**Logo

Description automatically generated**

**BUDT 704 - Data Processing and Analysis in Python**

**Professor** **Kunpeng 'KZ' Zhang**

**Loan Approval Prediction**

**Using Machine Learning**

Alexandra L. Dakhniuk

Nishant Patel

SIddhant Pillai

Vinu Ratnayake

**2. Project overview**

**2.1 General**

Machine learning is a field of data science that helps analyze the data and make decisions based on the predictions. Every industry, every area of business deals with data in one way or another and therefore data science, along with machine learning have become widely researched areas of computer science in the last decade. Machine learning offers various models using numerous algorithms for manipulation and analysis.

Our group had researched several methods while building the proposed model.

We have manipulated the data to reduce the time spent by loan officers and make the model more efficient and improve the accuracy.

Our project consisted of four major parts: data set selection and cleansing, analysis of several machine learning methods, training and testing the dataset. After a thorough data analysis, our group has concluded that in the future the loan approval system could be seamlessly integrated with the auto-processing system.

For the accuracy purposes, we need to collect and preserve customer’s financial information, specifically credit history, any remarks, or delinquencies to ensure the safe loan disbursement, and process the data with exploratory analysis. For loan generation, we suggested using the map function to predict the results based on the given data. We have also utilized a decision tree induction algorithm for model implementation, credit score review and applications approval. Credit score is one of the determining factors in the loan approval process. Our generated model has predicted that applications of the majority of low-income customers would be approved as those customers are heavily relying on the loans, carefully maintain their credit history and are more likely to repay the loan. We have used the test set for the application approval process.

For the purpose of this project our group has used the dataset found on Kaggle.com. The objective of the project is to create a model for loan repayment and prediction. For the development of our prediction model, we used data mining, specifically to go over three processes: data cleaning, classification and data adjustment.

We have also utilized the following machine learning algorithms: Logistic Regression, and the Decision Tree. After running the dataset on the algorithms mentioned above, Random Forest yielded the most accurate results.The purpose of the developed model was to screen out the applicants.

For our classifier technique we combined KNN algorithm and min-max Normalization, which yielded a 75.08% accuracy.

We were not able to single out the best performing model as each model has better performance in one way or another. Therefore, we have decided to create a hybrid model that would combine the features from several models at once to improve efficiency and accuracy.

**3. Implementation of Created Model**

3.1 Existing System

The purpose of the existing system is to analyze applicant’s information and predict the chances of loan repayment. We found out several loopholes in the existing system, specifically its inability to check whether all the information on the application is provided and could be verified. The system does not have an automatic verification process and therefore the bank could extend the loans to the individuals who may otherwise not qualify for them, have means and the ability to repay the loans.

3.2 Proposed System

Our proposed system will have the ability to predict an applicant’s chances of being approved for the loan. We have developed a model by using the loan approval datasets and recognizing  the patterns for loan approval or denial. This model is developed using the combination of several machine learning algorithms.

For the proposed model, we are using two different datasets: training and testing datasets. First, we train the datasets using the Decision Tree algorithm and develop a first version of the prediction model. Next, we use the training dataset to test the model. The objective of the proposed model is to predict the chances of loan repayment. For this project we have used such python libraries as pandas, numpy, sklearn etc. The first thing done with the dataset was data cleaning, which took a great deal of overall project time: we checked for missing values, erroneous entries and separating numerical and categorical variables.We then performed the analysis on the outliers by creating a boxplot diagram of the attributes and frequency analysis.

Diagram

Description automatically generated with medium confidence

Diagram

Description automatically generated with low confidence

## 3.2.1 Data

For this project, we have two subsets: training dataset and testing dataset.

The training dataset will be used to train the model which means the model will learn from this data. It contains all the independent variables and the dependent variable which is loan status.

The testing dataset contains all the independent variables, but not the target variable. The model we trained using the training data will predict the dependent variable which is the loan status for the testing data.

