**Loan Application Prediction Status**

****

**Introduction:**

A loan is the lending of money by one or more individuals/organization/bank to some borrower, which can again be an individual/organization/bank.

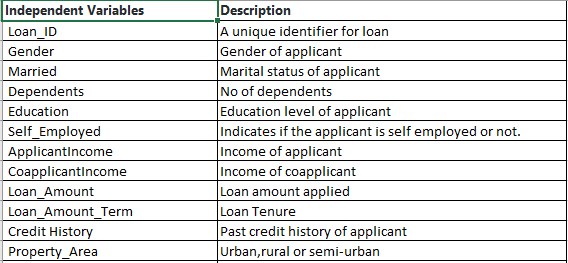
The borrowing party incurs a debt and has to repay the principal amount along with the interest to the lending party.

Loans are broadly divided as openended and close-ended loans. Open-ended loans are the loans for which the client has approval for a specific amount. Examples of open-end loans are credit cards and a home equity line of credit (HELOC). Close-ended loans decreases with each payment. In other words, it is a legal term that cannot be modified by the borrower. Personal loans, mortgages, auto payments, instalment loan and student loans are the most common examples of close-ended loans. Secured or collateral loan are those loans that are protected by an asset. Houses, Vehicles, Savings accounts are the personal properties used to secure the loan. Unsecured loans are also known as personal or signature loans. Here the lender believes that the borrower can repay the loan based on financial resources possessed by the borrower.

**Problem Definition:**

The dataset includes details of the applicants who have applied for loan. It includes different input features and one output variable i.e. Loan\_Status. Our aim is to predict whether the loan for a particular applicant will be approved or not basis the data provided for the applicant.

**Features:**



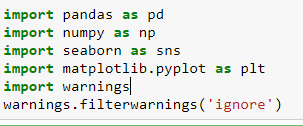


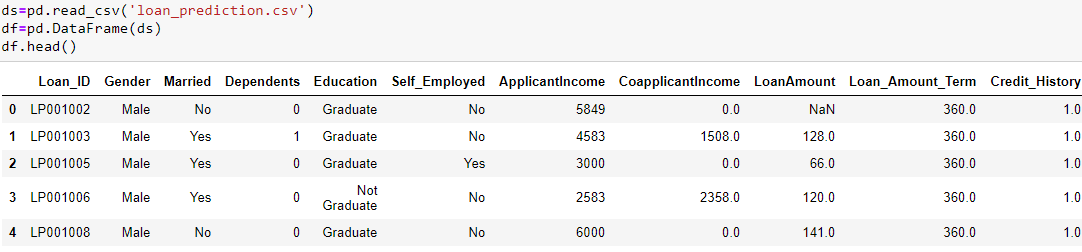
**Data Analysis:**

Now let’s work on dataset.

**Data Cleaning:**

Import Libraries

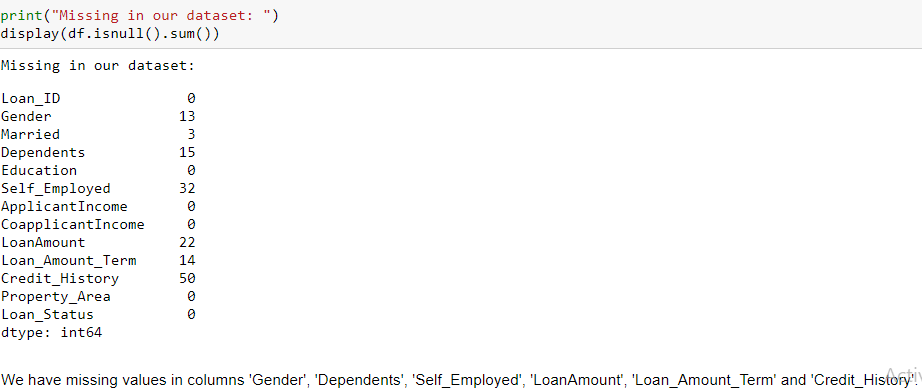




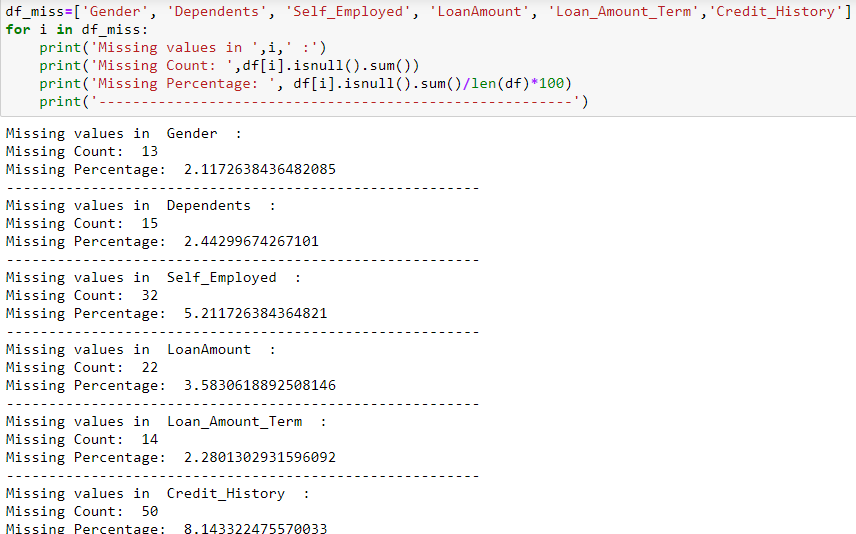
The target varible is 'Loan\_Status' which indicates whether an individual's loan will be approved or not. So it can have only 2 possible values 'N' and 'Y'.

