**Loan Approval Prediction**

**By**

**Pratima Paudel**

**Agenda**

The main purpose of this project is to do the analysis on the data and to decide whether the applicant is suitable for loan approval on not. Manual Process is effective, but it takes a huge amount of time for decision making when there are large number of applicants present. So, Basis on the number of criteria by analyzing different attributes of customer application( Credit History, Education, Loan Amount...) models can be used to assess the customer loan status and building future strategies such as focusing on specific customers to reach out to enhance and make loan approval process less risky.

With the help of Datamining techniques, dataset can be analyzed for preprocessing which includes extracting various information and normalizing the data and then on the basis of models and comparison of the results with high accuracy, appropriate decision can be made.

**Data**

|  |  |
| --- | --- |
| **Variable** | **Variable Description** |
| Loan\_ID | Unique Loan ID of the applicant |
| Gender | Male/Female |
| Married | Marital Status(Y/N) |
| Dependent | Number of Dependents |
| Education | Education Status of Applicant(Graduate/Undergraduate) |
| Self\_Employed | Yes/No |
| ApplicantIncome | Income of the applicant |
| CoapplicantIncome | Income of the co applicant |
| LoanAmount | Amount to be borrowed as loan |
| Loan\_Amount\_Term | Term in Months |
| Credit\_History | Credit history meets guidelines |
| Property\_Area | Urban/Semi Urban/Rural |
| Loan\_Status | Loan Approved(Yes/No) |
|  |  |

**Libraries Used**

|  |  |
| --- | --- |
| **Libraries** | **Description** |
| Pandas | Library for fast data analysis, manipulation and filtering |
| NumPY | Library for mathematical and logical operation on arrays |
| Matplotlib | Library for creating static, interactive and animated visualisation |
| SkiKit Learn | Library with efficient tools for machine learning and statistical models including range of supervised and unsupervised learning algorithms |

**Models Used**

1. **Decision Tree Classification**



* **Used for classification an regression problems**
* **Internal nodes represent the attributes of the dataset**
* **Branch represent the outcomes**
* **Each leaf node represent a class label**
* **It takes the decision on the basis of Yes or No(conditional)**
* **Decision Tree first select the best attribute based on the attribute selection measure(Entropy, information gain and Gini Index)**
* **Attribute with low entropy and high information gain is preferred**
* **Classifier then asks a relevant question to divide the dataset based on the answer**

1. **Naïve Based Classification**

Chart

Description automatically generated

* **Probabilistic machine learning algorithm based on the Bayes Theorem**
* **Assumptions made**

**-Independent**

**-Equal**

**-Contribution to the outcome**

***Source-*** Kaggle

***Language Used*** – Python

**Architecture**

**Data Preprocessing**

**Data Understanding**

1. **Reading the train dataset**

Graphical user interface, text

Description automatically generated with medium confidence

1. **Exploring the train dataset**

2.1) Finding the total number of rows and columns

Graphical user interface

Description automatically generated with medium confidence

2.2) Exploring the other features of Data

Graphical user interface

Description automatically generated with low confidence

2.3)Exploring the frequency distribution of Non numerical attributes

Table

Description automatically generated

2.4)Exploring the frequency distribution of numerical attributes

Table

Description automatically generated

Chart

Description automatically generated

Table

Description automatically generated

Through BoxPlot and histogram, we understand, normalization of data needs to be done as there are many outliers

Table

Description automatically generated

Chart, box and whisker chart

Description automatically generated

Through Boxplot and histogram, we understand, normalization of data needs to be done as there are many outliers

Chart, histogram

Description automatically generated

2.4)Normalizing the data

Chart, histogram

Description automatically generated

Chart, histogram

Description automatically generated

2.5)Identifying missing values in each attribute

Table

Description automatically generated

2.6)Removing the missing values by replacing it with mean or mode accordingly

Mode -For Categorical attribute

Mean- For Numerical attribute

Text

Description automatically generated

2.7)Now we will normalize the income by first adding applicant income and co-applicant income

Chart, histogram

Description automatically generated

2.8)Checking the data again after normalization

A screenshot of a computer

Description automatically generated with medium confidence

1. **Label Encoding** – Providing labels to categorical attributes for the convenience of the model training

A picture containing table

Description automatically generated

1. **Data Splitting into train and Test data**

Train data – data that will have the output label and model will be trained from that class label

X\_train and Y\_train will be two variables.

Y\_Train will have the class label ‘Loan Status’ that will be trained on the data through X\_train

X-Dependent variable

Y-Independent variable

Graphical user interface, text, application

Description automatically generated

Splitting the test data and train data by 80% of training and 20% of testing

Setting the random state to Zero so as to give the same result in every cycle

As it controls the shuffling of data before train and test split.

Graphical user interface, text, application, email

Description automatically generated

5)**Scaling the Data**

To perform better before model is applied

Graphical user interface, text, application

Description automatically generated

1. **Using Cross validation Function**

For better accuracy test of model as it creates multiple splits and test on each split and calculate the accuracy by taking the average of all results

Text

Description automatically generated

1. **Calling Decision Tree Classifier model**

Graphical user interface, text, application, email

Description automatically generated

As the accuracy is 67.47 which is less, Hence we will call another model

1. **Calling Naïve Bayes Model**

Text

Description automatically generated

1. **Prediction**

Labelling the class (y\_test) on the basis of x\_test

A picture containing application

Description automatically generated

So the outcome tells us the loan prediction as per Naïve Bayes Algorithm.