House Price Prediction using Random Forest Machine Learning

Technique

**Abstract**

Forecasting a price variance instead of an exact value is more practical and appealing in most real-life usage. Price prediction can be treated as a classification problem here. Yet, the House Price Index (HPI) is also a standard measure for approximating the inconsistencies of house prices. Housing prices tend to correlate closely with other variables like location, city, and population, individual housing prices require information beyond HPI.

The HPI is a repeat-sale index, which measures average price changes in repeat sales or refinancings of the same properties. Thus, HPI is poor at forecasting the price of an individual house since it is a crude predictor of all transactions. This research examines the application of Random Forest, Linear Regression and Gradient Boosting Regressor machine learning algorithm for prediction of house prices. Boston housing dataset containing 506 samples and 14 features was employed to assess the accuracy of the suggested model for prediction. Comparison of all the algorithm and calculated the Evaluation Metics and got a difference of 0.3.

**Introduction**

The population of urban dwellers. As the population of urban dwellers increases, the demand for accommodation increases. As the demand increases, the price of house also increases. In addition to these, the infrastructural developments in an area can result in a sudden rise in the price of houses in a particular area. For instance, once the

challenges of unmotorable road and unstable electricity a

residential area become resolved, house owners tend to increase the prices of house in that particular area. In several nations, such as the United States Federal

Housing Finance Agency HPI, the United Kingdom National Statistics HPI, the United Kingdom Land Registry's HPI, the United Kingdom Halifax HPI, the United Kingdom Rightmove HPI, and Singapore's URA HPI, the House Price Index (HPI) is often used to calculate price increases in residential housing. However, research has shown that the use of HPI is not enough in this 21 st century. Generally, house prices are influenced by a number of variables. Authors in identified these factors to be physical condition, concept and location. Physical conditions that can be observed by physical perception include the size of the property, the number of rooms, the size of the kitchen and garage, the availability of the yard, the area of land and structures, and the age of the property. Physical characteristics of a house, such as the size of the structure, the year it was built, the number of bedrooms and bathrooms, and other facts that may define the house's interior features, may affect the price of a house. Although concepts refer to various marketing tactics employed by developers to attract potential investors. This includes how close the property is to hospitals, markets, educational institutions, airports, major roads etc. The location of a property has a significant impact on its price. This is because the current land price is determined by the area. Therefore, understanding house price patterns and determining factors is not only a thing of interest to tenants alone; it is also an issue of interest to home owners, analysts and policy makers in the real estate industry as well as urban and regional planning authorities. A computer-based prediction system can help them to make informed decision about if a property should be acquired and the best time to acquire the property Residential real estate is the primary store of equity for the middle class that serves as leverage for new businesses. However, rising house prices can boost demand by increasing homeowners' income, but they can also promote debt-financed consumption and weaken financial resilience. Price forecast strategies can be divided into two categories. The first category of strategies was intended to forecast market trends in a time-series format, such as stock and oil price forecasting. The second category of approaches focuses on estimating the price of particular goods based on their characteristics, such as the cost of a house or an airline ticket. The second form of price prediction task is the focus of this article. The time-series strategy involves looking for a relationship between present and previous rates. The second method involve the use of hedonic pricing and linear regression. The second approach that involves the use of Random forest , Gradient Boosting Regressor algorithm was adopted into this study. Over the years, machine learning techniques have been greatly explored for price prediction. The results obtained have shown the predictive prowess of machine learning algorithm. Machine learning creates algorithms and

builds models from data, then applies them to new data to make predictions. The key distinction between a model and a conventional algorithm is that instead of simply executing a sequence of instructions, a model is constructed from input data. Unsupervised learning uses unlabelled data, while supervised learning uses data with results labelled. Regression, inference, neural networks, and deep learning are some of the most popular machine learning algorithms. However, this work explores the use of random forest machine learning technique for house price prediction. UCI Machine learning repository Boston housing dataset was used to evaluate the performance of the proposed model while Mean Absolute Error (MAE), R² or Coefficient of Determination and the Root Mean Square Error (RMSE) were used as performance evaluation metrics.

