Loan prediction system

Project submitted to the

SRM University – AP, Andhra Pradesh

for the partial fulfilment of the requirements to award the degree of

Bachelor of Technology

In

Computer Science and Engineering

School of Engineering and Sciences

Submitted by

N.SRI PRANAV (AP21110010885)

G.SAI KRISHNA (AP21110010892)

R.MADHU SUDHAN (AP21110010893)

S.KRISHNA MANIKANTA (AP21110010895)

Under the Guidance of

Dr. Mudassir Rafi

SRM University–AP

Neerukonda, Mangalagiri, Guntur

Andhra Pradesh – 522 240

Nov, 2023

Certificate

Date:

This is to certify that the work present in this Project entitled "**LOAN PREDICTION SYSTEM**" has been carried out by N.SRI PRANAV ,G.SAI KRISHNAR.MADHUSUDHAN, S.KRISHNA MANIKANTA

under my/our supervision. The work is genuine, original, and suitable for submission to the SRM University – AP for the award of Bachelor of Technology/Master of Technology in School of Engineering and Sciences.

Supervisor
Dr. Mudassir Rafi
Designation- Asst.Professor dept of CSE,
Affiliation.

Acknowledgements

Foremost, we express our deepest appreciation to our mentor, Dr. Mudassir Rafi, whose wisdom, insights, and steadfast assistance has been crucial in forming our project. Your guidance has illuminated our path and instilled in us a profound appreciation for the intricacies of machine learning and exploratory data analysis.

We extend our gratitude to our guide, Dr. Mudassar Rafi , who has generously shared their expertise, serving as our guiding light through this intricate journey. Your mentorship has been invaluable, and your patience in addressing our queries has been greatly appreciated.

Table of Contents

Abstract

A loan is basically borrowing money from the bank or other organization with a commitment of repaying it only on a monthly basis with interest added to it. Loans are a significant income for the banks. The repayment of the loan is very important for them so they have to verify the worthiness of the customer before accepting the loan. The project's main goal is to forecast whether the bank would accept loans for customers or not. Banks follow a set of standards and undertake background checks before lending money to customers. The rate of loan applications has increased significantly in recent years. The approval of loans is always highly risky. The necessity of customers repaying their loans is well understood by bank managers. Even after taking several measures and reviewing the loan application data, loan approval decisions are not always correct. As a result, the purpose of this study is to estimate loan eligibility using a decision tree , Random forest, Gaussian Naive Bayes machine learning models

Introduction

Loan is one of the ways to get funding for the needs of the people. The banks will beflooded for the loan request but they have to go through various checks like the credit\score of the person and all the factors because the repayment of loan is most important For the bank’s survival. Loan recovery is a major contributor in a bank's financials. It is quite difficult to predict if the customer will be able to repay the loan. So in this project,We predicted whether the bank can approve a loan for a customer or not based on a few parameters like customer education background, source of income, loan amount, andthe credit history of the customer using a machine learning model. First, we preprocessed the data by applying data preprocessing and feature engineering techniques. Then we used the Decision Tree Machine Learning algorithm on the preprocessed data to forecast whether the loan would be approved or not, and we calculated the algorithm's accuracy.

