LOAN APPROVAL PREDICTION

FINAL PROJECT REPORT

Submitted by:

1). Piyush Kulkarni - psk5@njit.edu

2). Payal Rane - pdr23@njit.edu

3). Sakshee Parnerkar - sp2772@njit.edu

Table of Contents

1) Introduction

Nowadays Loans have become the main source of business for banks. The amount of interest rate that is applicable while taking a loan adds up to be the main profit for the banks while granting the loan. There is an intensive procedure of verification, validation and approval followed by the loan companies while granting a loan. Even after these rigorous procedures the companies cannot assure whether the applicants will be able to repay it with no difficulties.

In this project, a predictive model is built to predict if an applicant can repay the lending company or not. The data is prepared using Jupyter Notebook and with the use of various models to predict the target variable - Loan Status.

2) Importing the Libraries

We use python as the programming language and the libraries we import for this project are as follows.

```
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns
```

3) Loading the datasets

We have taken the dataset for Loan approval prediction from the link below.

Training:
https://raw.githubusercontent.com/mridulrb/Predict-loan-eligibility-using-IBM-Watson-Studio/master/Dataset/train_ctrUa4K.csv

Testing:
https://raw.githubusercontent.com/mridulrb/Predict-loan-eligibility-using-IBM-Watson-Studio/master/Dataset/test_lAUu6dG.csv

Submission:
https://raw.githubusercontent.com/mridulrb/Predict-loan-eligibility-using-IBM-Watson-Studio/master/Dataset/sample_submission_49d68Cx.csv

4) Exploratory Data Analysis (EDA)

The different types of variables are Categorical, Ordinal, and Numerical.

Categorical variables: Gender, Married, Self_Employed, Credit_History, Loan_Status.
Ordinal variables: Dependents, Education, Property_Area.
Numerical variables: ApplicantIncome, Co-applicantIncome, LoanAmount, Loan_Amount_Term.

A) Univariate Analysis:

1) Categorical variables:
   a) The loan of 69% of people has been approved.
   b) Almost 80% (79.43%) of the applicants are males.
   c) Around 64.97% of the applicants are married.
   d) Here 80% of the applicants are not self-employed.

2) Ordinal variables:
   a) Most of the applicants i.e. 56.62% do not have any dependents.
   b) Almost 78.21% are graduate applicants while 21.79% are not graduated.
   c) The Semi-urban area has most of the applicants up to 40.33% whereas people who reside in Urban or Rural areas is comparatively less.

3) Numerical variables:
   a) The distribution of applicant income is not normally distributed as it is inclined more towards the left. According to the box plot, we infer that there are many outliers. As we are looking at people from different educational backgrounds, we segregate the data according to their education as well as their income and get to know that people who have graduated have high income.
   b) Co-applicant's income has a similar distribution as that of applicant's income.

B) Bivariate Analysis:

1) The loan approval and rejection ratio of male and females is almost the same.
2) For married people, the loan approval rate is comparatively higher than people who are married.
3) The approval rate of people with 2 dependents is high while people with 0, 1 or 3+ dependents are similar but less than 2 dependents.
4) People who graduate have a high rate of approved loans.
5) Self-employment or not is not a factor for approval of loans.
6) People with 0 credit history have high rejection rates. Whereas the ones with 1 credit history have high approval rates.
7) The ones who reside in Semi Urban areas have high chances of their loan being approved.
8) We considered the mean of applicant income of people with approved loans vs the mean of applicant income of people with rejected loans. There is no difference in both the cases. So, we made bins for the applicant income variable based on the values in that and we infer that there is still not much difference in loan status.
9) We then created bins for co-applicant income, and it states that people with low co-applicant income have high approval rates.
10) Later, we created a new variable as total income where we add applicant income as well as co-applicant income. We concluded that people with a high total income have high approval rates while those

with low total income have low approval rates. We created bins for total income as well and then got to know that people with low and average total income have higher loan approvals and those with higher total income have comparatively lesser loan approvals.

For exploratory data analysis, we created bins by changing 3+ dependents to 3 dependents to make it a numerical variable. Now we drop these bins and convert our target variable i.e. Loan_Status to binary to find correlation with numerical variables. We replace N with 0 and Y with 1. We use heatmap to visualize the correlation through variations in coloring. The most correlated variables are ApplicantIncome — LoanAmount and Credit_History — Loan_Status.

5) Replacing Missing Data

There are certain values missing in the dataset. To fill in the missing values we use mean or median for numerical variables and mode for categorical variables.

Outliers usually have a significant effect on mean and standard deviation which affects the distribution. Hence, we remove the outliers from our dataset. To remove skewness, we take log transformation resulting in a normal distribution.

6) Model Building: Part 1

1) Logistic Regression:

Logistic Regression is a classification algorithm used to predict a binary outcome (1 / 0, Yes / No, True / False). The estimation of the Logit function is known as logistic regression. The logit function is essentially a log of the event's chances in favor. This function produces an S-shaped curve with the probability estimate, which is quite close to the stepwise function that is required.

