Mapúa University
School of Electrical, Electronics, and
Computer Engineering

Design of Experiment: Loan Prediction

CPE106L (Software Design Laboratory)

Hannah Antaran
Jonathan Ignacio
Charlene Rabulan
Kathleen Tupas
Darrel Virtusio

Group: 1
Section: E04

ABSTRACT

Loan prediction is a very common real-life problem that every retail bank faces in their lending operations. If the loan approval process is automated, it can save a lot of man hours and improve the speed of service to the customers. The increase in customer satisfaction and savings in operational costs are significant. However, the benefits can only be reaped if the bank has a robust model to accurately predict which customer's loan it should approve and which to reject, in order to minimize the risk of loan default. With that being said, this project aims to help retail banks by depicting a model which predicts the customers eligibility for loan. The program made use of the Logistic Regression Modeling technique to show the prediction model and as inferred from the confusion matrix, more than half of the applicants are eligible for loan.

Keywords: Loan, Loan Prediction, Exploration Data Analysis, Logistic Regression

OBJECTIVES

The number of people applying for loans have increased for various reasons in recent years. The bank employees are not able to analyze or predict whether the customer can pay back the amount or not for the given interest rate. With that being said, the aim of this project is to analyze the nature of clients applying for personal loan and predict their eligibility for loan to help retail banks improve their speed of service to the customers.

1. Understand the nature of each client using the exploration data analysis.
2. Create a prediction model displaying the number of clients eligible for loan and those who are not based on the exploration data analysis.
3. Use the Logistic Regression Modeling technique to predict the target variable.

INTRODUCTION

Among all industries, the insurance domain has one of the largest uses of analytics & data science methods. Technologies' contributions are not limited only to the addition of value to the insurance sector, but they are now determining its very growth and evolution.

During the 20th Century, people have pioneered the usage of technology and the internet through wireless connection and paperless transactions. This sparked the change of convenient banking and easier access of loans and bank accounts. As more users switched to online banking, the demands for online agents also increased, this gave the bankers the idea to use online loaning systems and coded predictions. These websites or applications would measure if a person would be qualified to loan money and if they are capable of paying it back, some websites also check your credit scores based on your job and overall net worth. With the coded applications are getting more advanced, there are more factors that the banks are putting in to account to make the transactions more trustworthy and polished.

HYPOTHESIS GENERATION

Possible factors that can affect the outcome which will have an impact on whether a loan will be approved or not. Some of the hypothesis are listed below:

- Education – applicants with higher education level should have higher chances of loan approval.
- Income – applicants with higher income should have more chances of loan approval.
- Loan Amount – if the loan amount is less, the chances of loan approval should be high.
- Loan Term – loans with a shorter time period should have higher chances of approval.
- Previous Credit History – applicants who have repaid their previous debts should have higher chances of loan approval.
- Monthly Installment Amount – if the monthly installment amount is low, the chances of loan approval should be high.

DATA COLLECTION

The data have already been provided by Kaggle. The training set will be used for training the model which contains all the independent variables and the target variable. The test set contains all the independent variables, but not the target variable. To predict the target variable for the test data, the model will be applied. There are 13 columns of features and 614 rows of records in the training set and 12 columns of features and 367 rows of records in the test set. The dataset variables are summarized below:

Variable	Type	Description
Loan_ID	Numerical – Discrete	Unique Loan ID
Gender	Categorical – Nominal	Male/Female
Married	Categorical – Nominal	Married (Y/N)
Dependents	Categorical - Ordinal	Number of Dependents
Education	Categorical – Nominal	Graduate/ Undergraduate
Self_Employed	Categorical – Nominal	Self-employed (Y/N)
ApplicantIncome	Numerical – Continuous	Applicant Income

CoapplicantIncome	Numerical – Continuous	Co-applicant Income
Loan Amount	Numerical – Continuous	Amount in thousands
Loan_Amount_Term	Numerical – Discrete	Term of loan in months
Credit_History	Categorical - Nominal	Credit history meets guidelines
Property Area	Categorical - Ordinal	Urban/Semi-urban/Rural
Loan_Status	Categorical - Nominal	Loan Approved (Y/N)

Table 1. Summarized Dataset Variables

EXPLORATION DATA ANALYSIS

This analysis is an approach to analyse the datasets that summarizes its main features with visual methods. The purpose of using this analysis is to uncover the underlying structure of a relatively larger set of variables using visualization techniques.

