Predicting the loan status, of a customer based on the customers information

By - Sayan Das

APRIL, 2023

Contents

1 Introduction:

Borrowers frequently need loan approval in order to fund large purchases or investments, like a home or school, in the case of any loan application. Knowing whether a loan will be approved in advance enables borrowers to plan for their financial future with confidence and to avoid the disappointment of being turned down. In this project, we have a dataset of customer information together with a label indicating whether or not the customer's loan has been accepted. We will develop a machine learning model based on this dataset that will forecast whether a customer's loan will be authorised or not based on the information provided by the consumer.

2 Background:

Dream Housing Finance company deals in all home loans. They have presence across all urban, semi urban and rural areas. Customer first apply for home loan after that company validates the customer eligibility for loan. They have a dataset on the customer's information, that is relevant to the loan, along with the label, whether the loan of the customer was approved or not. The dataset is available in the following link –

https://www.kaggle.com/datasets/burak3ergun/loan-data-set

Company wants to automate the loan eligibility process (real time) based on customer detail provided while filling online application form. These details are Gender, Marital Status, Education, Number of Dependents, Income, Loan Amount, Credit History and others.

3 Dataset:

The dataset consists of data on 614 customers, with their details that are relevant to the loan. These details for each of the customers are given by 13 columns. The brief description of the columns are given below –

1. Loan_ID: Unique identity assigned by the bank to each of the customers.
2. Gender: The gender of the customer, unique values are 'Male' and 'Female'.
3. Married: Indicates whether the customer is married or not, unique values are 'Yes' and 'No'.
4. Dependents: The no of dependents of the customer.
5. Education: Indicates whether the customer is Graduated or not, unique values are 'Graduated' and 'Not Graduated'.
6. Self_Employed: Indicates whether the customer is self employed or not, unique values are 'Yes' and 'No'.
7. ApplicantIncome: Income of the loan applicant.
8. CoApplicantIncome: Income of the Co-applicant.
9. LoanAmount: The amount of loan that is requested.
10. Loan_Amount_Term:
11. Credit_History:

12. Property_Area: The property area of the customer, unique values are 'Urban', 'Semi-Urban' and 'Rural'.
13. Loan_Status: Indicator whether the loan of the customer is approved or not, unique values are 'Y' and 'N'.

Here the response variable is the Loan_Status, which indicates whether the loan request of the customer has approved or not. We are to predict the response on basis of the other variables provided. Here the response variables is a categorical type variable, the problem of prediction is going to be a classification problem.

4 The Dataset Exploration:

In the first step, we have deleted the column, Loan_ID because this unique identification column will be of no use to us. After deleting this column, we have explored the dataset to find out the following –

- We have 12 column, which can be classified as the following –
- 4 Numerical Columns: ApplicantIncome, CoApplicantIncome, LoanAmount, Loan_Amount_Term
- 8 Categorical Columns: Gender, Married, Dependents. Education, Self_Employed, Credit_History, Property_Area, Loan_Status,
- The columns Gender, Married, Dependents, Self_employed, LoanAmount, Loan_Amount_Term, Credit_History. The percentage of missing values in each column is shown in the below table –

Table 1 Table Showing the Column-wise % of Missing Values

Column Name	Type	% missing observation
Gender	Categorical	2.12
Married	Categorical	0.49
Dependents	Categorical	2.44
Self_Employed	Categorical	5.21
LoanAmount	Numerical	3.58
Loan_Amount_Term	Numerical	2.28
Credit_History	Categorical	8.14

- In the dataset, there are 422 cases where the response variable Loan_Status is labelled as 'Y' and 192 cases where the response variable Loan_Status is labelled as 'N'. Therefore, there is an imbalance in the response variable.
- We have obtained the bar-plots that represents the frequencies for the each of the categories for all the categorical variables, under the cases where the loan request have been approved and under the cases where the loan request have been rejected. The bar-plots are shown below.

