House Price Prediction Using Multiple Regression Models

Abstract

This study investigates the performance of various machine learning regression models in predicting house prices using the California housing dataset. The objective is to compare different models on the basis of training time, testing time, and predictive accuracy measured by Mean Squared Error (MSE). A comprehensive analysis of model performances is provided, supported by visualizations that illustrate the trade-offs between accuracy and computational efficiency.

1. Introduction

Predicting house prices is a complex but essential task in the real estate and financial industries, as accurate predictions enable better investment and pricing decisions. Machine learning provides a range of algorithms for regression tasks, each with unique advantages and computational requirements. This report evaluates the performance of seven popular regression models: Linear Regression, Support Vector Machine (SVM), K-Nearest Neighbors (KNN), Decision Tree, Random Forest, Gradient Boosting, and XGBoost.

The California housing dataset is used as a benchmark to test and compare these models. The study emphasizes key metrics such as training time, testing time, and prediction accuracy, aiming to identify a model that balances efficiency with accuracy for practical applications.

2. Methodology

2.1 Dataset

The California housing dataset from Scikit-Learn provides median house prices along with several features, including location and socio-economic factors. This dataset has been preprocessed to fit the requirements of each machine learning algorithm.

2.2 Models Trained

We trained seven machine learning models for regression analysis:

1. **Linear Regression** - A basic linear model that establishes a linear relationship between features and target.
2. **Support Vector Machine (SVM)** - A non-linear model that seeks an optimal hyperplane in higher-dimensional space, tuned for regression.
3. **K-Nearest Neighbors (KNN)** - A non-parametric model that predicts target values based on the closest training samples.
4. **Decision Tree** - A model that splits the data into branches to make decisions based on feature values.

5. **Random Forest** - An ensemble method that builds multiple decision trees and averages their outputs for better accuracy.
6. **Gradient Boosting** - An ensemble technique that sequentially builds models, each correcting the errors of its predecessors.
7. **XGBoost** - An optimized gradient boosting framework known for efficiency and accuracy in predictive modeling.

2.3 Evaluation Metrics

Each model was evaluated based on the following metrics:

- **Training Time**: The time taken by each model to fit on the training dataset.
- **Testing Time**: The time required to make predictions on the test dataset.
- **Mean Squared Error (MSE)**: A measure of the average squared difference between predicted and actual values, providing a quantifiable assessment of accuracy.

3. Results

3.1 Model Performance Comparison

The results for each model in terms of training time, testing time, and MSE are summarized below:

Model	Training Time (s)	Testing Time (s)	Mean Squared Error
Linear Regression	0.011	0.001	0.556
SVM	20.253	8.795	1.332
KNN	0.031	0.035	1.119
Decision Tree	0.671	0.003	0.494
Random Forest	51.693	0.149	0.256
Gradient Boosting	10.900	0.013	0.294
XGBoost	1.175	0.050	0.223

3.2 Analysis of Results

- **Linear Regression**: The model is computationally efficient with low training and testing times. However, its MSE is higher than other advanced models, indicating limited predictive accuracy.
- **SVM**: SVM has the highest training and testing times, and it also has a higher MSE compared to other models. This suggests that SVM may not be suitable for large datasets in real-time applications due to computational costs.
- **KNN**: While KNN has low training time, its testing time and MSE are relatively high, showing a trade-off between simplicity and predictive accuracy.
- **Decision Tree**: This model achieves a reasonable balance between speed and accuracy but is outperformed by ensemble methods.
- **Random Forest**: Random Forest demonstrates a low MSE, indicating strong predictive performance. However, it has a high training time, making it more computationally expensive.
- **Gradient Boosting**: With a relatively low MSE and moderate training time, Gradient Boosting is effective for accurate predictions, though it requires more computation than simpler models.
- **XGBoost**: This model achieves the lowest MSE, indicating the highest accuracy among all tested models, while maintaining moderate training and testing times, making it suitable for practical use in predictive tasks.

