Professor Mingdi Xin

BANA 273 Machine Learning for Analytics

30 November 2021

Machine Learning Final Project Report   
Classifying Credible Loan Applicants

Team 23:

Kate Zagrebneva

Prashasti Sharma

My Han Mai

Noor Zia

Samira Shafiei

**Table of Contents**

1. Executive Summary
2. Business Idea
3. Data Summary
4. Selected Machine Learning Techniques
   1. Decision Tree
   2. Random Forest
   3. Naive Bayes
   4. Logistic Regressions
5. Process of Analysis
6. Results of Analysis
7. Appendix

Executive Summary

## In this project we focused on analyzing loan approval data and figuring out which model to use in order to correctly classify people who are or are not likely to be approved for a loan. We also wanted to know what is the most important factor that influences loan approval.

We got the dataset that we used to build our models through Kaggle. It has twelve attributes and a loan approval class variable that is binary with values of Y and N. Deciding to capture the model result without preprocessing, we ran Naive Bayes, Decision Tree, Random Forest and Logistic Regression on WEKA to later compare the accuracy results and effectiveness of preprocessing. While the dataset was relatively clean, we still had to do some cleaning and preprocessing before running all the models for optimal results. Preprocessing steps included replacing null values with mean/mode, encoding and normalizing data as well as removing outliers.

For each of the models out of Naive Bayes, Logistic Regression, Decision Tree and Random Forest we ran the model on slightly different preprocessed data to see which preprocessing steps work the best and identify the best fitting model for our dataset and project. Comparing the results, and especially focusing on the accuracy percentage of the models, we identify multiple models that work the best. Out of them we looked into the differences and uniqueness of models to see which model would work the best. The conclusion was made that Naive Bayes model with removed outliers by performing log function works best to identify who would get approved for the loan. Decision Tree without removing the outliers turned out to be the best performing model to identify which attribute has the most influence on loan approval (Loan Amount Term attribute has the highest information gain and, therefore, has the highest influence on loan approval decision).

Business Idea

## Classifying credible loan applicants is an important part of today's world, especially for the banks, mortgage providers, and other loan providers. Loan officers require data analysis in order to determine which loan applications are safe and which pose a risk to their lending companies. For example, the main portion of the financial insitutions’s assets comes directly from the profit earned from the loans it distributes.The primary objective in the many financial environments is to invest their assets in situations where they can be assured of profit or being paid back. Therefore, it’s extremely important to analyze and classify the applicant's information and assign him/her to one of two categories: applicant with good credit or applicant with bad credit.

## Today many financial companies such as banks, mortgage providers, loan companies, et cetera approve loans after an aggressive process of verification and validation but still there is no surety whether the chosen applicant is the most dependable applicant out of all applicants. Through this project we hope to make more accurate predictions on whether or not a particular applicant is credible or not through the validation of attributes by applying machine learning techniques. Loan Prediction is very integral to the activities of lenders and also greatly impacts credible loan applicants.

## The aim of this project is to provide a quick, immediate and easy way to identify and classify the credibility of the loan applicants who are most likely to get a loan approval. It can provide many benefits to the lenders and applicants who are looking to apply for loans. Through this project we can automatically make decisions for each attribute and to what extent each attribute is affecting the credibility of the loan applicant. Moreover, the most influencing attributes can be considered while a new customer approaches the lending firms for a loan and these can identify the potential of the customer. Here by enabling the lending officers to identify the best applicants by using the final application of this project.

## 

## 

## Data Summary

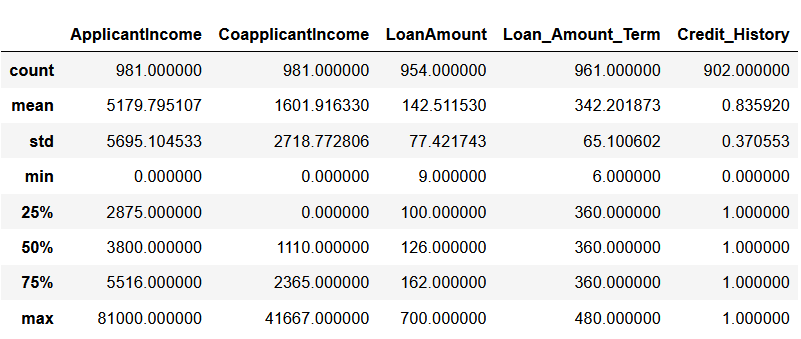
The data we are using for this project is pulled from Vikas Ukani’s ‘Loan Eligible Dataset’ off of the free datasets provided on Kaggle. The dataset consists of two parts: loan-test.csv and loan-train.csv. The dataset is based off of a fictitious loan company, Dream House Finance company, which offers home loans for urban, semi-urban, and rural areas. The data is collected from the company’s home loan applications in order to verify the applicant’s eligibility to be approved for a loan.

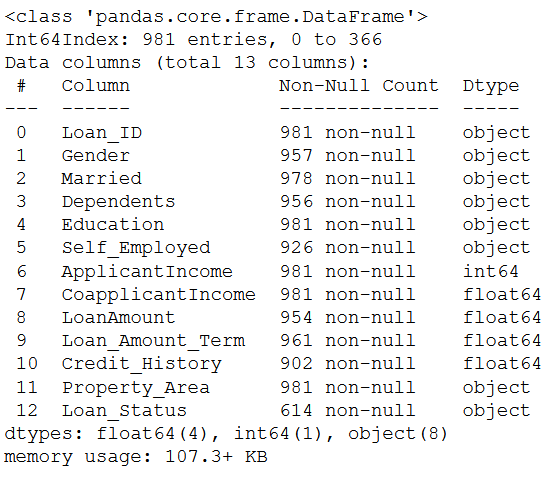
The application form requests information on gender, marriage status, dependents, education level, status of employment (self-employed or not), income level, co-applicant income level, amount of loan requested, length of loan, credit history of applicant, and area in which the property is located (urban, semi-urban, rural). It also provides applicants with a unique ID number that corresponds with the loan application. The loan amount is in units of thousands of US dollars and the loan length is in units of months.

Overall, the data provided is a partial list of 367 customers in the test set and 614 customers in the train set. The goal of using this dataset is to identify specific customer segments in order to predict and target those customers who are more likely to be eligible for loan approval.