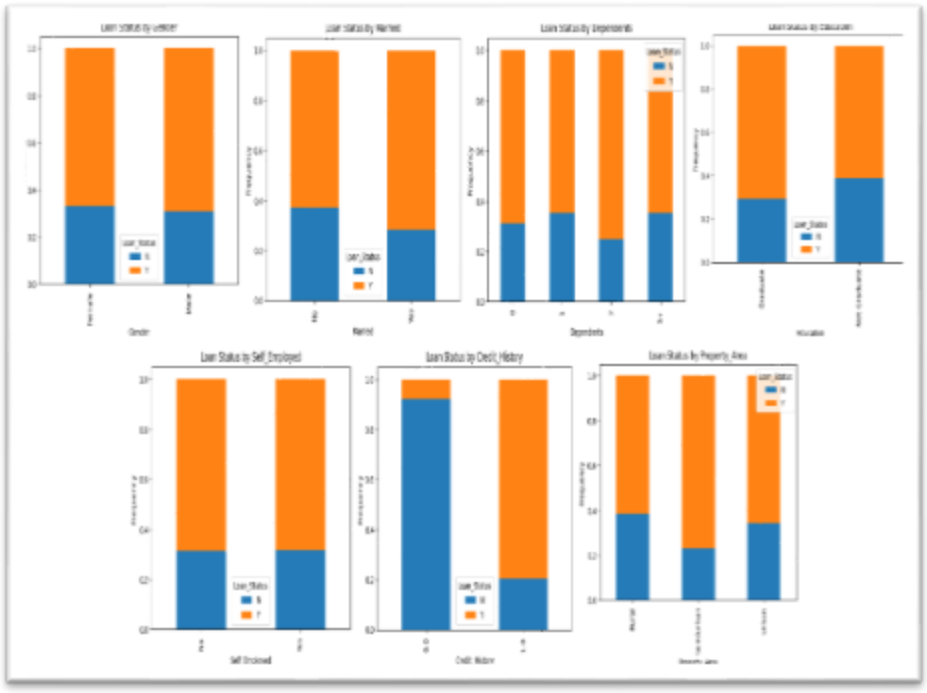

Figure 1 Bar-plots of the Categorical Columns

- From the bar-plot it seems that the variables like Credit History or Dependents strongly determines the decision that whether the loan request will be Processed or not. In addition, variables like property area or gender or marital status has contribution in decision whether the loan request will be processed or not.
- We have also plotted the box-plot of each of the numerical variables, for two labels of the Loan_Status. The graphs are shown below –

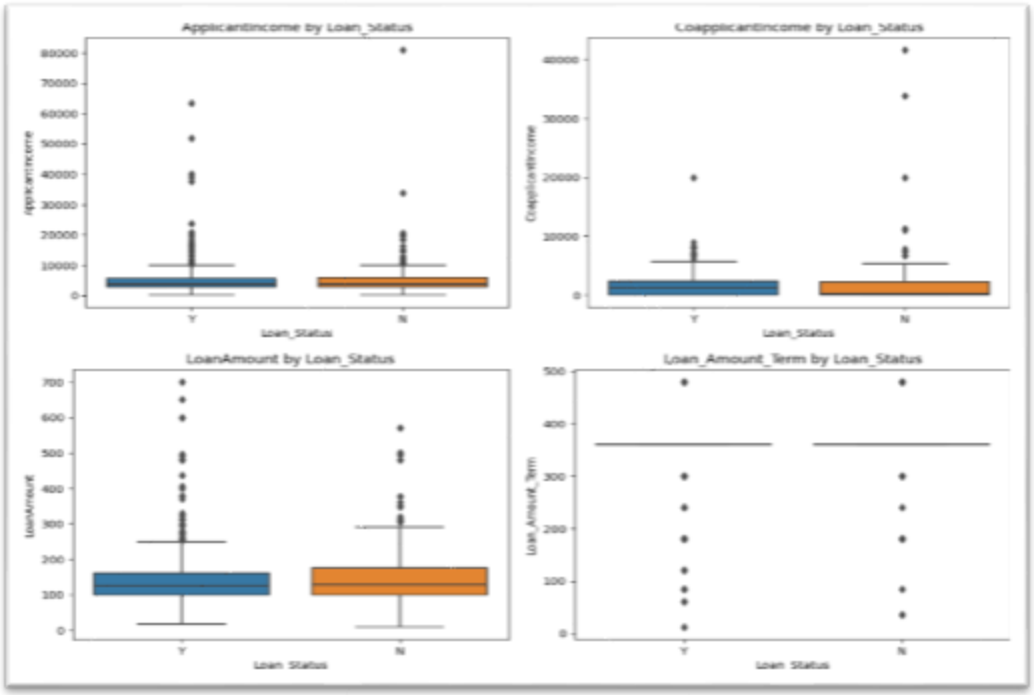

Figure 2 Box-plots of the Numerical Columns

- From the box-plot of the numerical variables it seems that the all the variables of the numerical have contributions in determining whether the loan of the customer will be approved or not.
- The bar-plot showing the variability of the numerical columns is given below.