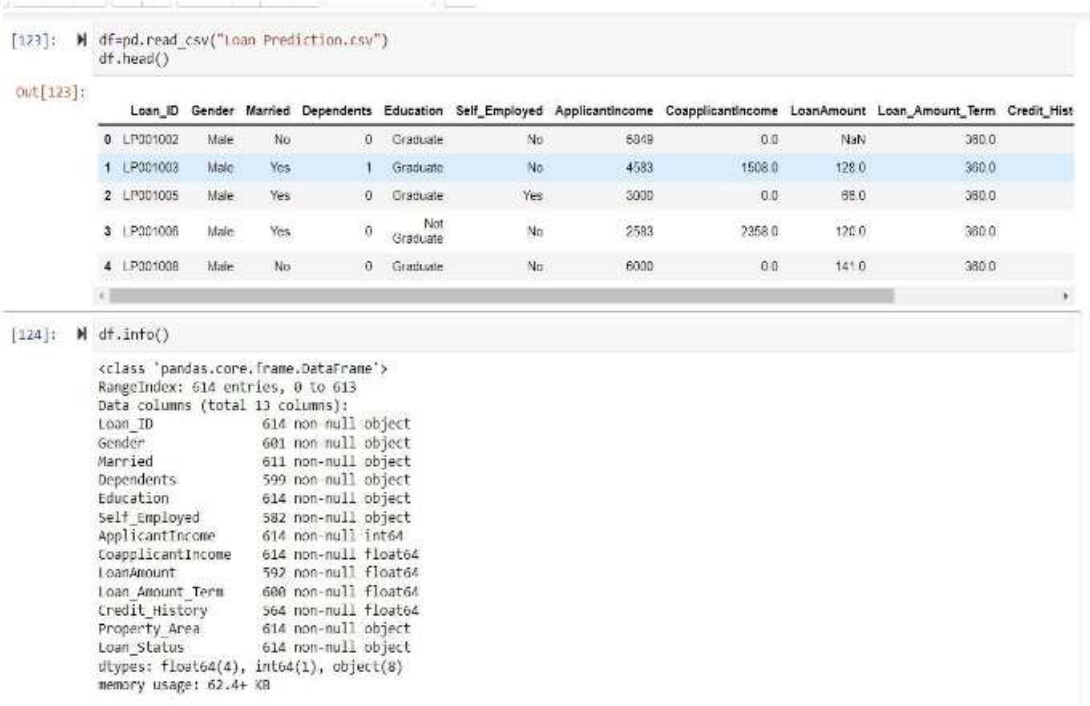
Decision tree is a classification algorithm which constructs a tree based on the training data and makes predictions.The models are compared using a variety of measures, including Mean Square Error (MSE), Root Mean Square Error (RMSE), and Root Mean Square Error (RMSE) (RMSE

Steps involved:

1. DataSet selection
2. Data Preprocessing
3. Feature Selection
4. Applying ML model
5. Error calculation

Methodology

DATASET SELECTION:
Finding a meaningful dataset for the topic we've chosen is the most important task in every machine learning research. The bank loan dataset was downloaded from the Kaggle website.



```
[123]: df=pd.read_csv("Loan Prediction.csv")
       df.head()
```

Out[123]:

	Loan_ID	Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome	CoapplicantIncome	LoanAmount	Loan_Amount_Term	Credit_Hist
0	LP001002	Male	No	0	Graduate	No	5849	0.0	NaN	360.0	
1	LP001003	Male	Yes	1	Graduate	No	4583	1508.0	128.0	360.0	
2	LP001005	Male	Yes	0	Graduate	Yes	3000	0.0	66.0	360.0	
3	LP001006	Male	Yes	0	Not Graduate	No	2583	2358.0	120.0	360.0	
4	LP001008	Male	No	0	Graduate	No	6000	0.0	141.0	360.0	

```
[124]: df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 614 entries, 0 to 613
Data columns (total 13 columns):
Loan_ID              614 non-null object
Gender               601 non-null object
Married              611 non-null object
Dependents           599 non-null object
Education            614 non-null object
Self_Employed        582 non-null object
ApplicantIncome      614 non-null int64
CoapplicantIncome    614 non-null float64
LoanAmount           592 non-null float64
Loan_Amount_Term     600 non-null float64
Credit_History       564 non-null float64
Property_Area        614 non-null object
Loan_Status          614 non-null object
dtypes: float64(4), int64(1), object(8)
memory usage: 62.4+ KB
```

2.2 DATASET DESCRIPTION:

The dataset we have taken from kaggle consists of a total of 614 rows and 13

columns. (NOA) and negation-of-verb (NOV).

