Graded Assignment 2 - Forecasting using Regression and Ensembles

Task 1

The objective of this project is to develop a robust forecasting system for daily demand utilizing regression models. Leveraging 2022 data, I will implement a sliding window technique to create and train various predictive models, including decision tree regressors, a custom random forest, and an ensemble of multilayer perceptron regressors. The model demonstrating the best performance will be selected to forecast daily demand for the remainder of 2023.

Task 2

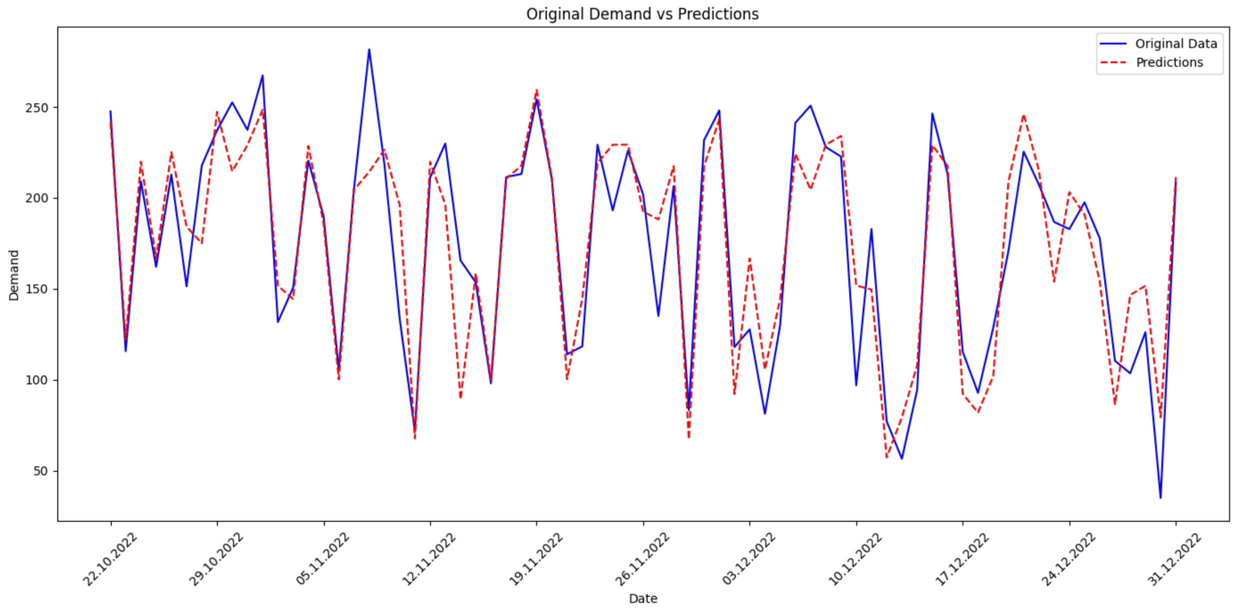
The goal of Task 2 was to implement a decision tree regressor to forecast demand effectively.

The process began with data preparation, where I read the data_2022.csv file and applied a sliding window approach. To facilitate this, I created a function named load_and_prepare_data, which reads the demand data and transforms it into a regression-ready dataset. The sliding window approach involved setting a window size of 10, meaning each prediction was based on the demand from the previous 10 days.

Next, I implemented the train_decision_tree function. In this step, I initialized the regression tree model using the DecisionTreeRegressor from scikit-learn. To ensure reproducibility, I set a fixed random state (seed value 0), which guarantees consistent results each time the model is trained.

To evaluate the model's performance, I split the data into training and testing sets using scikit-learn's train_test_split function. Specifically, 80% of the data was allocated for training and 20% for testing. This split ensures the model is trained on most of the data while maintaining a separate set to validate its generalization capabilities. The decision tree was then trained on the training dataset.

Finally, I used the trained model to make predictions on the test set. The predictions were obtained by passing the test set inputs through the decision tree model. To visualize the model's performance, I plotted the actual demand values alongside the predictions. The results demonstrated that the decision tree captured the general trends in the data, highlighting its potential for forecasting daily demand.