3.3 Visualization

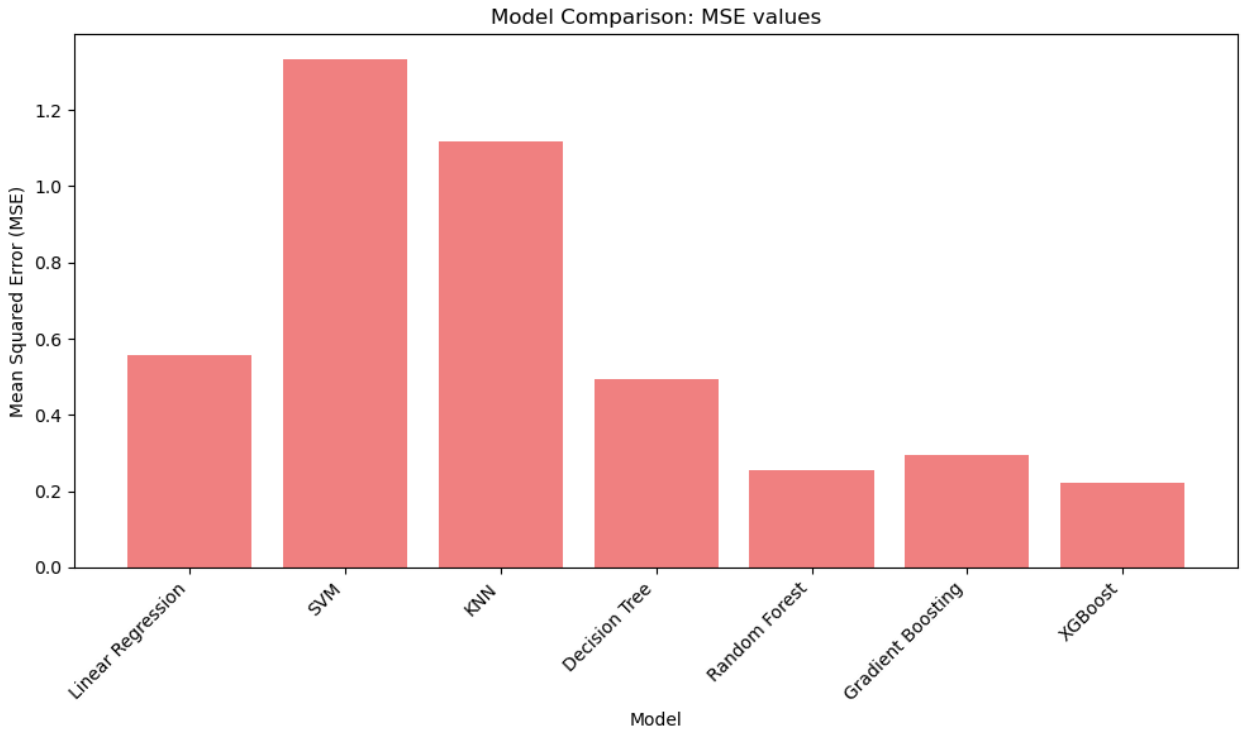

Figure 1: The following graph compares the Mean Squared Error (MSE) values of each regression model. This visualization helps illustrate the accuracy of each model, with lower MSE values indicating better performance:

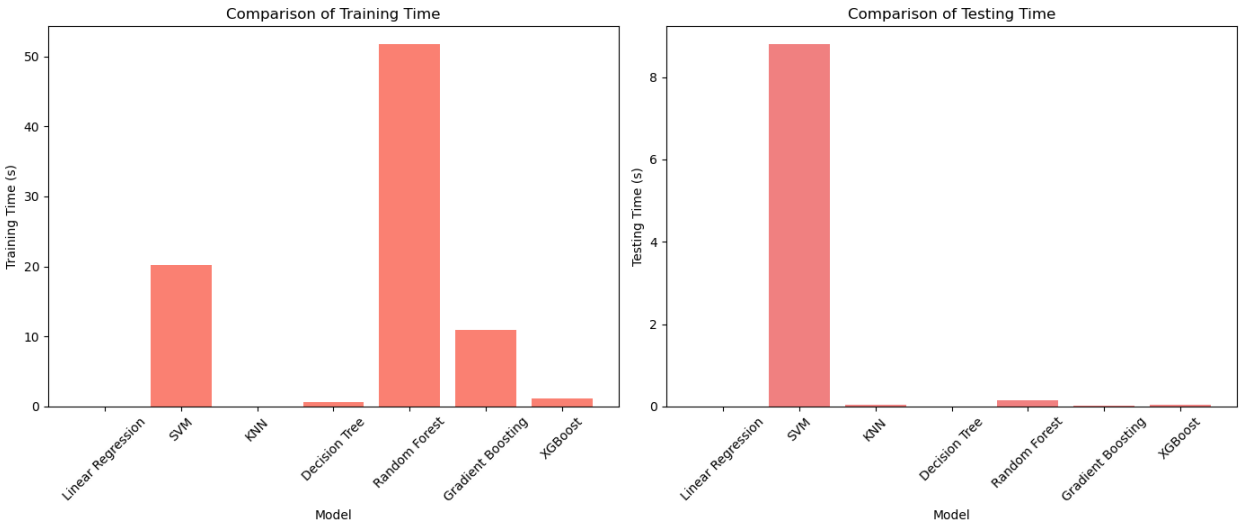

Figure 2: The following image presents a time comparison of the models in terms of training and testing times. The graph on the left shows the training times, while the graph on the right compares the testing times. This visualization illustrates the computation

4. Conclusion

This report highlights the trade-offs between model accuracy and computational efficiency in predicting house prices using the California housing dataset. While traditional models like Linear Regression and KNN are computationally efficient, their predictive accuracy is limited. Ensemble methods, especially XGBoost, show superior accuracy with a manageable computational cost, making it an ideal choice for high-stakes, real-time applications.

Future work could involve fine-tuning model hyper parameters, exploring other datasets, and investigating models' interpretability to enhance real-world applicability in house price prediction.