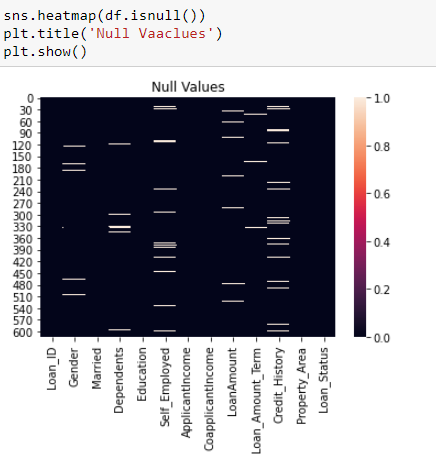
We have 12 different variables which can be used as feature to predict the outcome of our target.

**Missing Values:**

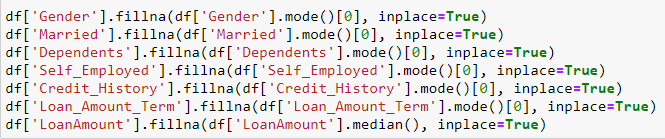


Missing Percentage:



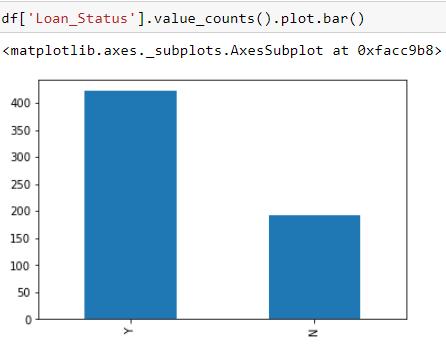


We can clearly see that there are missing values in our dataset. The missing values can irritate our algorithms, so cleaning is a crucial part.



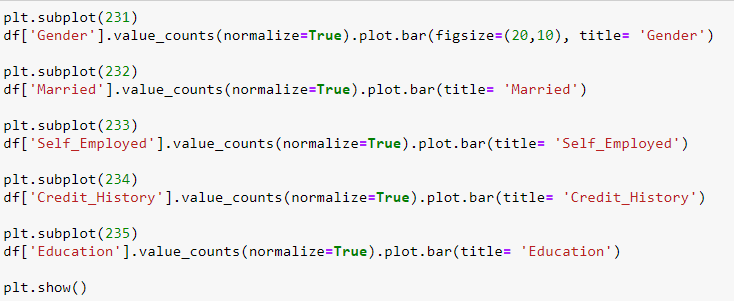
**Power of Visualization:**

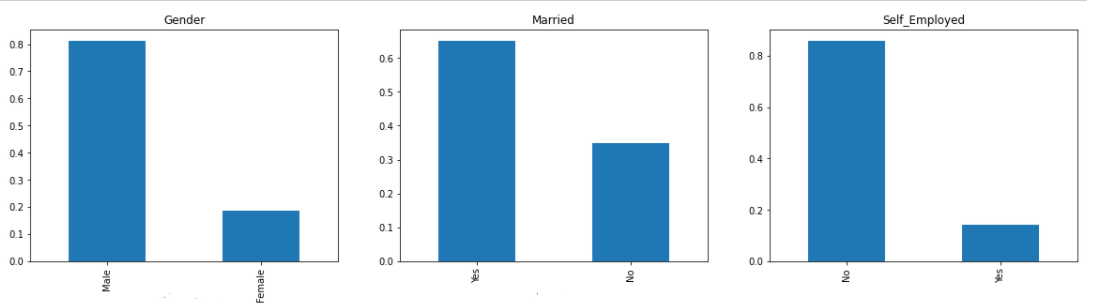
**Target variable distribution:**

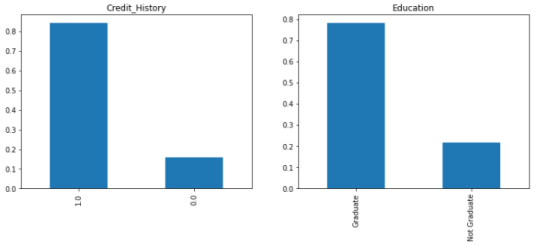


This is an unbalanced dataset as one value has very high count than the other.

**Independent Variables (Categorical) Distribution:**





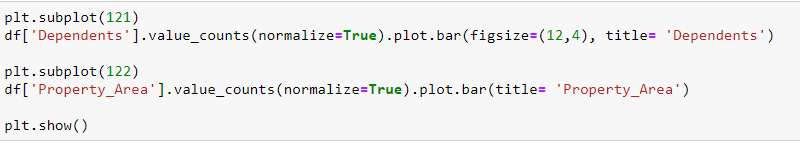


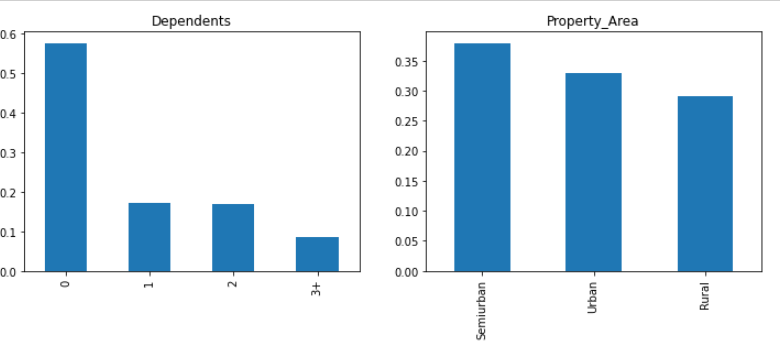
**Observations:**

* 80% applicants are Male.
* Approax 80% applicants are graduates.
* More than 60% applicants are married.
* More than 80% applicants are self-employed.
* More than 80% applicants have Credit history.

