











Loan Approval Prediction Machine Learning

Introduction:

In this article, we are going to solve the Loan Approval Prediction problem. 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.

Content:

-  Understanding the problem statement
-  Dataset
-  Load Python Libraries
-  Load Training, Testing datasets
-  Data Preprocessing
-  Exploratory data analysis (EDA)
-  Feature Engineering
-  Build Machine Learning Model
-  Predictions on the test dataset
-  Conclusion

Understanding the problem statement:

Dream Housing Finance company deals in all kinds of home loans. 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. The company wants to automate the loan eligibility process (real-time) based on customer detail provided while filling out online application forms. These details are Gender, Marital Status, Education, number of Dependents, Income, Loan Amount, Credit History, and others.

To automate this process, they have provided a dataset to identify the customer segments that are eligible for loan amounts so that they can specifically target these customers.

As mentioned above this is a Binary Classification problem in which 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

NO: if the loan is not approved

So using the training dataset we will train our model and try to predict our target column “Loan Status” on the test dataset.

Load python Libraries:

Load the following necessary Python Libraries .

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns
from sklearn.experimental import enable_iterative_imputer
from sklearn.impute import IterativeImputer
from sklearn.model_selection import train_test_split
```

Load training/test dataset:

```
Train=pd.read_csv("file path")
```

```
Test=pd.read_csv("file path")
```

Data Preprocessing:

- Concatenate the train and test data for data preprocessing
- drop the unwanted column
- Identify missing values
- Imputing the missing values

- Fill null values with mode

Exploratory Data Analysis (EDA):

- Split the data to new_train and new_test to perform EDA
- Mapping 'N' to 0 and 'Y' to 1

Univariate Analysis Observations:

- More Loans are approved than Rejections
- Count of Male applicants is more than Female
- Count of Married applicant is more than Non-married
- Count of graduate is more than non-Graduate
- Count of self-employed is less than that of Non-Self-employed
- Maximum properties are located in Semiurban areas
- Credit History is present for many applicants
- The count of applicants with several dependents=0 is maximum.

Bivariate Analysis:

- Correlation Matrix

Building Machine Learning Model:

- Creating X (input variables) and Y (Target Variable) from the new_train data.
`x=new_train.drop("Loan_status",axis=1)`
`y=new_train("Loan_status")`
- Using train test split on the training data for validation
from sklearn.model_selection train_test_split
`x_train, x_test,y_train,y_test=train_test_split(x,y,test_size=0.3)`

ML Algorithms for training:

We use multiple algorithms for training like Decision Tree, Random Forest, SVC, Logistic Regression, XGB Regressor, etc.

Among all the algorithms logistic regression performs best on the validation data with an accuracy score of 82.7%.

After getting an accuracy of 82.7% , tune it to improve accuracy score using GridSearchCV. After fine-tuning the logistic regression model the accuracy score improved from 82.7% to 83.24%.

Predict the test data.

Conclusions:

After the Final Submission of test data, my accuracy score was 78%. Feature engineering helped me increase my accuracy. Logistic Regression worked better than all other Ensemble models.