SALES FORECASTING

TECHNICAL REPORT

Prabhashi Mahawatta - 009





1.Introduction

1.1 Statement of purpose.

The purpose of this report is to create a strategy for accurately estimating sales that focuses on the top 25 moving goods in each department. Accurate sales forecasting optimizes inventory levels, reduces overstocking and understocking, and ensures product availability. This enhances customer satisfaction, increases operational efficiency, and boosts revenue by capturing more sales opportunities. By utilizing data-driven insights and advanced forecasting techniques, the company aims to drive business growth and competitiveness, ensuring the right products are in stock and meeting customer demand.

2.Methodology

This course work focuses on gathering necessary data and the data preparation procedure included cleaning and correcting missing values, creating features and target variables, and compiling all the data into a master table. In order to forecast sales, I used a variety of machine learning techniques, such as ARIMA (Time Series), Random Forest Regressor, XGBoost Regressor, LGBM Regressor, and Lasso Regression. A held-out test set helped me select the top model. Afterwards, grid search was used to improve the model's hyperparameters.

I employed the mean squared error (MSE) statistic to assess the model's precision. The average squared difference between the projected and actual sales numbers is what the MSE calculates.

3.Implementation

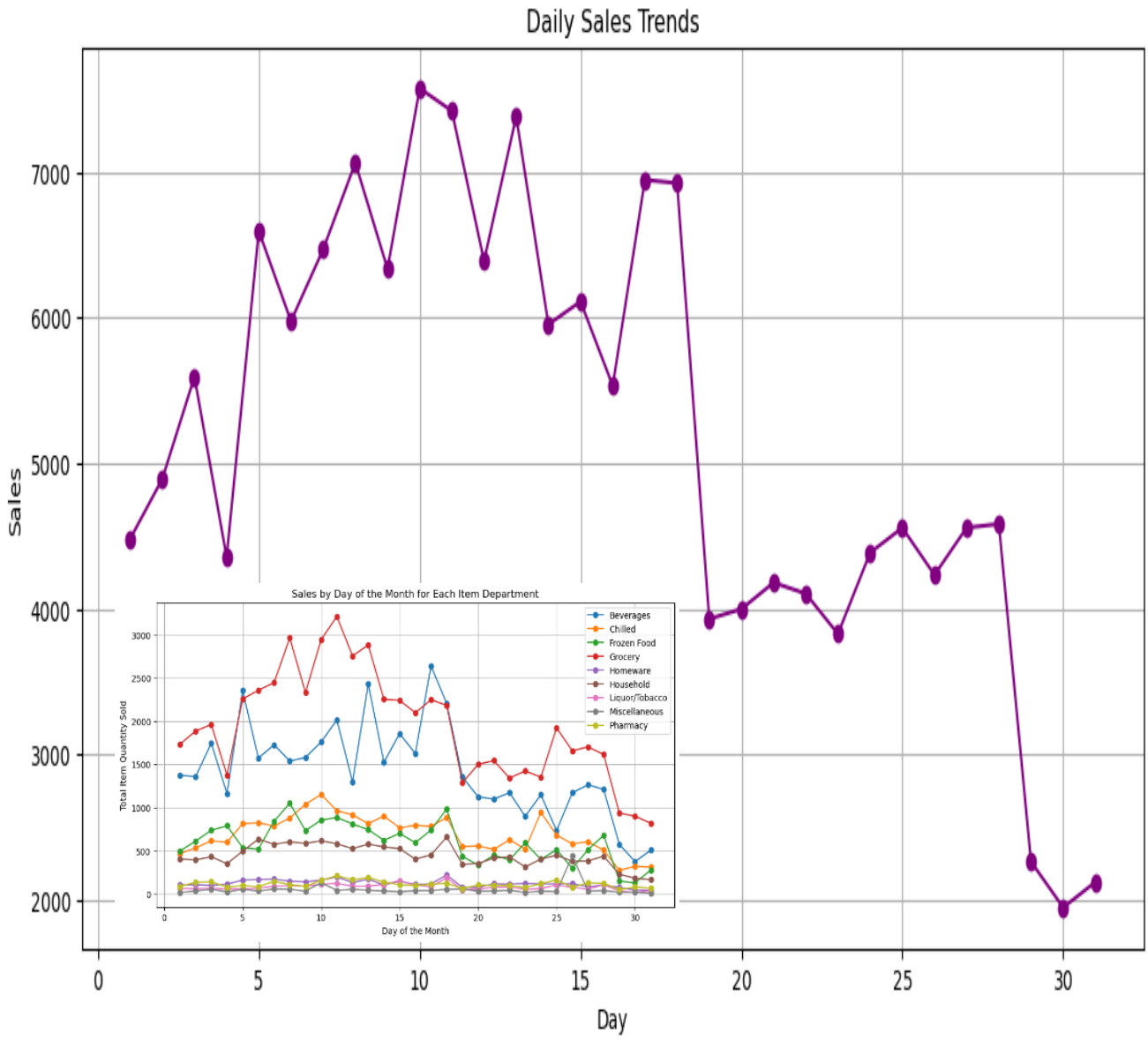
3.1 Data preparation

3.1.1 Data Sources

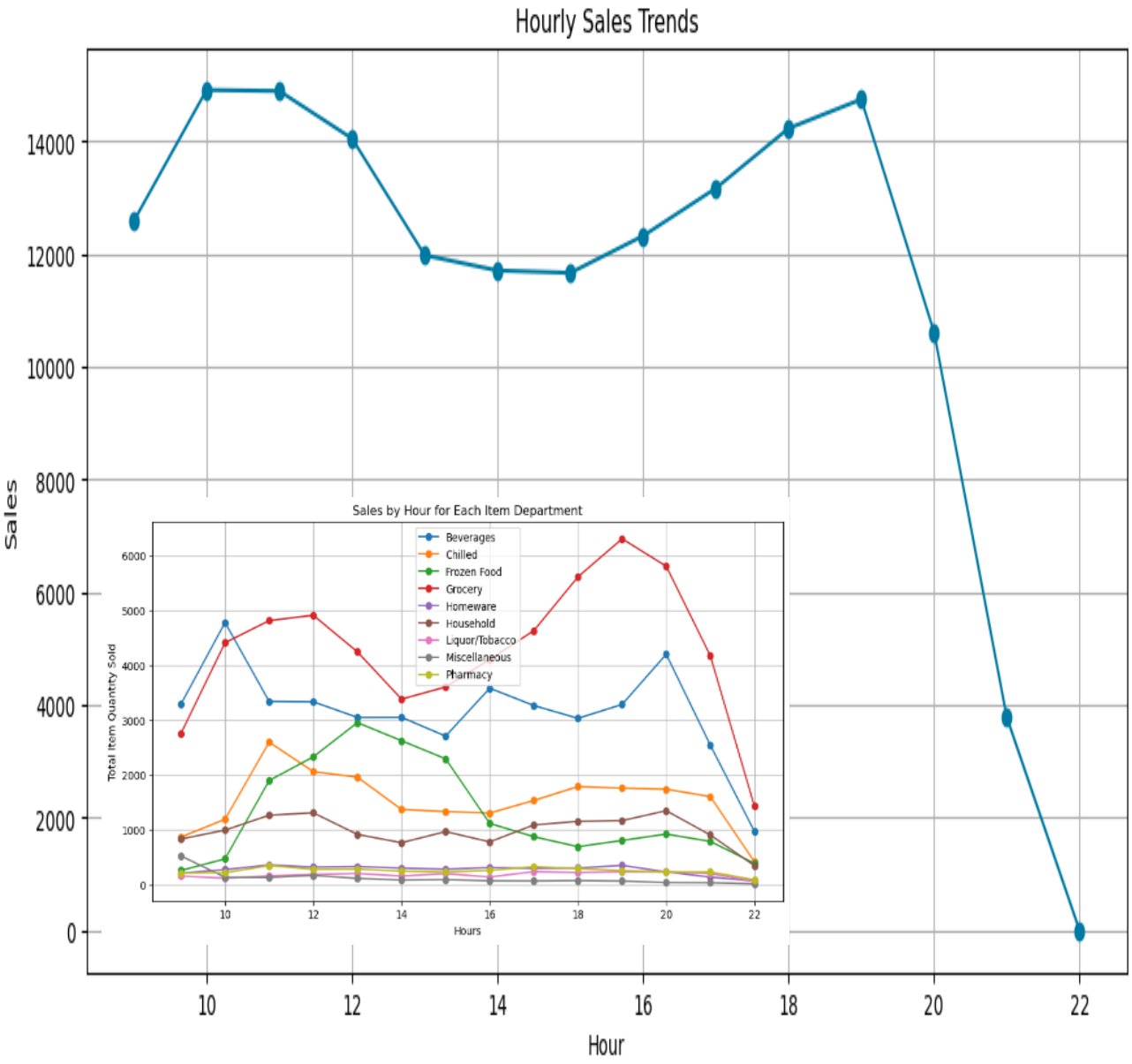
We have given item.csv and transformation.csv for Sales Forecasting.

