Project Description: Boston Housing Price Prediction

Overview

This project focuses on building and evaluating machine learning models to predict house prices using the Boston Housing dataset. The dataset contains various features about houses and neighborhoods, and the target variable is the median value of owner-occupied homes (MEDV).

Key Steps & Components in the Project

1. Data Loading and Preparation
- The dataset is loaded into a Pandas DataFrame.
- Features (x) and target (y) are separated; the target is the MEDV column.
- Data is split into training and testing sets using train_test_split. Stratification is done based on theCHAS feature

  to preserve its distribution.

2. Feature Engineering: Pipelines and ColumnTransformer -

Features are split into numerical and categorical columns.

- Two pipelines are created:
- Numerical pipeline: Imputes missing values using the median and scales features using StandardScaler.
- Categorical pipeline: Imputes missing values using the most frequent category.
- These pipelines are combined using ColumnTransformer.
- The full preprocessing pipeline is created using Pipeline.

3. Model Training
- The training data is transformed using the preprocessing pipeline.
- Three models are trained and evaluated:
- Linear Regression
- Decision Tree Regressor
- Random Forest Regressor
- Performance metrics: MAE, RMSE, and R^2 score.

4. Cross-Validation
- 10-fold cross-validation is performed on Linear Regression and Random Forest models.
- Negative mean squared error is used, converted to RMSE.
- Mean and standard deviation of RMSE scores are printed.

5. Model Saving and Loading
- Random Forest model is saved using joblib.
- It can be loaded later for prediction without retraining.

6. Final Model Testing
- The test data is preprocessed using the same pipeline.
- Final evaluation metrics (MSE and RMSE) are calculated.

7. Prediction on New Sample

- Demonstrates how to use the saved model on a new sample.
- Input must be preprocessed similarly.
-

Important Points to Highlight

- Data Preprocessing Pipelines ensure consistent and reproducible steps.
- Model Comparison helps in selecting the best-performing model.
- Cross-Validation provides model robustness.
- joblib saves time by persisting models.
- RMSE, MAE, and R^2 offer interpretable evaluation.
- Stratified Splitting retains key categorical distribution.
- Reusability is enabled through the saved model and pipeline.