We can remove the Loan ID variable because it has no bearing on the loan status. The test dataset will be modified in the same way as the training dataset.

For creating several models, we will utilize scikit-learn (sklearn), a Python open-source package. It is one of the most efficient tools, with many built-in functions that may be used for Python modeling.

Sklearn requires a distinct dataset for the target variable. As a result, we'll remove our target variable from the training dataset and save it in a different one.

For the categorical variables, we'll now create dummy variables. The dummy variable converts categorical variables into a series of 0 and 1 values, which makes them much easier to measure and compare.

Now we'll use the training dataset to train the model and make predictions for the test dataset. We do this by splitting our train dataset into two sections: train and validation. We can use this training phase to train the model and then use that to make predictions for the validation part. We can validate our predictions in this manner since we have the real predictions for the validation part which we do not have for the test dataset.

As a result, our predictions are 80% accurate, meaning we accurately recognized 80 percent of the loan status.

2) Logistic Regression using stratified k-folds cross-validation:

Stratification is the process of organizing data so that each fold is a decent representation of the entire dataset. For example, in a binary classification task where each class has 50% of the data, it is ideal to fold the data so that each class contains around half of the instances in each fold. When dealing with both bias and variance, it is often a preferable technique. In circumstances when there is a significant class imbalance, a randomly picked fold may not appropriately represent the minor class. This model's mean validation accuracy comes out to be 0.80. Therefore, we visualize the ROC curve. We get an AUC value of 0.7346.

7) Feature engineering

We came up with additional features based on our analysis that may affect the target variable. These are the following three new features:

a) Total Income:

We will combine the Applicant Income and Co-applicant Income, as discussed during bivariate analysis. If your total income is large, your chances of getting a loan are likely to be high as well.

b) EMI:

The applicant's monthly payment to repay the loan is referred to as the EMI. People with large EMIs may find it difficult to repay the loan, which is why this variable was created. We have computed the EMI by multiplying the loan amount by the loan duration.

c) Balance Income:

After the EMI has been paid, the remaining money is referred to as balance income. The theory behind this variable is that if it has a high value, it means that a person is more likely to repay the loan, improving the odds of loan acceptance.

We remove the variables (ApplicantIncome, CoapplicantIncome, LoanAmount, Loan_Amount_Term) we used to make these new features. This is because the correlation between the old and new features will be quite high, whereas logistic regression implies that the variables are not highly associated. We also want to get rid of the noise in the data, so deleting linked features will help with that.

8) Model Building: Part 2

We continue the model-building process after adding new features. So we'll start with logistic regression and work our way up to more complicated models like Random Forest and XGBoost.

1) Logistic Regression:

Using Logistic Regression, we get the mean validation accuracy as 0.7214.

2) Decision Tree:

A decision tree is a supervised learning algorithm that is commonly used to solve classification problems. Based on the most significant splitter/differentiator in input variables, we divide the population or sample into two or more homogenous groups.

To decide whether to break a node into two or more sub-nodes, decision trees employ a variety of techniques. The homogeneity of the generated sub-nodes improves with the generation of sub-nodes. To put it another way, the purity of the node improves as the target variable grows. We get the mean validation accuracy as 0.71

3) Random Forest:

Random Forest is a tree-based bootstrapping approach that combines several weak learners to produce a powerful prediction model. A decision tree model is built for each individual learner using a random sample of rows and a few randomly chosen factors. The final prediction can be a function of all the individual learners' guesses. The final prediction in a regression issue can be the mean of all the predictions. We get the mean validation accuracy as 0.8044

4) Grid Search:

To get the best values for hyper parameters, we use a grid search. Grid-search is a method for choosing the best hyper parameter from a set of hyper parameters that are parametrized by a grid of parameters. The max_depth and n_estimators parameters will be adjusted. The

maximum depth of the tree is determined by max depth, and the number of trees employed in the random forest model is determined by n estimators. Here, we get the mean validation accuracy as 0.80.

We now determine the important features to determine which features are the most important for this problem. To do so, we use the sklearn feature importances_ property. The most essential aspect is Credit History, which is followed by Balance Income, Total Income, and EMI. As a result of feature engineering, we were able to predict our target variable.

5) XGBoost:

We have previously replaced the categorical variables with numeric ones because XGBoost only works with numeric variables.

Here we have a few new parameters namely,

a) n_estimator: Number of trees in the model.
b) max_depth: To select the maximum depth of a tree.

Using XGBoost, we get the mean validation accuracy as 0.7882.

9) Conclusion

We performed exploratory data analysis on the dataset's features to see how they are distributed. We used charts to perform bivariate and multivariate analysis to understand how their features impacted one another. We looked at each variable to see if the data was clean and evenly distributed. The data was cleansed, and missing values were deleted. Based on our analysis, we added additional features that might affect the target variable. And we inferred whether there is a link based on analysis. We then continued our model building process with these new features. We get accuracy of 73.46% using model "Logistic Regression using stratified k-folds cross-validation". Thus, using Loan Approval Prediction, we can help banks to not go in debt.