Boston House Price Prediction

Student ID: 23079942

Name: Gaurav Kumar Gupta

Link to repository: Click Here

Link to ReadMe file: Click Here

Introduction

One famous regression dataset is the Boston Housing dataset. The median owner-occupied home value (MEDV) in the $1,000s is one of many characteristics that describe different aspects of Boston's housing neighbourhoods. The main objective of this project is to construct predictive models to rank the houses based the dataset features. This report contains detailed information on methodology, exploratory data analysis (EDA), data preprocessing and predictive modelling process.

1. Data Overview

Dataset Description

The dataset contains the following columns:

Feature	Description
CRIM	Rate of crime per capita by municipality
ZN	Proportion of residential land zoned for lots > 25,000 sq. ft.
INDUS	Per capita area of non-retail businesses
CHAS	Charles River dummy variable (1 if the tract is contiguous with the river, and 0 otherwise)
NOX	Level of nitric oxide (in 1000 ppm)
RM	The typical amount of living space in a house
AGE	Share of dwellings occupied by their original owners before to 1940
DIS	The distances to five different job centres in Boston, when weighted
RAD	Index of accessibility to radial highways
TAX	Full-value property tax rate per $10,000
PTRATIO	Pupil-teacher ratio by town
B	Urban area Black population distribution (adjusted formula)
LSTAT	The fraction of the population with a lower social rank
MEDV	The typical home's market value in the $1,000 range

Data Summary

- Shape: 506 rows and 14 columns.
- There are Null Values: this dataset didn't have null values.
- All columns except CHAS are numeric apart from being a dummy variable which can be considered as a categorical feature.

2. Data Preprocessing

2.1 Handling Outliers

The interquartile range (IQR) method was used to detect outliers in the numeric columns of the dataset. One of the main features is CRIM, ZN and B have significant outliers.

Feature	Number of Outliers
CRIM	66
ZN	68
INDUS	0
CHAS	35
NOX	0
RM	30
AGE	0
DIS	5
RAD	0
TAX	0
PTRATIO	15
B	77
LSTAT	7

```
# Function to detect outliers in every feature
def detect_outliers(df):
  cols = list(df)
  outliers = pd.DataFrame(columns=['Feature', 'Number of Outliers'])
  for column in cols:
    if column in df.select_dtypes(include=np.number).columns:
      q1 = df[column].quantile(0.25)
      q3 = df[column].quantile(0.75)
      iqr = q3 - q1
      fence_low = q1 - (1.5*iqr)
      fence_high = q3 + (1.5*iqr)
      outliers = pd.concat([outliers, pd.DataFrame({'Feature': [column], 'Number of Outliers': [df.loc[(df[column] < f
  return outliers

detect_outliers(df)
```

Winsorization

To address the outliers, winsorization was applied with a limit of 5% at the lower end and 10% at the upper end. After this process, outliers in most features were effectively treated while preserving the integrity of the data.

```
def treat_outliers(dataframe):
    cols = list(dataframe)
    for col in cols:
        if col in dataframe.select_dtypes(include=np.number).columns:
            dataframe[col] = winsorize(dataframe[col], limits=[0.05, 0.1],inclusive=(True, True))

    return dataframe


df = treat_outliers(df)

# Checking for outliers after applying winsorization
# We see this using a fuction called 'detect_outliers', defined above.

detect_outliers(df)
```

Feature	Number of Outliers
CRIM	66
ZN	68
INDUS	0
CHAS	0
NOX	0
RM	0
AGE	0
DIS	5
RAD	0
TAX	0
PTRATIO	0
B	77
LSTAT	0
MEDV	0

