Loan Application Status Prediction



* In this blog-post, I will go through the whole process of creating a machine learning model on the famous Loan Application Status Prediction, which is used by many people all over the world. It provides information on the Loan ID, Gender, Married, Dependents, Education, Self Employed, Applicant Income, Co-applicant Income, Loan Amount, Loan Amount Term, Credit History, Property Area.

**Problem Statement:**

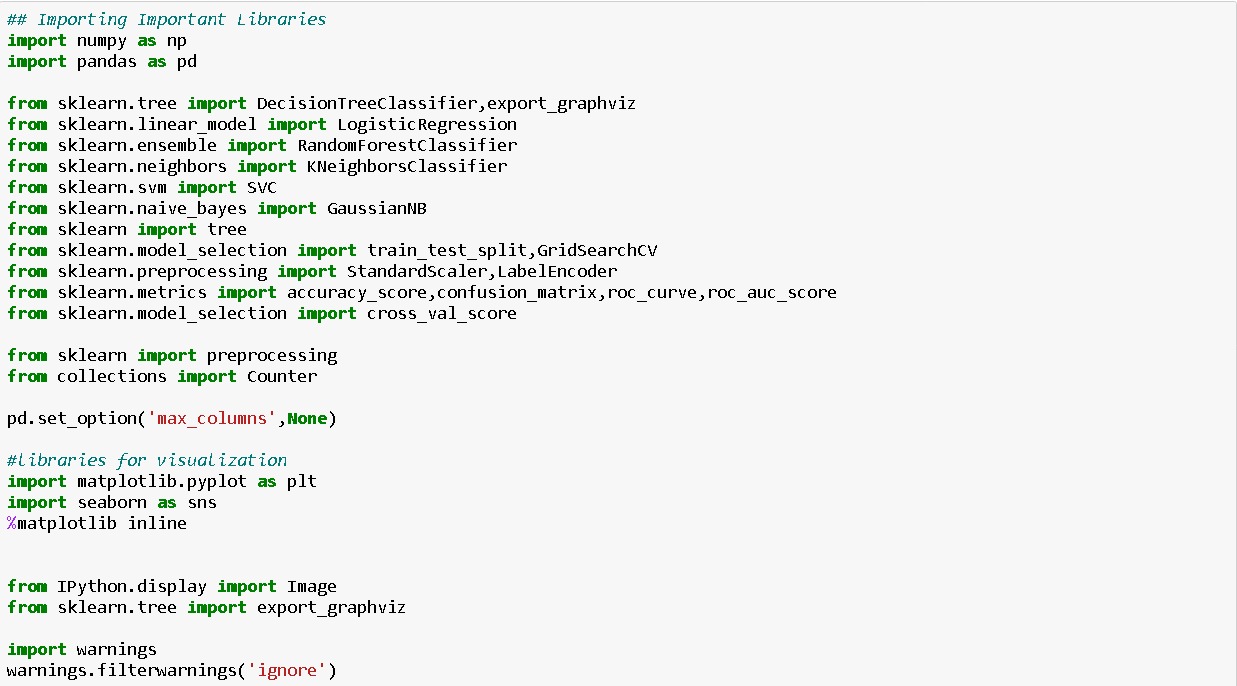
* In this case The Bank wants to automate the loan eligibility process based on customer detail provided while filling online application form.
* And to automate this process, they have given a problem to identify the customers segments, those are eligible for loan amount so that they can specifically target these customers.
* It’s a classification problem, given information about the application we have to predict whether the they’ll be to pay the loan or not.

**Steps to Proceed:**

* We’ll start by importing important Liberalises,
* exploratory data analysis,
* then data pre-processing and
* finally, we’ll be testing different models such as Logistic regression and decision trees.

