Machine Learning Algorithms to Predict Housing Prices

1. Introduction:

Machine Learning (ML) is a subset of artificial intelligence. It enables systems to automatically learn from experience without explicit programming. This involves training models on past data to discover patterns, make decisions, and predict outcomes. The ultimate goal of ML is to develop algorithms that can generalize from examples, allowing computers to handle tasks like classification, regression, clustering, and more.

ML algorithms can be categorized into three main types: **supervised, unsupervised, and reinforcement learning**. In supervised learning, we have **regression and classification** algorithms. Machine Learning algorithms for regression are used to predict continuous outcomes by understanding the relationship between input features and target variables. Common regression algorithms include Linear Regression, which fits a straight line to the data. Polynomial Regression, which captures curved trends and Support Vector Regression (SVR), which finds the best-fit hyperplane. Decision Trees split data into branches and Random Forests, which combine multiple trees for accuracy. Additionally, Logistic Regression models binary outcomes, and K-Nearest Neighbors (KNN) predicts values based on nearby data points.

Dataset Description:

The USA Housing dataset contains information about housing prices across different regions in the U.S. It contains multiple attributes that provide insights into social and economic factors affecting housing costs across different regions. The USA Housing dataset consists of 5,000 rows, each representing a house and contains 6 features and 1 output variable. The goal of this analysis is to predict house prices based on various features representing area characteristics.

Input/independent Variables (Features):

Avg. Area Income: Average income of residents in the area.
Avg. Area House Age: Average age of the houses in the area.
Avg. Area Number of Rooms: Average number of rooms in houses in the area.
Avg. Area Number of Bedrooms: Average number of bedrooms in houses in the area.
Area Population: Population size of the area.
Address: Address of the house

Output/dependent Variable (Target/Label):

Price: The price of the house, which we aim to predict.

Types of Algorithms Used:

To predict house prices, a variety of supervised learning algorithms were employed, including Multiple Linear Regression, Polynomial Regression, Support Vector Regression (SVR), Decision Trees, and Random Forests. These algorithms help capture both linear and non-linear relationships between the features and the target

2. Prediction Methods:

1. Multiple Linear Regression

Multiple Linear Regression models the relationship between a dependent variable and multiple independent variables. The general equation is represented as:

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + ... + \beta_{\square} x_{\square}$$

where:
- y: Predicted output (house price)
- β_0: Intercept
- $\beta_1, \beta_2, ..., \beta_{\square}$: Coefficients of the input features
- $x_1, x_2, ..., x_{\square}$: Input variables (features)

2. Polynomial Regression

Polynomial Regression is an extension of linear regression that models the non-linear relationship between the input variables and the target variable by adding polynomial terms. The general equation is:

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_1^2 + ... + \beta_{\square} x_1^n$$

where:
- y: Predicted output (house price)
- β_0: Intercept
- $\beta_1, \beta_2, ..., \beta_{\square}$: Coefficients of the input features and their polynomial terms
- $x_1, x_1^2, ..., x_1^n$: Polynomial terms of the input variable

3. Support Vector Regression (SVR)

Support Vector Regression aims to find a hyperplane that best fits the data points while maintaining robustness to outliers. **Support Vector Regression** works by finding the best line that closely matches the data while being able to ignore small errors or unusual values that don't follow the overall pattern. It tries to keep most predictions within a set range of acceptable error, ensuring the line is accurate but not overly sensitive to small, unexpected variations in the data. The SVR equation is represented as:

$f(x) = w^T x + b$

where:
- $f(x)$: Predicted output (house price)
- w: Weights vector
- x: Input variables (features)
- b: Bias term

4. Decision Tree

Decision Tree is a non-linear algorithm that splits the data into branches based on decision rules. At each step, it makes a decision based on the values of the features, creating "branches" that lead to different outcomes. It divides the data into sections, with each section representing a specific range of values. In each section, the model assigns an average value based on the data in that part. The tree continues to split until it can no longer divide the data meaningfully, making the predictions simple and straightforward. The algorithm uses the following recursive equation:

$y = \mathrm{mean}(y_i \mid x \in \mathrm{region})$

where:
- y: Predicted output (house price)
- y_i: Observed values of the target variable within the region
- x: Input variables (features)

5. Random Forest

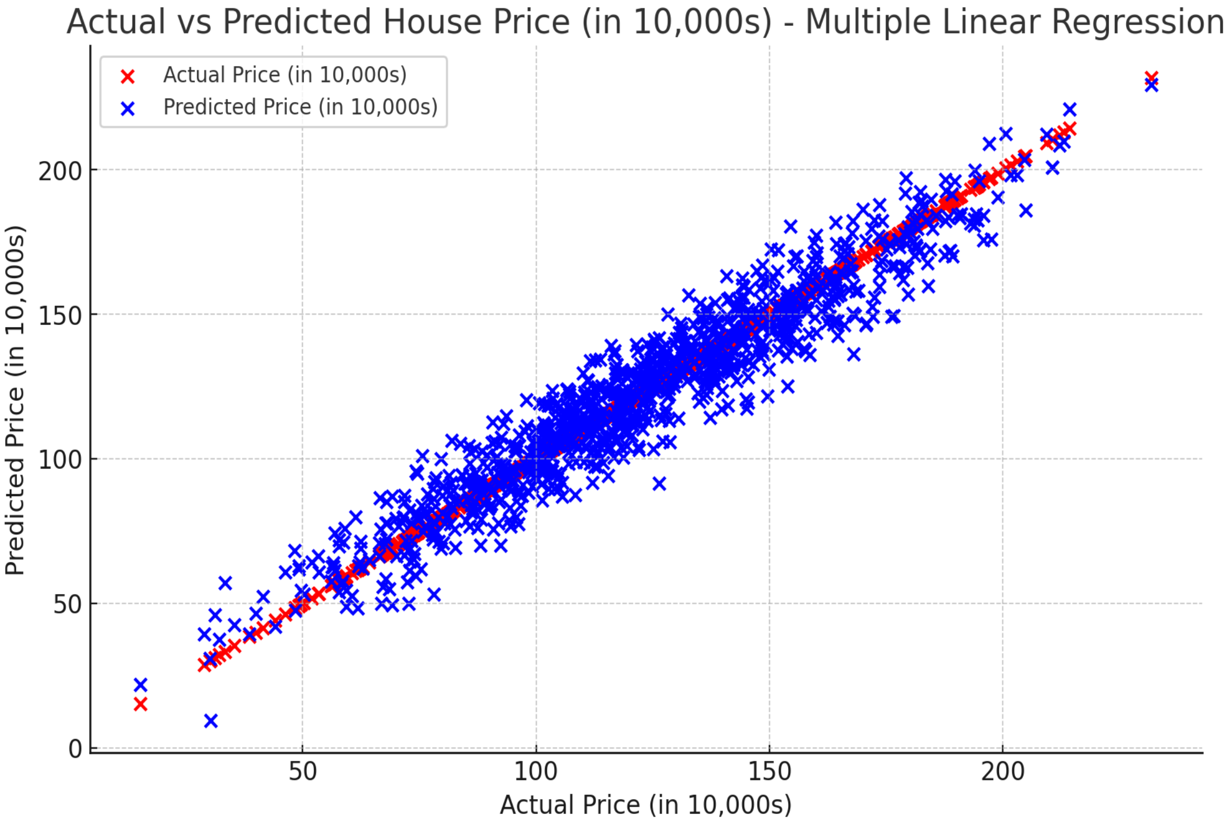
Random Forest is an ensemble learning algorithm that combines multiple decision trees to enhance accuracy and reduce overfitting. The prediction is based on the average of individual tree predictions. The equation is:

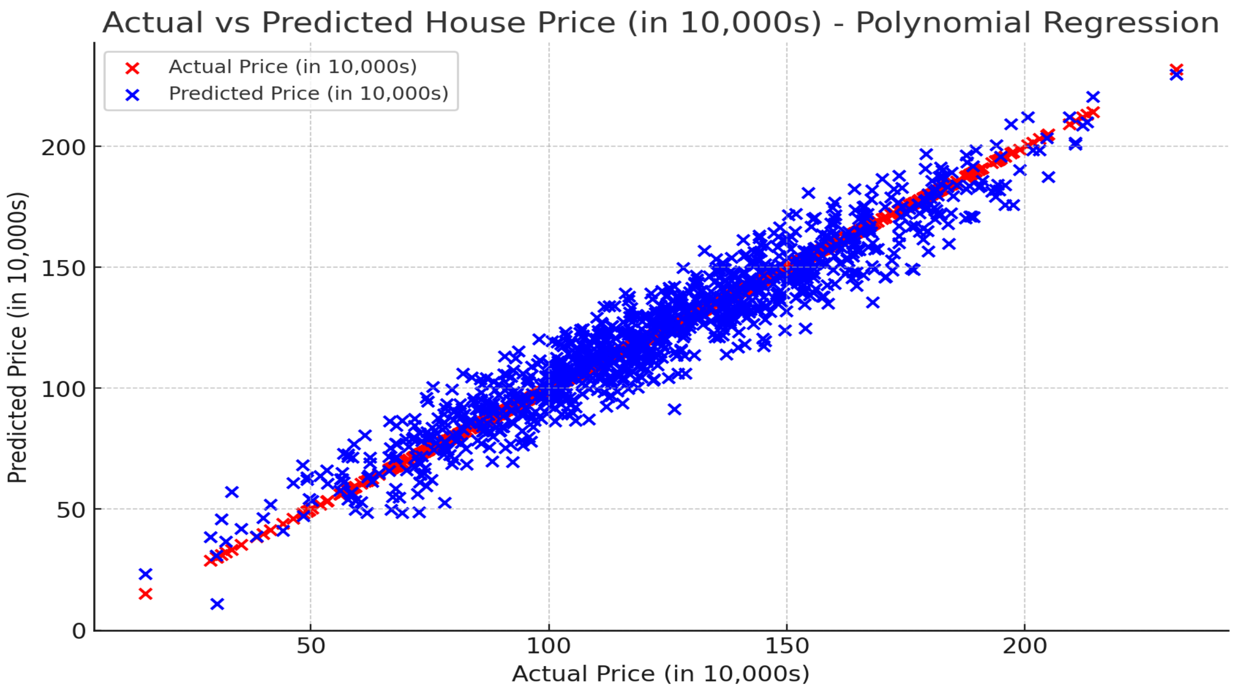
$y = (1/T) * \Sigma f_i(x)$

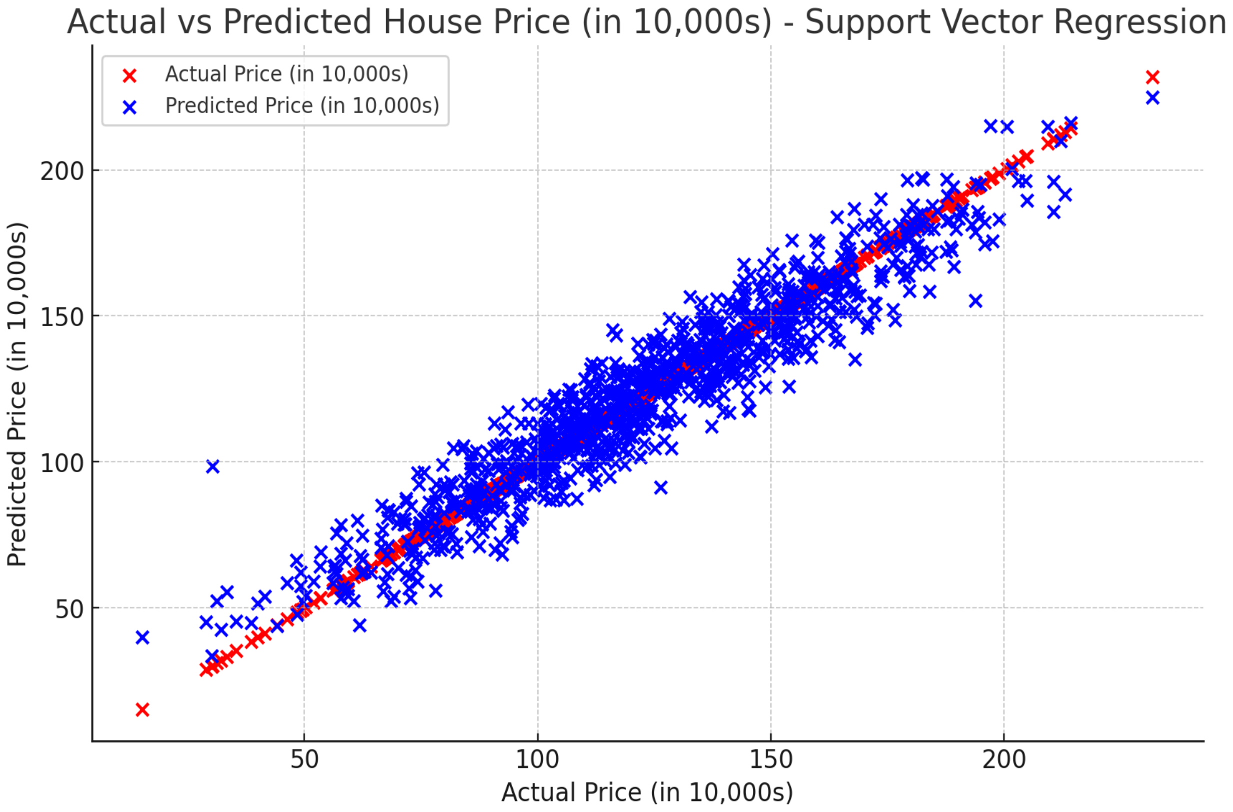
where:
- y: Predicted output (house price)
- T: Total number of decision trees in the forest
- $f_i(x)$: Prediction from the ith decision tree

