**Detailed Report on Forecasting Approach**

***1. Introduction***

The provided Python code aims to forecast sales data using time series forecasting techniques. The code utilizes various Python libraries and techniques to preprocess the data, detect outliers, visualize trends, perform seasonal decomposition, and conduct forecasting using neural network models. This detailed report provides an in-depth analysis of each step involved in the process.

***2. Libraries Used***

* ***neuralforecast***: This library offers a range of tools for building neural network-based forecasting models, including Long Short-Term Memory (LSTM) and NHITS models.
* ***pandas***: It provides data structures and operations for manipulating numerical tables and time series data, essential for data preprocessing and manipulation.
* ***numpy:*** This library is used for numerical computing with arrays and matrices, enabling efficient mathematical operations on data.
* ***matplotlib:*** Employed for data visualization, particularly for creating plots and graphs to visualize trends, outliers, and forecasted values.
* ***statsmodels:*** It provides classes and functions for statistical data analysis, including seasonal decomposition, which is utilized in the code.

***3. Data Preprocessing***

* a. The code starts by installing the **neuralforecast** library using pip. This step ensures that the necessary forecasting tools are available for subsequent use.
* b. Sales data is read from an Excel file into a pandas DataFrame (df). The data is likely structured with columns such as 'on\_date', 'SKU\_m', 'product category', and 'quantity sold'.
* c. The 'on\_date' column in the DataFrame is converted to datetime format to facilitate time series analysis.
* d. The DataFrame is sorted based on the date column to ensure the data is in chronological order, which is crucial for time series analysis.
* e. Filtering processes are applied to select relevant data:
  + i. SKUs with more than 350 entries are selected, ensuring that only products with a significant sales history are considered for analysis.
  + ii. Product categories with more than 3500 entries are selected, indicating a substantial volume of sales data for those categories.
* f. Outliers are detected using the z-score method, which measures the deviation of each data point from the mean in terms of standard deviations. A threshold of 3 standard deviations is used to identify outliers, which are then flagged for further analysis.
* g. Outliers are visualized using scatter plots, allowing for a clear identification of data points that deviate significantly from the overall trend.
* h. The overall sales trend is visualized using a line plot, providing insights into the general trajectory of sales over time.

***4. Seasonal Decomposition***

* a. Seasonal decomposition is performed on the selected product categories to isolate seasonal, trend, and residual components.