NUMBER	DATA COLUMNS	DATA TYPE
1	Loan_ID	Object
2	Gender	Object
3	Married	Object
4	Dependents	Object
5	Education	Object
6	Self_Employed	Object
7	ApplicantIncome	Int
8	CoapplicantIncome	Float
9	LoanAmount	Float
10	Loan_Amount_Term	Float
11	Credit_History	Float
12	Property_Area	Object
13	Loan_Status	Object

DATA PREPROCESSING:

The real-world data is subjected to noise and null values. So we need to clean the dataset and apply a machine learning model for better results. First, we checked for any duplicates in the dataset and then went for Data Cleaning which includes removing noise and null values.

There are a few methods to handle the null values:

1. Ignoring the tuple
2. Filling by mean
3. Filling by most repeated value
4. Filling by a constant

Here we used mean and mode methods to fill the null values.

Handling Null values

```
126]: df.isnull().sum()
```

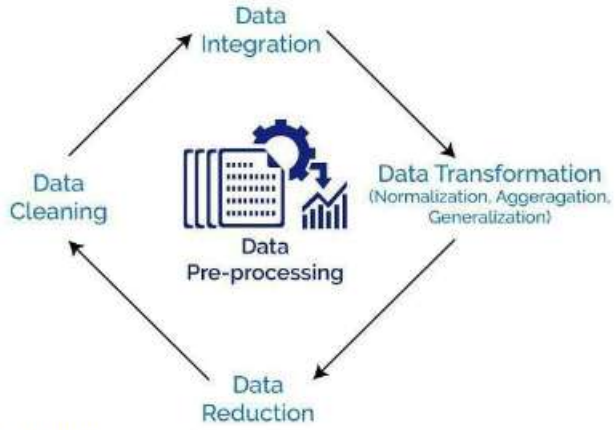
```
Out[126]: Loan_ID              0
          Gender              13
          Married              3
          Dependents          15
          Education            0
          Self_Employed       32
          ApplicantIncome      0
          CoapplicantIncome    0
          LoanAmount          22
          Loan_Amount_Term    14
          Credit_History      50
          Property_Area        0
          Loan_Status          0
          dtype: int64
```

```
127]: df["Gender"].fillna(df["Gender"].mode()[0],inplace=True)
      df["Married"].fillna(df["Married"].mode()[0],inplace=True)
      df["Dependents"].fillna(df["Dependents"].mode()[0],inplace=True)
      df["Self_Employed"].fillna(df["Self_Employed"].mode()[0],inplace=True)
      df["LoanAmount"].fillna(df["LoanAmount"].mean(),inplace=True)

      df["Loan_Amount_Term"].fillna(df["Loan_Amount_Term"].mode()[0],inplace=True)
      df["Credit_History"].fillna(df["Credit_History"].mode()[0],inplace=True)
```

