BOSTON HOUSE PRICES ANALYSIS

Project Overview

The objective of this capstone project is to develop a machine learning model to predict house prices using the Boston Housing dataset. This project involves data cleaning, exploratory data analysis (EDA), feature engineering, model training, and evaluation.

Dataset Description

The Boston Housing dataset contains various features related to houses in Boston and their corresponding prices (MEDV). The dataset includes the following columns:

- CRIM: Per capita crime rate by town
- ZN: Proportion of residential land zoned for lots over 25,000 sq. ft.
- INDUS: Proportion of non-retail business acres per town
- CHAS: Charles River dummy variable (1 if tract bounds river; 0 otherwise)
- NOX: Nitric oxides concentration (parts per 10 million)
- RM: Average number of rooms per dwelling
- AGE: Proportion of owner-occupied units built prior to 1940
- DIS: Weighted distances to five Boston employment centers
- RAD: Index of accessibility to radial highways
- TAX: Full-value property tax rate per $10,000
- PTRATIO: Pupil-teacher ratio by town
- B: 1000(Bk−0.63)21000(Bk - 0.63)^21000(Bk−0.63)2 where BkBkBk is the proportion of black people by town
- LSTAT: Percentage of lower status of the population
- MEDV: Median value of owner-occupied homes in $1000s

Phase 1: Data Collection and Preparation

The dataset used for this project, titled "The Boston Houseprice Data," was downloaded from Kaggle. The specific dataset can be found at this link.

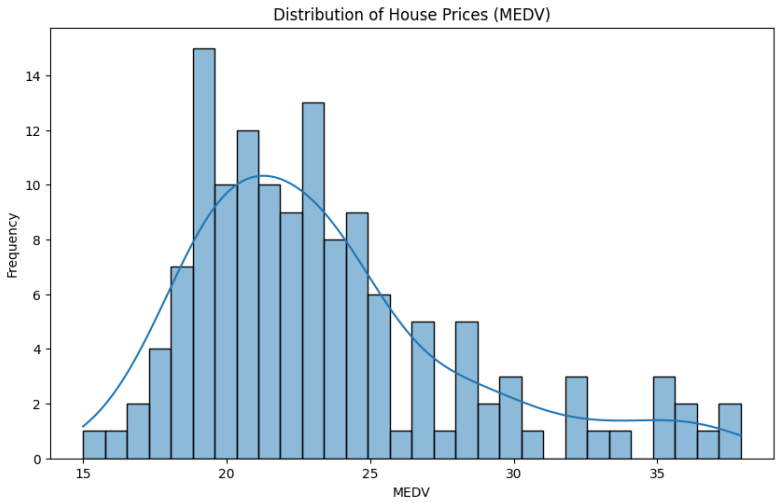
Once downloaded, the dataset was loaded into a Pandas DataFrame for ease of manipulation and analysis. Pandas is a powerful data analysis library in Python, and it provides various functions to inspect, clean, and preprocess the data effectively.

Upon loading the dataset, a thorough inspection was conducted to check for any missing values or anomalies. This inspection included checking the data types, identifying any null values, and understanding the basic statistics of each column.

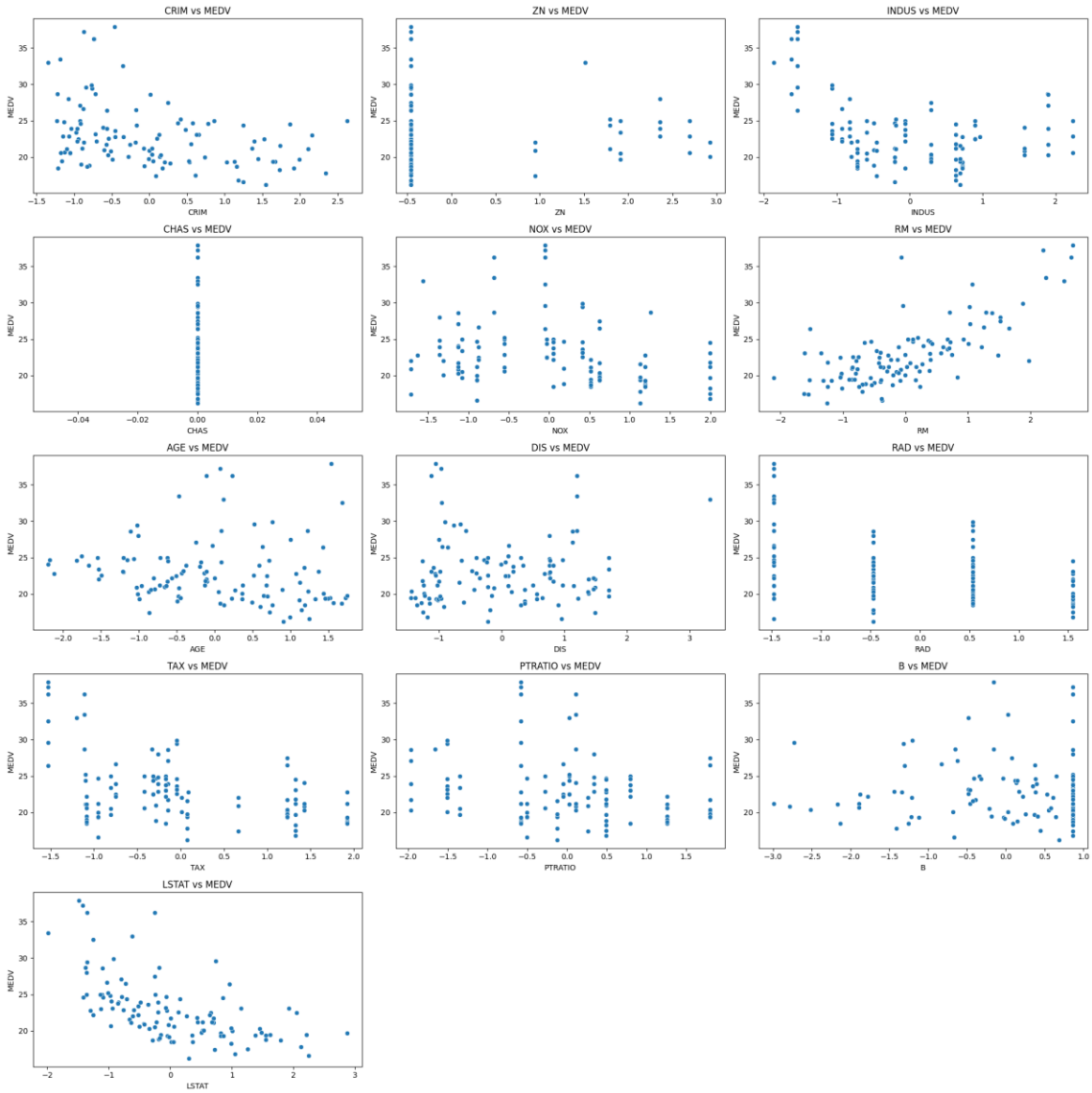
Fortunately, the dataset from Kaggle was already clean and in the right format upon inspection. This means there were no missing values, and all data points were within expected ranges. Thus, no additional data cleaning steps were necessary.