**Methodology**

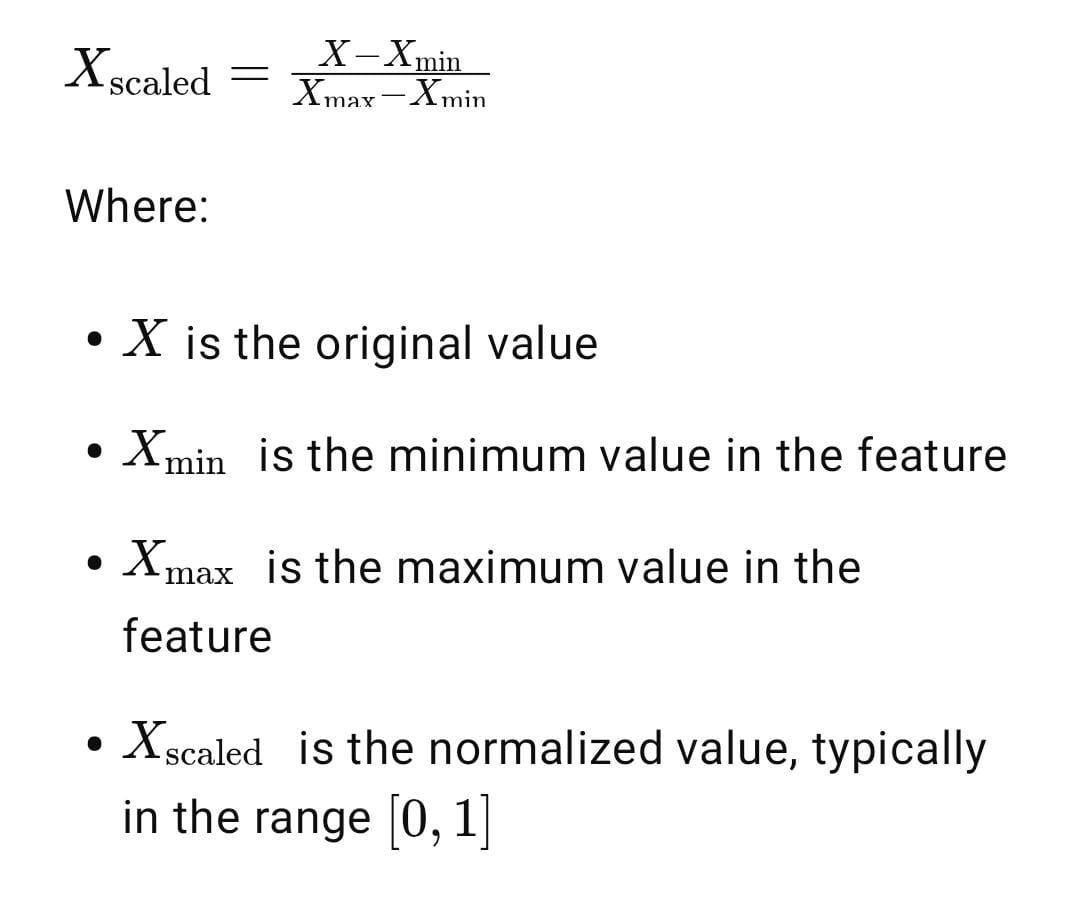
Regression model was used in this research to examine Boston housing datasets with the purpose of forecasting the prices of houses from the features that are present in the datasets. The core step that was taken towards the implementation are data collection, data exploration that was utilized in order to learn about the datasets and determine features within the dataset; data pre-processing phase that was utilized in cleaning the dataset so that it could be ready for model development. Subsequently, the model was created based on the suggested random forest algorithm.

***Data Collection and Exploration:***

In model development, the UCI Machine learning repository Boston housing dataset was employed. The dataset was obtained in 1978 and for each of the 506 entries, there is aggregated information about 14 features of homes from different suburbs in Boston, Massachusett dataset. Exploratory data analysis. Researchers can reveal the underlying trends in data through this way, which helps in the selection of appropriate machine learning methods. Data exploration was thus performed to know the features available in the dataset and their function. The features available in the dataset are: CRIM which represents the per capita crime rate by town, ZN which is the residential land area zoned for lots greater than 25,000sq.ft, INDUS which is ratio of non-retail business acres to town, CHAS which is dummy variable for Charles River (1 if tract is along river, 0 else), NOX which is concentration of nitric oxides (parts per 10 million), RM is average number of rooms in dwelling, AGE denotes proportion of owner-occupied units constructed before 1940, DIS is weighted distances to five Boston employment sub centers, RAD is accessibility index to radial highways, TAX is the tax-on-fair-assessment rate per $10,000, PTRATIO is the student-teacher ratio by town, B1000(Bk - 0.63) ^2 where Bk is the percentage of blacks by town, LSTAT is the lower status percentage of the population and MEDV is the median house-value of owners in $1000's. As the model has a supervised learning technique, the dataset should be split into the training dataset and testing dataset. For the training dataset, 70% of the dataset was utilized for training the model and the other 30% for testing and Cross Validation.

