# LOAN ELIGIBILITY DATA ANALYSIS & MACHINE LEARNING MODEL IMPLEMENTATION

# TABLE OF CONTENTS

TABLE OF CONTENTS........................................1

LIST OF FIGURES........................................2

LIST OF TABLES........................................3

1. Project Overview........................................4

1.1. Problem Statement........................................4

1.2. Proposed Methodology - Approach........................................4

2. Data Manipulation........................................5

2.1. Data Preprocessing........................................5

2.1.1. Data Encoding........................................6

2.1.2. Data Imputation........................................6

3. Data Analysis........................................7

3.1. Discrete data analysis........................................7

3.2. Continuous data analysis........................................12

3.3. Data Analysis – Summary........................................18

4. Outlier detection & elimination........................................19

4.1. Local Outlier Factor Method........................................19

5. Data Normalization – Standardization........................................19

5.1 Data Normalization........................................19

5.2. Data Standardization........................................19

6. Machine Learning........................................20

6.1. Machine Learning models........................................20

6.1.1. Logistic Regression........................................20

6.1.2. Decision Trees........................................21

6.1.3. KNN (K-Nearest Neighbor)........................................22

6.1.4. Random Forest Classifier........................................22

6.2. Machine Learning models results........................................23

6.2.1. Logistic Regression........................................23

6.2.2. Decision Trees........................................25

6.2.3. KNN (K-Nearest Neighbor)........................................27

6.2.4. Random Forest Classification........................................29

7. Conclusion........................................31

# LIST OF FIGURES

**Figure 1.** Dataset sample representation ........ 6
**Figure 2.** Encoded Dataset ........ 6
**Figure 3.** Gender Percentage Distribution (Left), Gender Loan Eligibility Distribution (Right)........ 8
**Figure 4.** Marital Status Percentage Distribution (Left), Loan Eligibility based on marital status (Right) .... 8
**Figure 5.** Number of dependents percentage distribution (Left), Loan Eligibility based on number of dependents (Right) ........ 9
**Figure 6.** Education percentage distribution (Left), Loan status based on education (Right)........ 9
**Figure 7.** Self-employed percent distribution (Left), Loan Status based on self-employment (Right)....... 10
**Figure 8.** Credit history percent distribution (Left), Loan status based on credit history (Right) ........ 10
**Figure 9.** Property Area percent distribution (Left), Loan status based on property type (Right) ........ 11
**Figure 10.** Loan status percent distribution........ 11
**Figure 11.** Pair plot analyzing continuous features ........ 13
**Figure 12.** Applicant Income distribution (Left), Co-applicant Income distribution (Right) ........ 14
**Figure 13.** Loan Amount Distribution (Left), Loan Amount Term Distribution (Right) ........ 14
**Figure 14.** Applicant’s data ........ 15
**Figure 15.** Income required to purchase properties........ 15
**Figure 16.** Loan Amount needed to purchase properties........ 15
**Figure 17.** Correlation matrix........ 16
**Figure 18.** Normalization Formula ........ 19
**Figure 19.** Standardization Formula ........ 20
**Figure 20.** Sigmoid Function ........ 21
**Figure 21.** Decision Tree Classifier Structure ........ 21
**Figure 22.** KNN Classification ........ 22
**Figure 23.** Random Forest Classifier Structure ........ 22
**Figure 24.** Logistic Regression model learning curve........ 23
**Figure 25.** Logistic regression classification report........ 23
**Figure 26.** Logistic regression model prediction on cross-validation data ........ 24
**Figure 27.** Logistic regression model prediction on testing data........ 24
**Figure 28.** Decision Tree model Learning Curve ........ 25
**Figure 29.** Decision Tree model classification report ........ 25
**Figure 30.** Decision Tree model predictions made on cross-validation data ........ 26
**Figure 31.** Decision Tree model predictions made on testing data........ 26
**Figure 32.** Decision Tree model ........ 27
**Figure 33.** KNN model learning curve ........ 28
**Figure 34.** KNN model classification report ........ 28
**Figure 35.** KNN model predictions made on cross-validation data ........ 28
**Figure 36.** KNN model predictions made on testing data ........ 29
**Figure 37.** Random Forest Classification model learning curve ........ 29
**Figure 38.** Random Forest Classification model classification report........ 30
**Figure 39.** Random Forest Classification model predictions made on cross-validation data........ 30
**Figure 40.** Random Forest Classification model predictions made on testing data ........ 30
**Figure 41.** Deployed Models Scores ........ 31

# LIST OF TABLES

**Table 1.** Dataset Feature Description ........................................ 4
**Table 2.** Encoded data representation ........................................ 6
**Table 3.** Features with missing values ........................................ 7
**Table 4.** Correlation Matrix Deductions made of features with correlation score >=0.1 .......................... 16

# 1. Project Overview

A housing finance company offers home loans to applicants that meet all the requirements imposed by the finance firm. At the moment, customers fill a detailed online application form. Based on the information provided by the customer, the finance firm within the company has to make a decision on whether this customer will be granted a loan to purchase a house or not. Overall, the process seems time consuming and requires human intervention in each applicant analysis case for a final decision to me made. As such, the company is planning to automate this process by understanding the relation between the collected information and the probability that the customer will be paying back his mortgage on a monthly basis.

## 1.1. Problem Statement

The objective is to design a machine learning model that is capable of predicting whether the applicant is capable of paying back their mortgage. Have to analyzing multiple attributes which are shown below in **Table 1**, shows that deriving a proper relation between the data is of the upmost importance, since redundant features could be eliminated by understanding such relationships, and by understanding relationships between features, a model that is highly biased towards insignificant features could be improved by eliminating such biases. As such, the model developed should take those aspects into consideration.

**Table 1.** Dataset Feature Description

| Feature | Description |
|---|---|
| Loan_ID | Unique Loan ID |
| Gender | Male - Female |
| Married | Customer married (Yes – No) |
| Dependents | Number of Dependents (0, 1, 2, 3+) |
| Education | Education (Graduate – Not Graduate) |
| Self_Employed | Self-employed (Yes – No) |
| ApplicantIncome | Customer Income |
| CoapplicantIncome | Co-applicant Income |
| LoanAmount | Loan amount in thousands |
| Loan_Amount_Term | Loan amount term in months |
| Credit_History | Has a good credit score (0 – 1) |
| Property_Area | Rural – Semiurban - Urban |
| Loan_Status | Eligible (Y – N) |

## 1.2. Proposed Methodology - Approach

To design such model which is capable of predicting eligibility of loan the following questions should be taken into consideration and check whether the data provided shows any insights that might help in analyzing the case.

- Is data biased towards gender?
- Are married applicants more prone to being accepted for a loan?
- Is number of dependents an important feature that affects the final decision made?
- Does level of education play a significant role in make such decision?
- Do self-employed applicants have a higher chance of being accepted?
- Is final decision made based on preference of property type?
- Is Credit History necessary?

Analyzing the dataset which was provided, should reveal what type of applicants are applying for loans and reveal whether the questions asked turn out to be an important decision-making factor.

Moreover, knowing that this is a classification problem more precisely binary classification, the following supervised learning algorithms will be deployed to perform such task.

- Logistic Regression
- Decision Trees
- KNN (K-Nearest Neighbor)
- Random Forest Classifier

Each of the developed models at the end will be compared to each other. Within the first trial, dataset as it is will be used to train such models, but on the second trial, insignificant features will be removed and dataset will be trained again on such models. Thus, by comparing such models generated on two trials, insights could be generated regarding the models, and indicate how well each model performs.

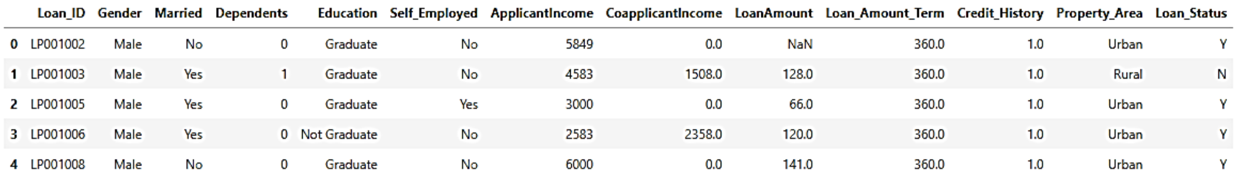
# 2. Data Manipulation

Within this section, the goal is to describe the process of how a dataset with multiple objects of different types will be preprocessed, encoded, and imputated in such a way that when deploying such model datasets can be used easily and interpreted by models in an effective way. The following procedure will be used to manipulate such dataset it its first phase:

- Data Preprocessing
    - Encode non numerical data into numerical encodings.
    - Replace missing and invalid entries with either their feature vector mean, median, or mode. In other words, use data imputation methods to replace invalid entries.

## 2.1. Data Preprocessing

For the proposed dataset which will be used to train the models, the data consists of 614 entries with 13 attributes. The given data has to be preprocessed in such a way that either all values are within the 0 to 1 range or values are centered around the mean with minimum variance. **Figure 1** shown below is a sample representation of the data that will be used to train such models.

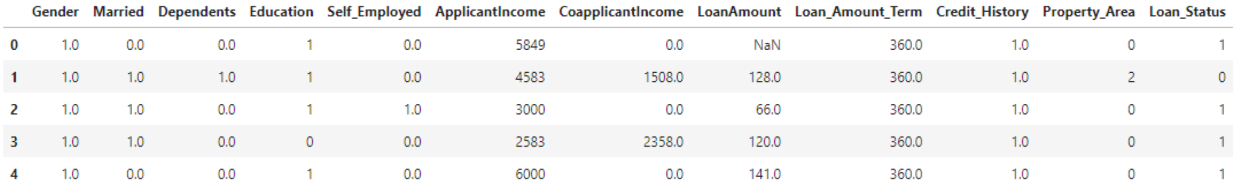


| | Loan_ID | Gender | Married | Dependents | Education | Self_Employed | ApplicantIncome | CoapplicantIncome | LoanAmount | Loan_Amount_Term | Credit_History | Property_Area | Loan_Status |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | LP001002 | Male | No | 0 | Graduate | No | 5849 | 0.0 | NaN | 360.0 | 1.0 | Urban | Y |
| 1 | LP001003 | Male | Yes | 1 | Graduate | No | 4583 | 1508.0 | 128.0 | 360.0 | 1.0 | Rural | N |
| 2 | LP001005 | Male | Yes | 0 | Graduate | Yes | 3000 | 0.0 | 66.0 | 360.0 | 1.0 | Urban | Y |
| 3 | LP001006 | Male | Yes | 0 | Not Graduate | No | 2583 | 2358.0 | 120.0 | 360.0 | 1.0 | Urban | Y |
| 4 | LP001008 | Male | No | 0 | Graduate | No | 6000 | 0.0 | 141.0 | 360.0 | 1.0 | Urban | Y |

**Figure 1.** Dataset sample representation

### 2.1.1. Data Encoding

In this step, binary and categorical data should be numerically encoded in such a way that the models can easily interpret what each feature label belongs to. **Table 2** shown below is a representation of how features will be encoded and as for **Figure 2** shown below, is a sample representation of the dataset which has been numerically encoded. Note that the Loan_ID Attribute has been dropped, simply because it serves no purpose and won't affect predictions.



