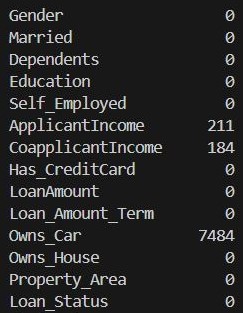
**Project: Putting It Together**

# **A) Preprocessing**

1. How many empty values did each column contain?

* Only three columns: ApplicantIncome, CoapplicantIncome, and Owns\_Car had missing or empty values. The column ApplicantIncome had 211 missing values, CoapplicantIncome had 184 missing values, and Owns\_Car had 7484 missing values.
* The details of all the columns with their missing value is listed in the following picture:



1. How many outliers did ApplicantIncome contain?

* There were 10,000 number of rows before removing outliers for ApplicantIncome columns and later after removing the outliers, the resultant number of rows were 9819. Therefore, there were 10,000 - 9819 = 181 number of rows that were classified as an outlier for ApplicantIncome column.

1. How many outliers did CoapplicantIncome contain?

* There were 9819 number of rows before removing outliers for CoapplicantIncome columns and later after removing the outliers, the resultant number of rows were 9623. Therefore, there were 9819 - 9623= 196 number of rows that were classified as an outlier for CoapplicantIncome column.

1. Which columns did you delete, and why?

* The columns that were dropped are as below:
  + Loan\_ID – This is an auto-increment column with all the values being distinct and does not provide any meaningful information for predicting bank loan approval.
  + Applicant\_ID - This is an auto-increment column with all the values being distinct and does not provide any meaningful information for predicting bank loan approval.
  + Owns\_Car - The column Owns\_Car has 7484 out of 10000 missing values which is greater than our threshold of 50% missing value.

**Summary**

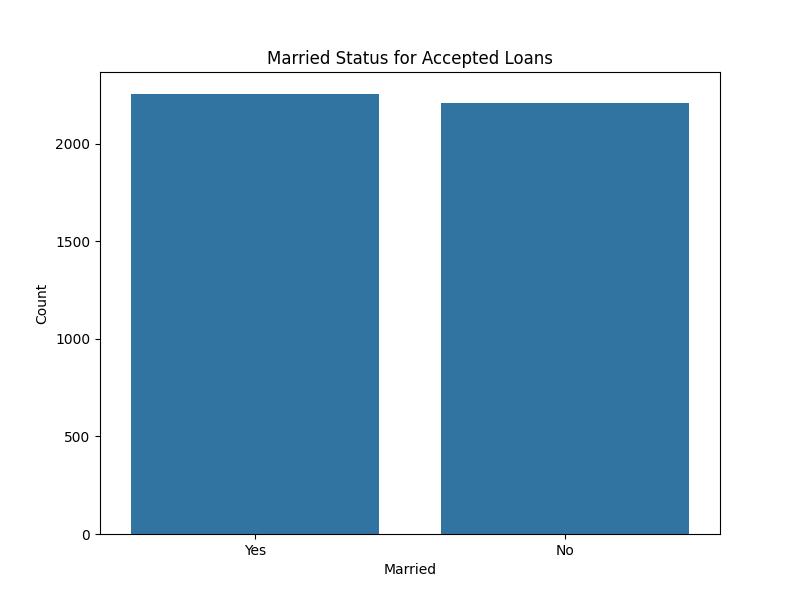
The initial dataset had values “Unknown” to which we initially replaced it with NAN value. This helped us to use the inbuilt python functions such as “dataset.isna().sum()” to calculate the missing values. As a result, the column “Owns\_Car” was dropped as it had more than 50% missing values whereas the missing values of “ApplicantIncome” and “CoapplicantIncome” were imputed using KNNImputer. Additionally, the columns “Loan\_ID” and “Applicant\_ID” where also dropped as it had all auto incremented unique values which does not provide any valuable information for predicting bank loan approval.

