**HOUSE PRICE PREDICTION**

**A hand holding a white arrow over a stack of coins

Description automatically generated**

**Introduction:**

The housing sector continues to be a vital component of economic expansion and stability, impacting both personal wealth and more general financial patterns. Predictive study of housing prices is therefore not only an academic endeavor but also an essential resource for investors, decision-makers, and prospective homeowners. Our project explores this exciting field of research by employing a traditional dataset that depicts the complex dynamics of the Boston property market in 1970. The Boston housing market offers a rich tapestry of data that allows us to investigate and forecast the median value of owner-occupied houses. Boston is recognized for its variety and historical significance within the urban fabric of the United States. The dataset provides a multidimensional picture of the elements that influence property prices. It is obtained from the Boston Standard Metropolitan Statistical Area and includes a range of variables, from per capita crime rates to the percentage of owner-occupied apartments built before 1940. Our study attempts to accurately estimate home prices by utilizing machine learning techniques on this information, while also revealing underlying patterns and linkages within the market. This study is situated at the nexus of data science and urban economics, offering perspectives that may influence future advances in both domains.

**Problem Statement/ Hypothesis:**

The housing market in metropolitan areas is a dynamic, intricate system that is impacted by several socioeconomic variables. In a system like this, figuring out the median price of owner-occupied houses is a difficult issue that has important real-world ramifications for a variety of stakeholders, including investors, buyers, sellers, and legislators. The challenge at hand is to precisely forecast the median value of homes (MEDV) in the Boston suburban using a set of factors that account for both the socioeconomic status of the communities where the homes are situated and their physical attributes. In addition to proving the viability of using these models to studies of the present and future housing markets, the effective prediction of MEDV from these features would also show promise for employing them in real estate investment and urban planning decision-making.

**Methodology:**

**Workflow Diagram**

A screenshot of a computer

Description automatically generated

Our machine learning project's workflow flowchart for predicting housing prices captures a methodical procedure that starts with obtaining the Boston Housing dataset. After that, the data is cleaned, normalized, and divided into training and testing sets as part of the preparation step. Statistical insights and visual signals on the characteristics and interaction of variables are obtained by further exploratory analysis of data, which is followed by feature engineering to enhance the models' predictive capability. The main steps in the procedure are choosing, configuring, and testing different regression models based on performance measures; next, by optimizing hyperparameters, the models are adjusted and their predicted accuracy is confirmed.

**Dataset**

The Boston Housing dataset was first gathered in 1978 by Harrison and Rubinfeld. It consists of 506 entries, each of which contains aggregated information about different facets of 14 characteristics from Boston, Massachusetts, suburbs. These characteristics include a blend of environmental, property-related, and socioeconomic factors.The dataset may be shown tabularly, with each row denoting a distinct town or suburb and each column representing a specific attribute of that location. It has the following variables:

CRIM: Town-specific per capita crime rate .

ZN: Area of residential property designated for lots larger than 25,000 square feet.

INDUS: The percentage of each town's non-retail business acres.

CHAS: Charles River dummy variable (zero otherwise, if tract is not limited by river).

NOX stands for concentration of nitric oxides (parts per million).

RM: The typical amount of rooms in a home.

AGE: The percentage of owner-occupied homes constructed before 1940.

DIS: Weighted travel times to five job hubs in Boston.

RAD stands for the Radial Highway Accessibility Index.

TAX: The property tax rate is $10,000 at full value.

PTRATIO: Town-specific student-teacher ratio.

B: 1000(Bk - 0.633)^2, where Bk represents each town's percentage of Black citizens.

LSTAT: The population's percentage of lower status individuals.

MEDV: Owner-occupied house median value expressed in $1,000s (Target variable).

**Exploratory data analysis**

Statistics and graphical representations, we examine the dataset in detail to find underlying trends, identify anomalies, and test hypotheses during the exploratory data analysis phase. This entails identifying outliers, analyzing the correlations between the characteristics, and showing the distribution of each variable.

The graph, which is a heatmap, shows the correlation coefficients between the target variable MEDV and other aspects of the Boston Housing dataset.

**A screenshot of a graph

Description automatically generated**

The graph, which is a bar chart with error bars to show variability, compares the median value of dwellings (MEDV) by the Charles River (CHAS=1) to properties not beside the river (CHAS=0).

**A graph of a bar chart

Description automatically generated with medium confidence**

A visual representation of the distribution and outliers for each variable is provided by the graph, which shows a succession of box plots for the different aspects of the Boston Housing dataset.

A group of blue boxes

Description automatically generated

Correlation Graph between all variables

A blue and green squares

Description automatically generated

The graph shows the distribution of each variable in the Boston Housing dataset as a set of histograms overlaid with estimates of kernel densities.

A group of blue and white graphs

Description automatically generated

The graph is made up of scatter plots that illustrate the link between the Boston Housing dataset's median value of houses (MEDV) and a number of other parameters. Trend lines indicate the intensity and direction of the correlations between the features.

