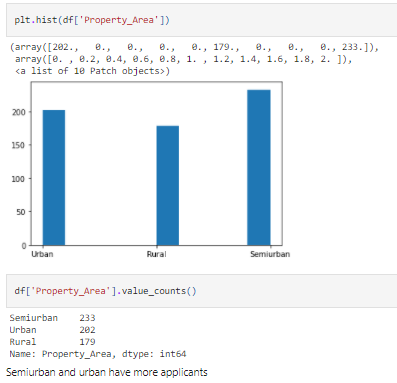
We will now visualize certain variables using visualization libraries

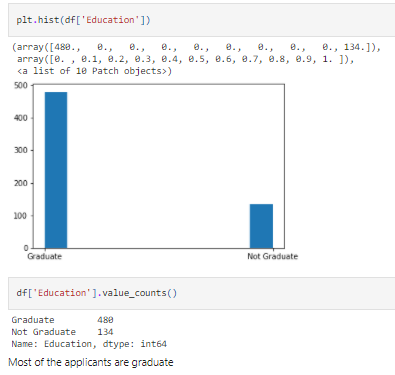


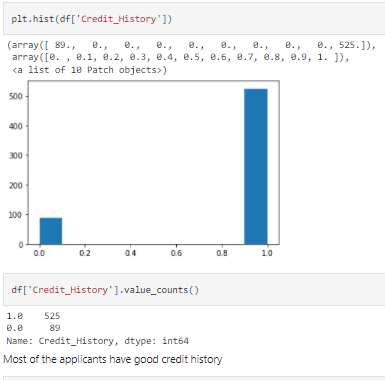
There are 422 applications whose applications were accepted and 192 rejected. Class imbalance is not a concern in this case.

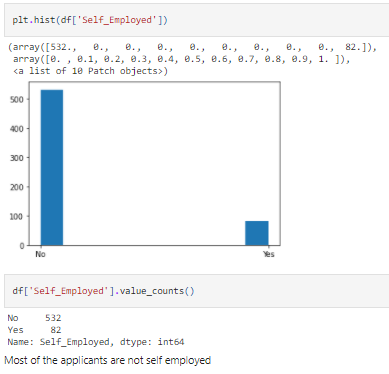
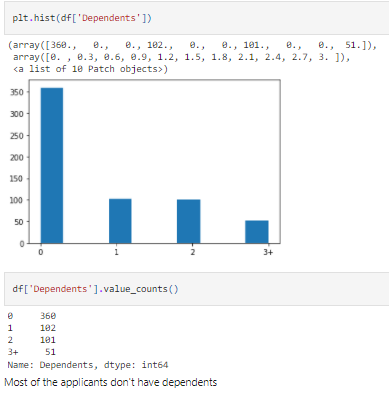
We will visualize in the same way for features data as well to see if certain attributes are more in data. Below are the histogram plots for the features

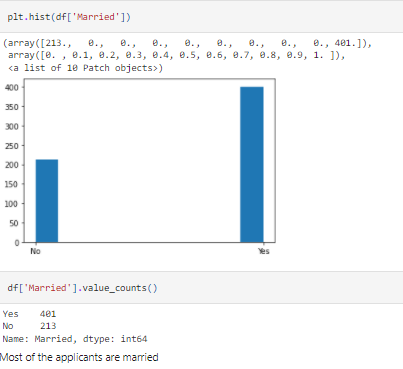






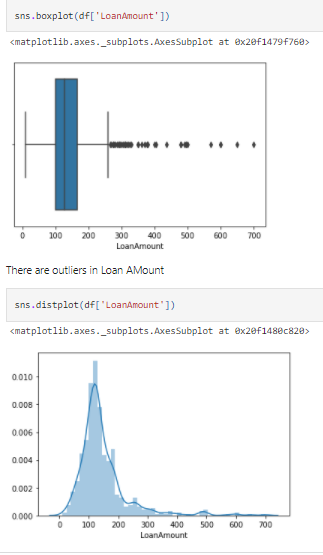


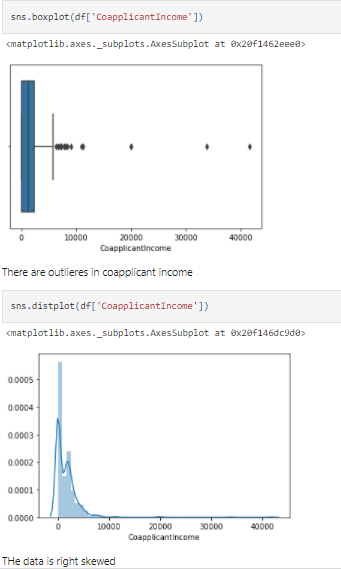


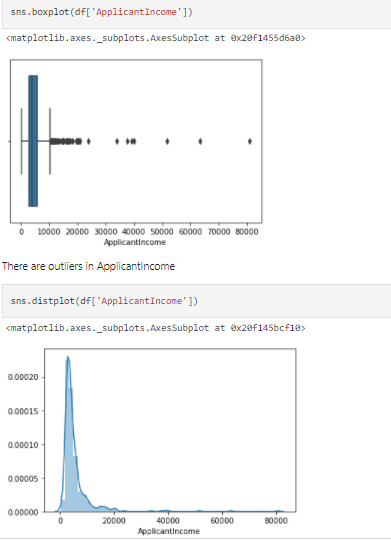


We have given the observation found in each plots at the bottom.