None of the other columns required any deletion. This was strengthened by calculating variance of each column and Pearson's correlation coefficient between each variable. The columns “ApplicantIncome” and “CoapplicantIncome” were however converted into data type of “number” before calculation.

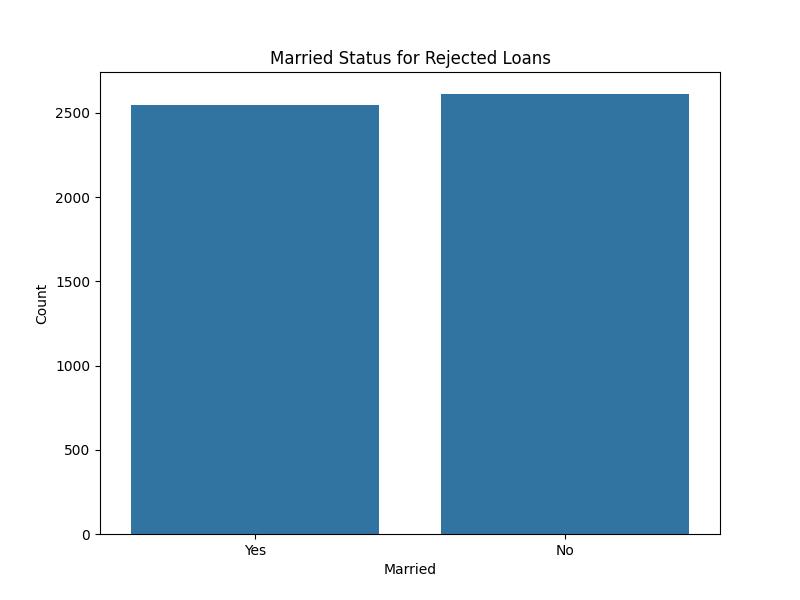
Finally, the outliers were removed for the columns “ApplicantIncome” and “CoapplicantIncome” which concluded our preprocessing step.

# **B) Visualization**

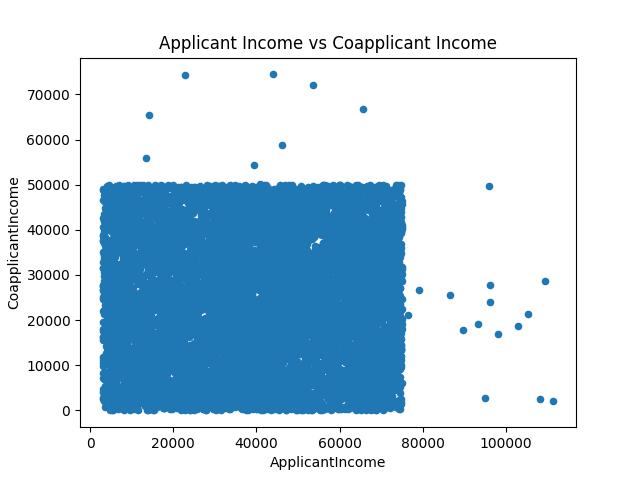
1. Figure 1: Married column for accepted loans