**DATA DICTIONARY**

|  |  |
| --- | --- |
| **Variable** | **Description** |
| Loan\_ID | Unique Loan ID for each loan application |
| Gender | Male/ Female |
| Married | Whether the Applicant is married or not (Y/N) |
| Dependents | Number of dependents in applicant family |
| Education | Level of Applicant Education (Graduate/ Undergraduate) |
| Self Employed | Is the applicant self-employed (Y/N) |
| ApplicantIncome | Applicant income |
| CoapplicantIncome | Co-applicant income |
| LoanAmount | Loan amount in thousands of Dollars |
| Loan Amount Term | Term of loan in months |
| Credit\_History | Credit history meets guidelines |
| Property\_Area | Urban/ Semi Urban/ Rural |
| Loan Status | (Target) Loan approved (Y/N) |

**Reading the Data**

**Text

Description automatically generated**

**Text

Description automatically generated**

**Understanding the Data**

We have three data types:

* Object: used for categorical variables. Categorical variables in our selected dataset are: Loan\_ID, Gender, Married, Dependents, Education, Self\_Employed, Property\_Area, Loan\_Status.
* Float64: used for numerical variables. In our dataset the variables of this format are: CoapplicantIncome, LoanAmount, Loan\_Amount\_Term, Credit\_History
* Int64: used for numerical variables. ApplicantIncome is the only variable of this type in our dataset

Graphical user interface, text, website

Description automatically generated

We have 614 rows and 13 columns in the selected dataset.

Graphical user interface, text, application, website

Description automatically generated

Using the value\_counts() function and setting “normalize” to True we could display proportions of “Yes” and “No”, instead of numbers.

Graphical user interface, text, application

Description automatically generated

The ratio of approved vs. denied loans is 2:1. Therefore, the majority of loan applications were approved (almost 69%)

Chart, bar chart

Description automatically generated

We can now move to analyze each variable separately. Throughout our project we had categorical, ordinal, and numerical variables.

* Categorical variables (having some categories to them): Gender, Married, Self\_Employed, Credit\_History, Loan\_Status.
* Ordinal variables (having some order): Dependents, Education, Property\_Area.
* Numerical variables (having numerical values): ApplicantIncome, Co-applicantIncome, LoanAmount, Loan\_Amount\_Term

**Univariate Analysis**

**Text

Description automatically generated**

We have created  bar plots to analyze  the distribution for each variable.

Chart, bar chart

Description automatically generated

Icon

Description automatically generated with medium confidence

A picture containing bar chart

Description automatically generated

A picture containing chart

Description automatically generated

From the above bar graphs, it can be summarized that 80% of the applicants in the dataset are males, approximately 65% of applicants in the dataset are married, 15% of applicants in the dataset are self-employed and 85% of applicants in the dataset have good credit history.

**Independent Variable (Ordinal)**

**Graphical user interface

Description automatically generated with low confidence**

This graph shows that a majority of the applicants do not have any dependents (around 60%).

**Chart

Description automatically generated**

This graph shows that a majority of the applicants are college graduates (around 80%)

**A picture containing chart

Description automatically generated**

This graph shows that the majority of applicants are from the semi urban areas (around 40%) and around 33% are from the urban area and around 30% is from the rural area.

**Icon

Description automatically generated**

**Independent Variable (Numerical)**

At this point we will start visualizing numerical variables. We will look at the distribution of the applicant’s income.

Text

Description automatically generated

Shape

Description automatically generated with medium confidence

This chart shows us the distribution of the ApplicantIncome.

Chart

Description automatically generated with medium confidence

Based on the graph above we can conclude that the data is not normally distributed. From the boxplot we can see that a lot of outliers or extreme values are present in our data, which can be tied to a problem of wealth distribution in our country. Another contributing factor could be the highest levels of education completed.

Graphical user interface, text, application

Description automatically generated

Chart

Description automatically generated

Based on the boxplot above, we can conclude that more applicants who have a higher level of education have much higher incomes, which seem to be outliers.

When we add the Co-Applicant income variable, we can also observe the unevenly distributed data with many outliers.

Text

Description automatically generated

We have created the chart for the Co Applicant Income distribution.A picture containing histogram

Description automatically generated

The box plot shows the existence of many outliers and extreme values for Co Applicant Income.

Chart, box and whisker chart

Description automatically generated

Text

Description automatically generated

We created a chart to visualize the distribution of Loan Amounts.

Histogram

Description automatically generated

Chart, box and whisker chart

Description automatically generated

When we look at the loan amount box plot and chart above, we can observe a lot of outliers, and normal distribution.

