AI-Driven Transaction Prediction for Workforce Planning in Retail Businesses

Godt Brød

Spis godt. Lev godt.

Contents

List of Figures

List of Tables

1 Introduction

This report is prepared for Godt Brød, a leading coffee and bakery brand, to enhance operational efficiency through AI-driven order forecasting and subsequent workforce planning. The study concentrates on exploratory data analysis, EDA, and the application of predictive models to estimate future order volumes across Godt Brød cafés in Oslo and Bergen, based on historical data and external variables such as weather and traffic patterns. The main objective is to identify the most accurate forecasting model and then develop an implementation plan that employs its results to optimize staffing levels.

A substantial dataset has been assembled from both internal sources, including transaction records and staffing schedules, and external sources such as weather forecasts, cruise ship traffic, and general traffic patterns. These data elements form a wide-ranging basis for predictive modeling and staffing recommendations. In compliance with the General Data Protection Regulation, GDPR, all personal data has been anonymized; employees have been assigned random identifiers, ensuring that no personally identifiable information is used beyond the staff-related data essential for this analysis.

Several predictive techniques have been explored to enhance forecasting accuracy, ranging from traditional statistical models to advanced deep learning approaches. The analysis began with exploratory data analysis (EDA) to uncover trends in revenue, orders, and staffing, as well as correlations with external factors. Baseline predictions were established using multiple regression, regularized regression (Lasso, Ridge), k-nearest neighbors (KNN), and time-series modeling with Autoregressive Integrated Moving Average (ARIMA). To capture nonlinear dependencies, machine learning models such as Random Forest and boosting methods (XGBoost, LightGBM) were employed. Further, deep learning models including neural networks with ReLU activation, recurrent networks (GRU), and hybrid CNN-LSTM architectures were implemented to assess their effectiveness in sequential pattern recognition. Model performance was evaluated through training and validation, with comparative visualizations providing insights into predictive accuracy.

Model effectiveness was quantified through error metrics such as Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), and Mean Absolute Percentage Error (MAPE). In keeping with established best practices, multiple models were evaluated to determine the one with the lowest predictive error. Once identified, the best-performing model was refined and subsequently integrated into a detailed implementation plan aimed at streamlining staffing decisions. While this plan provided a data-driven foundation for resource allocation, external circumstances and managerial judgment remained important for real-world workforce management.

By merging AI-driven order forecasting with a clear, systematic implementation strategy, this report aims to give Godt Brød a robust framework for workforce planning. This approach ensures that staffing is aligned with projected demand while concurrently minimizing inefficiencies in day-to-day operations.

2 Data Preparation

2.1 Feature Engineering

2.1.1 Raw Data

The raw sales, staffing, and order data consisted of 208,411 hourly sales records and additional staffing data spanning 2023–2024 across 20 bakeries. Preprocessing was conducted to ensure consistency and remove anomalies before analysis. The key steps were as follows:

1. *Column Standardization:* Variable names were renamed for clarity, and timestamps were converted to a structured format. Hourly features were extracted as:

$$Y_t = \text{year}(T_t), \quad M_t = \text{month}(T_t), \quad D_t = \text{day}(T_t), \quad H_t = \text{hour}(T_t)$$

   where T_t represents the original timestamp.

2. *Staffing Data Processing:* Employee work shifts were determined based on actual clock-in and clock-out times $(T_{\text{in}}, T_{\text{out}})$, which were rounded to the nearest hour. The number of employees present per hour was then aggregated for each department:

$$S_{d,h} = \sum_i 1(i \text{ present at } h)$$

   where $S_{d,h}$ represents the total number of employees in department d at hour h.

3. *Time-Based Expansion:* The recorded shift times were expanded to represent each hour an employee was present. Given the effective start and end times, hourly records were generated:

$$S_{d,h} = \sum_i 1(T_{\text{in},i} \leq h < T_{\text{out},i})$$

   ensuring each employee's work presence was accurately distributed across hours.

4. *Filtering Operating Hours:* Data was restricted to business hours ($06 \leq H_t \leq 20$) to exclude overnight noise for both sales and staffing data.

5. *Transaction and Staffing Volume Threshold:* Departments with fewer than 2000 transactions or unreliable staffing records were excluded from further analysis.

6. *City Classification:* Stores were categorized into Bergen or Oslo based on predefined department IDs, ensuring regional consistency.

7. *Historical Feature Engineering:* To capture temporal dependencies, past staffing levels were calculated:

$$S_{t-7} \text{ (same hour, previous week)}, \quad S_{t-365} \text{ (same hour, previous year)}$$

   These were used to model staffing patterns and their impact on orders.

8. *Outlier Removal:* Revenue and staffing anomalies outside the 1st and 99th percentiles were removed to mitigate extreme fluctuations:

$$X_t \in [Q_{0.01}, Q_{0.99}]$$

   where X_t represents either revenue or staffing data.

After preprocessing, the cleaned dataset retained only relevant transactions and staffing information, structured for robust analysis of business operations.

2.1.2 Lag Variables Historical Trends

Predictive modeling requires historical data to identify trends and seasonality. The following lag variables are used to capture temporal patterns:

- *Revenue_Same_Hour_Last_Week* (R_{t-7}) and *Staff_Same_Hour_Last_Week* (S_{t-7}) – Captures short-term weekly patterns by using revenue and staffing from the same hour and weekday one week ago:

$$R_{t-7} = R(t-7, h, d), \quad S_{t-7} = S(t-7, h, d)$$

- *Revenue_Same_Hour_Last_Year* (R_{t-365}) and *Staff_Same_Hour_Last_Year* (S_{t-365}) – Captures seasonal and annual event-based trends by using revenue and staffing from the same hour, weekday, and week of the previous year:

$$R_{t-365} = R(t-365, h, w, d), \quad S_{t-365} = S(t-365, h, w, d)$$

  where w is the week number.

Lag features were calculated in the same manner for both revenue and staffing data: X_{t-1} represents data from the same hour on the previous day, X_{t-7} captures data from the same hour and weekday one week prior, and X_{t-365} reflects data from the same hour, weekday, and week number in the previous year.

Lastly, the dataset was divided into subsets for 2023 and 2024. Data from 2023 is then merged with 2024 using a left join based on `Weekday_Name`, `Week_number`, `Hour`, and `Department_ID`. To ensure temporal consistency, the dataset is sorted by `Department_ID` and `Date`. This separation was necessary due to 2024 being a leap year, which introduced an extra day affecting date-based calculations.

Missing values were imputed hierarchically using median values at different aggregation levels:

$$X_{d,w,h,t} = \begin{cases} X_{d,w,h,t}, & \text{if observed} \\ \text{median}(X_{d,w,h}), & \text{otherwise} \end{cases}$$

If missing, values were replaced by the median within the same department and weekday:

$$X_{d,w,t} = \begin{cases} X_{d,w,t}, & \text{if observed} \\ \text{median}(X_{d,w}), & \text{otherwise} \end{cases}$$

Remaining missing values were filled using the median within each department:

$$X_{d,t} = \begin{cases} X_{d,t}, & \text{if observed} \\ \text{median}(X_{d}), & \text{otherwise} \end{cases}$$

2.1.3 External Historical Data

External variables provide crucial insights into customer flow.

The following features were derived from historical traffic data obtained from *Statens vegvesen*, covering three roads in Bergen and three roads in Oslo. The datasets were initially processed separately, including data cleaning and handling of missing values, before being merged into a unified dataset.

The dataset includes hourly traffic inflow and outflow, aggregated across monitoring points in each city:

$$IN_{c,h,t} = \sum_{p \in P_c} V^{\text{in}}_{p,h,t}, \quad OUT_{c,h,t} = \sum_{p \in P_c} V^{\text{out}}_{p,h,t}$$

where:

- $IN_{c,h,t}$ represents the total inbound traffic for city c at hour h on day t.
- $OUT_{c,h,t}$ represents the total outbound traffic for city c at hour h on day t.
- P_c is the set of monitoring points in city c.
- $V^{\text{in}}_{p,h,t}$ and $V^{\text{out}}_{p,h,t}$ denote the measured inflow and outflow at monitoring point p at hour h on day t.
- *Traffic_Trend_Prev_Week* (T_{t-7}) – 7-day moving average of traffic volume from the same hour and weekday in the previous week:

$$T_{t-7} = \frac{1}{7} \sum_{i=t-14}^{t-8} V_i$$

- *Traffic_Trend_Two_Weeks_Ago* (T_{t-14}) – 7-day moving average from the same hour and weekday two weeks ago:

$$T_{t-14} = \frac{1}{7} \sum_{i=t-21}^{t-15} V_i$$

- *Traffic_Change_Rate* (ΔT) – Percentage change in traffic volume between the two reference periods:

$$\Delta T = \left(\frac{T_{t-7} - T_{t-14}}{T_{t-14}} \right) \times 100$$

2.1.4 External Predictive Data

The following variables are based on historical data but can be forecasted up to 7 days in advance:

Data on the number of cruise ships docked at ports in Bergen and Oslo was obtained from *Kystverket*. This dataset, collected on a daily basis, serves as an indicator of potential fluctuations in customer volume, as the presence of cruise ships is associated with increased foot traffic and retail activity.

The dataset contains records of ship arrivals and departures, which were processed as follows:

1. *Data Preparation:* Column names were standardized, and the date was reconstructed from separate year, month, and day fields:

$$D_t = \text{datetime}(Y_t, M_t, D_t)$$

where D_t represents the converted date.

2. *Event Matching:* The dataset was sorted by ship name and date to correctly associate each arrival with the next recorded departure.

3. *Time in Port Calculation:* For each ship, the length of stay was determined by computing the difference between the departure and arrival dates:

$$T_{\text{port}} = D_{\text{departure}} - D_{\text{arrival}}$$

where T_{port} denotes the number of days the ship remained docked.

4. *Daily Aggregation:* The total number of ships present in port was computed for each city and date:

$$S_{c,t} = \sum_{s \in P_c} I_{s,t}$$

where $S_{c,t}$ represents the number of ships docked in city c on day t, and $I_{s,t}$ is an indicator function equal to 1 if ship s was present in port on that day.

Since this dataset was structured at a daily level, it was merged with the primary dataset to align with other time-dependent features and incorporate cruise ship presence as a potential explanatory factor for variations in customer volume.

Weather data was obtained from *Norsk Klimaservicesenter*, ensuring access to reliable meteorological observations for Bergen and Oslo. The dataset includes key explanatory metrics such as maximum temperature, hourly precipitation, minutes of rainfall per hour, and sunshine duration per hour.

The data collection process involved the following steps:

1. *Weather Station Identification:* Suitable weather stations were identified in each city to ensure comprehensive and accurate measurements of relevant climatic variables.

2. *Data Extraction and Standardization:* Observations were collected for each station and assigned a corresponding city label to distinguish between locations.

3. *Dataset Integration:* The datasets from Bergen and Oslo were merged, aligning weather variables with the existing dataset to facilitate further analysis.

This structured approach ensured that meteorological conditions were properly incorporated as potential explanatory factors in the analysis.

2.1.5 Final Merge

All datasets were standardized to ensure consistency in column naming conventions, enabling seamless integration during the merging process. The key steps involved:

1. *Column Standardization:* All tables were aligned to use uniform column names, ensuring compatibility across different data sources. The primary keys used for merging were:

$$\text{City}, \quad \text{Year}, \quad \text{Month}, \quad \text{Day}, \quad \text{Hour}$$

2. *Merging Process:* The datasets were merged sequentially using left joins, ensuring that all relevant information was retained. The merging process followed the structure:

$$DF = DF_1 \cup DF_2 \cup DF_3 \cup DF_{\text{expanded}}$$

where each dataset contributed different explanatory variables.

3. *Data Type Conversion:* After merging, numerical variables were converted to the appropriate data types:

   Integer: Department ID, Year, Month, Day, Hour, Ships in Port
   Floating Point: Revenue, Traffic Volume, Percentage Changes
   Categorical: City

4. *Feature Normalization:* The numeric columns that were not binary were normalized using min-max scaling based on their unique department ID for those analyses that required it, while leaving them unchanged for analyses where normalization was unnecessary. The min-max normalization was applied as follows:

$$X'_{d,t} = \frac{X_{d,t} - \min(X_d)}{\max(X_d) - \min(X_d)}$$

   where:

   - $X_{d,t}$ represents the original value of feature X for department d at time t.
   - $\min(X_d)$ and $\max(X_d)$ are the minimum and maximum values of feature X within department d.
   - $X'_{d,t}$ is the normalized value, scaled to the range $[0, 1]$ within each department.

   This transformation ensured comparability across departments while preserving relative differences within each department's dataset. However, for analyses that did not require normalization, the original scale of the numeric features was retained to maintain interpretability and consistency.

5. *Data Completeness Check:* To ensure data integrity, the number of unique months per department was verified:

$$M_d = |\{m \mid d, y\}|$$

   where M_d represents the number of unique months available for department d in a given year. Departments missing full-year data were flagged for further review.

This approach ensured that all datasets were harmonized, structured, and ready for analysis while maintaining completeness and correctness.