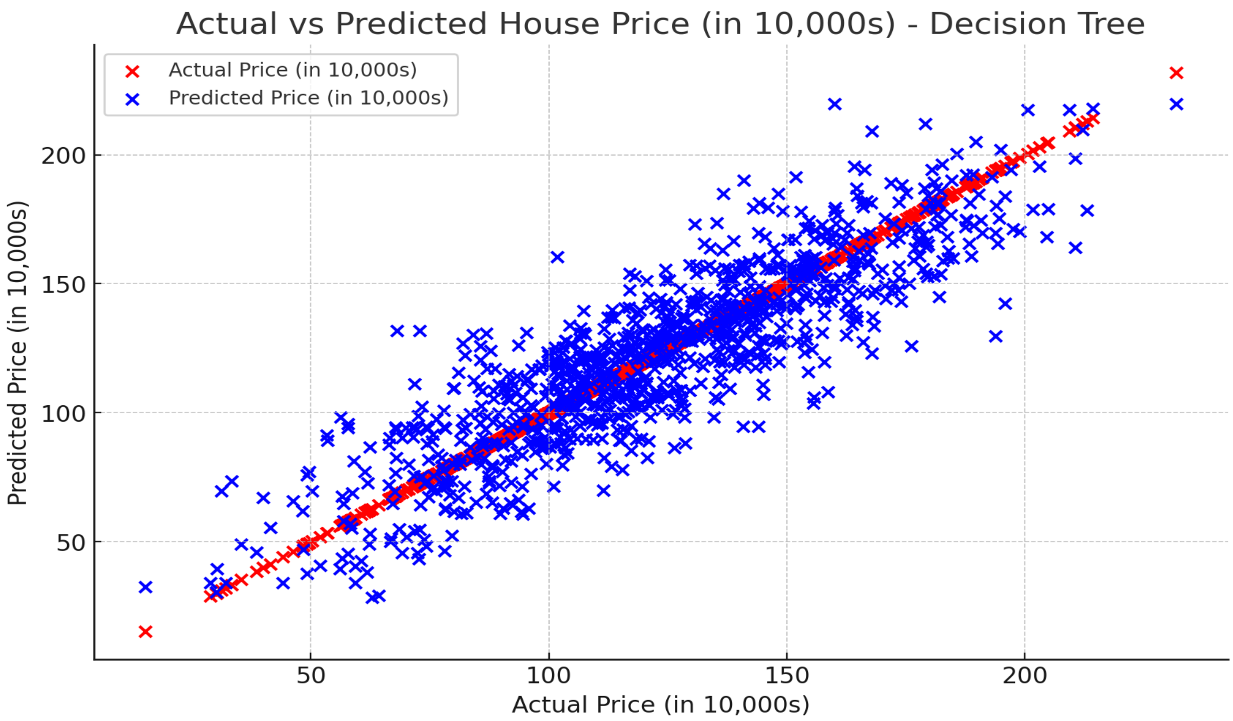
3. Results and Analysis

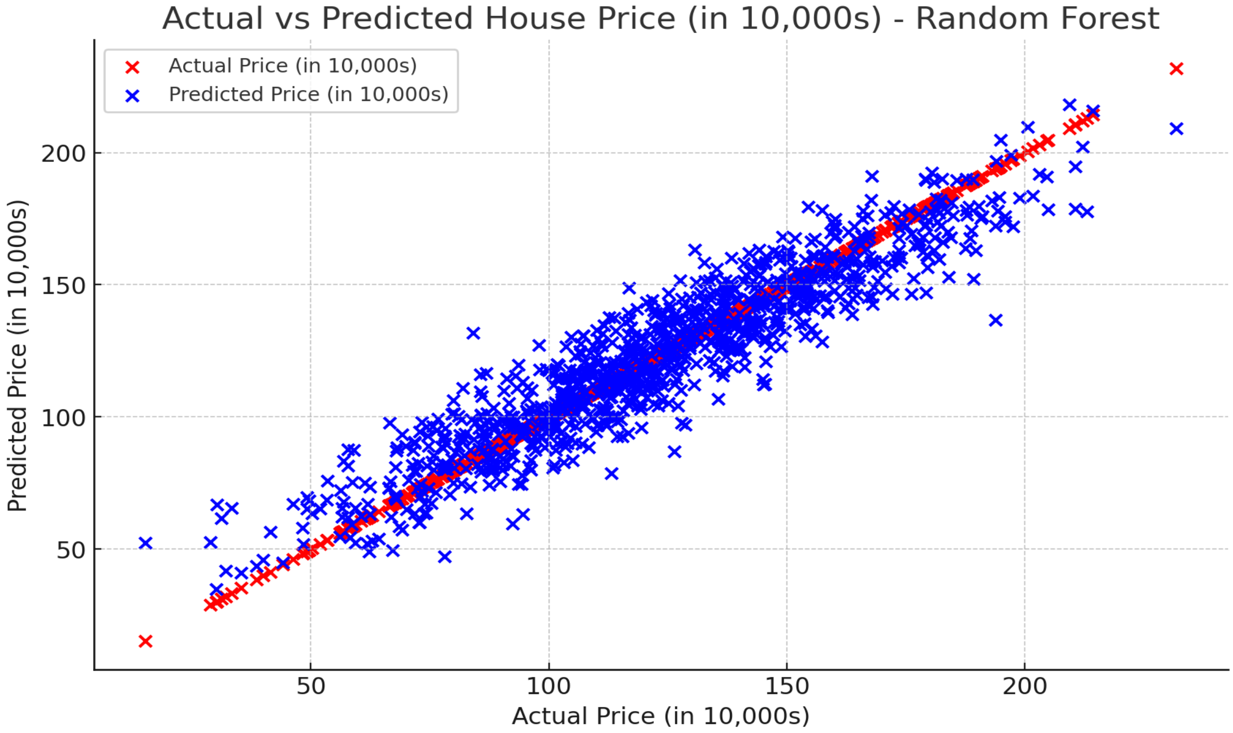
The scatter plots illustrate the actual vs. predicted house prices for each of the five models. In these plots, the red dots represent the actual house prices, while the blue dots represent the predicted prices.