***Data Pre-Processing***

The data obtained for model training and testing should be pre-processed in the right way before developing models such that the models learn the patterns faster. Numerical values were normalized and categorical values were encoded one-at-a-time. Following the data exploration and choosing the most appropriate feature using the heatmap, the following step is the pre-processing of the data of the chosen features to be utilized. Typically, the data sets obtained for the training and test task possess multiple features. It is most likely that the values of different features are at a different scale which might decrease the performance of the model, hence scaling was performed so that the features would be on a relatively consistent scale. The Normalization tool provided in Phyton Skitlearn module was used for this task. The Normalization makes the assumption that your data is normally distributed within each function and standardizes it to be now clustered and scale it down in the range of [0,1]. The normalization of the feature are calculated and then the feature is standardized on:



On scaling the features, a linear regression plot (regplot) was plotted to observe the correlation between the features and MEDV. This is to better comprehend the dataset as MEDV is the variable to be predicted.

**Model Development**

The model was developed with the random forest algorithm. The random forest was utilized with the RandomForestClassifier class in Phyton Scikit-learn (sklearn) machine learning library. Random Forest is a widely used supervised classification and regression machine learning algorithm. It utilizes the method of ensemble learning to solve complicated problems by using many classifiers to enhance the model's accuracy. Random Forest is a classifier that takes the average outcome of many decision trees tested on different subsets of a dataset to enhance the predictive accuracy of the dataset. Instead of using a single decision tree, the random forest utilizes the projections from every tree to decide the ultimate performance based on the majority votes. The *random forest algorithm* is

1. Construct an n-sample random bootstrap sample (by substitution, take n samples at random from the training set).
2. For each node, construct a decision tree based on the bootstrap sample: a. Take d functions at random without replacement.
3. b. Split the node based on the attribute that provides the best split in terms of the objective function, like maximizing knowledge gain here.
4. Iterate steps 1-2 k times more.

*Linear Regression:*

1. Training:
2. Analyze all features together on the entire dataset.
3. Find a single mathematical equation (a line or hyperplane) that best fits the data.
4. It uses least squares to minimize the total prediction error.
5. Prediction:
6. Input values are fed into the equation.
7. The equation directly returns a single prediction (e.g., price).
8. No trees, no bootstrapping, no multiple models — just one global linear function.

*Gradient Boosting Regression –*

* 1. Initialize the Model:

1. Start with a simple prediction (e.g., the average house price).
   1. Train Weak Models (Decision Trees) Sequentially:
2. Compute the error (residuals) between actual values and current predictions.
3. Train a small decision tree to predict those errors.
4. Add the output of that tree to the previous prediction to correct it.
5. Repeat this process for many rounds.
   1. Final Model:
6. The final prediction is the sum of all corrections made by the trees.
   1. Prediction:
7. Pass the input through each small tree.
8. Each tree gives a small adjustment.
9. Add all adjustments to the initial prediction → get final predicted price.

And by taking a majority vote of the predictions made by each tree, it assigns the class name. Also, n\_estimators of the Random Forest Classifier, Linear and Gradient Boosting Regression makes it possible for us to select how many trees we should make that we've set to 500.

Only a small collection of features will be utilized to bring variation into random forest subsets, however. We optimized the Random Forest Classifier, Linear and Gradient Boosting Regression by running the model multiple times and implementing a few parameters when we created it.

