MATTHEW TOMLINSON

CIS2717

Weekly sales Models

Table of Contents

[Introduction 2](#_Toc187000806)

[Objectives 2](#_Toc187000807)

[Report overview 3](#_Toc187000808)

[Methodology 3](#_Toc187000809)

[Algorithms chosen 3](#_Toc187000810)

[The datasets 3](#_Toc187000811)

[Data pre-processing 4](#_Toc187000812)

[Random Forest implementation 4](#_Toc187000813)

[Gradient Boosting implementation 5](#_Toc187000814)

[Explanation of parameter ranges 5](#_Toc187000815)

[Grid search configurations 5](#_Toc187000816)

[Weighting 5](#_Toc187000817)

[Algorithm strengths and weaknesses 6](#_Toc187000818)

[Experiments 6](#_Toc187000819)

[Evaluation metrics used 6](#_Toc187000820)

[Algorithm performances 7](#_Toc187000821)

[Random Forest Regressor 7](#_Toc187000822)

[Gradient Boosting Regressor 8](#_Toc187000823)

[Performance comparison 8](#_Toc187000824)

[Feature importance 9](#_Toc187000825)

[Outputs 10](#_Toc187000826)

[Summary 14](#_Toc187000827)

[References 16](#_Toc187000828)

# Introduction

In the industry of retail, an accurate forecasting of sales is crucial for effective decision making. Walmart is one of the largest retail chains internationally, and it faces the problem of predicting its department wide sales across its countless stores, this is also influenced by holidays. Effective weekly sales prediction can allow for the optimisation of operations and customer satisfaction.

This projects focus is to develop multiple machine learning models, using the Python programming language, to predict weekly department sales for Walmart stores using historical data provided. Machine learning algorithms can capture non-linear relationships and complex interactions between features, making them well suited for sales forecasting in a retail setting where factors like promotions and holidays play a significant role. The dataset provided spans over 45 different stores, each of these stores contain multiple departments, the dataset includes features such as weekly sales, promotional markdowns, regional temperature, fuel prices, consumer price index (CPI), and unemployment rates. In addition, the dataset makes an emphasis on holiday weeks, such as the Super Bowl, Labor Day, Thanksgiving, and Christmas. These periods significantly affect weekly sales, to reflect this, weekly sales over holiday weeks are weighted five times higher in the evaluation metric.

Jupyter notebook was used in this project to implement the code for both models.

## Objectives

These objectives aim to provide insights and tools that can optimise operations, reduce costs, and improve decision-making for Walmart.

* **Algorithm development –** develop machine learning algorithms capable of predicting weekly sales for each store.
* **Feature engineering –** preprocess features in the dataset, ensuring the model effectively captures the factors that drive sales, including promotional markdowns and holiday effects.
* **Performance evaluation –** evaluate the predictive accuracy of the model using evaluation metrics such as mean absolute error, root mean squared error etc. ensuring the model performs adequately during holiday periods.
* **Parameter optimisation –** fine-tune the models hyperparameters to achieve optimal performance on the test dataset.
* **Performance comparison –** compare the performances of the algorithms implemented using their evaluation metrics, making sure different parameter settings are used.

## Report overview

This report will include a summary of the methodology, experiments and results of this project. It starts with explaining the data preparation and the machine learning algorithms chosen for this task. The report then moves onto the experiments performed on the models and the corresponding outputs, highlighting the model’s performance using a variety of parameter settings and metrics. Finally, the report will conclude with a summary of the implementation of the project and how well it aligns with the initial requirements set out, a comparison of the algorithms used, along with discussing any implications and potential improvements that could be added.

# Methodology

## Algorithms chosen

For this project, the Random Forest Regressor (RF) and Gradient Boosting Regressor (GBR) machine learning algorithms were chosen. These algorithms were developed to predict weekly sales for Walmart stores based on historical sales data, external features and store characteristics. The main objective was to evaluate the performance of these models in accurately predicting sales, while having an emphasis on holiday weeks through a custom weighted evaluation metric.

The Random Forest Regressor is a versatile machine-learning technique used for predicting values. It achieves this by combining the predictions of multiple decision trees to reduce overfitting and improve accuracy according to (DUTTA, A, 2024). The Random Forest randomly selects features at each split, ensuring robust predictions.