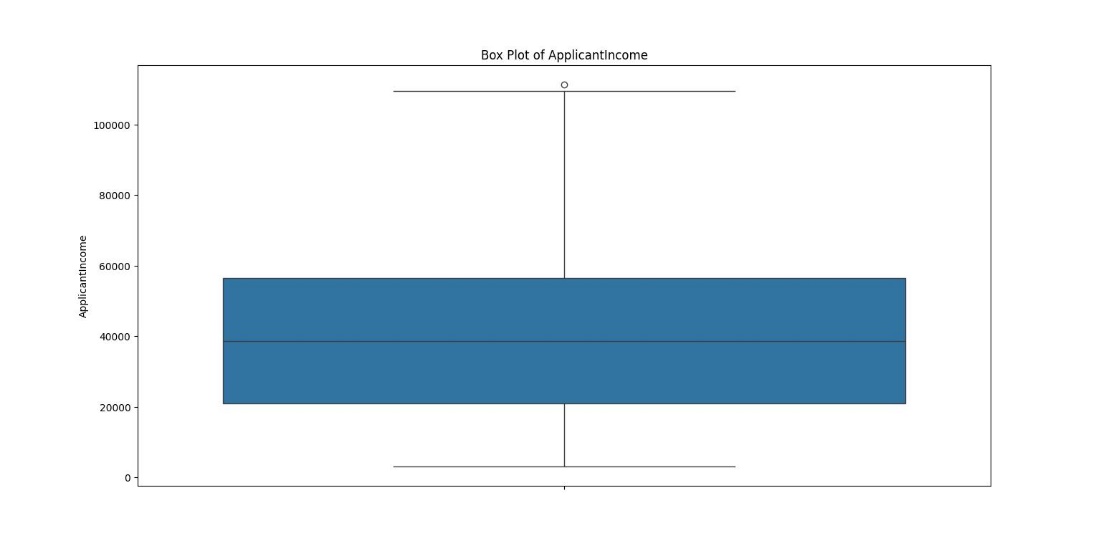
1. Figure 2: Married column for rejected loans



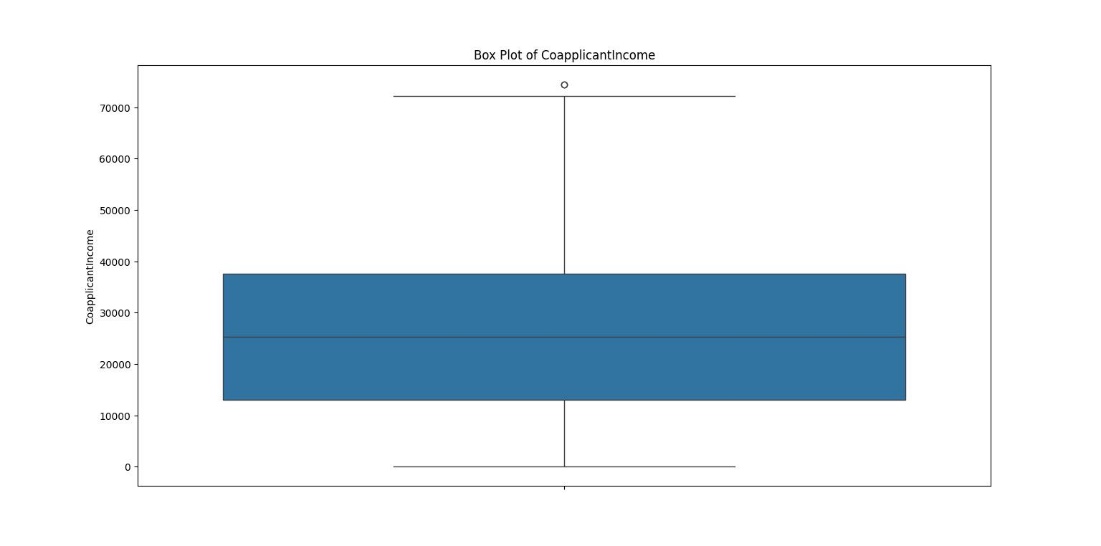
1. Figure 3: Scatter plot for ApplicantIncome and CoapplicantIncome