**Independent Variables (Ordinal) Distribution:**

Ordinal Variables have some order involved. Like we have values 'Urban', 'Rural' and 'Semiurban'. In the same way, we have values 0,1,2,3+ in 'Dependents' column.

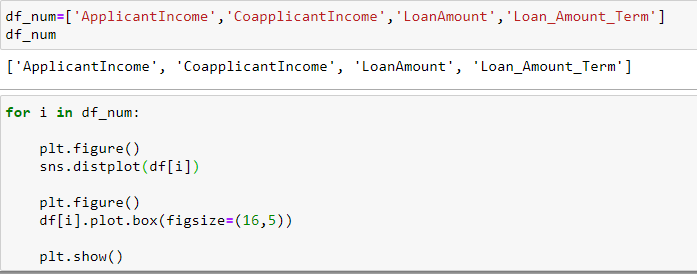


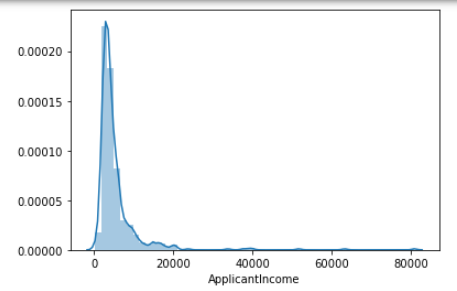


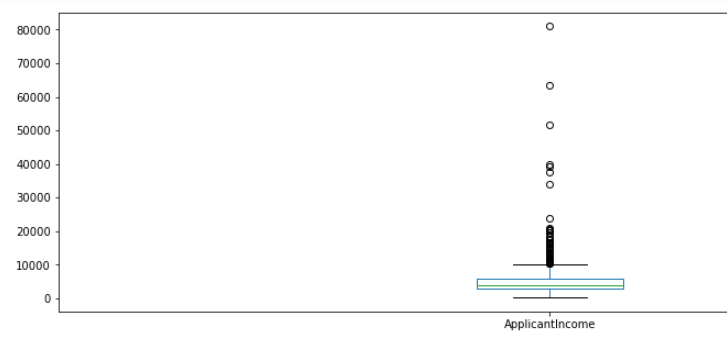
**Observations:**

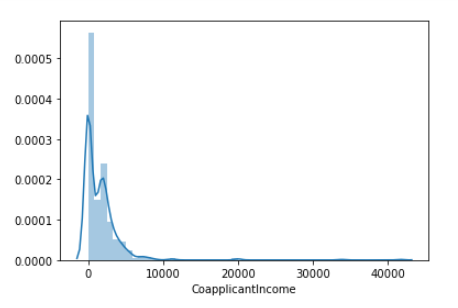
* More than 50% of applicants do not have any dependents.
* The applicants from 'Semiurban' area have maximum count, followed by Urban people and then Rural area people.

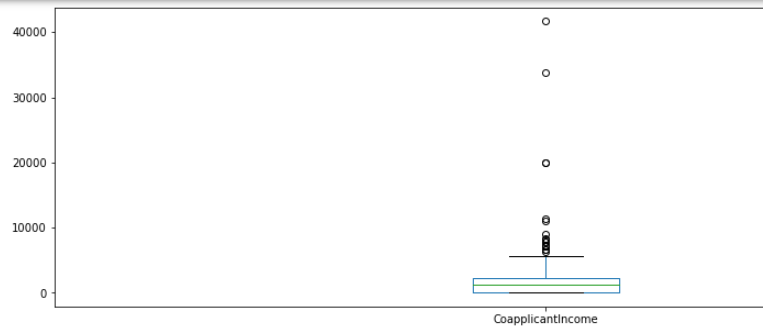
**Independent Variables (Numerical) Distribution:**