| | Gender | Married | Dependents | Education | Self_Employed | ApplicantIncome | CoapplicantIncome | LoanAmount | Loan_Amount_Term | Credit_History | Property_Area | Loan_Status |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1.0 | 0.0 | 0.0 | 1 | 0.0 | 5849 | 0.0 | NaN | 360.0 | 1.0 | 0 | 1 |
| 1 | 1.0 | 1.0 | 1.0 | 1 | 0.0 | 4583 | 1508.0 | 128.0 | 360.0 | 1.0 | 2 | 0 |
| 2 | 1.0 | 1.0 | 0.0 | 1 | 1.0 | 3000 | 0.0 | 66.0 | 360.0 | 1.0 | 0 | 1 |
| 3 | 1.0 | 1.0 | 0.0 | 0 | 0.0 | 2583 | 2358.0 | 120.0 | 360.0 | 1.0 | 0 | 1 |
| 4 | 1.0 | 0.0 | 0.0 | 1 | 0.0 | 6000 | 0.0 | 141.0 | 360.0 | 1.0 | 0 | 1 |

**Figure 2.** Encoded Dataset

**Table 2.** Encoded data representation

| Feature | Actual Label | Encoded Representation |
|---|---|---|
| Gender | Female, Male | Female = 0, Male = 1 |
| Married | Yes, No | No = 0, Yes = 1 |
| Dependents | 0, 1, 2, 3+ | 0 = 0, 1 = 1, 2 = 2, 3+ = 3 |
| Education | Not Graduated, Graduated | Not Graduated = 0, Graduated = 1 |
| Self_Employed | Yes, No | No = 0, Yes = 1 |
| Property_Area | Rural, Semiurban, Urban | Urban = 0, Semiurban = 1, Rural = 2 |
| Loan_Status | Y, N | N = 0, Y = 1 |

### 2.1.2. Data Imputation

The objective of performing data imputation is to replace invalid data entries with an appropriate value that best fits the scenario. Techniques involved make use of replacing missing / invalid entries with either their mean, median, or mode feature vector. In this case, for categorical values invalid entries will be replaced with their mode, and as for continuous values will be replaced with their median.

Upon analyzing the data, **Table 3** shows which features have missing values.

**Table 3.** Features with missing values

| Gender | Married | Dependents | Self_Employed | LoanAmount | Loan_Amount_Term | Credit_History |
|---|---|---|---|---|---|---|
| 13 missing | 3 missing | 15 missing | 32 missing | 22 missing | 14 missing | 50 missing |

Having encoded and adjusted the datasets content for appropriate analytic usage, in the upcoming section a detailed analysis will be performed where each feature will be examined and insights will be developed.

# 3. Data Analysis

In this section features will be analyzed and categorized into 2 groups, discrete and continuous data. As such, after analyzing insights will be developed and other aspects will be discussed.

- Discrete Features
    - Gender
    - Married
    - Dependents
    - Education
    - Self_Employed
    - Credit_History
    - Property_Area
    - Loan_Status

- Continuous Features
    - ApplicantIncome
    - CoapplicantIncome
    - LoanAmount
    - Loan_Amount_Term

## 3.1. Discrete data analysis

To begin, **Figure 3** represents the percentage distribution of applicants based on their gender which is shown on the left, and on the right-side percentage distribution of accepted and rejected loans based on gender. Based on this simple observation, majority of applicants are male, and as for when it comes to eligibility male gender holds the largest percentage distribution of accepted applicants with female accepted a small minority. Thus, from this simple observation we can see that data is imbalanced and majority of accepted applicants are male.

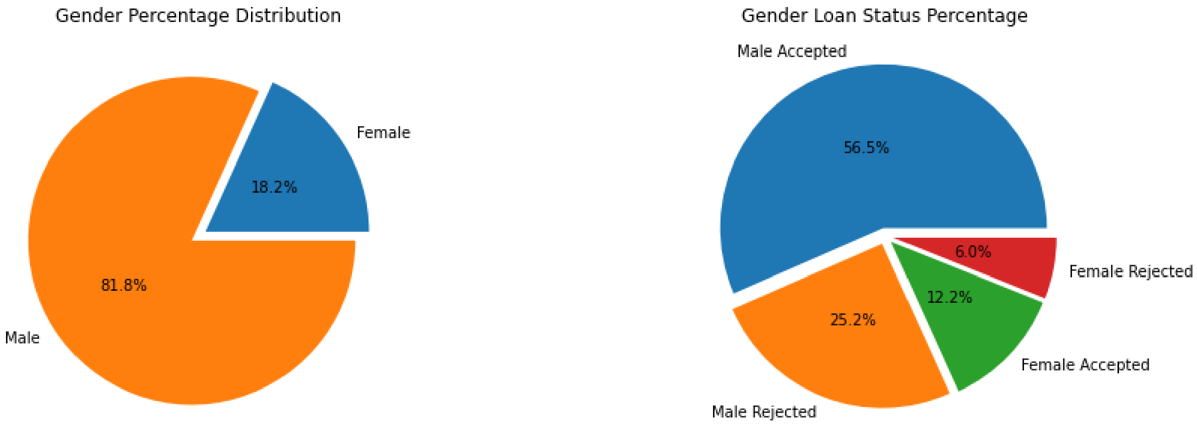


**Figure 3.** Gender Percentage Distribution (Left), Gender Loan Eligibility Distribution (Right)

Moving on, **Figure 4** represents the percentage distribution of married applicants on the left, and as well as percentage distribution of married applicants based on their loan status on the right. Based on the visualized observation, majority of applicants are married and the largest percentage distribution which is 46.9% is dominated by applicants that are married where their loan status is accepted. Following up, the second largest percentage distribution are non-married applicants that hold an acceptance ratio of 21.8%. Overall, based on the given data, married applicants have a better chance for being eligible for a loan.

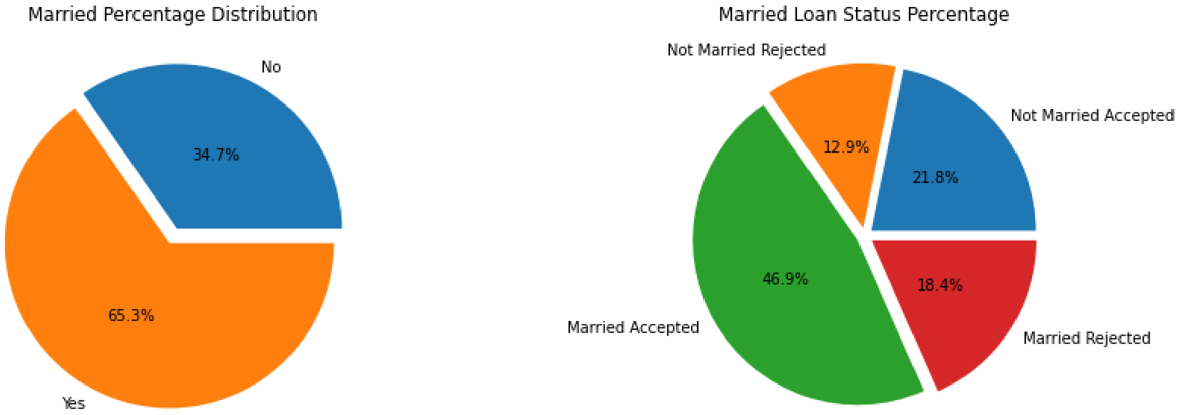


**Figure 4.** Marital Status Percentage Distribution (Left), Loan Eligibility based on marital status (Right)

Another attribute which is the number of dependents also might favor such decision, and **Figure 5** represents the percentage distribution of number of dependents and as well as percentage distribution of loan status with respect to the number of dependents. Based on the observed data, applicants with no dependents have a higher chance of being granted a loan.

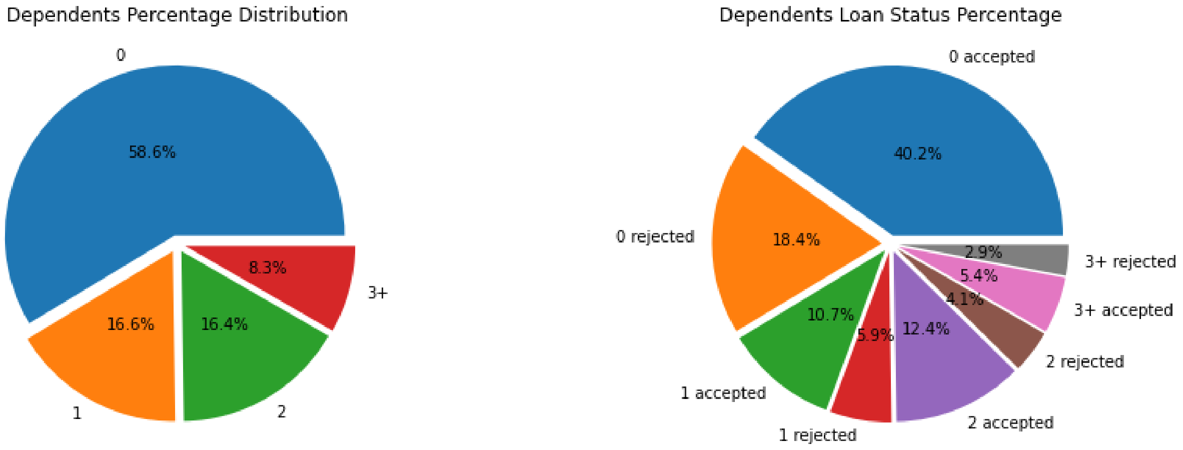


**Figure 5.** Number of dependents percentage distribution (Left), Loan Eligibility based on number of dependents (Right)

Adding to that another attribute which is the applicant's education status, whether the applicant has graduated or hasn't (either still studying or has no educational background). **Figure 6** shows the percentage distribution of applicants that have graduated and haven't graduated. Data visualized shows that majority of applicants have graduated covering 78.22% of the entire dataset population and 21.8% of the remaining applicants have not graduated. Moreover, majority of applicants that have graduated have a better chance of being eligible for a loan. But this isn't a governing factor that would affect such decision since even some graduate applicants are not eligible and other contributing factors would tend to shift such decision either in the positive or negative side.