Actual vs Predicted House Price (in 10,000s) - Polynomial Regression
× Actual Price (in 10,000s)
× Predicted Price (in 10,000s)
Predicted Price (in 10,000s)
0
50
100
150
200
50
100
150
200
Actual Price (in 10,000s)


Actual vs Predicted House Price (in 10,000s) - Support Vector Regression
× Actual Price (in 10,000s)
× Predicted Price (in 10,000s)
Predicted Price (in 10,000s)
50
100
150
200
50
100
150
200
Actual Price (in 10,000s)


Actual vs Predicted House Price (in 10,000s) - Decision Tree
× Actual Price (in 10,000s)
× Predicted Price (in 10,000s)
Predicted Price (in 10,000s)
200
150
100
50
50
100
150
200
Actual Price (in 10,000s)


Actual vs Predicted House Price (in 10,000s) - Random Forest
× Actual Price (in 10,000s)
× Predicted Price (in 10,000s)
Predicted Price (in 10,000s)
200
150
100
50
50
100
150
200
Actual Price (in 10,000s)

Based on the scatter plots for the five models, **Multiple Linear Regression** and **Polynomial Regression** show the closest alignment of predicted prices with actual prices. **Support Vector Regression** also performs relatively well but has slightly more deviation. **Decision Tree** and **Random Forest** models show more spread, with the Decision Tree model showing the most variation from actual values. Overall, the models that assume linearity seem to generalize better in this dataset, suggesting a primarily linear relationship between features and house prices.

Predicted values by the different models:

Actual Values	MLR	PR	SVR	DT	RF
932979.36	954717.19	950775.89	924464.08	849153.12	958170.04

Model Evaluation Results

Following are the evaluation results for each model:

Multiple Linear Regression

- R-Squared: 0.9146
- Mean Squared Error: 105.50
- Root Mean Squared Error: 10.27
- Normalized Root Mean Squared Error: 4.74%
- Mean Absolute Percentage Error: 7.48%

Polynomial Regression

- R-Squared: 0.9142
- Mean Squared Error: 106.05
- Root Mean Squared Error: 10.30
- Normalized Root Mean Squared Error: 4.75%
- Mean Absolute Percentage Error: 7.50%

Support Vector Regression

- R-Squared: 0.9041
- Mean Squared Error: 118.50
- Root Mean Squared Error: 10.89
- Normalized Root Mean Squared Error: 5.03%
- Mean Absolute Percentage Error: 8.07%

Decision Tree

- R-Squared: 0.7419
- Mean Squared Error: 319.05
- Root Mean Squared Error: 17.86
- Normalized Root Mean Squared Error: 8.25%
- Mean Absolute Percentage Error: 12.99%

Random Forest

- R-Squared: 0.8676
- Mean Squared Error: 163.67
- Root Mean Squared Error: 12.79
- Normalized Root Mean Squared Error: 5.91%
- Mean Absolute Percentage Error: 9.58%

Best Model for the Dataset

Based on the evaluation results, Multiple Linear Regression emerged as the best-performing model for predicting house prices in the dataset. It achieved the highest R-Squared value (0.9146), indicating a strong correlation between the features and the target variable. It also had the lowest MSE, RMSE, NRMSE, and MAPE compared to other models, making it the most suitable choice for this dataset.

References:

Link to dataset: https://www.kaggle.com/datasets/gopalchettri/usa-housing/data

https://www.geeksforgeeks.org/machine-learning/#supervised-learning-