The Gradient Boosting Regressor is a boosting algorithm in machine learning which is used for classification and regression tasks, it is an ensemble learning method which sequentially trains the decision trees and tries to correct the previous tree (NIKKI, 2023).

## The datasets

4 datasets had been provided to aid with the implementation of this project. These datasets are stores.csv, train.csv, test.csv, and features.csv. The store.csv dataset includes anonymized information about the 45 Walmart stores, displaying the size and type of store too. The features.csv dataset includes additional data related to these stores, departments, and the regional activity for the corresponding dates. These were all merged with the train.csv and the test.csv datasets to form train\_merged (train\_merged shown in Figure 3) and test\_merged datasets, ensuring all relevant information was centralised. The train\_merged dataset includes historical sales from the Walmart retail chain between 2010-02-05 and 2012-11-01. The dataset includes external such as temperature, fuel prices, consumer price index (CPI), and unemployment rates. The indicator named ‘IsHoliday’ is used to capture the impact of holidays on weekly sales. The test\_merged dataset contains the same features as the train\_merged dataset, but it excludes the ‘Weekly\_Sales’ column, as this is what will be predicted. The final training dataset consists of 42, 240 training instances and 45 features.

## Data pre-processing

Many steps were taken to clean and prepare the data before being processed. Firstly, categorical feature processing was carried out, this ensured that all columns in the train\_merged and test\_merged datasets will be numerical. Then the missing values were handled by calculating the mean of each column and filling in the missing values with the calculated mean. Then more feature engineering was carried out to add new features to the datasets and enhance the predictive capabilities of the models. Columns that act as a flag for each type of holiday were added (e.g. Is\_Christmas, Is\_Thanksgiving etc.), features were also added to capture the significant impact that the different holidays had on sales patterns. More features were added, these were time-based features like Year, Month, and Week. This allowed the models to capture temporal patterns in sales data, like seasonal trends or weekly and monthly fluctuations. The dataset after being pre-processed is shown in Figure 4.

## Random Forest implementation

The Random Forest algorithm (RF) was initially configured with 10 decision trees (‘n\_estimators = 10’) during early testing, allowing for the comparison of the model’s performance before and after hyperparameter tuning was performed. The initial predictions of the Random Forest Regressor are shown in Figure 5. According to (VERMA A, 2023), hyperparameters are the parameters set before training a model, they have a significant impact on the performance of the model, so they need to be carefully chosen. The model’s hyperparameters are tuned using GridSearchCV, according to (MALATO, G, 2021) a grid search is the simplest algorithm used for hyperparameter tuning, dividing the domain of hyperparameters into a discrete grid, trying very combination of values in the grid and calculating the performance using cross-validation. This grid search explores parameters such as ‘n\_estimators ’, ‘max\_depth’, and ‘min\_samples\_split’. It used values of 10, 20, and 50 for ‘n\_estimators’, this parameter specifies the number of decision trees in the forest, increasing this parameter can improve the performance of the model, however, it can also increase the computational cost of training and predicting (SHAFI, A, 2024). Values of 10, 20, and 50 were explored for ‘max\_depth’, this parameter is defined as the longest path between the tree’s root node and leaf node, it allows for the limitation of the tree’s depth (SAXENA, S, 2024). Finally, values of 2, 5, and 10 were explored for ‘min\_samples\_split’, this parameter represents the minimal number of samples required to split an internal node, it can vary between considering at least one sample at each node, to considering all samples at each node (MOHTADI, B F, 2017).

## Gradient Boosting implementation

The Gradient Boosting Regressor (GBR) was initially configured with 100 decision trees (‘n\_estimators’), a learning rate of 0.1(‘learning\_rate = 0.1’), this parameter is simply defined as how fast the model learns, the lower the learning rate, the slower the model learns, a lower learning rate can allow for a more robust and efficient model (MACHINE LEARNING+, n.d). A maximum depth of 3 (‘max\_depth = 3’) was also used in the model’s initial configuration. The initial predictions of the Gradient Boosting Regressor is shown in Figure 6. Hyperparameter tuning was performed on this model too, using GridSearchCV. Experimenting with values of 10, 20, and 50 for the number of estimators. Then 0.01, 0.05, and 0.1 for the learning rate, then 5, 10, and 20 for the maximum depth of the decision trees.