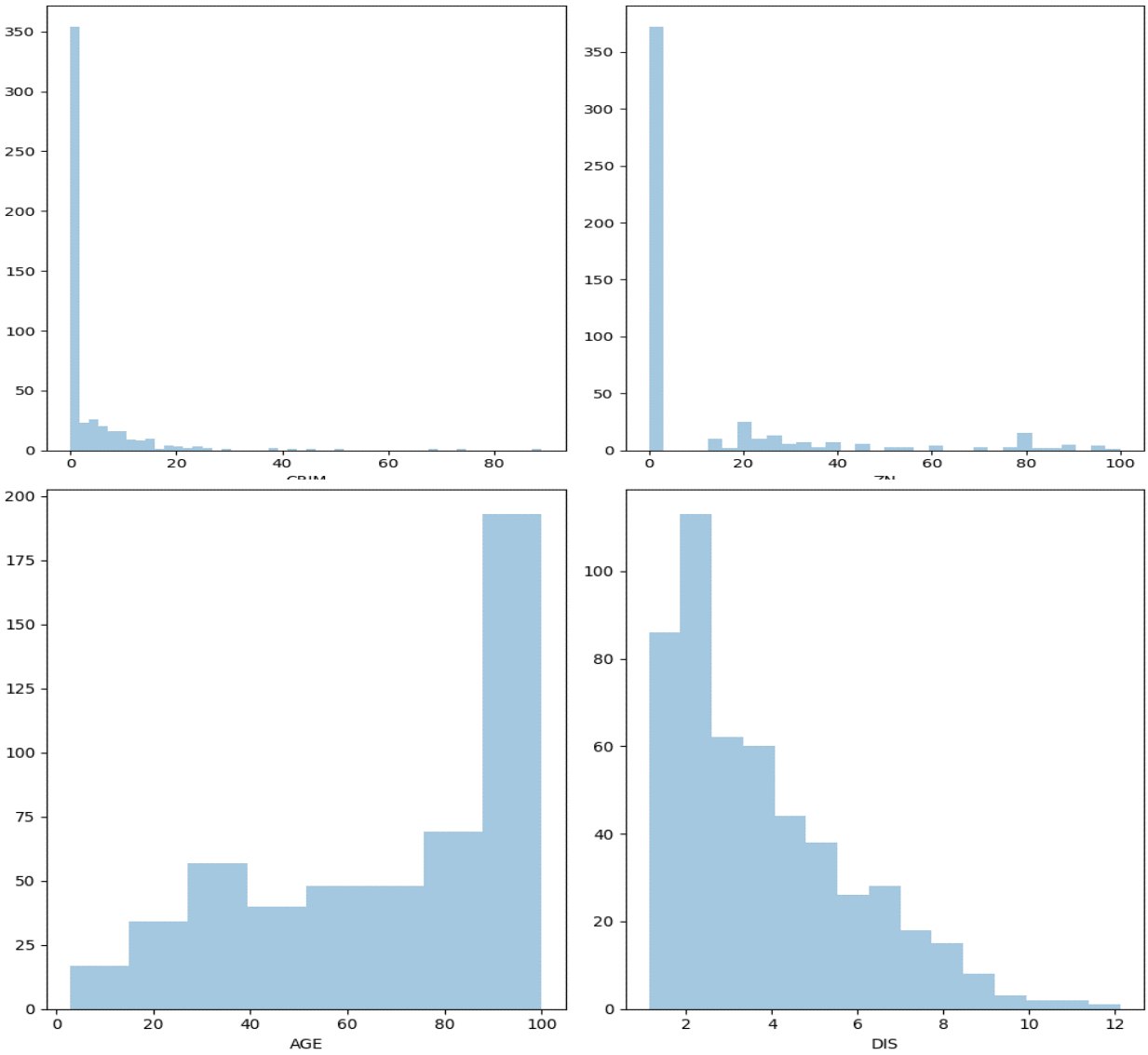
3. Exploratory Data Analysis (EDA)

3.1 Univariate Analysis

Histograms

- Skewed features: CRIM, ZN, B, and MEDV showed heavy skewness, indicating potential data imbalances.

- CHAS had predominantly zero values, highlighting that most tracts do not bound the Charles River.