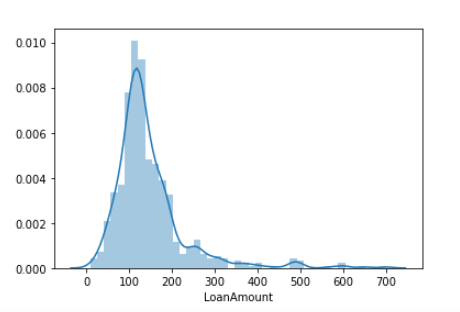


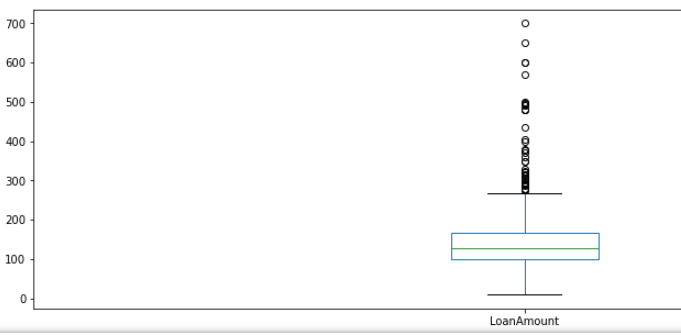


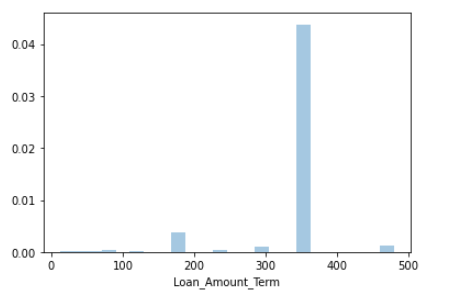


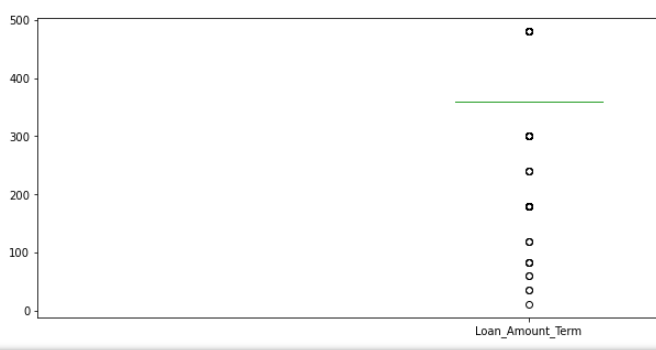








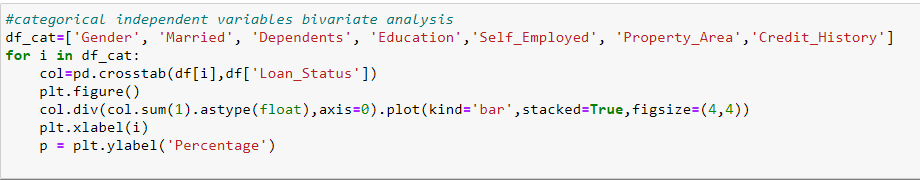


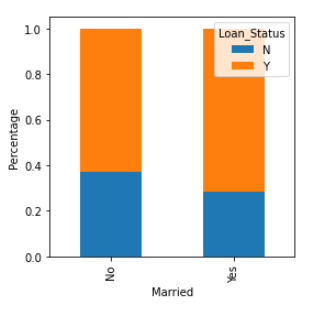
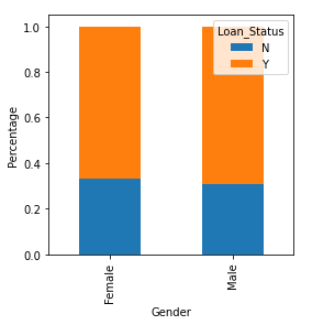


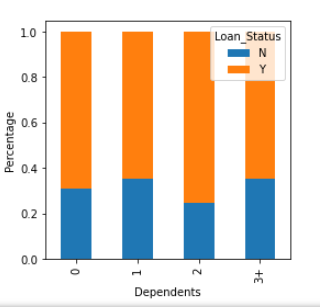
**Observations:**

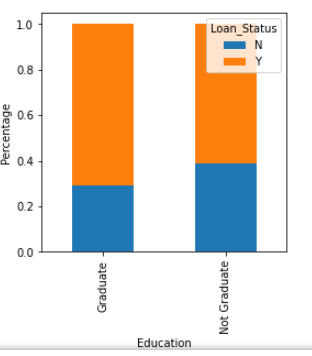
* 'ApplicantIncome' data is positive skewed, which means the distribution is towards left. We will make it normal in further steps.
* 'ApplicantIncome' boxplot confirms the presence of outliers. We will be removing outliers in further steps.
* 'CoapplicantIncome' data is also positive skewed. Also there are a lot of outliers.
* 'LoanAmount' distribution seems closer to normal distribution but still skewed.
* There are outliers present in 'LoanAmount' data. It means that for some of the cases, the loan amount is too high.
* There are outliers present in 'Loan\_Amount\_Term' data as well.

