Boston Housing Price Prediction

Introduction

The objective of this project is to predict house prices in Boston using the Boston Housing Dataset. This involves various steps, including data collection, data preparation, exploratory data analysis (EDA), feature engineering, model training and evaluation, and model interpretation.

The Dataset was found in Kaggle. Originally, it was a built-in dataset in scikit-learn, but was removed for reasons best known to them. The Boston Housing Dataset contains information collected by the U.S Census Service concerning housing in the area of Boston, Massachusetts. The dataset has 506 samples and 14 feature variables. The target variable is the median value of owner-occupied homes in $1000s.

Report

Phase 1 – Data Collection and Preparation

The dataset was sourced from https://www.kaggle.com/datasets/arunjangir245/boston-housing-dataset. The dataset was loaded into a Pandas DataFrame for efficient manipulation and analysis. An initial inspection was performed to check for missing values and any inconsistencies. Upon inspection, no missing values were found in the dataset, no duplicates were found in the dataset also, all data types were correct.

Phase 2 – Exploratory Data Analysis (EDA)

A group by function was used to check both the mean and the count of some features by the target variable. Summary statistics were computed to understand the basic characteristics of the dataset. This included measures of central tendency and dispersion for both the features and the target variable and also correlation or between a feature column and the target variable. Scatter plots were created for each feature against the target variable to visually inspect any linear or non-linear relationships. Box plots were also used to identify potential outliers and understand

the distribution of the data points. Outliers were identified using the Interquartile Range (IQR) method. A module was created to view outliers, visualize outliers using boxplots, and also to remove outliers; but for this project, I replaced the outliers with the mean of the outlier column. These outliers were subsequently removed to ensure they did not adversely affect the model performance greatly.

Phase 3 - Feature Engineering

New features were engineered to capture additional information that could improve the model's performance. For instance, a feature representing the ratio of tax to the average number of rooms (TAX_RM) was created. The dataset did not contain any categorical variables. Thus, this step was not applicable. If there had been categorical variables, techniques such as one-hot encoding or label encoding would have been used. To ensure that all features contributed equally to the model, numerical features were standardized. This involved scaling the features to have a mean of zero and a standard deviation of one.

Phase 4 – Model Training and Evaluation

The dataset was split into training and testing sets to evaluate the model's performance. An 80-20 split was used, with 80% of the data used for training and 20% reserved for testing with no random state. Several machine learning algorithms were selected for comparison: Linear Regression, Decision Tree, Random Forest, Gradient Boosting, KNeighbors and Support Vector. These models were chosen based on their varying complexities and predictive power. A dictionary was further created to host the model with a key representing the name of the model and the item representing the model itself. Each model was trained on the training set and evaluated on the test set using a for loop. Performance metrics such as Mean Squared Error (MSE), Mean Absolute Error (MAE), and R^2 score were used to compare the models also in the same loop. Another set of dictionaries were created which had a dictionary in it. The inner dictionary has the hyperparameter tuning in it, while the main dictionary has everything in it, including the model. These hyperparameter tuning was run in a loop of which a best performing model was eventually gotten.

Phase 5 – Model Interpretation and Reporting

A brief summary of my work was given alongside some visualization