Variables:

| Loan ID | Applicant Income |
| --- | --- |
| Gender (F/M) | Co-Applicant Income |
| Married (Y/N) | Loan Amount |
| Dependents (0,1,2,..) | Loan Amount - Term (12- 480) |
| Education (Graduate/Not Graduate) | Credit History (Past loan history: 0,1,..) |
| Self-Employed (Y/N) | Property Area (Urban/Semi Urban) |

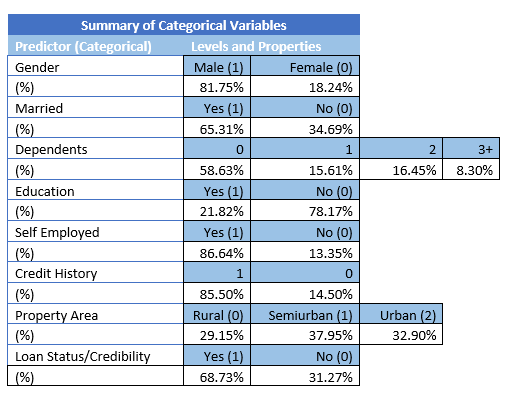
Descriptive statistics of the numerical variables within the dataset:  


Summary of overall dataset:  


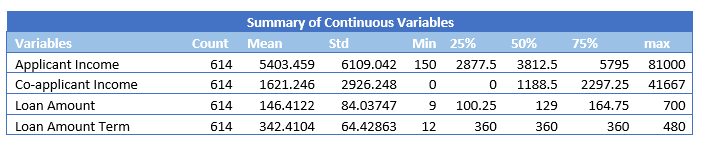
## Observations of Data:

1. Around eighty applicants (8%) are without a credit history.
2. There are null values in the loan amount and loan term column indicating that some applicants submitted an application without completing all the required information.
3. The variable, Loan Amount, has outliers that skews the entire dataset to the right.
4. The general applicant type is a married, male graduate with no children.

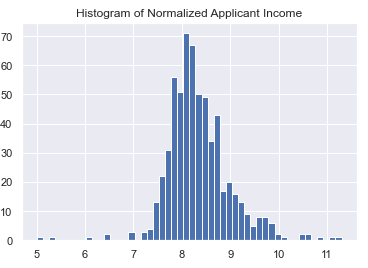
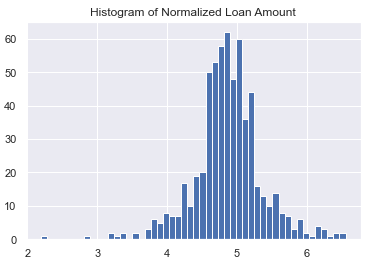
Summary of Categorical variables -

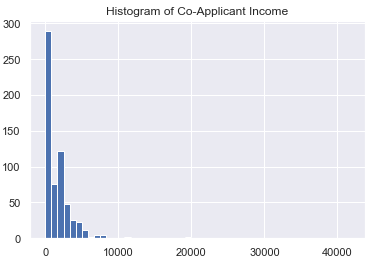
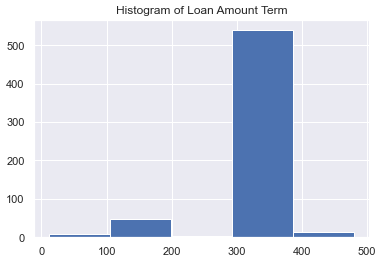


Summary of Continuous variables -

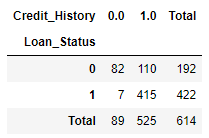
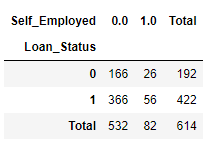
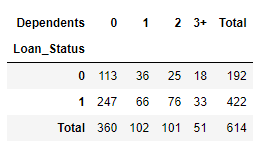
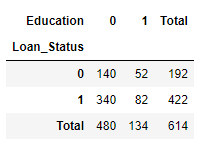
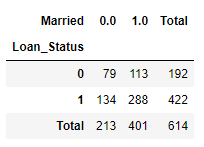
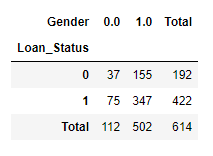


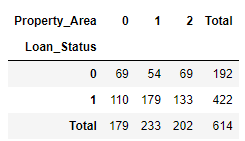
## Distribution of Continuous variables -