3 EDA

Exploratory Data Analysis (EDA) provides a systematic investigation of revenue, orders, staffing, and external factors, revealing trends, correlations, and anomalies. The cleaned and calculated dataset comprises 42,887 observations and 27 variables, encompassing business performance metrics across multiple time dimensions, traffic levels, and weather conditions. No missing values were detected, ensuring data completeness.

3.1 Revenue, Orders, and Staffing Trends

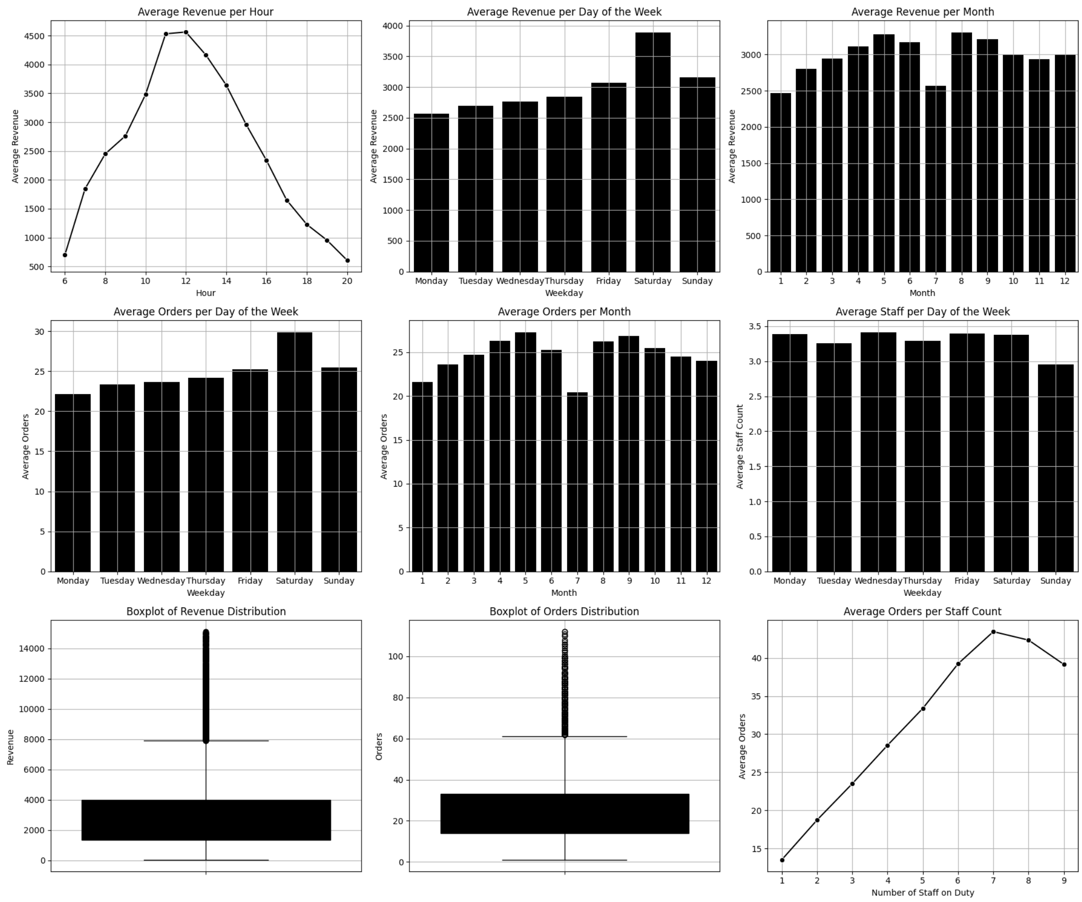

Figure 1: Revenue, orders, and staffing trends.

Revenue follows a predictable daily pattern, peaking at midday, aligning with peak consumer activity. Weekly trends indicate that Saturdays generate the highest revenue, while Mondays remain the lowest. Monthly revenue fluctuations suggest mid-year peaks, likely due to seasonal demand. Orders closely follow revenue trends, while staffing levels remain consistent, slightly increasing on high-revenue days. However, order efficiency per staff declines beyond a certain threshold, indicating diminishing productivity returns.

Boxplots reveal a right-skewed distribution for revenue and orders, with significant outliers. These high-revenue instances suggest periods of exceptional performance, likely driven by promotions, external demand surges, or special events. Retaining these outliers is essential, as they capture real-world fluctuations in demand and peak performance scenarios. Removing them would distort variability and underestimate revenue potential. Their inclusion ensures predictive models reflect true market dynamics rather than an artificially smoothed dataset.

3.2 Correlation Analysis and External Influence

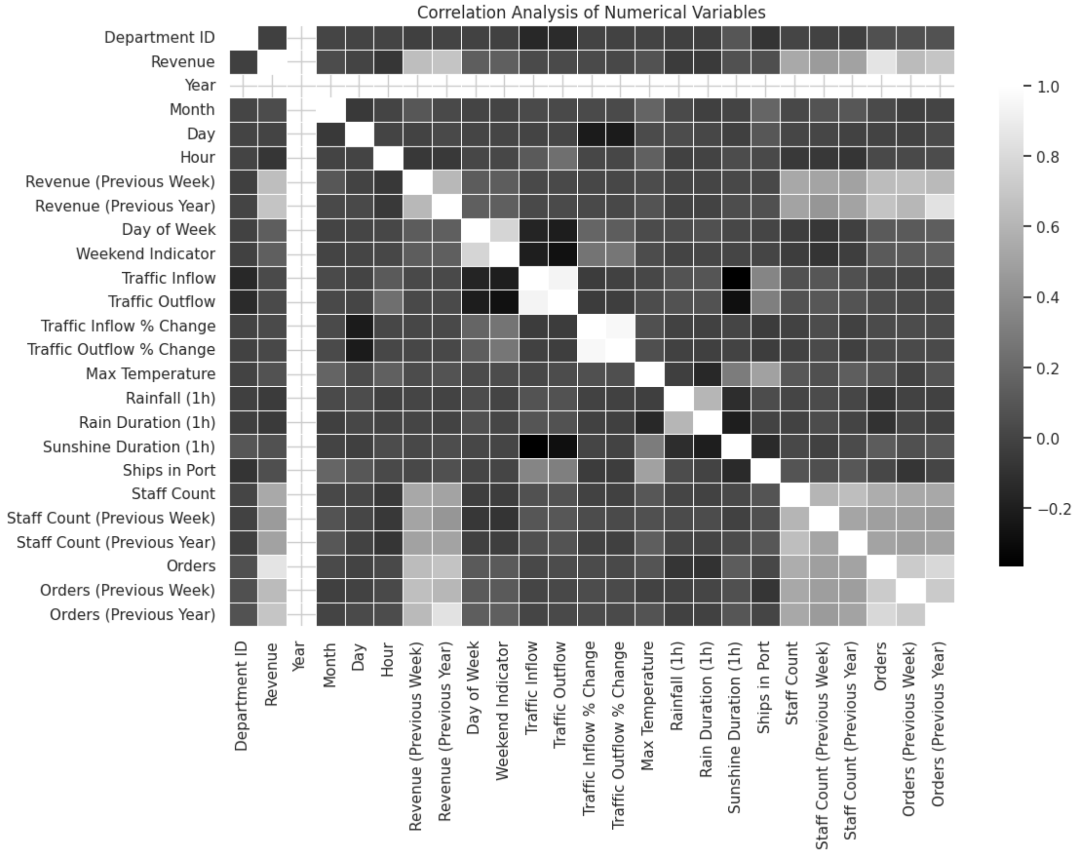

Figure 2: Correlation Analysis of Numerical Variables

The correlation analysis confirms that revenue is primarily driven by order volume. Staffing levels demonstrate a moderate correlation with revenue, indicating that workforce allocation plays a role but is not the sole determinant of business performance. Traffic inflow shows a weak but positive relationship with revenue, suggesting that customer movement patterns contribute to sales. However, weather conditions, including temperature and rainfall, exhibit minimal correlation, indicating that external climatic factors have limited impact on business operations.

4 Baseline Statistical Models

4.1 Regression Analysis

4.1.1 Model Description

To analyze the relationship between revenue, orders, and staff count across different departments, we employ regression analysis. The fundamental model is expressed as:

$$Y_i = \beta_0 + \sum_{j=1}^{n} \beta_j X_{ji} + \epsilon_i \tag{1}$$

where:

- Y_i represents the dependent variable (either *Revenue* or *Orders*).
- X_{ji} are the independent variables.
- β_0 is the intercept.
- β_j are the slope coefficients for each independent variable.
- ϵ_i is the error term.

The explanatory power of these regressions varies across departments, as shown in Figures **??** and **??**. This suggests that staffing and order patterns influence revenue differently depending on location-specific factors.

To further investigate how staff count interacts with orders, we extend the model by including an interaction term:

$$Revenue_i = \beta_0 + \beta_1 \cdot Staff_i + \beta_2 \cdot Orders_i + \beta_3 \cdot (Staff_i \times Orders_i) + \epsilon_i \tag{2}$$

where:

- β_1 represents the direct effect of staff count on revenue. A positive coefficient indicates that increasing staff generally leads to higher revenue, while a negative coefficient suggests that additional staff may reduce efficiency.
- β_2 captures the effect of orders on revenue, which is expected to be positive as an increase in orders should lead to increased revenue.
- β_3 is the interaction coefficient, which shows whether the impact of staff count on revenue depends on the number of orders.
  - If $\beta_3 > 0$, the positive effect of staff on revenue is stronger when order volume is high, suggesting that increased staffing is beneficial in high-demand scenarios.
  - If $\beta_3 < 0$, additional staff members have a diminishing or even negative effect on revenue at high order volumes, which may indicate inefficiencies such as overcrowding or redundant labor.
- ϵ_i represents the error term, accounting for unexplained variability in revenue.

This interaction model allows us to determine whether the relationship between staff count and revenue remains consistent across all order levels or whether staffing decisions should be adjusted dynamically based on order fluctuations.

4.1.2 Revenue Prediction

The results indicate that orders ($R^2 \approx 0.73$) is the strongest predictor of revenue, aligning with expectations. Historical revenue trends (*Revenue (Previous Year) and Revenue (Previous Week)*) also demonstrate relatively high explanatory power (R^2 between 0.40–0.50), reinforcing the importance of past performance in forecasting revenue. However, a notable observation is that (*staff count*) also serves as a moderate predictor of revenue, unlike past revenue, which cannot be adjusted. This highlights the role of workforce allocation in optimizing revenue generation, as illustrated in Figure 3.

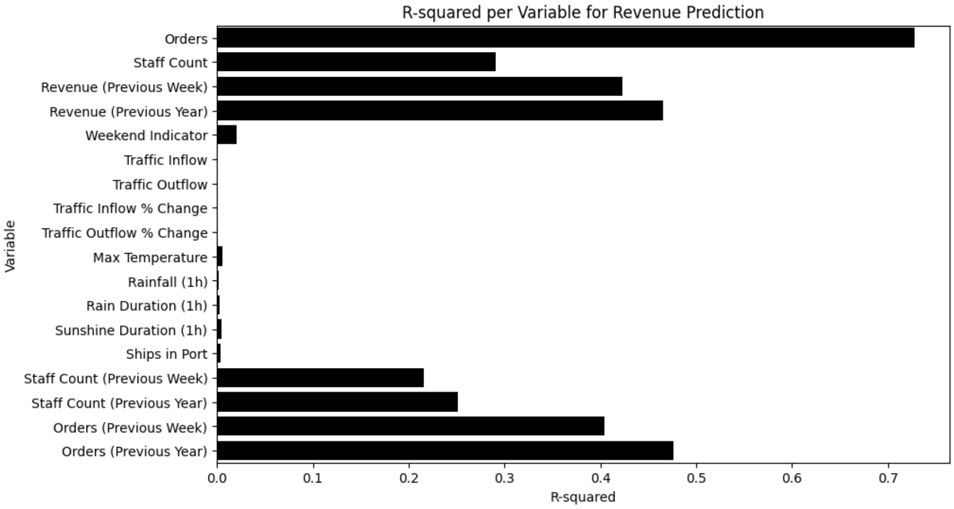

Figure 3: R-squared per Variable for Revenue Prediction

4.1.3 Staff Count as an Explanatory Variable for Revenue

When examining staff count as a predictor of revenue, the explanatory power varies across departments. Some departments exhibit moderate R^2 values, suggesting that staffing has a meaningful impact on revenue, while others show lower values, indicating that either staffing levels do not significantly impact financial performance, or there may be inefficiencies in how employees are allocated. These departments could benefit from a more data-driven approach to staffing, ensuring that labor costs align more effectively with revenue generation. Table 1 illustrates these differences across departments.