## Explanation of parameter ranges

It could be argued that the ranges of the different parameters used for training the two models are relatively small. Smaller parameter ranges allow for computational efficiency, due to the size of the dataset being used and the resource intensiveness of the algorithms used, a larger range of parameters would significantly increase training times. Smaller parameter ranges also allow me to use the full dataset, increasing generalisation and higher confidence in results.

## Grid search configurations

Both model’s grid searches use 2 folds (‘cv = 2’), the data is divided into several subsets, in this case 2 folds, then the model is trained and tested multiple times, with a new fold used for validation each time. This offers a more accurate estimation of the model’s performance and avoids overfitting using averaging (HENDRICKS, R, n.d). Both models use a verbose of 2 for their GridSearchCV’s (‘verbose = 2’), this parameter controls the level of information displayed in the console during the grid search process, a verbose of 2 provides detailed information, this helps us track the progress of the grid search. However, the output can become overwhelming, this needs to be considered. Next, the models both utilise all available CPU cores for their grid searches (‘n\_jobs = 1’), allowing for the use of all available resources, making the hyperparameter tuning process a lot faster. The grid search output for the RF is shown in Figure 9, and the grid search output for the GBR is shown in Figure 10.

## Weighting

The custom weighted mean absolute error (WMAE) was employed as the primary evaluation metric for the models, it places a higher weight (5x) on holiday weeks to ensure the models perform well during these periods. This adjustment reflects the business importance of accurate predictions during weeks of high revenue.

## Algorithm strengths and weaknesses

Both algorithms have strengths and weaknesses. The Random Forest is highly interpretable and performs well with busy datasets because of its averaging mechanism. However, it struggles with fine-tuned adjustments due to the trees being trained independently. On the other hand, Gradient Boosting is more prone to overfitting on smaller datasets but is highly effective in capturing detailed relationships between features.

# Experiments

This section focuses on presenting the experiments performed to evaluate the performance of the implemented machine learning algorithms: Random Forest Regressor, and Gradient Boosting Regressor. These experiments were designed to assess and evaluate the predictive accuracy and robustness of the models in forecasting weekly sales, with a strong emphasis on handling holiday periods. This is achieved through evaluation metrics, both models were assessed using multiple metrics to provide a cohesive comparison of their effectiveness.

## Evaluation metrics used

The experiments focused on predicting the target variable “Weekly\_Sales” using the features provided in the merged datasets. Two ensemble algorithms, Random Forest Regressor and Gradient Boosting Regressor were selected because of their ability to handle non-linear relationships. The models were evaluated using the Weighted Mean Absolute Error (WMAE), Holiday Mean Absoluter Error (Holiday MAE), Non-holiday Mean Absolute Error (Non-holiday MAE), Root Mean Squared Error (RMSE), and R^2 score. The WMAE metric was prioritised because it penalised errors during holiday periods five times more heavily than regular weeks, aligning with the business goal of effectively predicting weekly sales during holiday periods.

Mean Absolute Error (MAE) refers to the magnitude of difference between the prediction of an observation and the actual value, taking the average of absolute errors for a group of predictions (C3.AI, n.d). RMSE is the standard deviation of the prediction errors, measuring how spread out these errors are (STATISTICS HOW TO, n.d). R^2 measures how well the independent variables in a model explain the variation of the dependent variable, ranging from 0 to 1 (FERNANDO, J, 2024).

The weighted mean absolute error (WMAE) gives five times more weight to holiday weeks than non-holiday weeks, it combines the errors from both holiday and non-holiday weeks but adjusts the importance of holiday weeks in the calculation. This metric reflects the overall performance of the model while prioritising accurate predictions during holiday weeks. Useful for when holiday periods are more important for the business.

The holiday mean absolute error (Holiday MAE) represents the mean absolute error for holiday weeks specifically. The purpose of this metric is to isolate the performance of the model during holiday periods, allowing for the assessment of how well it predicts during these weeks. It helps understand the contribution of holiday periods to the overall performance of the model.

The non-holiday mean absolute error (Non-holiday MAE) represents the mean absolute error for non-holiday weeks. The purpose of this metric is to isolate the model’s performance during non-holiday weeks, which accounts for most of the dataset. It helps understand whether the model overemphasises holidays at the expense of non-holiday week accuracy.