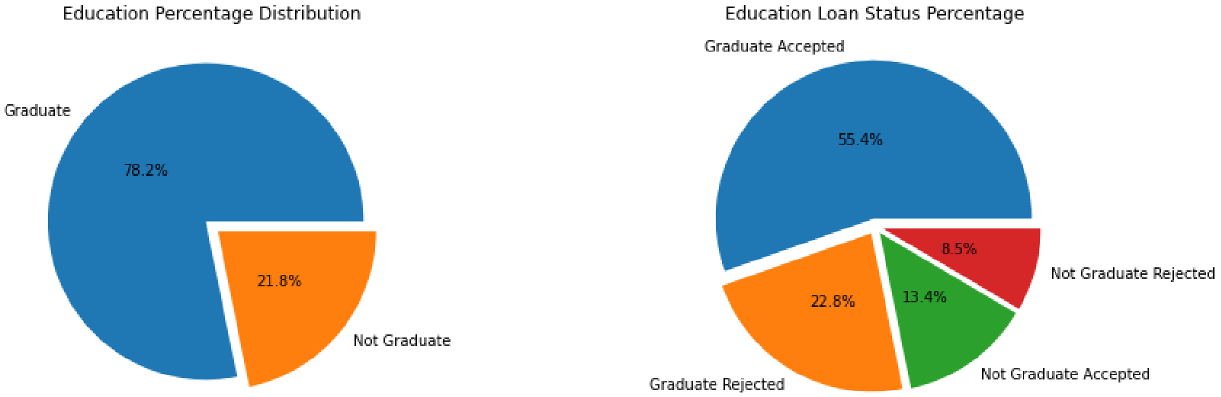


**Figure 6.** Education percentage distribution (Left), Loan status based on education (Right)

Moreover, anther attribute is self-employment, Figure 7 shows the percentage distribution of applicants that are self-employed and loan status based on self-employment. Overall based on the graphical representation shown below, self-employed applicants have a very low chance of being granted a loan simply due to the fact that applicants might not have a steady stream of income on a monthly basis that would effectively cover their expenses and contributions towards their mortgages. As such data reveals that self-employed applicants have a higher chance of rejection.

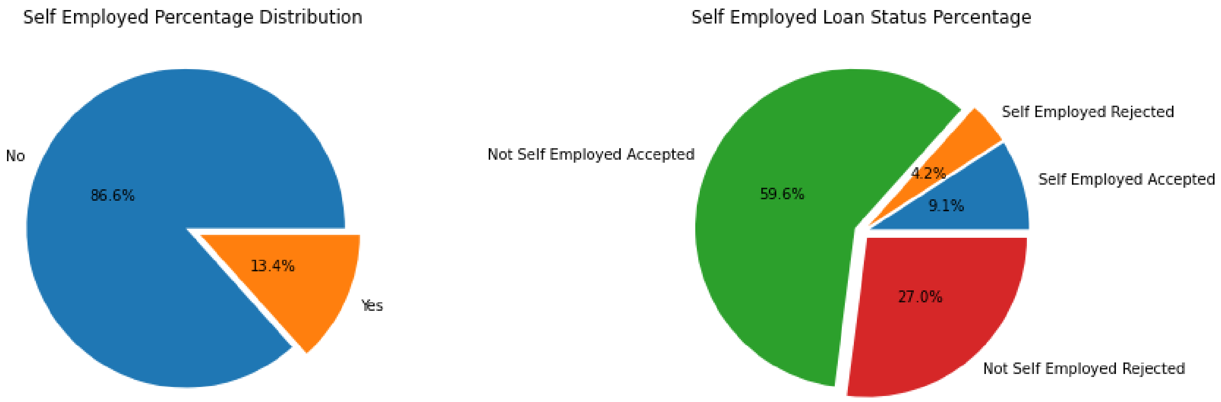


**Figure 7.** Self-employed percent distribution (Left), Loan Status based on self-employment (Right)

Another feature worth analyzing is credit history, since for a person to be eligible for a loan most financial institutes look at the applicant credit history and for first time applicants that have no previous history it is common that they are not eligible if they don't meet other requirements. Figure 8 shows the percent distribution of applicants with and without credit history, and as well as loan status based on credit history. As observed 85.5% of applicants have credit history, majority of applicants have also been granted a loan since this attribute reveals previous data when it comes to mortgage payments and other finances.

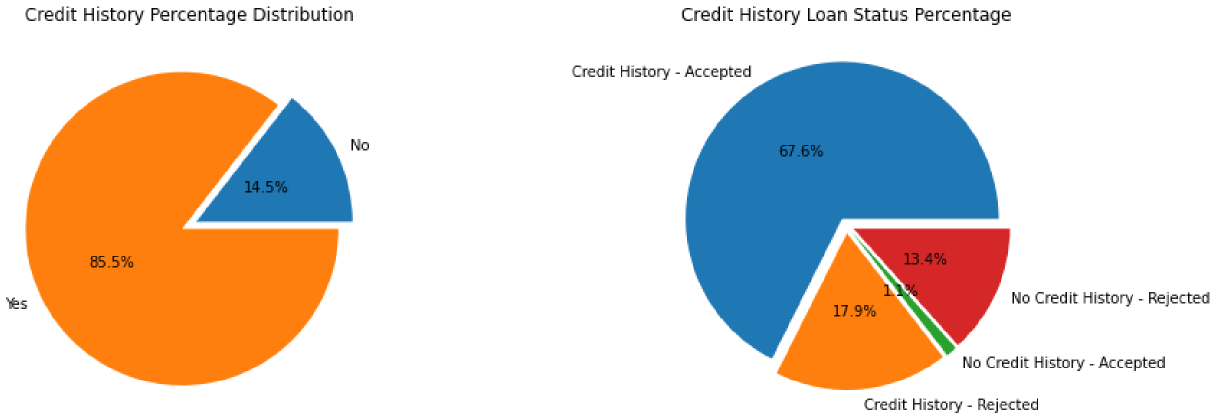


**Figure 8.** Credit history percent distribution (Left), Loan status based on credit history (Right)

Another feature which is property area, applicants that are purchasing properties most of them choose either to buy urban, semiurban, or rural property areas. **Figure 9** shows an almost equal distribution amongst those 3 types of properties. Moreover, applicants that choose semiurban areas are more favorable when compared to others. Data shows that this attribute does have a major impact on the final decision since all accepted applicants are equally distributed along other types of properties and such feature isn't of massive importance for the final decision to be affected.

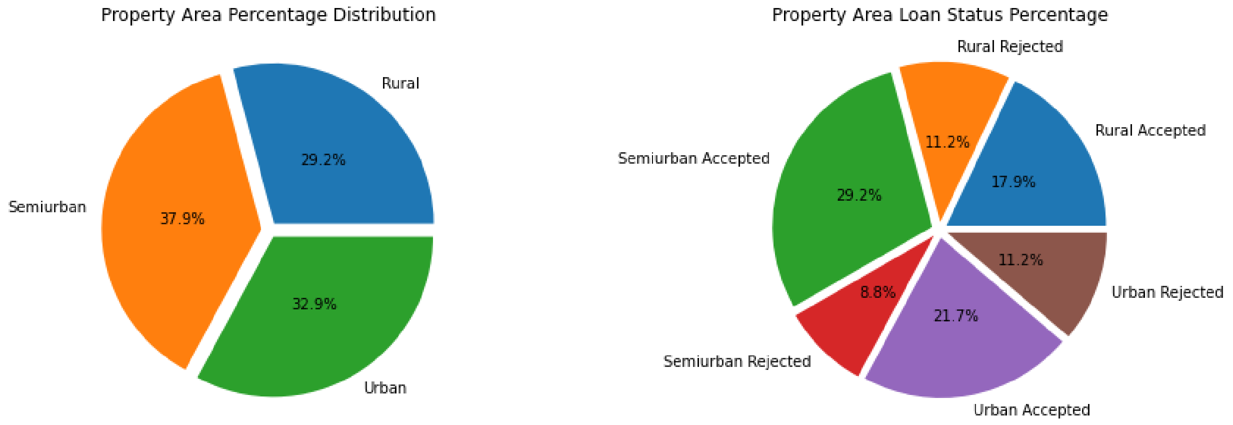


**Figure 9.** Property Area percent distribution (Left), Loan status based on property type (Right)

Finally, having analyzed all discrete data the outcome of such model is either an applicant is eligible or not eligible for a loan and as such **Figure 10** shown below displays the percent distribution of applicants that are eligible and applicants that are not eligible. As such, majority of applicants are eligible, but have 614 applicants where 68.7% have been accepted and a small percentage being rejected reveals that data is imbalanced and this could lead to the model having a bias to predict most outcomes as eligible.

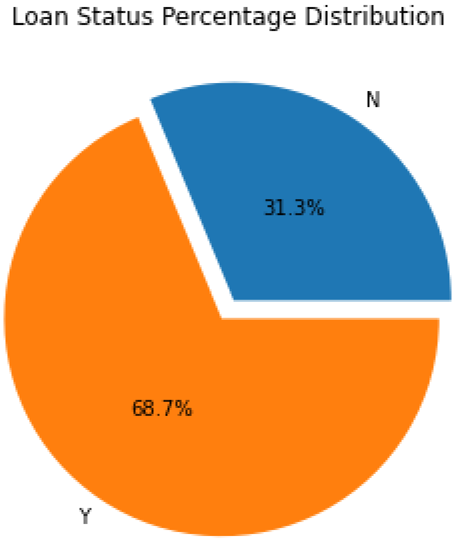


**Figure 10.** Loan status percent distribution

## 3.2. Continuous data analysis

For numerical data to be considered continuous values should range over a certain interval. The dataset contains 4 features which relate to the data that represents each applicant's case. Attributes are the following:

- ApplicantIncome
- CoapplicantIncome
- LoanAmount
- Loan_Amount_Term

Using seaborn which is a python visualization library, a graphical visualization can be performed where all continuous values can be analyzed with respect to each other. As shown in **Figure 11,** the following deductions can be made:

- Applicant Income vs Co-applicant Income: No relation
- Applicant Income vs Loan Amount: based on the scatter plot, there seems to be some linear relation with majority of income ranges from 0 to 300,000 and loan amount ranging from 0 to 650.
- Applicant Income vs Loan amount term: No relation, but applicants tend to select a loan term of 360 months more often.
- Co-applicant income vs loan amount: there seems to be a linear relationship with co-applicant income varying from 0 to 20,000 and loan amount being concentrated in the range of 0 to 600.
- Co-applicant income vs loan amount term: No relation, but applicants tend to select a loan term of either 180 or 360 months more commonly.
- Loan amount vs loan amount term: based on the scatter plot, as loan amount increases so does loan amount term increase in majority of the cases.