Department ID	Intercept	Slope	R-squared	P-value
11	804.03	640.55	0.2058	7.00e-166
12	607.49	816.01	0.3045	9.40e-289
13	6.31	863.76	0.3262	1.96e-233
17	178.03	843.94	0.3239	1.42e-291
20	1666.67	247.03	0.0573	9.25e-47
21	1622.7	267.6	0.0776	7.65e-69
22	1351.5	475.66	0.1668	5.74e-147
24	-44.26	837.55	0.2863	1.54e-238
25	1043.64	835.07	0.2064	1.94e-182
26	945.63	499.86	0.1043	1.96e-63
27	1018.19	439.97	0.1132	7.69e-83
32	213.7	895.65	0.3534	2.30e-295
36	803.79	612.95	0.258	3.99e-218

Table 1: Regression results per Department ID Revenue

Similarly, Table 2 presents the regression results for the relationship between staff count and orders. Departments such as 17, 24, and 32 show stronger associations, suggesting that their staffing levels are well-adjusted to customer demand. Again, the explanatory power varies significantly, highlighting the need for customized staffing strategies.

Department ID	Intercept	Slope	R-squared	P-value
11	8.00	3.97	0.2124	7.03e-172
12	8.66	5.14	0.3405	0.00e+00
13	0.99	5.93	0.371	9.97e-274
17	4.92	6.20	0.4115	0.00e+00
20	15.48	1.62	0.0642	2.32e-52
21	15.32	2.40	0.1292	1.56e-116
22	13.13	3.22	0.196	2.61e-175
24	2.27	5.71	0.3992	0.00e+00
25	14.04	6.44	0.2105	1.99e-186
26	9.72	3.09	0.1091	1.79e-66
27	12.54	2.79	0.0988	6.14e-72
32	5.56	6.22	0.395	0.00e+00
36	13.84	4.03	0.2458	2.28e-206

Table 2: Regression results per Department ID Orders

4.1.4 Interaction Analysis

To gain deeper insights into how staff count influences revenue under varying order volumes, we introduce an interaction term in the regression model. Analyzing interaction effects helps determine whether the effect of staff count on revenue is more pronounced under high or low order volumes. The results reveal substantial variation across departments, as illustrated in Figure 4.

Figure 4: Heatmap of staff count and order interaction on revenue.

This heatmap illustrates the interaction effect of staff count and orders on revenue across different departments. Lighter shades indicate a stronger positive interaction, suggesting that increasing staff is more beneficial when order volumes are high. Conversely, darker shades represent a negative interaction effect, where additional staffing at higher order levels may reduce efficiency.

Key findings from the interaction analysis include:

- *Departments 27, 11, 36, 13, and 32* benefit from increased staffing when orders are high, indicating that a flexible workforce strategy improves performance in these areas.
- *Departments 26, 22, 25, 17, 20, and 12* also show positive interactions, though excessive staffing during low-demand periods may lead to inefficiencies.
- *Departments 24 and 21* exhibit a negative interaction effect, meaning that additional staff reduces efficiency at high order volumes, possibly due to overcrowding, redundancy, or misallocation of labor.

These variations emphasize the importance of tailoring staffing decisions to individual department characteristics, ensuring that workforce levels align with actual demand to maximize efficiency and revenue generation.

4.2 KNN

4.2.1 Model Description

The k-Nearest Neighbors (KNN) regression model is a non-parametric approach that estimates a target variable Y by averaging the values of the k closest observations in the feature space. Given a dataset $\mathcal{D} = \{(X_i, Y_i)\}_{i=1}^{N}$, where X represents the input features and Y the target variable, the KNN prediction for a new input x^* is computed as:

$$\hat{Y}(x^*) = \frac{1}{k} \sum_{i \in \mathcal{N}_k(x^*)} Y_i, \tag{3}$$

where $\mathcal{N}_k(x^*)$ denotes the set of k nearest neighbors of x^* according to a chosen distance metric, typically the Euclidean distance:

$$d(X_i, X_j) = \sqrt{\sum_{m=1}^{M} (X_{im} - X_{jm})^2}. \tag{4}$$

4.2.2 Training and Evaluation

The model is trained using a fixed $k = 5$ for all departments. This value is chosen based on standard practice in KNN regression to balance bias and variance.

The Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE) are used as evaluation metrics:

$$\text{MAE} = \frac{1}{N} \sum_{i=1}^{N} \left| Y_i - \hat{Y}_i \right|. \tag{5}$$

$$\text{RMSE} = \sqrt{\frac{1}{N} \sum_{i=1}^{N} (Y_i - \hat{Y}_i)^2}. \tag{6}$$

Department	**Fixed K**	**MAE**	**RMSE**	**Avg. Orders**	**Error Margin (%)**
11	5	3.55	4.84	13.73	25.9
12	5	5.87	7.73	31.89	18.4
13	5	5.44	7.61	35.48	15.3
17	5	5.71	7.73	36.47	15.7
20	5	4.03	5.34	19.18	21.0
21	5	4.78	6.42	21.55	22.2
22	5	4.24	5.55	22.07	19.2
24	5	4.14	5.49	20.87	19.8
25	5	4.97	6.77	24.93	19.9
26	5	4.95	6.64	17.92	27.6
27	5	4.45	5.90	18.36	24.2
32	5	6.20	8.36	37.70	16.5
36	5	6.23	8.46	29.97	21.5

Table 3: KNN Prediction Performance per Department with Fixed $k = 5$, Including Avg. Orders and Error Margin

4.2.3 Results

The results show that with a fixed $k = 5$, the model performs consistently across different departments, achieving reasonable MAE and RMSE values.

4.2.4 Visualization of Predictions

To illustrate the performance of the KNN model, we visualize actual versus predicted orders for a randomly selected department and week. The figure below shows the comparison of predicted and actual order volumes for Department 24 in Week 20.

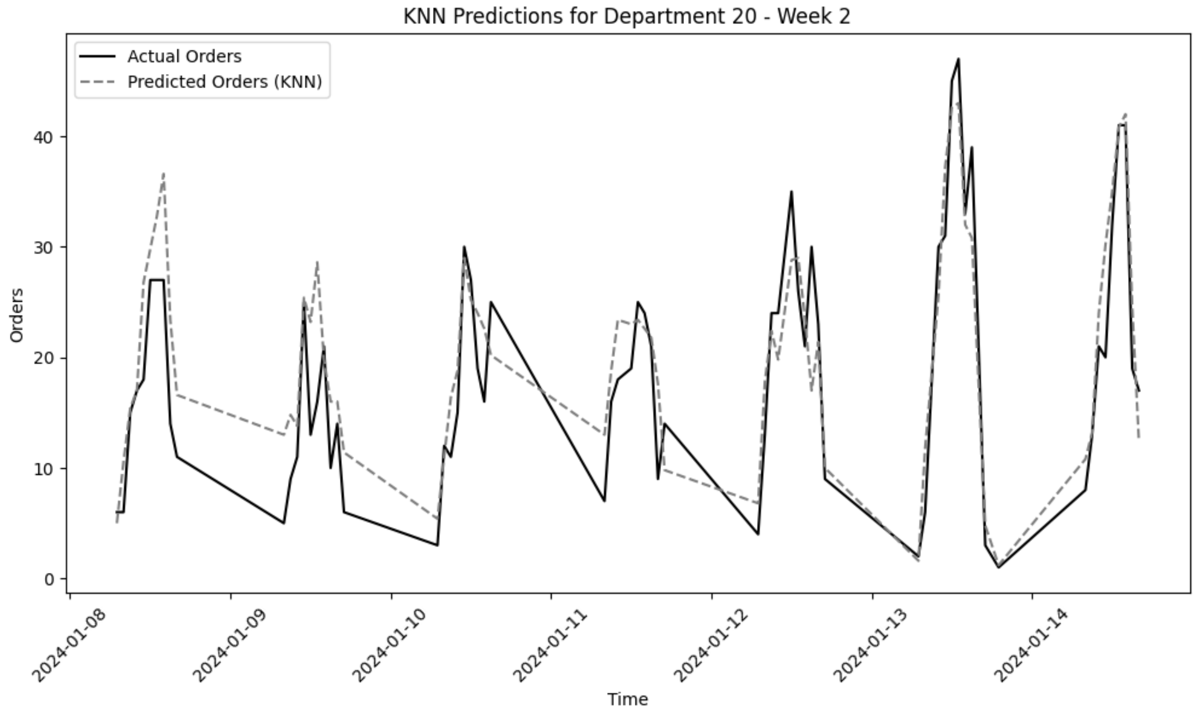

Figure 5: Actual vs. Predicted Orders for Department 20 - Week 2 using KNN with $k = 5$

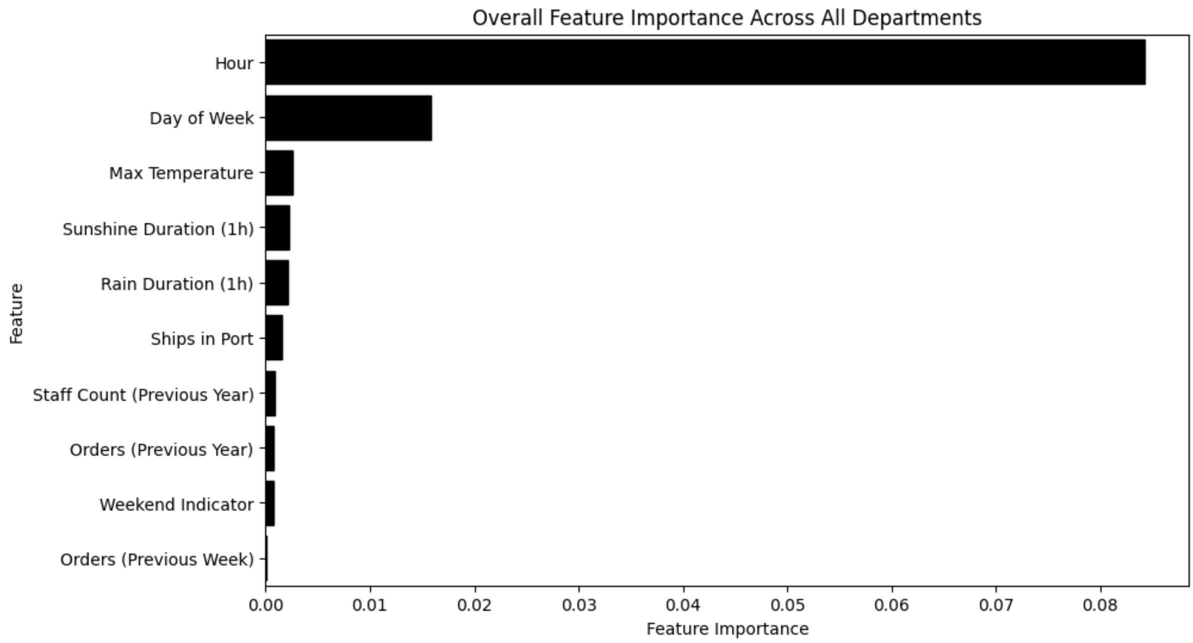

Figure 6: Feature Importance for KNN Overall

4.3 Regularized Regression (Lasso, Ridge)

4.3.1 Model Description

The Ridge and Lasso regression models were trained for various departments to predict the target variable. Regularization strength was optimized to balance bias-variance tradeoff.

Mathematically, Ridge regression minimizes the following objective function:

$$\min_{\beta} \sum_{i=1}^{n} (y_i - X_i\beta)^2 + \lambda \sum_{j=1}^{p} \beta_j^2 \tag{7}$$

where λ is the regularization parameter that controls the penalty on large coefficients.

Similarly, Lasso regression minimizes:

$$\min_{\beta} \sum_{i=1}^{n} (y_i - X_i\beta)^2 + \lambda \sum_{j=1}^{p} |\beta_j| \tag{8}$$

which results in sparsity by driving some coefficients to zero.

4.3.2 Training and Evaluation

The models were trained on historical data, and evaluation was conducted using Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE). The performance metrics for selected departments are as follows:

Department	Ridge MAE / RMSE	Lasso MAE / RMSE	Avg. Orders	Error Margin (%)
11	3.96 / 5.22	3.96 / 5.23	13.73	28.5%
12	7.00 / 9.21	7.02 / 9.24	31.89	20.5%
13	7.11 / 9.33	7.10 / 9.33	35.48	16.5%
17	7.19 / 9.65	7.18 / 9.64	36.47	17.2%
20	4.82 / 6.41	4.82 / 6.41	19.18	23.6%
21	5.42 / 7.39	5.38 / 7.37	21.55	23.5%
22	5.07 / 6.72	5.07 / 6.71	22.07	22.3%
24	5.07 / 6.58	5.06 / 6.58	20.87	22.0%
25	6.20 / 8.52	6.20 / 8.52	24.93	20.8%
26	5.57 / 7.70	5.56 / 7.71	17.92	32.0%
27	5.56 / 7.13	5.55 / 7.12	18.36	26.9%
32	7.91 / 10.52	7.92 / 10.53	37.70	18.3%
36	7.87 / 10.56	7.86 / 10.56	29.97	21.4%

Table 4: Performance of Ridge and Lasso models across departments

4.3.3 Results

The Ridge and Lasso models exhibited similar performance across departments, with minor variations in MAE and RMSE. Ridge generally showed slightly lower RMSE values, indicating better generalization. Lasso had more feature sparsity, suggesting its usefulness for feature selection.

