Syed Aslam Sheik Dawood

INFO ISTA 521 421 – Machine Learning Final Project

Predictive Modeling of Housing Prices in Boston: A Regression Analysis

Introduction:

The Boston Housing Dataset provides information about various factors affecting housing prices in Boston. This project aims at developing a machine learning model to predict the median value of owner-occupied homes (MEDV) based on the provided features. This project focuses on predicting housing prices in various towns of Boston using regression analysis. The goal is to implement three distinct regression models—Linear Regression, K-Nearest Neighbors (KNN) Regression, and Random Forest Regression—and evaluate their performance using Mean Squared Error, Mean Absolute Error, and R-Squared Error. The objective is to discern the most effective model for accurately predicting the median value of owner-occupied homes (MEDV) based on the given dataset. This project delves into the exploration of three potent regression models—Linear Regression, K-Nearest Neighbors (KNN) Regression, and Random Forest Regression aiming to unravel the intricate relationships between the features and the median value of owner-occupied homes (MEDV). Through the lens of regression analysis, we seek to analyse patterns, discern trends, and uncover insights. Our aim is to derive actionable insights that resonate with the pulse of Boston's housing market. Understanding the factors influencing housing prices can guide efforts to enhance economic opportunities, improve infrastructure, and foster a sustainable and inclusive environment.

Models:

I am planning to implement the following machine learning methods –

Linear Regression - A method to establish relationships between the independent variables and the target variable.

Random Forest - Robust algorithms suitable for regression tasks that can capture non-linear relationships and interaction among features.

K-Nearest Neighbours - KNN operates on the principle of proximity and makes predictions based on the average of the target variable of its K-Nearest Neighbours.

Linear Regression:

Description:

Linear Regression is a fundamental supervised learning algorithm used for predicting a continuous outcome variable based on one or more predictor variables. The model assumes a linear relationship between the predictors and the target variable. The basic equation for simple linear regression is given by:

$Y=b0+b1X+\epsilon$

where,

Y is the dependent variable,

X is the independent variable,

b0 is the y-intercept,

b1 is the slope, and

ϵ represents the error term.

Applicability:

Linear Regression is suitable for this problem because the relationship between the independent and dependent variables is approximately linear.

Citations:

G. James, D. Witten, T. Hastie and R. Tibshirani (2013). An Introduction to Statistical Learning.

K-Nearest Neighbours (KNN) Regression:

Description:

KNN is a non-parametric algorithm used for both classification and regression tasks. In the context of regression, the predicted value for a data point is the average of the values of its k-nearest neighbours. The distance metric is used to determine proximity.

Hat(Y) = 1/k $\sum_{i=1}^{K} Yi$

where,

Hat(Y) is the predicted value for the target variable,

Yi is the value of the target variable for the 'i' neighbour.

Applicability:

KNN is effective when local patterns in the data are important. It adapts well to irregular boundaries and is versatile across different types of datasets.

Citations:

T. Hastie, R. Tibshirani and J. Friedman (2009). The Elements of Statistical Learning.

Random Forest Regression:

Description:

Random Forest is an ensemble learning method that constructs a multitude of decision trees during training and outputs the average prediction of the individual trees for regression tasks. Each tree is constructed using a subset of the data and features, and the final prediction is an aggregation of the individual tree predictions.

Hat(Y) = 1/N $\sum_{i=1}^{N} Yi$

where,

Hat(Y) is predicted value for the target variable,

Yi is the prediction from the 'i' decision tree,

N is the number of decision trees in the forest.

Applicability:

Random Forest is versatile, robust to overfitting, and can handle complex relationships in the data. It is suitable for large and high-dimensional datasets.

Citations:

L. Breiman (2001). Random forests. Machine learning, 45(1), 5-32.

Data:

Data Description: Boston Housing Dataset

The Boston Housing Dataset aims to analyse the factors influencing housing prices in various towns around Boston. This dataset is commonly used for regression and predictive modelling tasks. The target variable is the median value of owner-occupied homes (MEDV) in thousands of dollars. This variable serves as the focal point for regression analysis, where the goal is to predict the median home value based on the given features.