**Bivariate Analysis**

Some of the conclusions generated from the univariate analysis were that:

1. Applicants with higher income should be given more preference for a loan
2. Applicants with a good credit score, who repay their debts on time should be given a higher preference for a loan.
3. Loans with a lower principal should have a higher chance of approval.
4. The lesser the monthly loan payment amount is, the higher is the rate of approval.

After analyzing all the variables independently, we now analyze each variable with respect to the target variable.

**Categorical Independent Variable vs Target Variable**

In this section we will find the relationship between the dependent variable and categorical independent variables. Plotting a stacked bar chart will show us the proportion of approved and unapproved loans.

Text

Description automatically generated

Chart, bar chart

Description automatically generated

The chart above shows that the proportion of male and female applicants is approximately the same for both approved and unapproved loans.

Text

Description automatically generated

The chart below shows that the proportion of married applicants is higher for approved loans.

Chart, bar chart

Description automatically generated

The chart below shows that the distribution of applicants with 1 or 3 plus dependents is similar across approved and unapproved loans.

Graphical user interface, text

Description automatically generated

The chart below shows that the proportion of college graduates is higher for approved loans.

A picture containing chart

Description automatically generated

The chart below shows that the attribute self employed has little effect on loan approval.

Graphical user interface, text

Description automatically generated

According to the chart below people with good credit history represented as 1 in the chart, has a higher percentage of loan approval.

Graphical user interface, text

Description automatically generated

According to the chart below the proportion of loans getting approved in the semi urban areas is higher as compared to that in the rural or urban area

Chart, bar chart

Description automatically generated

**Numerical Independent Variable VS Target Variable**

In this section , we discuss the impact numerical variables have with the target variable , specifically the mean income of applications whose loan applications were denied versus the mean income of applications who received the loans.

Graphical user interface, text, application, website

Description automatically generated

Logo

Description automatically generated with medium confidence

On Y-axis we have the Application Income. By dividing it in five bins we can assign the proportion of decisions for each bin.

Text

Description automatically generated

Chart, bar chart

Description automatically generated

Based on the above analysis we can see that the bank decision does not necessarily depend on the applicant’s income, which, in turn, contradicts our conclusion from before.

We will further analyze  the relationship between the co-applicant’s income and the loan amount.

Text

Description automatically generated

Chart, bar chart

Description automatically generated

Based on the chart above, we could conclude that if the co-applicant’s income is low, the chances for loan approval are better. However, it is not logical and against the banking principles. The only explanation for it may be the fact that not every applicant has a co-applicant and therefore this number is zero, which adds to the “Low” income column.

Let’s try to combine the applicant’s and the co-applicant’s income to see how these two combined affect the loan status and the total income.

Text

Description automatically generated

Chart, bar chart

Description automatically generated

From the chart above we can observe that when the Total Income is Low, the chances of loan approval are significantly lower than when the Total Income is Average, HIgh or Very High.

In the next step, we will be visualizing the Loan Amount.

Text

Description automatically generated

Chart, bar chart

Description automatically generated

From the chart above we can conclude that the chances of approval are better for Low LoanAmount, compared to Average and High LoanAMount, which supports our hypothesis formulated at the beginning of the project.

Since we will be using Logistic Regression, we need to convert N to zero and yes to one, we will change the 3+ dependent variable to a numeric value of 3. We should also convert the target variable categories to 0 and 1, so we could find their correlation with numerical variables in our data.

Let’s look at the correlation of all numerical variables using the heatmap. The areas of darker color mean higher correlation and vice versa.

Graphical user interface, application, Teams

Description automatically generated

We can observe that the most highly correlated variables are (ApplicantIncome — LoanAmount) and (Credit\_History — Loan\_Status). LoanAmount is correlated to CoapplicantIncome.

**DATA PREPROCESSING**

**Missing Value Imputation**

We want to find the number of missing values in the dataset and impute them with the necessary substitutions. We mostly use the mode for the imputation of all missing values in our model.