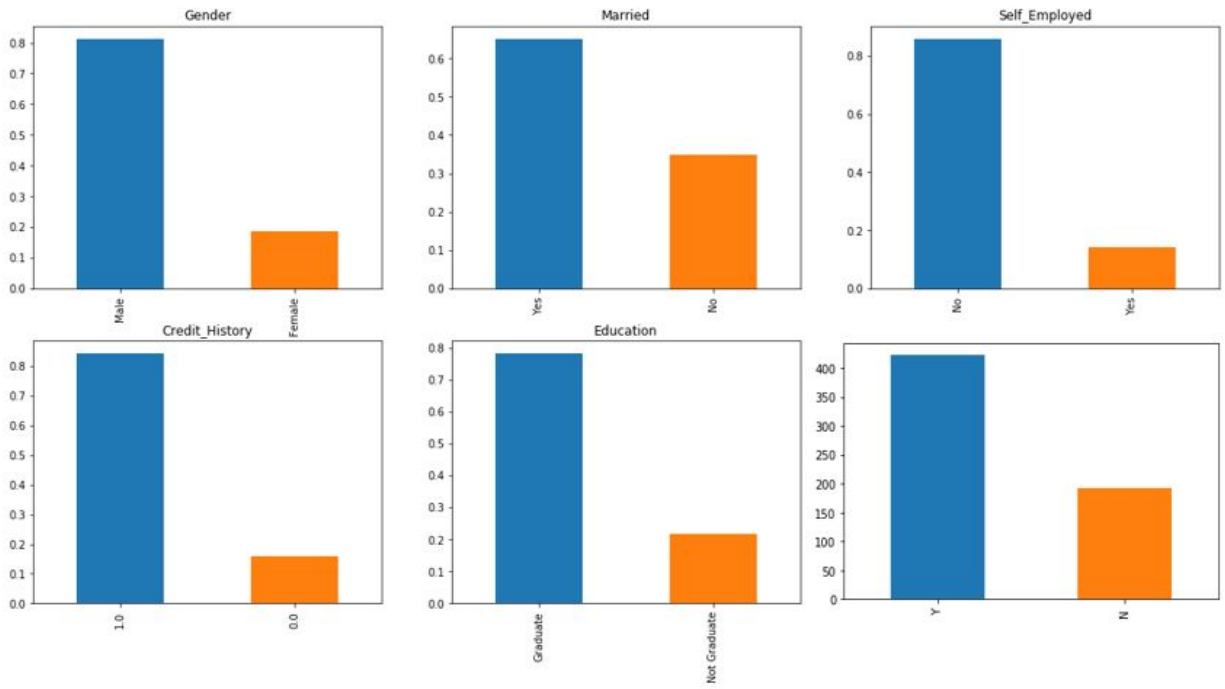


Figure 1.1 Independent Variable and Target Variable (Categorical)

It can be inferred from the above bar plots that:

- There are more men than women.
- Around 65% of the applicants are married.
- Around 15% of the applicants are not self-employed.
- Around 85% of the applicants have repaid their debts.
- Around 80% of the applicants are graduates.
- Around 69% of the applicants were approved for loan.

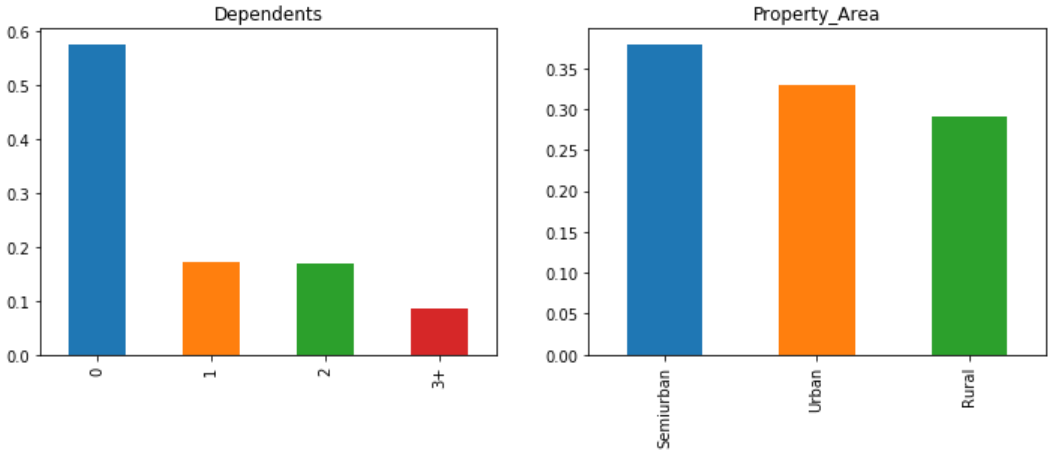


Figure 1.2 Independent Variable (Ordinal)

Following inferences can be made from the above bar plots:

- Majority of the applicants have zero dependents.
- Most of the applicants are from the suburban area.