**Data Analysis:**

We’ll start import the necessary libraries and then load the data:



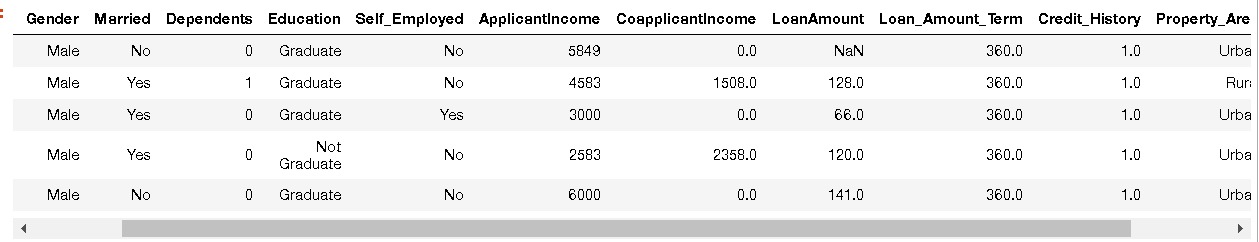
The data consists of the following rows:

Independent Variable

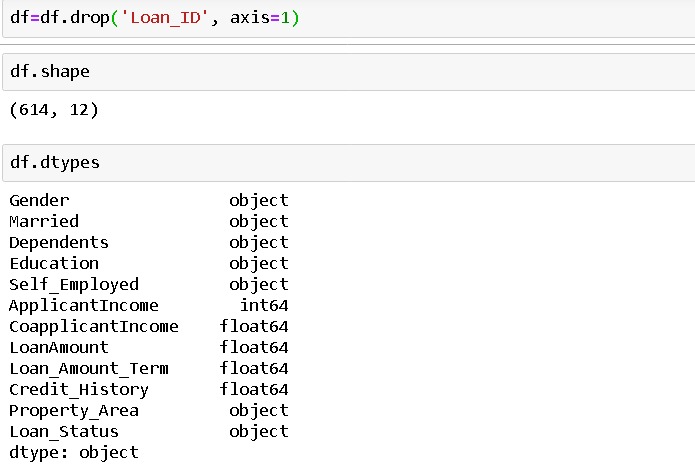
* Loan\_ID – individual Loan ID
* Gender – Male/Female
* Married – (Yes/No)
* Dependents – Number of dependents
* Education – (Graduate/ Under Graduate)
* Self\_Employed - (Yes/No)
* ApplicantIncome – Applicant Income
* CoapplicantIncome – Coapplicant Income
* Loan\_Amount – Loan Amount
* Loan\_Amount\_Term – Term of loan in months
* Credit History - Credit history meets guidelines (Yes/ No)
* Property\_Area – Urban/ Rural/ Semi Urban

Dependent Variable (Target Variable)

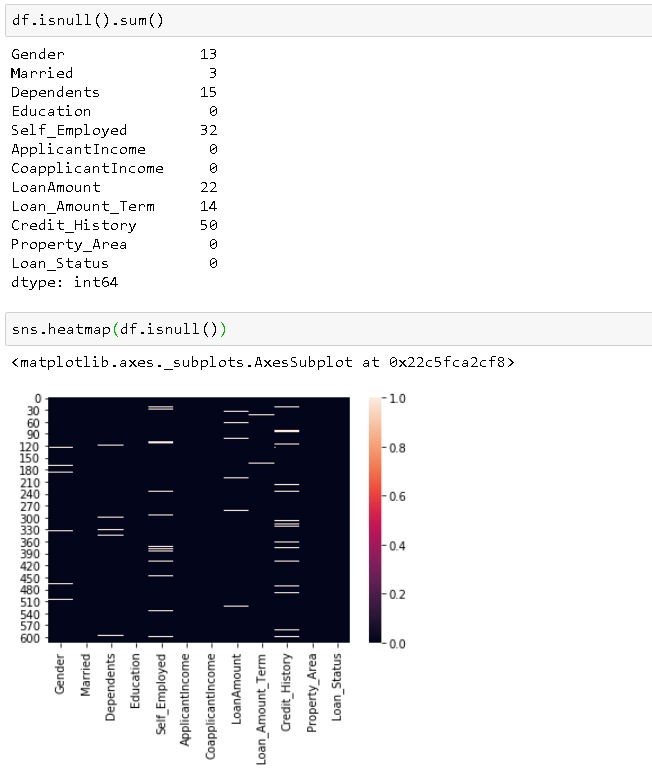
* Loan\_Status – Loan approved (Y/N)



