A REPORT ON LOAN APPROVAL PREDICTION MACHINE LEARNING PROJECT

Introduction

In this article, we are going to solve the Loan Approval Prediction. This is a classification problem in which we need to classify whether the loan will be approved or not. Classification refers to a predictive modelling problem where a class label is predicted for a given example of input data.

Problem statement

Develop a machine learning model to predict the likelihood of loan approval based on applicants' information.

About dataset

The dataset is collected from-https://www.kaggle.com/datasets/rishikeshkonapure/home-loan-approval.

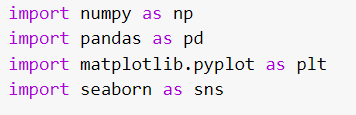
The dataset is in csv format. There is total 615 numbers of rows and 13 columns. Column names are - Loan\_ID, Gender, Married, Dependents, Education, Self\_Employed, ApplicantIncome, CoapplicantIncome, LoanAmount, Loan\_Amount\_Term, Credit\_History,Property\_Area, Loan\_Status.

These details are Gender, Marital Status, Education, number of Dependents, Income, Loan Amount, Credit History, and others. They have a presence across all urban, semi-urban and rural areas. The customer first applies for a home loan and after that, the company validates the customer eligibility for the loan. We need to predict our Target label which is “Loan Status”. Loan status can have two values: Yes or NO. Yes: if the loan is approved and NO: if the loan is not approved. So using the training dataset we will train our model and try to predict our target column that is “Loan Status” on the test dataset.

Categorical Columns: Gender (Male/Female), Married (Yes/No), Number of dependents (Possible values:0,1,2,3+), Education (Graduate / Not Graduate), Self-Employed (No/Yes), credit history(Yes/No), Property Area (Rural/Semi-Urban/Urban) and Loan Status (Y/N)(i. e. Target variable)

Numerical Columns: Loan ID, Applicant Income, Co-applicant Income, Loan Amount, and Loan amount term.

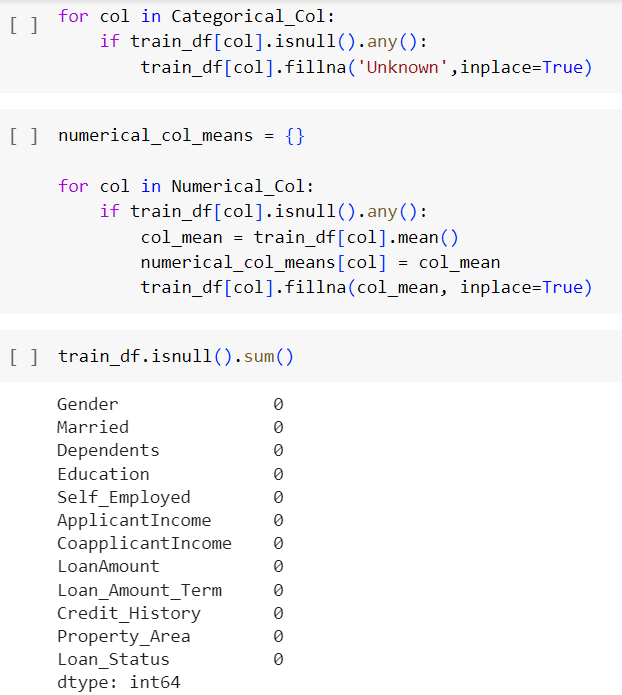
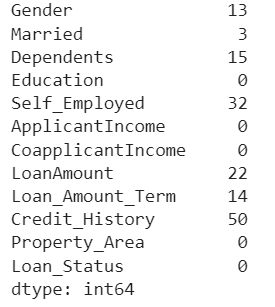
Essential Python Libraries



Data preprocessing

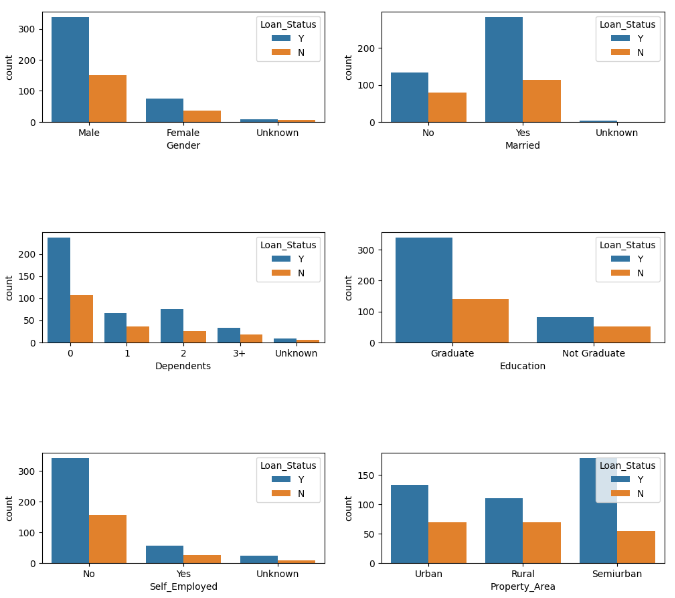
* Data cleaning

There are several columns present in the Training Data.csv dataset where null value is present. For removing the null values, we replace the null value present in categorical data by “Unknown” and in numerical data by the mean value of that column.