4.3.4 Visualization of Predictions

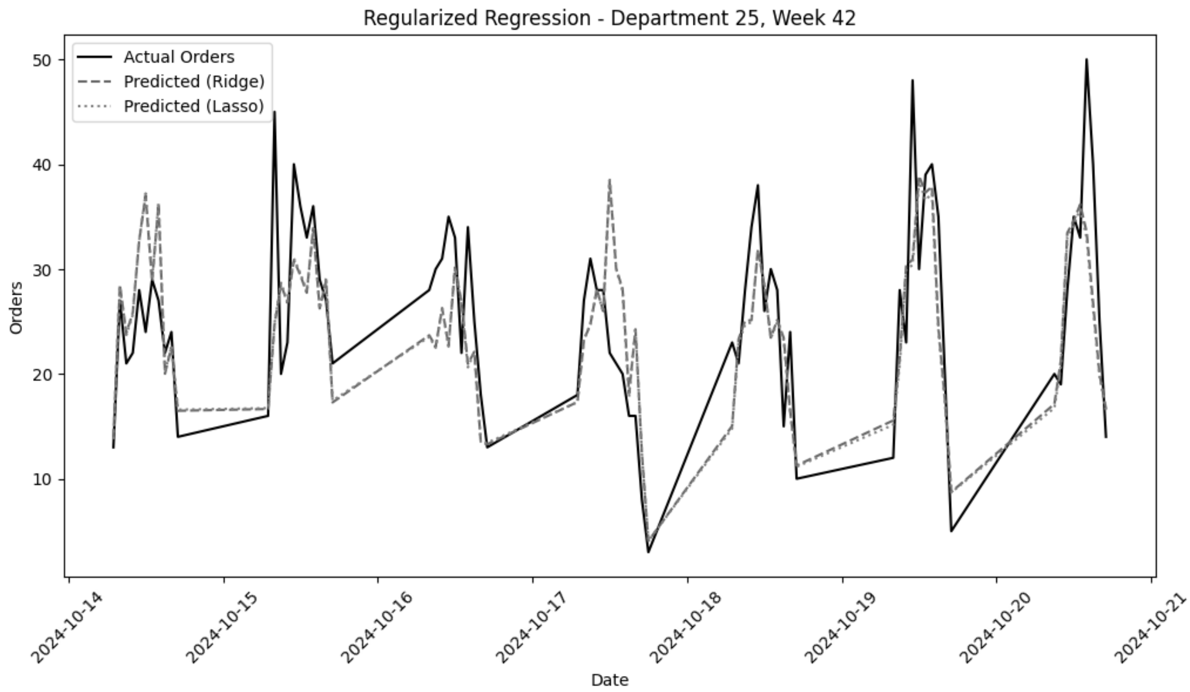

Figure 7: Comparison of Ridge and Lasso regression models

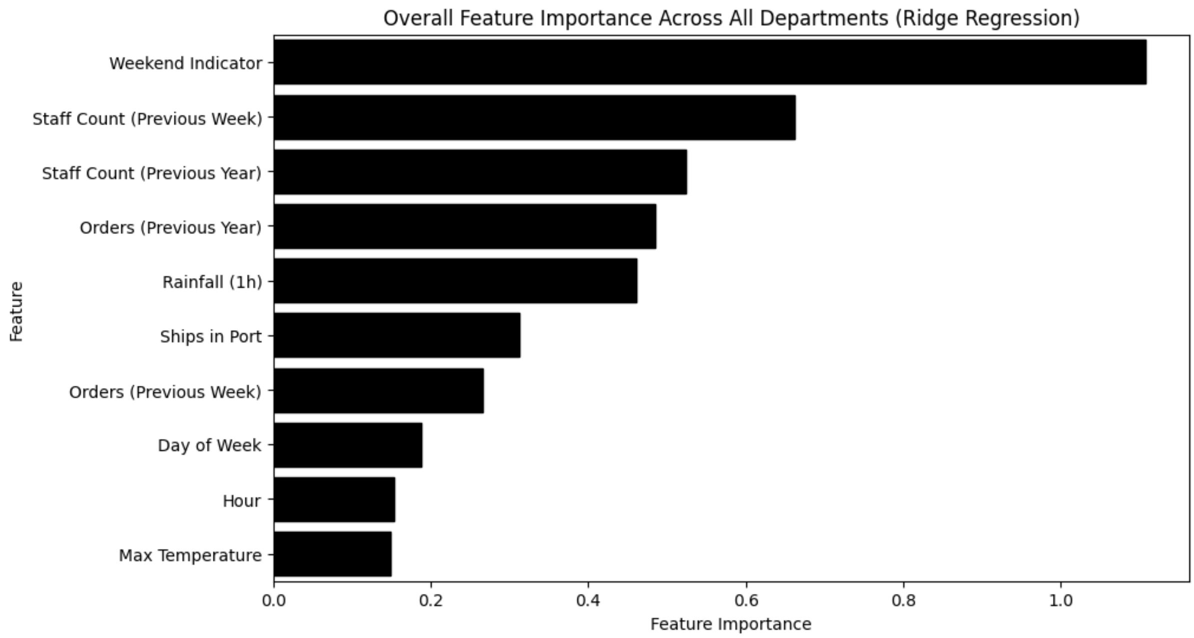

Figure 8: Overall feature importance for Ridge and Lasso models

Visual analysis of predicted vs. actual values showed that both models captured trends accurately, though residuals indicated some variance in predictions across different departments.

4.4 Time Series Models (ARIMA)

4.4.1 Model Description

The ARIMA models were trained to predict hourly order volumes for different departments using the most recent four months of data. The models capture temporal dependencies and trends in the order data to provide accurate forecasts.

An ARIMA(p, d, q) model consists of three components:

- p: the number of autoregressive (AR) terms.
- d: the number of differencing operations needed to make the time series stationary.
- q: the number of moving average (MA) terms.

Mathematically, an ARIMA(p, d, q) model is defined as:

$$\Phi(B)(1 - B)^d Y_t = \Theta(B)\epsilon_t \tag{9}$$

where:

- Y_t is the observed time series.
- B is the backshift operator, i.e., $BY_t = Y_{t-1}$.
- $(1 - B)^d$ represents the differencing operation to make the series stationary.
- $\Phi(B) = 1 - \phi_1 B - \phi_2 B^2 - \cdots - \phi_p B^p$ is the autoregressive (AR) polynomial.
- $\Theta(B) = 1 + \theta_1 B + \theta_2 B^2 + \cdots + \theta_q B^q$ is the moving average (MA) polynomial.
- ϵ_t is a white noise error term with mean zero and variance σ^2.

The parameters (p, d, q) are selected based on model evaluation criteria such as the Akaike Information Criterion (AIC) and Bayesian Information Criterion (BIC).

4.4.2 Training and Evaluation

Each department's data was used to train an individual ARIMA model. The models were evaluated using Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE) metrics. The results for different departments are shown in Table 5.

Department	ARIMA MAE	ARIMA RMSE	Avg. Orders	Error Margin (%)
11	5.50	7.39	13.73	40.06
12	10.15	12.44	31.89	31.83
13	11.15	13.58	35.48	31.43
17	11.67	14.31	36.47	32.00
20	7.65	9.37	19.18	39.89
21	12.56	15.39	21.55	58.29
22	8.95	11.09	22.07	40.56
24	11.75	14.74	20.87	56.30
25	9.26	11.45	24.93	37.15
26	12.45	16.75	17.92	69.49
27	7.62	9.77	18.36	41.50
32	15.82	19.32	37.70	41.96

Table 5: ARIMA Model Evaluation Results for Different Departments (Corrected Error Margins)

4.4.3 Results

The ARIMA models attempted to capture order trends and variations across different departments. However, the performance varied significantly, with some departments exhibiting high forecasting errors. Departments with more stable order patterns achieved lower MAE and RMSE scores, while those with greater volatility showed less reliable predictions.

4.4.4 Visualization of Predictions

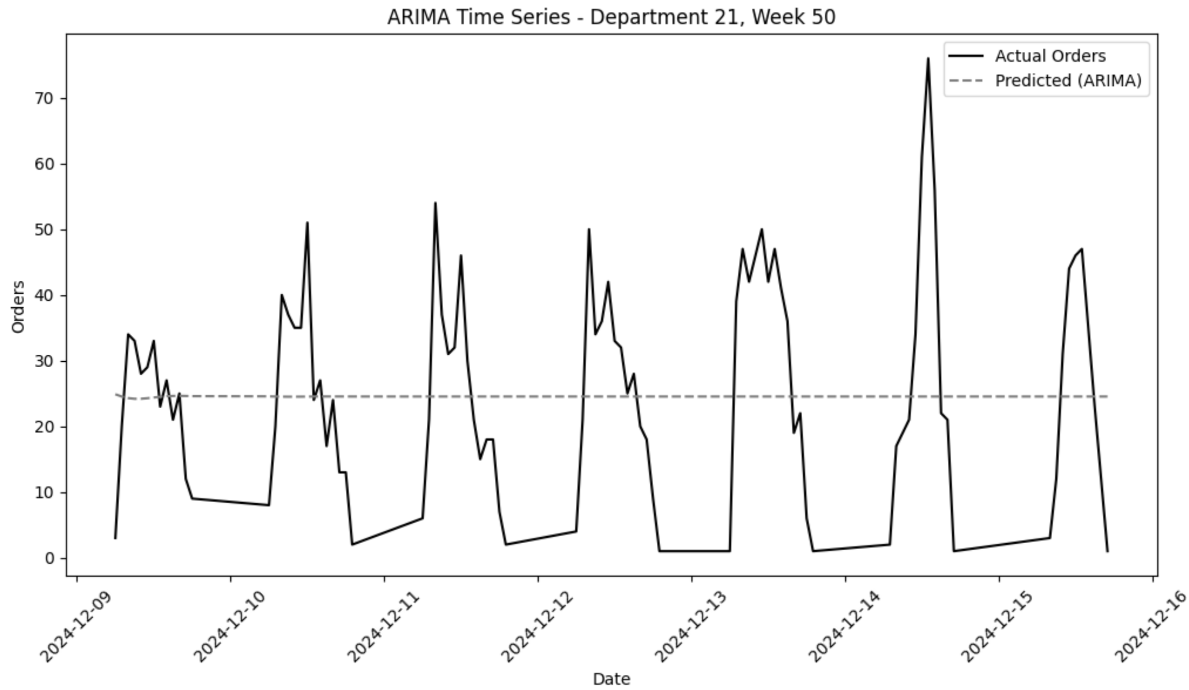

Figure 9: ARIMA model predictions compared to actual order volumes. The discrepancy between predicted and actual values highlights the challenges in forecasting volatile order patterns.

5 Machine Learning Models

The performance of the predictive models was assessed using two standard error metrics: Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE), defined as follows:

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{10}$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{11}$$

5.1 Random Forest Model for Order Prediction

5.1.1 Model Overview

To forecast the hourly order volume for each department, we employed a extitRandom Forest Regressor, an ensemble learning approach leveraging multiple decision trees. The model aggregates predictions from individual trees, formulated as:

$$\hat{y} = \frac{1}{N} \sum_{i=1}^{N} T_i(X) \tag{12}$$

where $T_i(X)$ represents the output of each decision tree, and N denotes the total number of trees in the ensemble.

5.1.2 Training and Evaluation Methodology

The dataset was partitioned into an 80% training set and a 20% test set for model validation. The Random Forest model was configured with 100 decision trees, each with a maximum depth of 10.

5.1.3 Results

Table 6 presents the MAE and RMSE for each department.

Table 6: Random Forest Prediction Performance per Department

Department	MAE	RMSE	Avg. Orders	Error Margin (%)
11	3.07	3.91	13.73	22.4%
12	5.51	7.54	31.89	17.3%
13	6.35	8.95	35.48	17.9%
17	6.35	8.21	36.47	17.4%
20	4.28	5.47	19.18	22.3%
21	5.33	7.12	21.55	24.7%
22	4.89	6.37	22.07	22.2%
24	5.21	6.71	20.87	25.0%
25	4.22	5.50	24.93	16.9%
26	4.79	6.21	17.92	26.7%
27	5.02	6.95	18.36	27.3%
32	6.62	8.52	37.70	17.6%
36	4.97	6.19	29.97	16.6%

5.1.4 Visualization of Predictions

Figure 10 illustrates the predicted and actual orders for a randomly selected department over a single week.