Cross Tabulation of predictor variables –





## 

## 

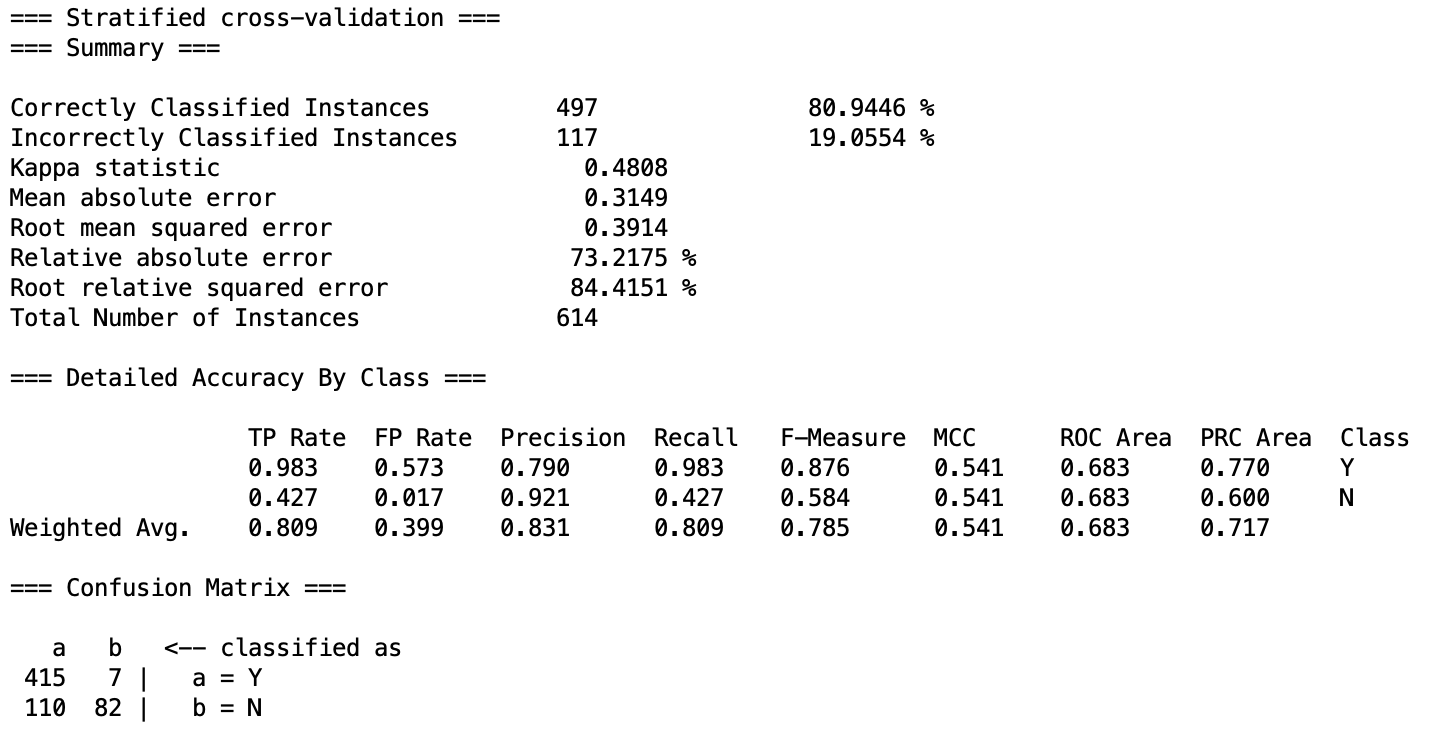
## Selected Machine Learning Techniques

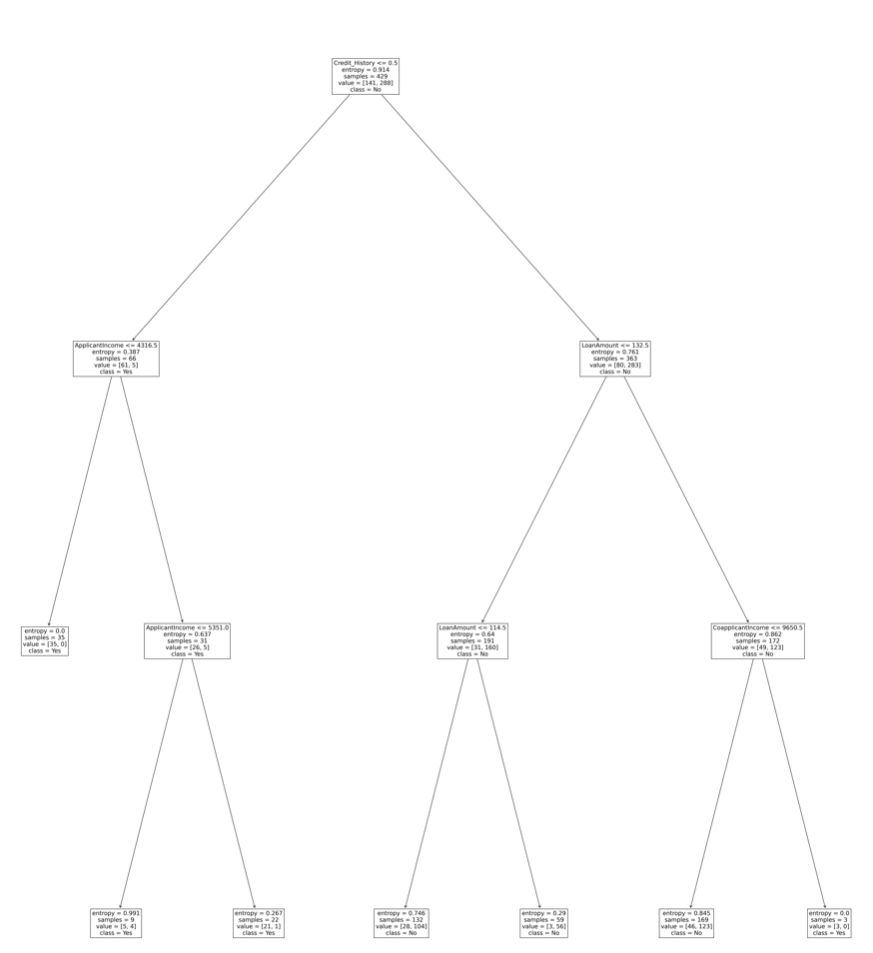
**Decision Tree**

One of the most important goals of the project is to determine which class a new client with specific characteristics will fall into. The provided dataset is structured and supervised. Based on different variables, two classes of interest are defined; approved and not approved. For example, we would like to predict that the loan application for a new client who is male, married, not graduate, self-employed, and with $75,000 income would be approved or not. To do so, the best Model to perform this classification is the Decision Tree classifier. Based on the different attributes, the decision Tree algorithm identifies whether the client loan should be approved or not.

One of the reasons that we chose the Decision Tree technique is that our dataset contains both categorical and continuous variables and the Decision Tree algorithm works well with both types within a dataset. Another reason is that it is easy to understand and interpret and does not require very complex and precise data. Decision Tree does not require much preprocessing to get output.

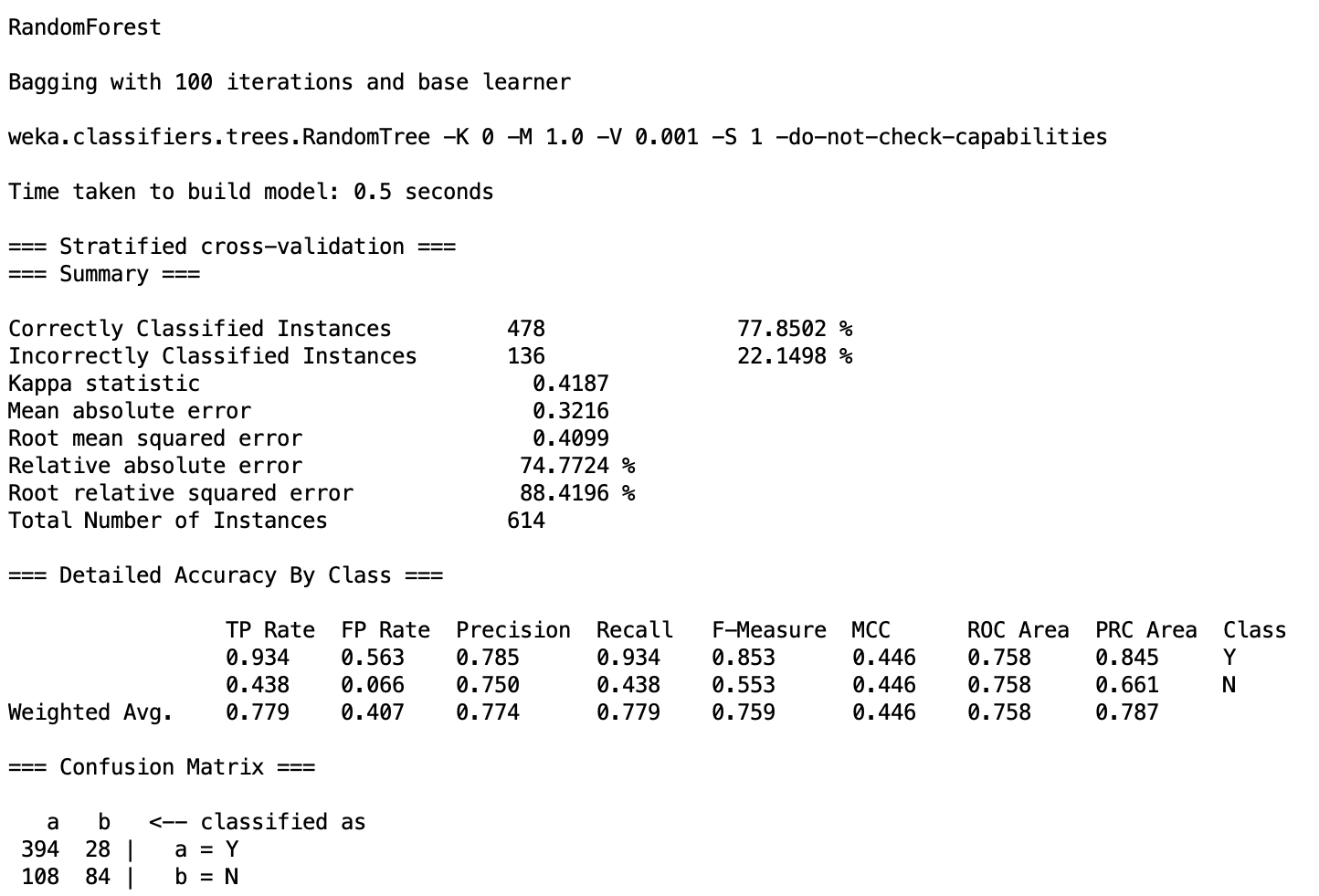
One of the challenges that we faced with the Decision Tree algorithm is selecting the right attributes. We need to choose the best attributes as the root of the different levels of the decision Tree. In order to tackle this problem, we measure the information gain and entropy for each variable. We set the variable with the highest information gain as the root of the decision Tree and split the data that gives the best classification and purer classes.

