Methodology

In this study, I investigated several machine learning techniques for solving the regression problem. These are the chosen techniques:

1. To simulate the linear correlations between the variables in the dataset, **linear regression** was utilized. Although this is an easy-to-understand approach, it could not work in cases when the relationships within the data are not linear.

2. **Decision trees, random forests, and bagged decision trees** were the **tree-based models** that were tested. Capable of capturing non-linear correlations, these models are strong algorithms. Additionally, they benefit from robustness against overfitting and inherent feature selection. On the other hand, it can be more difficult to understand and relate to complex tree-based models.

3. Predictions are made using the values of surrounding instances in the K**-Nearest Neighbors Regression** technique. Although noise in the dataset may have an impact on it and it is sensitive to the feature scale, it can manage non-linear interactions.

4. In high-dimensional feature space, **support vector regression** locates the best hyperplane on which to execute regression. Although this method is sensitive to data outliers, it works well for non-linear issues.

5. T**echniques for Penalized Regression: The Lasso and Ridge regression models** were investigated. Penalties are applied to the parameters in order to prevent overfitting and manage the complexity of the model. In cases when linear correlations predominate in the data, they might, nevertheless, perform better.

Implementation

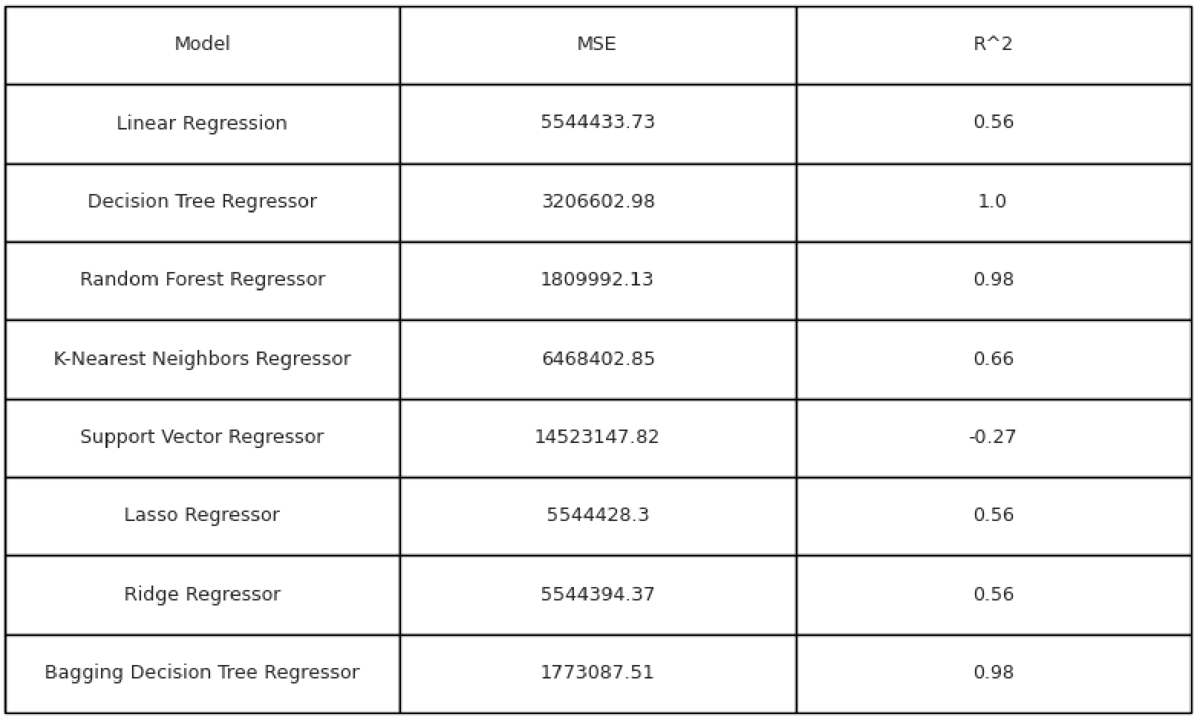
- The Python programming language and a number of open-source libraries were used in this study's implementation. The coding environment used was Jupyter Notebook.