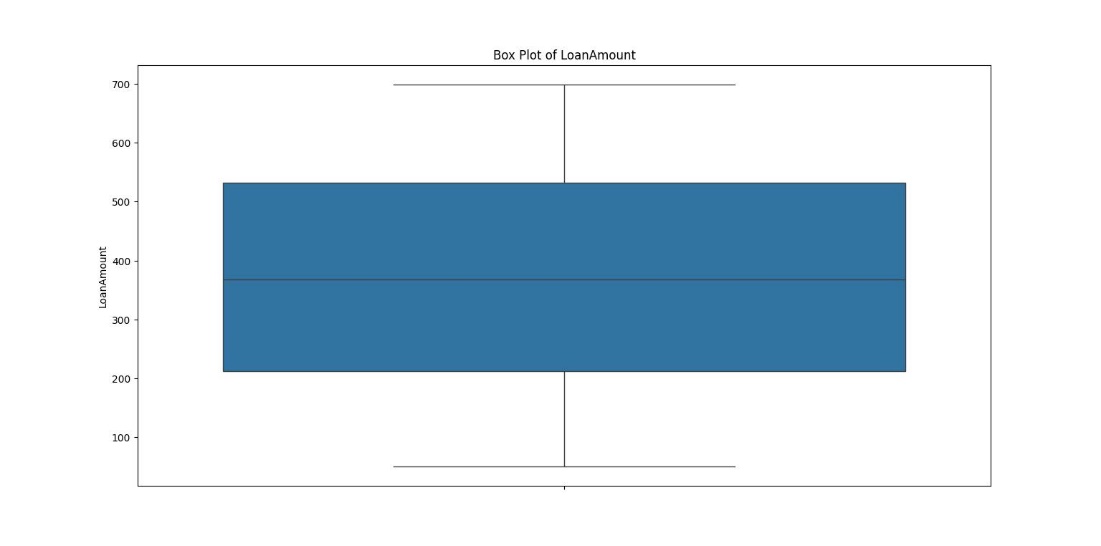
1. Figure 4: Box plot for ApplicantIncome, CoapplicantIncome, and LoanAmount
   1. ApplicantIncome



* 1. CoapplicantIncome



* 1. LoanAmount



# **C) Descriptive Analytics**

* 5 Number Summary for ApplicantIncome
  1. ApplicantIncome Minimum Value = 3002.0
  2. ApplicantIncome 1st Quartile = 21045.5
  3. ApplicantIncome Median = 38693.0
  4. ApplicantIncome 3rd Quartile = 56548.5
  5. ApplicantIncome Maximum Value = 111410.0
* 5 Number Summary for CoapplicantIncome
  1. CoapplicantIncome Minimum Value = 0.0
  2. CoapplicantIncome 1st Quartile = 13028.5
  3. CoapplicantIncome Median = 25272.0
  4. CoapplicantIncome 3rd Quartile = 37516.0
  5. CoapplicantIncome Maximum Value = 74440.0
* 5 Number Summary for LoanAmount
  1. LoanAmount Minimum Value = 50.0
  2. LoanAmount 1st Quartile = 212.0
  3. LoanAmount Median = 368.0
  4. LoanAmount 3rd Quartile = 532.0
  5. LoanAmount Maximum Value = 699.0

# **D) Predictive Analytics**

**Summary**

The clean data which were dumped into the new CSV file after preprocessing were used to segregate the data into training and testing dataset. 20% of the data were used for testing i.e., 20% of 9623 = 1924 number of rows were used for testing and the remaining 80% i.e., 9623 – 1924 = 7699 number of rows were used for training.

These training and testing dataset were used to build two machine learning models: DecisionTreeClassifier and GaussianNB. DecisionTreeClassifier have 64% accuracy rate which indicates that the model correctly predicts the approval or rejection status for 64% of the loan applications in the dataset used for evaluation, whereas GaussianNB have 75% accuracy rate which indicates that the model correctly predicts the approval or rejection status for 75% of the loan applications in the dataset used for evaluation.

Considering this result, the Gaussian Naïve Bayes model is better suited for predicting bank loan approval. It achieves a higher accuracy compared to the Decision Tree Classifier. Additionally, the Gaussian Naïve Bayes model demonstrates better precision and recall for both approval (Y) and rejection (N) classes. Specifically, it shows higher precision and recall for the rejection class (N), which is crucial in banking scenarios to avoid approving risky loans.