***1.Raw data- remove Loan-ID***

****

**Random Forest**

The Decision Tree is a simple classification model, however, in terms of prediction accuracy, it is not as robust as other models. We chose the Random Forest classifier to improve the prediction accuracy of the Decision Tree model. After growing some individual decision trees on the training dataset, the Random Forest model combines all the predictions and produces the outcome with the highest accuracy.

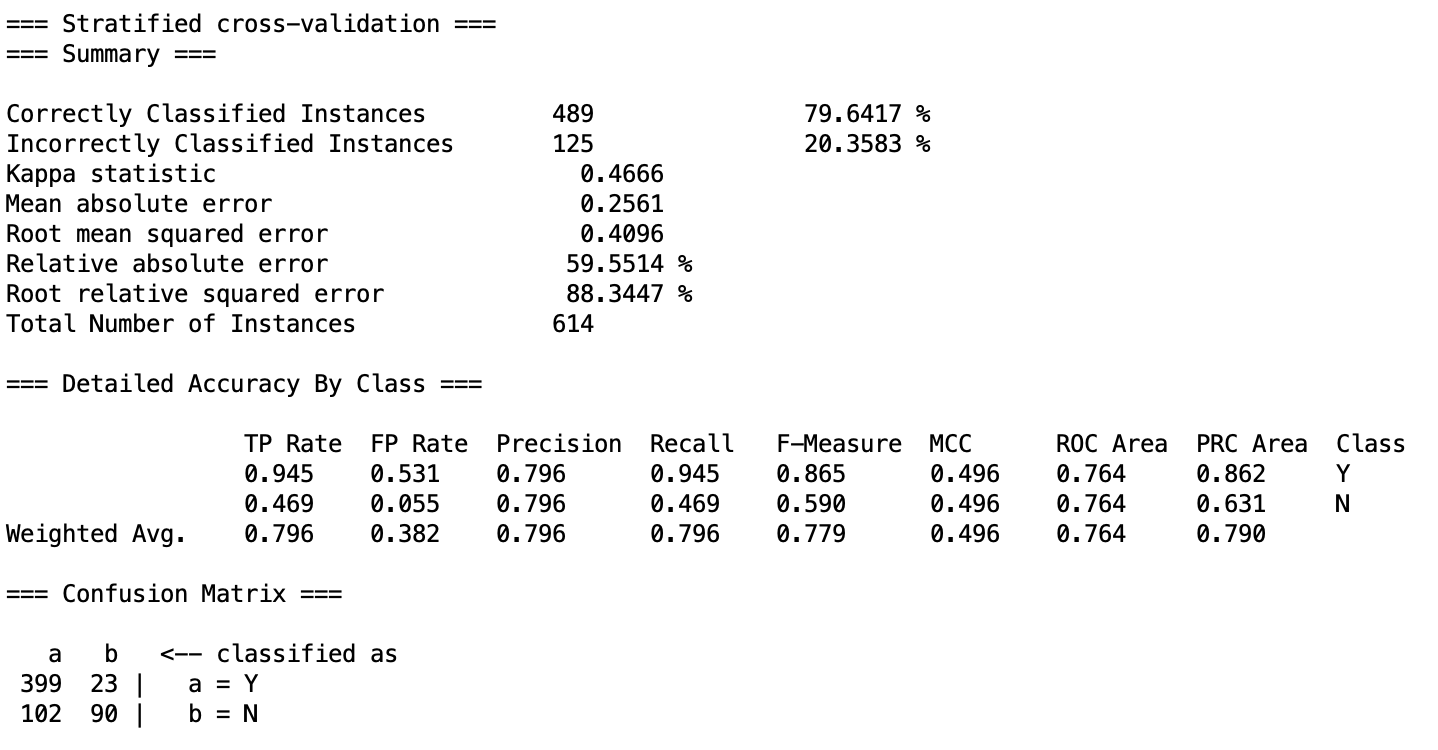
***1.Raw data- remove Loan-ID, Dependents***