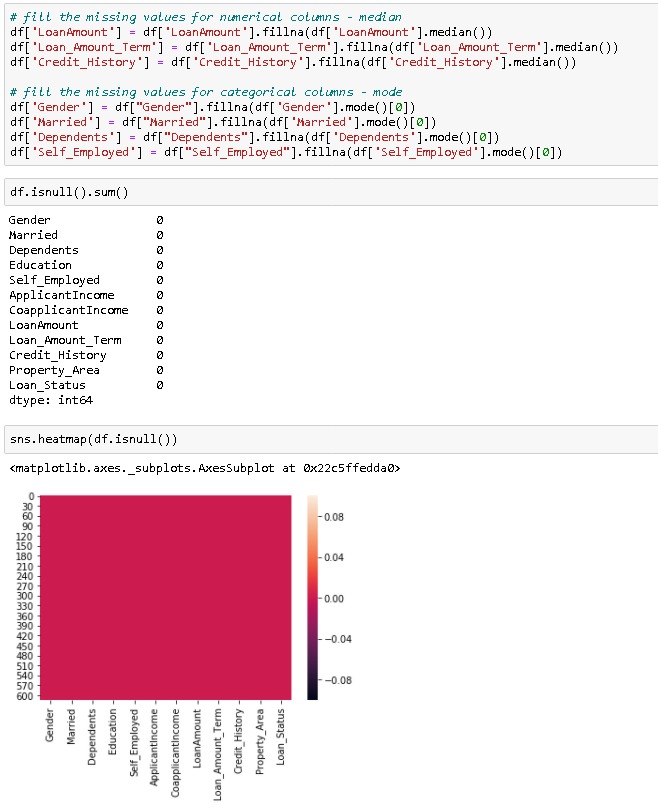
# **Exploratory Data Analysis (EDA):**

Well start removing unnecessarily data for better model building and checking the data types of available to know more about data. 

We can see that there’s some missing data

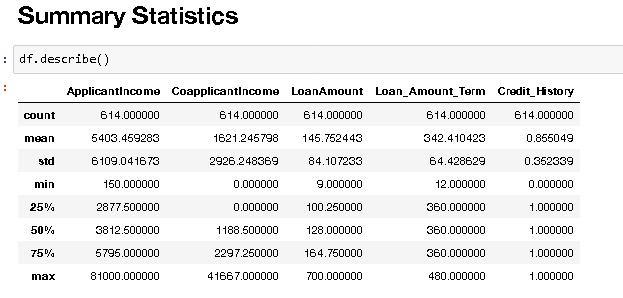


And we fix that by Filling Null values with their Median (numerical columns) & Mode values (categorical columns)

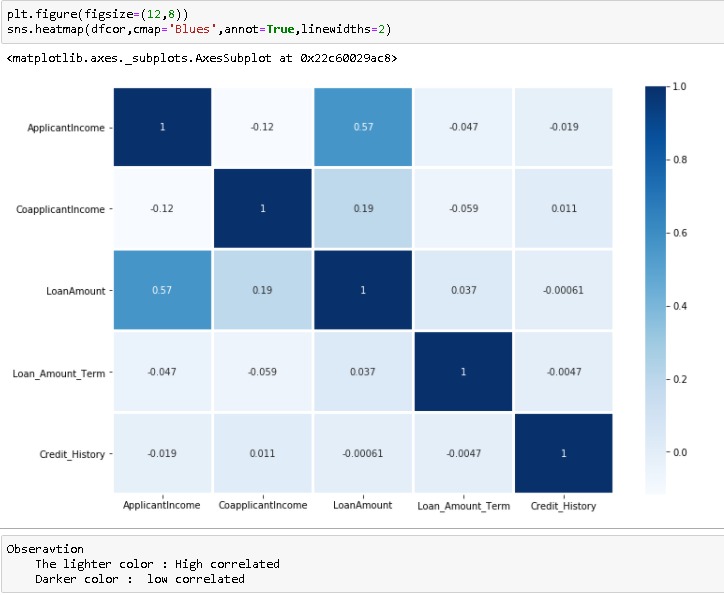
 No null values are there If there were any, you would've noticed in figure represented by different colour shade.