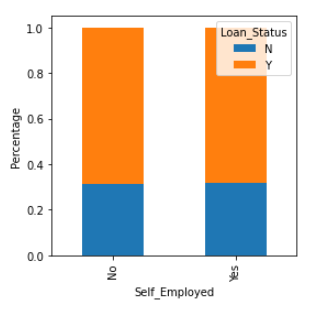
**Bivariate Analysis:**

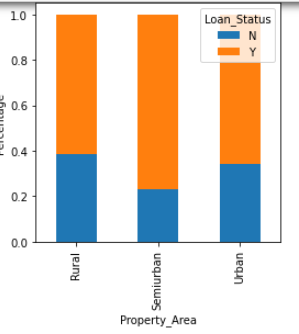














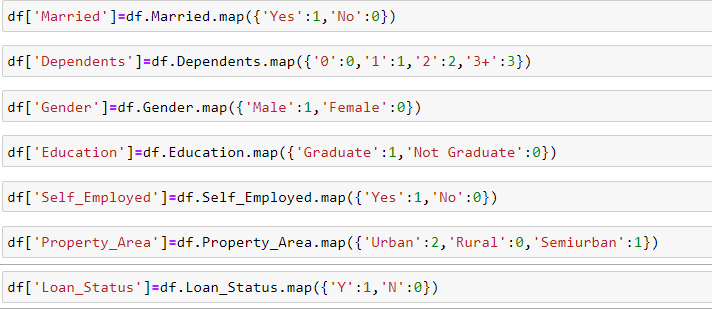
**Observations:**

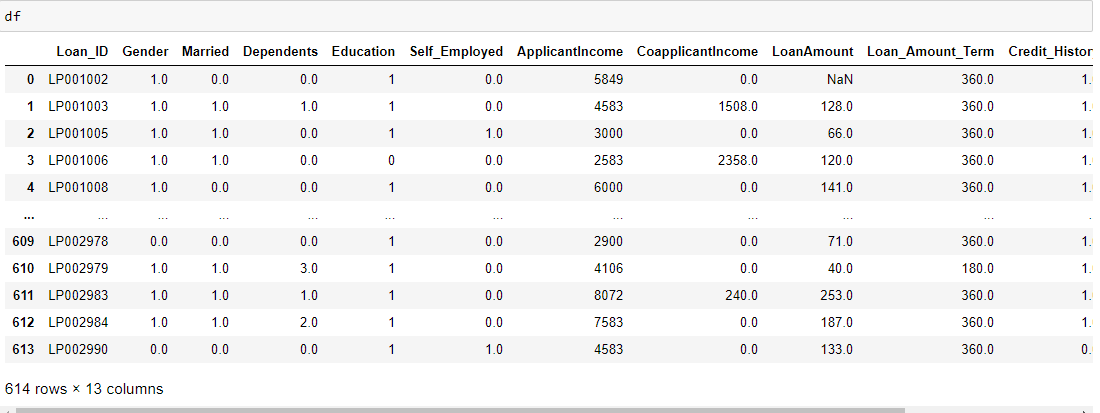
* Gender column has no impact on Loan Status.
* Married applicants have more loan approval ratio than unmarried applicants.
* Applicants having 1 and 3+ dependents have almost same approval rate.
* Graduates have more approval rate than non-graduate applicants.
* We are unable to conclude any relationship between Self\_Employed and Loan\_Status.
* Semiurban applicants have got more loans approved than Rural and Urban applicants.
* Applicants with credit history 1 have more chances for loan approval than applicants with credit history 0.

**EDA Conclusions:**

1. Loan Approval Status: About 2/3rd of applicants has been granted loan.
2. Sex: There are more Men than Women (approx. 3x)
3. Marital Status: 2/3rd of the population in the dataset is Marred; Married applicants are more likely to be granted loans.
4. Dependents: Majority of the population have zero dependents and are also likely to accepted for loan.
5. Education: About 5/6th of the population is Graduate and graduates have higher proportion of loan approval
6. Employment: 5/6th of population is not self-employed.
7. Property Area: More applicants from Semi-urban and also likely to be granted loans.
8. Applicant with credit history is far more likely to be accepted.
9. Loan Amount Term: Majority of the loans taken are for 360 Months (30 years).

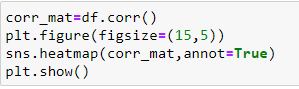
**Converting Categorical Data into Numerical Data:**

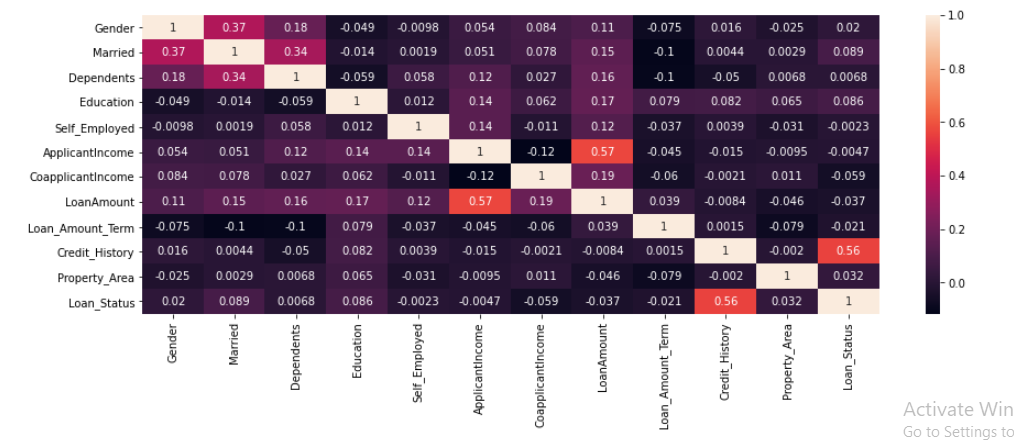


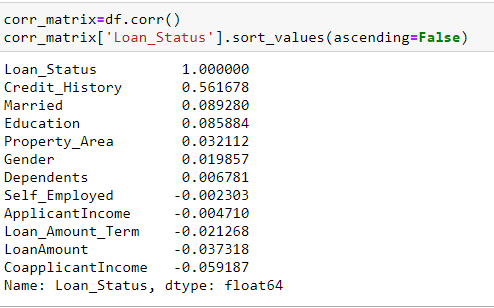


**Correlation:**

Let’s see how each variable is related to the target variable and to each other.







POSITIVE CORRELATION: If an increase in feature A leads to increase in feature B, then they are positively correlated. A value 1 means perfect positive correlation. We can see some variables have positive correlation with the target variable.

NEGATIVE CORRELATION: If an increase in feature A leads to decrease in feature B, then they are negatively correlated. A value -1 means perfect negative correlation. We can see some variables have negative correlation with the target variable

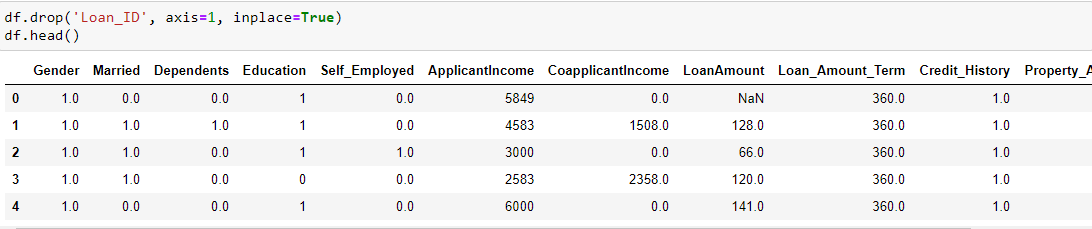
Now lets say that two features are highly or perfectly correlated, so the increase in one leads to increase in the other. This means that both the features are containing highly similar information and there is very little or no variance in information. This is known as MultiColinearity as both of them contains almost the same information.

While making or training models, we should try to eliminate redundant features as it reduces training time and many such advantages.

Now from the above heatmap,we can see that the features are not much correlated. So we can carry on with all features.

Credit\_History is the most closely related to the target variable. Other features like Marital Status and Education also impact the Loan Status.