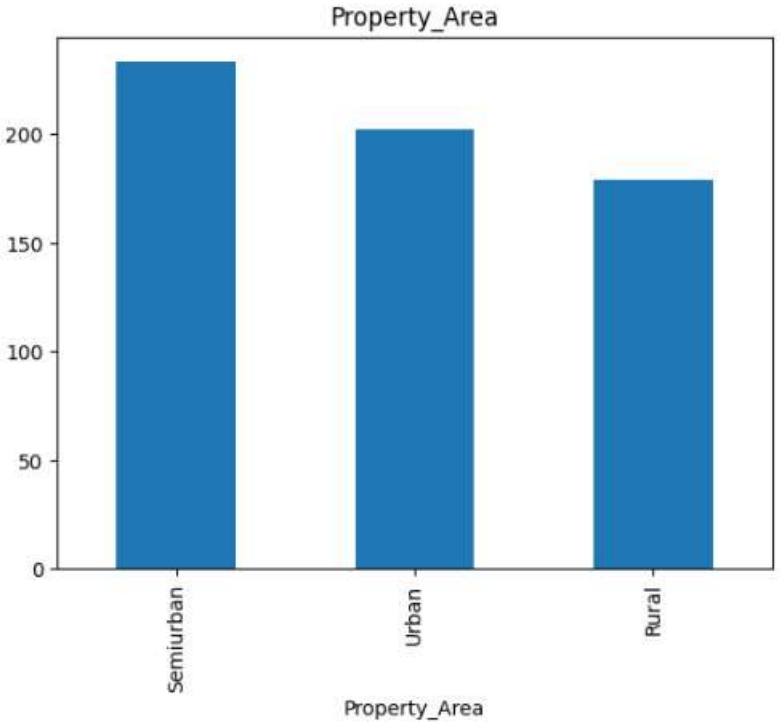
```
128]: df.isnull().sum()
```

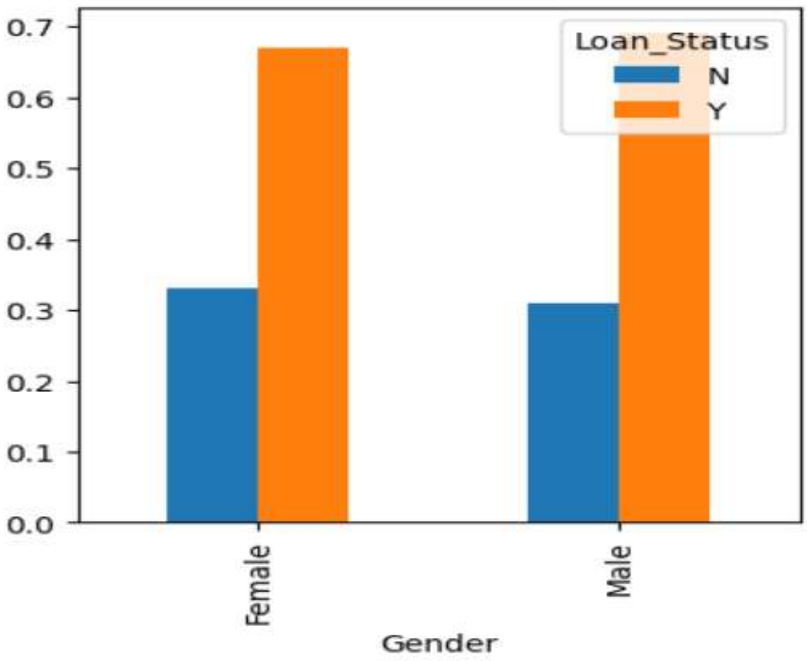
```
Out[128]: Loan_ID              0
          Gender               0
          Married              0
          Dependents           0
          Education            0
```

After dealing with null values, we need to convert categorical columns into numerical data columns. We cannot apply the machine learning model to categorical columns. So we Found the categorical columns from the dataset and used an inbuilt module from sklearn LabelEncoder to convert categorical data into numerical data.



Some Data Visualization:





This above Bar Graph shows the Loan status of Male and Female.

2.4 NORMALIZATION:

After merging data from various sources, the data may also need to transform into forms
appropriate for mining. The data transformation includes the following techniques:

1. Smoothing
2. Aggregation
3. Normalization

1. Min-Max Normalization
2. Z score normalization
3. Decimal scaling

We need to normalize the data so that the accuracy of the model improves. There are
various methods to normalize the data like Min-Max normalization, Z-score normalization, and also NumPy has an inbuilt function to normalize certain columns in the data

