**PROJECT REPORT**

**SUMMARY OF DATA CLEANING ACTIVITIES**

The data cleaning process began with splitting the dataset into numerical and categorical variables for easier handling. For the categorical data, I first checked for inconsistencies and observed that the Applicant Income and Coapplicant Income columns were incorrectly stored as string values. I converted these columns to numeric data types to allow for statistical analysis. Additionally, I removed the Loan\_ID and Applicant\_ID columns, as they are unique identifiers with no predictive value and could introduce noise into the model. Since the income columns had already been converted to numerical values, they were moved to the numerical dataset. I then examined the categorical data for missing values and found none. However, I identified 9,548 duplicate rows, which were subsequently removed to ensure data quality and model integrity.

For the numerical data, there were no duplicate entries after checks. I identified missing values in the Applicant Income and Coapplicant Income columns—211 and 184 entries respectively—amounting to roughly 2% of the numerical data. These missing values were imputed using the median rather than the mean, as the distributions of these columns were positively skewed, and the mean would have been disproportionately influenced by outliers. I then performed outlier detection on both columns using the 95% confidence interval based on the normal distribution. This revealed 155 outliers in Applicant Income and 172 in Coapplicant Income, all of which were removed to improve the robustness of the analysis. Afterward, I merged the cleaned numerical and categorical datasets. Due to prior outlier removal, the two datasets had mismatched indices, resulting in missing values, which I subsequently dropped to ensure data consistency.

Finally, I encoded the categorical variables to prepare the dataset for modeling. Binary variables such as Married, Education, Self\_Employed, Has\_CreditCard, and Loan\_Status columns were label-encoded into Boolean values. For columns with more than two categories—Gender, Owns\_Car, and Property\_Area—I applied One-Hot Encoding to convert them into multiple binary features without introducing ordinal relationships.

**PREPROCESSING**

1. The Applicant Income and Coapplicant Income were the only columns that has missing values.

Missing values for Applicant Income: 211

Missing values for Coapplicant Income: 184

1. The number of outliers for Applicant Income column: 155
2. The number of outliers for Coapplicant Income column: 172
3. The Loan\_ID and Applicant\_ID were deleted because these are unique numbers that do not contribute to the predictive power of the model.

**VISUALIZATION**

A graph of a couple of blue and pink squares

AI-generated content may be incorrect.

A graph of a couple of blue squares

AI-generated content may be incorrect.

A graph of a blue and green dotted chart

AI-generated content may be incorrect.

A graph with a blue and white line

AI-generated content may be incorrect.

A diagram with a blue and white box

AI-generated content may be incorrect.

A diagram of a box plot

AI-generated content may be incorrect.

**DESCRIPTIVE ANALYTICS**

A screenshot of a white and black text

AI-generated content may be incorrect.

**PREDICTIVE ANALYTICS**

To predict the Loan status of applicants, I developed and evaluated two machine learning models: Random Forest and Naïve Bayes. Prior to modeling, the cleaned dataset was split into training and testing sets using an 80/20 split to ensure that the models could generalize well to unseen data. Upon training and testing, the Random Forest model demonstrated a significantly higher predictive performance, achieving an accuracy of 98%, compared to 75% accuracy from the Naïve Bayes model.

In addition to accuracy, I evaluated both models using the Root Mean Squared Error (RMSE) metric. The Random Forest model achieved an RMSE of 0.15, whereas the Naïve Bayes model recorded an RMSE of 0.50—indicating a 70% reduction in prediction error when using Random Forest. This substantial difference in both accuracy and error rates clearly suggest that the Random Forest algorithm is the more effective model for predicting loan approval outcomes in this dataset. Overall, Random Forest has a better recall accuracy in predicting loan approvals than Naïve-Bayes.