Phase 2: Exploratory Data Analysis (EDA)

In the Exploratory Data Analysis (EDA) phase, I delved into understanding the distribution of features and the target variable, house prices (MEDV). The dataset was inspected for any anomalies, and visualizations were created to better grasp the relationships between different features and house prices. I observed that the MEDV values were roughly normally distributed but had a slight skew to the right.



Key features such as RM, which indicates the average number of rooms per dwelling, showed a strong positive correlation with house prices. Similarly, LSTAT, representing the percentage of the population with lower socioeconomic status, and PTRATIO, the pupil-teacher ratio by town, exhibited significant correlations with house prices.



Scatter plots, histograms, and box plots were employed to visualize these relationships, making it evident how these variables influence house prices. Furthermore, I identified and addressed outliers in the dataset, ensuring that our analysis was robust and the data was clean for subsequent modeling steps.

Phase 3: Feature Engineering

In the Feature Engineering phase, I focused on enhancing the dataset to improve the model's performance. This involved creating new features that could provide additional insights and better predictive power. I also encoded categorical variables using appropriate techniques; specifically, the CHAS variable, which indicates proximity to the Charles River, was encoded effectively to ensure the

model could interpret this information. Furthermore, numerical features were normalized or standardized as necessary. This standardization process was crucial as it improved the model's performance by ensuring that all features contributed equally to the prediction, avoiding any bias due to differing scales.

Phase 4: Model Training and Evaluation & Phase 5: Model Interpretation and Reporting

In the Model Training and Evaluation phase, I began by splitting the dataset into training and testing sets to ensure that the model's performance could be evaluated on unseen data. I chose several machine learning algorithms, including Linear Regression, Decision Tree, Random Forest, and Gradient Boosting, to determine which model would best predict house prices. Each model was trained and evaluated using metrics such as Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), and R^2. Hyperparameter tuning was performed to optimize each model's performance.

After thorough evaluation, the Random Forest Regressor emerged as the best-performing model with an RMSE of 2.901809, MAE of 2.204227, and R^2 of 0.614449. The Gradient Boosting Regressor also showed competitive performance with an RMSE of 2.666115, MAE of 2.091347, and R^2 of 0.674537. Although the Decision Tree and Linear Regression models were less accurate, they provided valuable insights into the data structure.

In the Model Interpretation and Reporting phase, I interpreted the results of the best-performing model, focusing on the importance of critical features. The most significant features in predicting house prices were RM (average number of rooms per dwelling), LSTAT (percentage of lower status population), PTRATIO (pupil-teacher ratio), CRIM (per capita crime rate), and TAX (full-value property tax rate). Visualizations, including scatter plots, histograms, box plots, and feature importance plots, were created to support our findings and enhance the model interpretations.

Recommendations

- Further improvements could be achieved by exploring additional feature engineering techniques and trying advanced algorithms like XGBoost or Neural Networks.

- Regularization techniques like Lasso or Ridge Regression could be explored to improve the model's performance and handle multicollinearity.

Model Performance Summary

Model	RMSE	MAE	R^2
Linear Regression	2.526514	2.064111	0.707728
Decision Tree	3.149675	2.622727	0.545770
Random Forest	2.901809	2.204227	0.614449
Gradient Boosting	2.666115	2.091347	0.674537

Best Random Forest Performance

- RMSE: 2.987628
- MAE: 2.292419
- R^2: 0.591307