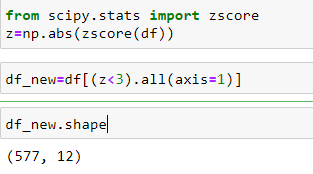
Loan\_ID does not play any role in prediction of our target variable, so we can drop it while building our model.



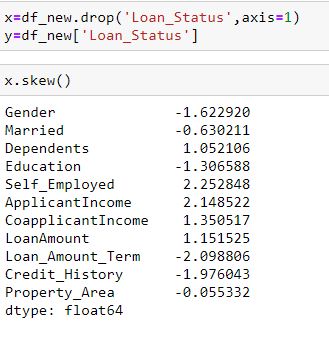
**Handling Outliers:**

Outliers are data points which are at abnormal distance from the mean.

We will remove outliers based on z-score. Z-score indicates how many standard deviations a datapoint is away from the mean.

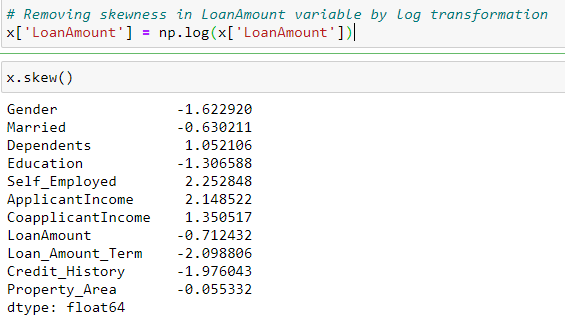


**Handling Skewness:**



We can see that there is skewness present in Gender, Credit History, Applicant Income, Self Employed and other columns as well. But all these columns are categorical variables, so they can have only some discrete set of values. Like Gender can have value 0 and 1 only, so skewness is expected in these categorical variables and we prefer not to transform it to avoid any information distortion.

We can remove skewness from LoanAmount as it is not a categorical variable.



**Model Development:**

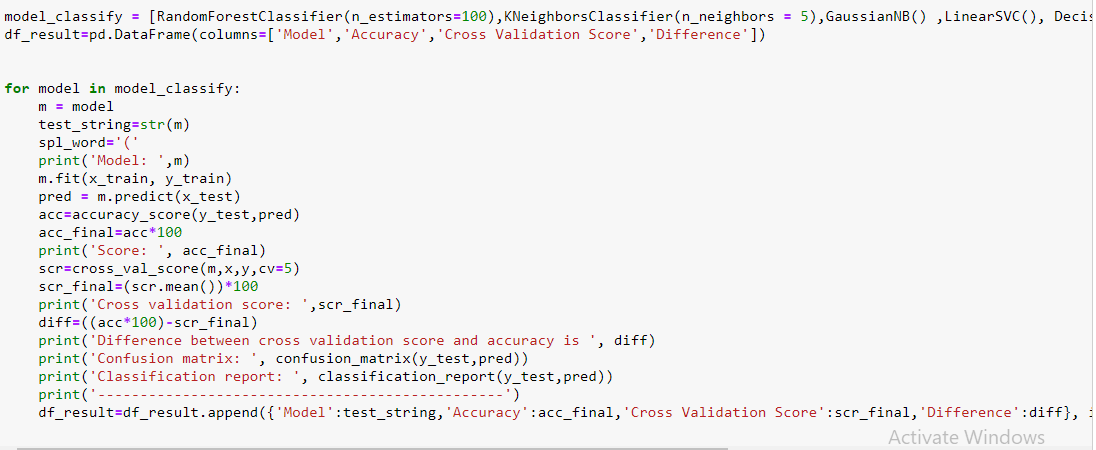


**Building the model:**

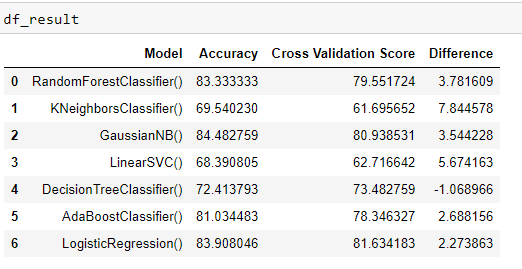
Let's try different algorithms and find out the accuracy for each model. We will also consider the cross validation score to check if the accuracy is due to overfitting.

The model with high accuracy and low difference between accuracy score and cross validation score will be considered as the best fit model.

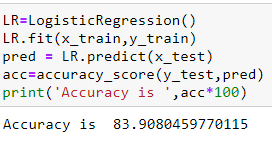




We have tried different classification models. Let’s see the result for all these models.



We can see that Logistic Regression has maximum accuracy score with less overfitting, so this will be our best fit algorithm.