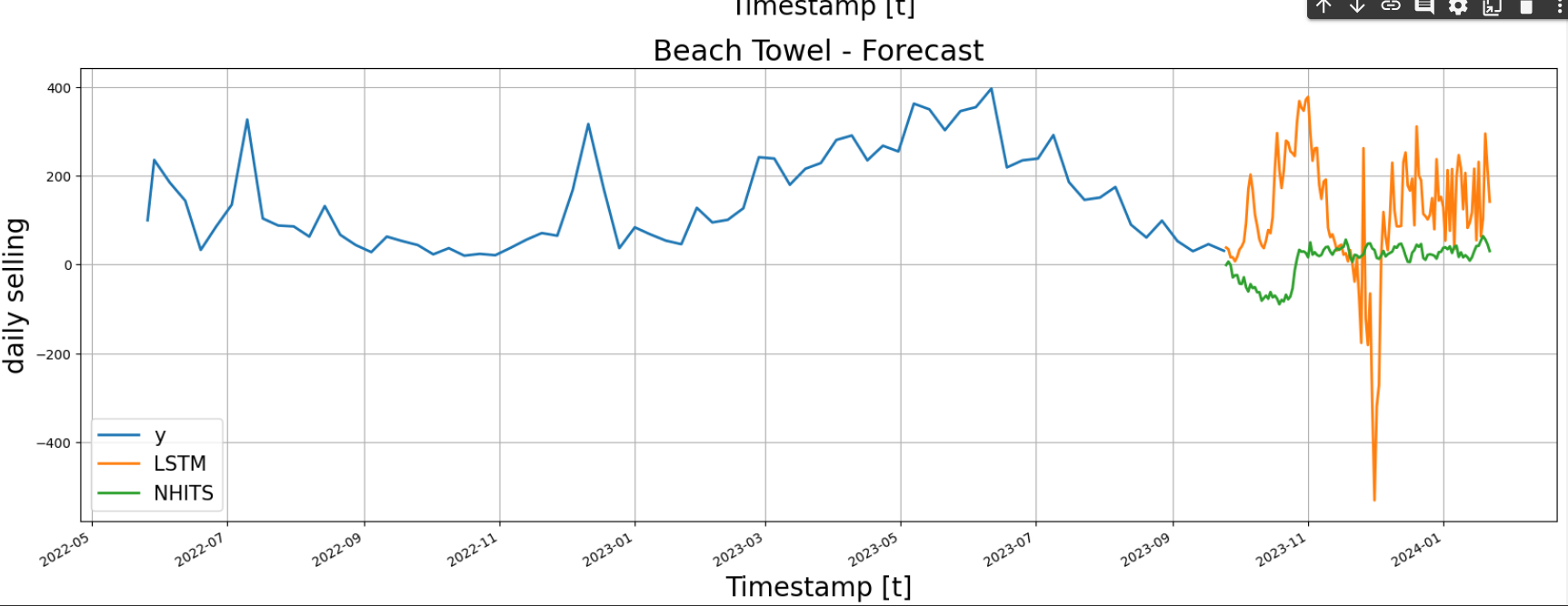
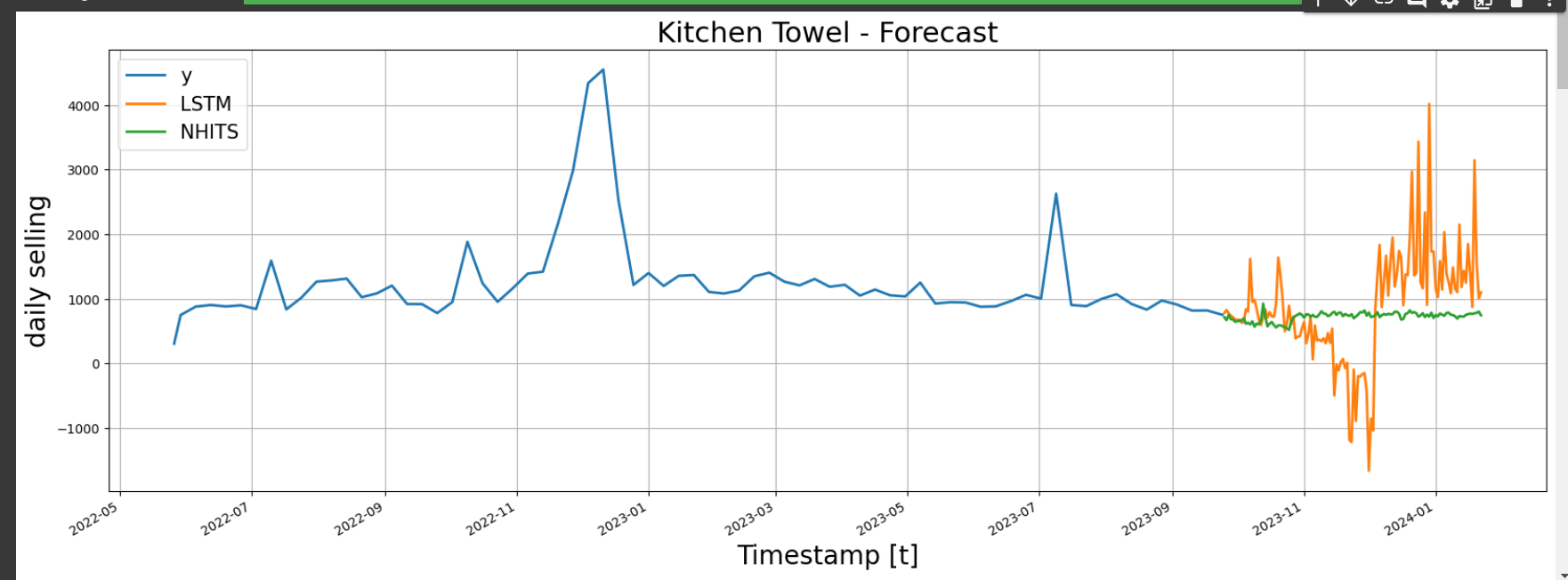
b. The seasonal\_decompose function from statsmodels is utilized for this purpose. This function decomposes the time series into seasonal, trend, and residual components using either additive or multiplicative models.

