***Loan Approval Prediction***

The loan prediction machine learning model can be used to assess a customer's loan status and build strategies. This model extracts and introduces the essential features of a borrower that influence the customer's loan status. Finally, it produces the planned performance (loan status). These reports make a bank manager's job simpler and quicker.

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

**Table of Content**

1. Understanding the problem statement
2. About the dataset
3. Load essential Python Libraries
4. Load Training/Test datasets
5. Exploratory data analysis (EDA).
6. Feature Engineering.
7. Build Machine Learning Model
8. Make predictions on the test dataset
9. Prepare submission file
10. Conclusion

# Problem Statement:

This dataset includes details of applicants who have applied for loan. The dataset includes details like credit history, loan amount, their income, dependents etc.

Independent Variables:

* Loan\_ID
* Gender
* Married
* Dependents
* Education
* Self\_Employed
* ApplicantIncome
* CoapplicantIncome
* Loan\_Amount
* Loan\_Amount\_Term
* Credit History
* Property\_Area

Dependent Variable (Target Variable):

* Loan\_Status

You have to build a model that can predict whether the loan of the applicant will be approved or not on the basis of the details provided in the dataset.

## *1. Understanding the Problem Statement*

This is a classification problem in which we need to classify whether the loan will be approved or not. Classification refers to a predictive modeling problem where a class label is predicted for a given example of input data.

The bank 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 dataset we build our model and try to predict our target column that is “Loan Status” on the test dataset.

***2. Exploratory Data Analysis (EDA)***

Importing Libraries for Data Analysis

**Pandas:** Pandas is a Python package to work with structured and time series data. The data from various file formats such as csv, json, sql etc can be imported using Pandas. It is a powerful open source tool used for data analysis and data manipulation operations such as data cleaning, merging, selecting as well wrangling.

**Seaborn:** Seaborn is a python library for building graphs to visualise data. It provides integration with pandas. This open source tool helps in defining the data by mapping the data on the informative and interactive plots. Each element of the plots gives meaningful information about the data.

**Sklearn:** This python library is helpful for building machine learning and statistical models such as clustering, classification, regression etc. Though it can be used for reading, manipulating and summarizing the data as well, better libraries are there to perform these functions.

We run the *df.info()* command, which gives us the complete information about the dataset column names, number of values present in each column, and data types of each column

We observe that we have 614 rows in our dataset. We have 13 features in total out of which we have 12 independent variables and 1 dependent variable i.e. Loan\_Status. In our dataset 8 columns have object data type, 4 columns have float data type, and 1 column has integer data type.

**Feature description**:

|  |  |
| --- | --- |
| **Variables** | **Description** |
| Loan\_ID | Unique loan ID |
| Gender | (Male/Female) |
| Married | Applicant’s Marital Status (Y / N) |
| Dependents | Number of dependents |
| Education | Applicant’s Education (Graduate / Under Graduate |
| Self\_Employed | Self employed (Y/ N) |
| ApplicantIncome | Applicant Income |
| CoapplicantIncome | Co-applicant Income |
| Loan\_Amount | Loan amount in thousands |
| Loan\_Amount\_Term | Term of loan in months |
| Credit\_History | Credit history meets guidelines |
| Property\_Area | Urban / Semi Urban / Rural |
| Loan\_Status | Loan approved (Y / N) |

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. our Target variable)

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

***3. Data Pre-processing:*** We run the *df.isnull().sum()* command to check the count of NaN (null) values in our dataset.

Loan\_ID 0

Gender 13

Married 3

Dependents 15

Education 0

Self\_Employed 32

ApplicantIncome 0

CoapplicantIncome 0

LoanAmount 22

Loan\_Amount\_Term 14

Credit\_History 50

Property\_Area 0

Loan\_Status 0

We observe that 4 Categorical columns and 3 Numeric columns have Null values

**Imputing the missing values:**

For categorical values: We use most frequent category because our dataset is small and missing values also in small amount. This method works well for small dataset.

For numeric values: We impute missing values with mean.

After deal with null values we drop the unwanted columns: We drop the Loan\_ID column. Now we have 12 columns.

### *4. Visualization*

Proceeding with the plotting and analyzing the data using seaborn, matplotlib libraries. Here we use count plot for categorical data and distplot numeric data.

**Bivariate Analysis:** Bivariate analysis is finding some kind of empirical relationship between two variables. Specifically the dependent vs independent Variables

**Gender:** There are 502 males and 112 females in our dataset. The count plot shows that loan approval for male is higher than female.