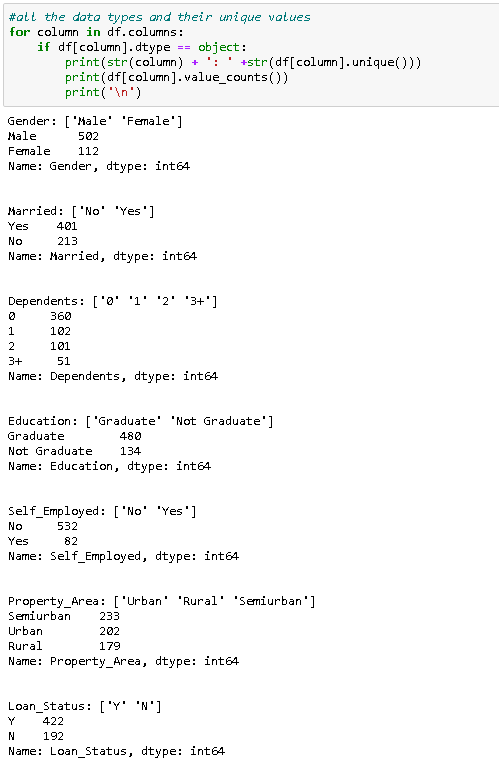
we can further explore this using the pandas describe function:

we can see that some outliers for the Applicant Income, Coapplicant income and Loan Amount. We also see that about 85% applicants have a credit history. Because the mean of Credit History field is 0.855 and it has either (1 for having a credit history or 0 for not)



Here Corelation used to find the pairwise correlation of all columns in the data frame. For any non-numeric data type columns in the data frame, it is ignored