## Algorithm performances

Hyperparameter tuning was conducted for both models using GridSearchCV. For the Random Forest Regressor, parameters such as ‘n\_estimators’, ‘max\_depth’, and ‘min\_samples\_split’ were optimised. While the Gradient Boosting Regressor was tuned for ‘learning\_rate’, ‘n\_estimators’, and ‘max\_depth’. Both models were trained on a dataset of around 40,000 instances and tested on a separate dataset of around 12,000 instances.

### Random Forest Regressor

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | WMAE | Holiday MAE | Non-holiday MAE | RMSE | R^2 |
| Pre-tuning | 2198 | 2400 | 1449 | 3817 | 0.972 |
| Post-tuning | 2169 | 2346 | 1439 | 3750 | 0.973 |

Table 1, RF performance before and after tuning

These metrics suggest that the untuned Random Forest Regressor (shown in Table 1 and Figure 7) performs exceptionally well, with a high R^2 value of 0.972, indicating the model explains almost all variance in the dataset. The Holiday MAE is 2400, which is significantly higher than the Non-holiday MAE of 1449, this is expected since sales patterns during holiday periods are harder to predict due to them fluctuating more. The Weighted mean absolute error (WMAE) lies closer to the Holiday MAE, which is accurate given that holidays are weighted five times higher in the evaluation. The RMSE value is also relatively low, suggesting good predictive accuracy is displayed by the model.

After tuning, results are only slightly affected. As shown in Table 1 and Figure 13, the Holiday MAE and Non-holiday MAE decrease slightly, indicating the model deals better with holiday and non-holiday related sales patterns. The WMAE and RMSE values remain very close to pre-tuning levels, suggesting the hyperparameter tuning had minimal impact, this is expected because Random Forests are generally robust at default settings. It is possible that changed could be seen if a wider range of parameters was explored in the GridSearchCV.

### Gradient Boosting Regressor

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | WMAE | Holiday MAE | Non-holiday MAE | RMSE | R^2 |
| Pre-tuning | 9116 | 7763 | 6847 | 11474 | 0.748 |
| Post-tuning | 2273 | 2512 | 1486 | 4164 | 0.967 |

Table 2, GBR performance before and after tuning

The untuned Gradient Boosting Regressor (shown in Table 2 and Figure 8) achieves a relatively low R^2 values, it displays higher errors than the Random Forest, because its RMSE, WMAE, and other MAE values are significantly higher. Boosting algorithms require careful parameter initialisation, so this is expected. The R^2 value of 0.748 suggests the model explains less variance in the data. The Holiday MAE is higher than the Non-holiday MAE, but the difference isn’t as big as in the Random Forest, indicating that the untuned Gradient Boosting model struggles equally with both holiday and non-holiday predictions.

After the Gradient Boosting Regressor model was tuned, dramatic improvements across all evaluation metrics were displayed. As shown in Table 2 and Figure14, the R^2 score increases to 0.967, showing how more variability is now explained by the model. Additionally, the RMSE value drops substantially, suggesting a better overall prediction accuracy. The Holiday MAE (2512) and Non-holiday MAE (1486) both show large reductions, though the model still struggles slightly more with holiday predictions, which is expected given the higher complexity of holiday sales patterns. The WMAE also dramatically decreases to a value of 2273, suggesting a major improvement in the model’s overall predictive accuracy.

## Performance comparison

Comparing the two models after they have been tuned, the Random Forest achieves slightly better R^2 score (0.973 vs 0.967), suggesting it explains more variance. The Random Forest also has a lower RMSE value (3750 vs 4164) indicating better predictive accuracy. The Random Forest model achieves a slightly lower WMAE (2169 vs 2273) indicating the model’s predictions are more accurate overall. However, the Gradient Boosting model shows a much larger improvement after being tuned, with much larger shifts in the evaluation metrics. The final predictions for the RF are shown in Figure 11, and the final predictions for the GBR are shown in Figure 12

Both models show higher errors during holiday periods compared to non-holiday periods, this is expected due to greater variability in holiday sales. The Gradient Boosting model appears to close the gap between holiday and non-holiday performance more effectively after tuning.

## Feature importance

A screenshot of a computer screen

Description automatically generated

Figure 1, RF feature importance

A screenshot of a computer code

Description automatically generated

Figure 2, GBR feature importance

Figure 1 shows the feature importance for the Random Forest Regressor (RF), and Figure 2 shows the feature importance for the Gradient Boosting Regressor (GBR).