***Data Exploration Process Results***

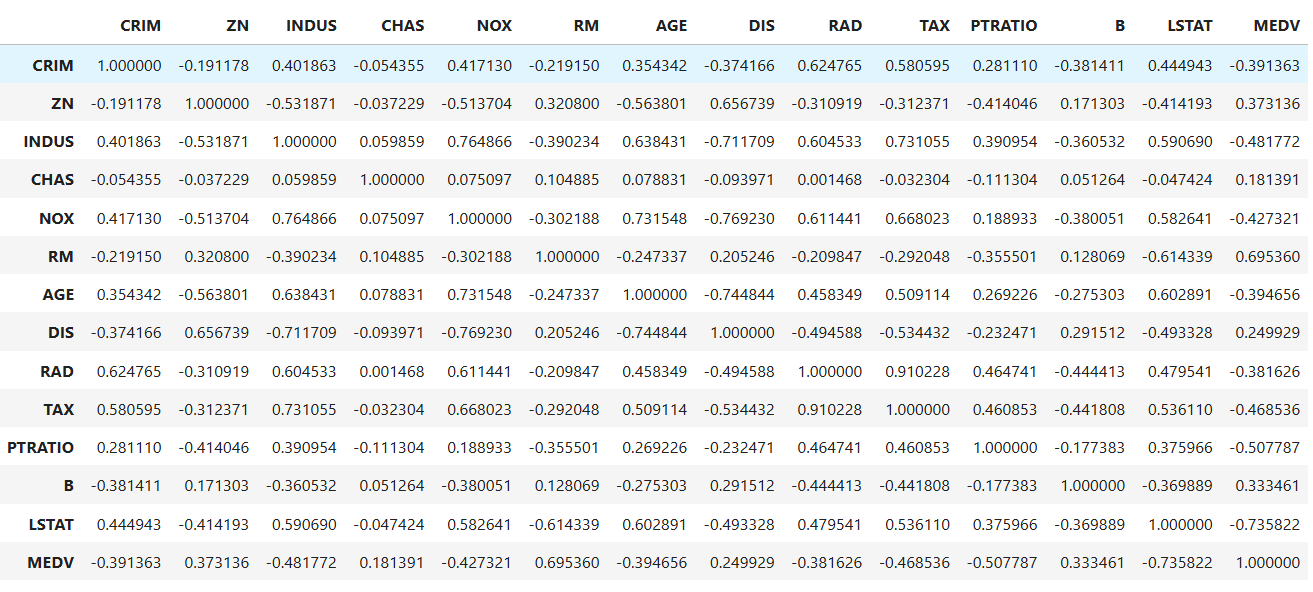
To gain a deeper understanding of the dataset, a comprehensive data exploration process was conducted. Figure 1 presents the distribution of all the features within the dataset. This includes the total count of data entries, along with statistical measures such as mean, standard deviation, minimum, maximum, and quartile values (25%, 50%, 75%).

Two specific features showed notable characteristics. The **ZN** column (proportion of residential land zoned for lots over 25,000 sq. ft.) displays 0 as the value for both the 25th and 50th percentiles. Similarly, the **CHAS** variable (Charles River dummy variable: 1 if the tract bounds the river; 0 otherwise) shows 0 for the 25th, 50th, and 75th percentiles. These patterns suggest that these features are heavily skewed or binary in nature, especially in the case of CHAS, which is categorical.

Since both ZN and CHAS are either conditional or categorical, their statistical summaries are expected to be sparse or concentrated around specific values. As a result, they may not contribute significantly to predicting the target variable **MEDV** (Median value of owner-occupied dwellings) in regression models.

To evaluate the effectiveness of predictive modeling on this dataset, two regression algorithms were employed: Linear Regression and Gradient Boosting Regressor (GBR). Linear Regression provides a simple baseline to understand linear relationships between independent variables and MEDV. In contrast, GBR is a powerful ensemble technique capable of capturing complex non-linear interactions in the data.