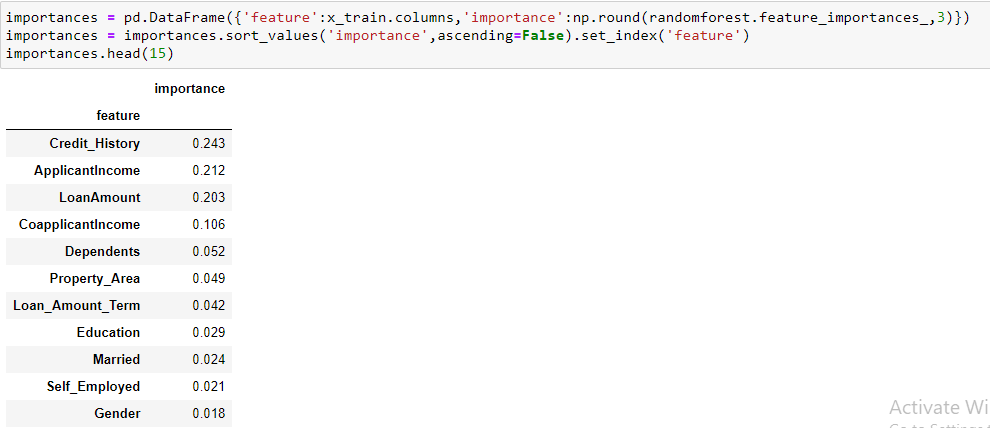


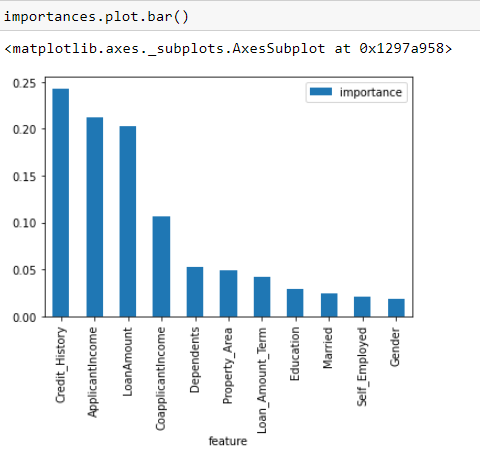
**Feature Importance:**

Feature importance refers to techniques that assign a score to input features based on how useful they are at predicting a target variable.

Feature importance scores play an important role in a predictive modelling project, including providing insight into the data, insight into the model, and the basis for dimensionality reduction and feature selection that can improve the efficiency and effectiveness of a predictive model on the problem.

Sklearn measures a feature importance by looking at how much the tree nodes uses that feature, reduced impurity on average (across all trees in the forest). It computes this score automatically for each feature after training and scales the results so that the sum of all importance is equal to 1.



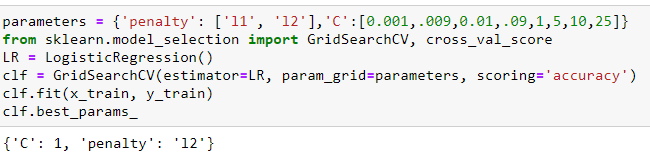


**Hyper Parameter Tuning using Grid Search CV:**

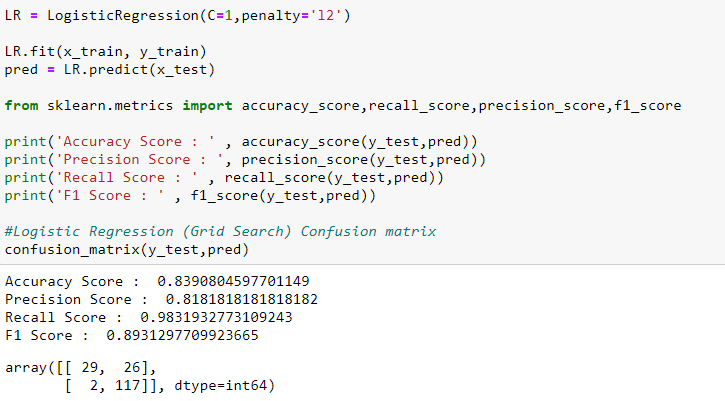
We will try to improve the accuracy by tuning the hyperparameters for this model. We will use grid search to get the optimized values of hyper parameters. GridSearch is a way to select the best of a family of hyper parameters, parametrized by a grid of parameters.

We will use GridSearchCV in sklearn.model\_selection for an exhaustive search over specified parameter values for an estimator. GridSearchCV implements a “fit” and a “score” method

We will tune the penalty and C parameter for Logistic Regression.

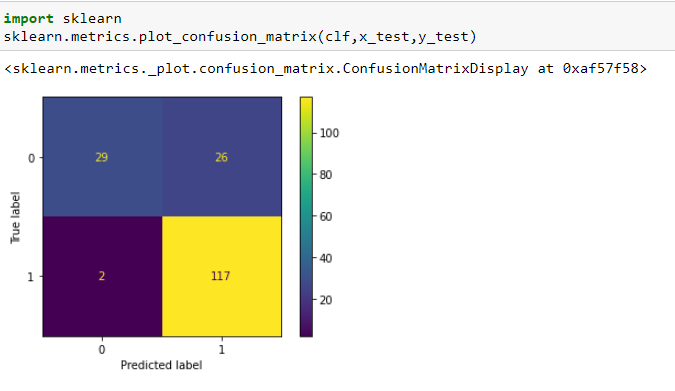


Testing the model with our best fit parameters:



**Confusion Matrix:**

A Confusion matrix is an N x N matrix used for evaluating the performance of a classification model, where N is the number of target classes. The matrix compares the actual target values with those predicted by the machine learning model. This gives us a holistic view of how well our classification model is performing and what kinds of errors it is making.



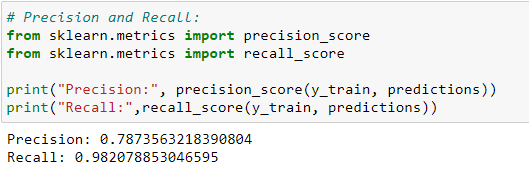
From the above plotting, we can see that 29 is True Positive Value and 117 is the True Negative Value.

26 and 2 are error terms and represent False Positive and False Negative respectively. This indicates that 26 applicants which are predicted positive for Loan approval are wrong and 2 applicants are predicted as rejected which were not rejected.

**Precision and Recall:**

Precision is a useful metric in cases where False Positive is a higher concern than False Negatives.

Recall is a useful metric in cases where False Negative trumps False Positive.