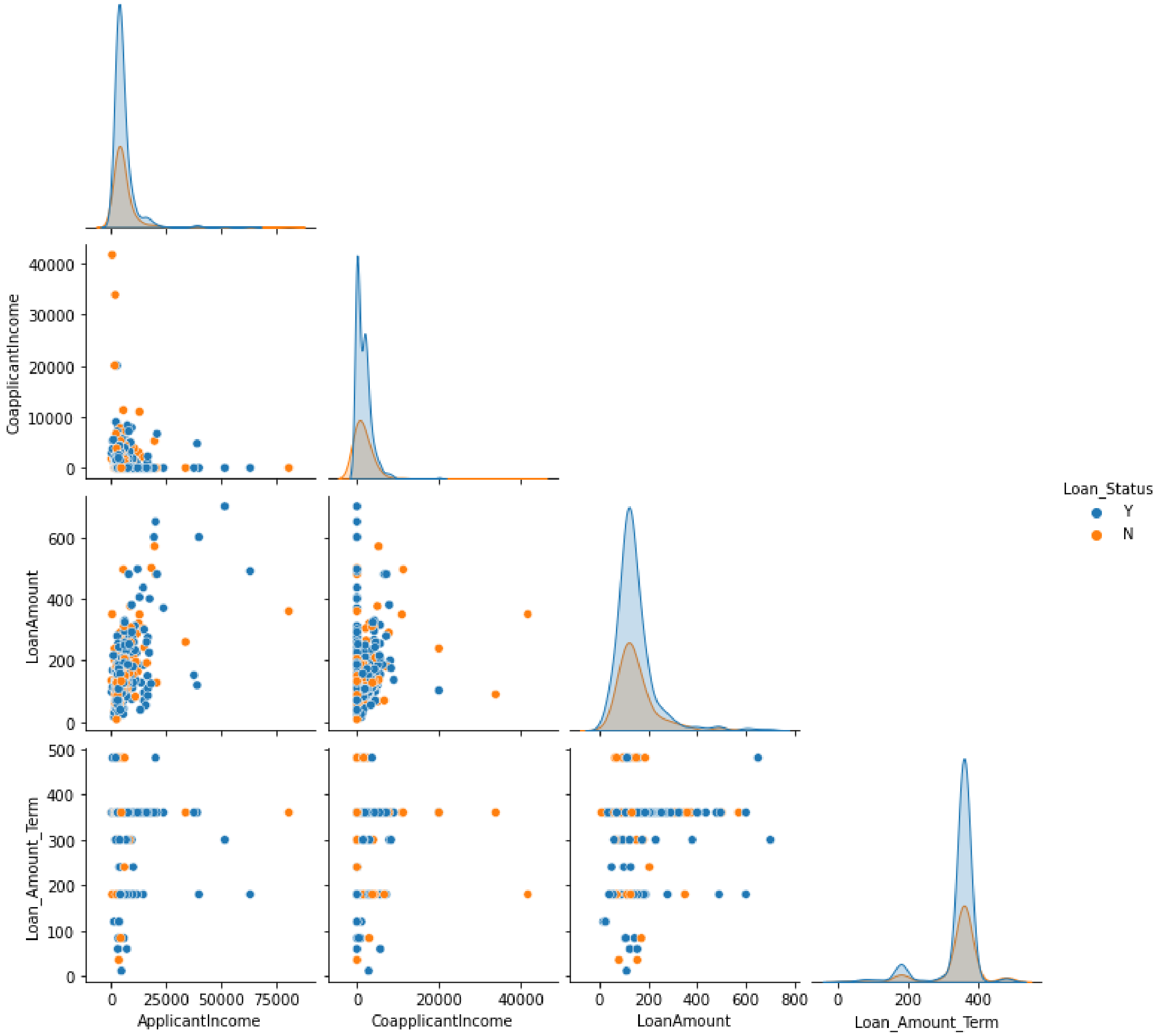


**Figure 11.** Pair plot analyzing continuous features

Another feature which reasonably holds an importance in making such decision is applicants' income and co-applicants' income, **Figure 12** shown below visualizes distribution of incomes and it is safe to say that distribution resembles a Gaussian function. Moreover, further details regarding income will be discussed below.

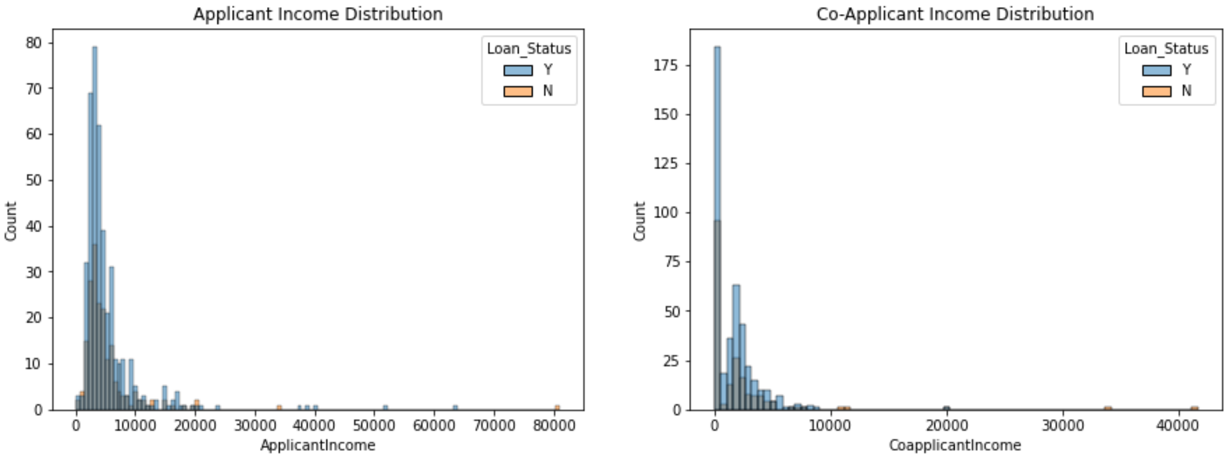


**Figure 12.** Applicant Income distribution (Left), Co-applicant Income distribution (Right)

Furthermore, **Figure 13** shown below displays count of applicants based on the loan amount they select and loan terms. Simply most applicants that are eligible apply for a loan amount of 100,000$ to 150,000$, and as for how many months most commonly 360 months is selected.

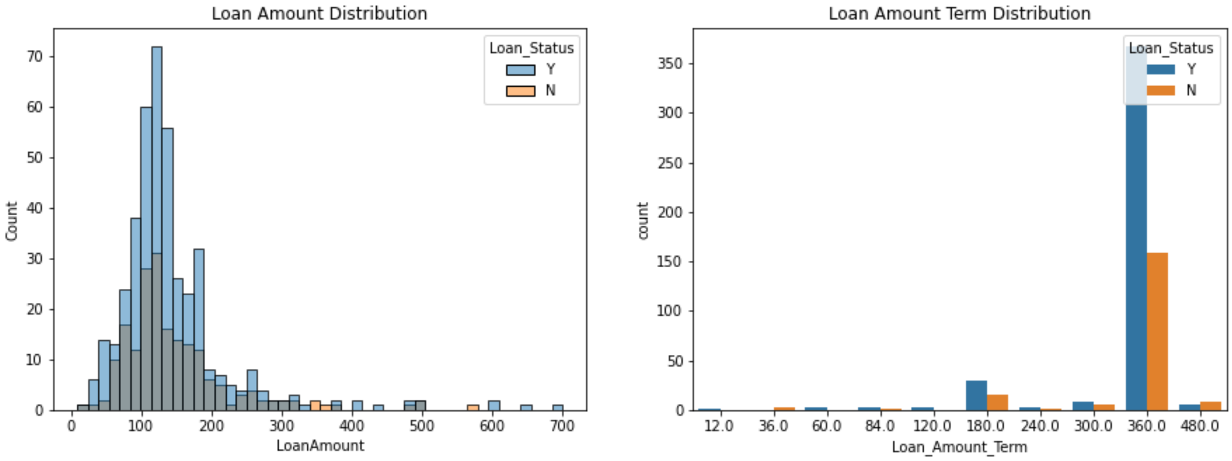
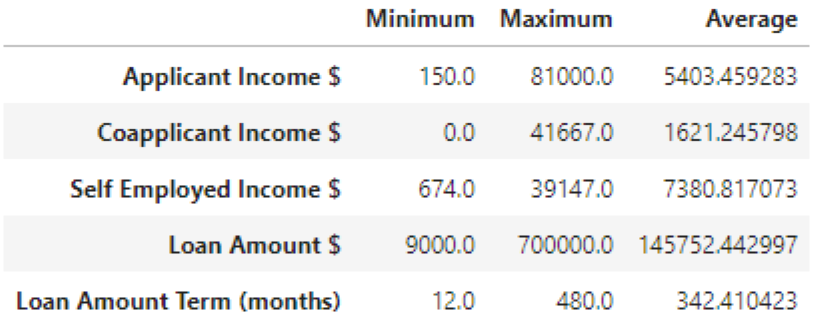


**Figure 13.** Loan Amount Distribution (Left), Loan Amount Term Distribution (Right)

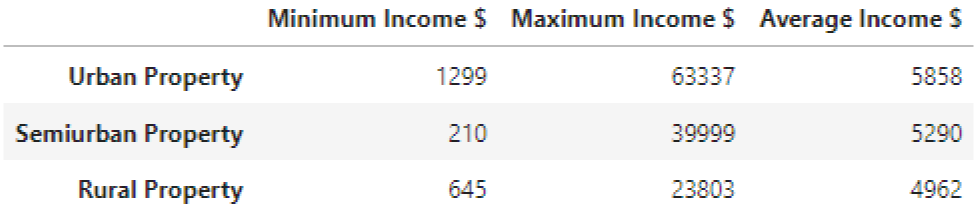
Adding on, **Figure 14** displays the overall data needed to understand applicant's requirements, as seen applicants that apply their income ranges from 150$ to 81, 000$, Co-applicant income ranges from 0$ to 41,667$, Self-employed applicants make roughly 674$ to 39,147$, and applicants select loan amounts of 9,000$ being the minimum up to 700,000$ as their maximum budget and prefer their loan to be extended over a period ranging from 12 months up to 480 months.



| | Minimum | Maximum | Average |
|---|---|---|---|
| Applicant Income $ | 150.0 | 81000.0 | 5403.459283 |
| Coapplicant Income $ | 0.0 | 41667.0 | 1621.245798 |
| Self Employed Income $ | 674.0 | 39147.0 | 7380.817073 |
| Loan Amount $ | 9000.0 | 700000.0 | 145752.442997 |
| Loan Amount Term (months) | 12.0 | 480.0 | 342.410423 |

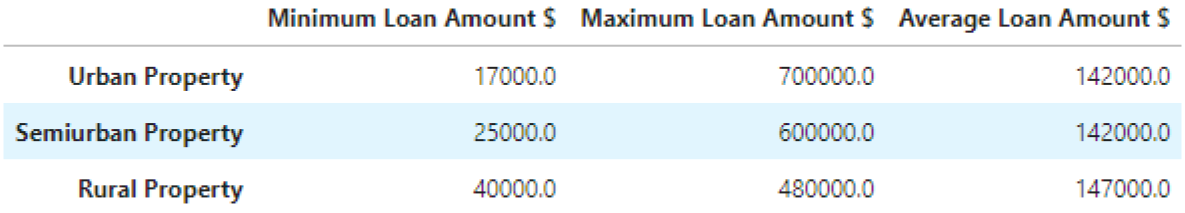
**Figure 14.** Applicant's data

In order to know how much an applicant's income should be and what amount to ask for, data shown below in **Figure 15** and **Figure 16** are computations of applicants that are eligible for loans, and as such the data below can be used as a safety margin that would reveal eligibility status.



| | Minimum Income $ | Maximum Income $ | Average Income $ |
|---|---|---|---|
| Urban Property | 1299 | 63337 | 5858 |
| Semiurban Property | 210 | 39999 | 5290 |
| Rural Property | 645 | 23803 | 4962 |

**Figure 15.** Income required to purchase properties



| | Minimum Loan Amount $ | Maximum Loan Amount $ | Average Loan Amount $ |
|---|---|---|---|
| Urban Property | 17000.0 | 700000.0 | 142000.0 |
| Semiurban Property | 25000.0 | 600000.0 | 142000.0 |
| Rural Property | 40000.0 | 480000.0 | 147000.0 |

**Figure 16.** Loan Amount needed to purchase properties

Another visualization shown below in **Figure 17** which is a correlation matrix, this matrix computes how each feature is correlated with respect to other features. As seen, we have values that range from the negative side up to the positive side, a negative correlation coefficient indicates that if one value increases then the other value decreases and as for positive correlation if a value increases, then the other value increases. **Table 4** shown below describes the correlation matrix that have a positive correlation with a score greater or equal to 0.1.