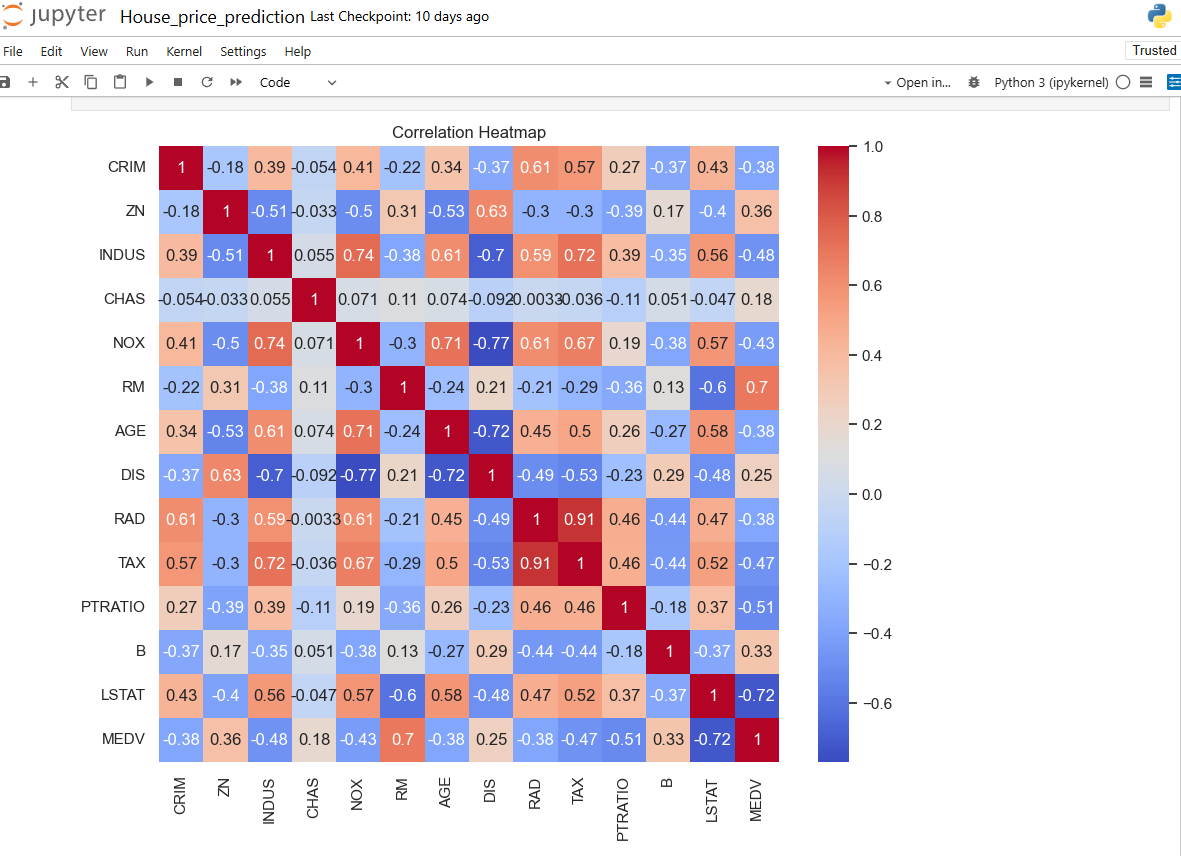
Insights from the data exploration phase informed the feature selection and preprocessing steps for both models. Understanding the distribution and relevance of variables such as ZN and CHAS played a role in interpreting model performance and refining the predictive approach.



**The Correlation Metrics**

*Correlation Analysis*

To understand the interdependence among variables and identify the most influential features affecting housing prices, we performed a correlation analysis using a Pearson correlation matrix. The correlation matrix provides a symmetric overview of pairwise linear relationships between all numerical features in the dataset, with values ranging from -1 (perfect negative correlation) to +1 (perfect positive correlation). A heatmap visualization was generated to aid in interpreting the strength and direction of these relationships, as illustrated in Figure.



Particular attention was given to the correlation between the predictor variables and the target variable MEDV (Median value of owner-occupied homes in $1000s), as these relationships are critical in determining model performance.

From the correlation matrix, several key insights were obtained:

The variable LSTAT (percentage of lower-status population) exhibited a strong negative correlation with MEDV (), suggesting that housing prices tend to decrease as the proportion of lower-income residents increases. The variable RM (average number of rooms per dwelling) demonstrated a strong positive correlation with MEDV (), indicating that homes with more rooms are typically valued higher.

Other variables such as PTRATIO (pupil-teacher ratio) and TAX (property tax rate) showed moderate negative correlations with MEDV ( and respectively), implying that higher taxes and student-teacher ratios are associated with lower housing prices. Notably, some features like RAD (access to radial highways) and TAX showed a high inter-correlation (), which may introduce multicollinearity concerns in linear models. Such relationships necessitate careful feature selection or the use of regularization techniques during model training.

This correlation analysis not only informed the feature engineering process but also provided a foundation for selecting variables with the most predictive power for downstream modeling, while also identifying and addressing potential redundancy among features.

**Cross Validation**

To ensure robust estimation of model generalizability and mitigate the risk of overfitting, we employed K-Fold Cross-Validation as the primary evaluation strategy. Cross-validation is a well-established resampling technique used to assess the performance of predictive models on independent datasets by partitioning the available data into subsets for systematic training and validation.