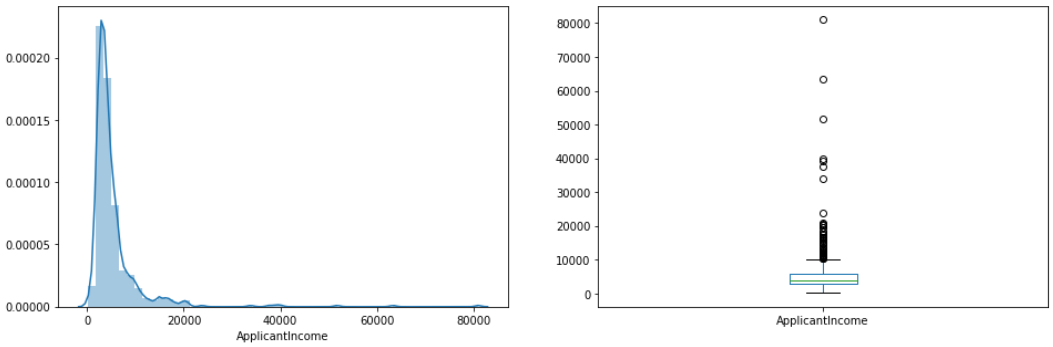


Figure 1.3 Independent Variable for the Applicant's Income (Numerical)

It can be inferred that most of the data in the distribution of applicant income is towards the left which means it is not normally distributed. The boxplot confirms the presence of a lot of extreme values. This can be attributed to the income disparity in the society. Part of this can be driven by the fact that we are looking at people with different education levels.

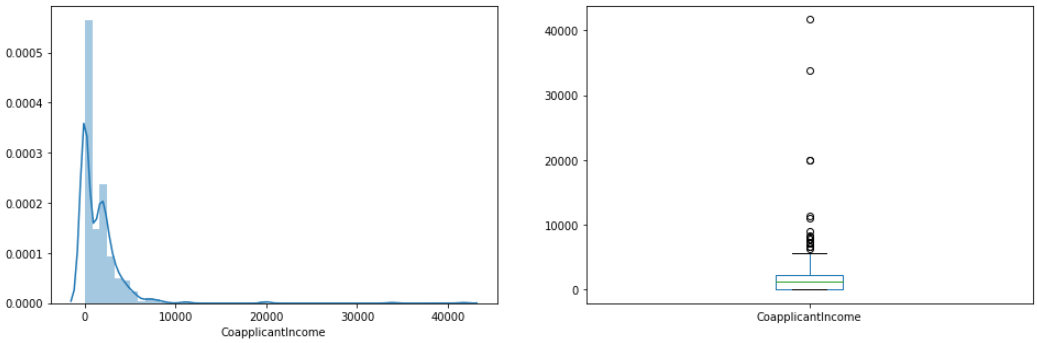


Figure 1.4 Independent Variable for the Co-applicant's Income (Numerical)

In this figure, a similar distribution as that of the applicant income can be inferred. Majority of the co-applicant's income ranges from 0 to 5000. This shows a lot of extreme values in the co-applicant's income and it is not normally distributed.

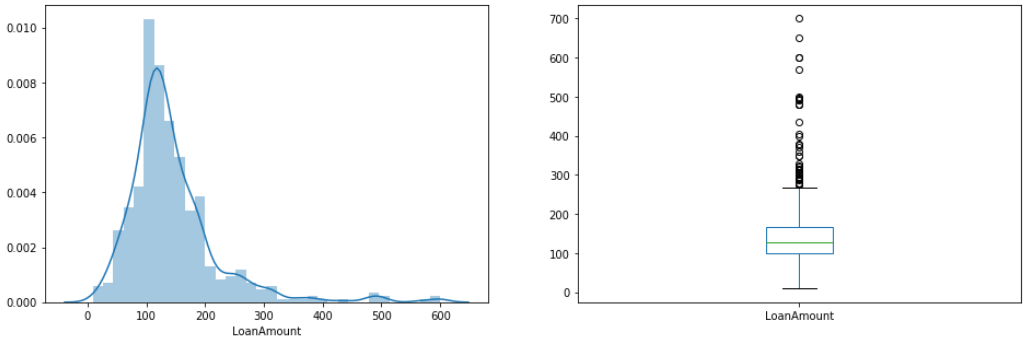


Figure 1.5 Independent Variable for the Loan Amount (Numerical)

This figure shows a fairly normal distribution, however, there are a lot of extreme values.

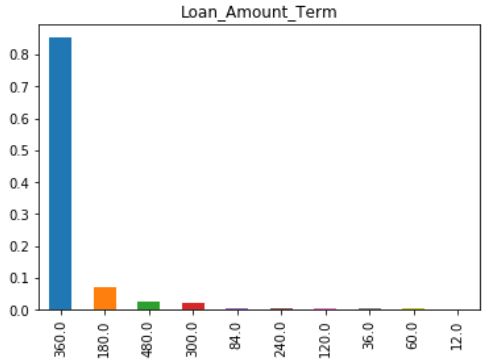


Figure 1.6 Independent Variable for the Loan Term (Numerical)

It can be inferred from the bar plot above that around 85% of the loans are 360 months term or 30 years period.