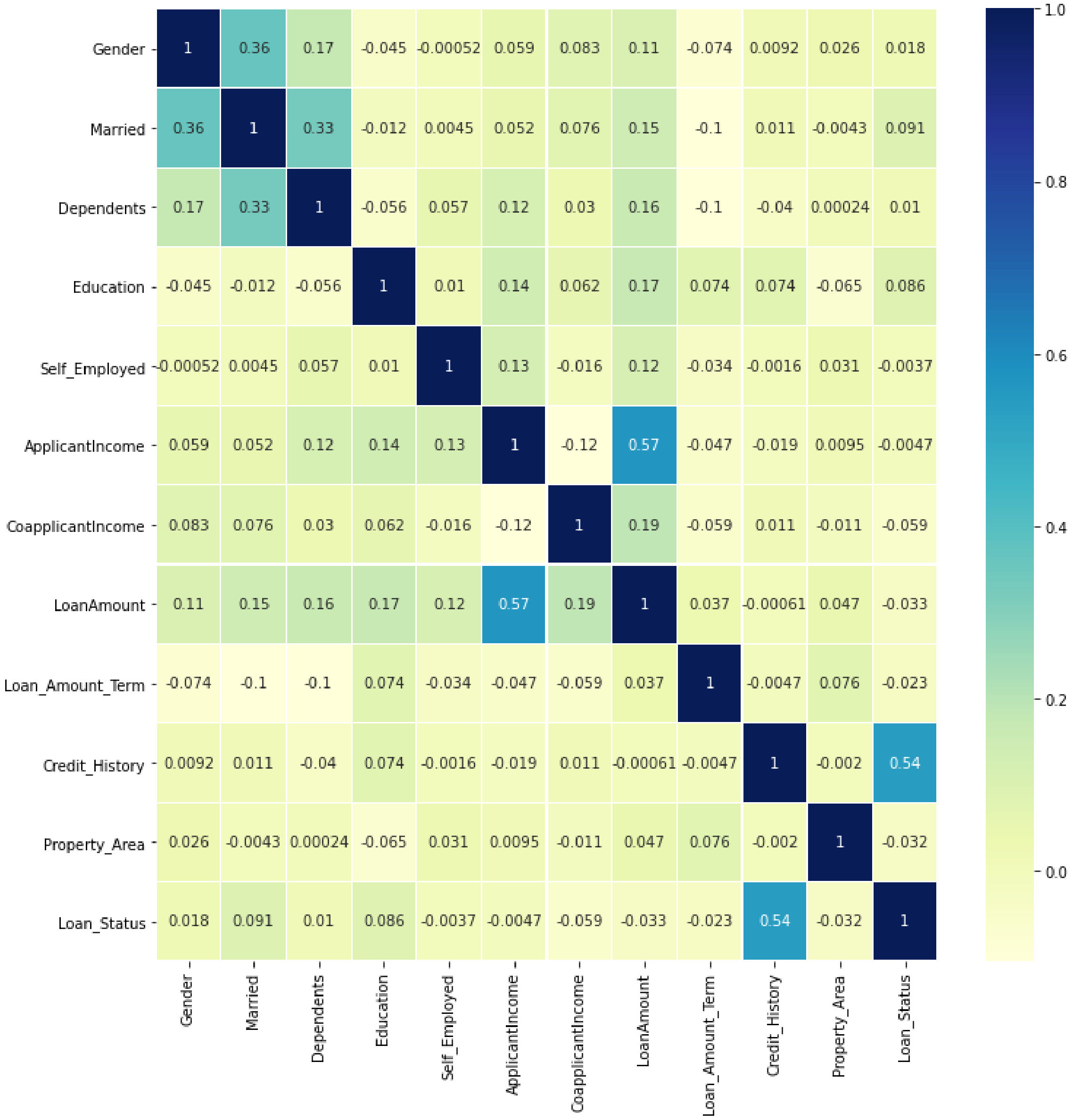


**Figure 17.** Correlation matrix

**Table 4.** Correlation Matrix Deductions made of features with correlation score >=0.1

| Feature 1 & Feature 2 | Assumptions | Correlation Coefficient Score |
|---|---|---|
| Married – Gender | ? | 0.36 |
| Dependents – Gender | ? | 0.17 |
| Dependents – Married | An assumption can be made that if an applicant is married then there is a possibility that they have dependents and this as a result when predicting eligibility for loan, indicates whether | 0.33 |

| | | |
|---|---|---|
| | after all expenses does the applicant have enough income to pay back his loan | |
| ApplicantIncome - Dependents | Similarly, if a final decision had to be made regarding loan eligibility, income minus expenses of dependents indicates whether applicant has enough money to make monthly payments. | 0.12 |
| ApplicantIncome – Education | Education is an indicator of whether the applicant is still studying, or has graduated. If applicant is still studying then it is highly possible that applicant has other payments to make on time. Another scenario is the level of education indicates a job position with high level income. Moreover, for a final decision to be made applicants income and education level is taken into consideration. | 0.14 |
| ApplicantIncome – Self_Employed | A self-employed applicant simply put doesn’t have a steady source of income on a monthly basis and there could be a possibility that there are high fluctuations in the applicant’s monthly income and as such for a final decision to be made when granting loans, this is another feature that should be looked into. | 0.13 |
| LoanAmount – Gender | ? | 0.11 |
| LoanAmount – Married | Marital status of an applicant indicates whether after all expenses would there be enough to pay back their loan. In other words, if applicant’s other half has an occupation, then this would be fine, but if it weren’t the case then this aspect should be taken into consideration. | 0.15 |
| LoanAmount – Dependents | This correlation explores a side where the applicants expenses determine the amount of loan the applicant is eligible to take. | 0.16 |
| LoanAmount – Education | This correlation explains that loan amount is related to the level of education an applicant, whether the applicant is currently studying, has graduated, or has no educational background. | 0.17 |
| LoanAmount – Self_Employed | Similarly, applicant’s income should be enough to supply for paying back their | 0.12 |

| | | |
|---|---|---|
| | loans, and for such case of being a self-employed applicant monthly income should be enough to cover their own expenses and pay back their mortgage. | |
| LoanAmount – ApplicantIncome | This correlation indicates that based on the applicant’s income a loan amount should be taken based on their income in such a way that they are capable of paying back their monthly payments on time. | 0.57 |
| LoanAmount – CoapplicantIncome | Similarly, this feature determines whether co-applicant income is sufficient enough to cover and payback their mortgage and as well as supplement their monthly expenses. | 0.19 |
| LoanStatus - CreditHistory | This correlation, indicates whether an applicant has previous credit history. this is a good indicator of previous historical data which indicates whether applicant pays their mortgage on time or whether they have unpaid previous mortgages. This is an important feature to look at since it reveals financial historical data regarding their mortgages and other finances. | 0.54 |

## 3.3. Data Analysis – Summary

- Data is male dominant.
- 65.3% of applicants are married.
- 78.2% of applicants have graduated.
- 13.4% of applicants are self-employed.
- 85.5% of applicants have credit history.
- 68.7% of applicants are eligible for loans.
- Income needed to be eligible for a loan and to purchase urban property ranges from 1,299 USD to 63,337 USD.
- Income needed to be eligible for a loan and to purchase semiurban property ranges from 210 USD to 39,999 USD.
- Income needed to be eligible for a loan and to purchase rural property ranges from 645 USD to 23,803 USD.
- Loan amount needed to be eligible for a loan and to purchase urban property ranges from 17,000 USD to 700,000 USD.
- Loan amount needed to be eligible for a loan and to purchase semiurban property ranges from 25,000 USD to 600,000 USD.

- Loan amount needed to be eligible for a loan and to purchase urban property ranges from 40,000 USD to 480,000 USD.
- 68.7% of applicants are eligible for loans, meaning that dataset is imbalanced.

# 4. Outlier detection & elimination

## 4.1. Local Outlier Factor Method

To eliminate outliers that would cause the model to perform badly, the Local Outlier Factor (LOF) Algorithm will be used. Simply, it is an unsupervised detection algorithm which computes the local density deviation of a given data point with respect to its neighbors. In this case, samples that have a substantially lower density than their neighbors are considered to be as outliers.

As such, using LOF algorithm dataset is reduced from 614 data samples down to 552 data samples where 62 data points have been eliminated.

# 5. Data Normalization – Standardization

When a dataset contains multiple features that has various scales for numeric values, machine learning algorithms tend to take time to converge to an optimal solution. Thus, a technique called normalization or either standardization is introduced, such method allows the learning algorithm to perform better on the dataset since this would reduce the sensitivity in terms that any change wont significantly affect predictions.

## 5.1 Data Normalization

Normalization is a scaling technique in which values are shifted and rescaled so that they end up ranging between 0 and 1, another term for such method is commonly referred to as the Min-Max Scaling method. The formula to normalize such set of data to range in between 0 and 1 is shown in **Figure 18** below.

$$X^{'} = \frac{X - X_{min}}{X_{max} - X_{min}}$$

**Figure 18.** Normalization Formula

Moreover, $X_{min}$ if taken as a vector holds the minimum value of each feature and as for $X_{max}$ if taken as vector, it holds the maximum value of each feature. Thus, when the datapoint is normalized using the formula shown in **Figure 18**, the value of $X'$ should range in between 0 and 1.

## 5.2. Data Standardization

Standardization is another scaling technique where the values are centered around the mean with a unit standard deviation. This means that the mean of the attribute becomes zero and resultant distribution has a unit standard deviation. The formula to normal such dataset is shown in **Figure 19** below.

$$X' = \frac{X - \mu}{\sigma}$$

**Figure 19.** Standardization Formula

The mean represented by the symbol $\mu$ is the average of the feature values, and the $\sigma$ is the standard deviation of the feature values.

# 6. Machine Learning

In this section before starting training and deploying models, data preprocessing has to continue from where we left which was at encoding categorical data. Following this step outliers should be eliminated, once done data will either be normalized or standardized depending on what technique best suits each model. As such once the mentioned process is done, data will be split into training and testing data where training data holds 80% of the original data and the remaining goes to testing. Following up, machine learning models will be discussed briefly and the following models will be trained and deployed.

- Logistic Regression
- Decision Trees
- KNN (K-Nearest Neighbor)
- Random Forest Classification

## 6.1. Machine Learning models

In this section, machine learning models will be discussed briefly and results will be compared with respect to each of the trained models.