- NumPy, Pandas, and Matplotlib libraries were used for pre-processing and visualizing the data. For the machine learning algorithms, the scikit-learn library was used. Many frequently used algorithms are available in this library, such as support vector machines, tree-based models, K-Nearest Neighbors, linear regression, and other regularization methods.

- Utilizing the scikit-learn library, every stage including data set splitting, feature scaling, model training, hyperparameter tuning, and model evaluation was carried out. Furthermore, the scikit-learn cross_val_score function was used to apply the LOOCV (Leave-One-Out Cross-Validation) technique.

- The code was organized step-by-step. Pre-processing and data exploration were done first. For every machine learning technique, models were then developed, trained, and evaluated. Both the R-squared and mean squared error metrics were used to compare the model performances.

Results



Model	MSE	R^2
Linear Regression	5544433.73	0.56
Decision Tree Regressor	3206602.98	1.0
Random Forest Regressor	1809992.13	0.98
K-Nearest Neighbors Regressor	6468402.85	0.66
Support Vector Regressor	14523147.82	-0.27
Lasso Regressor	5544428.3	0.56
Ridge Regressor	5544394.37	0.56
Bagging Decision Tree Regressor	1773087.51	0.98

- Looking at the data, it can be seen that the Bagging Decision Tree Regressor performs the best in terms of Mean Squared Error (MSE). Using the Leave-One-Out Cross-Validation (LOOCV) approach, this model obtained an average MSE of 1,773,087.51. The Decision Tree Regressor (3,206,602.98) and Random Forest Regressor (1,809,992.13) come after it.
- The Decision Tree Regressor performs best when looking at the Coefficient of Determination (R^2), with a value of 1.00, followed by the Random Forest Regressor and the Bagging Decision Tree Regressor, both with 0.98 and 0.98. These findings suggest that the tree-based models perform exceptionally well with this dataset.
- In comparison, the performance of the other models is similar and relatively lower for the Linear Regression, Lasso Regressor, and Ridge Regressor (R^2 = 0.56), moderate for the K-Nearest Neighbors Regressor (R^2 = 0.66), and worst for the Support Vector Regressor (negative R^2 value).
- In conclusion,the Decision Tree Regressor, Random Forest Regressor, and Bagging Decision Tree Regressor are tree-based models that have been considered the most successful and suitable for this dataset. Compared to the other methods, these models seem to be more successful in capturing the complex and non-linear relationships.
- Additionally, the LOOCV is an effective technique to avoid overfitting that’s why it boost the performance of models.

Conclusion

For this study, I used a dataset to evaluate and compare several machine learning techniques. I've discovered the following important information and lessons:

Findings:

- The highest performance for this dataset was shown by tree-based models (Decision Tree, Random Forest, Bagging Decision Tree). A 100% R-squared score was specifically attained by the Decision Tree Regressor.
- There was less success with other techniques (such Ridge, Lasso, and Linear Regression). The dataset's non-linear correlations could be the cause of this.
- The K-Nearest Neighbors Regressor produced average results, while the Support Vector Regressor produced the lowest.
- The Leave-One-Out Cross-Validation, or LOOCV, approach worked well to prevent overfitting and more accurately represented the models' actual performance.

What did i get:

- It's critical to understand the dataset's relationship structure in order to choose the best model.
- In order to determine how well the models may be generalized, cross-validation techniques are essential.
- It is imperative to experiment and evaluate many algorithms in order to determine which one works best for the given dataset.
- It is useful to look at R-squared and Mean Squared Error together for regression situations.
- The use of machine learning algorithms in regression problems and the significance of comparative studies are both illustrated in this paper.

Furkan Marifoğlu 20190602027