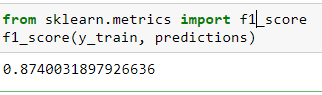


This indicates that the model predicts 78% of the time, an applicant got approval which was predicted correctly (precision). The recall tells us that it predicted the approval of 66 % of the applicants who actually got approval.

**F1 Score:**

F1-score is a harmonic mean of Precision and Recall, and so it gives a combined idea about these two metrics. It is maximum when Precision is equal to Recall.

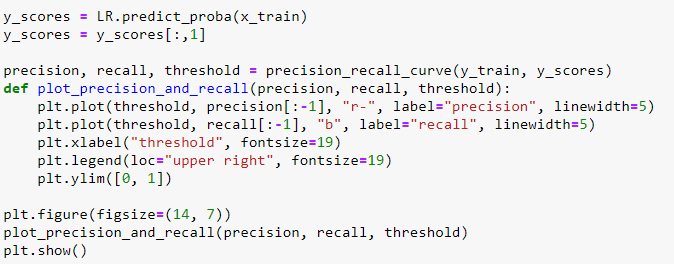
But there is a catch here. The interpretability of the F1-score is poor. This means that we don’t know what our classifier is maximizing – precision or recall? So, we use it in combination with other evaluation metrics which gives us a complete picture of the result.

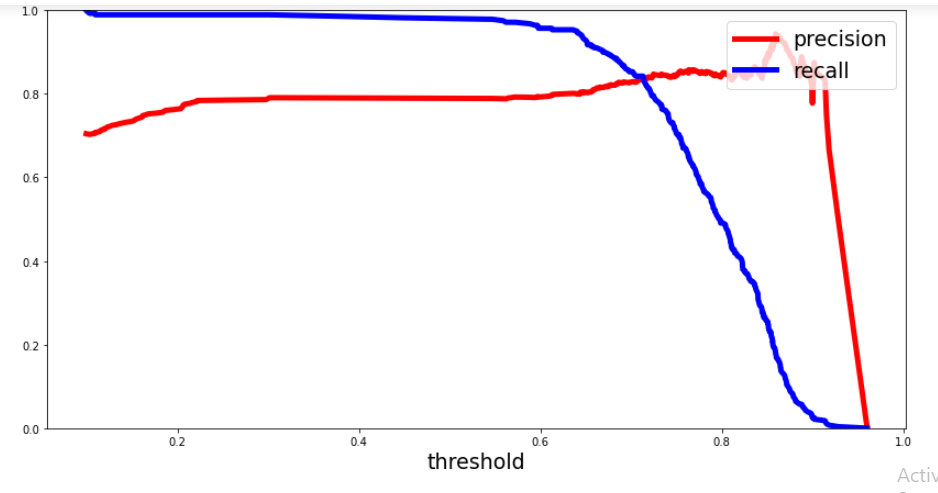


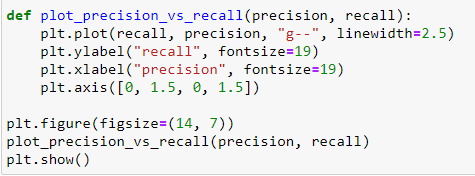
**Precision Recall Curve:**

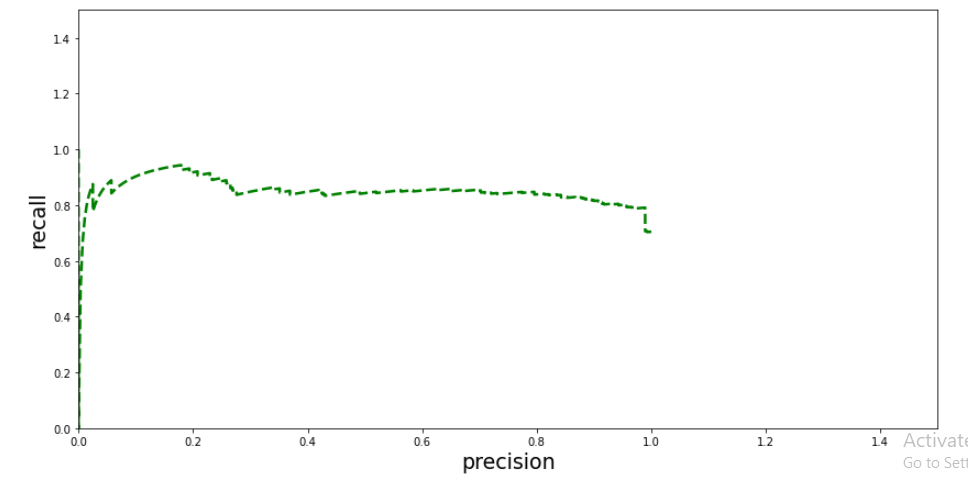
For each applicant, the algorithm has to classify, it computes a probability based on a function and it classifies the application as approved (when the score is bigger the than threshold) or as not approved (when the score is smaller than the threshold). That’s why the threshold plays an important part. We will plot the precision and recall with the threshold using matplotlib.

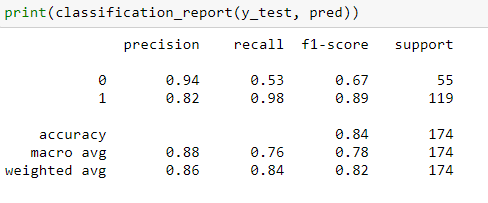
Getting the probabilities of our predictions.



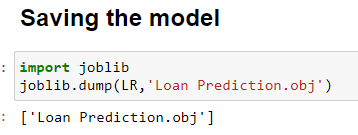








We just completed our model. Lets save it for production.



**Conclusion:**

From the Exploratory Data Analysis, we could generate insights from the data and how each of the features relates to the target feature.

Also, it can be seen from the evaluation of three models that Logistic Regression performed better than others