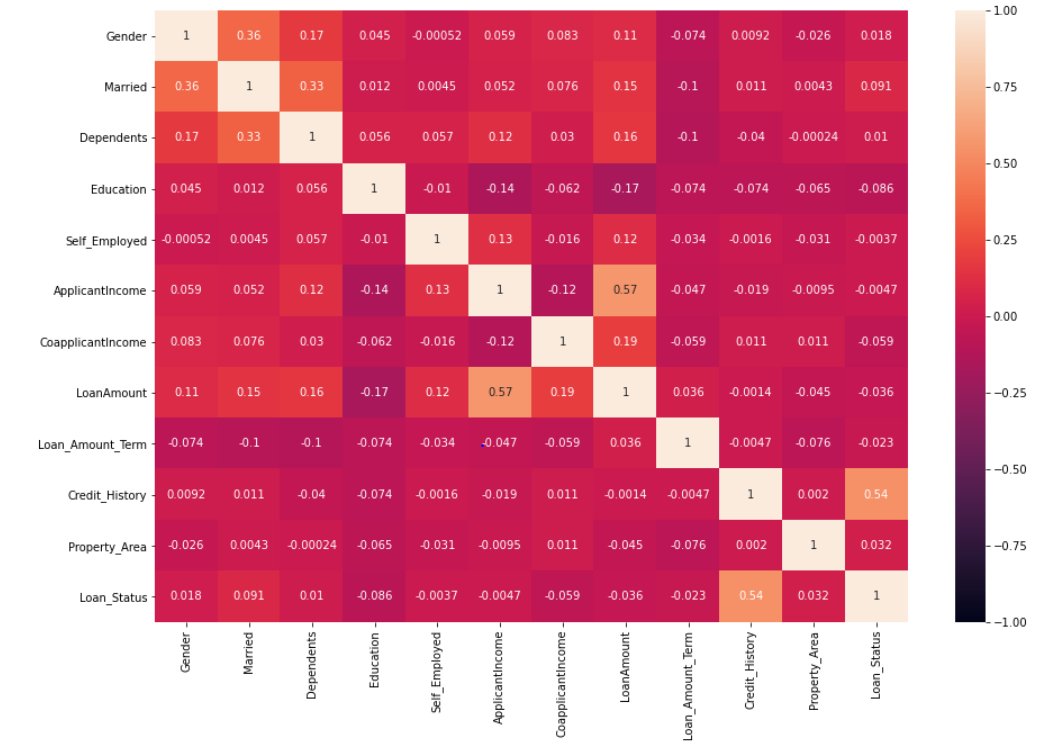
**Naive Bayes**

Naive Bayes is a probabilistic classifier and is one of the most powerful and simple classification techniques to make fast predictions. Naive Bayes performs better than other classifiers when the variables are independent. It makes the assumption that the presence of the predictor variables is unrelated to each other in the dataset. We chose this technique because the attributes in the dataset are independent of each other. To make sure we have chosen the right technique and to have a robust analysis we examined the interaction between the variables in the dataset. To do so, we analyzed the correlations between variables in Python. Also, we created a heatmap visualization in Python to be able to explain the correlation between the attributes. According to the below plot, the dataset’s attributes have either very weak correlations or almost no correlation between themselves. So, the first assumption needed to use the Naive Bayes technique is fulfilled, which is that the predictors are independent.

Moreover, the provided dataset is structured and supervised, and some of the attributes are categorical. We chose the Naive Bayes technique because it works better with categorical variables compared to numerical variables. Subsequently, we changed the numerical variables to categorical to improve the model’s accuracy

***1.Raw data- remove Loan-ID, Dependents***



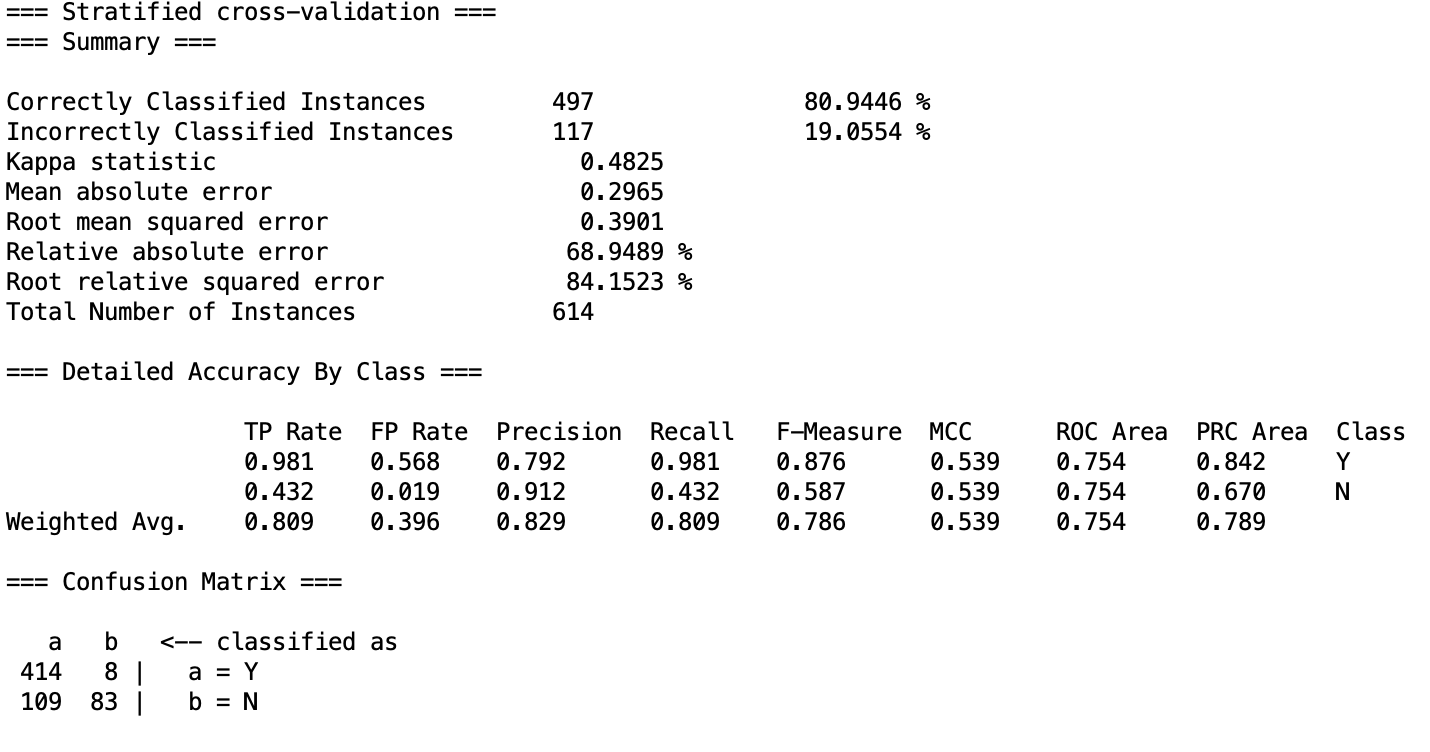
Heat Map of Variable Correlation:

**Logistic Regression**

Another purpose of this project is to determine which variables have the most impact on the loan approval variable. Moreover, we want to produce probability estimates for loan approval rather than classification. We would like to find the probability of each class; the likelihood that a loan will be approved or not. To do so, we decided to use the logistic regression model. Since the provided dataset is small, the logistic regression model is best for prediction as it has better performance. To run logistic regression in Weka we had to remove LoanID and Dependents. We used ‘logistic’ and cross-validation with 10 folds:

