# **Loan Application Status Prediction**

The loan is one of the most important products of the banking. Loans default will cause huge loss for the banks, so they pay much attention on this issue and apply various method to detect and predict default behavior of their customers. Here let us discuss a basic loan application status prediction using Machine Learning.

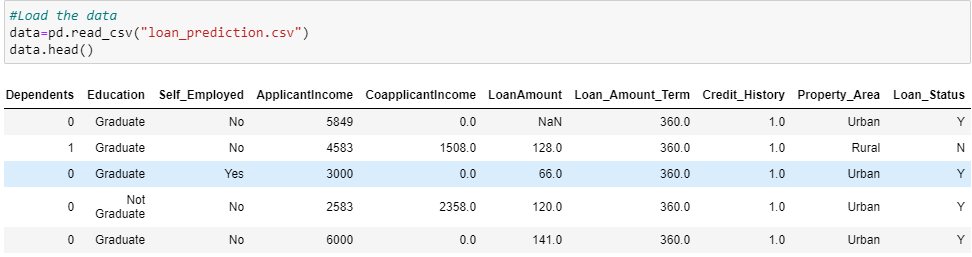
**Problem Statement:**

The dataset contains the details of applicants who have applied for loan. We must build a model that can predict whether the loan of the applicant will be approved or not based on the details provided in the dataset.

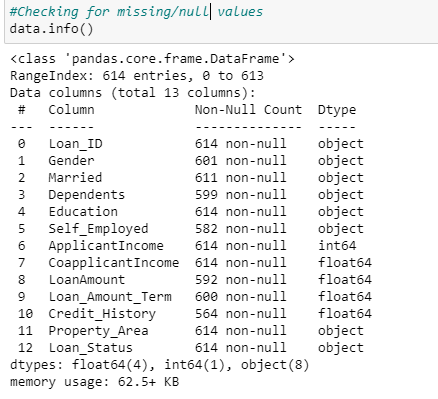
**Data Preparation:**

The data set includes details like Credit history, Loan\_Amount, ApplicantIncome, Loan\_Amount\_Term Etc. By looking at the problem statement, we can make out that the target variable is binary value, hence this is a classification problem.

The data can be downloaded from [here](https://github.com/dsrscientist/DSData/blob/master/loan_prediction.csv).

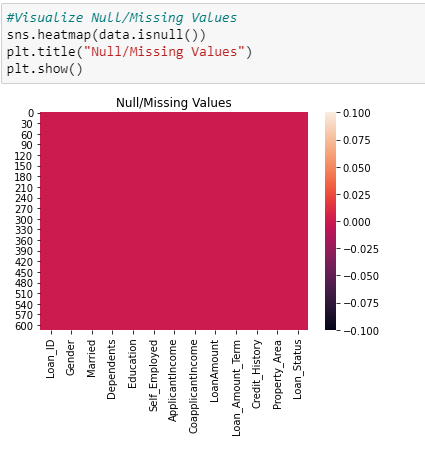


Next step is to check if the dataset has any missing or null values.



The data contains 614 entries and there are missing values in Gender, Dependents, Self\_Employed, LoanAmount, Loan\_Amount\_term and Credit\_History. The majority of the columns are categorical values.

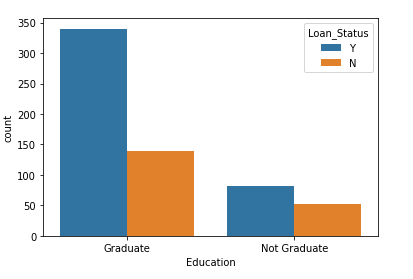
We have filled all the categorical missing values by most frequent values and the continuous missing values by mean of the respective columns. Now all our missing values are treated well. Let’s verify by looking at heatmap for missing values.