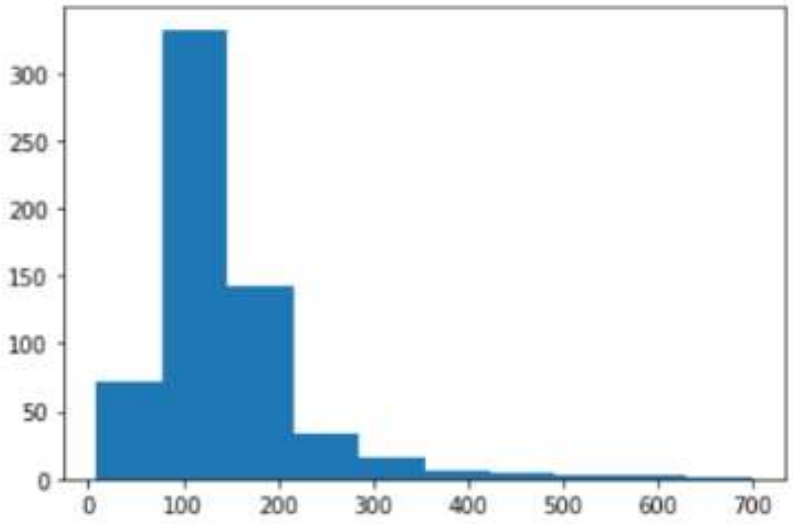
dtype: object

```
from sklearn.preprocessing import StandardScaler   # importing module

Zscaler=StandardScaler()   # creating object for standard scaler

df2=Zscaler.fit_transform(df)
df2=pd.DataFrame(df2,columns=df.columns)
df2.drop("Loan_Status",axis=1,inplace=True)
df2["Loan_status"]=df["Loan_Status"]
df2.head()
```

11]:

	Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome	CoapplicantIncome	LoanAmount	Loan_Amount_Term	Credit_History	Pro
0	0.472343	-1.372089	-0.737806	-0.528362	-0.392601	0.072991	-0.554487	0.250640	0.273231	0.411733	
1	0.472343	0.728816	0.253470	-0.528362	-0.392601	-0.134412	-0.038732	-0.020226	0.273231	0.411733	
2	0.472343	0.728816	-0.737806	-0.528362	2.547117	-0.393747	-0.554487	-1.355252	0.273231	0.411733	
3	0.472343	0.728816	-0.737806	1.892641	-0.392601	-0.462062	0.251980	-0.150299	0.273231	0.411733	
4	0.472343	-1.372089	-0.737806	-0.528362	-0.392601	0.097728	-0.554487	0.174727	0.273231	0.411733	