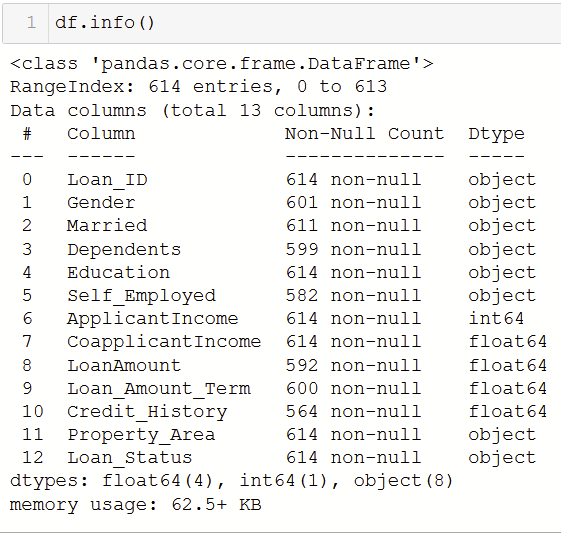
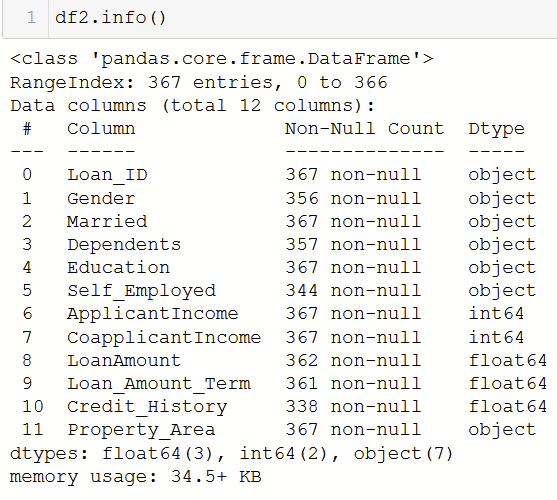
Another reason that we chose this technique is that logistic regression is a robust model to predict the binary variable. In the dataset, the dependent variable is loan approval and it is a binary (approve: yes =1, approve: No=0). Another assumption of the logistic regression which is made by the provided dataset is that independent variables should not be correlated. As mentioned in the Naive Bayes technique, we proved that the attributes have no correlation by conducting a heatmap plot (see above). In order to have a more robust model and increase the overall accuracy, we will remove the variables that are not relevant.

***1.Raw data- remove Loan-ID, Dependents***



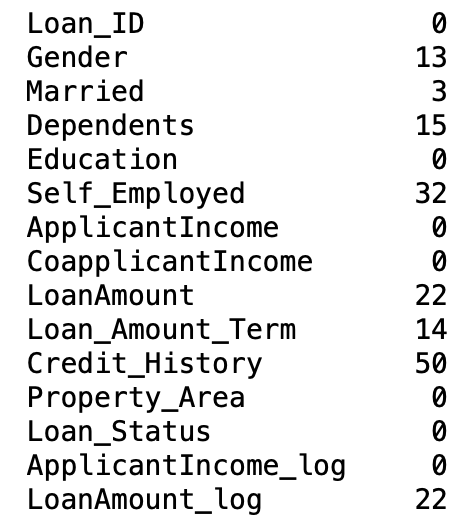
## Process of Analysis

Initially, we gathered our data from kaggle. As part of the data exploration phase, we looked at the shape (rows and columns) and the descriptive summary of the unprocessed data set. We found that our overall dataset was split into a train and test set. The train set had 614 rows with 13 columns. Other than Loan\_ID, ApplicantIncome/CopplicantIncome, Property\_area, and Loan\_status, every other column contains null values. The test set had 367 rows and 12 columns (missing the Loan\_status column). The test set was more complete than the train set, but still had a few null values in most columns. Looking at the descriptive statistics of the numeric columns, we found the values to be within reasonable range for its column.

Once we assessed our data set, we decided to focus on classification. The project goal was to use the attributes to classify whether applicants are credible or not. For this goal, we planned to use naive bayes, logistic regression, and decision tree/random forest.

Before we could run those techniques, we needed to clean and preprocess our data sets. To clean, we checked for duplicates and nulls. Fortunately, there were no duplicates, however there were many null values. For null values, we tried dropping them but found that it lost too much data (171 rows) so instead we replaced categorical null data with mode and numeric null values we replaced with mean. We also dropped columns that were irrelevant: loan\_id. Furthermore, we scaled all the features to be within the same range. We also found several obvious outliers and tried running our models with both outliers and without outliers to see if any information was lost. As outliers skewed the data of Loan Amount, ApplicantIncome, CoapplicantIncome to the right, we chose to normalize the data by combining Applicant Income and Co-Applicant Income into a single variable: Total Income.

Number of Nulls in each column:  


Furthermore, given that we were running different machine learning techniques that required different data types, we had to convert the datasets into the appropriate data types. In Weka, we changed all numeric and string values into nominal to create a fully nominal dataset in order to run Naive Bayes. In Python, we converted nominal attributes into numeric by creating dummy variables. We used the fully numeric dataset in order to run logistic regression. Lastly, we kept the unprocessed (aside from cleaning duplicates and nulls) dataset to run our decision trees and random tree algorithms.

Once the datasets were cleaned and ready, we used two programs to run our selected techniques. In Weka, we used the nominal dataset to run Naive Bayes. We also used Weka to run Random Forest to see if accuracy would improve compared to Decision Trees. In Python, we used the cleaned, original dataset to run decision trees and logistic regressions. For all these methods, we first ran the unprocessed dataset in order to have a baseline to compare our results.

In the end, we chose the model that had the best combination of accuracy and fit in order to run our test data set. Interestingly, Loan Amount Term has the highest information gain. We found that Naive Bayes performs the best, then Decision Tree. For the Decision Tree model, we found the max depth is 3, after which information gain decreases.

## 

## Results of Analysis

**Benchmarks before preprocessing data**

To compare how preprocessing influences the models and its accuracy, we ran the models and documented accuracy and proportion of class variable to compare after doing preprocessing.

**Accuracy**

Decision Tree: 80.9446%

Naiive Bayes: 79.6417%

Logistic Regression: 80.9446%

Random Forest: 77.8502%

**Proportion of class variable**

Decision Tree: Y: 422, N: 192

Naive Bayes: Y: 422, N: 192

Logistic Regression: Y: 422, N: 192

Random Forest: Y: 422, N: 192

We have not done any unique preprocessing to specific models yet and, therefore, class variable proportion is the same for all models.

**General Preprocessing**

The general preprocessing steps that we took for all models included:

* Dropping LOAN\_ID

Loan ID does not bring us any information, since it is a unique consecutive number for each loan. Including this attribute would only make the model performance worse regardless of the type of the model.

* Label Encoding

To make sure that the models would run, we changed all categorical variables to nominal.

* Replacing Null Values with column averages

Due to a low amount of rows and overall dataset that we have, we decided that we need to keep as many values as we can. So instead of removing the rows that had missing values, we calculated the mean or mode for rows (depending on the type of data in the column) and replaced the null values with those.