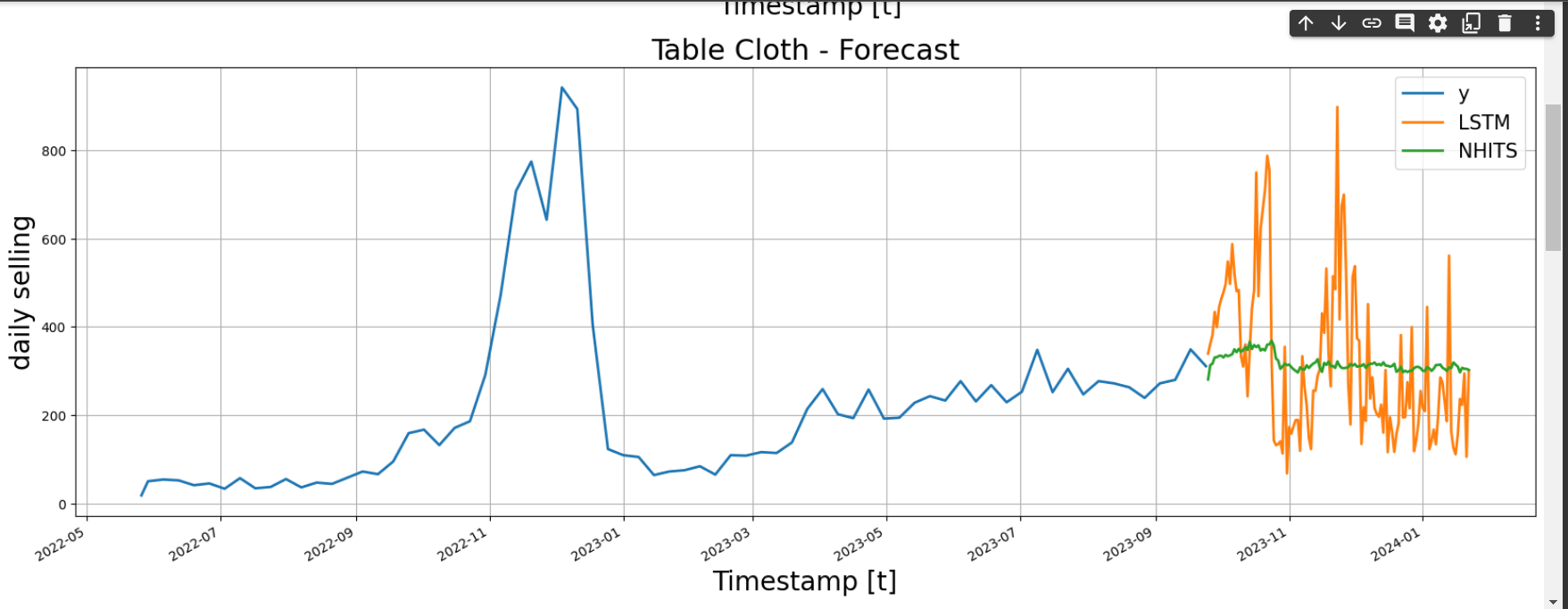
* c. Seasonal components for each product category are visualized using line plots, allowing for the examination of seasonal patterns and fluctuations in sales over time.

***5. Forecasting***

* a. Two types of neural network models, namely LSTM and NHITS, are employed for forecasting sales data.
* b. Data is prepared for each product category by transforming it into a suitable format for training the models.
* c. For each product category, both LSTM and NHITS models are trained using the training data and then used to forecast future sales.
* d. The NeuralForecast class from the neuralforecast library is utilized for training and predicting. This class provides a convenient interface for training neural network models and generating forecasts.
* e. The forecasted values are plotted along with the actual sales data for visualization.

Outputs:





More outputs are in the ***mensa\_assignment\_solution.ipynb*** file.

***6. Conclusion***

The provided code demonstrates a comprehensive approach to forecasting sales data using time series analysis and neural network models. By leveraging Python libraries such as neural forecast, pandas, and matplotlib, the code achieves efficient preprocessing, visualization, and forecasting of sales data, providing valuable insights for decision-making and planning. Further optimization and fine-tuning of the forecasting models can be explored to improve accuracy and reliability.