Get That Loan

Caitlin Sizemore, csizemore@bellarmine.edu

ABSTRACT
Using a loan eligibly dataset obtained through Kaggle.com and Python, I applied a logistic regression model. Before this, the data required preparation, also performed with Python. The logistic regression model applied uses variables from the dataset to predict if the applicant is available for the loan. Once predicted, I determined the accuracy of the model by comparing the actual and predicted values through Python.

I. INTRODUCTION

I obtained a dataset about loan eligibility. This dataset considers many variables, both continuous and discrete, like age, marital status, education and more to determine if you are eligible for the loan requested. Using Python, I prepared the data for analysis, applied a logistic regression model, predicted loan eligibility, and tested the accuracy of the regression model.

II. BACKGROUND

A. Data Set Description

The Loan Eligibility dataset used for analysis was obtained from Kaggle.com. Separate test and train csv files were provided; however, only the train file is used in this analysis. The data provides details about each applicant and their loan, ending with the loan status. These details include gender, marital status, number of dependents, education level, self-employment status, applicant and coapplicant income, loan amount, loan term, credit history, and property area. The purpose of the dataset is to determine loan eligibility based upon this information so the loan eligibility process can be automated. As a young adult in their college career, I will have to apply for loans in the near future. I chose this dataset to learn about the process and my eligibility.

B. Machine Learning Model

The logistic regression model applied with Python predicts the loan status using the variables previously listed using mathematical figures. The model determines the relationship between the independent and dependent variables and uses this relationship to predict the outcome. This outcome, also called classification, is a finite prediction, usually a simple yes or no. After predictions have been made, a confusion matrix and classification report are determined. These objects portray the accuracy of the model. A confusion matrix breaks each classification down by comparing the predicted to the actual. A classification report details the accuracy of several components of the regression model. Most important is accuracy, which uses the sum of true negative and true positives over the number of samples to determine the precision of the model. The closer this number is to 1, the more accurate the model is.

III. EXPLORATORY ANALYSIS

This dataset contains 614 samples and 12 columns of various data types. In total, 7 columns had missing values. Of the 7, 4 were discrete and 3 were continuous. Gender, Married, Dependents, and Self-Employed are categorical/discrete data and LoanAmount, Loan_Amount_Term, and Credit_History are continuous. Mode is required and was used to fill in the categorical missing data. Referring to Figures 1 (LoanAmount), 2 (Loan_Amount_Term), and 3 (Credit_History) it is clear that each of the continuous variables with missing data are skewed, meaning median must be and was used to fill in.

Table 1: Data Types

Variable Name	*Data Type*
Gender	object/discrete
Married	object/discrete
Dependents	object/discrete
Education	object/discrete
Self-Employed	object/discrete
ApplicantIncome	int64/continuous
CoapplicantIncome	float64/continuous
LoanAmount	float64/continuous

Loan_Amount_Term	float64/continuous
Credit_History	float64/continuous
Property_Area	object/discrete
Loan_Status	object/discrete

IV. METHODS

A. Data Preparation

The column "Loan_ID" was dropped from the dataset due to its lack of relation for analysis. This column provides a unique ID number for each loan applicant. Since it simply serves as an identifier and provides no information about the applicant or loan numbers, it is unrelated and unnecessary for prediction with the model. To drop this column, I called every column of the dataset except for "Loan_ID" under its previous name "dataset."

B. Experimental Design

Table X: Experiment Parameters

Experiment Number	Parameters
1	80/20 split for Train and Test
2	75/25 split for Train and Test
3	65/35 split for Train and Test

C. Tools Used

The following tools were used for this analysis: Python v6.4.8 running the Anaconda 2022.05 environment for Apple Macintosh computer was used for all analysis and implementation. In addition to base Python, the following libraries were also used: Pandas 1.4.2, Numpy 1.21.5, Matplotlib 3.5.1, Seaborn 0.11.2, SKLearn 1.0.2.

Python Libraries Used:

- Pandas: Used to upload and view the csv file and to convert categorical variables to numerical values
- Numpy: Used to impute missing values – median and mode
- Matplotlib: Used to graph the confusion matrix
- Seaborn: Used to graph columns containing missing values to visualize the measure of centrality
- SKLearn: Used to split into training and test set, fit the logistic regression model, and determine the confusion matrix and classification report

V. RESULTS

A. Classification Measures

See the Appendix for visual representations of the confusion matrix and classification report for each experiment. As stated, Figure 4 represents Experiment 1, Figure 5 represents Experiment 2, and Figure 6 represents Experiment 3.

B. Discussion of Results

Overall, the logistic regression model is accurate at predicting loan eligibility with each of the three experiments scoring an accuracy of at least 75%. In addition, each confusion matrix's highest number of predictions resides in True Positive while the lowest in False Negatives. Unfortunately, each experiments False Positive was the 2nd highest prediction. This is not preferable because it is predicting that the applicant receives the loan when they actually do not receive the loan.

C. Problems Encountered

As discussed, 4 of the 7 missing values were categorical/discrete in nature. I had trouble imputing these missing values and had to do additional research to learn how to impute and fill in missing categorical data. Through the research, I discovered that mode is most accurate for imputing categorical data. I had not originally realized my imputations were not working and did not discover this until the logistic regression model would not work. Once the categorical missing values were imputed, I did not encounter any additional problems.