item_info.csv : Contains item related the information (Columns: item_code, item_sub_segment, item_code , item_segment , item_sub_segment, item_category, item_sub_department and item_department)segment of an item
transactions_info: Contains transactions related information (Columns: item_code, invoice_num, invoice_time and item_qty)

Descriptive Analysis.



The graph shows sales peak at the start and end of the month, declining in the middle due to increased spending after payment, indicating department popularity. And the graph illustrates the popularity of several departments relative to one another.



The graph shows sales peak in the morning and evening, with a peak around noon due to lunch breaks, and peaking at 6:00 PM due to shopping after work. This pattern is observed across all departments, likely due to people shopping at the beginning or end of the day.

Test Hypothesis

Null hypothesis (H0): There is no correlation between the historical sales data and the future sales of the top 25 fast moving items for each department in Dry at a supermarket.

Alternative hypothesis (H1): There is a correlation between the historical sales data and the future sales of the top 25 fast moving items for each department in Dry at a supermarket.

Here are some additional hypotheses that might be investigated:

- The historical sales data of an item is a good predictor of its future sales.
- The sales of other products in the same category or department have an impact on the sales of a particular item.
- External aspects, such the day of the week, season (new year), holidays or the weather, have an impact on an item's sales.

3.1.2 Features Construction

Different factors, such as time-related features, sales-related features, and item-related features would influence sales demand; I identify and create the most representative features.

Time-Related Features

- same_day_hour_before: This feature calculates the amount of an item sold in the previous hour on the same day.
- previous_day: This function determines the overall amount of a product sold the day before.
- previous_day_avg_sales: This function determines the typical amount of a product sold the day before.
- d2_sales: This function figures out how many units of an item were sold overall two days ago.
- sales_previous_week: This function determines the overall amount of a product sold the previous week.
- day_of_week: This function determines the transaction's day of the week.
- day_of_week_mean_sales: This function determines the typical amount of a product sold on each day of the week.
- rolling_mean_3h, 6h,13h: This function determines the rolling mean of an item's quantity sold over the last three hours, last six hours, last thirteen hours (previous day). It can be used to spot trends and smooth out data noise.
- rolling_mean_3h,6h,13h: The rolling standard deviation of the quantity of an item sold over the previous three hours, previous six hours, previous thirteen hours (previous day).

- is_holiday: A binary variable in this feature determines if a transaction occurred on a holiday.
- is_weekend: A binary variable in this feature determines if a transaction occurred on a weekend day.
- item_hourly_average: This function figures out how many units of a particular item are typically sold throughout each hour of the day.
- time_of_day: The four categories included in this feature are night, morning, afternoon, and evening.

Sales-Related Features

- sales_trend: This computes the slope of a linear regression line fitted to the total number of goods sold over the preceding seven hours.
- Sales_Volatility: This function determines the standard deviation of an item's seven-hour sales volume.
- weekly_sales_index: This feature calculates the proportion of an item's weekly sales that occur on a given day of the week

Item-Related Features

- weekly_sales_department: This determined by dividing sales data by department and week, and averaging item amounts. This process is repeated for item_department, item_category and item_ sub_segment.

3.1.3 Target Construction

The target variable is next_hour_sales. Calculate the quantity of a particular item that will be sold in the ensuing hour based on the invoice date. The feature is created by combining the desired_structure and outcome data frames on the invoice_date, item_code, and hour columns. Any missing values in the item_qty column are filled in with 0 and item_qty is then shifted by 1 to calculate the next_hour_sales.