Now we will find outliers in numerical data and data skewness with the following code.



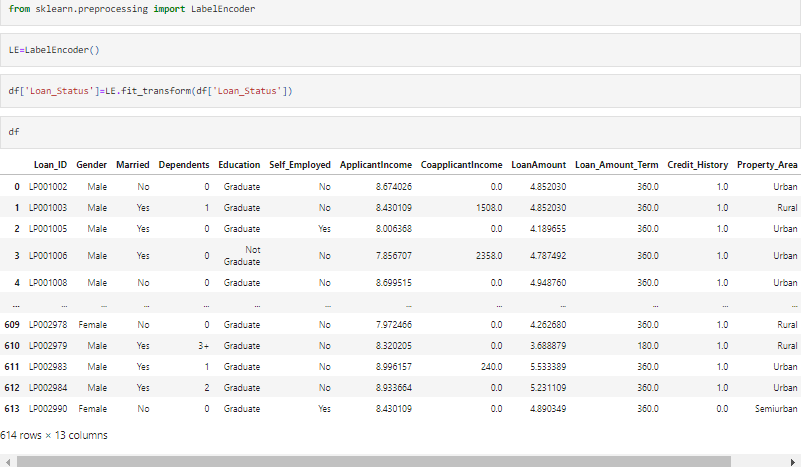


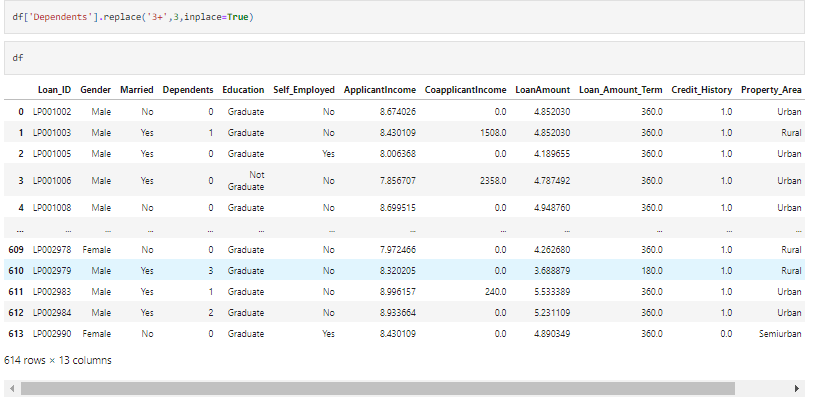


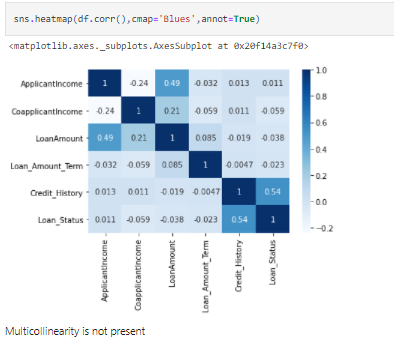
It can be inferred that most of the data in the distribution of applicant income are towards the left which means it is not normally distributed. We will try to make it normal in later sections as algorithms work better if the data is normally distributed.

The boxplot confirms the presence of a lot of outliers/extreme values. This can be attributed to the income disparity in the society. We can see that there are a higher number of graduates with very high incomes, which are appearing to be outliers. We see a similar distribution as that of the applicant's income. The majority of co-applicants income ranges from 0 to 5000. We also see a lot of outliers in the applicant's income and it is not normally distributed. We will treat all the skewness and outliers and skewness present in the dataset later on.

Now that we have visualized the data, we will check the multi-collinearity problem using heatmap of correlation. In order to make the map, the data should be of numerical form. We shall encode the target variable using label encoder so that it is assigned a numerical value and next we plot the heatmap.