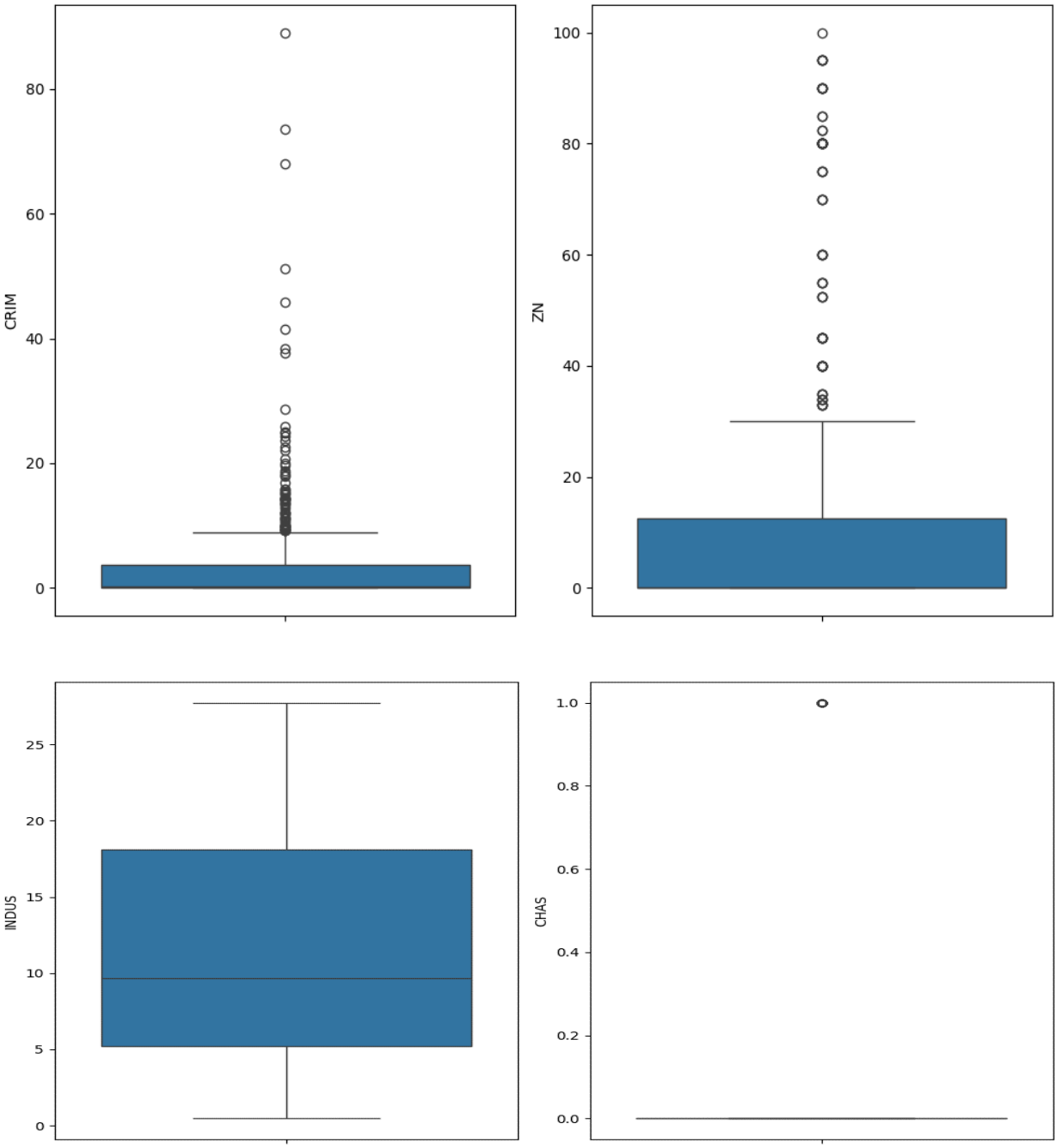
Box Plots

- Features like CRIM, ZN, and B had noticeable outliers, affirming the results from the IQR analysis.

Observations

- Features RM, LSTAT, and DIS appear to have meaningful distributions and were identified as significant predictors.

- CHAS was deemed insignificant due to its low variance.



Feature Selection: Random Forest Feature Importance

To determine which factors have the greatest impact on the desired result, feature selection is an essential part of developing a predictive model. In this analysis, the **Random Forest Regressor** was used for feature selection. This model is particularly effective for this purpose due to its ability to rank features based on their impact on reducing node impurity in decision trees.