To compare how the models increase or decrease their accuracy before and after doing preprocessing, we ran all four models on not preprocessed data and included the accuracy in the results for the comparison.

**Creating and Testing models**

Since the data and the classification problem are binary (the class variable is Y/N), we used the four models below:

1. Naive Bayes
2. Logistic Regression
3. Decision Tree
4. Random Forest

Within those models we created models with different preprocessing steps to test the accuracy of the models and see which preprocessing steps work the best with specific models. Specific to the model preprocessing steps included removing outliers by dropping outlying data points using binning or by using log function to normalize and further testing if feature scaling improved model performance.

Due to the small size of the dataset, we did not perform data validation. As for evaluating the models, we used accuracy as a key measure. All the models were built using python package sklearn.

**Naive Bayes**

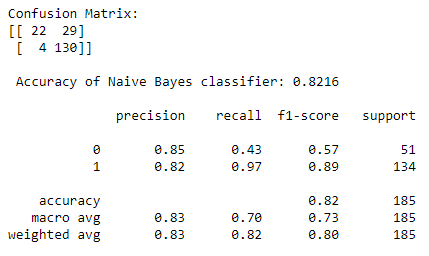
There were a total of six Naive Bayes models that we ran to test which preprocessing steps would work best for the model and will improve its accuracy.

One of the key features of our analysis was understanding the impact of outliers on the dataset.

Our goal was to check how the treatment of outliers would affect the model’s accuracy, as well as check if feature scaling has any significant impact. For treatment of outliers, we used two different techniques:

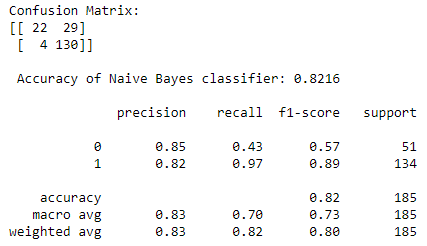
1. Performing binning and removing the bins that had extremely low amounts of values in the bin
2. Performing normalization using logarithmic function.

To begin, we first ran the pre-processed data without removing outliers and without feature scaling on Naive Bayes and here are the results:



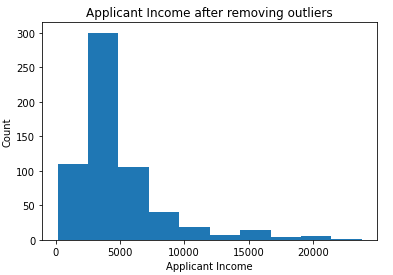
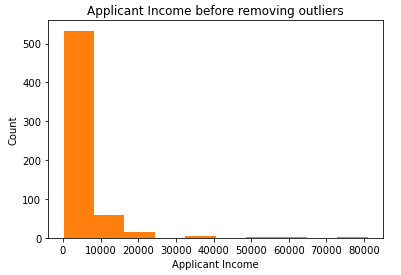
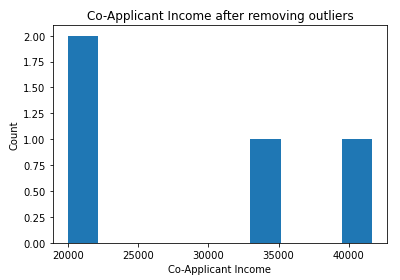
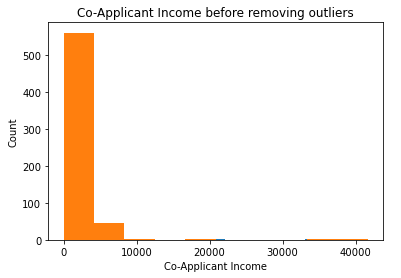
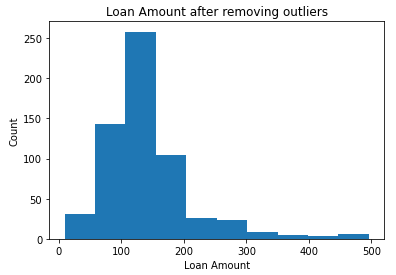
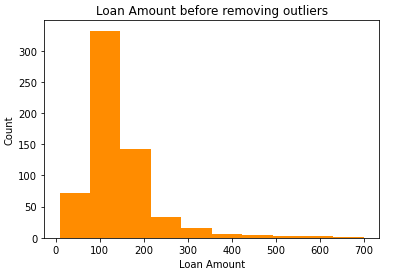
We can see that the model does pretty good job of correctly classifying the customers to whom loan should not be provided (i.e. Loan\_Status = N) and also has an accuracy of 82.16%.

Next we decided to introduce feature scaling in the model and check if the results varied.



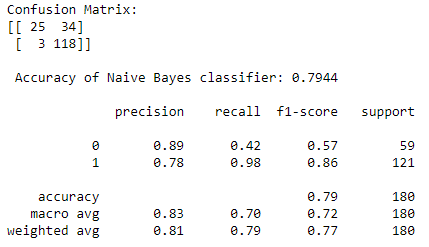
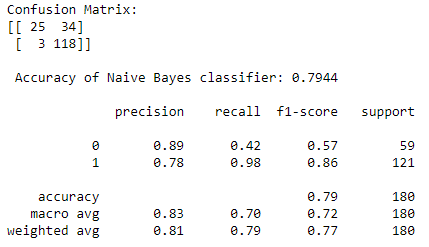
The results remained unchanged. We can imply that feature scaling had no effect on the dataset and model accuracy.

Now, we decided to treat outliers in the dataset. Firstly, we visualized all the columns and saw that three columns - ApplicantIncome, CoApplicantIncome and Loan\_Amount - are skewed to the right and have maximum outliers. We used the binning technique to identify the customers that were resulting in skewness.

1. Remove outlier customers with outlying ApplicantIncome (above $30000) - count 7  
   
2. Remove outlier customers with outlying CoaaplicantIncome (above $15000) - count 4  
   
3. Remove outlier customers with outlying loan amount (above $500) - count 5  
   

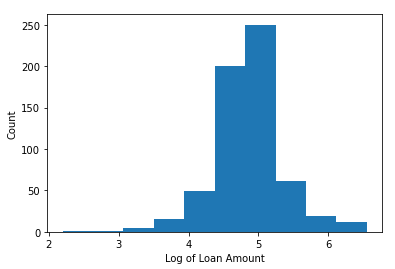
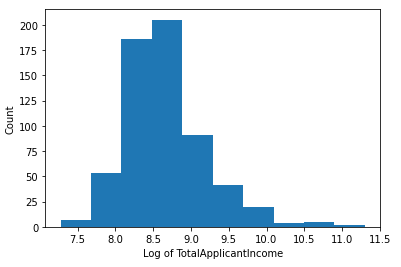
In total, there were a count of 14 unique customer entries that were outliers. We decided to drop these outliers and ran model twice - with and without feature scaling. The results were as follows:

Without feature scaling: With feature scaling:



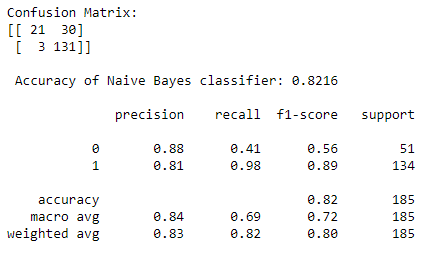
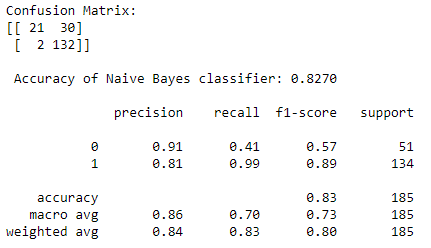
With the above results, we can easily conclude that there exists no impact of feature scaling on the accuracy of Naive Bayes model without outliers (using binning technique).