CLASS DIAGRAM

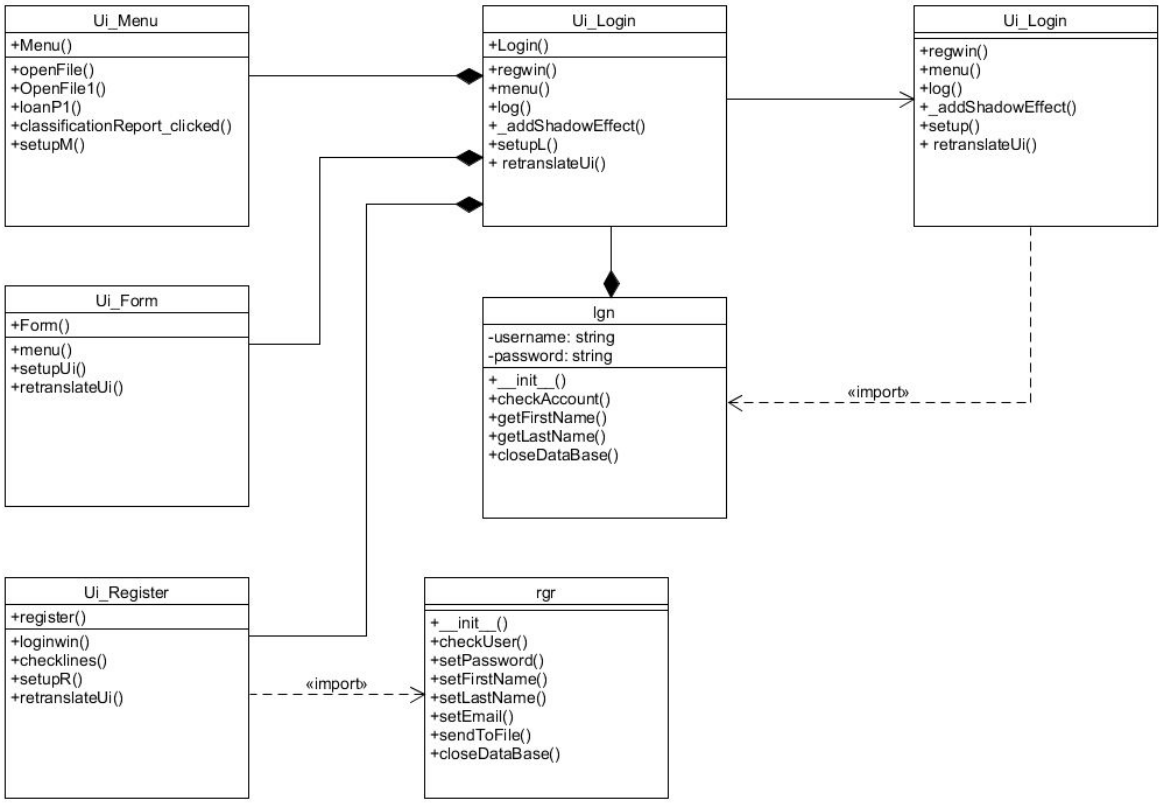
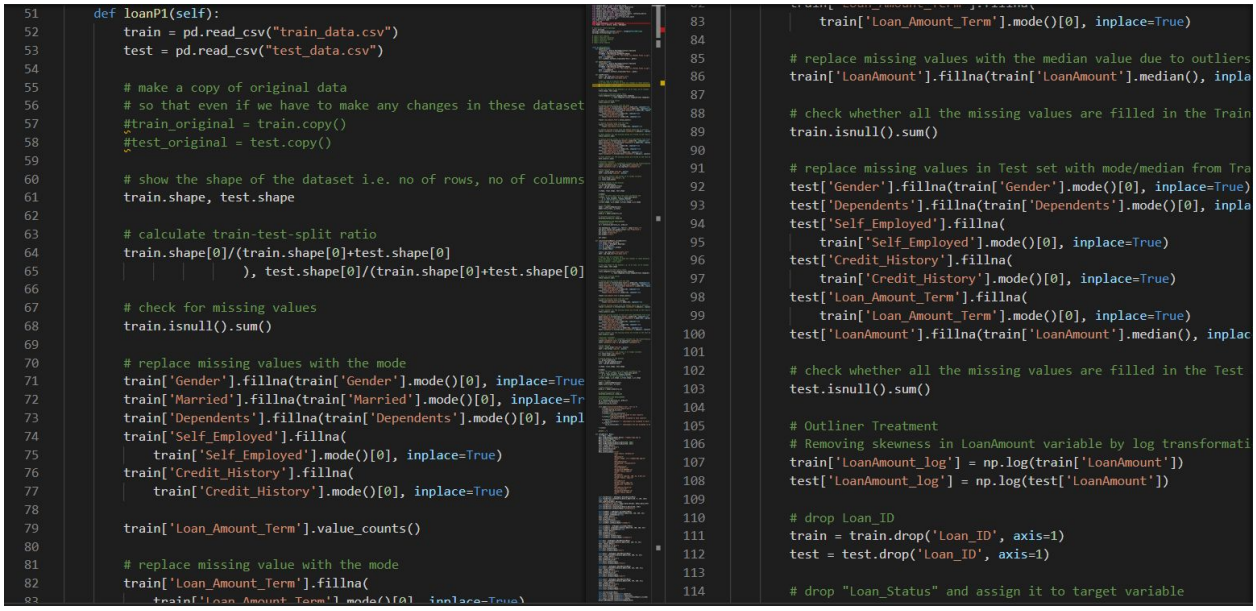