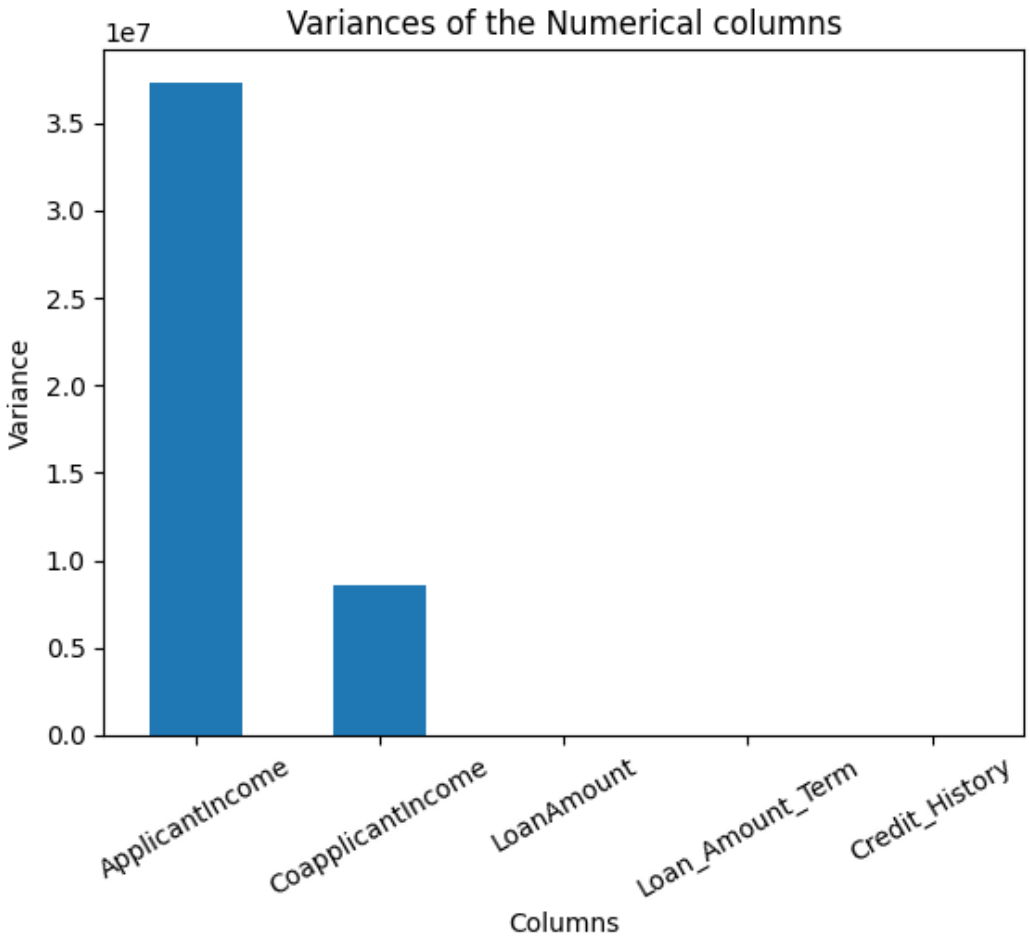

Figure 3 Bar-plots showing the variances of the numerical columns

- The plot suggests that different numeric variables in the dataset has different levels of variability.

5 Pre-processing:

In this stage we are going to pre-process the dataset, so that it could be used for the modelling purpose.

5.1 Treating the Missing Values:

Observe that both the categorical type and numerical type of variables have missing values. Since the number of observations are so less, only 614 rows, we will be going to impute this missing observations.

For the categorical variable, we have imputed the missing observation based on the mode imputation technique, depending on the response. Let us review the method with example of Gender column.

To impute the missing values in the gender column, we have prepared a cross table of Gender and Loan_Status as following

Table 2 Cross table for Gender and Loan Status

		Loan_Status	
		Y	N
Gender	Male	347	155
	Female	75	37

Now observe that in the cross-classification table we can see that among the customers whose loan request has been approved the number of Males are high. So we replace the missing values that has as labelled as 'Y' in the Loan_Status, by 'Male'. In addition, we have seen from the table, the customers whose loan requests has not been approved, most are Males. Hence we replace the missing values that has as labelled as 'N' in the Loan_Status, by 'Male'. We have used the same, missing value imputation method for the other categorical columns.