Dataset Source: https://geodacenter.github.io/data-and-lab/boston-housing/

Structure of the Data:

The dataset comprises 506 instances, each representing a different suburb of Boston. The columns in the dataset provide a comprehensive set of features, with a mix of numerical and categorical variables. The features are:

Variable	Description
ID	Sequential ID
TOWN	A factor with levels given by town names
TOWNNO	A numeric vector corresponding to TOWN
TRACT	A numeric vector of tract ID numbers
LON	A numeric vector of tract point longitudes in decimal degrees
LAT	A numeric vector of tract point latitudes in decimal degrees
X	X Coordinates (UTM Zone 19)
Y	Y Coordinates (UTM Zone 19)
MEDV	A numeric vector of median values of owner-occupied housing in USD 1000
CMEDV	A numeric vector of corrected median values of owner-occupied housing in USD 1000
CRIM	A numeric vector of per capita crime
ZN	A numeric vector of proportions of residential land zoned for lots over 25000 sq. ft per town (constant for all Boston tracts)
INDUS	A numeric vector of proportions of non-retail business acres per town (constant for all Boston tracts)
CHAS	A factor with levels 1 if tract borders Charles River; 0 otherwise
NOX	A numeric vector of nitric oxides concentration (parts per 10 million) per town
RM	A numeric vector of average numbers of rooms per dwelling
AGE	A numeric vector of proportions of owner-occupied units built prior to 1940
DIS	A numeric vector of weighted distances to five Boston employment centres

Variable	Description
RAD	A numeric vector of an index of accessibility to radial highways per town (constant for all Boston tracts)
TAX	A numeric vector full-value property-tax rate per USD 10,000 per town (constant for all Boston tracts)
PTRATIO	A numeric vector of pupil-teacher ratios per town (constant for all Boston tracts)
B	A numeric vector of 1000*(Bk - 0.63) ^2 where Bk is the proportion of blacks
LSTAT	A numeric vector of percentage values of lower status population

Procedure:

Data Preprocessing:

- Load the Boston Housing Dataset using the pandas and read csv file into 'df' variable.
- Handled any missing values, outliers, or data transformations. Thus, used various techniques such as imputation, removal, or transformation to handle these issues.
- To find the relationship between the different features of the dataset, correlation plot is created. A correlation plot is a graphical representation of the correlation matrix of the dataset. heatmap() function has been used from the seaborn module to create a correlation plot.
- After analysing the dataset, it is split into a training set and a testing set. The training set is used to train the machine learning model, while the testing set is used to evaluate the performance of the model. We are using the train_test_split() function from the sklearn.model_selection module to split the dataset.

Model Training:

Regression models have been implemented using a machine learning library like scikit-learn. Scikit-learn offers a variety of regression models and tools to streamline the process of training, testing, and evaluating regression algorithms.

Linear Regression:

Linear Regression is applied to the Boston Housing Dataset to predict housing prices based on various features. The steps are given below:

Feature Selection:

Identify relevant features for training the regression models based on domain knowledge or feature importance analysis. Excluded non-contributing or redundant variables to enhance model efficiency. A subset of features related to housing characteristics, crime rates, environmental factors, and socio-economic indicators is selected from the dataset. This subset, denoted as 'hm,' includes columns such as 'CRIM,' 'ZN,' 'INDUS,' 'NOX,' 'RM,' 'AGE,' 'DIS,' 'RAD,' 'TAX,' 'PTRATIO,' 'B,' and 'LSTAT.'

Data Splitting:

The dataset is divided into input (independent variables, 'X') and output (dependent variable, 'y') values. The 'X' matrix includes the selected features, and 'y' represents the median house values ('MEDV'). The data is split into training and testing sets using the train_test_split function, with 70% of the data used for training and 30% for testing.

Model Instantiation and Training:

A Linear Regression model is instantiated using the LinearRegression class from the scikit-learn library. The model is then trained using the training data ('x_train' and 'y_train') with the fit method.

Prediction:

The trained model is used to predict housing prices on the testing set ('x_test'). The predicted values are stored in the 'predicted_house_price' array.

Visualization:

A scatter plot is generated to visually compare the predicted house prices with the actual values from the testing set. This provides a qualitative assessment of the model's performance.

K-Nearest Neighbors (KNN) Regression:

Model Configuration:

The K-Nearest Neighbors Regression model is configured with a specified hyperparameter, 'n_neighbors=7,' indicating that the algorithm will consider the predictions of the seven nearest neighbors when making predictions.

Model Training:

The dataset is divided into input (independent variables, 'X') and output (dependent variable, 'y') values. The 'X' matrix includes the selected features, and 'y' represents the median house values ('MEDV'). The data is split into training and testing sets using the train_test_split function, with 70% of the data used for training and 30% for testing. The KNN model is trained using training set ('x_train' and 'y_train') with the fit method.

Prediction:

The trained KNN model is used to predict housing prices on the testing set ('x_test') with the predict method. The predicted values are stored in the 'Y_pred' array.