A group of blue dots

Description automatically generated

Info of all the variables in the dataset

A screenshot of a computer

Description automatically generated

Description of all the variables in the data. It has all statistical values.

A screenshot of a computer

Description automatically generated

Finding null values in the dataset.

A screenshot of a computer

Description automatically generated

Splitting the data in train and test using train\_test\_split library

A close-up of a computer code

Description automatically generated

**Implementation**

Algorithm LinearRegression:

Input: Training data X, target values y

Output: Model parameters θ

Begin

Initialize θ to random values or zeros

Loop until convergence or for a set number of iterations:

Calculate predictions: ŷ = X \* θ

Calculate loss: L = (1/2m) \* Σ(ŷ - y)^2

Calculate gradients: ∇L = (1/m) \* X' \* (ŷ - y)

Update parameters: θ = θ - α \* ∇L

End Loop

Return θ

End

Algorithm RandomForest:

Input: Training data X, target values y, number of trees n\_trees

Output: A set of decision trees

Begin

Initialize an empty set of trees

For i = 1 to n\_trees:

Bootstrap a sample X', y' from X, y

Tree t = DecisionTree(X', y')

Add t to the set of trees

End For

Return the set of trees

End

**Linear Regression Model**

A screenshot of a computer program

Description automatically generated

**Random Forest Model**

A screenshot of a computer program

Description automatically generated

**SVM Model**

A screenshot of a computer program

Description automatically generated

Libraries for Implementation:

Scikit-learn (from sklearn.linear\_model import LinearRegression) is used for linear regression.

Scikit-learn (from sklearn.ensemble import RandomForestRegressor) is used for Random Forest.

Scikit-learn (from sklearn.svm import SVR) is used for SVM.

A screenshot of a computer program

Description automatically generated

In order to use linear regression, a line that minimizes the sum of squared discrepancies between the observed and predicted values is fitted across the high-dimensional data. In order to provide a forecast that is more reliable and accurate, Random Forest constructs several decision trees and combines them. In order to provide a varied collection of models, each tree in the Random Forest operates on a random selection of data and characteristics. The hyperplane with the largest margin that most effectively divides the data classes is found by the SVM algorithm. It attempts to fit the best line under a threshold value for regression tasks (SVR). To handle these techniques, Scikit-learn offers well-optimized libraries in every situation; all that is needed to train the models and provide predictions is the data and hyperparameters.

**Results**

R2 and MSE, and RMSE performance metrics are taken into consideration for the evaluation of the three machine learning models employed in your project: Random Forest, and Support Vector Machine (SVM),Linear Regression.

**linear regression method**

R2: 0.712 meaning , the model can account for around 71.2% of the variation in the target variable. The adjusted R2 for the no of predictors in the model is 0.686, which shows a good fit but takes into account the possible cost of including further predictors. The average squared difference between the observed actual outturns and the model's predictions is displayed as MSE: 30.05. The residuals' standard deviation (RMSE) is 5.48, indicating that the model's predictions are, on average, within $5480 of the actual values.

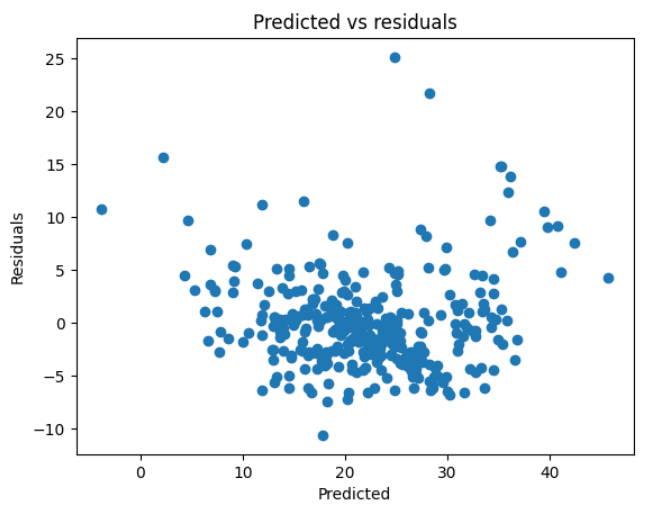
A screenshot of a computer program

Description automatically generated

A graph showing the difference between prices and price

Description automatically generated

The above graph is a scatter plot that contrasts anticipated prices, which are plotted on the y-axis and were most likely determined using a linear regression model, with actual prices, which are plotted on the x-axis. Plotting the actual price against the matching forecasted price, each point on the graph represents a distinct data point. The points' pattern indicates a relationship between the actual and expected prices, suggesting that the linear regression model can forecast prices with some degree of accuracy. All points should ideally sit on a straight line with a slope of 1, where the projected and actual prices are equal, if the forecasts were correct.



The scatter plot of the residuals against the expected values from a linear regression model is shown in the graph above. The discrepancies between the actual observed values and the values the regression model anticipated are known as residuals. The residuals, which may be computed as actual value - anticipated value, are represented on the y-axis. The regression model's anticipated values are presented on the x-axis.