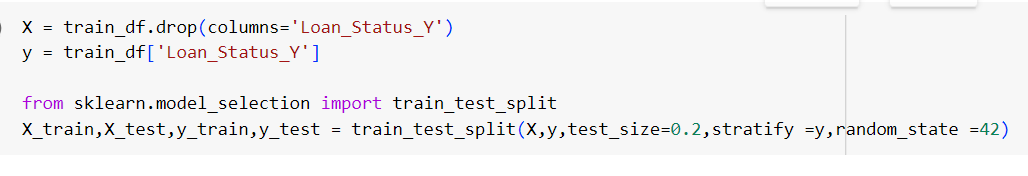
Exploratory Data Analysis (EDA)

* Data visualization



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

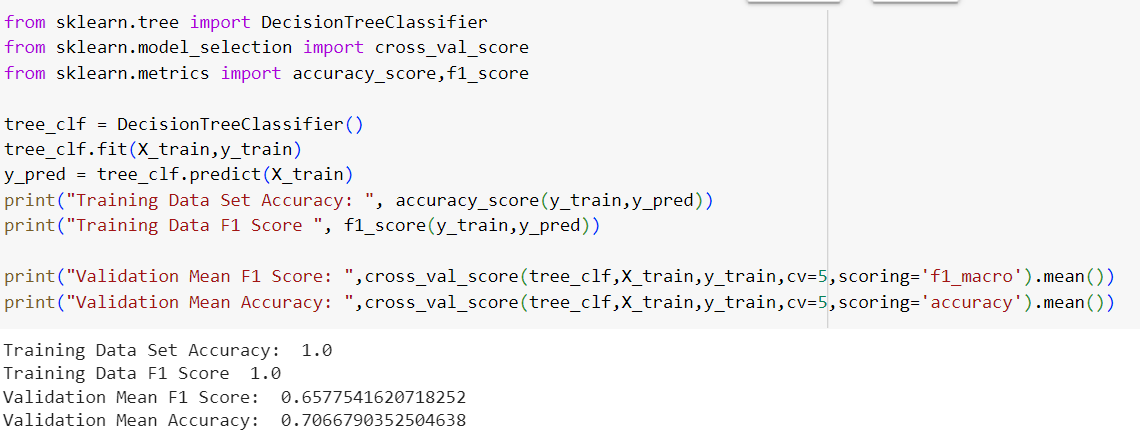
* Data splitting



We split the dataset as “test\_size=0.2” means 20% data will be used for testing purpose and rest 80% for training the model. And we used “random\_state=42” to avoid getting different result each time running the program.

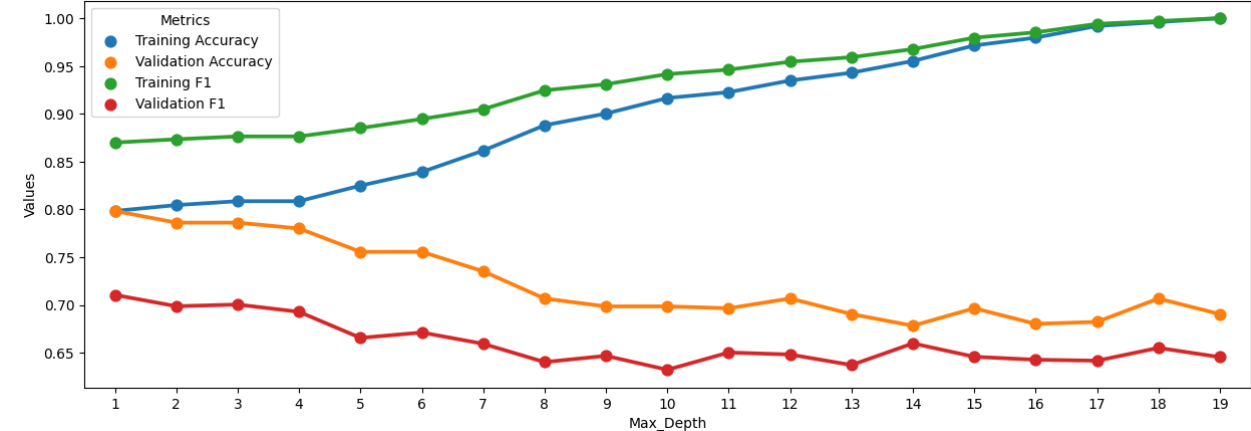
* Training & Accuracy prediction

Here we used Decision Tree Classifier for training the model.



We can see from above metrics that Training Accuracy > Test Accuracy with default settings of Decision Tree classifier. Hence, model is overfit. We will try some Hyper-parameter tuning and see if it helps.

We change the depth of the decision tree from 1 to 19 and check which value for the max depth we get the optimal accuracy and F1 value-



From the graph we can conclude that keeping 'Max\_Depth' = 3 will yield optimum Test accuracy and F1 score Optimum Test Accuracy ~ 0.805; Optimum F1 Score: ~0.7

Conclusion

After the Final Submission of test data, accuracy score of the model is 85%.

Hyper-parameter tuning helped increasing the accuracy of the model and

Decision tree classifier worked better than all other Ensemble models.