Predicting Housing Prices using Machine Learning

A Comparative Analysis of Regression Models on Two Datasets by Ameen Hussain and Sarthak Bhardwaj

Project Overview and Significance

Accurate housing price prediction is crucial for the real estate industry, buyers, sellers, and investors

The project aims to compare the performance of different regression models on two housing datasets (Boston Housing and Miami Housing) to predict housing prices accurately

The goal is to identify the best-performing models and provide insights into the factors influencing housing prices

Methodology

Conducted a literature review to understand the existing research in housing price prediction using machine learning techniques

Explored and preprocessed two housing datasets: Boston Housing and Miami Housing

Boston Housing dataset: 506 instances, 13 features

Miami Housing dataset: 13,932 instances, 16 features

Implemented data preprocessing steps, including handling missing values, feature selection, and train-test split

Applied various regression models, including Linear Regression, K-Nearest Neighbors Regressor, Decision Tree Regressor, Random Forest Regressor, and Support Vector Regressor

Evaluated the models using the R-squared (R2) score metric

Our Datasets

Dataset 1: Boston Housing

Description: The Boston Housing dataset contains information about various features of houses in Boston and their corresponding median values

Number of instances: 506

Number of features: 13

Target variable: 'medv' (median value of owner-occupied homes in $1000s)

Dataset 2: Miami Housing

Description: The Miami Housing dataset provides information about housing properties in Miami, including their features and sale prices

Number of instances: 13,932

Number of features: 16

Target variable: 'SALE_PRC' (sale price of the property)

Data Preprocessing

Data Preprocessing

Handling missing values: Removed instances with missing values in the Boston Housing dataset using df.dropna()

Feature selection: Dropped the target variable from the feature set using df.drop()

Train-test split: Split the data into training and testing sets using train_test_split from scikit-learn

Regression models used

Linear Regression: A simple and interpretable model that assumes a linear relationship between features and the target variable

K-Nearest Neighbors Regressor: A non-parametric model that predicts based on the average of the k nearest neighbors

Decision Tree Regressor: A model that learns decision rules from the data to make predictions

Random Forest Regressor: An ensemble model that combines multiple decision trees to improve performance and reduce overfitting

Support Vector Regressor: A model that tries to fit the data points within a certain margin while minimizing the error

