MarketMatch - Sales Forecast Report

Business Framing: Sales Demand Forecasting for Optimized Operations

Sales demand forecasting is essential for increasing operational efficiency and managing overstocking and supply shortages. Accurate predictions allow:

- **Optimized Inventory Management:** Lower storage costs, decrease waste, and maintain product availability.
- **Resource Efficiency:** Reduce operational costs by streamlining production, labor, and logistics.
- **Cash Flow Stability:** Improve capital management by minimizing wasteful cash expenditures or shortages.
- **Strategic Decision Making:** Align pricing, marketing initiatives, and promotions with current demand trends.
- **Enhanced customer satisfaction:** Meet demand quickly to foster trust, loyalty, and long-term connections.
- **Risk Mitigation and Competitiveness**: Respond proactively to market shifts while maintaining agility and service quality.

Effective demand forecasting promotes sustainable operations, waste reduction, and long-term corporate success.

Problem Statement

Rohlik Group aims to forecast warehouse inventory sales for the next 14 days using historical sales data. The task is to predict the sales column for a given ID, constructed from unique_id and date (e.g., id 1226_2024-06-03 from unique_id 1226 and date 2024-06-03).

The challenge involves balancing the trade-off between overstocking and stock shortages, with the following cost-benefit considerations:

- Benefit: 10 per unit for fulfilling demand.
- Cost: 5 per unit for over-estimating sales

Data and Data Processing

The sales forecasting task involves the integration and preparation of three datasets:

1. Sales Train Data: Contains 4,007,419 records of historical sales, prices, and other relevant details.
2. Inventory Data: Includes 5,432 records providing information about product availability.
3. Calendar Data: Consists of 23,016 records with time-related attributes, including holidays and school schedules.

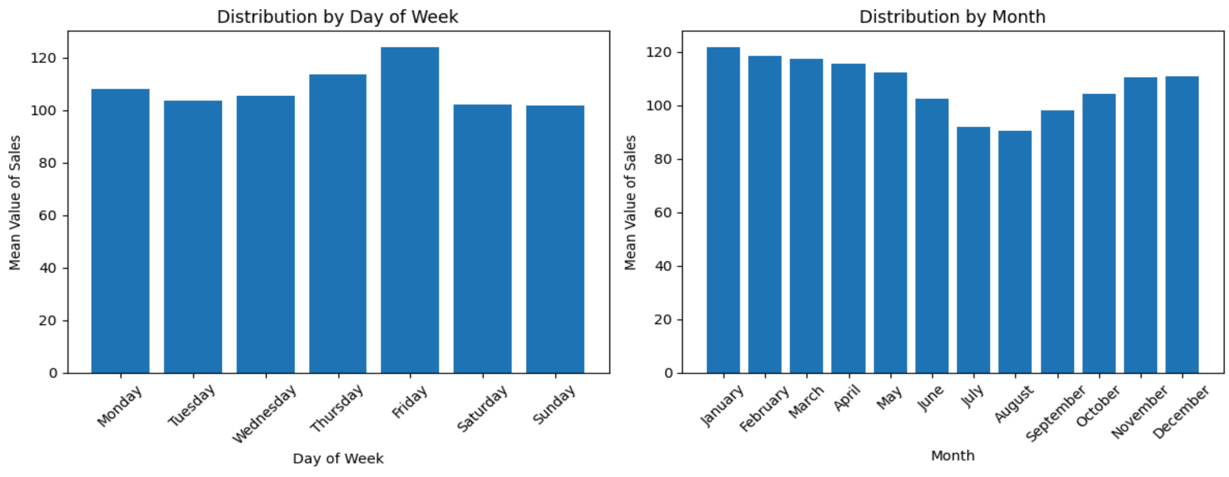
Steps Involved

1. Handling Missing Values
   An initial analysis identified 52 missing values in the sales train dataset. Given their minimal impact on the overall data size, these records were removed to maintain data quality.
2. Feature Engineering
   - Data Integration: The three datasets (sales train, inventory, and calendar) were merged to form a unified dataset for analysis and modeling.
   - Discount Summarization: As multiple discount types can apply simultaneously, only the highest possible discount was considered to reflect its actual impact on sales.

3. Data Cleaning
   To focus on relevant variables for prediction, non-essential features were excluded. The cleaned dataset retained attributes such as:
   - Product and warehouse identifiers
   - Sales and price details
   - Discounts and relevant calendar features

4. Encoding Categorical Variables
   - Product categories and warehouse locations were converted into numerical representations through one-hot encoding, ensuring the dataset was suitable for machine learning models. This step also reduced memory usage by excluding redundant or irrelevant features.

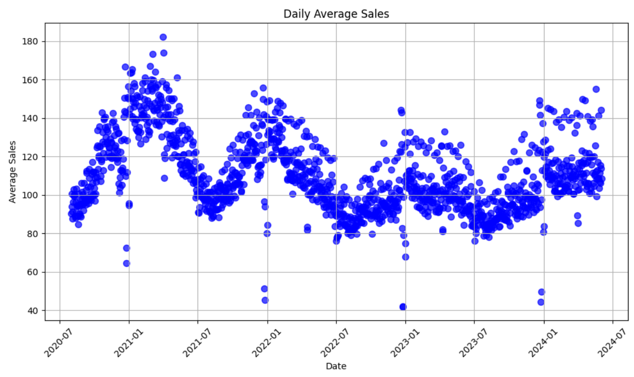
5. Time-Based Feature Creation
   - Extracted time-related information, such as weekly and monthly trends, to capture seasonality and time-based sales patterns.