To imputed the missing values of the numerical column, we used the median imputation techniques since we have seen that there are presence of outliers in the numerical values. Lets understand this imputation with the help of the LoanAmount column.

To impute the missing observations from the LoanAmount column we first compute the median of the LoanAmount column for both the customers whose loan request has been approved and whose loan request has not been approved. The following table shows the computed medians.

Table 3 Table showing the median of the Loan Amount for different Loan Status

		Median LoanAmount
Loan_Status	Y	126.0
	N	129.0

Now, we are going to replace missing values in the LoanAmount column that are labelled as 'Y' in the Loan_Status by 126.0 and missing values in the LoanAmount column that are labelled as 'N' in the Loan_Status by 129.0. We have used this technique to impute the missing values of all the numerical columns.

5.2 Scaling down the observations:

The Figure 3 suggests that the variances of the numerical variables are far apart from each other. Therefore, it would be helpful to scale down the numerical columns. To scale down the numerical columns we have used standard scaling or z-scaling. If the original variable is given by x and the scaled quantity is given by z then -

$$z = \frac{x - \bar{x}}{sd(x)}$$

We have applied this scaling procedure to all the numerical variables that are present in the dataset.

5.3 Encoding the Categorical Variables:

We have already seen that there are 8 categorical variables in the dataset. The values of this categorical variables are not of numeric. This non-numeric type of values would be difficult to handle when we apply any kind of modelling on the dataset. So this categorical variables need to be encoded before we apply any kind of model on the dataset . We have used the 1-Hot-Encoding technique to convert the data categorical columns into the numerical variables. Let us understand this technique with the help of the Property_Area column.

Consider the column Property_Area. It has 3 unique values: rural, semi-urban and urban. Under 1-Hot-Encoding scheme, we will replace the Property_Area column by two new columns Property_Area_rural and Property_Area_semi_urban. If the original value of Property_Area is "rural" then (Property_Area_rural, Property_Area_semi_urban) would (1, 0). If the original value of Property_Area is "semi-urban" then (Property_Area_rural, Property_Area_semi_urban) would (0, 1) and finally if the original value of Property_Area is "urban" then (Property_Area_rural, Property_Area_semi_urban) would (0, 0).

We have applied this encoding scheme to all the categorical variables that are present in the dataset.

6 Model Building:

In this stage, we are going to build up a proper model to predict whether the any loan request with certain specification will be approved or not. For this purpose of classification we have done some experiments, by trying on different models, like logistic regression with L_1 regularization, logistic regression with first 8 principal components, support vector machines. Among these models the prediction accuracy, coming out to be greatest for support vector machine model with a Radial Basis Function(rbf) kernel function.

6.1 Train-Test Split:

To apply those models, in the first stage we have divided the dataset, into training and testing dataset using a 70-30 split. We have also stratified according to the Loan_Status when we make this split. As the dataset was imbalanced with respect to the Loan_Status, stratification will make sure that sufficient amount of observation from the both types of Loan_Status values come into the sample.

6.2 Fitting Model:

In the next step, we have fitted a Logistic Regression model with L_1 regularisation, a Logistic Regression model with 8 principal components (explaining 80.34% of the variation), Support Vector machine model, with RBF as the kernel function on the training dataset. For comparison of the 3 model, we have obtained the Receiver Operating Characteristics (ROC) curve. The curves are shown below –