Getting a list and the count of the missing values:

Text

Description automatically generated with low confidence

We now use the mode for the categorical variables for imputing:

Text

Description automatically generated

Imputing Numerical Values with the Mode:

Graphical user interface, website

Description automatically generated

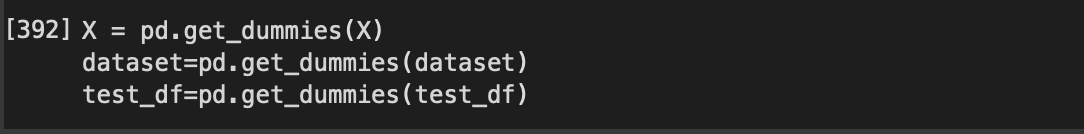
As we can see, after these commands we are left with no missing values in our dataset

Text

Description automatically generated with medium confidence

**BUILDING THE MODEL**

We will start building our model by making dummies for our categorical variables. The category like Gender has 2 values - Male and Female. Using dummy variables helps us to quantify our categorical variables and get better results.

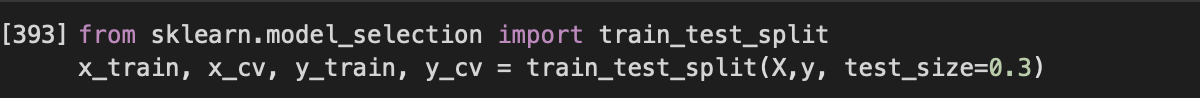


**Logistic Regression**

The Logistic Regression model is contained in the Scikit-learn library. Logistic Regression is a statistical model and a classification algorithm. It is used to predict a binary outcome (1/0, Yes/No, True/False) given a set of independent variables. In the project, only the variables that have a direct impact on the loan eligibility of the applicant are considered (Credit history, Education, Self-employment, Property area). We do not consider all the variables are not considered at the same time since this might lead to the issue of over fitting. Adding more attributes to the model will cause it to learn more particular to the data set rather than generalizing, resulting in overfitting and inaccurate findings for other general situations. We have split our data into training and testing sets. We use 70% of our data as training data and 30% as testing data.

The graph for logistic regression looks like:





Building the logistic regression model for our dataset

Text

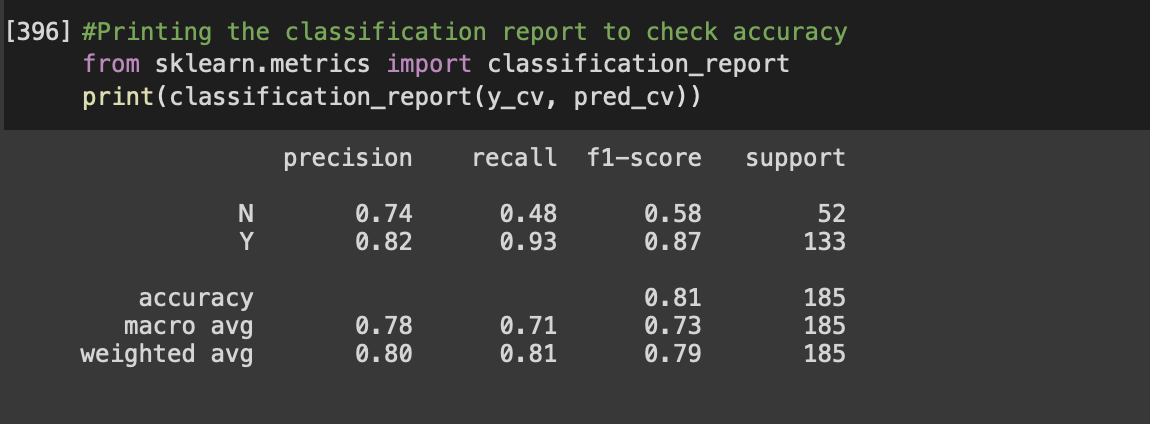
Description automatically generated

We get an accuracy of approximately 80%, which is pretty high without risking overfitting.