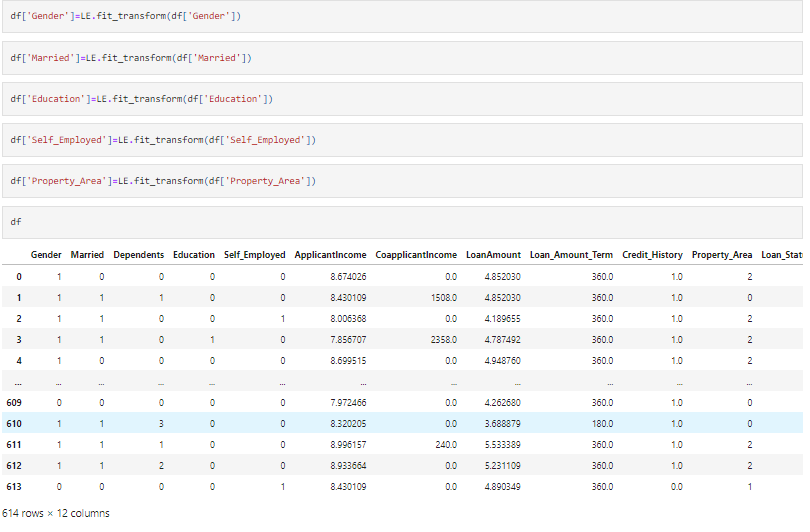


1. EDA Concluding Remark

We have done univariate and multivariate analysis of the data in which we have found that multi-collinearity problem is not present. There are outliers and missing data in the dataset which needs to be corrected. There is no class imbalance problem in the data and some data that are in object form needs to be label encoded to numerical value before applying the algorithm on the dataset.

1. Pre-processing Pipeline

Based on the analysis of our dataset, we shall now start with the cleaning up of the data.We shall first fill missing data with the mean or median or mode of the column based on the datatype. We shall then start with label encoding the data using LabelEncoder to give it numerical form. This will be followed by removing the unwanted columns from the dataset that has no impact on the target attribute. After all cleaning process is completed, we will separate target attribute from the rest of the data and perform scaling on features.



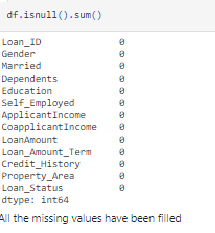
We shall now carry out Data Cleaning. First we will impute missing values with the following code









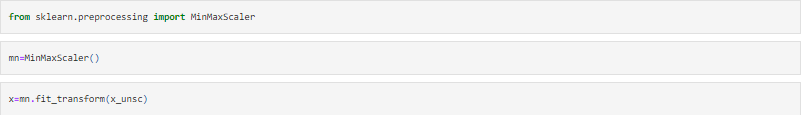


Now we will remove data that has no effect on Loan\_Status





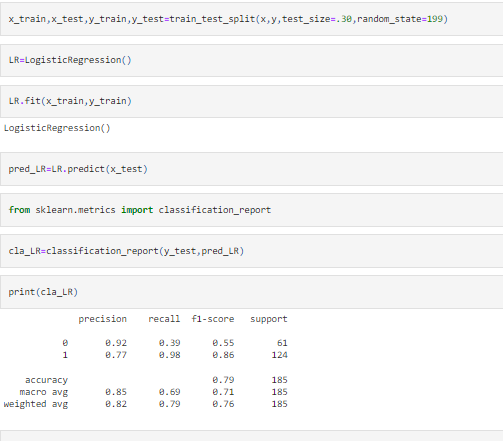
Now scaling the data



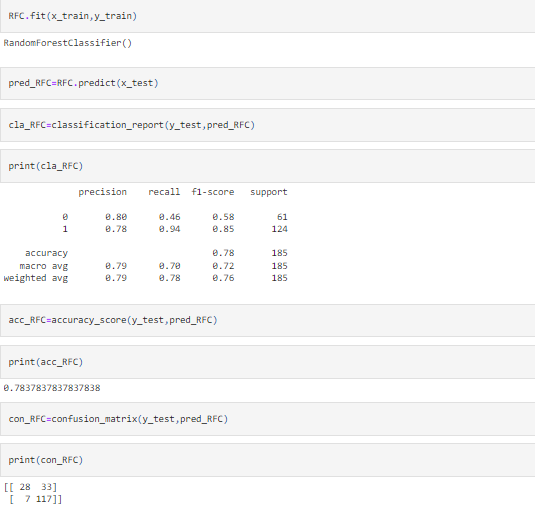
1. Building Machine Learning Models

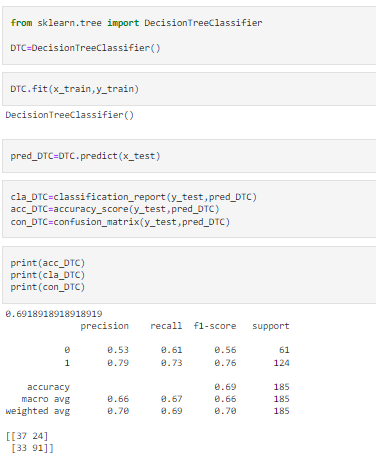
We have done pre-processing of data. Now we shall do the model building. Since out target attribute is categorical, we will use Classifier algorithm. Let us import the algorithms from libraries.