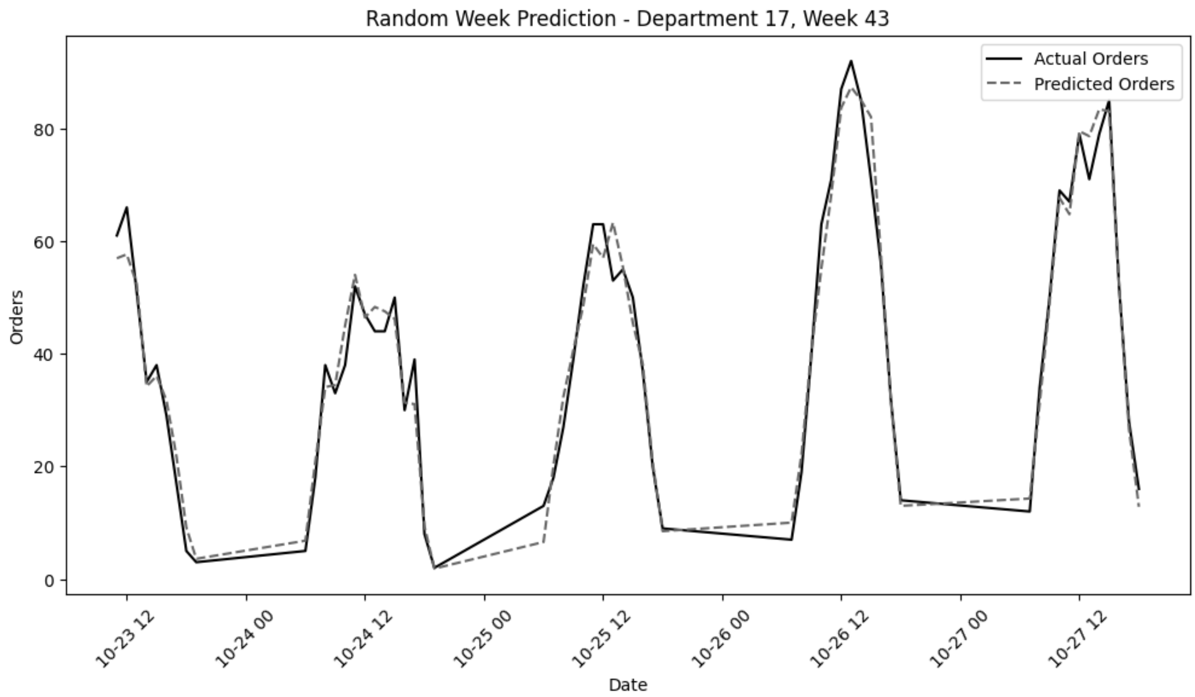

Figure 10: Random Forest Predictions vs. Actual Orders for Department 13, Week 51

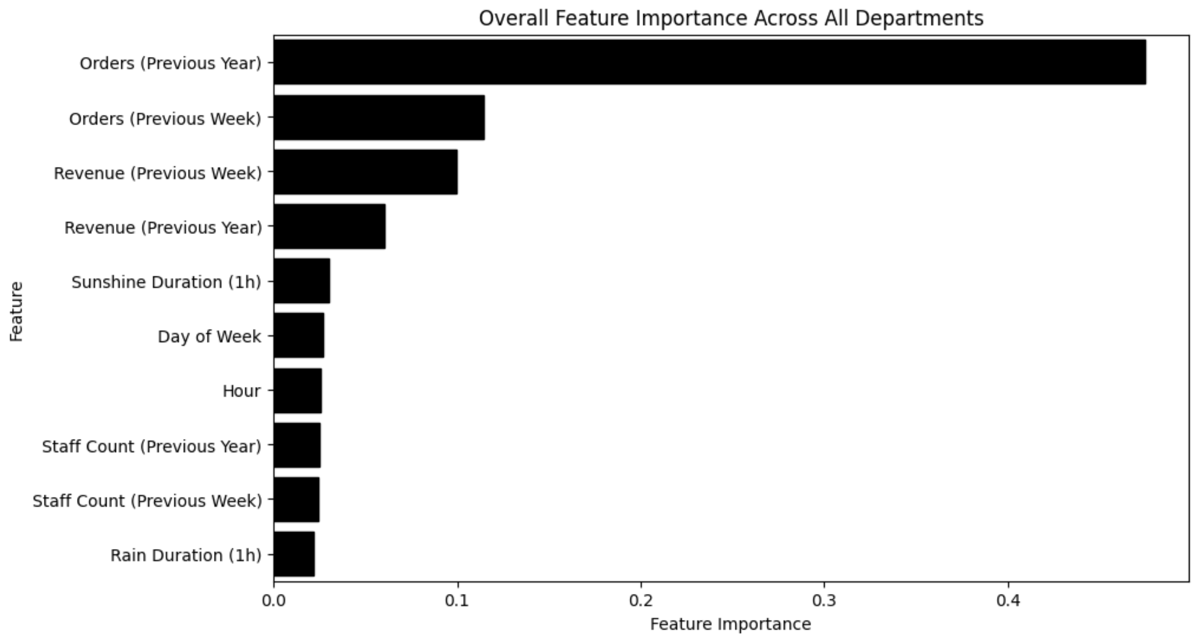

Figure 11: Random Forest Overall Feature Importance

The model effectively captures the overall trend in order fluctuations, with minor deviations observed in peak hours.

5.2 Boosting Models: XGBoost and LightGBM

5.2.1 Model Overview

Boosting models such as XGBoost and LightGBM iteratively enhance weak learners by minimizing a predefined loss function $L(y, \hat{y})$. The objective function is expressed as:

$$\mathcal{L}(\theta) = \sum_{i=1}^{n} L(y_i, f(x_i; \theta)) + \Omega(f) \quad (13)$$

where:

- $L(y_i, f(x_i; \theta))$ denotes the loss function (e.g., Mean Squared Error for regression)
- $\Omega(f)$ represents a regularization term to prevent overfitting
- $f(x_i; \theta)$ is the predictive function based on decision trees

Each new boosting iteration refines previous residuals as follows:

$$h_t(x) = -\gamma \frac{\partial L(y, \hat{y})}{\partial \hat{y}} \quad (14)$$

where γ is the learning rate controlling the impact of each subsequent tree.

5.2.2 Training and Evaluation

Both models were trained on historical sales data across multiple departments, using an 80% training and 20% validation split. Hyperparameter tuning was conducted, optimizing:

- Learning rate (η)
- Maximum tree depth (d)
- Minimum child weight (w_{min})
- Subsample ratio (s)

5.2.3 Results

The models' performance metrics across different departments are summarized in Table 7.

Department	XGBoost MAE	LightGBM MAE	Avg. Orders	Error Margin (%)
11	3.12	3.09	13.73	22.51
12	5.56	5.55	31.89	17.42
13	7.07	6.86	35.48	19.87
17	6.64	6.41	36.47	17.57
20	4.66	4.32	19.18	24.37
21	5.93	5.60	21.55	26.00
22	5.06	5.04	22.07	22.89
24	5.57	5.22	20.87	26.35
25	4.23	4.49	24.93	17.71
26	4.90	4.92	17.92	27.45
27	5.29	4.97	18.36	27.10
32	6.64	6.41	37.70	17.00

Table 7: Model performance across departments with average orders and error margins

5.2.4 Visualization of Predictions

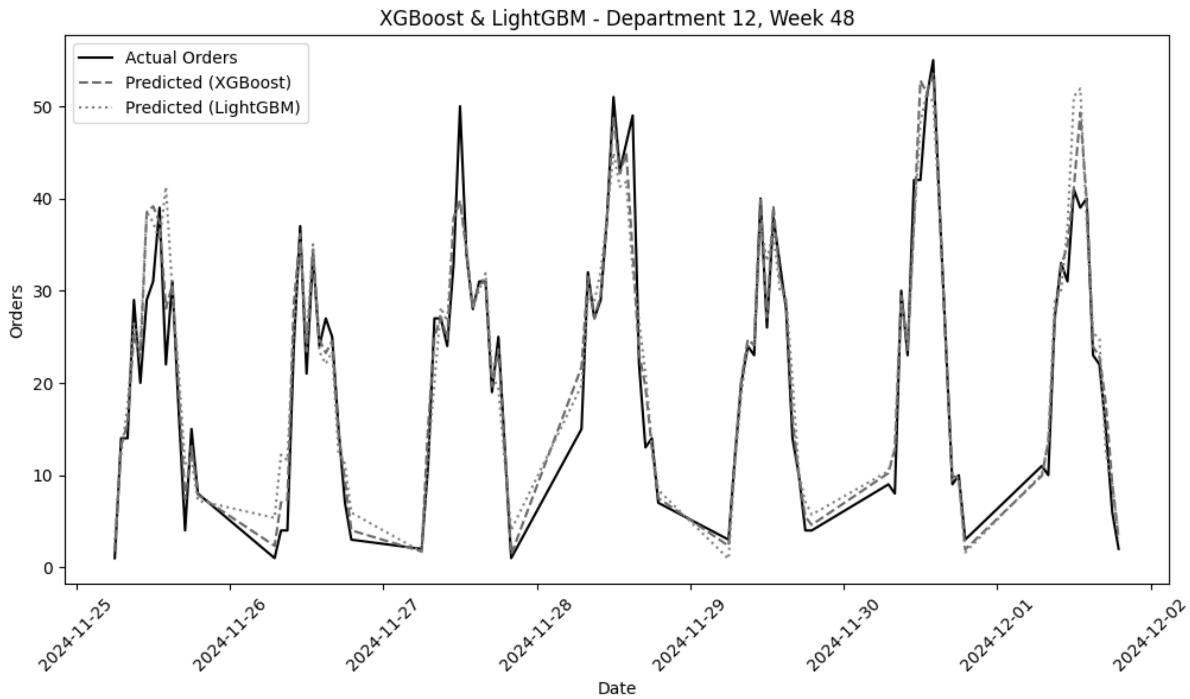

Figure 12: Predictions from XGBoost and LightGBM models

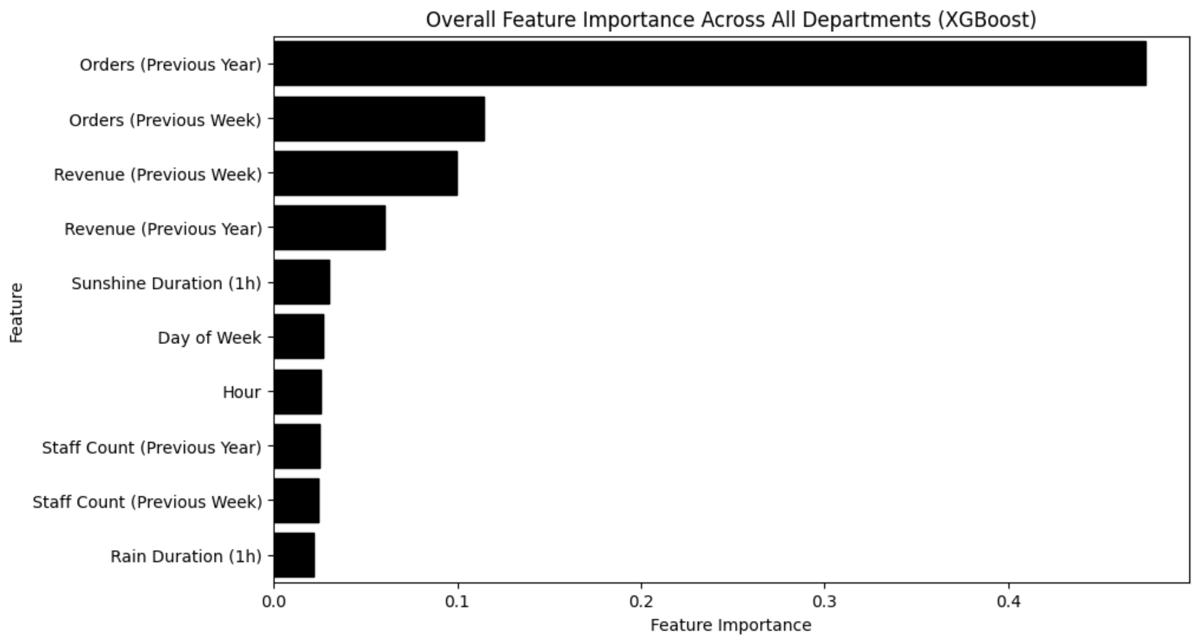

Figure 13: Feature importance analysis for XGBoost and LightGBM

6 Deep Learning

The model's predictive performance was evaluated using two key metrics:

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n} |y_i - \hat{y}_i| \tag{15}$$

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{16}$$

To improve model convergence and stability, the numeracal input features and target values are that was not binary or categorical was normalized using Min-Max scaling:

$$X' = \frac{X - X_{\text{min}}}{X_{\text{max}} - X_{\text{min}}} \tag{17}$$

$$y' = \frac{y - y_{\text{min}}}{y_{\text{max}} - y_{\text{min}}} \tag{18}$$

where X_{min} and X_{max} represent the minimum and maximum values in the training dataset for each feature, and similarly for y_{min} and y_{max} for the target variable.

After training, the predictions are denormalized using the inverse transformation:

$$\hat{y} = y'(y_{\text{max}} - y_{\text{min}}) + y_{\text{min}} \tag{19}$$

6.1 Neural Network (ReLU)

6.1.1 Model Description

The neural network model is designed to predict the number of orders per department using multiple input features. The model consists of three dense layers, with ReLU activation in the hidden layers and a linear activation function in the output layer. The architecture can be described as follows:

$$\hat{y} = f(W_3 \cdot \max(0, W_2 \cdot \max(0, W_1 X + b_1) + b_2) + b_3) \tag{20}$$

where:

- X represents the input feature vector.
- W_i and b_i are the weight matrices and bias vectors for each layer.
- $\max(0, \cdot)$ denotes the ReLU activation function.
- $f(\cdot)$ is the linear activation function in the output layer.
- $\hat{y}$ is the predicted order volume.