**Data Analysis:**

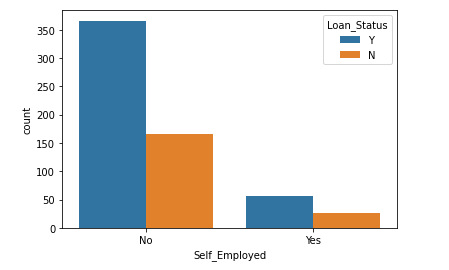
The column Loan\_ID can be deleted as this does not provide any information while predicting the status. Let’s look at some important features that contributes in predicting the Loan Status. Let’s analyze by visualizing each of the feature variable with target variable.

On plotting a count plot for Education with Loan status, there is no good relationship with these two variables. The percentage of Loan approval is higher in both Graduate and Not Graduate customers.

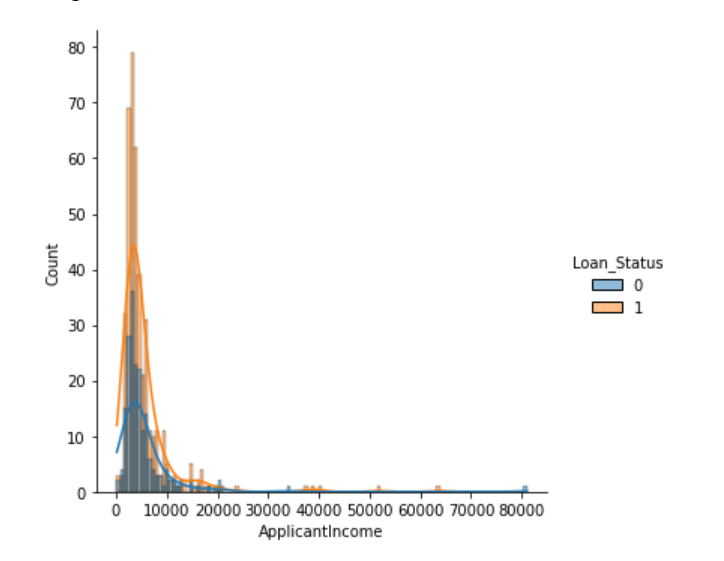


People with better education will have better income hence should get the loan approved.

By looking at the count plot below for Self\_Employed Vs Loan\_Status, half of the loan applications are rejected for non self employed and self employed customers.

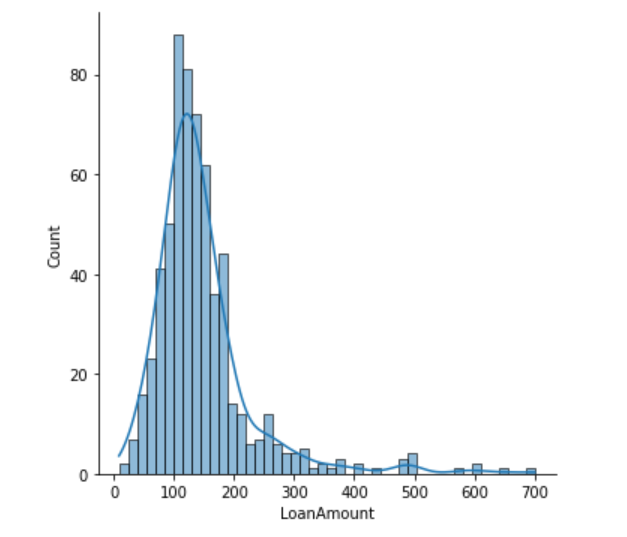


It would be interesting to study the distribution of ApplicantIncome and Loan\_status.