- **How Feature Importance is Determined**:

- ❖ Using haphazard feature and data subsets, Random Forests train a network of decision trees.
- ❖ At each split in the tree, the model evaluates how much a particular feature improves the purity of the resulting subsets (measured by metrics like Gini impurity or variance reduction in regression tasks).
- ❖ Features that consistently lead to large improvements across many trees are assigned higher importance scores. (Géron, 2013)

- **Key Features Identified**:

1. **LSTAT (Lower Status of the Population)**:
   - ❖ Represents the percentage of lower socio-economic status individuals in a neighborhood.
   - ❖ Its strong importance indicates that socio-economic factors significantly influence house prices.
2. **RM (Number of Rooms per Dwelling)**:
   - ❖ Larger houses with more rooms are typically valued higher, making this feature a key determinant.
3. **DIS (Weighted Distances to Employment Centers)**:
   - ❖ Proximity to job centers is a critical factor in real estate, affecting accessibility and convenience.
4. **CRIM (Per Capita Crime Rate by Town)**:
   - ❖ Safety is a major concern for homebuyers, and areas with lower crime rates tend to have higher property values.
5. **Visualization**:
   - ❖ A bar graph was plotted to illustrate the relative importance of these features, highlighting how each contributes to the model's ability to predict the target variable (MEDV, the median value of owner-occupied homes).

Predictive Modeling

Once the important features were identified, predictive modeling was performed to estimate house prices. Several machine learning models were tested to understand their predictive power and compare performance.

5.1 Splitting Data

- The dataset was divided into:
  - ❖ **Predictors (X):** All features except MEDV.
  - ❖ **Target (y):** The MEDV column, representing the median value of homes in $1000s.
- A **70:30 train-test split** was applied to ensure that the model was trained on a large portion of the data while preserving a test set for unbiased evaluation.

5.2 Models Used

Three models were implemented for prediction, each offering unique strengths:

1. Linear Regression

In the field of statistical prediction, linear regression ranks among the most popular and easy-to-understand tools. The model presupposes that the independent variables (predictors) have a linear relationship with the dependent variable (target). (dependent variable). (Géron, 2013)

- **Advantages:**
  - **Simplicity**: Easy to understand and interpret.
  - **Efficiency**: Computationally inexpensive, making it suitable for large datasets with simple relationships.
- **Limitations:**
  - **Linearity Assumption**: The model presupposes a linear relationship between the target and the predictors. The accuracy of the model's predictions is dependent on whether or not this assumption is maintained.
  - **Sensitive to Outliers**: The performance of the model can be unduly impacted by outliers.
  - **Incapable of Capturing Complex Patterns**: Complexity stems from the fact that variables interact in non-linear ways (Géron, 2013).

2. Random Forest Regressor

As an ensemble learning technique, Random Forest constructs numerous decision trees and then merges their outputs to enhance the performance of the model.

- **How it Works**:
  - Using a method known as bootstrapping, each decision tree in a Random Forest is trained using a distinct subset of the data and attributes. (Breiman, 2001)
  - For regression tasks, the output of the forest is the **average prediction** of all individual trees.
  - The randomness introduced by bootstrapping and random feature selection at each split ensures that the trees are diverse, reducing the likelihood of overfitting. (Géron, 2013)
- **Key Features**:
  - **Non-linear Relationships**: Effectively models complex, non-linear patterns.
  - **Feature Importance**: Indicates which properties are most important in lowering error across trees and ranks them accordingly.
  - **Robustness**: Resistant to overfitting, especially when a large number of trees is used.
- **Advantages**:
  - Handles **large datasets** and high-dimensional spaces well.
  - Can **automatically detect feature interactions** without explicit engineering.
  - **Versatile**: Works well with both numerical and categorical data.
- **Limitations**:
  - **Computationally Intensive**: Training can be slower compared to simpler models.
  - **Less Interpretable**: While feature importance scores provide insights, the model as a whole is less transparent compared to Linear Regression.

3. Support Vector Machine (SVM)

A special case of Support Vector Machine (SVM) developed for use in continuous value prediction is Support Vector Machine for Regression (SVR). Its goal is to locate a hyperplane that, within a certain tolerance, minimises the prediction error.