In both models, ‘Dept’ is the most important feature (62.8 % in RF and 63.3% in GBR) indicating that the department plays a crucial role in determining weekly sales. High levels of importance are also assigned to features like ‘Size’ and ‘Store’. Holiday related features such as ‘Is\_Thanksgiving’ (1.8% in both models) also have a noticeable impact, likely due to significant sales spikes during holidays. Less important features such as ‘Year’, ‘Is\_LaborDay’, ‘Is\_Superbowl’ suggesting minimal influence on weekly sales.

The RF model handles a wider range of features effectively, assigning small but meaningful importance to almost all variables. This contributes to its robustness in handling data.

The GBR model focuses more on impactful features (‘Dept’, ‘Size’, ‘CPI’), which may explain its significant improving in performance after being tuned. However, it appears to remove less influential features more aggressively.

# Outputs

A screenshot of a computer screen

Description automatically generated

Figure 3, unprocessed training dataset

A screenshot of a computer screen

Description automatically generated

Figure 4, training dataset after being pre processed



Figure 5, RF initial predicitions

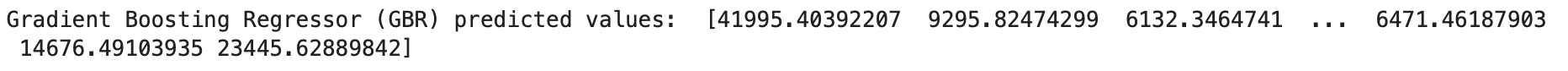


Figure 6, GBR initial predictions

A black text with black text

Description automatically generated

Figure 7, RF performance before tuning

A black text on a white background

Description automatically generated

Figure 8, GBR performance before tuning

A screenshot of a computer code

Description automatically generated

Figure 9, RF grid search

A screenshot of a computer code

Description automatically generated

Figure 10, GBR grid search

A screenshot of a computer

Description automatically generated

Figure 11, RF final predictions

A screenshot of a computer

Description automatically generated

Figure 12, GBR final predictions

A black numbers on a white background

Description automatically generated

Figure 13, RF performance after tuning

A number on a white background

Description automatically generated

Figure 14, GBR performance after tuning

# Summary

The purpose of this project was to develop machine learning models to predict weekly sales across various stores and departments for Walmart, placing emphasis on accurately forecasting holiday sales.

The task required building accurate predictive models and evaluating them, specifically Random Forest Regressor and Gradient Boosting Regressor, whilst ensuring holiday weeks were weighted five times higher in the evaluations. For the models to be accurately trained on the dataset, the data was pre-processed, and new features were added to improve the predictive capabilities of the models. Key metrics such as Weighted Mean Absolute Error, Holiday Mean Absolute Error, Non-holiday Mean Absolute Error Root Mean Squared Error, and R^2 were used to assess the model performance.

Both models were trained with their initial configurations, they were then tested and evaluated, allowing model’s performance to be reported at different parameter settings. Grid search was then employed to systematically explore hyperparameters and identify the best configuration for each model. Both models were then trained using these configurations.

To ensure the models met the project requirements, several evaluation tests were performed on the predictions:

* Comparison of metrics before and after tuning to assess the impact of hyperparameter optimisation.
* Holiday-specific evaluation by calculating separate MAEs for holiday and non-holiday periods to establish to models sensitivity to the variability in holiday sales.
* Cross-validation to ensure robustness of the model performance across different data splits

After the models had been tuned, their evaluation metrics displayed high accuracy, WMAE was prioritised to capture the impact of holiday sales more accurately. The models’ WMAE’s sat at 2169 (RF) and 2273 (GBR), their RMSE’s at 3750 (RF) and 4164 (GBR), and their R^2’s at 0.973 (RF) and 0.967 (GBR). These evaluations confirmed the models’ ability to predict sales effectively during volatile holiday periods, meeting the primary goals of the project.

Several challenges were faced across the implementation and evaluation phases of this project, particularly around feature engineering, holiday weighting, and hyperparameter tuning. There was a significant issue with correctly implementing the holiday weighting in the WMAE metric. Debugging was required to ensure holiday weeks were weighted five times higher in error calculations. Another problem encountered was the handling of missing values in the dataset, this affected both models’ predictive accuracy. This problem was solved through calculating the mean of each column and filling in missing values with the value calculated. Additionally, the hyperparameter tuning for both models were very computationally intense due to a large parameter grid, this was solved through reducing the size of the parameter grid. The disadvantage to this is the decrease in accuracy of the models, because less parameters are explored.