### 6.1.1. Logistic Regression

Logistic regression is used to predict discrete values from a set of independent variables. It helps predict the probability of an event by fitting the data to a logit function. Moreover, since our application involves predicting whether applicant is eligible or not, this can be interpreted as the case of a binary classification and thus logistic regression can be used for such application.

Mathematically speaking, the logistic regression uses the output of a linear regression model and passes through the sigmoid activation function which outputs a probability. the sigmoid function can be defined and visualized as shown in **Figure 20.** Briefly speaking, the logistic function outputs values that range from 0 to 1 which can be interpreted as a probability where if output is greater or equal to 0.5 then predict true otherwise false.

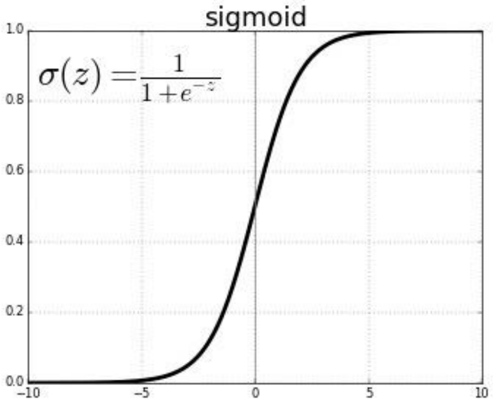


**Figure 20.** Sigmoid Function

### 6.1.2. Decision Trees

The decision tree Algorithm belongs to the family of supervised machine learning algorithms. It can be used for both a classification problem as well as for regression problem. The goal of this algorithm is to create a model that predicts the value of a target variable, for which the decision tree uses the tree representation to solve the problem in which the leaf node corresponds to a class label and attributes are represented on the internal node of the tree.

Since multiple variants of algorithms exist to generate trees, this approach focuses on the entropy factor, in other words the tree uses the ID3 algorithm to generate the tree and its nodes. **Figure 21**, shows the typical structure of a decision tree.

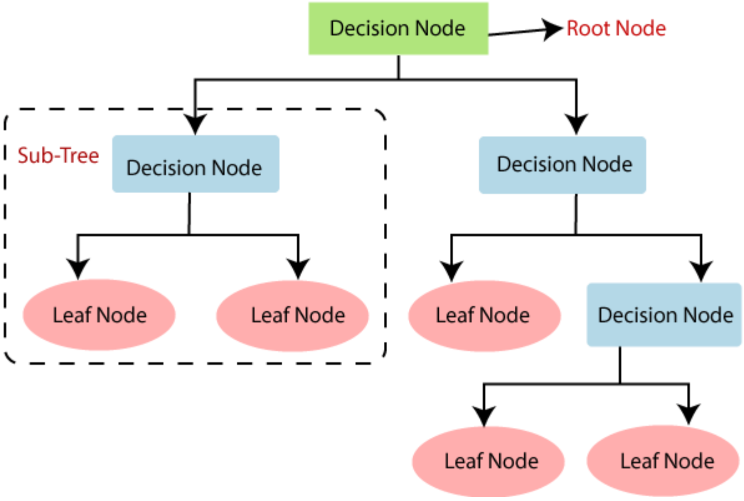


**Figure 21.** Decision Tree Classifier Structure

### 6.1.3. KNN (K-Nearest Neighbor)

KNN is an approach to data classification that estimates how likely a data point is to be a member of one group or the other depending on what group the data points nearest to it are in.

The k-nearest-neighbor is an example of a "lazy learner" algorithm, meaning that it does not build a model using the training set until a query of the data set is performed. More specifically, KNN uses the distance approach to estimate whether this datapoint belongs to a certain group or not. **Figure 22** shows a visualization of how KNN basically estimates it closest neighbors and produces a prediction.

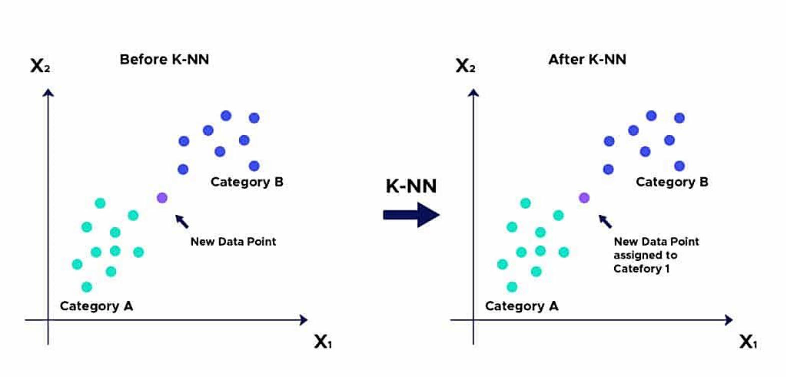


**Figure 22.** KNN Classification

### 6.1.4. Random Forest Classifier

Random forest is a supervised learning algorithm. The "forest" it builds, is an ensemble of decision trees, usually trained with the “bagging” method. The general idea of the bagging method is that a combination of learning models increases the overall result.

Simply put, random forest builds multiple decision trees and merges them together to get a more accurate and stable prediction. One big advantage of random forest is that it can be used for both classification and regression problems, which form the majority of current machine learning systems. Let's look at random forest in classification, since classification is sometimes considered the building block of machine learning.

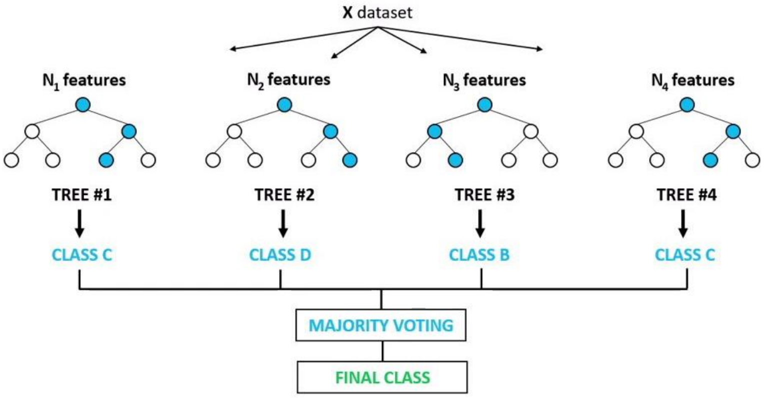


**Figure 23.** Random Forest Classifier Structure

## 6.2. Machine Learning models results

### 6.2.1. Logistic Regression

Upon training a logistic model on the given data, after tuning its hyperparameters the model performed as such, on the training set an accuracy of 81.48% was obtained and as for the cross-validation set an accuracy of 88.07% was obtained. As shown in **Figure 24,** as number of samples increase the learning curve of both training and cross-validation tend to meet at a certain point where the curve is considered to be a good fit, in other words it doesn't overfit nor underfit and as such we can say that this logistic model deployed does indeed have a good performance.

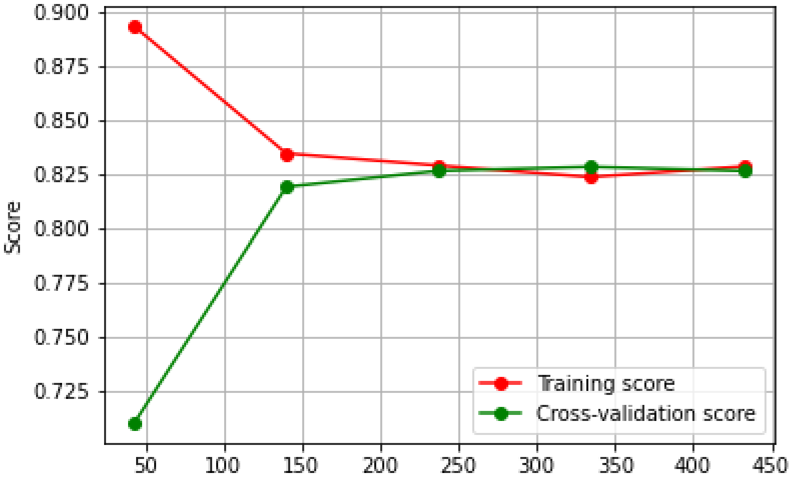
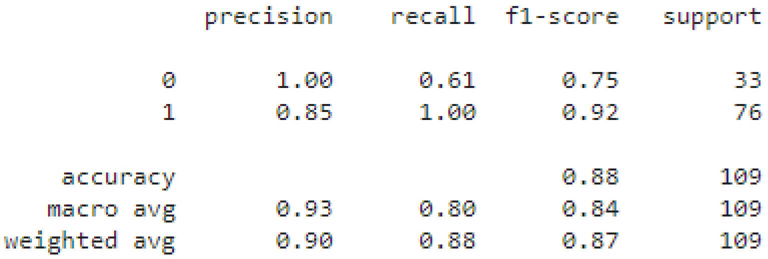


**Figure 24.** Logistic Regression model learning curve

Moving on, another metric to evaluate the performance of a model is to use a classification report which basically covers the precision, recall, f1-score, and support on the predictions made by the logistic regression model. As shown in **Figure 25,** the f1-score will be taken into consideration as our main metric for evaluation since we have an uneven class distribution this metric is more useful than accuracy. As observed, an f1-score greater than 0.5 is considered to be a good value since it represents the weighted average of precision and recall, which means that this score takes both false positives and false negatives into account and as such it is a good indicator of the model's performance.



```
              precision    recall  f1-score   support

           0       1.00      0.61      0.75        33
           1       0.85      1.00      0.92        76

    accuracy                           0.88       109
   macro avg       0.93      0.80      0.84       109
weighted avg       0.90      0.88      0.87       109
```

**Figure 25.** Logistic regression classification report

Another metric for evaluating the deployed model, is to use a confusion matrix since it reveals how the model is classifying each of the cross-validation datapoints. As observed in **Figure 26,** 20 out of 33 datapoints labeled as 0 are correctly classified and the remaining 13 samples are misclassified as 1. And as for data with label 1, data is 100% classified correctly. With this simple observation, we know that data is unbalanced so we can explain why a few samples labeled as 0 are being misclassified.

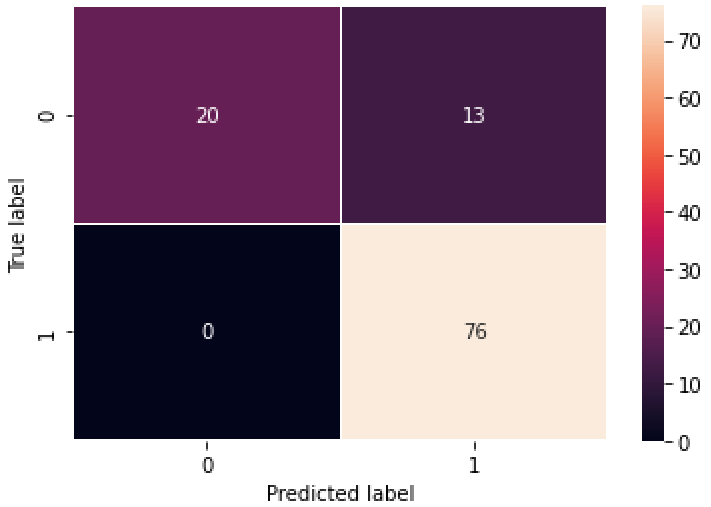


**Figure 26.** Logistic regression model prediction on cross-validation data

Moreover, have to better understand how well the model is performing, the model will be tested on the testing data, and **Figure 27** shows a confusion matrix on the predictions made by the logistic regression model.

