Loan Approval Prediction

Sai Rohith Gogineni,Ritesh Jadhav,Uday sai Chaganti,Chaitanya Sai Nutakki, Jithin chowdary Atluri, Sumalatha Saleti

***Abstract*— With the enhancement in the banking sector lots of people are applying for bank loans but the bank has its limited assets which it has to grant to limited people only, so finding out to whom the loan can be granted which will be a safer option for the bank is a typical process. So in this project we try to reduce this risk factor behind selecting the safe person so as to save lots of bank efforts and assets. This is done by mining the Big Data of the previous records of the people to whom the loan was granted before and on the basis of these records/experiences the machine was trained using the machine learning model which gives the most accurate result. The main objective of this report is to predict whether assigning the loan to a particular person will be safe or not.**

I. INTRODUCTION

Loan Prediction is very helpful for employees of banks as well as for the applicant also. The aim of this Project is to provide a quick, immediate and easy way to choose the deserving applicants. It can provide special advantages to the bank. The Loan Prediction System can automatically calculate the weight of each feature taking part in loan processing and on new test data the same features are processed with respect to their associated weight .A time limit can be set for the applicant to check whether his/her loan can be sanctioned or not. Loan Prediction System allows jumping to specific applications so that it can be checked on priority basis. This Project is exclusively for the managing authority of a Bank/finance company, the whole process of prediction is done privately no stakeholders would be able to alter the processing. Results against particular Loan Id can be sent to various departments of banks so that they can take appropriate action on application. This helps all other departments to carry out other formalities.

Distribution and approval of loans is one of the core business parts of any operational retail bank. It resembles a major income for the bank's capital for its assets. With the enhancement of the technologies in the banking sector and the daily processes becoming much more streamlined and efficient, this caused a larger influx of applicants for loans seeking capital from banks which readily expands any bank reservoir of data for this type of decision making. The process of selecting an eligible candidate is a typical process. This type of decision making has an inherent risk associated with it. Moreover, selecting and developing the proper resources to make that judgment also possess another operational risk and cost.

The current work aims at automating the process of loan approval by using statistical learning methods based on a bank's historical data that outlines certain characteristics of the applicants.

II. LITERATURE REVIEW

Loan prediction is a much-talked-about subject in the sectors of banking and finance. Credit scoring has become a key tool for the same in this competitive financial world. Furthermore, following the recent improvements in data science and several notable developments in the field of artificial intelligence, this topic has gained more attention and research interest. In recent years, it has attracted more focus towards research on loan prediction and credit risk assessment. Due to the high demands of loans now, demand for further improvements in the models for credit scoring and loan prediction is increasing significantly. A multitude of techniques have been used to assign individuals a credit score and much research has been done over the years on the topic. Unlike previously, where experts were hired and the models depended on professional opinions were used for assessing the individual's creditworthiness, the focus has shifted to an automated way of doing the same job. In recent years, the researchers and banking authorities have been focused on applying machine learning algorithms and neural networks for credit scoring and risk assessment

III. METHODOLOGY

3.1 Dataset Description

Every new applicant details filled at the time of application form acts as a test data set. After the operation of testing, model predict whether the new applicant is a fit case for approval of the loan or not based upon the inference it conclude on the basis of the training data sets.These details are Gender, Marital Status, Education, Number of Dependents, Income, Loan Amount, Credit History and others. To automate this process, they have given a problem to identify the customers segments, those are eligible for loan amount so that they can specifically target these customers. Here they have provided a partial data set.

Variable Name	Description of Variable	Data Type
Dependents	Number of dependents	Integer
Education_Qualification	Graduate/ Undergraduate	String
Self_Imployed	Self Employed (Y/N)	Character
Application_Income	Applicant income	Integer

Variable Name	Description of Variable	Data Type
Dependents	Number of dependents	Integer
Education_Qualification	Graduate/ Undergraduate	String
Co_Applicant_Income	Coapplicant income	Integer
Loan_Amount	Loan amount in Thousands	Integer

3.2 Preprocessing

The data processing techniques covered data transformation and missing data imputation following the approach. There were seemingly extreme values present in both loan amount applied for and the income of certain individuals.

Accordingly, different forms of data transformation have been implemented to normalize the variables.

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

```
dataset=pd.read_csv("C:\\Users\\Dell\\Downloads\Loan approval.csv")
```

```
dataset
```

	Loan_ID	Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome	CoapplicantIncome	LoanAmount	Loan_Amount_Term
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
...	...	...	...	...	...	...	...	...	...	...
609	LP002978	Female	No	0	Graduate	No	2900	0.0	71.0	360.0
610	LP002979	Male	Yes	3+	Graduate	No	4106	0.0	40.0	180.0
611	LP002983	Male	Yes	1	Graduate	No	8072	240.0	253.0	360.0
612	LP002984	Male	Yes	2	Graduate	No	7583	0.0	187.0	360.0
613	LP002990	Female	No	0	Graduate	Yes	4583	0.0	133.0	360.0