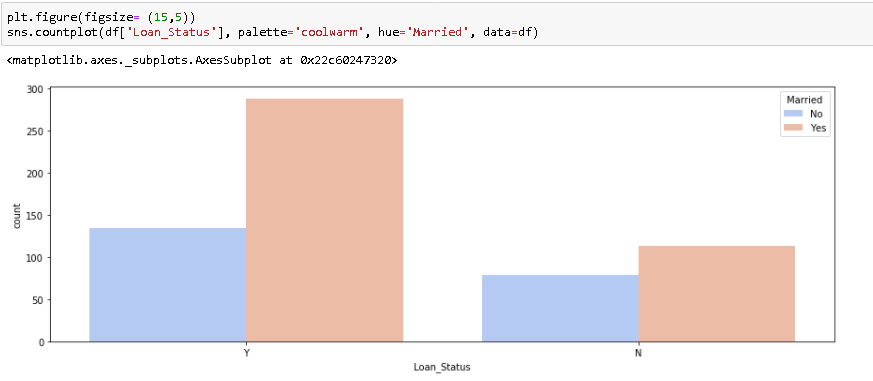
**Key Visualizations:**

Now will look at through Some insight about features

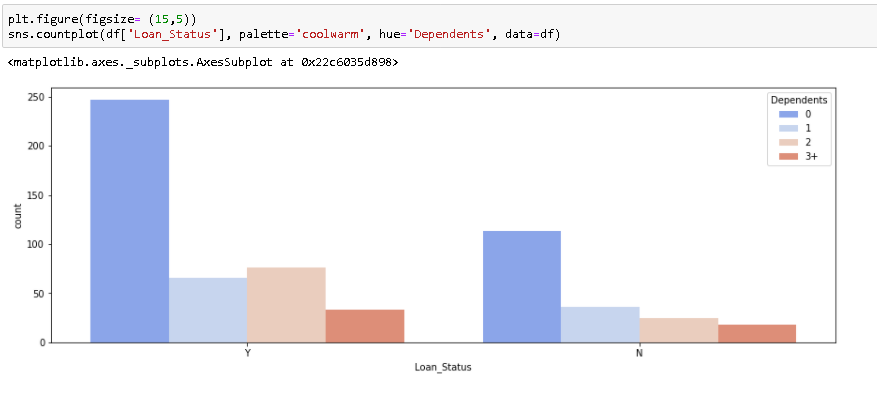
Loan status with respect Gender



Loan Status with respect to Applicant married status



Loan Status with respect to Dependents



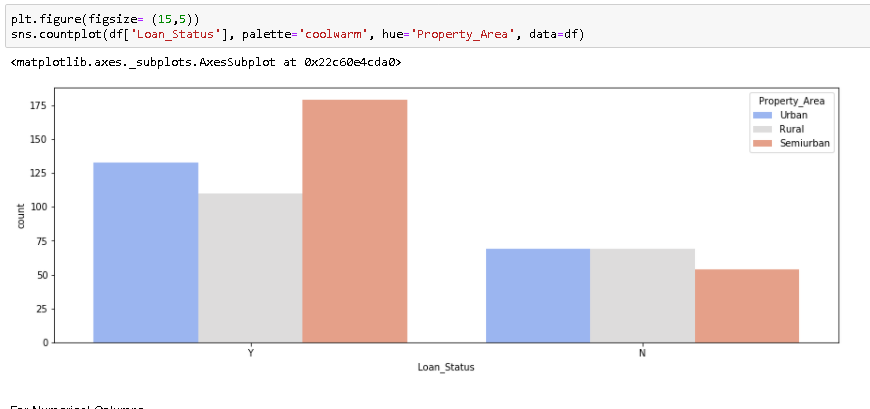
Loan status with respect to Education



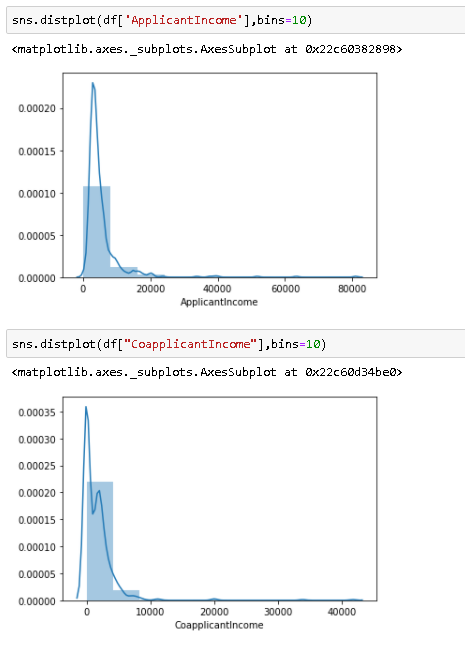
Loan Status with respect to Self Employed

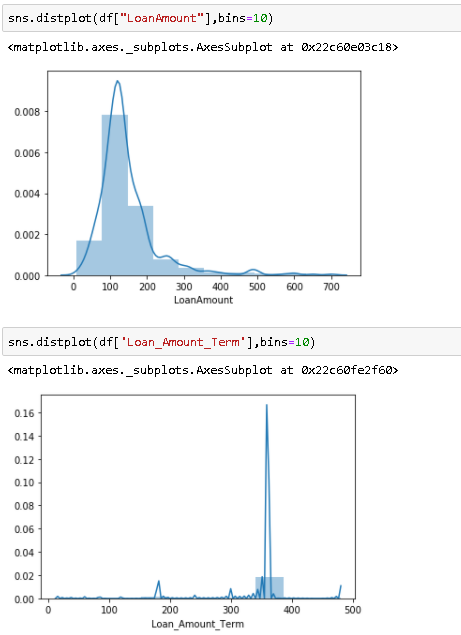


Loan Status with respect to Property Area



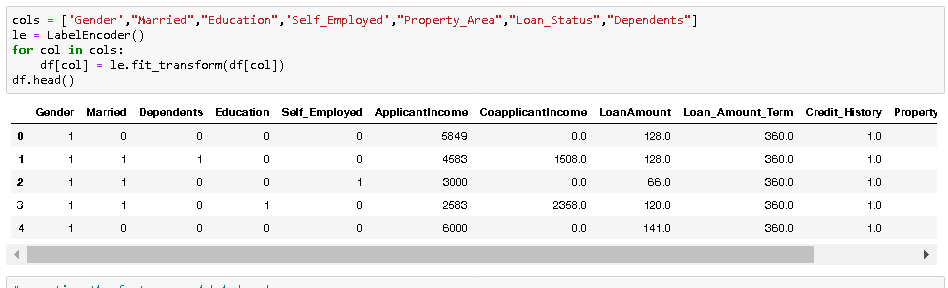
After Categorical variables It would be interesting to study the distribution of the numerical variables mainly the Applicant income and the loan amount. To do this we’ll use distplot for visualization.





