# Documentation

## **Data Preprocessing**

### **Steps Taken:**

1. **Data Loading:** Loaded the training and test datasets using `pandas`.
2. **Handling Missing Values:** Checked for and handled any missing values in the datasets.
3. **Feature Engineering:** Created additional features or modified existing ones to enhance model performance.
4. **Label Encoding:** Converted categorical variables into numerical format using label encoding.

## **Model Selection**

### **Models Chosen:**

1. **Logistic Regression:** Selected for its simplicity and interpretability.
2. **Decision Tree:** Chosen for its ability to capture non-linear relationships.
3. **Random Forest:** Selected to address overfitting issues seen in Decision Trees.
4. **Gradient Boosting:** Chosen for its strong predictive performance.

### **Reasoning:**

1. Logistic Regression provides a good baseline with interpretable coefficients.

2. Decision Trees are easy to visualize and understand, making them useful for identifying key decision points.

3. Random Forests improve Decision Trees by averaging multiple trees to reduce variance.

4. Gradient Boosting iteratively improves model performance by focusing on errors of previous models.

## **Hyperparameter Tuning**

### **Steps Taken:**

1. **Grid Search:** Used `**GridSearchCV**` to find the best hyperparameters for each model.
2. **Cross-Validation:** Employed 5-fold cross-validation to ensure the models generalize well to unseen data.

## **Model Evaluation**

**Evaluation Metrics:**

**Accuracy:** Primary metric used to evaluate model performance.

**Cross-Validation Scores:** Used to assess model stability and generalization.

## **Prediction and Submission**

**Steps Taken:**

**Prediction:** Used the best-performing model to make predictions on the test set.

**Submission Preparation:** Created a DataFrame for submission and saved it as an Excel file.