614 rows × 13 columns

3.3 Data cleaning

The data can have many irrelevant and missing parts. To handle this part, data cleaning is done. It involves handling of missing data, noisy data etc.

```
dataset.info()     # there are 13 attributes along with the class attribute, here the class attribute is Loan_Status
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 614 entries, 0 to 613
Data columns (total 13 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   Loan_ID            614 non-null    object
 1   Gender             601 non-null    object
 2   Married            611 non-null    object
 3   Dependents         599 non-null    object
 4   Education          614 non-null    object
 5   Self_Employed      582 non-null    object
 6   ApplicantIncome    614 non-null    int64
 7   CoapplicantIncome  614 non-null    float64
 8   LoanAmount         592 non-null    float64
 9   Loan_Amount_Term   600 non-null    float64
 10  Credit_History     564 non-null    float64
 11  Property_Area      614 non-null    object
 12  Loan_Status        614 non-null    object
dtypes: float64(4), int64(1), object(8)
memory usage: 62.5+ KB
```

```
dataset.describe()    # exploring the data
```

	ApplicantIncome	CoapplicantIncome	LoanAmount	Loan_Amount_Term	Credit_History
count	614.000000	614.000000	592.000000	600.00000	564.000000
mean	5403.459283	1621.245798	146.412162	342.00000	0.842199
std	6109.041673	2926.248369	85.587325	65.12041	0.364878
min	150.000000	0.000000	9.000000	12.00000	0.000000
25%	2877.500000	0.000000	100.000000	360.00000	1.000000
50%	3812.500000	1188.500000	128.000000	360.00000	1.000000
75%	5795.000000	2297.250000	168.000000	360.00000	1.000000
max	81000.000000	41667.000000	700.000000	480.00000	1.000000

3.4 Missing data

This situation arises when some data is missing in the data. It can be handled in various ways.

```
plt.figure(figsize=(12, 10))
sns.heatmap(dataset.isnull(),yticklabels=False,cbar=False,cmap='plasma')
# using seaborn library and heatmap() function, we are able to visualise the missing values in the data set.
dataset.isnull().sum()
```

```
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
```

```
# filling columns containing null values with mean values of respective columns
mean_crehis = dataset['Credit_History'].mean()
mean_LoanAmo = dataset['LoanAmount'].mean()
mean_Loan_AmoTerm = dataset['Loan_Amount_Term'].mean()

dataset['Credit_History'].fillna(value = mean_crehis,inplace = True)
dataset['LoanAmount'].fillna(value = mean_LoanAmo,inplace = True)
dataset['Loan_Amount_Term'].fillna(value = mean_Loan_AmoTerm,inplace = True)

# dropping the missing values
dataset = dataset.dropna()
```

```
plt.figure(figsize=(12, 10))
sns.heatmap(dataset.isnull(),yticklabels=False,cbar=False,cmap='plasma')
# using seaborn library and heatmap() function, we are able to visualise the missing values in the data set.
dataset.isnull().sum()
```

```
Loan_ID              0
Gender               0
Married              0
Dependents           0
Education            0
Self_Employed        0
ApplicantIncome      0
CoapplicantIncome    0
LoanAmount           0
Loan_Amount_Term     0
Credit_History       0
Property_Area        0
Loan_Status          0
dtype: int64
```

3.5 Handle Categorical values

```
dataset.replace({"Loan_Status":{'N':0,'Y':1}},inplace=True)
```

```
dataset.head(5)
```

	Loan_ID	Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome	CoapplicantIncome	LoanAmount	Loan_Amount_Term
0	LP001002	Male	No	0	Graduate	No	5849	0.0	146.412162	360.0
1	LP001003	Male	Yes	1	Graduate	No	4583	1508.0	128.000000	360.0
2	LP001005	Male	Yes	0	Graduate	Yes	3000	0.0	66.000000	360.0
3	LP001006	Male	Yes	0	Not Graduate	No	2583	2358.0	120.000000	360.0
4	LP001008	Male	No	0	Graduate	No	6000	0.0	141.000000	360.0

```
dataset['Dependents'].value_counts()
```

```
0     319
1      95
2      94
3+     46
Name: Dependents, dtype: int64
```

```
dataset=dataset.replace(to_replace='3+',value=4)
```

```
dataset
```

	Loan_ID	Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome	CoapplicantIncome	LoanAmount	Loan_Amount_Term
0	LP001002	Male	No	0	Graduate	No	5849	0.0	146.412162	360.0
1	LP001003	Male	Yes	1	Graduate	No	4583	1508.0	128.000000	360.0
2	LP001005	Male	Yes	0	Graduate	Yes	3000	0.0	66.000000	360.0
3	LP001006	Male	Yes	0	Not Graduate	No	2583	2358.0	120.000000	360.0
4	LP001008	Male	No	0	Graduate	No	6000	0.0	141.000000	360.0
...	...	...	...	...	...	...	...	...	...	...
609	LP002978	Female	No	0	Graduate	No	2900	0.0	71.000000	360.0
610	LP002979	Male	Yes	4	Graduate	No	4106	0.0	40.000000	180.0
611	LP002983	Male	Yes	1	Graduate	No	8072	240.0	253.000000	360.0
612	LP002984	Male	Yes	2	Graduate	No	7583	0.0	187.000000	360.0
613	LP002990	Female	No	0	Graduate	Yes	4583	0.0	133.000000	360.0