   - Defined the chronological order of the dataset to ensure sequential predictions, a critical step for accurate time-series forecasting.

6. Dataset Reduction
   Given the large dataset size, a subset of 50 unique_ids (representing products) was selected. These IDs were chosen to ensure complete timeline coverage, maintaining the integrity of time-based analysis while reducing computational overhead.

7. Exploratory Analysis
   - Average sales patterns were analyzed over time to identify trends, seasonal variations, and periods of higher or lower demand.
   - This analysis provided key insights to guide the forecasting approach.



Models and Methods

· ***Baseline Model-Auto ETS Model***

To address the forecasting challenge, the AutoETS model was used to predict sales for a 14-day horizon. This model was chosen for its automated feature selection, which optimizes the components of Error (E), Trend (T), and Seasonality (S). The focus was to capture weekly seasonality in the dataset, as the sales data exhibits clear periodic patterns. Additionally, the Naive Forecast was used as a baseline to provide a straightforward comparison for model performance.

AutoETS was an appropriate choice for the following reasons:

1. Automated Component Selection: AutoETS identifies the best combination of error, trend, and seasonality, ensuring an optimal fit for the data.
2. Seasonality Handling: Given the 7-day weekly sales cycle, AutoETS efficiently models recurring patterns.
3. Scalability: AutoETS can be applied across multiple warehouses, maintaining consistency in forecasting performance.

A Naive Forecast was used as a baseline model, where sales for the next 14 days were predicted based on the most recent 7-day sales data. This simple method allows us to compare the performance of the more advanced AutoETS model.

To ensure robust and unbiased evaluation, I employed Leave-One-Out Cross-Validation (LOOCV), focusing on the last 56 days of the dataset for the cross-validation procedure. For each warehouse, the following steps were taken:

1. Training: The model was trained on historical sales data leading up to the current time point.
2. Testing: Predictions for the next 14 days were made, and the error between predicted and actual sales was calculated.
3. Comparison to Naive Forecast: The performance of AutoETS was then compared to that of the Naive Forecast for the same test period.

The models were evaluated using Mean Absolute Error (MAE), which considers the weighted importance of different warehouses in the forecast, providing a more accurate reflection of the prediction quality.

· ***Neural Prophet***

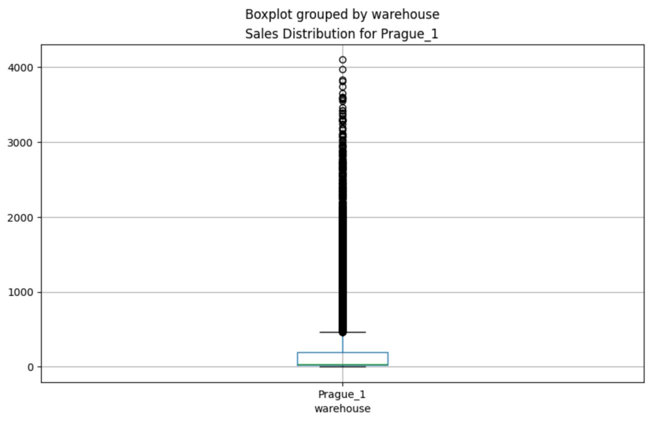
Neural Prophet was chosen for its ability to efficiently handle time-series forecasting with recurring seasonality and external regressors. It integrates weekly and yearly seasonality and supports external variables like holidays, shops_closed, and school holidays, which are essential for accurate sales demand forecasting. Compared to deep learning models, Neural Prophet requires minimal tuning while remaining scalable and interpretable.

Code Workflow

Columns were renamed ('sales' → 'y', 'date' → 'ds') to match Neural Prophet's format. Forecast Horizon setting to 14 days. The model used 7-day lags to capture recent trends. Weekly and yearly seasonality were enabled to reflect periodic trends. Daily seasonality was disabled to reduce noise. Configured with 10 changepoints and a range of 0.8 to identify trend shifts. MASE was calculated, comparing predictions against actual values and a Naive Forecast baseline.