**Random Forest**

R2 value is 0.834 shows that, compared to the Linear Regression model, the random forest model better explains 83.4% of the variability in the response variable. A better fit when taking into consideration the amount of characteristics utilized is shown by the adjusted R2 of 0.819, which is greater than that of linear regression. MSE: 17.27, which indicates higher prediction accuracy compared to linear regression. RMSE: 4.16 shows that, on average, the model's predictions and the actual values are within $4160 of each other.

A screenshot of a computer code

Description automatically generated

A graph showing the difference between prices and price

Description automatically generated with medium confidence

The difference between the actual and expected prices is positively correlated. Predicted prices typically rise in tandem with real prices, indicating that the model is capturing the data's trend. Several points, particularly in the upper price range, dramatically depart from the overall trend.

**Hyper parameter tuning for Random Forest:**

The following are the main RandomForestRegressor hyperparameters:

n\_estimators : no of trees in the node

max\_features: The quantity of features to take into account while determining the optimal split.

max\_depth: The tree's maximum depth.

min\_samples\_leaf: The bare minimum amount of samples necessary for a leaf node to exist.

bootstrap: useful to constructing the tree.

A screenshot of a computer program

Description automatically generated

A screenshot of a computer

Description automatically generated

**SVM Model**

R2 values 0.590 indicates that the model is less successful than both the Random Forest and Linear Regression models, explaining 59.0% of the variation in the target variable. The model with the lowest adjusted R2 value (0.551) out of the three suggests that the model's complexity, rather than the features themselves, may be partially to blame for the variation explained. SVM has the lowest prediction accuracy in this case, as evidenced by its MSE of 42.81, the highest error score of the three models. RMSE: 6.54 indicates an average $6540 difference between the SVM model's predictions and the actual values.

A graph with blue dots

Description automatically generated

In the scatter plot supplied, real prices are plotted on the x-axis and anticipated prices are plotted on the y-axis. A Support Vector Machine (SVM) regression model was used to make the predictions. The actual and forecasted prices have a definite positive correlation. Predicted prices rise in tandem with real prices, suggesting that the SVM model can accurately represent the data's trend.

A screenshot of a computer code

Description automatically generated

**Hyper Parameter Tuning:**

Optimizing hyperparameters is a crucial step in enhancing a machine learning model's performance. Key hyperparameters to adjust for an SVM regressor such as SVR from sklearn are as follows: The parameter for regularization. The regularization's strength is inversely related to C. It has to be unwaveringly positive. Kernel, Indicates the kind of kernel that will be used to the algorithm. One of the following must be true: "linear," "poly," "rbf," "sigmoid," "precomputed," or callable.

A screenshot of a computer program

Description automatically generated

A close-up of a computer code

Description automatically generated

**Visual diagrams for results:**

A graph with blue rectangles

Description automatically generated

Evaluates and contrasts the R-squared values of three distinct regression models: Random Forest, Linear Regression, and Support Vector Machines. With respect to its R-squared score, each bar indicates how well the model explains the variability of the target variable. The predicted accuracy of each model is shown by the length of the bar, which reflects the percentage of the R-squared score. Random Forest outperforms the other two models in terms of predictive accuracy, while Support Vector Machines get the lowest score.

**Project Management**

Data Preprocessing: The dataset for Boston Housing was successfully obtained. The first stages of preparation, which included data normalization, outlier identification, and handling of missing information, were finished.

Exploratory Data Analysis (EDA): Carried out thorough EDA, which included feature connections, distributions, and correlations as well as statistical summaries and visualizations.

Model Building: Three prediction models were developed, Support Vector Machine (SVM), Random Forest, and Linear Regression.

Model Evaluation:

R-squared, Adjusted R-squared, MSE, and RMSE were used to evaluate the models. With the lowest error metrics and the greatest R-squared value, Random Forest turned out to be the top performer.

The project's goal was to use a variety of machine learning approaches to forecast the median value of properties in the Boston suburbs. The project was well documented at every point, guaranteeing that the procedure was clear and repeatable. The project has followed a defined methodology, starting with data preparation to create a clean dataset and moving on to exploratory data analysis to provide guidance for feature selection and obtain insights. Three models were then put into practice, and the goal was to comprehend both the individual and comparative predicted performances of each model. The Random Forest model was found to be the most promising model for this application due to its higher performance metrics.

**Responsibility:**

**References:**

1. Harrison, D., & Rubinfeld, D. L. (1978). Hedonic housing prices and the demand for clean

air. Journal of Environmental Economics and Management, 5(1), 81-102.

<https://doi.org/10.1016/0095-0696(78)90006-2>

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

<https://doi.org/10.1023/A:1010933404324>

1. Cortes, C., & Vapnik, V. (1995). Support-vector networks. Machine Learning, 20(3), 273-297. <https://doi.org/10.1007/BF00994018>
2. James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). An Introduction to Statistical Learning: with Applications in R. New York, NY: Springer.