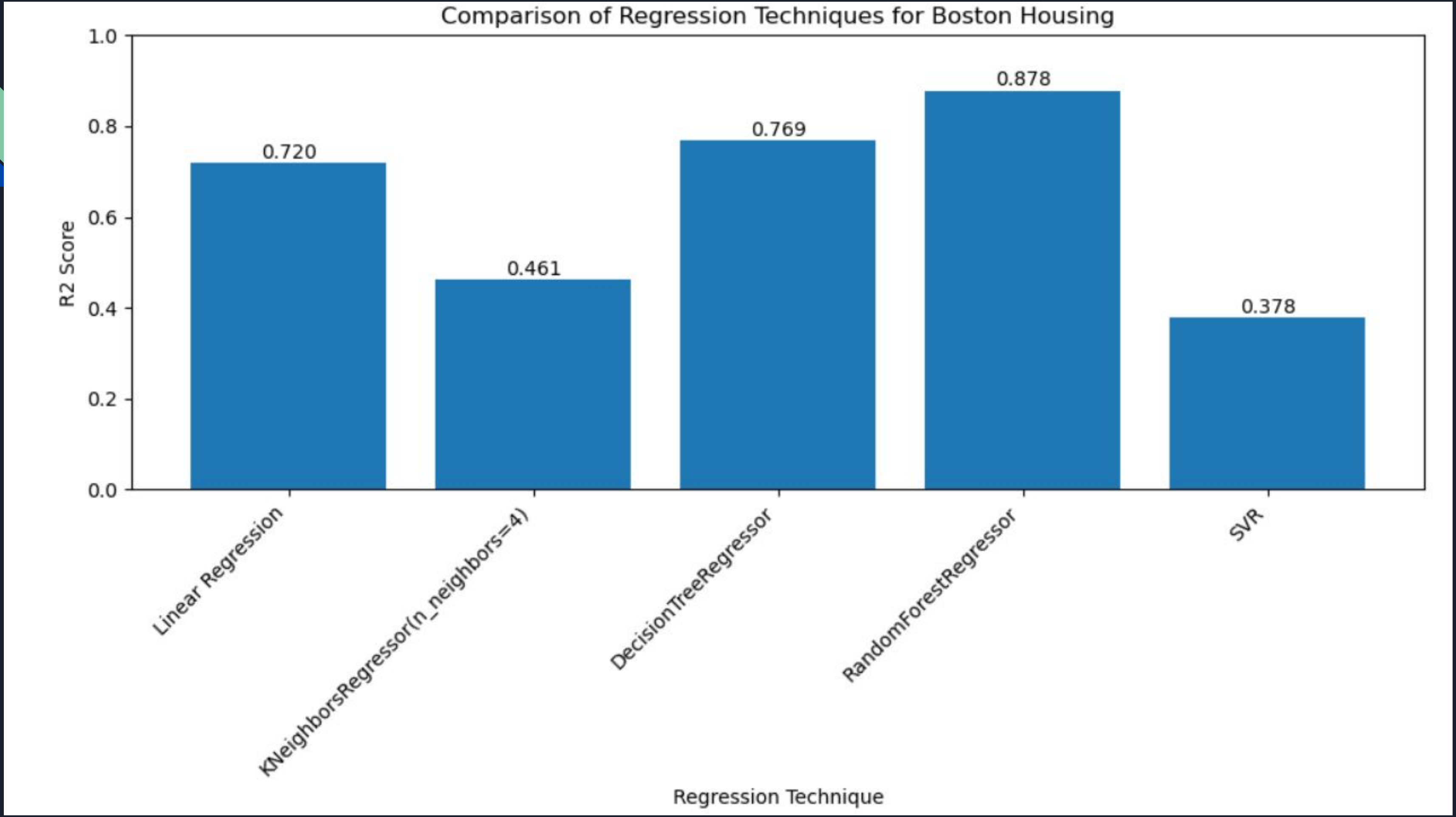
Preliminary Results and conclusions

Boston Housing Dataset:

The Random Forest Regressor achieved the highest R2 score of 0.878329, indicating a strong fit to the data. This suggests that the Random Forest model was able to capture the complex relationships between the features and the target variable effectively.

The Decision Tree Regressor also performed well with an R2 score of 0.769046, followed by Linear Regression with an R2 score of 0.720028. These models were able to capture a significant portion of the variance in the target variable.

The K-Nearest Neighbors Regressor (with n_neighbors=4) and Support Vector Regressor (SVR) had lower R2 scores of 0.461170 and 0.378342, respectively. This indicates that these models may not be the best choices for this particular dataset.


Comparison of Regression Techniques for Boston Housing
R2 Score
1.0
0.8
0.6
0.4
0.2
0.0
0.720
0.461
0.769
0.878
0.378
Linear Regression
KNeighborsRegressor(n_neighbors=4)
DecisionTreeRegressor
RandomForestRegressor
SVR
Regression Technique

Preliminary Results and conclusions

Miami Housing Dataset:

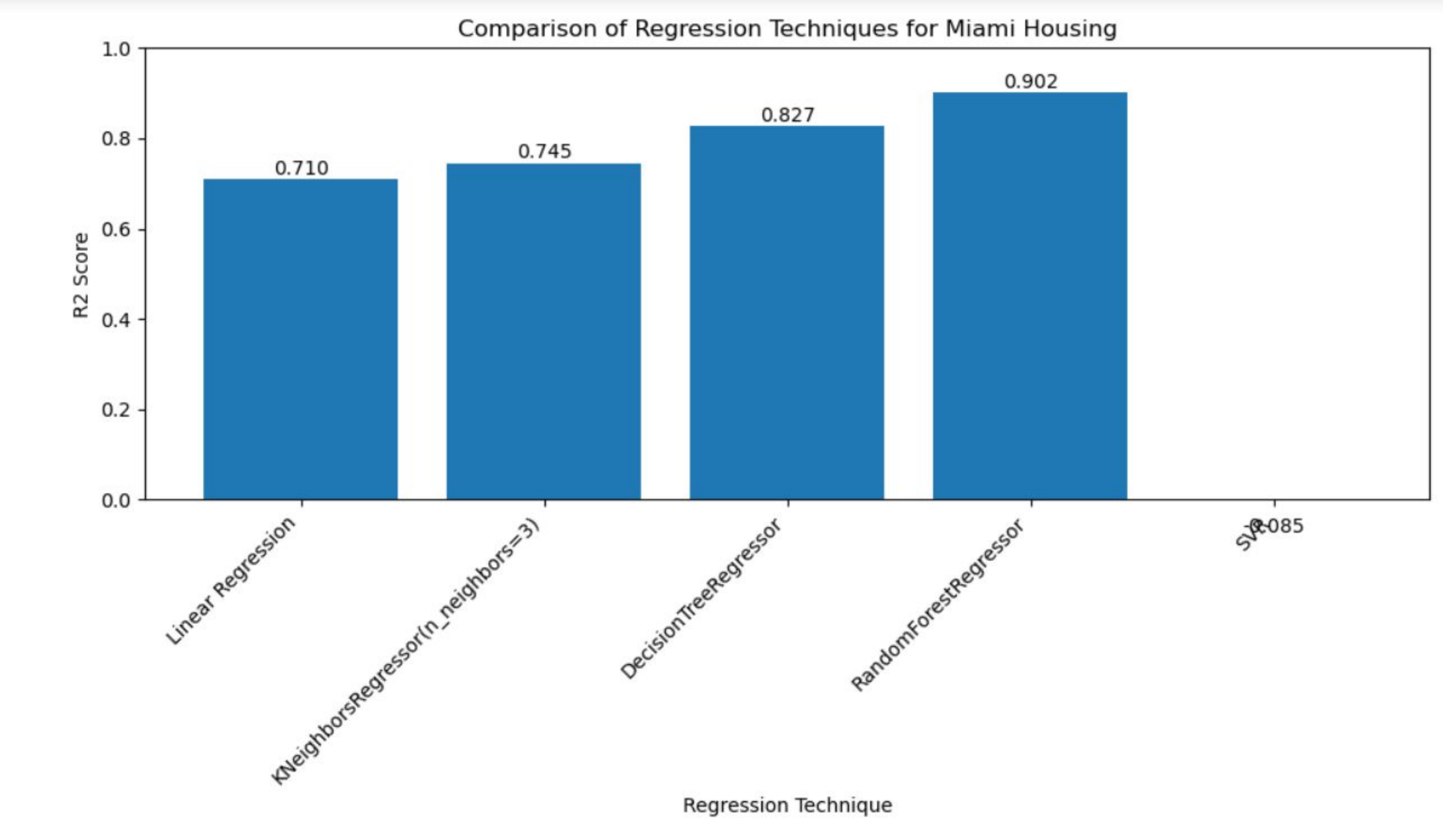
The Random Forest Regressor again achieved the highest R2 score of 0.902451, demonstrating its ability to capture the complex relationships in the data and provide accurate predictions.