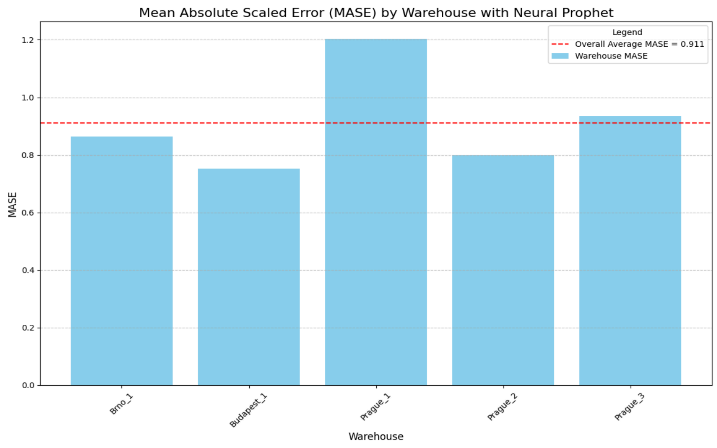
Results Analysis

The model's performance was evaluated using MASE for each warehouse. The results indicated that while most warehouses achieved satisfactory forecasts (e.g., Budapest_1 with 0.75), Prague_1 exhibited higher errors (1.20), suggesting further optimization opportunities. The boxplot below highlights significant outliers and skewed sales data in Prague_1, which likely contributed to its higher MASE value.



To address outlier effects in the Prague_1 warehouse, I implemented a 5% cap on the sales data and calculated the 95th percentile (quantile(0.95)) for sales (y) in Prague_1, identifying 1402.21 as the cap value. The result for this model changing in Prague_1 shows significant improvement, with MASE dropping from 1.20 to 0.55. Prague_2's MASE increased slightly (from 0.80 to 1.06), likely due to global normalization effects across warehouses. Performance improved for most warehouses, except those with extreme values that were not capped. The changes in results can be attributed to the following reasons:

1. **Capped Extreme Values**:
   - Limiting the impact of large outliers in Prague_1 reduced prediction error significantly.
2. **Global Normalization Effects**:
   - Neural Prophet applies normalization across the dataset, so modifying one warehouse (Prague_1) indirectly impacts others (e.g., Prague_2).
3. **Shared Regressors**:
   - Features like holidays and shops_closed are shared among warehouses. Changing Prague_1's data distribution could alter how these regressors influence predictions for other warehouses.

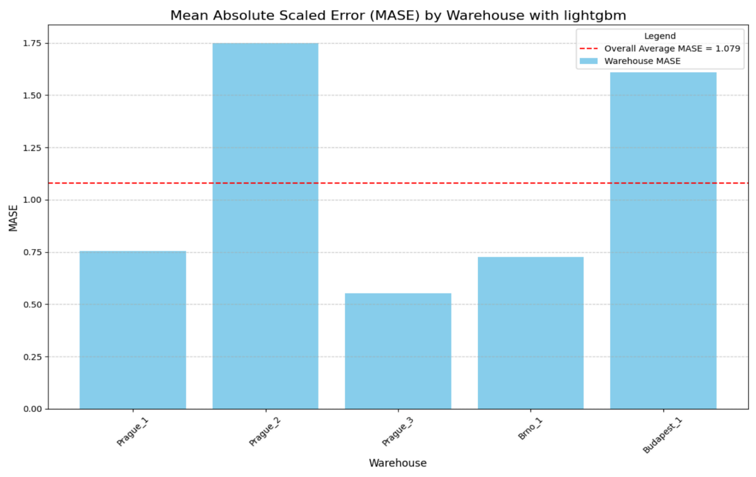


This improvement demonstrates our commitment to refining the model by addressing outliers in key warehouses. While the performance for Prague1 improved, slight shifts in other warehouses highlight the interconnected nature of data and regressors. Further adjustments, such as capping extreme values in Prague2, could optimize overall results.

· ***LightGBM***

I introduced the LightGBM model as an enhancement to the Neural Prophet-based forecasting process. The main motivation for integrating LightGBM is its flexibility in incorporating additional features, which helps refine predictions. I first aggregated data by date and warehouse and generated time-based features. Then used the LOOCV function to split data into training and validation sets. Iteratively training the model for each warehouse to predict the next 14 days of sales.

The result in Prague1 and Brno1 showed lower MASE, indicating better performance. Prague2 had the highest MASE, likely due to extreme values in the dataset. The figure below shows MASE distribution across warehouses.



Compared to the Neural Prophet model, LightGBM showed: Improved handling of additional regressors and time-based features. Slightly higher MASE overall, suggesting trade-offs between feature complexity and model performance.

Conclusion

In this project, I aimed to forecast 14-day sales demand across multiple warehouses using time-series models. The initial steps included data cleaning (handling duplicates, missing values, and outliers) and feature engineering to ensure high data quality. I implemented Baseline Auto ETS, Neural Prophet, and LightGBM models. The Auto ETS model served as a benchmark for performance comparison, while Neural Prophet and LightGBM explored more advanced approaches by integrating seasonality and external future regressors (e.g., holidays, school closures).

The Neural Prophet model performed well, particularly after capping extreme values in Prague_1, reducing its MASE from 1.20 to 0.55. However, global normalization and shared regressors led to trade-offs, such as increased MASE in Prague_2 (1.06). LightGBM, with additional time-based features and greater flexibility, demonstrated competitive results, achieving an overall MASE of 1.08. Despite its robustness, LightGBM showed sensitivity to extreme values in certain warehouses.

Overall, this project highlights the importance of data preprocessing, model selection, and targeted improvements for enhancing forecast accuracy. Future work can focus on optimizing outlier treatment and refining model inputs to achieve balanced performance across all warehouses.