Task 3

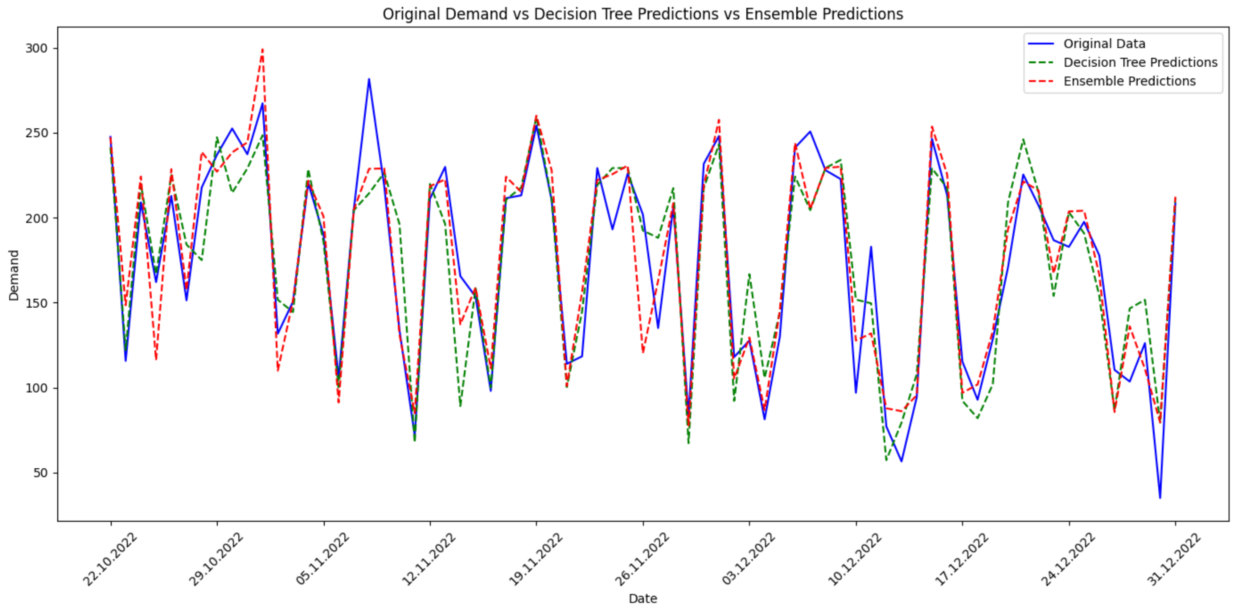
In Task 3, I implemented a custom random forest to enhance the accuracy of demand forecasting for Angstrom Fruitcake Co. Below, I provide a brief explanation of the steps involved.

I created an ensemble consisting of 100 trees, each initialized with randomly generated hyperparameters. To build this ensemble, I designed a function called create_ensemble, which initializes an empty list to store individual decision trees. For each of the specified n_estimators, a bootstrapped sample of the training dataset was generated using the bootstrap_sample function.

Bootstrapping was employed to train each tree, providing them with slightly different subsets of the data. This approach ensures that the ensemble benefits from diverse perspectives of the training data, leading to a more robust and accurate model. The bootstrap_sample function was instrumental in creating these varied subsets.

To combine predictions from all the trees, I implemented the predict_ensemble function. This function aggregates predictions by averaging the outputs of all individual trees. Averaging reduces prediction variance, smoothens the output, and minimizes the risk of overfitting, resulting in a more stable and accurate forecasting model.

In the final step, I visualized and compared the predictions generated by the random forest ensemble, the single decision tree from Task 2, and the original demand data. The results revealed that the ensemble's predictions were significantly smoother and more closely aligned with the true demand curve compared to the single decision tree. This suggests that the ensemble effectively captures the general trends in the data and is likely to generalize better to unseen scenarios.