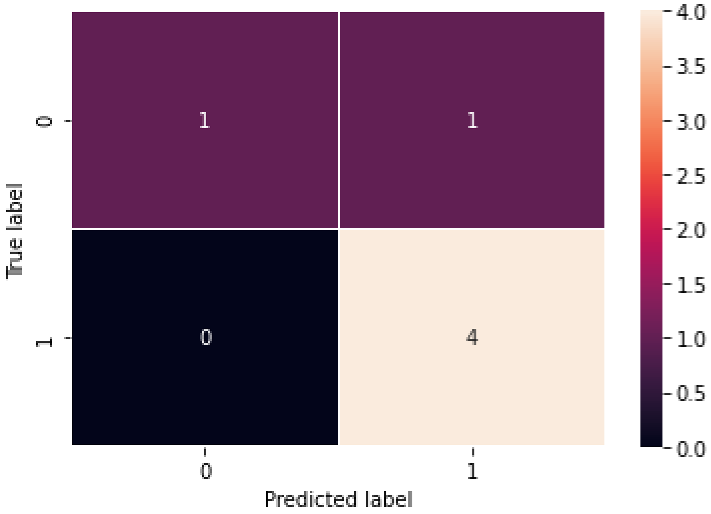


**Figure 27.** Logistic regression model prediction on testing data

### 6.2.2. Decision Trees

The decision tree model after tuning its hyperparameters, on the training set an accuracy of 83.79% was obtained and as for the cross-validation set an accuracy of 86.23% was obtained. As shown in **Figure 28**, the learning curve shows that the model is overfitting, meaning that the model hasn't generalized well enough.

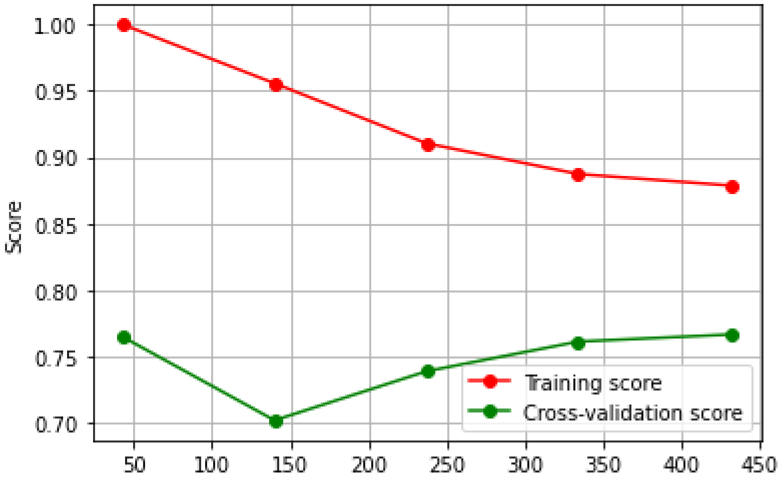
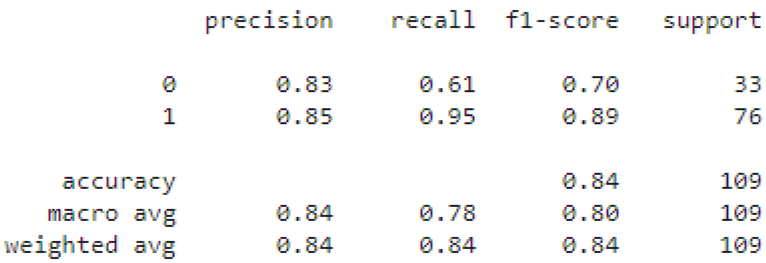


**Figure 28.** Decision Tree model Learning Curve

Moreover, taking into consideration the f1-score of the classification report generated shown in **Figure 29,** an observation can be made that both f1-scores are acceptable.



```
              precision    recall  f1-score   support

           0       0.83      0.61      0.70        33
           1       0.85      0.95      0.89        76

    accuracy                           0.84       109
   macro avg       0.84      0.78      0.80       109
weighted avg       0.84      0.84      0.84       109
```

**Figure 29.** Decision Tree model classification report

Moving on, **Figure 30** shows the predictions made by the Decision tree model on the cross-validation dataset. As observed data labeled as 0, out of the 33 20 have been labeled correctly and the remaining data points have been misclassified. As for data with label 1, out of 76 samples 4 have been misclassified and 72 have been correctly classified.

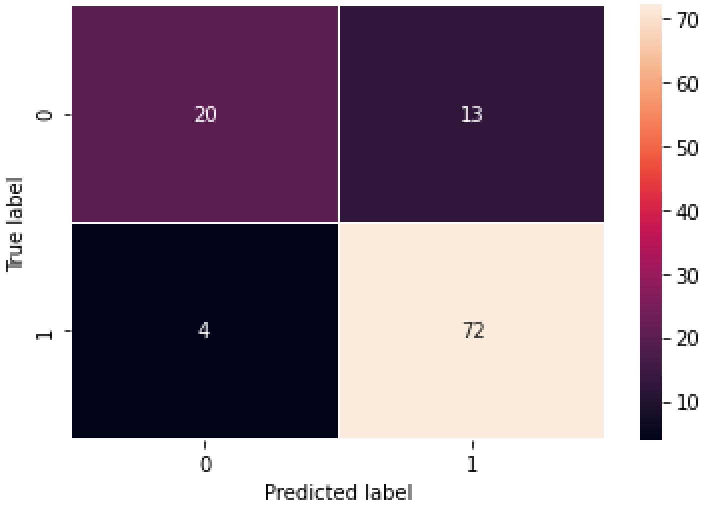


**Figure 30.** Decision Tree model predictions made on cross-validation data

Having evaluated the decision tree model on cross-validation data, upon final inspection the model will be tested on the testing data which basically contains 6 samples and results obtained can be seen in **Figure 31** as shown below.

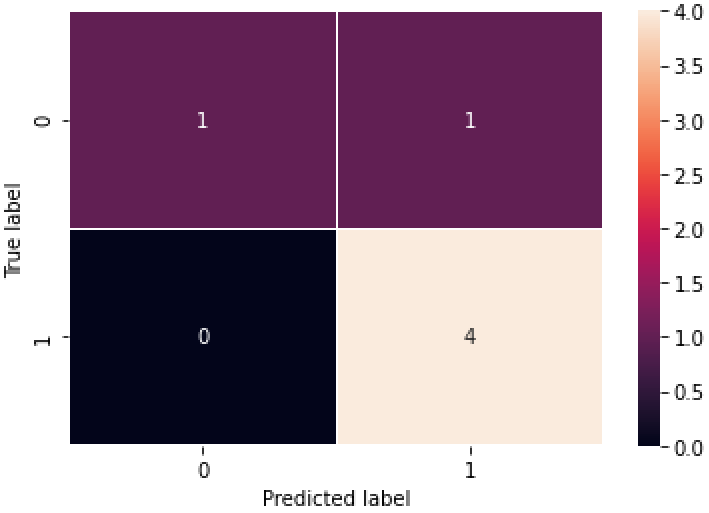


**Figure 31.** Decision Tree model predictions made on testing data

Moreover, after pruning and tuning all hyper-parameters, **Figure 32** which is a visualization of the decision tree deployed to predict whether an applicant is eligible for a loan or not.

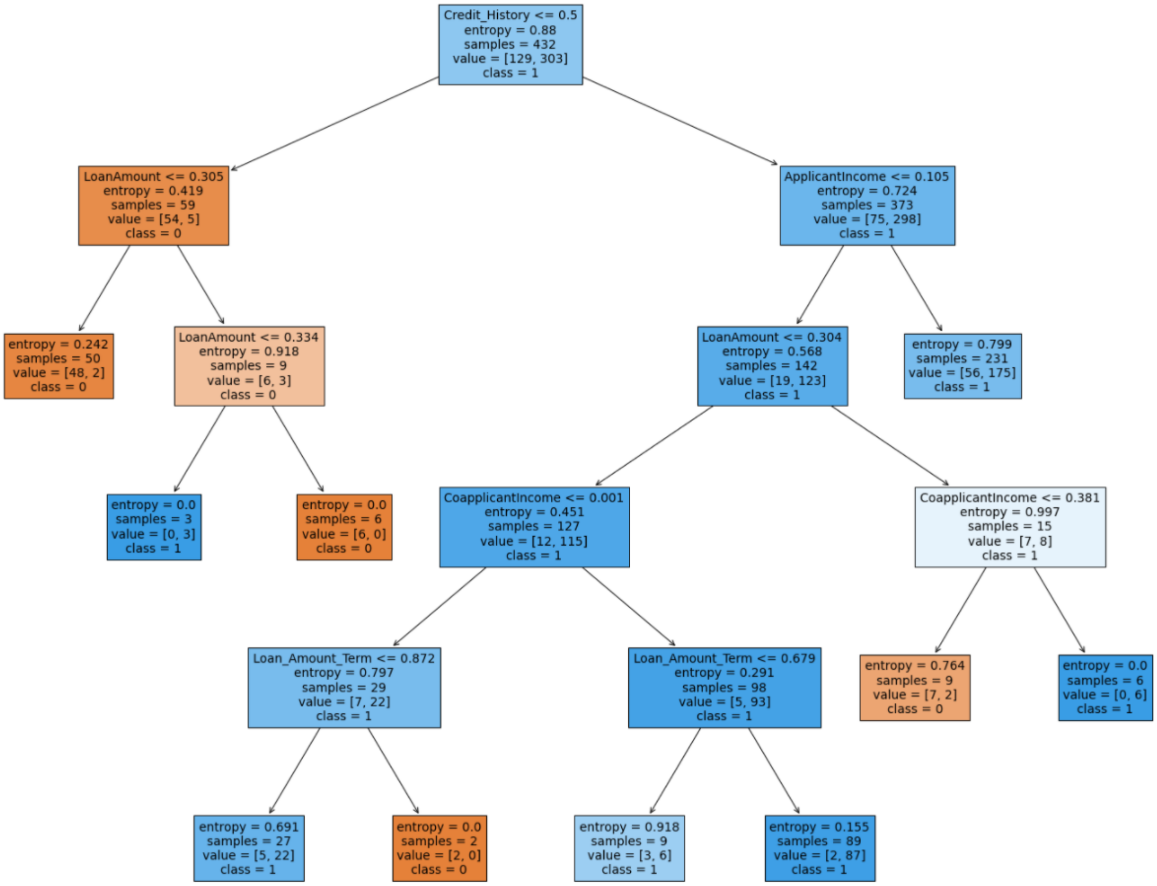


**Figure 32.** Decision Tree model

### 6.2.3. KNN (K-Nearest Neighbor)

Adding on, KNN another learning algorithm has been trained and deployed on this dataset and results are as such, upon training the accuracy is 81.01% and as for the cross-validation accuracy, a score of 88.07% has been obtained. By briefly observing the learning curve shown in **Figure 33,** it is clear that the model ………………………………