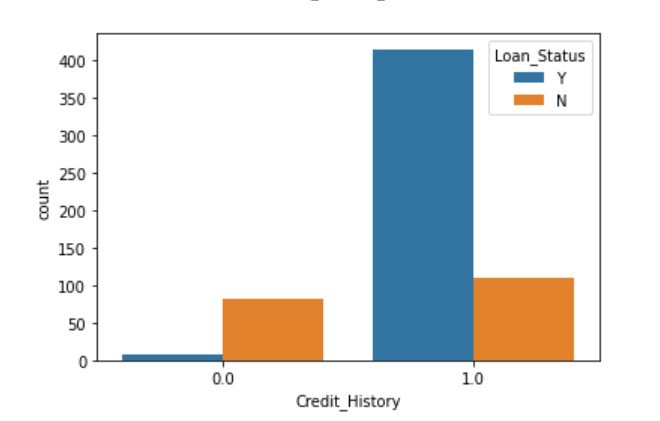
ApplicantIncome contributes to Loan\_Status, the distribution is skewed, and we can notice few outliers.

Below displot shows the distribution of Loan Amount.



The distribution is skewed, and we can notice some outliers as well, these outliers can be ignored as it is genuine data.

Another important feature is the Credit History. This plays a higher role in predicting the loan status.



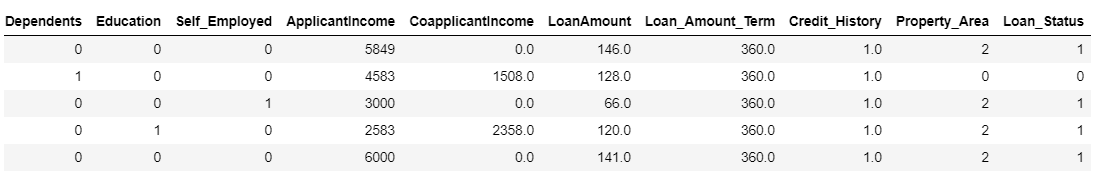
Loan status is approved where the credit history is True. There are higher chances of rejecting the loan if the customer does not have Credit History. Customers with credit history are more likely to pay their loan.

**Data Preprocessing:**

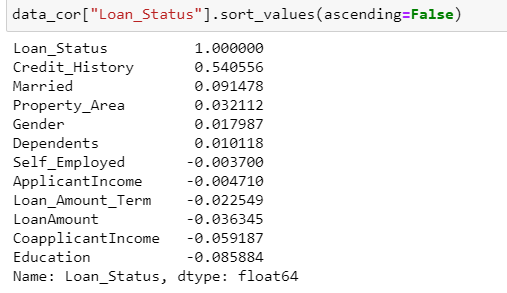
We must handle the skewness of some of the numerical columns, which is transformed using log transformation.

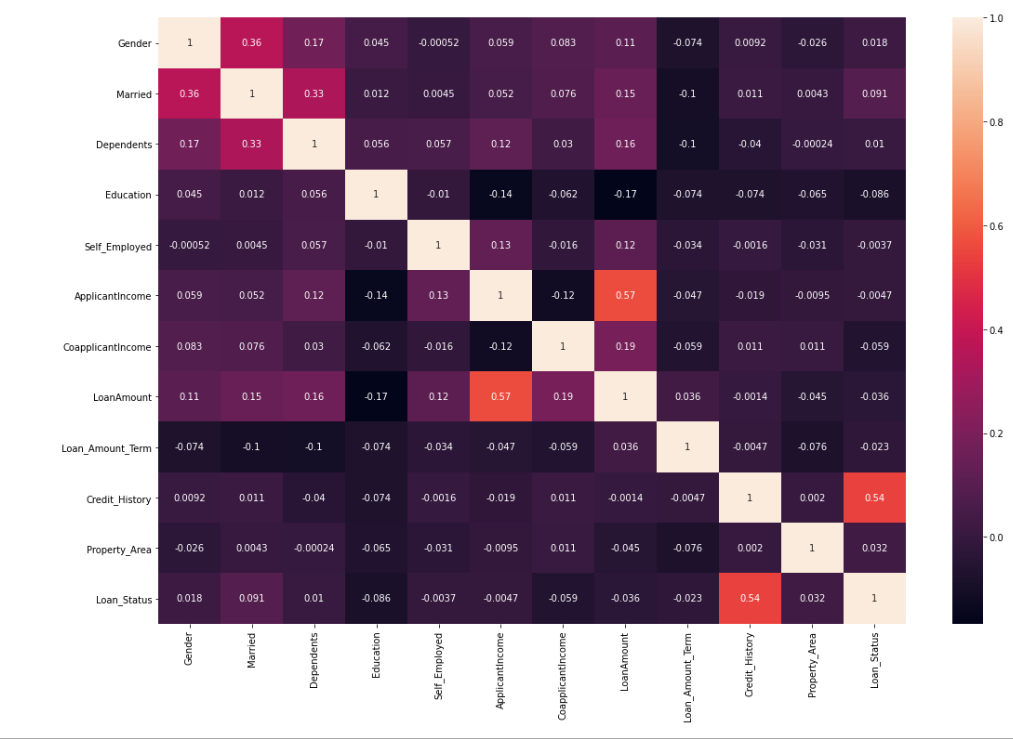
The next task is to convert all the categorical values to numerical values. We use LabelEncoder utility from scikit learn library to convert the categorical values to numeric.