Visualization:

A scatter plot is generated to visually compare the predicted and actual house prices. The x-axis represents the actual values from the testing set ('MEDV'), and the y-axis represents the corresponding predicted values ('Predicted_MEDV').

Random Forest Regression:

The Random Forest Regression algorithm is applied to predict housing prices in the Boston Housing Dataset. The process is described step by step:

Model Configuration:

A Random Forest Regression model is instantiated with the RandomForestRegressor class from scikit-learn. The hyperparameter 'n_estimators' is set to 100, indicating that the model will be an ensemble of 100 decision trees. The 'random_state' is set to 10 for reproducibility.

Model Training:

The Random Forest model is trained using the training set ('x_train' and 'y_train') with the fit method. This involves the creation of an ensemble of decision trees, each trained on a different subset of the data. The dataset is divided into input (independent variables, 'X') and output (dependent variable, 'y') values. The 'X' matrix includes the selected features, and 'y' represents the median house values ('MEDV'). The data is split into training and testing sets using the train_test_split function, with 70% of the data used for training and 30% for testing.

Prediction:

The trained Random Forest model is used to predict housing prices on the testing set ('x_test') with the predict method. The predicted values are stored in the 'Y_pred' array.

Visualization:

A scatter plot is generated to visually compare the predicted and actual house prices. The x-axis represents the actual values from the testing set ('y_test'), and the y-axis represents the corresponding predicted values ('Y_pred').

Evaluation:

The evaluation of regression models for the Boston Housing Dataset involves a comprehensive set of measures to assess the performance of each model in predicting housing prices. The following evaluation metrics are employed:

1. **Mean Squared Error (MSE):**

   Rationale: MSE is used as a metric for regression tasks as it penalizes larger errors more heavily. It provides a measure of the average squared difference between predicted and actual values.

   Formula:

   MSE= 1/n $\sum_{i=n}^{n}\left(Yi - Hat(Y)\right)^2$

   Interpretation: A lower MSE indicates better accuracy in predicting housing prices.

2. **Mean Absolute Error (MAE):**
   Rationale: MAE is suitable for regression tasks, providing a measure of the average absolute differences between predicted and actual values.
   Formula:
   MAE= 1/n $\sum_{i=n}^{n}|Yi - Hat(Y)|$

   Interpretation: MAE is easily interpretable and offers insights into the magnitude of errors.

3. **R-Squared (R^2) Score:**
   Rationale: R^2 measures the proportion of variance in the dependent variable (housing prices) that is predictable from the independent variables. It provides an indication of how well the model fits the data. R^2 is essential for understanding how well the regression models capture the variability in housing prices. A higher R^2 indicates that a larger proportion of the variance is explained by the model, reflecting a more effective predictive capability.

   Formula:
   R^2 = 1 - ($\sum_{i=0}^{n}\left(Yi - Hat(Yi)\right)^2 / \sum_{i=0}^{n}\left(Yi - Hat(Y)\right)^2$)
   Interpretation: R^2 ranges from 0 to 1, with higher values indicating a better fit.

After applying Linear Regression, K-Nearest Neighbours (KNN) Regression, and Random Forest Regression to the Boston Housing Dataset, the models were evaluated using Mean Squared Error (MSE), Mean Absolute Error (MAE), and R-Squared (R^2) Score as performance metrics.

Model	Mean Square Error (MSE)	Mean Absolute Error (MAE)	R-Squared (R^2) Score
Linear Regression	30.25	3.72	0.69
KNN Regression	55.74	4.83	0.43
Random Forest Regression	12.66	2.48	0.87

Linear Regression:

The linear regression model demonstrated moderate performance. The MSE and MAE values were reasonable, the R-squared value of 0.69 suggested that the model explained 69% variance in housing prices.

KNN Regression:

The KNN regression model exhibited moderate performance, with higher MSE and MAE compared to linear regression. The R-squared value of 0.43 indicated a lower level of explanatory power.