*K-Fold Cross-Validation*

In this study, we utilized K-Fold Cross-Validation with, a commonly used configuration that balances bias and variance in error estimation. The dataset was randomly partitioned into five approximately equal-sized folds. For each iteration, the model was trained on folds and validated on the remaining fold. This process was repeated five times, with each fold serving as the validation set exactly once. The overall model performance was then computed as the mean and standard deviation of the evaluation metric across all folds.

This methodology ensures that each data point is used for both training and validation, thereby improving the statistical power of performance estimation. Furthermore, it provides insights into model stability by observing the variance in scores across folds.

***Performance Metrics***

Model performance during cross-validation was evaluated using [insert relevant metric(s), e.g., Mean Squared Error (MSE), Root Mean Squared Error (RMSE), R-squared (R²), Accuracy, F1-Score, etc.], depending on the problem type (regression or classification). For regression tasks in this study, we primarily report the Root Mean Squared Error (RMSE) and R² score, calculated for each fold and subsequently averaged to obtain the final cross-validated score.

**Evaluation Metris**

***Performance Analysis of the Developed Model***

Once the model was trained and tested, the performance measurement metrics were utilized to obtain the performance of the model. They are the Mean Absolute Error (MAE), R² or Coefficient of Determination and the Root Mean Square Error (RMSE). The model's performance as per the metrics is given in.

*Random Forest Model Evaluation*

Subsequent to obtaining the model's performance a scatter plot was created to indicate the linear regression between the predicted value and actual value from the model. This is indicated in good performance of the forecasted values versus the actual values may be due to the K-fold cross-validation and Coupling effect of multivariate regressions. The k-fold cross-validation technique is an efficient method of achieving a good bias-variance trade-off. This technique is employed by Stacking Regression in order to find out the general- inization efficiency of every variable model. Different regression techniques can reinforce each other. The second level of stacking will learn and accurately predict house prices from the pre-estimated prices of the first stacking level.

*Random Forest with Cross-Validation (5-fold):*

*MAE: 3.0867114624505927*

*MSE: 21.888164814545473*

*RMSE: 4.678478899658036*

*R² Score: 0.7407216312052729*

*Linear Regression Model Evaluation*

Following the model training, a scatter plot was generated to illustrate the linear relationship between the predicted values and the actual values. The alignment of the points along the diagonal suggests a reasonably strong performance of the linear regression model, albeit with more variance than ensemble models. Linear regression provides a baseline model, capturing linear trends between input features and the target variable. While it may underperform in capturing complex nonlinear relationships, the model’s interpretability and simplicity make it a valuable benchmark. Cross-validation was employed to mitigate overfitting and assess generalization capability. The consistent performance across folds indicates that the linear model does not suffer from high variance, although it may have limited capacity to capture interactions and higher-order feature dependencies.

*Linear Regression with Cross-Validation:*

*MAE: 4.269089732145535*

*MSE: 37.590610731066064*

*RMSE: 6.1311182284364785*

*R² Score: 0.5547167926170073*

*Gradient Boosting Regressor (GBR) Model Evaluation*

Upon evaluating the GBR model, a scatter plot was drawn to compare the predicted values with the actual values, revealing a tightly clustered distribution along the ideal fit line. This demonstrates the superior predictive capability of GBR, especially in modeling nonlinear relationships. The performance is likely enhanced due to the model's iterative boosting mechanism, where each weak learner improves upon the residuals of its predecessor. The use of K-fold cross-validation helped fine-tune the hyperparameters and avoid overfitting, ensuring robust generalization. The ensemble nature of GBR, combining multiple decision trees in sequence, captures complex patterns and interactions among features. Additionally, when used in stacking regression frameworks, GBR effectively contributes as a strong base learner, providing accurate initial estimates that can be further refined at higher stacking levels.

*Gradient Boosting Regressor with Cross-Validation:*

*MAE: 3.038138797504764*

*MSE: 19.19706734832672*

*RMSE: 4.381445805704633*

*R² Score: 0.7725992859662214*

Results and Discussion

To evaluate and compare the performance of various regression algorithms for house price prediction, three models were trained and assessed using cross-validation: Linear Regression, Random Forest, and Gradient Boosting. The evaluation metrics used were Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and R² Score. Figure presents a visual comparison of the models based on these metrics.