After the encoding process the data looks as shown below:



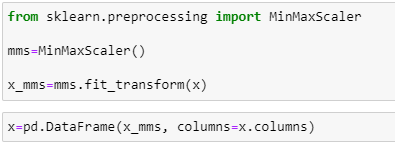
Now we can create a seaborn heatmap to see the correlation between the feature variable and target variable. Also, below we can see the correlation with respect to target variable Loan\_Status.



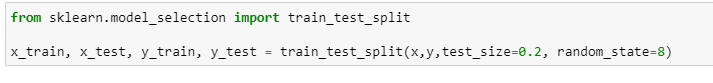


Credit\_History has highest correlation with Loan\_Status. Rest all has very less correlation with Loan\_Status.

Now the whole feature set is scaled using MinMaxScaler utility from sklearn preprocessing library to bring all the datapoints to similar scale.



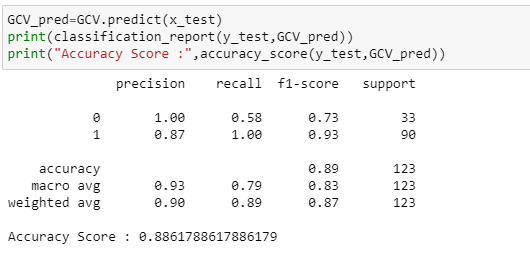
We then split the dataset into training set and testing set using train\_test\_split utility from the scikit learn model selection library.



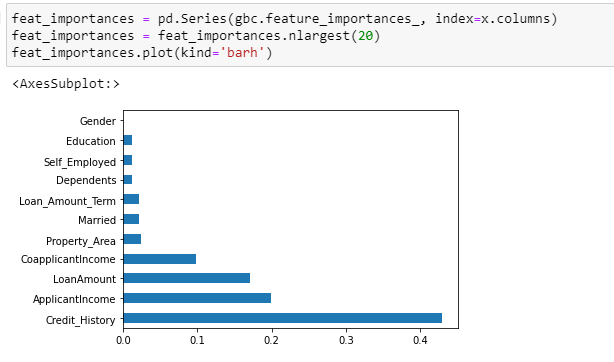
**Model Building:**

The GradientBoostingClassifier is ensemble learning method which is chosen over the other algorithms as the predictive performance/accuracy is higher compared with the other classifiers. Gradient Boostingclassifier has lots of flexibility, it can optimize on different loss functions and provides several hyper parameter tuning options that make the function fit very flexible.

We then train the model and apply the gradientBoostingClassifer to testing data set. We check the accuracy score or f1 score with cross validation as well to overcome the problem of overfitting of the data. Then we perform the hyper parameter tuning to get the best parameters with highest scores.



Our accuracy\_score is 88.61% with GradientBoostingClassifier, model is performing very good.



Also, when we check the feature importance, we found that Credit\_history contributes higher followed by ApplicantIncome and LoanAmount.