Figure 3.1 Class Diagram of the Program

As you can see from the figure above, this is the class diagram of the program where it shows the relationship between the classes. One of types of relationship in this diagram is the composition where a class cannot exist independently or on its own. The other relationships in this diagram are dependency and association, depency shows the relationship of imports and association represents the connection of two classes.

SOURCE CODE



```
def loanP1(self):
    train = pd.read_csv("train_data.csv")
    test = pd.read_csv("test_data.csv")

    # make a copy of original data
    # so that even if we have to make any changes in these dataset
    #train_original = train.copy()
    #test_original = test.copy()

    # show the shape of the dataset i.e. no of rows, no of columns
    train.shape, test.shape

    # calculate train-test-split ratio
    train.shape[0]/(train.shape[0]+test.shape[0]
                    ), test.shape[0]/(train.shape[0]+test.shape[0]

    # check for missing values
    train.isnull().sum()

    # replace missing values with the mode
    train['Gender'].fillna(train['Gender'].mode()[0], inplace=True
    train['Married'].fillna(train['Married'].mode()[0], inplace=Tr
    train['Dependents'].fillna(train['Dependents'].mode()[0], inpl
    train['Self_Employed'].fillna(
        train['Self_Employed'].mode()[0], inplace=True)
    train['Credit_History'].fillna(
        train['Credit_History'].mode()[0], inplace=True)

    train['Loan_Amount_Term'].value_counts()

    # replace missing value with the mode
    train['Loan_Amount_Term'].fillna(
        train['Loan_Amount_Term'].mode()[0], inplace=True)

    # replace missing values with the median value due to outliers
    train['LoanAmount'].fillna(train['LoanAmount'].median(), inpla

    # check whether all the missing values are filled in the Train
    train.isnull().sum()

    # replace missing values in Test set with mode/median from Tra
    test['Gender'].fillna(train['Gender'].mode()[0], inplace=True)
    test['Dependents'].fillna(train['Dependents'].mode()[0], inpla
    test['Self_Employed'].fillna(
        train['Self_Employed'].mode()[0], inplace=True)
    test['Credit_History'].fillna(
        train['Credit_History'].mode()[0], inplace=True)
    test['Loan_Amount_Term'].fillna(
        train['Loan_Amount_Term'].mode()[0], inplace=True)
    test['LoanAmount'].fillna(train['LoanAmount'].median(), inplac

    # check whether all the missing values are filled in the Test
    test.isnull().sum()

    # Outliner Treatment
    # Removing skewness in LoanAmount variable by log transformati
    train['LoanAmount_log'] = np.log(train['LoanAmount'])
    test['LoanAmount_log'] = np.log(test['LoanAmount'])

    # drop Loan_ID
    train = train.drop('Loan_ID', axis=1)
    test = test.drop('Loan_ID', axis=1)

    # drop "Loan_Status" and assign it to target variable
```

Figure 4.1 Sample Code

The sample source code in figure _ shows the main function of the program. It is the part of the program where machine learning is applied. This means that our loan prediction dataset were trained and tested to acquire the prediction model. Mainly, we used scikit-learn as the machine learning library for this project. We also used the pandas library to read our datasets. And for data visualization, we used seaborn and matplotlib as the main library.