3.1.4 Master Table Creation

The code provided creates a master table with all features for machine learning training, using the Master_Table class to create a single data frame from the Fast_Moving_Items, Pre_Processing, Target_Variable, Time_Related_Features, Sales_Related_Features, Item_Related_Features and Primary_Keys classes.

```
class Master_Table(BaseEstimator, TransformerMixin):

    def fit(self, X, y= None):
        return self

    def transform(self,X):

        # Execute the Fast_Moving_Items.ipynb notebook using %run
        %run Fast_Moving_Items.ipynb
        %run Pre_Processing.ipynb
        %run Target_Variable.ipynb
        %run Time_Related_Features.ipynb
        %run Sales_Related_Features.ipynb
        %run Item_Related_Features.ipynb
        %run Primary_Keys.ipynb

        MovingT = Fast_Moving_Items()
        PP = Pre_Processing()
        TV = Target_Variable()
        TR_F = Time_Related_Features()
        Sales_F = Sales_Related_Features()
        Item_F = Item_Related_Features()
        PK= Primary_Keys()

        df = TR_F.fit_transform(TV.fit_transform(PP.fit_transform(MovingT.fit_transform(X))))
        df1 = Item_F.fit_transform(Sales_F.fit_transform(df))

        X = PK.fit_transform(df1)

        return X
```

3.2 Model Development

3.2.1 Choosing the Algorithm

I chose to use the LGBM Regressor algorithm to predict sales because it had the lowest MAPE value among the models I was considered. MAPE (Mean Absolute Percentage Error)

Used Models:

- ARIMA: Statistical model that forecasts time series data by identifying and modeling historical data patterns.
- Random Forest: Ensemble learning model that combines multiple decision tree predictions to learn complex feature-target relationships.
- XGBoost: Ensemble learning model that uses a different decision tree and is more efficient for training than Random Forest Regressors.
- LGBM: Gradient boosting machine that is more efficient and can handle larger datasets than XGBoost Regressors.
- Lasso Regression: Linear regression technique that uses L1 regularization to improve model performance and interpretability.

3.2.2 Hyperparameter Optimization

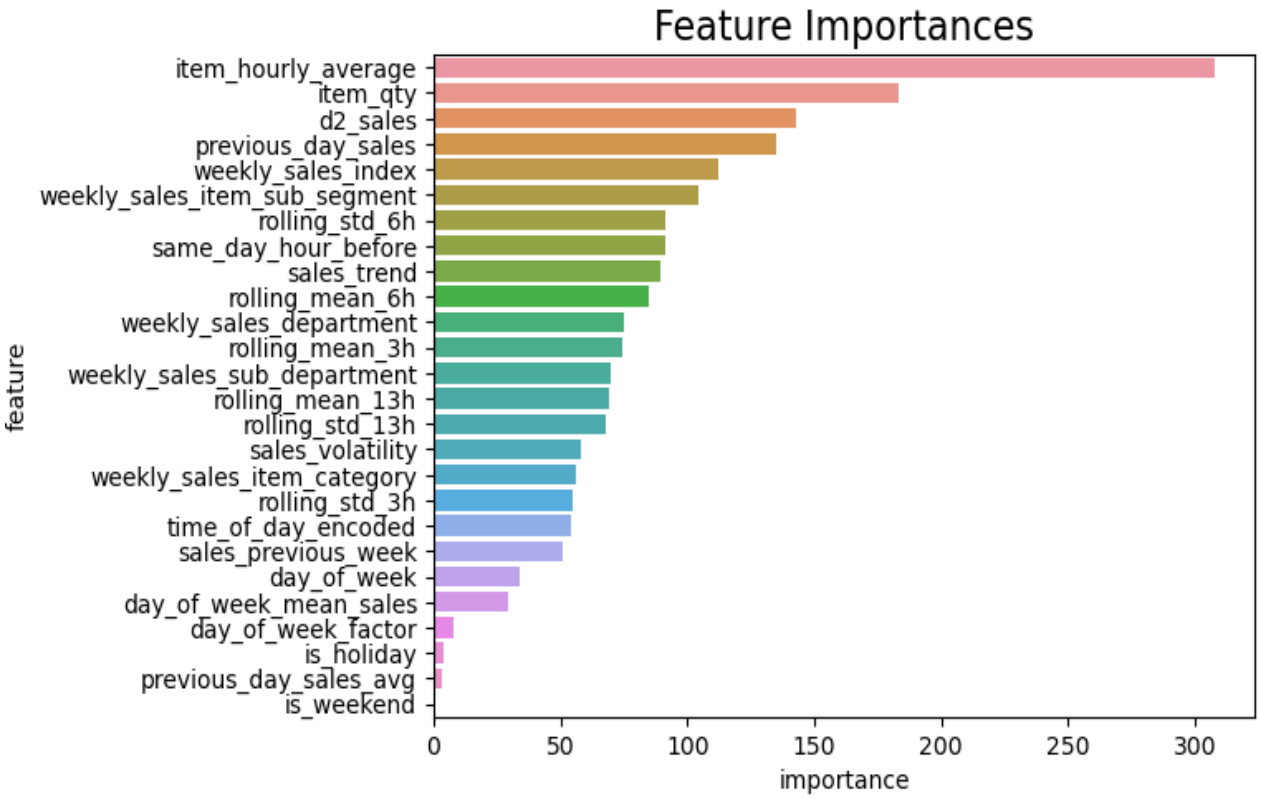
I have selected the LGBMRegressor model. And used a grid search to optimize the hyperparameters and tuned the following hyperparameters:

