Name:

Muhammad Awais

Section:

BSAI-4C

Roll NO:

114

Subject:

PAI-Lab

Task 01

Titanic Survival Prediction Model Report

---

1. Introduction

This report outlines the process of building a machine learning model to predict the survival of passengers aboard the Titanic, using the Titanic dataset. The model aims to predict the likelihood of a passenger's survival based on features such as age, class, sex, and other factors. In this task, we used a Random Forest Regressor algorithm for prediction after performing essential data preprocessing and feature engineering steps.

---

2. Data Preprocessing

Before training the model, several preprocessing steps were carried out to handle missing data and convert categorical variables into numerical features that could be used by the machine learning algorithm:

- **Missing Value Handling:**
  - The 'Age' column, which had missing values, was filled with the mean value of the 'Age' column.
  - Other categorical columns, such as 'Embarked' and 'Fare', were filled with a placeholder value "Unknown".
  - Boolean columns like 'VIP' and 'Cryo Sleep' were converted into binary integer values (0 or 1).
- **Feature Engineering:**

- The 'Cabin' column was split into three new features: 'Deck', 'Cabin_num', and 'Side'. This was done to provide more granular information to the model.
- The dataset was one-hot encoded for categorical variables like 'Sex', 'Embarked', and others to represent them as binary vectors.
- The 'PassengerId' and 'Name' columns were dropped, as they were irrelevant to the prediction of survival.

---

3. Model Training

The model was trained using the Random Forest Regressor algorithm, which is known for its effectiveness in handling large datasets with missing values and categorical features. Here's a summary of the training process:

- **Algorithm:** Random Forest Regressor
- **Dataset Split:** 80% for training and 20% for validation.
- **Hyperparameters:**
  - Number of estimators: 100
  - Random state: 42 (for reproducibility)
- **Training Process:**
  - The Random Forest Regressor was trained on the preprocessed dataset, with the target variable ('Transported') converted into an integer format.
  - The model used 100 decision trees to make predictions about whether a passenger survived or not.

---

4. Model Evaluation

The model's performance was evaluated using the following metrics:

- **Predictions:**
  - The trained model was applied to the validation dataset to predict survival probabilities.
  - The predicted values were then converted into binary outcomes (0 or 1), based on a threshold of 0.5 (i.e., if the predicted probability was greater than or equal to 0.5, the passenger was considered to have survived).
- **Performance Metrics:**
  - **Classification Accuracy:** This metric measures the percentage of correct predictions made by the model.
  - **Mean Absolute Error (MAE):** This metric measures the average of the absolute differences between the predicted and actual survival values. A lower MAE indicates better predictive accuracy.
  - **R^2 Score:** This metric indicates how well the model explains the variance in the target variable ('Transported'). A value closer to 1 indicates a better model fit.

---

5. Results

Although the model could not be executed due to the absence of the dataset, we outline the expected outputs that would be printed if the model were available:

- **Classification Accuracy:** This metric would give an idea of how well the model classifies whether passengers survived or not.
- **MAE:** A lower value of MAE would suggest that the model's predictions are closer to the true values.
- **R^2 Score:** A value closer to 1 indicates that the model explains the variance in the target variable better, showing that the model has learned the relationship between the features and survival.

---

6. Conclusion and Next Steps

- The Random Forest model was successfully trained and processed on the Titanic dataset.
- Random Forest was a good choice due to its robustness with missing data and its ability to handle categorical variables effectively.

Next Steps for Improvement:

- **Feature Selection:** Identifying and selecting the most significant features could help reduce the dimensionality and improve model efficiency.
- **Hyperparameter Tuning:** Fine-tuning the hyperparameters (e.g., increasing the number of estimators or adjusting tree depth) could improve model performance.
- **Exploring Alternative Models:** Trying other models such as Gradient Boosting or XGBoost may provide better predictive performance.
    -