**Loan Approval Prediction using Machine Learning**

LOANS are the major requirement of the modern world. By this only, Banks get a major part of the total profit. It is beneficial for students to manage their education and living expenses, and for people to buy any kind of luxury like houses, cars, etc.

But when it comes to deciding whether the applicant’s profile is relevant to be granted with loan or not. Banks have to look after many aspects.

So, here we will be using Machine Learning with Python to ease their work and predict whether the candidate’s profile is relevant or not using key features like Marital Status, Education, Applicant Income, Credit History, etc.

The dataset contains 13 features:

|  |  |  |
| --- | --- | --- |
| 1 | Loan | A unique id |
| 2 | Gender | Gender of the applicant Male/female |
| 3 | Married | Marital Status of the applicant, values will be Yes/ No |
| 4 | Dependents | It tells whether the applicant has any dependents or not. |
| 5 | Education | It will tell us whether the applicant is Graduated or not. |
| 6 | Self-employed | This defines that the applicant is self-employed i.e. Yes/ No |
| 7 | Applicant Income | Applicant income |
| 8 | Co-applicant Income | Co-applicant income |
| 9 | Loan Amount | Loan amount (in thousands) |
| 10 | Loan Amount Term | Terms of loan (in months) |
| 11 | Credit History | Credit history of individual’s repayment of their debts |
| 12 | Property Area | Area of property i.e. Rural/Urban/Semi-urban |
| 13 | Loan Status | Status of Loan Approved or not i.e. Y- Yes, N-No |

## **Importing Libraries and Dataset**

Firstly, we have to import libraries:

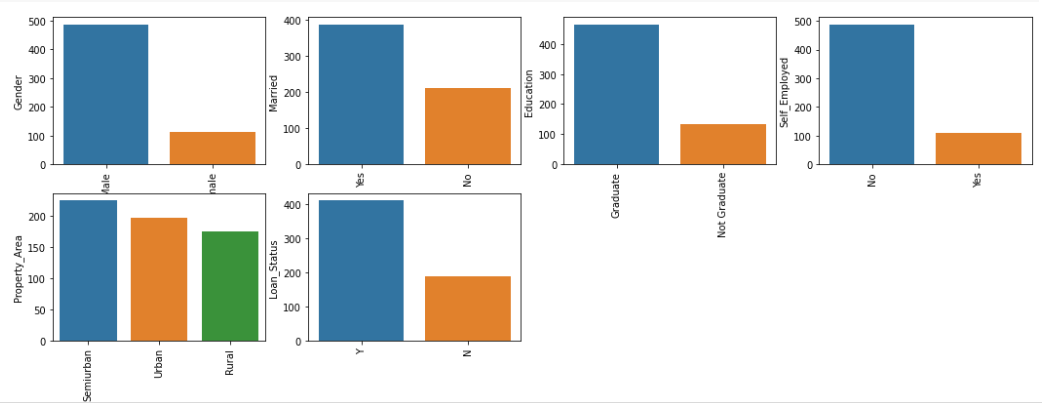
* [Pandas](https://www.geeksforgeeks.org/python-pandas-dataframe/) – To load the Data frame
* [Matplotlib](https://www.geeksforgeeks.org/python-matplotlib-an-overview/) – To visualize the data features i.e. barplot
* [Seaborn](https://www.geeksforgeeks.org/introduction-to-seaborn-python/) – To see the correlation between features using heatmap

## **Data Pre-processing and Visualization**

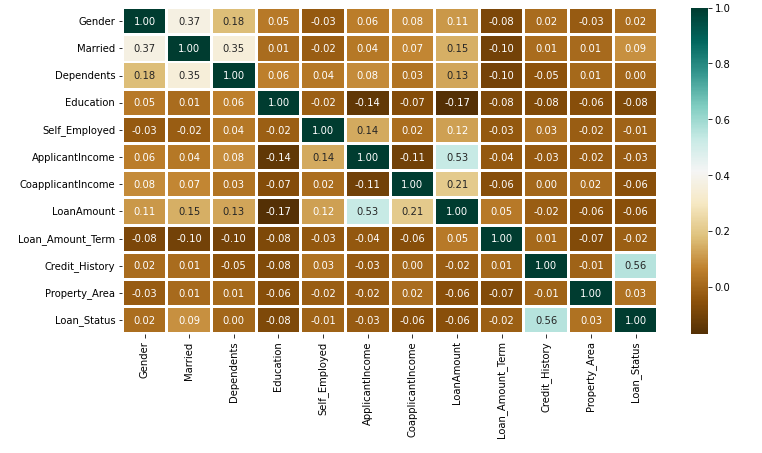
Get the number of columns of object datatype.

As Loan ID is completely unique and not correlated with any of the other column, So we will drop it using .[drop()](https://www.geeksforgeeks.org/python-delete-rows-columns-from-dataframe-using-pandas-drop/) function.

Visualize all the unique values in columns using [barplot](https://www.geeksforgeeks.org/bar-plot-in-matplotlib/). This will simply show which value is dominating as per our dataset.

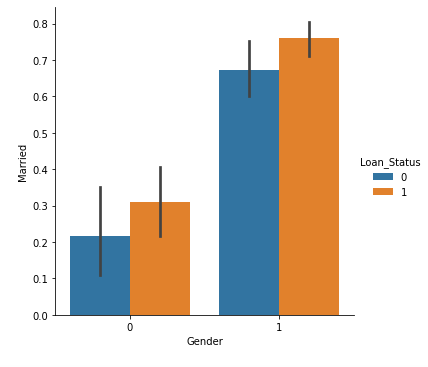


As all the categorical values are binary so we can use [Label Encoder](https://www.geeksforgeeks.org/ml-label-encoding-of-datasets-in-python/) for all such columns and the values will change into **int** datatype.

Again, check the object datatype columns. Let’s find out if there is still any left.

The above heatmap is showing the correlation between Loan Amount and Applicant Income. It also shows that Credit History has a high impact on Loan Status.

Now we will use [Catplot](https://www.geeksforgeeks.org/python-seaborn-catplot/) to visualize the plot for the Gender, and Marital Status of the applicant.



Now we will find out if there are any missing values in the dataset using below code.

for col in data.columns:

data[col] = data[col]. fillna(data[col]. mean ())

data.isna().sum()

If there is no missing value then we must proceed to model training.

## **Model Training and Evaluation**

As this is a classification problem so we will be using these models:

* [KNeighborsClassifiers](https://www.geeksforgeeks.org/k-nearest-neighbor-algorithm-in-python/)
* [RandomForestClassifiers](https://www.geeksforgeeks.org/random-forest-classifier-using-scikit-learn/)
* [Support Vector Classifiers (SVC)](https://www.geeksforgeeks.org/classifying-data-using-support-vector-machinessvms-in-python/)
* [Logistics Regression](https://www.geeksforgeeks.org/understanding-logistic-regression/)

To predict the accuracy we will use the accuracy score function from [scikit-learn](https://www.geeksforgeeks.org/learning-model-building-scikit-learn-python-machine-learning-library/) library.

## **Conclusion:**

Random Forest Classifier is giving the best accuracy with an accuracy score of 82% for the testing dataset. And to get much better results ensemble learning techniques like [Bagging](https://www.geeksforgeeks.org/ml-bagging-classifier/) and [Boosting](https://www.geeksforgeeks.org/xgboost/) can also be used.