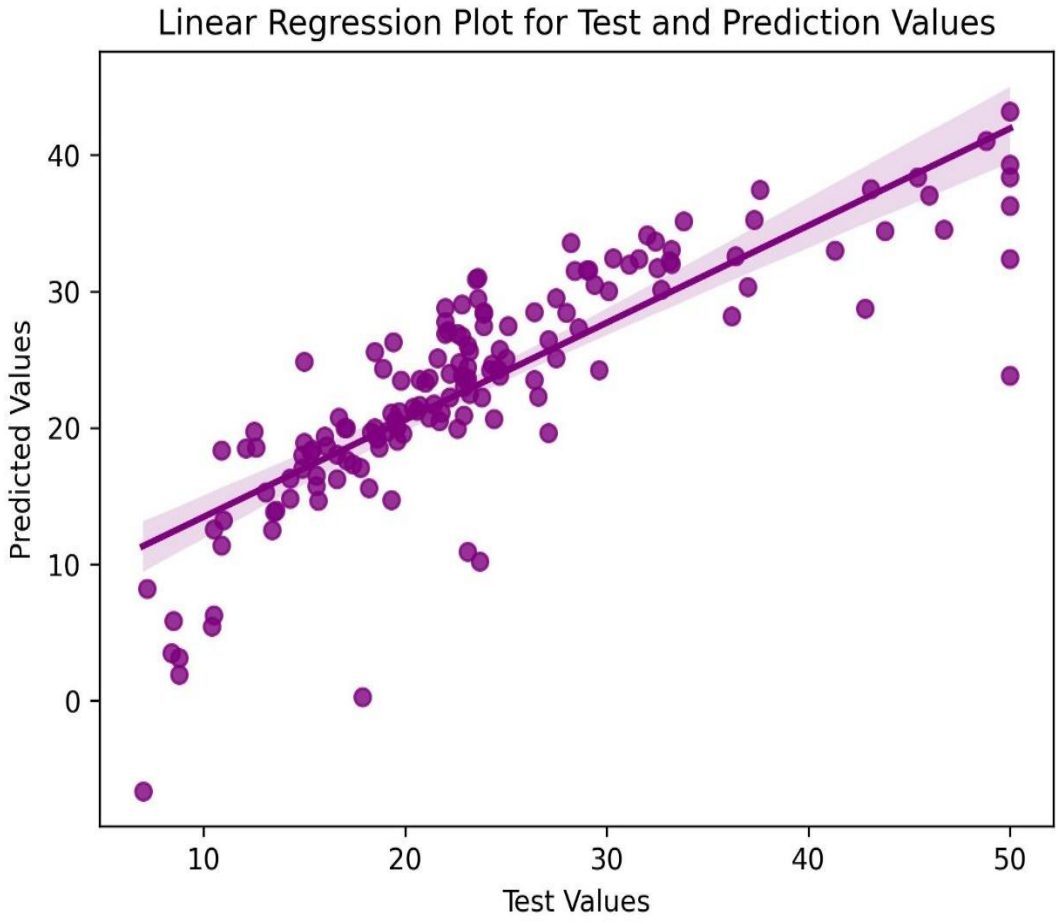
Random Forest Regression:

The Random Forest regression model outperformed the other models significantly. It showed the lowest MSE and MAE values and the highest R-squared value of 0.87, indicating accurate predictions and a strong fit to the data.

Results:

Linear Regression:

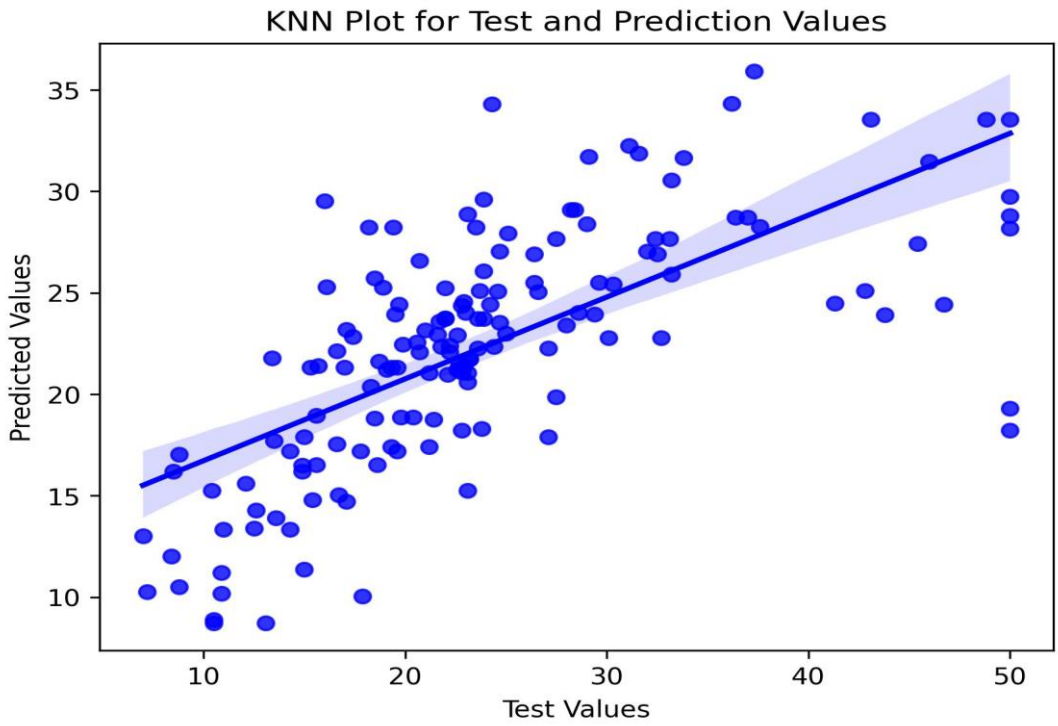
Figure 1: Linear Regression Model predicts Test Values with high accuracy