Graphical user interface, application

Description automatically generated

We also built the classification report for the Model:



**Performing 10-Fold Cross Validation:**

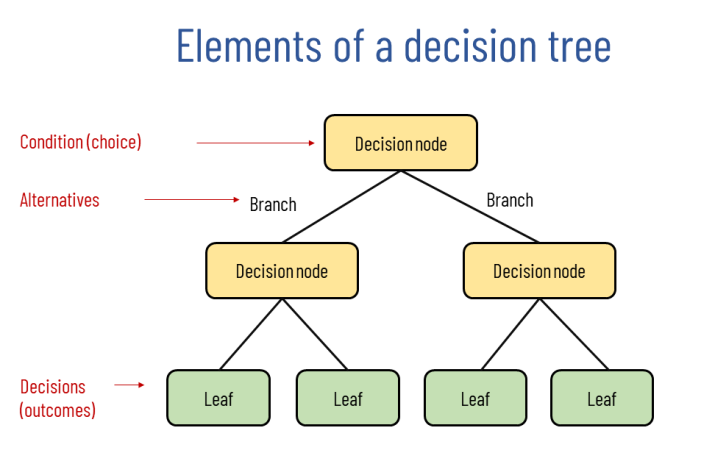
Text

Description automatically generated

**DECISION TREE**

Decision Tree is a versatile algorithm used to solve classification and regression problems. They comprise of several branches, leaf nodes, and root nodes. The algorithm generates a structure like a tree by classifying the instances and utilizing a Recursive Partioning Algorithm. A class label is represented by a leaf node and the branches represent test results. These tests are represented by internal nodes for an attribute.

This is what a decision tree functioning looks like:

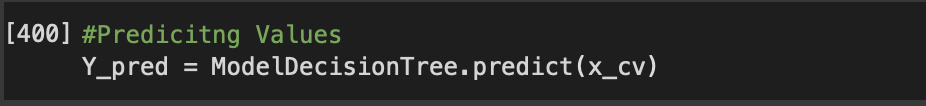


Building a decision Tree Model for our dataset:

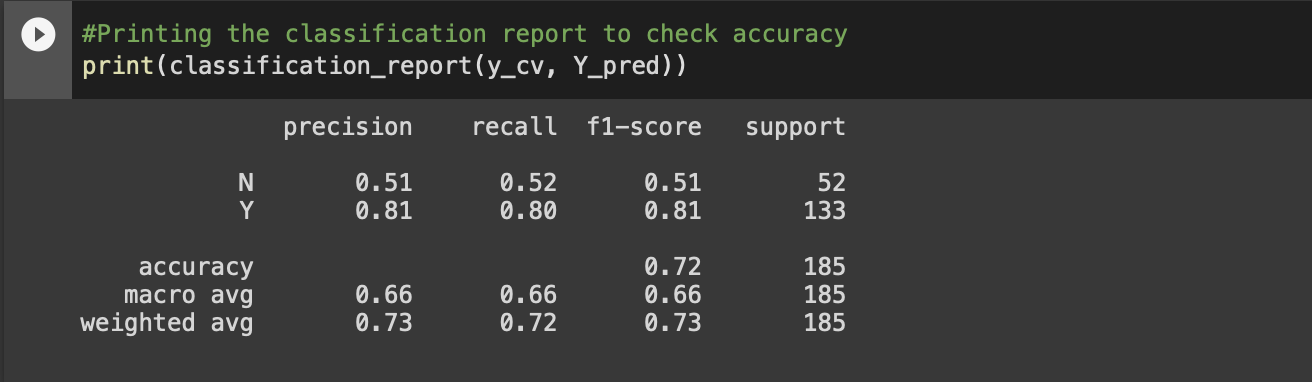
Graphical user interface, text

Description automatically generated

Finding the accuracy



Printing the classification report:



In our analysis, Decision tree proved to function worse than logistic regression. Logistic Regression had an accuracy of approximately 80% whereas Logistic Regression had an accuracy of approximately 72%.

**Conclusion**

This model compares the logistic regression algorithm and the decision tree algorithm and lets the user know which algorithm gives us the better prediction accuracy. In this case it is the logistic regression algorithm. Using this model, we can conclude that the logistic regression version is extremely efficient and gives a higher result. This model will predict whether the applicant is able to repay their loan. This model will significantly reduce the amount of work a banker has to do to approve a loan and it will increase the efficiency of operations of the bank. Machine learning can also be applied to other areas within the banking industry which will make operations much more efficient.

**REFERENCES**

[1] https://www.saedsayad.com/logistic\_regression.html

[2] https://why-change.com/2021/11/13/how-to-create-decision-trees-for-business-rules-analysis/