- n_estimators: The number of trees in the model.
- learning_rate: The learning rate controls how quickly the model learns.
- max_depth: The maximum depth of the trees in the model.
- subsample: The fraction of samples used to build each tree.

```
best_params
```

```
{'learning_rate': 0.1, 'max_depth': 5, 'n_estimators': 100, 'subsample': 0.8}
```

3.2.3 Feature Importance



The feature importance analysis reveals that the most crucial features for predicting sales are as follows:

- item_hourly_average
- item_qty
- d2 sales
- previous_day_sales
- weekly sales index
- weekly_sales_item_sub_segment

Indicating a deeper understanding of overall sales trends, the sales_trend element is more significant than the sales volatility factor.The model's ability to identify key sales predictor features, aiding in improved performance and decision-making in inventory management, staffing, and promotions.

3.2.4 Model Accuracy Measures

To evaluate the model's accuracy, I used the Mean Absolute Percentage Error (MAPE) metric and Mean Absolute Error (MAE).

$$MAPE = \frac{1}{Number\ of\ Predictions} \cdot \Sigma \left[\frac{Actual - Predictions}{Actual} \right] \cdot 100\%$$

$$MAE = \frac{1}{Number\ of\ Predictions} \cdot \Sigma \left[Actual - Predictions \right]$$

3.3 Risks and assumptions

Sales forecasting faces risks from external factors beyond business control, such as economic changes, successful or unsuccessful marketing campaigns, seasonal sales patterns, and competitor activity.

The assumption that the model's exclusive use of dry goods is that these sales are less influenced by outside forces than sales of other kinds of goods. This is due to the fact that dry goods are often more enduring and have a longer shelf life than other goods. Additionally, under forecasting is unnecessary for dry goods.