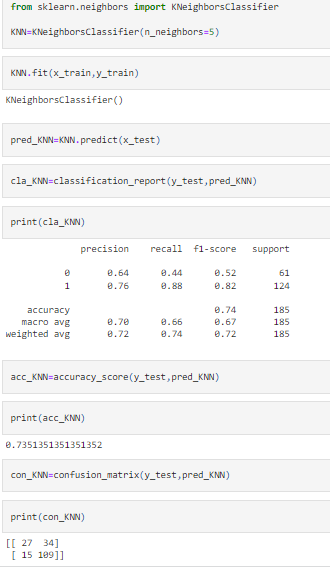








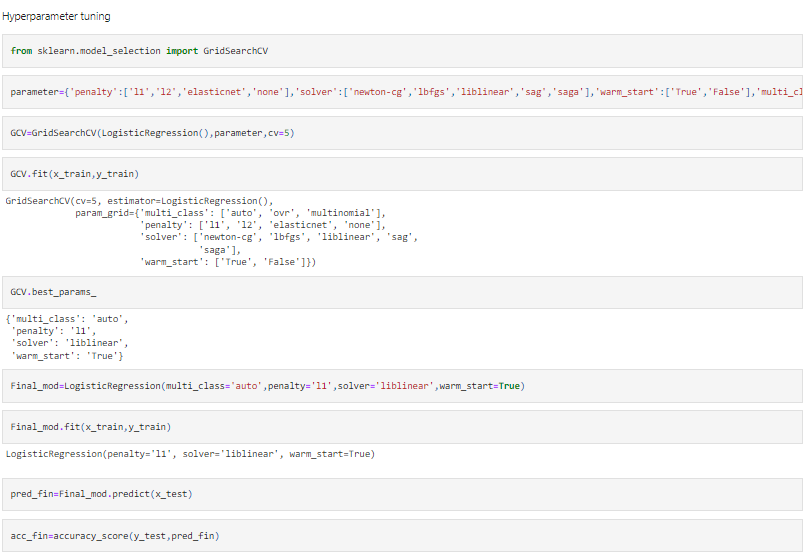


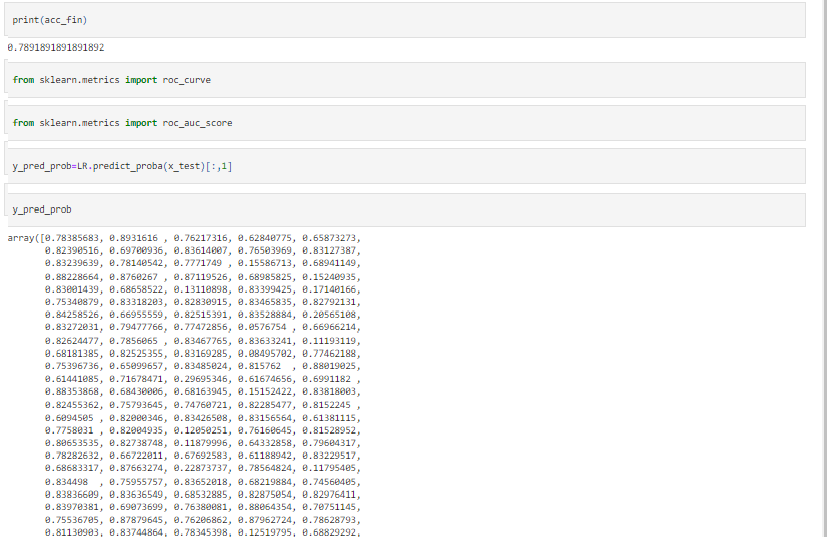


Cross Validating the scores of each Model

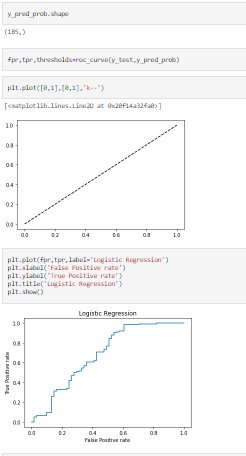


Using Hypermarameter Tuning

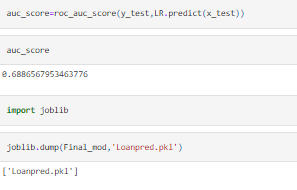




Plotting AUC-ROC curve



Finding AUC-Roc Score and saving the model



1. Concluding remark

We have built a model which has accurately predicted the target with 79% accuracy. This model will save time and assist in making decision of granting loan to a particular applicant.