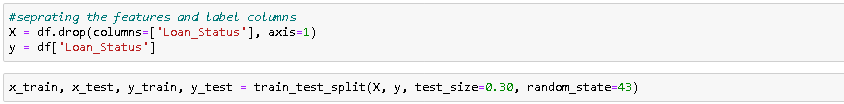
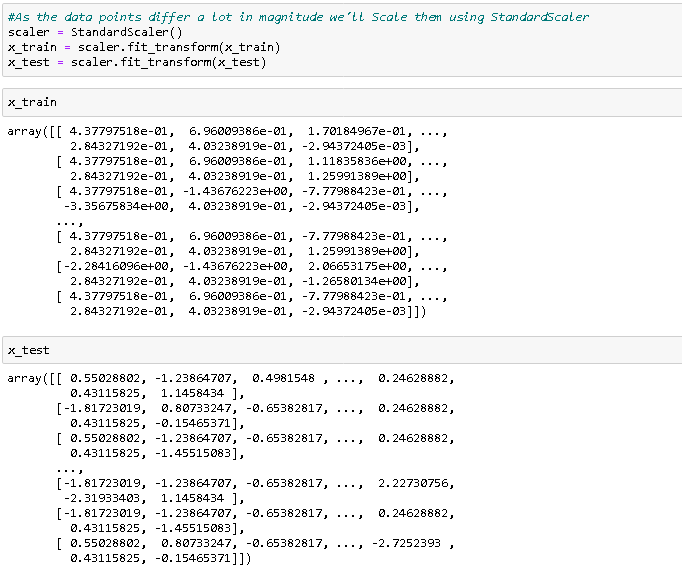
**Data Preprocessing:**

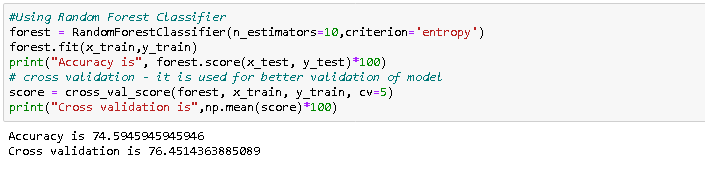
We’re going to use sklearn for our models, before doing that we need to turn all the categorical variables into numbers. We’ll do that using the Label Encoder from sklearn.

 All our variables have become numbers that our models can understand.

**Building Machine Learning Models**

Now we will train several Machine Learning models and compare their results. we need to use the predictions on the training set to compare the algorithms with each other.



**Model Hyper Parameter Tuning:**

As there is very less difference between Accuracy and Cross Validation score in Logistic Regression, we select it for Hyperparameter Tuning.

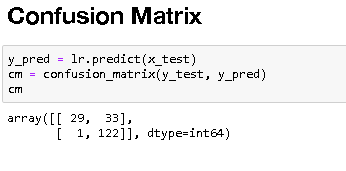


# **Further Evaluation**

## Confusion Matrix:

 A confusion matrix treats a binary process of classification. The resulting table is composed of two rows and two columns, filled with four values – true positives, false positives, true negatives and false negatives.

A confusion matrix gives you a lot of information about how well your model does.



# 

# **Conclusion:**

# We’ve gone through a good portion of the data science pipeline in this article, mainly EDA, Preprocessing and Modeling and we’ve used essential classification models such as Logistic Regression, Decision tree and Random forests.

**Summary**

We started with the data exploration where we got a feeling for the dataset, checked about missing data and learned which features are important. During this process we used seaborn and matplotlib to do the visualizations. During the data preprocessing part, we computed missing values, converted features into numeric ones, grouped values into categories and created a few new features. Afterwards we started training 5 different machine learning models, picked Logistic regression and applied cross validation on it. Then tuned its performance through optimizing it’s hyperparameter values. Lastly, we looked at its confusion matrix.

There is still room for improvement, like doing a more extensive feature engineering, by comparing and plotting the features against each other and identifying and removing the noisy features. Another thing that can improve the overall result.