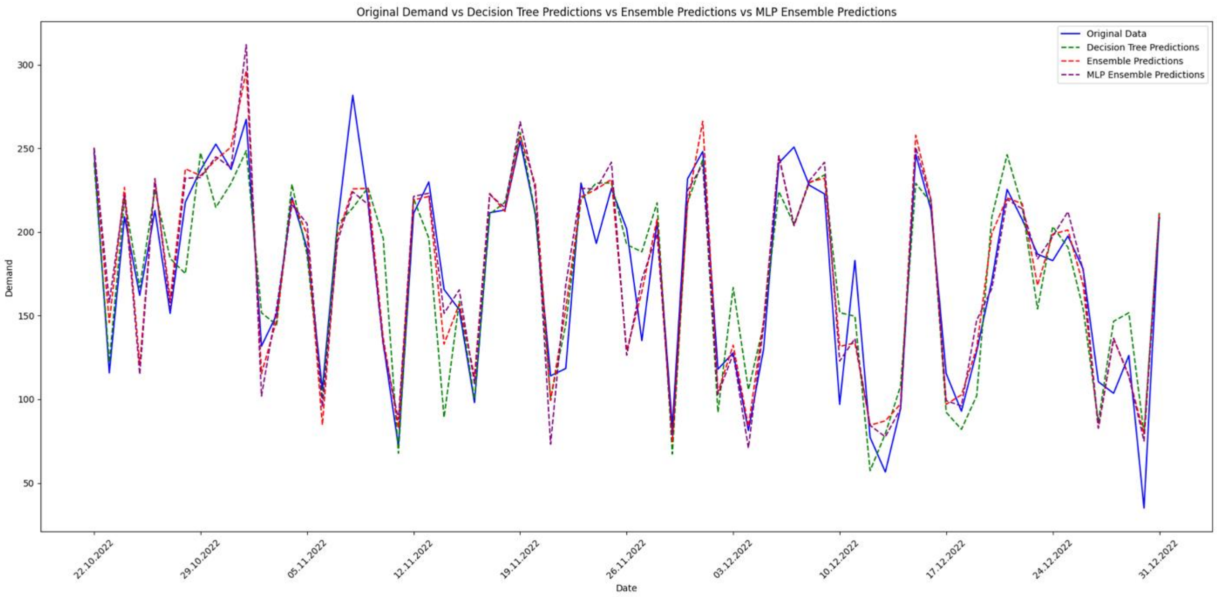
Task 4

To further enhance the forecasting model, I constructed an ensemble of MLPRegressor instances using the create_mlp_ensemble function. Each MLP was configured with a single hidden layer containing 100 neurons to maintain consistency across models. To ensure reproducibility, I set random_state=0, following the same approach used throughout the assignment. The ensemble consisted of 100 estimators.

For training, I employed a bagging approach. Each MLP was trained on a unique bootstrap sample of the data, introducing diversity and randomness into the ensemble. This step added computational complexity but was crucial for creating a robust and varied model ensemble.

For prediction, I implemented the predict_with_mlp_ensemble function, which aggregates the forecasts of all MLPs by averaging their outputs. This approach effectively reduces noise and minimizes individual model errors, resulting in a more stable and reliable prediction.

When comparing all the models, the MLP ensemble appears well-balanced, capturing the complexity of the data without overfitting. However, certain patterns and predictions across all models remain relatively similar in several areas, suggesting a shared understanding of the underlying trends in the data.

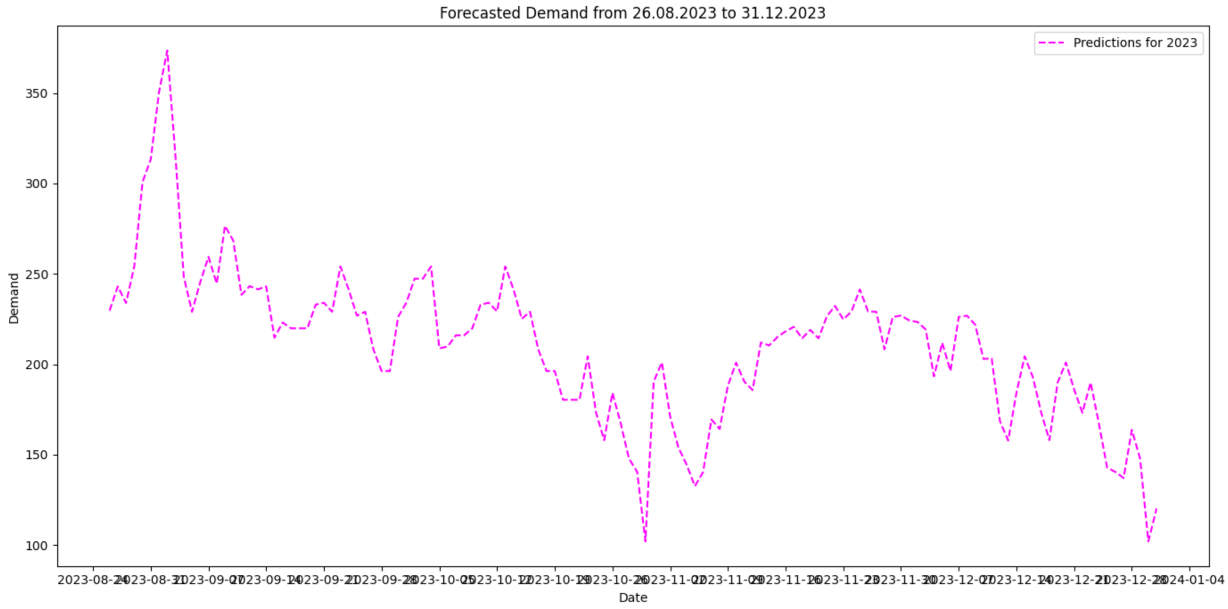


Task 5

In Task 5, I began by loading the data_2023.csv file and preparing the data up to August 25, 2023. To ensure consistency with the previous tasks, I utilized the familiar sliding window technique with a window size of 10. This approach created a suitable foundation for generating predictions from August 26, 2023, to the end of the year.

For the forecasting, I selected the decision tree model trained on the 2022 data, as it was identified as the best-performing model earlier. I implemented a predict_demand function to generate daily demand predictions for the specified range, from August 26 to December 31, 2023. However, I was surprised to find that the decision tree model from Task 2 outperformed the more complex ensemble models from subsequent tasks. This unexpected result led me to suspect a potential issue in my implementation for Task 5, though I was unable to identify and resolve the problem within the available time.

Finally, I visualized the predictions for the given date range, plotting the forecasted demand values. Despite the uncertainty regarding the model selection, the plotted predictions provided an insight into the expected trends for the remainder of the year.


Forecasted Demand from 26.08.2023 to 31.12.2023
Predictions for 2023
Demand
Date