554 rows × 13 columns

3.6 Decision Tree Implementation

In this project we have implemented the “Decision Tree” classification. A decision tree is a non-parametric supervised learning algorithm, which is utilised for both classification and regression tasks. It has a hierarchical tree structure, which consists of a root node, branches, internal nodes and leaf nodes.It is the most important and powerful tool for classification and prediction. A Decision tree is a flowchart-like tree structure, where each internal node denotes a test on an attribute, each branch represents an outcome of the test, and each leaf node (terminal node) holds a class label. The basic algorithm of decision tree [7] requires that all attributes or features should be discretised. Feature selection is based on greatest information gain of features. The knowledge depicted in the decision tree can represented in the form of IF-THEN rules.

```
from sklearn.model_selection import train_test_split

X=dataset.drop(columns=['Loan_ID','Loan_Status'],axis=1)
Y=dataset["Loan_Status"]

X_train,X_test,Y_train,Y_test=train_test_split(X,Y,test_size=0.3,random_state=101)
#splitting the data set for testing and training

from sklearn.tree import DecisionTreeClassifier

model=DecisionTreeClassifier(max_depth=3,random_state=0,criterion="entropy")

model.fit(X_train,Y_train)# training the model with training data

DecisionTreeClassifier(criterion='entropy', max_depth=3, random_state=0)

predictions=model.predict(X_test)# testing the algorithm using test data

compare=pd.DataFrame({"Actual":Y_test,"Predicted":predictions})
compare
```

	Actual	Predicted
471	0	0
20	0	0
403	1	1
476	1	1
473	1	1
...	...	...
279	1	1
76	0	1
451	1	1
599	1	1
253	1	1

167 rows × 2 columns

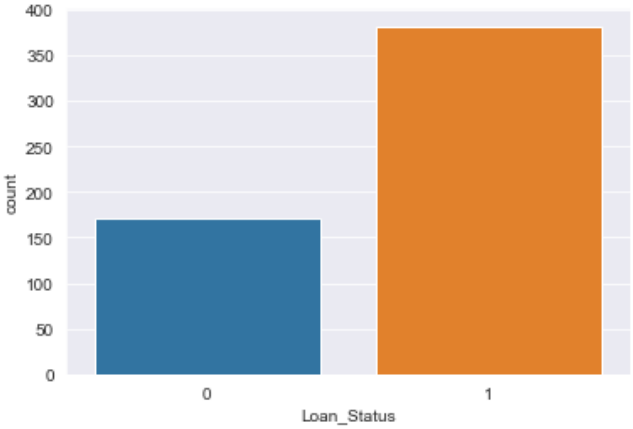
```
from sklearn.metrics import classification_report

print(classification_report(Y_test,predictions))# we can observe that the accuracy is 83%
```

```
              precision    recall  f1-score   support

           0       0.89      0.56      0.68        45
           1       0.86      0.98      0.91       122

    accuracy                           0.86       167
   macro avg       0.87      0.77      0.80       167
weighted avg       0.87      0.86      0.85       167
```

```
from sklearn import tree

# Visualising the algorithm using tree structure. here, we have taken the depth as 3.
fig, axes = plt.subplots(nrows = 1,ncols = 1,figsize = (25,20), dpi=300)
tree.plot_tree(model,feature_names = list(dataset.drop(columns=["Loan_Status"],axis=1).columns),class_names=['0','1'],filled = True,rounded=True);
#fig.savefig('imagename.png')# we can do this to save the figure.
```

IV. Resultant Graphs

```
sns.set_style('darkgrid')
sns.countplot(x='Loan_Status',data=dataset)
# many people got loans
```



V. Conclusion

From a proper analysis of positive points and constraints on the component, it can be safely concluded that the product is a highly efficient component. This application is working properly and meeting all Banker requirements. This component can be easily plugged in many other systems. There have been numbers cases of computer glitches, errors in content and most important weight of features is fixed in automated prediction system, So in the near future the so –called software could be made more secure, reliable and dynamic weight adjustment .In near future this module of prediction can be integrate with the module of automated processing system. The system is trained on old training dataset so future software can be made such that new testing data should also take part in training data after some fixed time.

References

1. Madaan, Mehul, et al. "Loan default prediction using decision trees and random forest: A comparative study." *IOP Conference Series: Materials Science and Engineering*. Vol. 1022. No. 1. IOP Publishing, 2021.
2. S.S. Keerthi., E.G. Gilbert., Convergence of a generalize SMO algorithm for SVM classifier design, Machine Learning, Springer, Vol. 4, Issue 1, pp. 351-360, 2002.
3.
4. Andy Liaw., Matthew Wiener., Classification and Regression by random Forest, Vol. 2, Issue 3, pp. 9-22, 2002.
5.
6. Ekta Gandotra., Divya Bansal., Sanjeev Sofat., Malware Analysis and Classification: A Survey, Journal of Information Security, Vol. 05, Issue 02, pp. 56-64, 2014.
7.
8. Rattle data mining tool, http:// rattle.togaware.com/rattle -download.html.