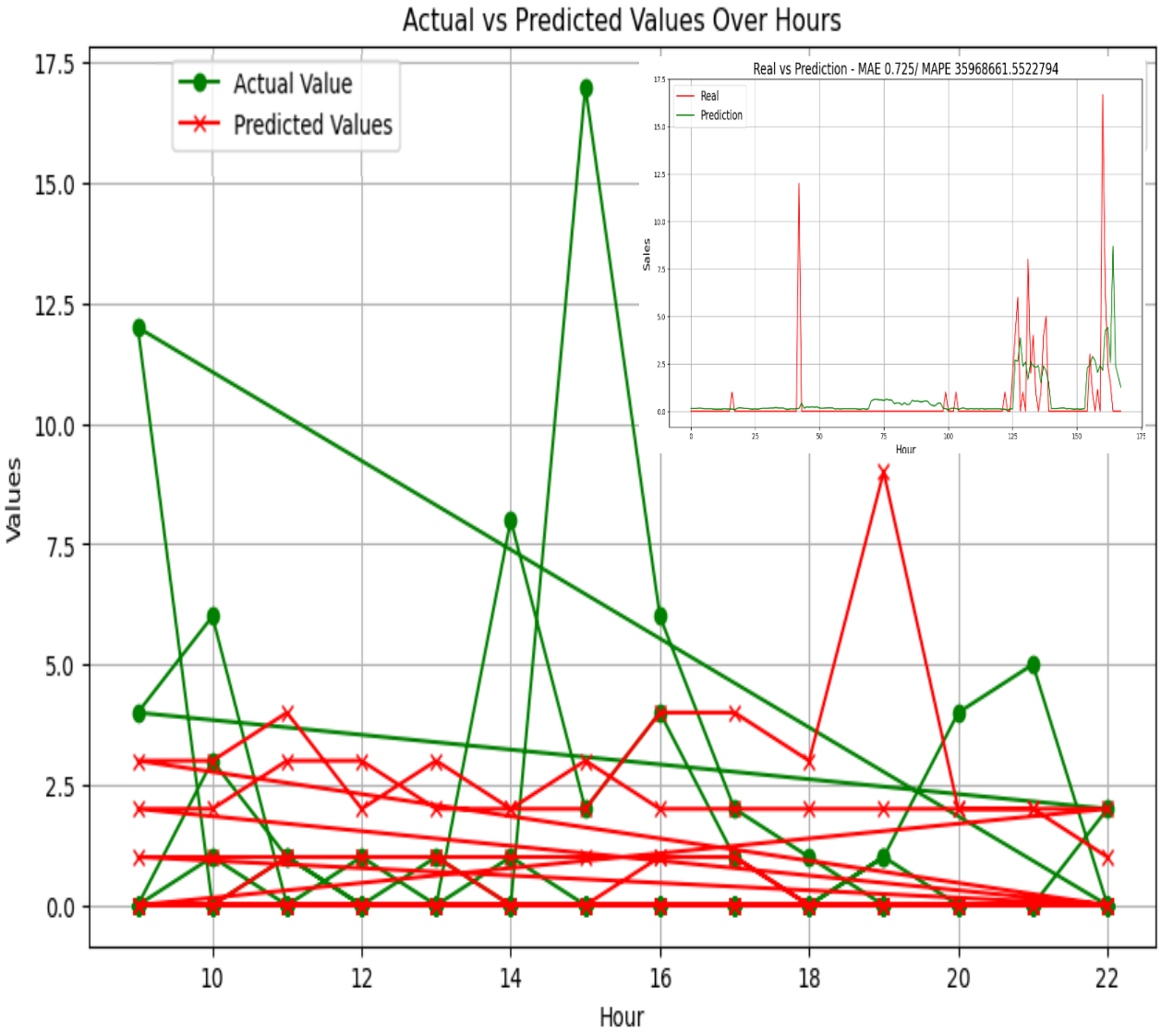
3.4 Pipelines

Two pipelines were produced by me.

Two pipelines: One for the master table and the other for the sales forecasting.

4. Findings and Conclusions

Model Performance: With a Mean Squared Error (MSE) value of 0.725, the model demonstrates reasonable performance. However, it's important to note that a lower MSE value would indicate better model performance.



Feature Significance: The dominance of historical sales data and product information among the most important features indicates that the model relies heavily on these factors for forecasting future sales. This aligns with the understanding that past performance and product-related variables are significant drivers of sales prediction.

Recommendations:

Based on the findings and conclusions, the following recommendations can be made:

- Continued Use of the Model: Continue to use the model for sales predictions, as it demonstrates reasonable performance.
- Performance Monitoring: Regularly monitor the model's performance over time. Assess its accuracy and make necessary adjustments or updates to ensure it continues to perform well.
- Enhanced Data Collection: Consider collecting additional data related to product information, such as promotions and pricing. Incorporating more detailed product-related variables can help further improve the model's accuracy, as it seems to heavily depend on product characteristics for forecasting.

Overall, the model shows promise in predicting sales, but there is room for improvement, especially by incorporating additional relevant data and maintaining a vigilant approach to model performance.