The model is optimized using the Adam optimizer and minimizes the Mean Squared Error (MSE) loss function:

$$\mathcal{L} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{21}$$

where y_i is the true value and $\hat{y}_i$ is the predicted value.

6.1.2 Training and Evaluation

The model was trained using the Adam optimizer with a mean squared error (MSE) loss function. The training configuration included:

- Learning rate: 0.001
- Batch size: 32
- Number of epochs: 50
- 80-20 train-test split

6.1.3 Results

The results for each department are summarized in Table 9.

Department	MAE	RMSE	Avg. Orders	Error Margin (%)
11	3.41	4.58	13.73	24.84
12	5.74	7.55	31.89	18.00
13	6.46	8.36	35.48	18.21
17	5.97	7.74	36.47	16.38
20	4.05	5.22	19.18	21.12
21	5.13	6.72	21.55	23.81
22	4.81	6.22	22.07	21.79
24	5.18	6.48	20.87	24.82
25	4.82	6.11	24.93	19.34
26	5.04	6.66	17.92	28.14
27	4.56	5.94	18.36	24.83
32	6.66	8.51	37.70	17.68
36	5.67	7.46	25.10	22.60

Table 8: Neural Network Prediction Performance per Department with Avg. Orders and Error Margin

6.1.4 Visualization of Predictions

The training process is visualized by plotting the loss function over epochs. Figure 15 shows the training and validation loss for a randomly selected department.

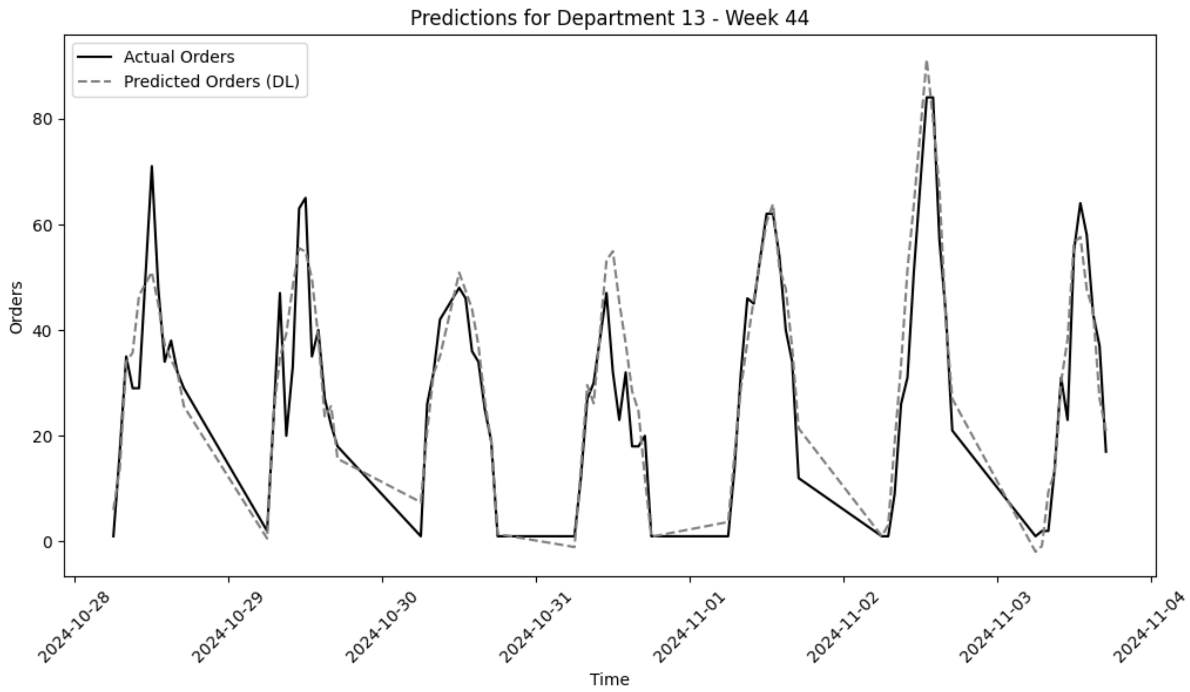

Figure 14: Predictions from Relu

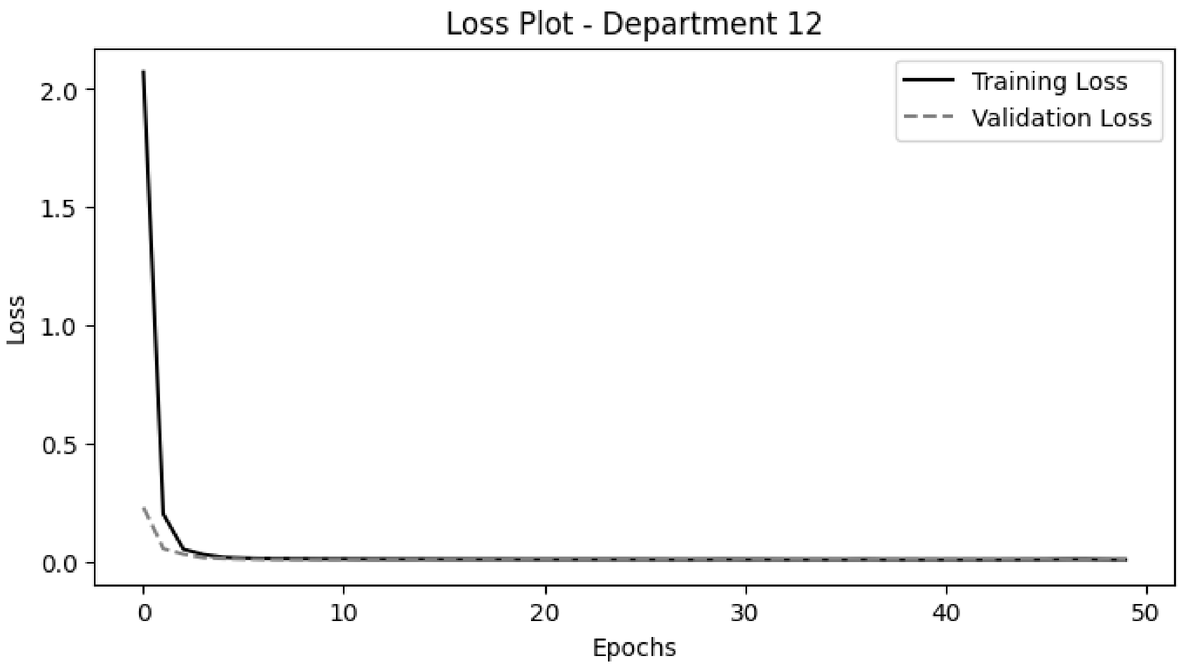

Figure 15: Loss plot for a randomly selected department

The black solid line represents the training loss, while the gray dashed line represents the validation loss. The decreasing trend in loss indicates that the model is learning and generalizing to new data.

6.2 Recurrent Neural Network GRU

6.2.1 Model Description

A Gated Recurrent Unit (GRU) is a type of recurrent neural network (RNN) that is particularly effective for time series forecasting. The GRU architecture simplifies traditional LSTMs by reducing the number of gates while retaining long-term dependencies. The GRU cell is defined by the following equations:

$$z_t = \sigma(W_z x_t + U_z h_{t-1} + b_z) \tag{22}$$

$$r_t = \sigma(W_r x_t + U_r h_{t-1} + b_r) \tag{23}$$

$$\tilde{h}_t = \tanh(W_h x_t + U_h(r_t \odot h_{t-1}) + b_h) \tag{24}$$

$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \tilde{h}_t \tag{25}$$

where z_t is the update gate, r_t is the reset gate, and h_t represents the hidden state. The GRU network is trained using a Huber loss function to handle outliers in the dataset.

6.2.2 Training and Evaluation

The GRU model is trained on historical order data per department. The dataset is split into training (80%) and validation (20%) sets. Sequences of length $L = 10$ are used for input, where each sequence represents historical data over 10 consecutive time steps.

The training process optimizes the weights using the Adam optimizer. Regularization is applied via dropout layers ($p = 0.2$) and L2 kernel regularization to prevent overfitting. Early stopping and learning rate reduction callbacks are used to improve convergence.

6.2.3 Results

The table below presents the prediction performance of the GRU model across different departments, including Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), average order volume, and relative error margin.

Table 9: Neural Network Prediction Performance per Department with Avg. Orders and Error Margin

Department	MAE	RMSE	Avg. Orders	Error Margin (%)
11	4.55	6.12	13.73	33.13
12	7.90	10.43	31.89	24.77
13	9.70	12.75	35.48	27.35
17	8.04	10.85	36.47	22.05
20	5.06	6.51	19.18	26.38
21	6.83	8.73	21.55	31.74
22	6.17	7.80	22.07	27.99
24	5.80	7.70	20.87	27.82
25	6.24	8.34	24.93	25.04
26	6.99	9.22	17.92	39.02
27	5.99	7.75	18.36	32.59
32	9.19	11.79	37.70	24.39
36	8.20	10.65	37.70	21.74

6.2.4 Visualization of Predictions

Figures 16 and 17 illustrate the performance of the GRU model. The first plot shows the predicted vs. actual orders for Department 24 in Week 34, while the second visualizes the loss function during training for Department 27.

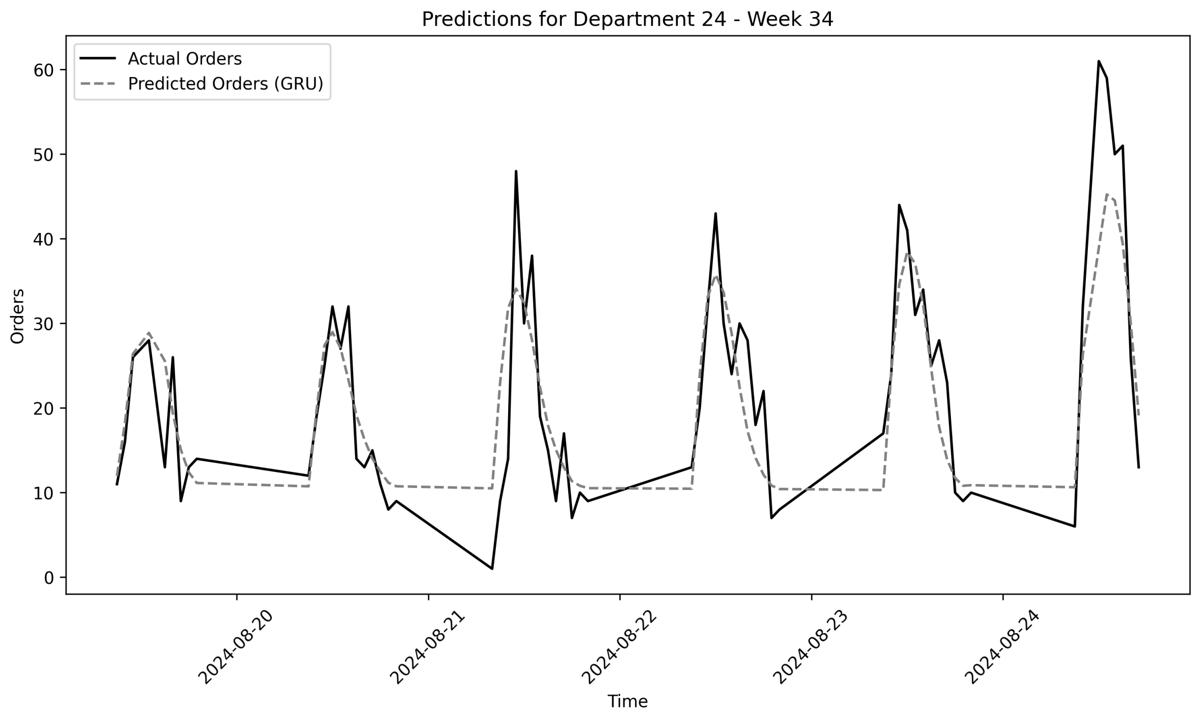

Figure 16: Predicted vs. actual orders for Department 24, Week 34.