The Decision Tree Regressor also performed well with an R2 score of 0.827196, indicating its effectiveness in modeling the housing prices.

The K-Nearest Neighbors Regressor (with n_neighbors=3) showed a reasonably good performance with an R2 score of 0.745114.

Linear Regression had a moderate R2 score of 0.709649, suggesting a linear relationship between the features and the target variable.

The Support Vector Regressor (SVR) had a negative R2 score of -0.084694, indicating that it performed poorly on this dataset. This suggests that the SVR model may not be suitable for this particular problem.


Comparison of Regression Techniques for Miami Housing
R2 Score
1.0
0.8
0.6
0.4
0.2
0.0
0.710
0.745
0.827
0.902
-0.085
Linear Regression
KNeighborsRegressor(n_neighbors=3)
DecisionTreeRegressor
RandomForestRegressor
SVR
Regression Technique

Preliminary Results and conclusions

Comparison between Datasets:

The Random Forest Regressor consistently performed the best on both datasets, with R2 scores above 0.87. This highlights the robustness and effectiveness of the Random Forest model in capturing complex patterns and providing accurate predictions.

The Decision Tree Regressor also demonstrated good performance on both datasets, with R2 scores above 0.76. This suggests that decision trees can be a viable option for regression tasks in housing price prediction.

Linear Regression showed moderate performance on both datasets, with R2 scores around 0.70. This indicates that there is a linear relationship between the features and the target variable to some extent.

The K-Nearest Neighbors Regressor performed better on the Miami Housing dataset compared to the Boston Housing dataset. This suggests that the effectiveness of KNN may depend on the specific characteristics of the dataset.

The Support Vector Regressor (SVR) performed poorly on both datasets, indicating that it may not be the best choice for these particular regression problems.

Next steps

Explore advanced regression techniques:

Implement Gradient Boosting Regressors, such as XGBoost and LightGBM, to capture complex non-linear relationships and improve prediction accuracy

Investigate the use of Neural Networks, particularly Deep Learning models like Multi-Layer Perceptrons (MLPs) and Convolutional Neural Networks (CNNs), to learn intricate patterns in the housing data

Compare the performance of these advanced techniques with the existing regression models to identify the most effective approach

Perform feature engineering:

Create new informative features by combining or transforming existing features, such as calculating the price per square foot or the ratio of bedrooms to bathrooms

Explore domain-specific features that may influence housing prices, such as proximity to amenities, school districts, or crime rates

Utilize techniques like polynomial features, interaction terms, or feature scaling to capture non-linear relationships and improve model performance

Next Steps

Conduct hyperparameter tuning:

Use techniques like Grid Search or Random Search to systematically explore and optimize the hyperparameters of the regression models

Fine-tune the hyperparameters of the advanced regression techniques, such as the learning rate, depth, and number of estimators in Gradient Boosting Regressors

Employ cross-validation to ensure the robustness and generalization ability of the tuned models

Expand evaluation techniques:

Implement additional evaluation metrics, such as Mean Absolute Error (MAE), Mean Squared Error (MSE), or Root Mean Squared Error (RMSE), to assess the prediction accuracy and error distribution

Employ techniques like cross-validation or bootstrapping to obtain more robust and reliable performance estimates

Compare the performance of the models across different subsets or segments of the housing data to identify any variations or biases

References

Hastie, T., Tibshirani, R., & Friedman, J. (2009). The Elements of Statistical Learning: Data Mining, Inference, and Prediction. Springer Science & Business Media.

Altman, N. S. (1992). An introduction to kernel and nearest-neighbor nonparametric regression. The American Statistician, 46(3), 175-185.

Quinlan, J. R. (1986). Induction of decision trees. Machine Learning, 1(1), 81-106.

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

Drucker, H., Burges, C. J., Kaufman, L., Smola, A. J., & Vapnik, V. (1997). Support vector regression machines. Advances in Neural Information Processing Systems, 9, 155-161.