Laura Thornton, Zelin Wang, Vijay Kalavakolanu
CSC 2302
April 27, 2020

Automated Loan Eligibility Project Report

Problem Statement

Given sets of test and training data, our goal was to automate the process of approving a loan for individuals based on information they volunteer via an application. This process could be utilized on behalf of banks or financing companies to optimize their loan deals.

Preprocessing of input

The data provided has a number of columns characterizing each entry of data. The variables we had to work with were the applicant's unique loan ID number, gender, marital status, number of dependents, education, whether or not they were self employed, their income, their co-applicant's income, the loan amount, the loan term, their credit history, the property area, and for the training data we had the applicant's loan status.

The first step was to organize and familiarize ourselves with the data. We sorted the variables between categorical variables and numerical variables. The categorical variables included Gender, Married, Dependents, Education, Self_Employed, Loan_Amount_Term, Credit_History, Property_Area, and Loan_Status since the values associated with these columns were not int values. On the other hand we identified our numerical variables as ApplicantIncome, CoapplicantIncome, and LoanAmount.

Our next step was to look for trends in the data between certain variables and the target variables (Loan_Status). We used pivot tables in the pandas library to generate probability charts for each categorical variable in relation to Loan_Status. We saw the most impactful variables to be Credit_History, LoanAmount, and Education. Using the matplotlib library to create histograms of the continuous variables we looked for outliers. The histograms showed us a skewed distribution and were somewhat vague so we used boxplots on the numerical variables to confirm the presence of outliers in our ApplicantIncome and LoanAmount data. To counteract this and normalize our continuous variables, we redefined our continuous variables using the logarithmic function and looked at the distribution of the log histograms to see it was normal. For the categorical variables, most of the columns were missing data points. In order to normalize the rest of the data we filled in the missing values with the mode for each categorical variable and the mean of the numerical variables.

In order to actually use this data in our evaluation we had to use the Label Encoder in the sklearn library to convert categorical variables to continuous variables and use them in our evaluation function. After all this, we were familiar with our data and ready to apply a prediction model to the data.

Method and Evaluation

In order to use a model on our data, we created a function that takes a classification model, a dataset, predicting variables, and the desired outcome variable as arguments. The function uses a cross validation technique to select a sample for testing, in this case our test data, for finalizing the model. The function prints out an accuracy score and a cross validation score.

We decided to use the Logistic Regression model from the sklearn library because our target variable is binary in nature and the prediction is a probability of an event occurrence. We chose Credit_History, Education, and LoanAmount as our predicting variables. Based on our analysis of the training data in pre-processing, those variables showed the most variance in probability. We created a new dataset which combined our training and testing data and used the model on it. Our outcome variable or target variable was the loan status.

Results

When we used the model on our combined dataset we used the logistic regression algorithm from the sklearn library, we got an accuracy score of 80.945% and a cross-validation score of 80.944%.

Our results were relatively good but there is definitely room for improvement.

Conclusion

With an accuracy score of 80.945% we are happy with our results; they provide a great starting point for accurately predicting loan approvals for this data. There are a number of ways we could improve our data such as using more data, testing different metrics, hyper-parameters, and/or algorithms, or tuning the algorithm we used.

References

M. One, "Using matplotlib in jupyter notebooks — comparing methods and some tips [Python]," Medium, 5 April 2018. [Online]. Available: https://medium.com/@1522933668924/using-matplotlib-in-jupyter-notebooks-comparing-methods-and-some-tips-python-c38e85b40ba1. [Accessed April 23, 2020].

N. Chauhan, "Real world implementation of Logistic Regression" Medium: Towards Data Science, 11 March 2019. [Online]. Available: https://towardsdatascience.com/real-world-implementation-of-logistic-regression-5136cefb8125. [Accessed April 13, 2020].

S. Ray, "8 Proven Ways for improving the "Accuracy" of a Machine Learning Model ," Analytics Vidhya, 29 December 2015. [Online]. Available: https://www.analyticsvidhya.com/blog/2015/12/improve-machine-learning-results/. [Accessed April 25, 2020].

S. Ray, "A Comprehensive Guide to Data Exploration," Analytics Vidhya, 10 January 2016. [Online]. Available: https://www.analyticsvidhya.com/blog/2016/01/guide-data-exploration/. [Accessed April 2, 2020].

Sklearn Documentation, "sklearn.model_selection.KFold," Scikit Learn. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.KFold.html. [Accessed April 15, 2020].