SAMPLE OUTPUT

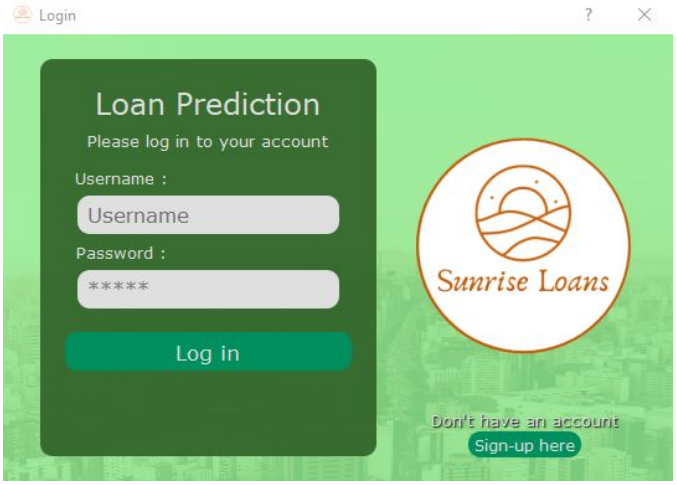


Figure 4.2 Log-in window

The log-in window requires you to enter a registered account before you access the loan prediction window. If the user does not have any registered account, the user can sign up using the sign up button located in the lower right corner and the user will be directed to the register

window and then can proceed to register his/her account. For future upgrades, we would like to recommend another feature that lets the user to reset their password whenever the user forgot their old password.

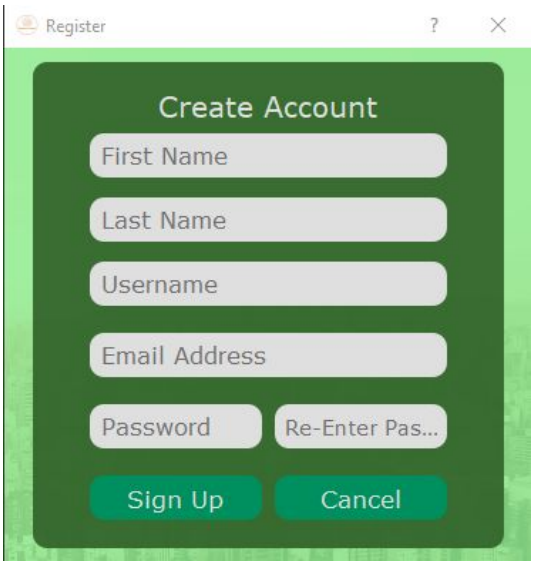


Figure 4.3 Register window

The register window looks like this. If the user has no account, he/she needs to register one before accessing the loan prediction window. In the register window, the user is required to enter their first name, last name, username, email address, and password. If the username is already taken by another user, it will prompt the user to enter another username. As the user clicks the sign up button, it will save their data into the database created by the group.

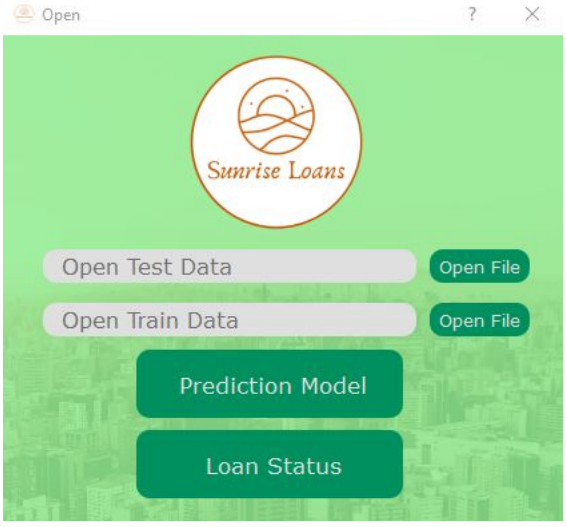


Figure 4.4 Main window

Once the user has successfully created an account and has logged in, the main window will prompt. This window allows the user to enter their loan prediction datasets and have it predicted whether the data stored in that dataset will get their loans accepted or not. The prediction model

button will display a confusion matrix that corresponds to the loan prediction dataset. The loan status button will output a text file that is the summary of the confusion matrix. It will explain how many people from the dataset get their loan accepted or rejected.