Despite the overall success of the project, few challenges remain unsolved. Accurately predicting holiday sales continues to pose a challenge due to the high variability during these periods. Also, the minimal improvement in the Random Forest performance after being tuned suggests that the hyperparameter grid may not have been broad enough. For example, the Random Forests WMAE improved by 29, whilst the Gradient Boosting model’s WMAE improved by 6,843.

While both models perform well after tuning, the Random Forest Regressor (RF) consistently outperforms the Gradient Boosting Regressor across all key metrics, suggesting that the RF model is better suited for accurately forecasting sales during both holiday and non-holiday periods.

Overall, the project successfully delivered models that met the objectives of developing and evaluating models that could predict weekly sales, with focus on holiday periods, assessing their performance against key metrics. Whilst some challenges remain, such as improving holiday predictions further, the models represent a significant step towards accurate sales forecasting and provide a strong foundation for future expansion, like exploring alternative feature engineering, ensemble methods, or additional data sources.

# References

* DUTTA, A, 2024. *Random Forest Regression in Python* [online]. Available from: <https://www.geeksforgeeks.org/random-forest-regression-in-python/> [Accessed 10 December 2024]
* NIKKI, 2023. *Gradient Boosting in ML* [online]. Available from: <https://www.geeksforgeeks.org/ml-gradient-boosting/> [Accessed 10 December 2024]
* VERMA A, 2023. *GridSearchCV in scikit-learn: A Comprehensive Guide* [online]. Available from: <https://dev.to/anurag629/gridsearchcv-in-scikit-learn-a-comprehensive-guide-2a72> [Accessed 12 December 2024]
* MALATO, G, 2021. *Hyperparameter tuning. Grid search and random search* [online]. Available at: <https://www.yourdatateacher.com/2021/05/19/hyperparameter-tuning-grid-search-and-random-search/> [Accessed 12 December 2024]
* SHAFI, A, 2024. *Random Forest Classification with Scikit-Learn* [online]. Available at: <https://www.datacamp.com/tutorial/random-forests-classifier-python> [Accessed 13 December 2024]
* SAXENA, S, 2024. *A Beginner’s Guide to Random Forest Hyperparameter Tuning* [online]. Available at: <https://www.analyticsvidhya.com/blog/2020/03/beginners-guide-random-forest-hyperparameter-tuning/#:~:text=The%20max_depth%20of%20a%20tree,my%20random%20forest%20to%20grow>. [Accessed 17 December 2024]
* MOHTADI, B F, 2017. *In Depth: Parameter tuning for Random Forest* [online]. Available from: <https://medium.com/all-things-ai/in-depth-parameter-tuning-for-random-forest-d67bb7e920d> [Accessed 12 December 2024]
* MACHINE LEARNING+, n.d. *An Introduction to Gradient Boosting Decision Trees* [online]. Available at: <https://www.machinelearningplus.com/machine-learning/an-introduction-to-gradient-boosting-decision-trees/> [Accessed 21 December 2024]
* HENDRICKS, R, n.d. *Can Cross-validation Be Used for Hyperparameter Tuning?* [online]. Available at: <https://www.deepchecks.com/question/can-cross-validation-be-used-for-hyperparameter-tuning/#:~:text=Cross%2Dvalidation%20is%20used%20when,provide%20an%20overall%20performance%20score>. [Accessed 21 December 2024]
* C3.AI, n.d. *What is Mean Absolute Error (MAE)?* [online]. Available at: <https://c3.ai/glossary/data-science/mean-absolute-error/> [Accessed 27 December 2024]
* STATISTICS HOW TO, n.d. *RMSE: Root Mean Squared Error* [online]. Available at: <https://www.statisticshowto.com/probability-and-statistics/regression-analysis/rmse-root-mean-square-error/> [Accessed 28 December 2024]
* FERNANDO, J, 2024. *R-Squared: Definition, Calculation, and Interpretation* [online]. Available at: <https://www.investopedia.com/terms/r/r-squared.asp> [Accessed 30 December 2024]