The second technique used to deal with outliers was normalization using logarithmic function. Again, we intended to perform normalization on the three continuous variable columns discussed previously - ApplicantIncome, CoApplicantIncome and Loan\_Amount. The challenge faced here was that a large number of customers had CoApplicantIncome stated at zero. This prevented us from using the log function directly on the variable. We decided to create a new variable named TotalIncome = ApplicantIncome + CoApplicantIncome, then take a log on the new variable to fix skewed distribution.



Post normalization, the Naive Bayes model with and without feature scaling gave the below results:

Without feature scaling: With feature scaling:



The model’s accuracy reduced from 82.70% to 82.16% on introducing feature scaling. We concluded that feature scaling was ineffective since none of the three modeling variations showed any improvements. The overall accuracy improved after removing outliers by normalizing the skewed variables to 82.70%

Hence, the best performing model out of the six Naive Bayes models was the Naive Bayes model without outliers by doing normalization using log function.

| **Method** | **Accuracy** |
| --- | --- |
| Naïve Bayes before any preprocessing (Weka) | 79.64% |
| Naïve Bayes with Outliers (without Feature Scaling) | 82.16% |
| Naïve Bayes with Outliers (with Feature Scaling) | 82.16% |
| Naïve Bayes without Outliers (Binning) | 79.44% |
| Naïve Bayes without Outliers (Binning + Feature Scaling) | 79.44% |
| Naïve Bayes without Outliers (Normalization using log function) | 82.70% |
| Naïve Bayes without Outliers (Normalization using log + Feature Scaling) | 82.16% |

**Logistic Regression**

For logistic regression we added one more testing compared to Naive Bayes models. In addition to testing how the models perform with and without outliers by doing two different techniques for removing the outliers of removing bins and doing normalization using the log function. We also tested if dropping Insignificant Variables would improve the model.

The results showed that dropping insignificant variables does not improve the model, but removing the outliers actually decreased the accuracy of the model. This happens due to logistic regression working best when there are outliers in the data. Overall, since we have nominal data and not continuous, this model is not the best model for our dataset and analysis.

| **Method** | **Accuracy** |
| --- | --- |
| Logistic Regression before Preprocessing | 80.94% |
| Logistic Regression with Outliers with All Variables | 83.24% |
| Logistic Regression with Outliers after dropping Insignificant Variables | 83.24% |
| Logistic Regression without Outliers by doing Binning with All Variables | 79.44% |
| Logistic Regression without Outliers by doing Binning after dropping Insignificant Variables | 82.70% |
| Logistic Regression without Outliers by doing normalization using log function with All Variables | 82.70% |
| Logistic Regression without Outliers by doing normalization using log function after dropping Insignificant Variables | 76.22% |

**Decision Tree**

Similarly to the previous models we again tested models with and without outliers and two different techniques for removing outliers. The results showed that the model with outliers performs better than the ones without outliers. The reason for this is that the decision tree already handles the outliers on its own while running the model, so when we remove the outliers ourselves, we simply remove data and therefore the model accuracy decreases.

Loan Amount Term attribute has the highest information gain and is the first attribute that the decision tree is using. The Gini index for it is 0.441.

| **Method** | **Accuracy** | **Precision** | **Recall** |
| --- | --- | --- | --- |
| Decision Tree before Preprocessing | 80.94% |  |  |
| Decision Tree with Outliers | 82.70% | 81.48% | 98.51% |
| Decision Tree without Outliers by doing Binning and dropping Bins | 78.33% | 77.33% | 95.87% |
| Decision Tree without Outliers by doing normalization using log function | 81.08% | 81.13% | 96.27% |

**Random Forest**

Since Random Forest is generally very similar to Decision Tree and usually provides more accurate results, we used the same set of models for it. Similarly to Decision Tree, the best performing model is the one with Outliers.

The reason for it is the same as with Decision Tree: the model already takes into account outliers and removing it ourselves only removes the rows and decreases the amount of data to create the model on, therefore, decreasing its accuracy.

Even though Random Forest usually works better than Decision Tree in a lot of cases, out data works better with Decision Tree as shown by the accuracy comparison.

| **Method** | **Accuracy** | **Precision** | **Recall** |
| --- | --- | --- | --- |
| Random Forest without Preprocessing | 77.85% |  |  |
| Random Forest with Outliers | 80.00% | 80.54% | 93.75% |
| Random Forest without Outliers by dropping rows | 77.78% | 79.87% | 92.25% |
| Random Forest without Outliers by doing normalization using log function | 78.92% | 79.87% | 92.97% |

**Analysing Results**

Choosing the best model to use for this dataset in order to identify who would get approved for loan and which factors are the most influential for the decision we have to compare the accuracy results for all models, but also to consider how the models work and if the process of model creation works for our dataset.

Simply comparing the accuracy results of the top performing models in each of the Naive Bayes, Logistic Regression, Decision Tree and Random Forest models, we get the following results in the descending order:

| **Model** | **Accuracy** |
| --- | --- |
| Logistic Regression with Outliers | 83.24% |
| Logistic Regression with Outliers | 82.70% |
| Naïve Bayes without Outliers by doing normalization using log function | 82.16% |
| Random Forest with Outliers | 80.00% |

We can see that Logistic Regression wins when comparing accuracy results, however, it is not the best fitting model due to the type of data that we have. Logistic Regression got rid of almost all variables in order to get that high of an accuracy. It also takes into account the outliers and is skewed.

Decision Tree and Naive Bayes both have similar accuracy, but due to Decision Tree tendency to overfit, we decided that Naive Bayes model with removed outliers by performing log function works best to identify who would get approved for the loan. Decision Tree is the best performing model to identify which attribute has the most influence on loan approval (Loan Amount Term attribute has the highest information gain).

### 

### Appendix

Visualization of Initial (Un-preprocessed)Datasets:  