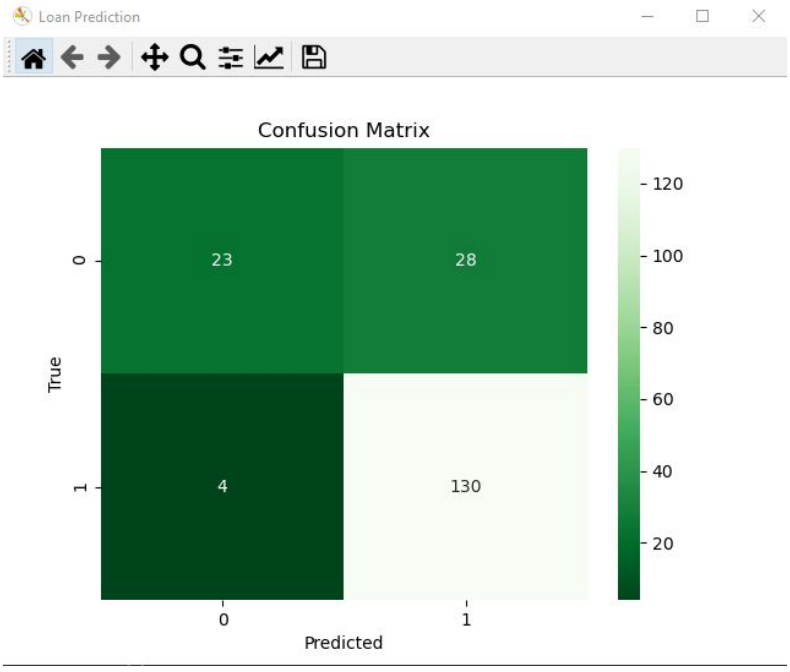


Figure 4.5 Confusion matrix

The confusion matrix displayed here is the predicted model of the loan prediction dataset. It represents the number of people who are eligible or not eligible for a loan. The meaning of these numbers predicted by the program is explained in the analysis of the result.

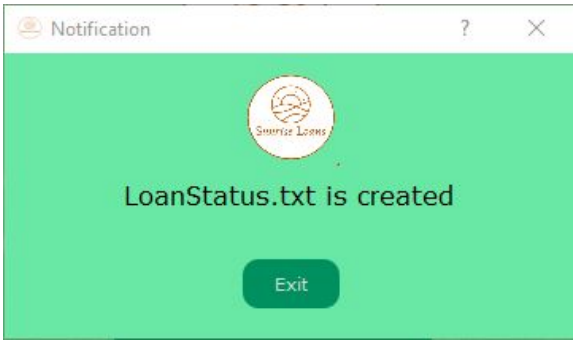


Figure 4.6 Notification window

The notification window is an alert that lets the user know that the text file for the loan status summary report is created. This will only appear when the user clicks the loan status button from the main window and wishes to read the summary of the report.

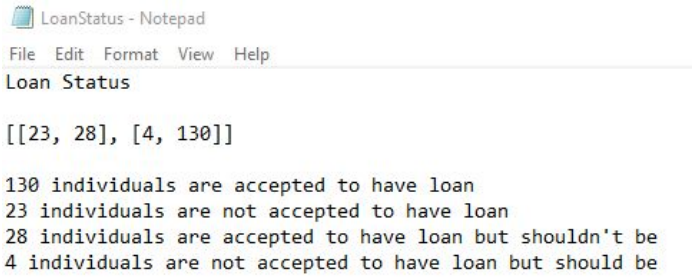


Figure 4.7 Loan status summary report

The text file of the loan status displays the confusion matrix in a multidimensional array version. Below that, it contains the summary of the report of the number of people who are eligible or not eligible to apply for the loan.

ANALYSIS OF THE RESULTS

With the Loan Prediction Model working, the program produces results through a confusion matrix which presents the number of people who are eligible and not eligible for a loan. A confusion matrix is a table that easily classifies a test data if its values are true or not. For the Loan Prediction Model, the group has classified the registered individuals into four categories which portions the confusion matrix into four parts. As seen below, the Confusion Matrix is divided into two rows and two columns with labels of 0s and 1s. In the x-axis, the label of 0 and 1 means that the program predicts whether an individual should receive a loan or not. On the other hand, the y-axis labels of 0 and 1 shows if the program either accepts or rejects the individuals registered in the program. When the label is 0, it means that the program either does not accept the individuals to have a loan or predicts that they should not receive any loan for they are considered as a false or negative output in the Confusion Matrix Concept. If the label for the row and column is 1, the program either accepts the individuals to have a loan or predicts that they should receive a loan for the label 1 is considered as true or positive in the Confusion Matrix Concept.