**Married:** According to the dataset 401 people are married and 213 people are unmarried. The count plot shows that loan approval for married is higher than unmarried.

**Dependents:** In dependent feature there are 4 categories i.e. 0, 1, 2 and 3+ it means number of dependents, 360 applicant have 0 dependent, 102 applicant have 1 dependent, 101 applicant have 2 dependents, 51 applicant have 3+ dependents. The count plot shows that loan approval for 0 dependent is higher than other and loan approval for 3+ dependents is lower than others.

**Education:** The value count shows that 480 applicants are Graduate and 134 applicants are Not Graduate The count plot shows loan approval for graduate people is higher than not graduate.

**Self\_Employed:** Out of 614 applicants 532 applicants are not self-employed and 82 applicants are self-employed. The count plot shows loan approval for the applicants who are not self-employed is higher than self-employed applicants.

**Property\_Area:** Value count of semi urban – 233, Urban – 202 and Rural – 179. The count plot shows maximum properties are located in Semiurban areas than rural and urban property area and loan approval for the rural is mimimum than semi urban and urban property area.

**Credit\_History:** It seems people with credit history as 1 are more likely to get their loans approved.

**Target Variable –** **Loan Status:** Loan status is the independent variable which is our target variable as well. The count plot shows that 422 people out of 614 is approved the loan and 192 is not approved the loan.

We make the below mentioned observations using the plots above –

1. Count of Male applicants is more than Female
2. Count of Married applicant is more than Non-married
3. The count of applicants with several dependents=0 is maximum.
4. Count of graduate is more than non-Graduate
5. Count of self-employed is less than that of Non-Self-employed
6. Maximum properties are located in Semi urban areas
7. Credit History is present for many applicants
8. More Loans are approved vs Rejected

Numerical Features

**ApplicantIncome:** It can be inferred that most of the data in Applicant income is towards left which means it is not normally distributed

**CoapplicantIncome:** It can be inferred that most of the data in Co-applicant income is towards left which means it is not normally distributed

For correlation we use the heatmap and bar chart:The variables with darker color means their correlation is more

For outliers we use box plot: The boxplot confirms the presence of outliers

***5. Pre-processing Pipeline.***

After visualization we convert the dataset into numeric form by using

***from sklearn.preprocessing import OrdinalEncoder***

To convert categorical text data into model-understandable numerical data, we use the Ordinal Encoder class. So all we have to do, import the OrdinalEncoder class from the sklearn library, fit and transform the column of the data, and then replace the existing text data with the new encoded data.

After Encoding we check the outliers and considering the outlier removal using zscore method. After removing the outliers the dataset shape is 577 rows and 12 columns. If we remove outliers data loss score is 6 % which is ok. So, we decide to remove outlier store the new dataset in df for model building.

Once again we visualize loan status using count plot Now the count plot shows that 398 people out of 577 is approved the loan and 179 is not approved the loan.

After that we drop the target variable which is Loan\_Status

Spliting data by train\_test\_split: Now we split the data into test and train and drop the price column from the test set because we have to predict Loan approved or loan not approved with our test data set

***6. Model Building***

Building Machine Learning Models: After separating independent variables and dependent variables (target variable) Using train test split on the training data for validation. We use two machine learning models for the prediction of loan approvals. Below are the descriptions of the models used

**Logistic Regression:** This is a classification algorithm which uses a logistic function to predict binary outcome (True/False, 0/1, Yes/No) given an independent variable. The aim of this model is to find a relationship between features and probability of particular outcome. The logistic function used is a logit function which is a log of odds in the favor of the event. Logit function develops a s-shaped curve with the probability estimate similar to a step function

**Decision Tree Classifier:** This is a supervised machine learning algorithm mostly used for classification problems. All features should be discretized in this model, so that the population can be split into two or more homogeneous sets or subsets. This model uses a different algorithm to split a node into two or more sub-nodes. With the creation of more sub-nodes, homogeneity and purity of the nodes increases with respect to the dependent variable.

***7. Concluding Remarks:***

Logistic Regression:

Accuracy Score: 0.8850574712643678

Confusion Matrix: [[23 19]

[ 1 131]]

Cross validation score: 0.816292735042735

Decision Tree Classifier:

Accuracy Score: 0.735632183908046

Confusion Matrix: [[29 13]

[33 99]]

Cross validation score: 0.7295673076923077

Logistic Regression worked better than Decision Tree Classifier