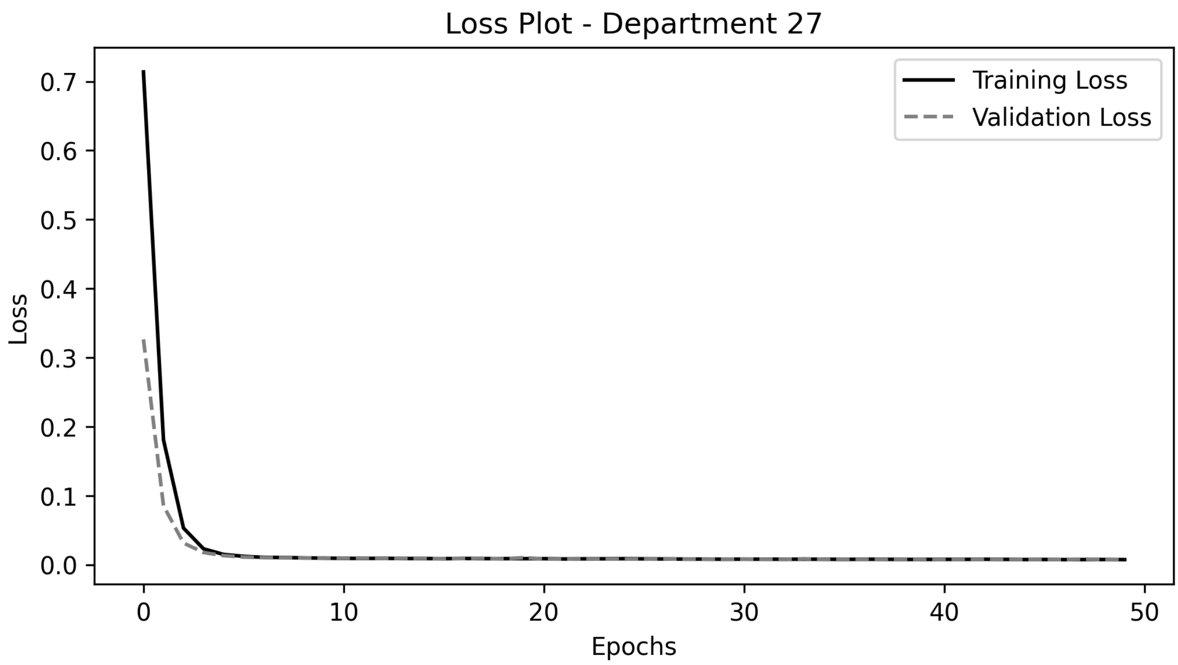

Figure 17: Training and validation loss for Department 27.

6.3 Hybrid Models - (CNN-LSTM)

6.3.1 Model Description

The CNN-LSTM model leverages both convolutional neural networks (CNN) and long short-term memory (LSTM) networks to extract spatial and temporal dependencies in the data. The model architecture is structured as follows:

- A 1D convolutional layer with 64 filters and a kernel size of 3 to extract spatial patterns from input sequences.
- A max-pooling layer with a pool size of 2 to downsample feature maps.
- Two stacked LSTM layers with 50 units each to capture long-range dependencies.
- A fully connected (dense) layer with 32 neurons and ReLU activation.
- An output layer with a single neuron using a linear activation function to predict order volume.

Mathematically, given an input sequence $X = \{x_1, x_2, \ldots, x_T\}$, the CNN extracts feature maps $F = \{f_1, f_2, \ldots, f_T\}$, which are then passed to the LSTM network:

$$h_t = \sigma(W_f F_t + W_h h_{t-1} + b) \tag{26}$$

where h_t is the hidden state at time step t, W_f and W_h are weight matrices, and b is a bias term.

6.3.2 Training and Evaluation

The model was trained using the Adam optimizer with a mean squared error (MSE) loss function. The training configuration included:

- Learning rate: 0.001
- Batch size: 32
- Number of epochs: 50
- 80-20 train-test split

where y_i represents the actual order volume, and $\hat{y}_i$ represents the predicted order volume.

6.3.3 Results

Table 10 presents the performance metrics for each department.

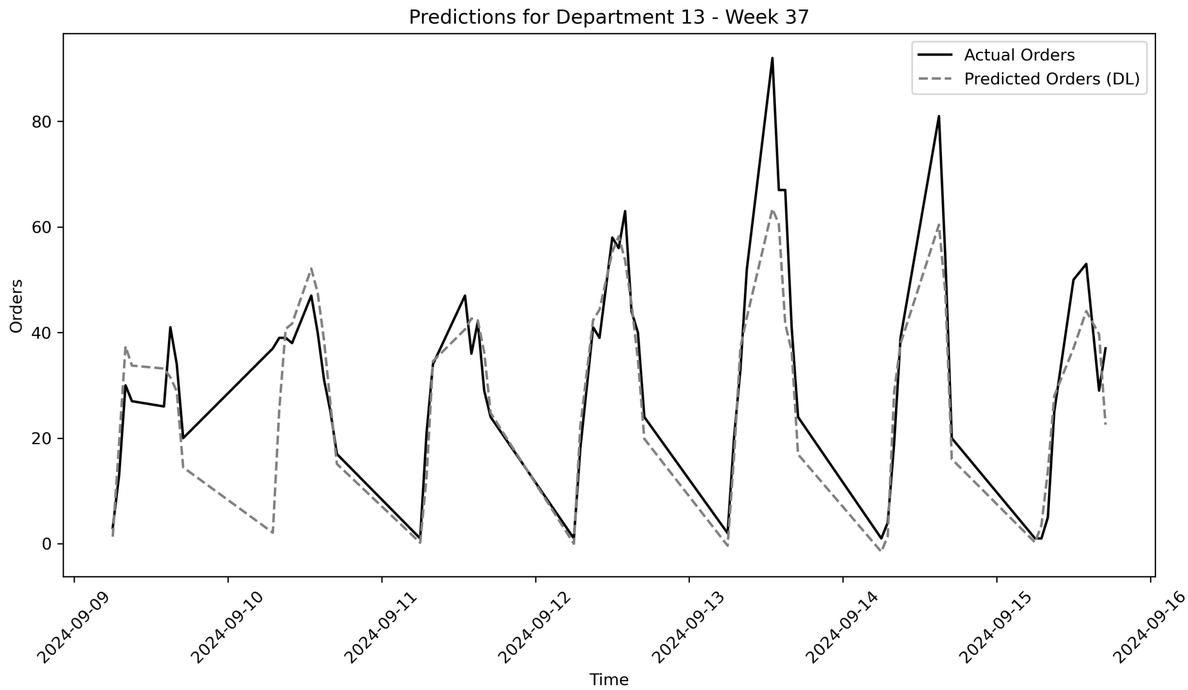

Figure 18: Comparison of actual vs. predicted order volumes for a selected department and week for CNN-LSTM.

Department	MAE	RMSE	Avg.orders	Error Margin
11	3.72	4.96	13.73	27.10%
12	6.15	8.03	31.89	19.29%
13	7.71	10.33	35.48	21.73%
17	6.58	8.57	36.47	18.05%
20	4.32	5.56	19.18	22.52%
21	5.27	7.03	21.55	24.46%
22	5.07	6.64	22.07	22.98%
24	5.27	6.77	20.87	25.24%
25	4.77	6.25	24.93	19.14%
26	5.45	7.12	17.92	30.43%
27	4.67	6.08	18.36	25.44%
32	7.35	9.44	37.70	19.50%
36	6.22	8.05	29.97	20.76%

Table 10: CNN-LSTM model performance across different departments.

6.3.4 Visualization of Predictions

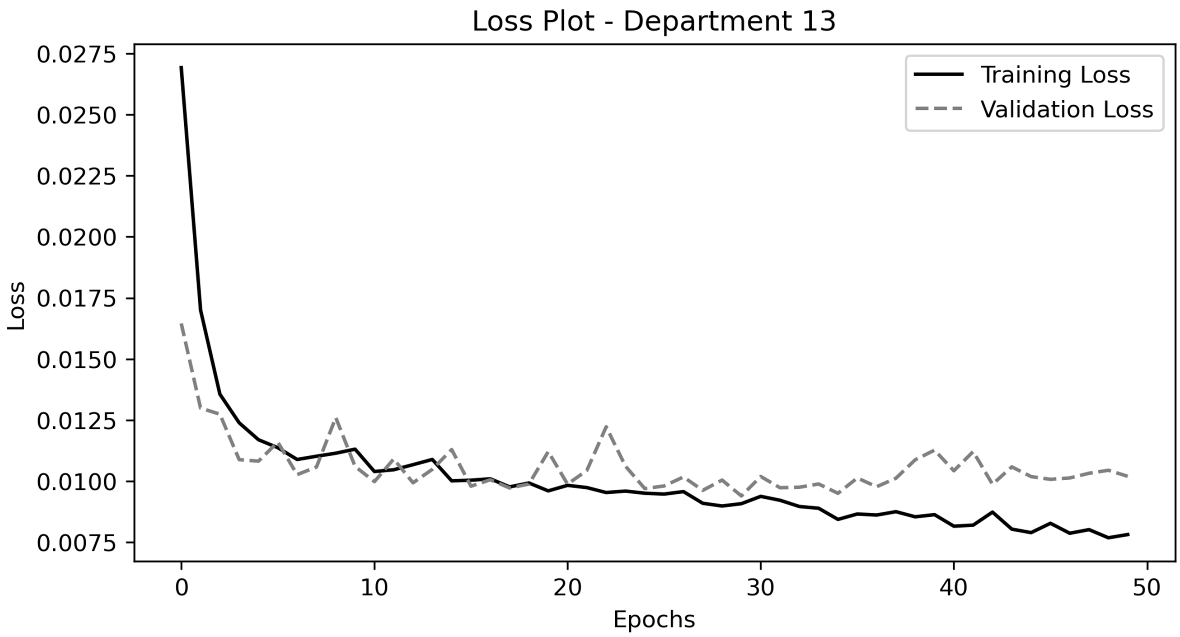

Figure 19: Training and validation loss over epochs for CNN-LSTM.

7 Model Evaluation and Selection

7.1 Performance Metrics

7.1.1 Mean Absolute Error (MAE)

D	KNN	Ridge	Lasso	ARIMA	XGBoost	LightGBM	Relu	CNN-LSTM
11	4.35	3.96	3.96	5.50	3.12	3.09	3.41	3.72
12	7.51	7.00	7.02	10.15	5.56	5.55	5.74	6.15
13	6.76	7.11	7.10	11.15	7.07	6.86	6.46	7.71
17	7.02	7.19	7.18	11.67	6.64	6.41	5.97	6.58
20	5.40	4.82	4.82	7.65	4.66	4.32	4.05	4.32
21	5.98	5.42	5.38	12.56	5.93	5.60	5.13	5.27
22	5.70	5.07	5.07	8.95	5.06	5.04	4.81	5.07
24	5.14	5.07	5.06	11.75	5.57	5.22	5.18	5.27
25	6.38	6.20	6.20	9.26	4.23	4.49	4.82	4.77
26	6.66	5.57	5.56	12.45	4.90	4.92	5.04	5.45
27	5.89	5.56	5.55	7.62	5.29	4.97	4.56	4.67
32	7.88	7.91	7.92	15.82	6.64	6.41	6.66	7.35
36	7.47	7.87	7.86	8.00	6.22	6.05	5.67	6.22
Total	83.15	78.75	78.68	132.53	70.89	68.93	67.50	72.55
Average	6.40	6.06	6.05	10.19	5.45	5.30	5.19	5.58

Table 11: Overall MAE Comparison for All Models by Department

7.1.2 Root Mean Square Error

D	KNN	Ridge	Lasso	ARIMA	XGBoost	LightGBM	Relu	CNN-LSTM
11	5.96	5.22	5.23	7.39	3.91	3.91	4.58	4.96
12	9.77	9.21	9.24	12.44	7.54	7.54	7.55	8.03
13	9.37	9.33	9.33	13.58	8.95	8.95	8.36	10.33
17	9.42	9.65	9.64	14.31	8.21	8.21	7.74	8.57
20	7.06	6.41	6.41	9.37	5.47	5.47	5.22	5.56
21	7.66	7.39	7.37	15.39	7.12	7.12	6.72	7.03
22	7.36	6.72	6.71	11.09	6.37	6.37	6.22	6.64
24	6.73	6.58	6.58	14.74	6.71	6.71	6.48	6.77
25	8.34	8.52	8.52	11.45	5.50	5.50	6.11	6.25
26	8.57	7.70	7.71	16.75	6.21	6.21	6.66	7.12
27	7.62	7.13	7.12	9.77	6.95	6.95	5.94	6.08
32	10.34	10.52	10.53	19.32	8.52	8.52	8.51	9.44
36	9.92	10.56	10.56	10.65	8.05	8.05	7.46	8.05
Total	111.19	104.94	104.95	166.25	89.51	89.51	87.55	94.83
Average	8.55	8.07	8.07	12.79	6.89	6.89	6.73	7.29

Table 12: Overall RMSE Comparison for All Models by Department

7.2 Comparison of Model Performance

In evaluating model performance, we consider the Mean Absolute Error (MAE) and its variability across different departments. The models compared include K-Nearest Neighbors (KNN), Ridge Regression, Lasso Regression, ARIMA, XGBoost, LightGBM, Neural Networks (Relu), and CNN-LSTM.

