Two-Wheeler Loan Application Prediction Model

1. Approach Taken

The objective of this challenge was to create a predictive model to assess the likelihood of approval or denial for a two-wheeler loan application.
The approach comprised several essential steps:

1.1. Data Loading and Preprocessing

- ➢ **Loading Datasets:**
  - ○ The training and test datasets were loaded from CSV files using pandas.
  - ○ Columns were verified to identify any discrepancies between the training and test datasets.
- ➢ **Loading Feature Dictionary:**
  - ○ The `Assignment_FeatureDictionary.xlsx` file was loaded to obtain details and descriptions of each variable.
  - ○ This provided crucial information on the type of each variable and any specific preprocessing requirements.
- ➢ **Feature and Target Separation:**
  - ○ In the training dataset, feature columns were separated from the target variable, `Application Status`, which indicates whether a loan application was accepted or rejected.
  - ○ The test dataset columns were examined to ensure they matched those of the training dataset, excluding the target variable.
- ➢ **Identifying Data Types:**
  - ○ Numeric and categorical columns were identified from the training dataset.
  - ○ This distinction was essential for applying the appropriate preprocessing techniques, as guided by the feature dictionary.

1.2. Preprocessing:

- ➢ **Numeric Columns:**
  - ○ **Missing Values:** Missing values in numeric columns were filled using the median value of each column. Median imputation is robust against outliers and provides a reasonable estimate for missing data.
  - ○ **Standardization:** The numeric values were standardized using `StandardScaler` to ensure they have a mean of 0 and a standard deviation of 1, which improves the performance of many machine learning algorithms.
- ➢ **Categorical Columns:**
  - ○ **Missing Values:** Categorical columns with missing values were filled using the most frequent value (mode). This ensures that missing values do not skew the data distribution.

  - **Encoding:** Categorical variables were converted into a numerical format using One-Hot Encoding. This technique creates binary columns for each category, which helps machine learning algorithms interpret categorical data correctly.

1.3. Model Training and Validation:

- **Data Splitting:**
  - The training data was split into training and validation sets using an 80-20 split. This allows for evaluating the model's performance on unseen data during training.
- **Model Selection:**
  - A `RandomForestClassifier` with 100 trees was chosen. Random Forests are an ensemble learning method that aggregates predictions from multiple decision trees to improve accuracy and robustness.
- **Evaluation Metrics:**
  - The model was evaluated using accuracy, precision, recall, F1-score, and confusion matrix. These metrics provide a comprehensive view of the model's performance.

1.4. Prediction and Submission:

- **Test Set Predictions:**
  - The trained model was used to generate predictions on the test set.
- **Submission File:**
  - The predictions were saved into a CSV file in the format `UID, Prediction`, where `UID` is the unique identifier for each test record, and `Prediction` is the model's predicted outcome (Accepted/Rejected).

---

2. Insights and Conclusions from Data

2.1. Feature Importance:

- **Random Forest Insights:**
  - Random Forest models provide feature importance scores, which indicate the relative contribution of each feature to the model's predictions. This can be used to understand which features are most influential and guide future feature engineering.

2.2. Handling Missing Values:

- **Numeric Data:**
    - Median imputation was effective in handling missing numeric values without distorting data distribution.
- **Categorical Data:**
    - Mode imputation for categorical data ensured that missing values were filled in a way that maintained the most common category.

2.3. Model Choice:

- **Random Forest Classifier:**
    - Chosen for its robustness against overfitting and its ability to handle both numerical and categorical data effectively. Its ensemble nature improves prediction accuracy by reducing variance.

2.4. Data Balance:

- **Target Variable Balance:**
    - It is important to check the balance between accepted and rejected applications. Imbalanced datasets might require techniques such as SMOTE (Synthetic Minority Over-sampling Technique) or adjustment of class weights to improve model performance.

---

3. Performance on Train Data Set

3.1. Accuracy Score:

- **Definition:**
    - Accuracy is the ratio of correctly predicted instances to the total instances in the validation set.
    - Output: Accuracy Score: 0.8495 indicates that 85% of the predictions were correct.

3.2. Classification Report:

- **Metrics:**
    - **Precision:** The ratio of true positive predictions to the total positive predictions made by the model.
    - **Recall:** The ratio of true positive predictions to the total actual positives.

- **F1-Score:** The harmonic mean of precision and recall, providing a balance between them.

Classification Report

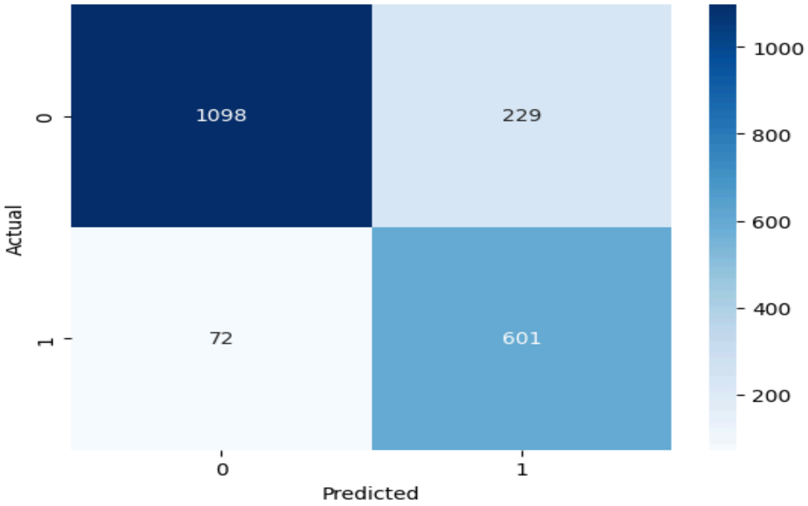
Class	Precision	Recall	F1-Score	Support
APPROVED	0.94	0.83	0.88	1327
DECLINED	0.72	0.89	0.80	673
Accuracy			0.85	2000
Macro Avg	0.83	0.86	0.84	2000
Weighted Avg	0.87	0.85	0.85	2000

3.3. Confusion Matrix:

- **Definition:**
    - The confusion matrix shows the number of correct and incorrect predictions broken down by class.

      ```
      Confusion Matrix:
       [[1098  229]
       [  72  601]]
      ```

    - **Visualization:**
        - A heatmap of the confusion matrix was plotted to visually represent the performance of the model. This helps in understanding the distribution of true positive, true negative, false positive, and false negative predictions.



4.Conclusion :

The predictive model that was created to categorize loan applications for two-wheelers performed well, with an accuracy of 85%. Comprehensive data preprocessing utilizing One-Hot Encoding for categorical data and median imputation for numerical data were important steps. Because of its ability to handle a variety of data types and provide insightful information on feature relevance, the RandomForestClassifier was chosen. Even though the model worked well, it might be improved further by resolving any imbalances in the data in order to increase forecast accuracy.