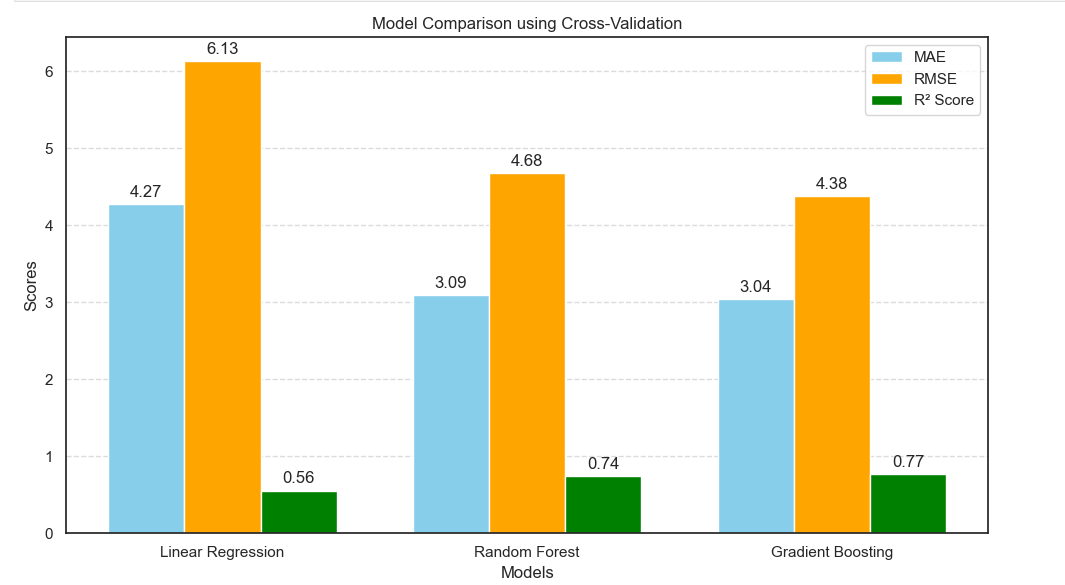


Figure shows that Gradient Boosting and Random Forest significantly outperform Linear Regression across all three metrics. The Gradient Boosting model achieved the lowest MAE (3.04) and RMSE (4.38), indicating that it provided the most accurate predictions. Additionally, it achieved the highest R² Score of 0.77, reflecting a better fit and greater explained variance in the target variable.

Random Forest also performed well with an MAE of 3.09, an RMSE of 4.68, and an R² Score of 0.74, suggesting that it is a robust model for this regression task. In contrast, Linear Regression had the highest error metrics (MAE: 4.27, RMSE: 6.13) and the lowest R² Score (0.56), highlighting its limited capacity to capture complex nonlinear relationships in the dataset.

These results confirm the superiority of ensemble models, particularly Gradient Boosting, in predicting house prices due to their ability to handle nonlinearities and interactions between features more effectively.

Conclusion

Each year, house prices increase, making it necessary to develop a mechanism for predicting future house prices.

House price prediction can be applied by landowners, estate valuers, and policymakers in order to determine the valuation of a house and the reasonable price for sale. This will help prospective buyers decide on the time to buy a home. Although physical condition, style, and location are the three primary factors that determine a house's price, the individual variables affecting the price of a house differ. A flawless prediction model hence needs to fit the particular variables that affects the price of a house in the area under consideration. This research has again confirmed the capability of random forest machine learning method in forecasting prices of a house according to variables available in Boston housing dataset. The comparison between the forecasted and real prices sown in

Table 1 indicated that the model had a prediction margin of ±5. This demonstrated that the model can be utilized to make predictions on house prices. A number of other machine learning models particularly deep learning models can also be utilized for predicting house prices.

Conclusion

Based on the comparative analysis illustrated in the figure, it is evident that Gradient Boosting is the most effective model for house price prediction among the three evaluated. It yielded the lowest Mean Absolute Error (3.04) and Root Mean Squared Error (4.38), along with the highest R² Score (0.77), indicating superior accuracy and predictive capability.

Random Forest also demonstrated strong performance, surpassing Linear Regression in all metrics. In contrast, Linear Regression showed the weakest performance, suggesting that simple linear models may not be sufficient for capturing the complex patterns in housing data.

Overall, ensemble methods—particularly Gradient Boosting—are recommended for building robust and accurate house price prediction systems.