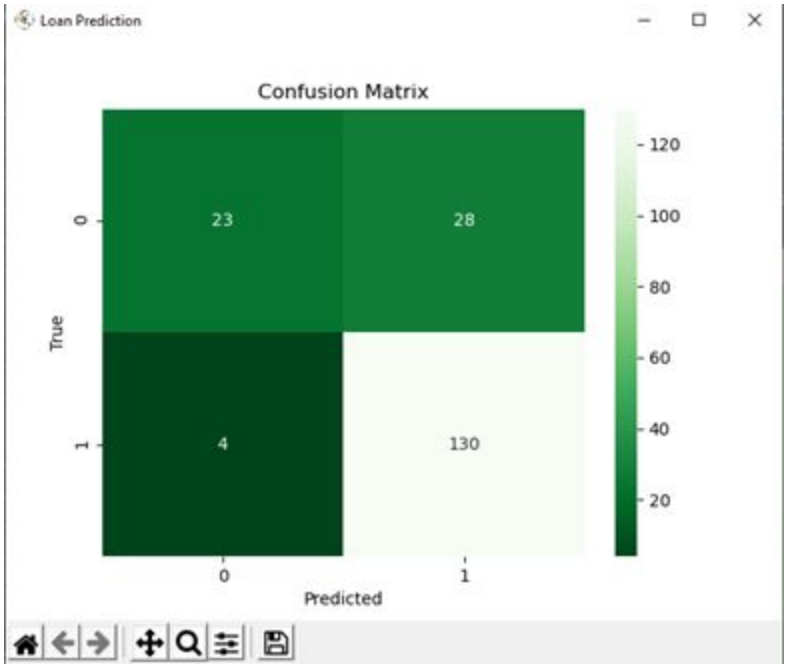


Figure 5.1 Confusion Matrix

This figure shows the produced Confusion Matrix of the given data set used for this Loan Prediction Model. Here, a large square is portioned into four smaller squares where the first square on the upper left can be labelled as box A, upper right square as box B, lower left square as box C, and lower right square as box D. In box A, the calculated number inside is the number of individuals registered in the Loan Prediction Program who were not accepted to receive a loan for they are not eligible enough to receive a loan. In box B, the number inside represents the individuals who have registered and been accepted by the program but should not have received a loan. In box C, the number shown is the number of individuals who were not accepted by the program to have loan but should receive a loan. In box D, the number inside shows the individuals who have registered in the Loan Prediction Program and have been accepted to receive a loan because they are eligible for a loan. The matrix also shows a color legend beside the matrix on the right where the group has used shades of green to represent the number of individuals involved, partitioning the color legend through an interval of 20 people. As seen in the color legend, the shade of green gets lighter as the number increases.

The concept between being accepted or not and predicting if the individuals should receive or not receive a loan is different. At a general outlook, the confusion matrix allows an audience to see if values involved are true or not, however, it specifically details correct and incorrect values which is why it is portioned into four: True Positive (1-1), True Negative (1-0), False Positive (0-1), False Negative (0-0). From the boxes, box A is False Negative as the number of individuals inside are not accepted by the program and are not eligible enough for a loan. Box B is False Positive as the program accepts these individuals but are actually not eligible for a loan. Box C is True Negative as the individuals here are not accepted by the program but are eligible for a loan. For box D, it is a True Positive since the individuals here have been approved or

accepted by the program and are actually eligible for a loan. The results of this matrix are summarized as the program outputs a text file for a loan status report as seen below:

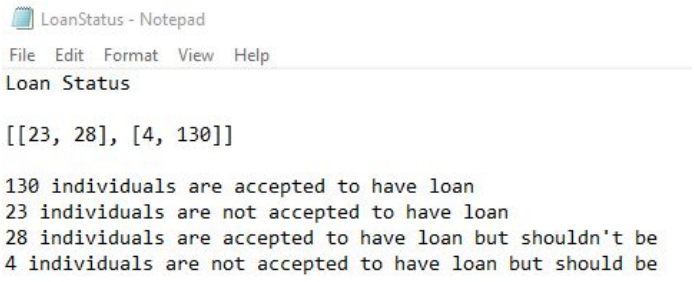


Figure 5.2 Loan Status Report

In this figure, the number of people who have registered are classified into four parts: accepted to have a loan, not accepted to have a loan, accepted to have a loan but should not have a loan, not accepted but should have a loan. As discussed in figure 1, box A shows a number of 23 individuals who are not accepted by the program to receive a loan for they are not eligible to have a loan. For box B, 28 individuals were accepted by the program to have a loan but are actually not eligible to have a loan. For box C, 4 individuals are not accepted by the program to have a loan but should receive a loan for they are eligible to have a loan. In box D, 130 individuals are accepted by the program to receive a loan because they are eligible enough to have a loan. Hence, the Loan Prediction Model made has predicted not only if the registration of the individuals for a loan is approved or not, but rather the program has specifically calculated the number of individuals registered through four categories as seen in the Loan Status Report .txt file.

CONCLUSION

As the application requires an account to log-in, the users are needed to be identified by the app first. This gives an added security buffer for people who are using the software and assurance of only you and the developer only knows if you are qualified or not. Multiple tests are made to justify the screening and all tests passed. As seen in the analysis of the results, the confusion matrix has separated the users into 4 different sectors. They are screened through the code if their qualifications are passing and due to the simplicity of the code, there is a little margin of error that is also calculated. This margin represents the users who got a different evaluation that they are supposed to have, but all of these are part of the calculation and are predicted to appear. All in all, the software did its job with 82.70% accuracy and is very effective.