This plot shows a positive linear relationship between the Test and Prediction Values. The plot suggests that the model is performing well in predicting the values.

KNN Regression:

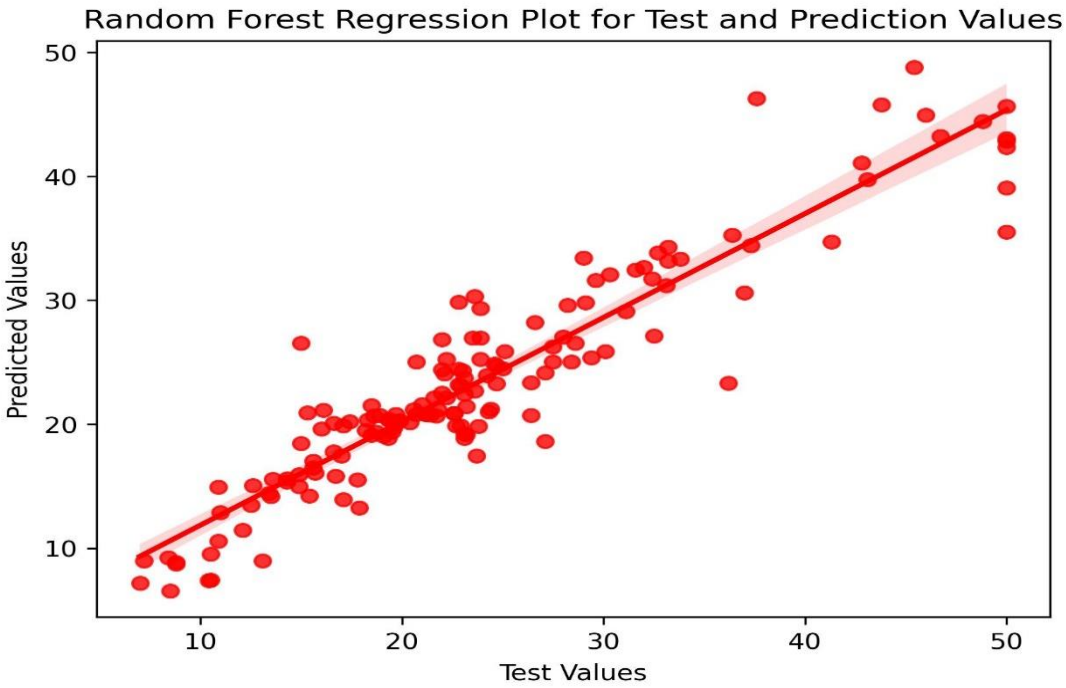
Figure 2: KNN Model predicts Test Values with high accuracy



This plot shows a positive linear relationship between the Test and Prediction Values. The plot suggests that the model is performing well in predicting the values.

Random Forest Regression:

Figure 3: Random Forest Regression Model predicts Test Values with high accuracy



This plot shows a strong positive correlation between the Test and Prediction Values. The plot suggests that the model is performing well in predicting the values.

Discussion of Results:

Linear Regression: The model exhibits a good fit, as evidenced by the tight clustering of points around the diagonal line. However, the performance metrics provide a more nuanced evaluation.

KNN Regression: The KNN model shows competitive performance, with the scatter plot indicating a notable degree of accuracy in predicting housing prices.

Random Forest Regression: The Random Forest model demonstrates strong predictive capabilities, as indicated by the close alignment of points with the diagonal line.

Comparison:

The Random Forest regression model consistently outperformed both Linear Regression and KNN Regression across all metrics. It achieved the lowest prediction errors for MSE and MAE and the highest explanatory power for R-squared, making it the most accurate and robust model for predicting housing prices.

Conclusion:

Based on the comprehensive evaluation of the regression models, the Random Forest regression model is recommended for predicting housing prices in the Boston Housing dataset. Its superior performance in terms of accuracy and explanatory power positions it as the preferred choice for this specific regression task.