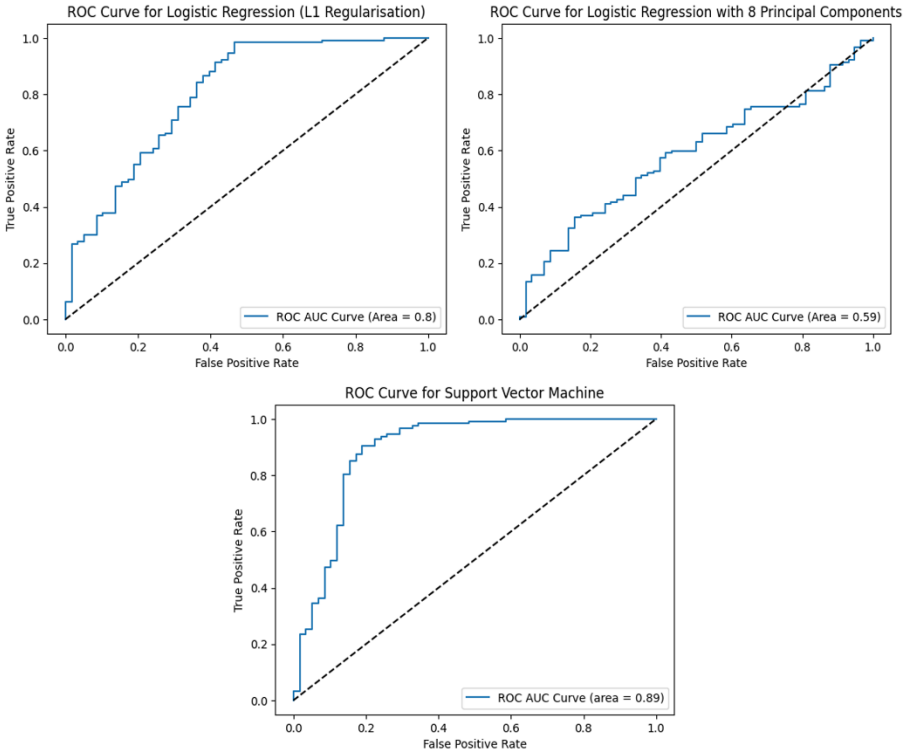

Figure 4 Figure showing the ROC curves of Different models applied

From the ROC curves it is clear that the Area Under Curve is greatest for the Support Vector Machine model (about 0.89). So we choose this model as our final model.

6.3 Selection of the Best Threshold Probability:

The classification model would provide us with the probabilities of the positive response ($P(Y = 1 | x)$ here in this case it is the probability of a bank loan getting accepted). We need to select a threshold such that whenever this probability is greater than that threshold we classify the observation to be a as rejected loan request. For this we have calculated the TPR (True Positive Rate: here Probability of the model, correctly classifying a loan request to be accepted) and FPR

(False Positive Rate: here Probability of the model, wrongly classifying a loan request to be accepted) for a number of threshold then we calculated TPR.(1-FPR) for each threshold. Finally we selected that threshold that maximises TPR.(1-FPR). For out dataset the threshold is selected to be 0.80. i.e, whenever the predicted probability is greater than 0.80 then we classify it as the loan request that it will get accepted. We have shown the model accuracy based on this threshold value.

6.4 Model Accuracy:

Based on the threshold selected (mentioned above) we have predicted the value of the response for the testing dataset, using the support vector machine model trained on the training dataset.

The confusion matrix of the prediction is shown below –

Table 4 Confusion Matrix of the Prediction

		True Value	
		1	0
Predicted Value	1	125	2
	0	26	32

From the confusion matrix we compute the testing accuracy rate = (125+32)/(125+32+2+26) = 0.85. Which is very high. We have also shown the precision and recall and the f1 score in the following table –

Table 5 Table for showing the accuracy score

	Scores	Precision	Recall	F_1 score
Classification	0	0.94	0.55	0.70
	1	0.83	0.98	0.90

The overall values of the accuracy scores ate very high. Judging these high values we decide that the model fitted good for the dataset.

7 Using the Model to build a Web-application:

Once we are done with preparing the model, we have used it to build a simple web app, using streamlit library in python, that takes Gender, Marital Status, Income, Loan Amount etc. as input and it predicts the output regarding whether the loan request with specification as inserted as input, will be approved or not.

8 Conclusion:

In conclusion, the machine learning project that predicts loan application approvals based on customer and loan details has successfully demonstrated the application of various techniques such as data cleaning, missing value handling, scaling and encoding to pre-process the data. The project has also involved model comparison and accuracy determination, which has helped to select the best performing model.

Overall, this project has shown the potential of machine learning techniques to provide insights and predictions in the field of financial services, particularly in the loan approval process. The project's results can be used to inform the decision-making process and improve the efficiency of the loan approval process. Further work can be done to improve the accuracy and performance of the model, as well as to integrate it into a real-world application.

9 References:

1. The dataset is available in kaggle website, the link to the dataset : https://www.kaggle.com/datasets/burak3ergun/loan-data-set
2. All the data explorations, pre-processing, calculations and model building are done using python codes. The codes are available in the form of Jupyter Notebook and python file in the following link : https://github.com/sayandas1302/Loan-Approval-Classifier-Project