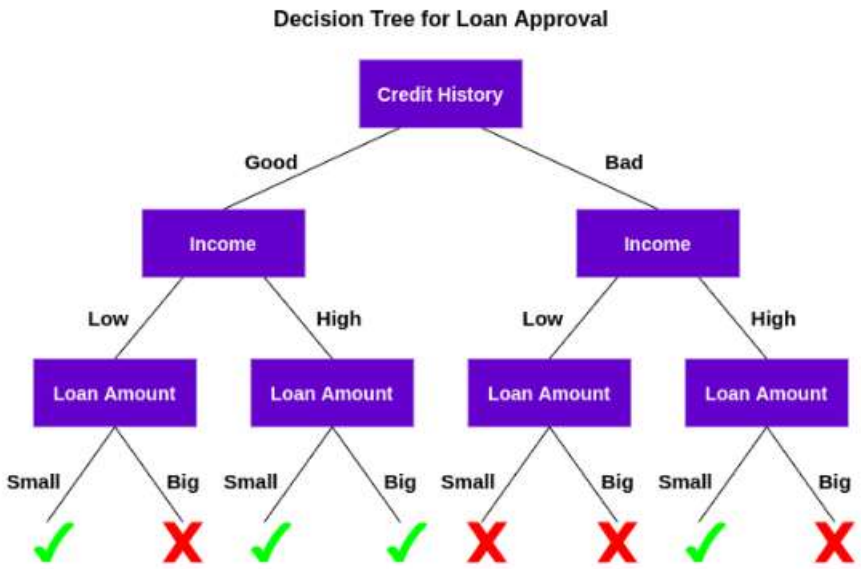
Before Normalization



2.5 Feature Selection:

Machine learning is based on a basic principle: if you put garbage in, rubbish will come out. When I say garbage, I'm referring to data noise. When there are a lot of features, this becomes much more significant. When constructing an algorithm, youdon't have to employ every feature available to you. You may help your algorithm byproviding only the most critical features into it.Feature selection is the process of selecting required or most valued features from the dataset which contribute most to our predicted variable. Feature selection helps to improve accuracy and reduce the running time of the model since we are removing the columns which are less significant to our predicted variable. It reduces the overfittingof the model

3. Discussion
3.1 DECISION TREE:
Algorithm:
It is a greedy algorithm
The tree is constructed in a top-down recursive divide and conquer
method At the start, all the training examples are the root.
Attributes are categorical(if continuous-valued they are discretized in advance)
Examples are partitioned recursively based on selected attributes
Test attributes are selected on the basis of heuristic or statistical measures.
Conditions for stopping partitioning:
A node's samples all belong to the same class.
There are no remaining attributes for further partitioning
There are no samples left.
Advantages:
1. Simple to understand and to interpret. Trees can be visualized.
2. Able to handle multi-output problems.
3. Even if the underlying model from which the data were created violates some
of its assumptions, it still performs well.



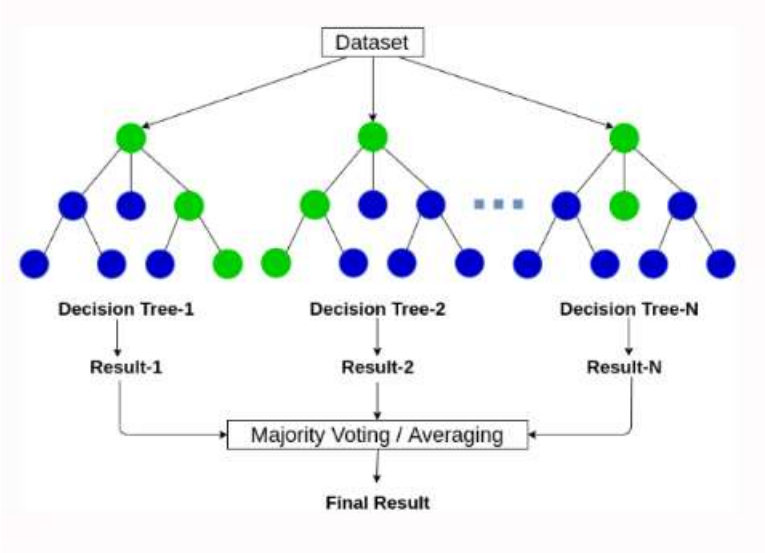
The determination of the characteristic for the root node in each level is a key difficulty in the Decision Tree. This process is known as attribute selection. We have two popular attribute selection measures:
1. Information Gain
2. Gini Index
Given the dataset, we need to find the root node for the tree. To do so, we must first determine the information gain of each column. The root node will be determined by the column with the highest information gain. To compute the information gain, we must first calculate the entropy of each possible outcome in each column, as well as the
overall entropy. Similarly, the creation of the root node continues by picking other nodes in the appropriate columns.

Aggregate Results:

● Combine the predictions from all trees to obtain the final ensemble prediction.

Advantages of Random Forest:

1. Reduces overfitting by combining multiple trees.
2. Handles noisy data well.
3. Provides feature importance information.



4. Concluding Remarks

The model began with the data cleaning and processing followed by implementation. The decision tree model gave 74% accuracy for the dataset. The Random forest model gave accuracy of 74.79% .The Gaussian Naive Bayes model gave accuracy of 82.92% As a result applicants with poor credit history are rejected and applicants with higher income have more chances to get the loan as the chances to get back the loan amount are higher. Gender and marital status play less role in determining loan eligibility.