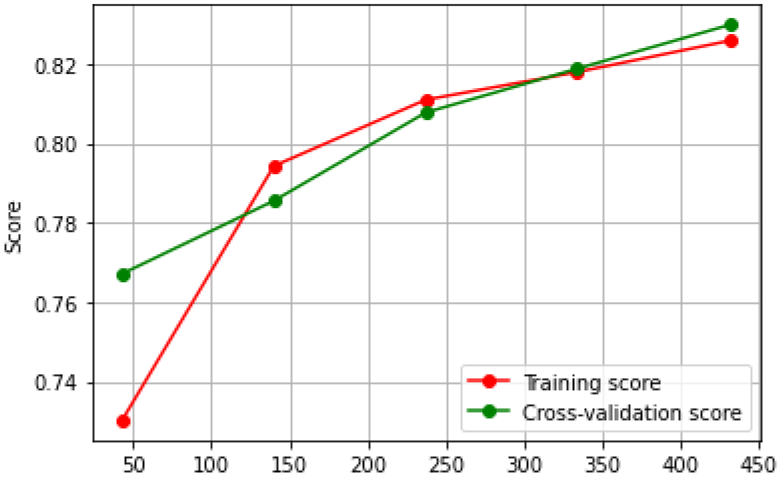
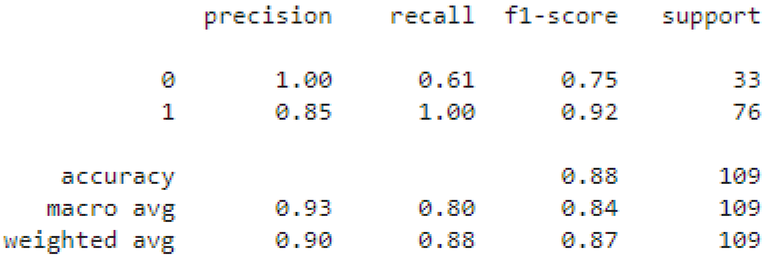


**Figure 33.** KNN model learning curve

Having trained the KNN model on the cross-validation dataset, **Figure 34** shows the classification report generated that displays the f1-score to evaluate the performance of the model.



```
              precision    recall  f1-score   support

           0       1.00      0.61      0.75        33
           1       0.85      1.00      0.92        76

    accuracy                           0.88       109
   macro avg       0.93      0.80      0.84       109
weighted avg       0.90      0.88      0.87       109
```

**Figure 34.** KNN model classification report

Adding on, the predictions made by the KNN model can be observed in **Figure 35** where data labeled as 1 have been correctly classified, and as for data labeled as 0 out of the 33 datapoints labeled as 0, 20 have been classified correctly and remaining have been misclassified.

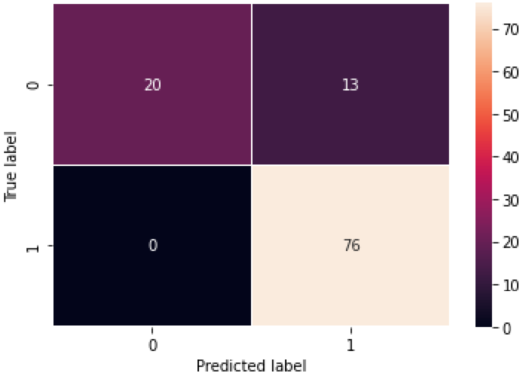


**Figure 35.** KNN model predictions made on cross-validation data

**Figure 36** is a confusion matrix of the KNN model evaluated on the testing dataset, and as observed out of the 6 samples only one sample has been misclassified.

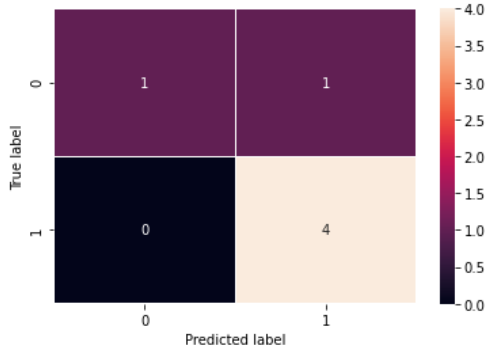


**Figure 36.** KNN model predictions made on testing data

### 6.2.4. Random Forest Classification

…………………………………………………………………J

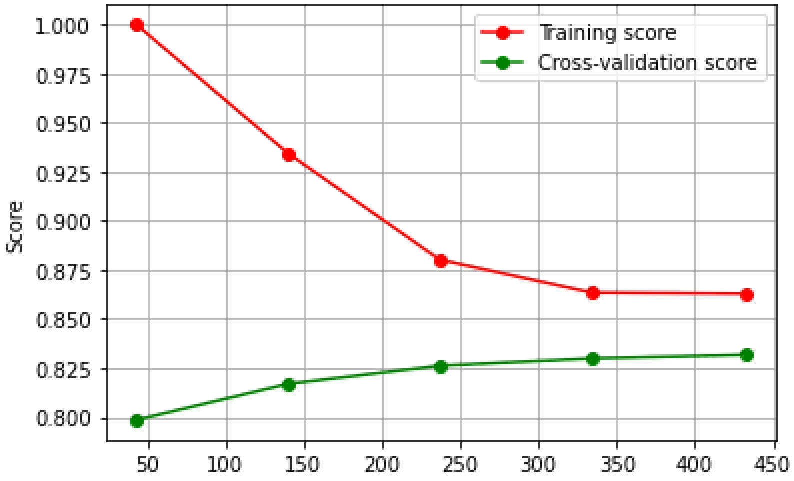
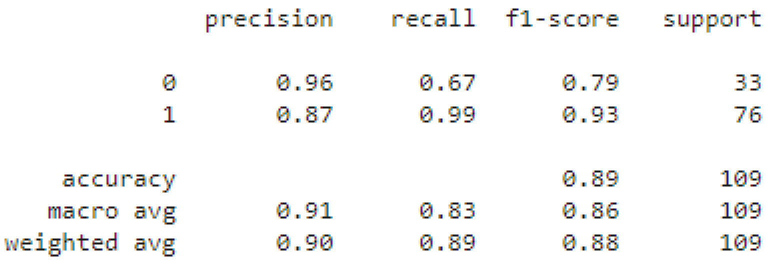


**Figure 37.** Random Forest Classification model learning curve

Having trained the random forest model, a classification report can be generated which basically covers most necessary metrics used to evaluate the performance of the model, and as such for our use case the f1-score metric will be used. As observed in **Figure 38,** we can see that the f1-score is above 0.5 which is a good indication that the model is taking into consideration the false classifications into its evaluation as well.



```
              precision    recall  f1-score   support

           0       0.96      0.67      0.79        33
           1       0.87      0.99      0.93        76

    accuracy                           0.89       109
   macro avg       0.91      0.83      0.86       109
weighted avg       0.90      0.89      0.88       109
```

**Figure 38.** Random Forest Classification model classification report

Moreover, **Figure 39** shows the predictions made by the random forest model on the cross-validation set. As observed out of the 33 samples labeled as 0, 22 are classified correctly and 11 misclassified. And as for samples labeled as 1, out of the 76 labeled as 1only 1 sample has been misclassified.

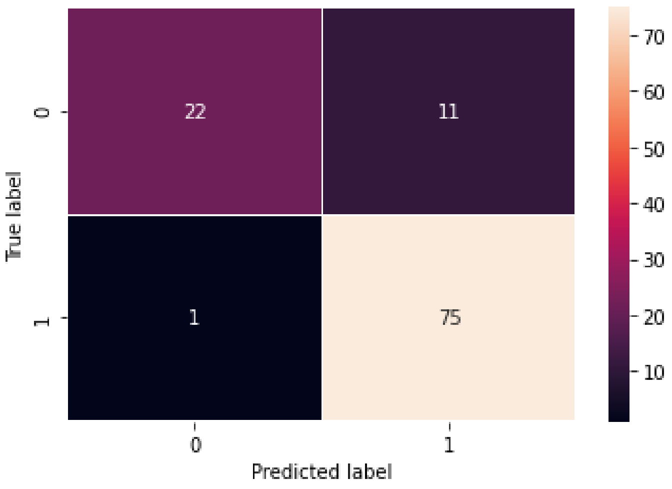


**Figure 39.** Random Forest Classification model predictions made on cross-validation data

Moreover, using the trained random forest model on the testing dataset the following predictions can be observed as shown in **Figure 40** where only 1 sample has been misclassified.

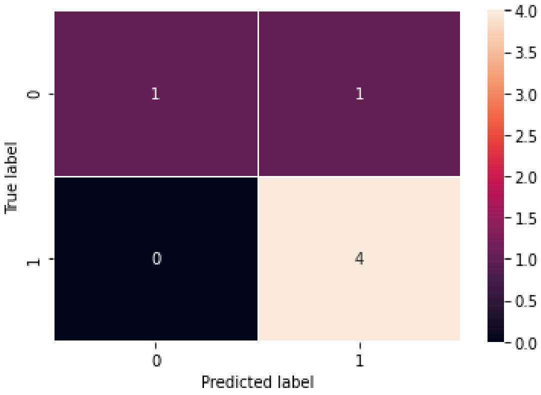
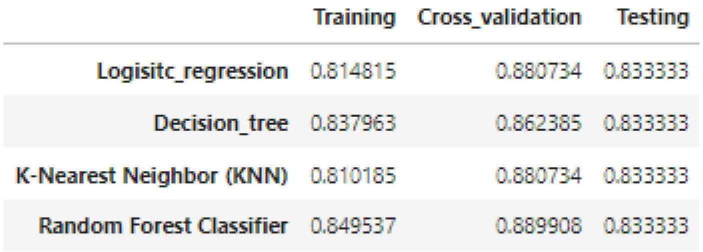


**Figure 40.** Random Forest Classification model predictions made on testing data

# 7. Conclusion

Having developed and tested multiple machine learning models, and evaluating models based on the produced metrics, the decision tree model and random forest classifier model seems to be the best model in this case for such application. As we can observe the results shown in **Figure 41,** basically every model performs the same on the testing data, where each model misclassified 1 wrong sample out of the 6 testing samples. Overall, the models need to be tuned properly in such a way that it maximizes its performance and minimizes its prediction error.



| | Training | Cross_validation | Testing |
|---|---|---|---|
| **Logisitc_regression** | 0.814815 | 0.880734 | 0.833333 |
| **Decision_tree** | 0.837963 | 0.862385 | 0.833333 |
| **K-Nearest Neighbor (KNN)** | 0.810185 | 0.880734 | 0.833333 |
| **Random Forest Classifier** | 0.849537 | 0.889908 | 0.833333 |

**Figure 41.** Deployed Models Scores