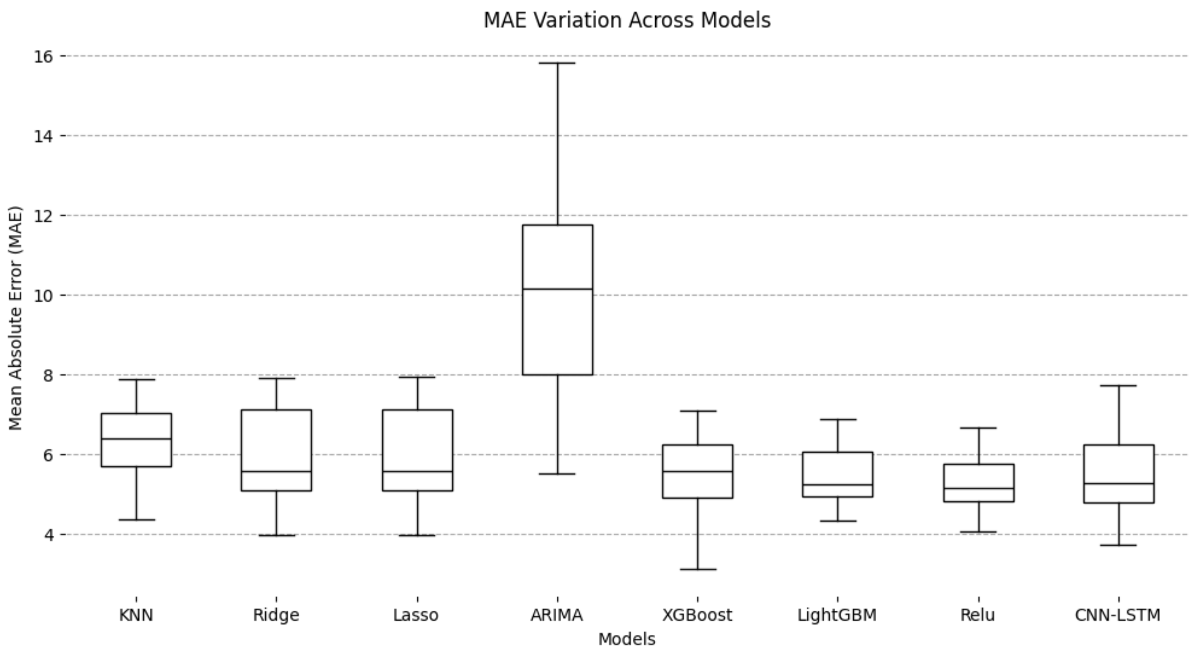

Figure 20: Evaluation of models

To quantify performance, the mean MAE (μ_{MAE}) and standard deviation (σ_{MAE}) are calculated for each model:

$$\mu_{MAE} = \frac{1}{n}\sum_{i=1}^{n} MAE_i \tag{27}$$

$$\sigma_{MAE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(MAE_i - \mu_{MAE})^2} \tag{28}$$

where n represents the number of departments analyzed.

Model	Mean MAE	Std Dev MAE
Neural Network (ReLU)	5.19	0.88
LightGBM	5.30	0.98
XGBoost	5.45	1.05
CNN-LSTM	5.58	1.12
Lasso	6.05	1.20
Ridge	6.06	1.20
KNN	6.32	0.99
ARIMA	10.19	2.63

Table 13: Best Model for Minimizing Variation and MAE. Recommended Model: Neural Network (ReLU).

Additionally, a complexity factor (C) is introduced to account for ease of implementation and scalability, with simpler models receiving a lower complexity weight. The final ranking metric (R) is defined as:

$$R = \mu_{MAE} + \sigma_{MAE} + 0.5C \tag{29}$$

The model with the lowest ranking metric is deemed the most suitable.

Model	Mean MAE	Std Dev MAE	Complexity Factor	Ranking Metric
LightGBM	5.30	0.98	2	7.28
XGBoost	5.45	1.05	2	7.50
Neural Network	5.19	0.88	3	7.57
Lasso	6.05	1.20	1	7.76
Ridge	6.06	1.20	1	7.76
CNN-LSTM	5.58	1.12	3	8.20
KNN	6.32	0.99	2	8.31
ARIMA	10.19	2.63	3	14.33

Table 14: Best Model for Minimizing Variation, MAE, and Implementation Complexity. Recommended Model: LightGBM.

8 Discussion of Results

The results of this study demonstrate the effectiveness of AI-driven forecasting models for workforce planning in retail. Through rigorous evaluation, it is evident that advanced machine learning methods outperform traditional statistical models in predicting order volumes, particularly in terms of accuracy and robustness across different departments.

8.1 Performance Comparison

The comparative evaluation highlights the superiority of gradient boosting techniques, particularly LightGBM, in achieving the lowest error rates across multiple performance metrics. With an average Mean Absolute Error (MAE) of 5.30 and a Root Mean Squared Error (RMSE) of 6.89, LightGBM provides a strong balance between predictive accuracy and computational efficiency. This aligns with prior research indicating that tree-based models excel in structured forecasting tasks where complex interactions exist among features.

Neural networks, especially the CNN-LSTM hybrid model, demonstrated competitive performance, particularly in capturing temporal dependencies. However, their increased training complexity and marginal improvements over boosting models raise considerations about the trade-off between computational cost and accuracy. For real-world deployment, the interpretability and scalability of boosting models offer a distinct advantage.

8.2 Key Insights from Feature Importance Analysis

Feature importance analysis reveals that order history remains the strongest predictor of future orders, reaffirming the necessity of lag variables in time series forecasting. Additionally, staffing levels exhibit a moderate correlation with revenue, but the impact varies by department. This suggests that while workforce allocation plays a crucial role, optimizing staffing must account for other contextual factors, such as store location and customer flow.

Contrary to initial assumptions, external factors such as weather and traffic data contributed minimally to the predictive performance. This finding aligns with empirical observations in the dataset, where fluctuations in customer demand appear to be driven more by historical patterns and operational variables rather than external environmental factors. Nevertheless, these factors could have a non-linear influence that may not be fully captured through traditional feature importance rankings, warranting further exploration through alternative modeling approaches.

8.3 Implications for Workforce Planning

The insights from this study provide a clear roadmap for improving workforce scheduling strategies. By leveraging predictive models, businesses can optimize staffing levels to better align with demand fluctuations, reducing labor costs while maintaining service quality. The deployment of an automated forecasting pipeline using LightGBM ensures that predictions remain updated with the latest transactional data, allowing for dynamic adjustments in staffing decisions.

However, effective implementation requires managerial oversight to account for operational contingencies not captured by the model, such as unexpected promotional events or changes in customer behavior due to macroeconomic shifts. Future iterations of the model could integrate additional business-specific factors, such as marketing activities or competitor dynamics, to enhance adaptability.

8.4 Model Limitations and Future Research

This study initially considered external factors such as weather conditions and traffic patterns for revenue forecasting. However, their feature importance was found to be significantly low, leading to their exclusion from the final implementation to maintain an efficient and interpretable model. This suggests that these factors may not be strong predictors of demand fluctuations in this food service context. Prior research, such as Döring et al. (2024), has shown that integrating broader macroeconomic indicators (e.g., national GDP growth, consumer confidence indices) can enhance forecasting models. Future studies could explore whether such variables offer stronger predictive power for a café and bakery business model.

Second, while the selected models performed well overall, individual department-level variations indicate that a single modeling approach may not be optimal for all locations. Future work could explore ensemble methods that dynamically adjust model selection based on department-specific characteristics.

Lastly, given the importance of interpretability in workforce planning, integrating explainable AI (XAI) techniques could enhance managerial trust in model outputs. Methods such as SHAP (Shapley Additive Explanations) could provide deeper insights into how individual factors influence predictions, fostering more informed decision-making.

8.5 Conclusion

In summary, this study confirms the effectiveness of AI-based forecasting for optimizing workforce planning in retail. The results emphasize the value of machine learning models in improving demand prediction accuracy, with LightGBM emerging as the preferred model for deployment due to its superior performance and scalability. While challenges remain in fully capturing all demand drivers, continued advancements in AI and data integration hold promise for further refining predictive capabilities. Future research should focus on expanding data sources, refining model selection strategies, and enhancing interpretability to maximize the impact of AI-driven forecasting in operational decision-making.

9 Implementation and Deployment

9.1 Data Ingestion and Processing

For our implementation example, we manually uploaded an Excel file containing order data for the year 2024, using 2023 as lagged parameters. However, in a real-world production environment, data ingestion would be managed through a structured pipeline within a data warehouse or data lake, ensuring standardized and automated data updates. Incremental data refreshes would maintain an up-to-date historical dataset for model training and forecasting.

9.2 Using PySpark for Big Data Processing

Given the IT architecture and cost considerations, Azure ML Notebook is the recommended environment for scalable machine learning workflows. It provides a flexible and cost-effective approach for running distributed computations. However, to illustrate the workflow, we implemented a PySpark notebook within Microsoft Fabric. PySpark, the distributed computing framework for Apache Spark, is well-suited for handling large-scale datasets.

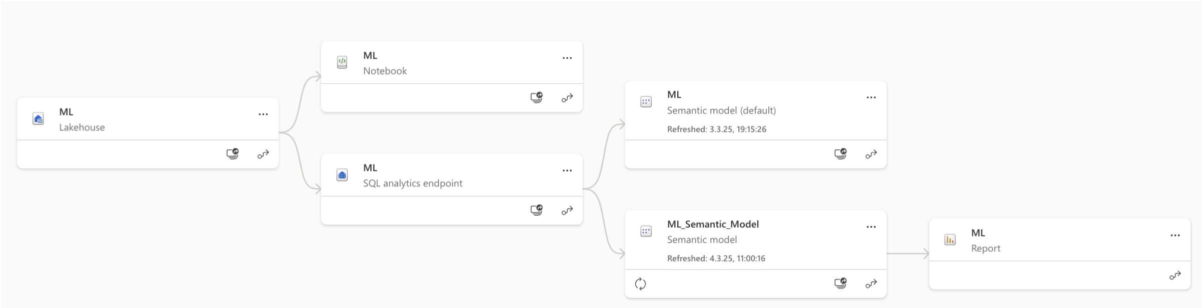

Figure 21: Fabric Lakehouse and Power BI Integration

9.3 Scheduling and Model Execution

To automate the process, the PySpark notebook is scheduled to run once per week. This scheduling ensures that the staffing plan for the upcoming week is continuously updated based on the latest internal order trends.

For simplicity, only internal historical data was used in our implementation. Although external factors such as weather conditions and traffic patterns were considered, their feature importance was significantly low in the modeling process. As a result, they were excluded from the final implementation to maintain an efficient and interpretable forecasting model.

9.4 Deployment and Future Considerations

The predicted orders are then stored in a Power BI semantic model, enabling real-time visualization and decision-making for staffing adjustments.

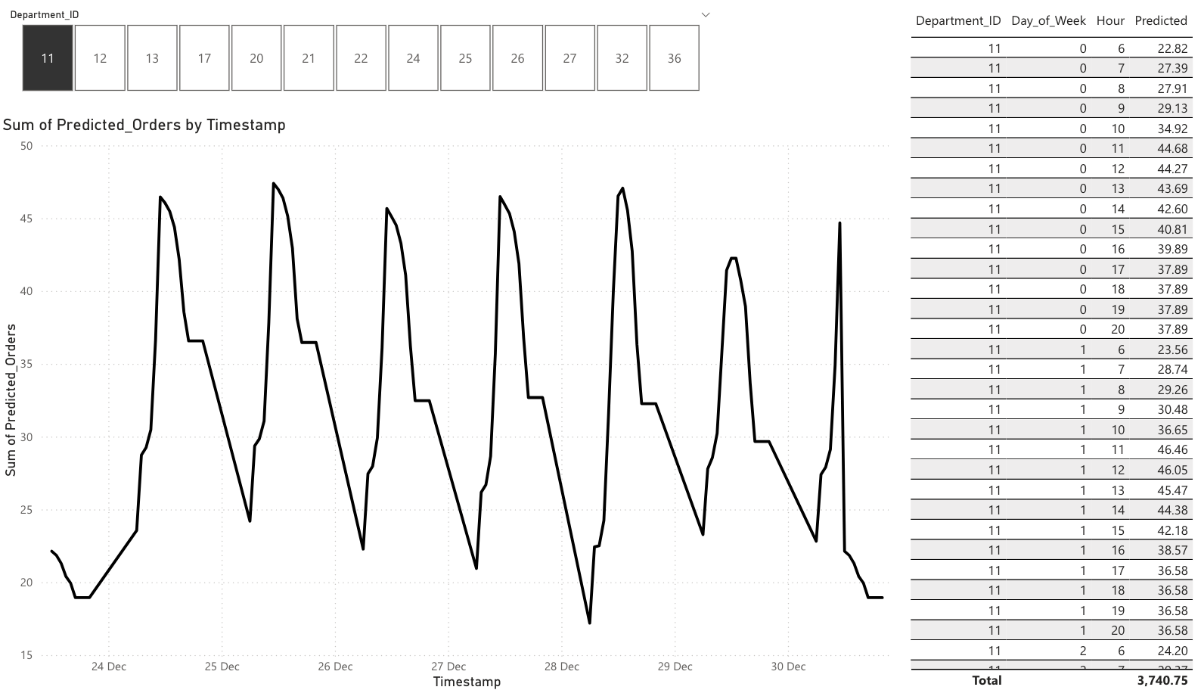

Figure 22: Power BI Visualization of Predicted Orders

This setup ensures that Godt Brød has a scalable, automated, and data-driven approach to weekly staffing optimization based on predicted order volumes.

References

Shehzadi, T. (2025). *Time Series Analysis with Machine Learning: A Comprehensive Review and Future Directions*. ResearchGate.
Döring, L., Grumbach, F., Reusch, P. (2024). *Optimizing sales forecasts through automated integration of market indicators.* arXiv.