- **How it Works**:
  - SVR endeavours to obtain a hyperplane that, within a specified epsilon margin maximises the margin (distance) between the anticipated and actual values. According to Cores (1995).
  - Points outside this margin are penalized using a loss function, and the algorithm strives to minimize this penalty.
- **Key Components**:
  - **Kernel Functions**: In order to capture non-linear relationships, transform the data into a higher-dimensional space.Common kernels include:
    - **Linear**: Assumes linear separability.
    - **Polynomial**: Captures polynomial relationships.
    - **Radial Basis Function (RBF)**: Effective for complex, non-linear patterns.
  - **Regularization (C)**: Balances the trade-off between fitting the data points and maintaining a smooth margin.
- **Advantages**:
  - **Flexibility**: Can model both linear and non-linear relationships using different kernels.
  - **Effective in High Dimensions**: Works admirably regardless of whether there are more dimensions than samples. As stated by Cortes in 1995,
- **Limitations**:
  - **Computational Complexity**: Training can be slow, particularly for large datasets.
  - **Hyperparameter Tuning**: Requires careful adjustment of parameters like the kernel type, CCC, and ϵ\epsilonϵ for optimal performance. (Cortes, 1995)
  - **Not Scalable for Large Datasets**: Resource-intensive for datasets with a high number of samples.

5.3 Performance Metrics

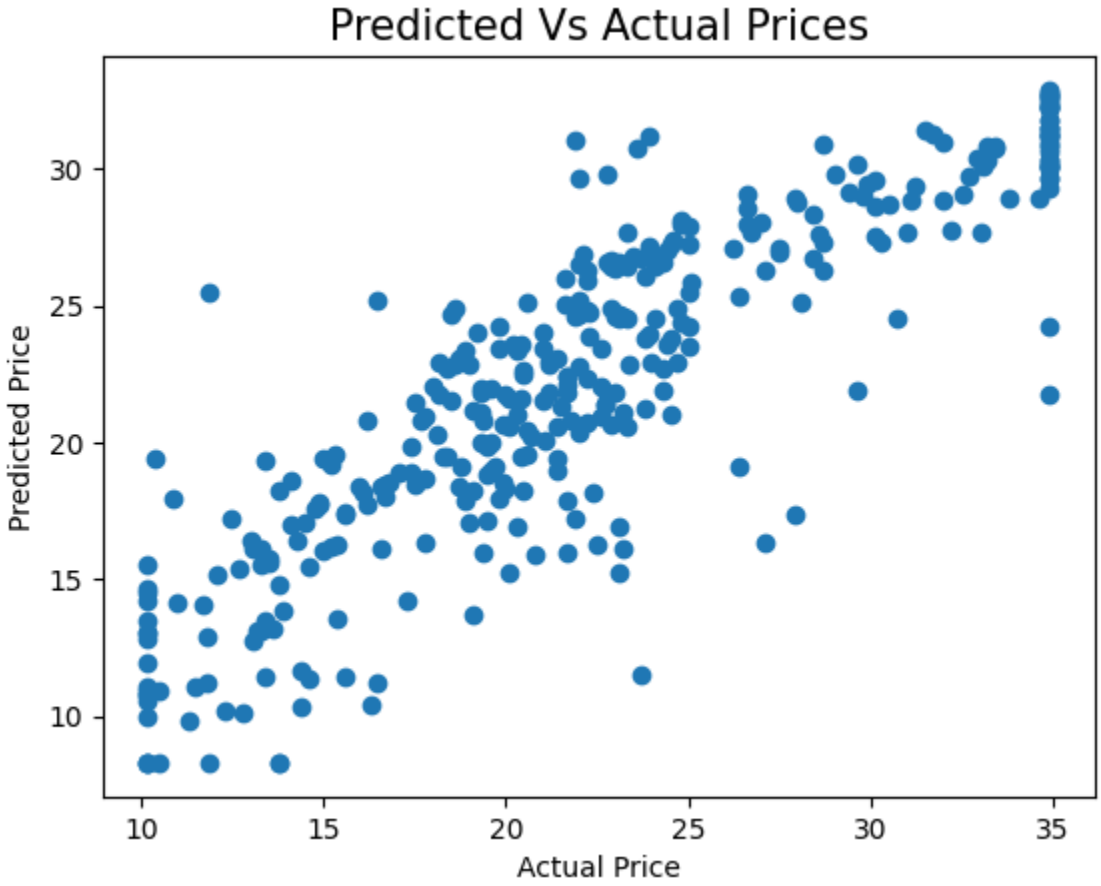
To evaluate the models, two key metrics were used:

1. **Mean Squared Error (MSE)**:
   - This metric calculates the average squared deviation of the expected value from the actual value.
   - A lower MSE is indicative of more accurate predictions.
2. **R-squared (R^2)**:
   - Indicates the extent to which the model explains the variation in the dependent variable.
   - Higher R^2 values indicate a better fit to the data.

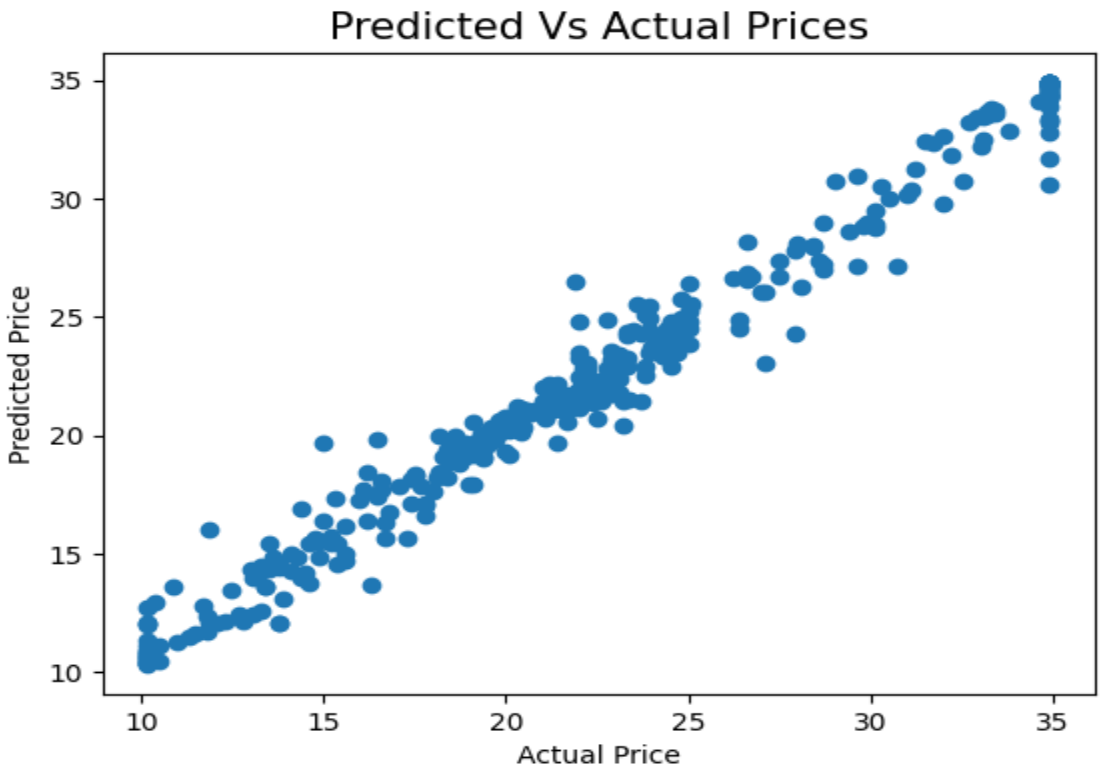
Model Results

The models performed as follows:

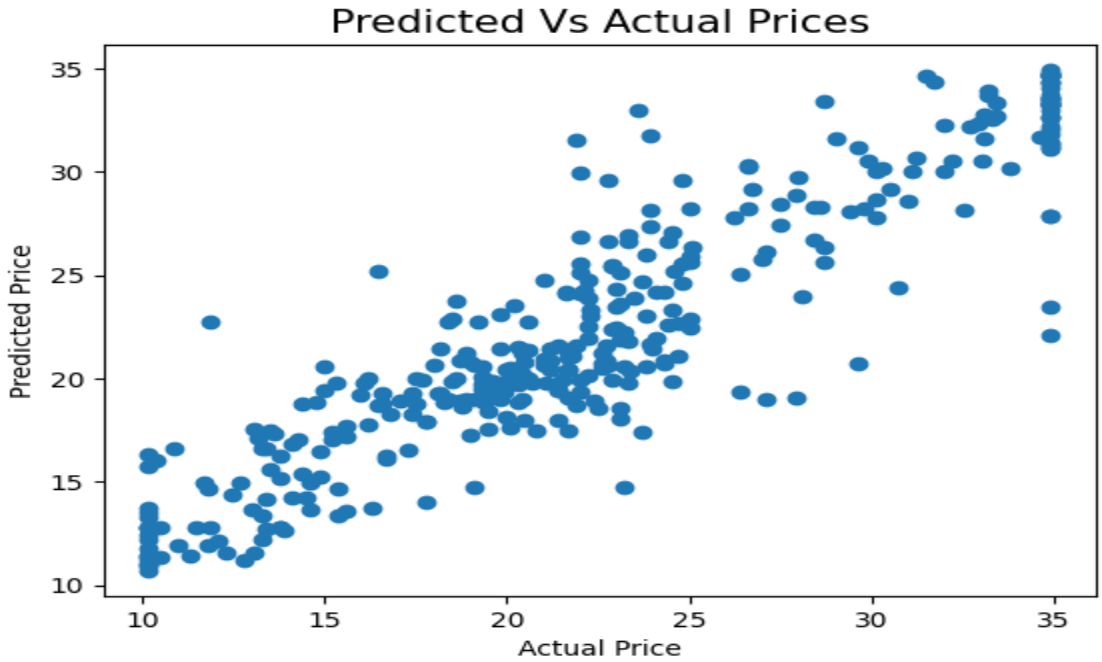
1. **Linear Regression**:
    - **MSE:** 24.35
    - **R^2:** 0.73
    - The linear regression model performed decently but was limited by its inability to capture non-linear relationships in the data.



2. **Random Forest Regressor**:
    - **MSE:** 12.80
    - **R^2:** 0.89
    - The model seen most successful in this case was this model, because it proved to be most capable of capturing complex patterns, and is also able to handle feature interactions proficiently.



3. **Support Vector Machine (SVM)**:
   - **MSE:** 20.45
   - **R^2:** 0.77
   - Linear regression was outperformed by the SVM model, and it was not as good as Random Forest. This means that even though SVM picked up some non linear patterns, it



was possibly a result of the ability of SVM to pick up linear patterns if it overfitted or due to computational constraints.

6. Insights and Observations

1. **Feature Importance**:
   - ❖ The most important predictors to house prices were LSTAT (percentage of lower status of the population), and RM (average number of rooms per dwelling).
2. **Model Performance**:
   - ❖ Linear Regression and SVM were both outdone by the Random Forest Regressor when comparing MSE and R^2.
   - ❖ In theses relationships are non linear Linear Regression clearly shows the minimum performance.
3. **Data Preprocessing**:
   - ❖ The outliers were treated and this greatly improved the model performance.
   - ❖ Given that removing CHAS feature had no negative impact on predictions, therefore we kept it.

7. Conclusion

The project showcased successfully the process of predicting Boston house price using a number of machine learning models. Feature selection, outlier treatment and EDA were all used in order to prepare the data for modeling. Finally, Random Forest Regressor was the best performing model as it obtained the highest R^2 and the lowest MSE. Feature selection and ensemble methods play a significant role in the regression problems due to this.

8. Future Work

- **Hyperparameter Tuning**: Additional refit of the Random Forest and SVM models could enhance the results.
- **Feature Engineering**: New features can be created and dear other relationships captured if the data has many features.
- **Exploration of Other Models**: It may be more beneficial to include gradient boosting methods such as XGBoost or LightGBM.
- **Addressing Skewness**: Skewed features can be further improved by log transforming or scaling.

References:

1. Breiman, L. (2001). *Random Forests*. Machine Learning, 45(1), 5–32.
2. Cortes, C., & Vapnik, V. (1995). *Support-Vector Networks*. Machine Learning, 20(3), 273–297.
3. Géron, A. (2019). *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*. O'Reilly Media.