D. Limitations of Implementation

Logistic regression is the best model to be applied for analysis. Because we are predicting eligibility on a yes/no classification basis, linear regression would be inapplicable and insufficient.

E. Improvements/Future Work

For future improvement, I would like to combine the given test and train csv files to gather more information for prediction. I believe the model would be more accurate with this information. Running additional experiments with this information will be helpful.

VI. CONCLUSION

Overall, despite the smaller size of samples, the model is largely accurate. While it has an unfortunately large amount of False Positive predictions, it also has a remarkably large amount of True Positive with less than two False Negative Predictions along with an accuracy of over 75%. The dataset required a large amount of preparation, including removing the "Loan_ID" column and imputing missing values for 7 columns. Using Python and its related libraries, it was easy to apply the logistic regression model for three separate experiments of varying splits to find these favorable and accurate results. Despite the varying splits, each experiment had a similar confusion matrix and classification report showing the overall accuracy of the model.

VII. APPENDIX

Figure 1:

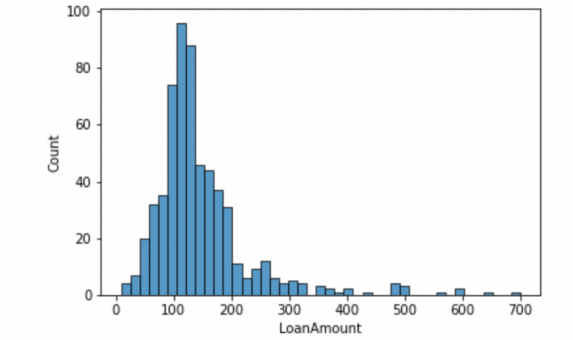


Figure 2:

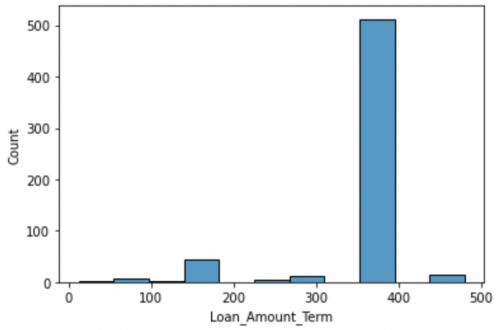


Figure 3:

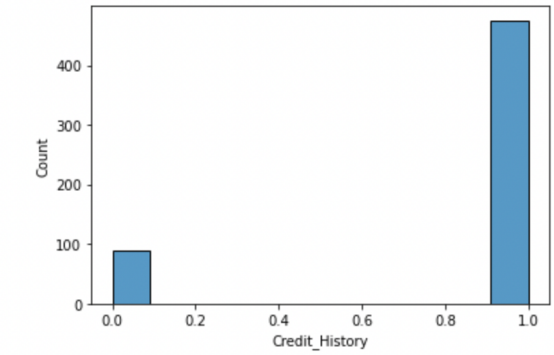
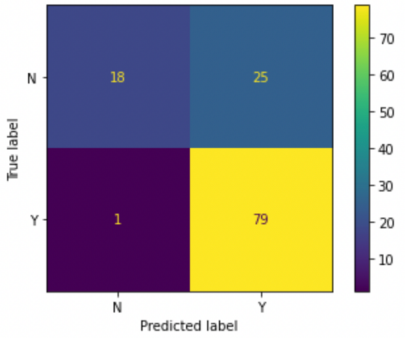
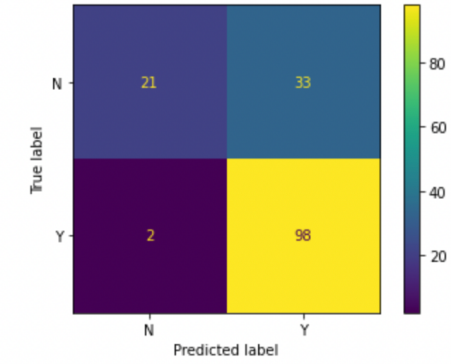


Figure 4: Experiment 1



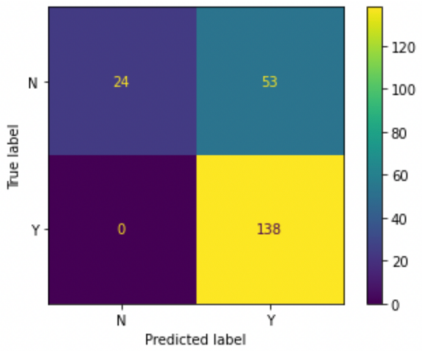
	precision	recall	f1-score	support
N	0.95	0.42	0.58	43
Y	0.76	0.99	0.86	80
accuracy			0.79	123
macro avg	0.85	0.70	0.72	123
weighted avg	0.83	0.79	0.76	123

Figure 5: Experiment 2



	precision	recall	f1-score	support
N	0.91	0.39	0.55	54
Y	0.75	0.98	0.85	100
accuracy			0.77	154
macro avg	0.83	0.68	0.70	154
weighted avg	0.81	0.77	0.74	154

Figure 6: Experiment 3



	precision	recall	f1-score	support
N	1.00	0.31	0.48	77
Y	0.72	1.00	0.84	138
accuracy			0.75	215
macro avg	0.86	0.66	0.66	215
weighted avg	0.82	0.75	0.71	215