AI-Driven Transaction Prediction for Workforce Planning in Retail Businesses

STR459: Artificial Intelligence and Robotics

Candidate numbers: 4, 6, 27, 49, 83

Handed out:	February 25, 2025
Deadline:	April 8, 2025

Number of pages:	30
Number of words:	4880

Do you allow your report to be used as an example in the future: **No**

Contents

1 Introduction **1**

2 Data Analysis **2**

2.1 Feature Engineering . . . 2
2.1.1 Raw Data . . . 2
2.1.2 Lag Variables and Historical Trends . . . 3
2.1.3 External Historical Data . . . 3
2.1.4 External Predictive Data . . . 4
2.1.5 Final Merge . . . 5

2.2 EDA . . . 5
2.2.1 Revenue, Orders, and Staffing Trends . . . 6
2.2.2 Correlation Analysis and External Influence . . . 7

3 Building AI Models **8**

3.1 Libraries and Tools Overview . . . 8

3.2 Classification Models . . . 8
3.2.1 Regression Analysis . . . 8
3.2.2 KNN . . . 11
3.2.3 Regularized Regression (Lasso, Ridge) . . . 13

3.3 Building Basic Models . . . 16
3.3.1 Random Forest Model for Order Prediction . . . 16
3.3.2 Boosting Models: XGBoost and LightGBM . . . 18

3.4 Building Deep Learning Models . . . 20
3.4.1 Neural Network (ReLU) . . . 20
3.4.2 Hybrid Models - (CNN-LSTM) . . . 23

4 Evaluating AI Models **25**

4.1 Performance Metrics . . . 25
4.1.1 Mean Absolute Error (MAE) . . . 25
4.1.2 Root Mean Squared Error (RMSE) . . . 25

4.2 Comparison of Model Performance . . . 25

5 Summary of Findings **27**

5.1 Discussion of Results . . . 27
5.1.1 Performance Comparison . . . 27
5.1.2 Key Insights from Feature Importance Analysis . . . 28
5.1.3 Conclusion . . . 28
5.1.4 Model Limitations and Future Research . . . 28

5.2 Implementation and Deployment . . . 28

List of Figures

1 Revenue, orders, and staffing trends. . . . 6
2 Correlation Analysis of Numerical Variables . . . 7
3 R-squared per Variable for Revenue Prediction . . . 9
4 Interaction Coefficients by Department (p-value <0.05 Highlighted). . . . 11
5 KNN Cross-Validation scores . . . 12
6 KNN Prediction Performance per Department with Fixed k = 8, Including Avg. Orders and Error Margin . . . 13
7 Comparison of Ridge and Lasso regression models . . . 14
8 Overall feature importance for Ridge regression model . . . 15
9 Overall feature importance for LASSO model . . . 15
10 Random Forest predictions vs. actual orders for department 17, week 11 . . . 17
11 Overall feature importance across all departments . . . 17
12 Predictions vs. actual orders for XGBoost and LightGBM models. Department 17, week 11 . . . 19
13 Feature importance analysis for XGBoost model . . . 19
14 Feature importance analysis for LightGBM model . . . 20
15 ReLu model predictions for department 13, week 37 . . . 22
16 Loss plot for the randomly selected department 13 . . . 22
17 Predicted vs. actual order volumes for department 13, week 37 . . . 24
18 Training and validation loss over epochs for CNN-LSTM . . . 24
19 MAE variation across models . . . 26
20 Fabric Lakehouse and Power BI Integration . . . 29
21 Power BI Visualization of Predicted Orders . . . 30

List of Tables

1 Regression results per Department ID Revenue . . . 10
2 Regression results per Department ID Orders . . . 10
3 KNN Prediction Performance per Department with Fixed $k = 8$, Including Avg. Orders and Error Margin . . . 12
4 Performance of Ridge and Lasso models across departments with individual error margins . . . 14
5 Random Forest Prediction Performance per Department . . . 16
6 Model performance across departments with average orders and error margins . . . 18
7 Neural Network Prediction Performance per Department with Avg. Orders and Error Margin . . . 21
8 CNN-LSTM model performance across different departments. . . . 23
9 Overall MAE Comparison for All Models by Department . . . 25
10 Overall RMSE Comparison for All Models by Department . . . 25
11 Model Performance Analysis across n departments. . . . 26
12 Model Performance Analysis including Complexity. . . . 27

1 Introduction

Predicting demand assists businesses in optimizing staffing, reducing overstaffing, and shortages. In recent years, tools such as machine learning have significantly improved time series forecasting, enhancing predictive accuracy in dynamic environments (Shehzadi, 2025). These methods have been widely used across industries to optimize resource allocation and operational efficiency. For instance, machine learning has been applied in workforce planning for call centers, where it outperforms traditional models by leveraging real-time data and predictive features (Meller et al., 2020). Similarly, it has been used in food delivery logistics to predict demand and efficiently allocate staff (Crivellari et al., 2023).

Building on these advancements, this report explores how machine learning can be applied to Godt Brød, a Norwegian bakery chain, to enhance operational efficiency through order forecasting and subsequent workforce planning. A comprehensive dataset was assembled, incorporating internal transaction records and staffing schedules along with external factors such as weather forecasts, cruise ship activity, and traffic patterns. In compliance with the General Data Protection Regulation, GDPR, all personal data has been anonymized where employees have been assigned random identifiers, ensuring that no personally identifiable information is used beyond the staff-related data essential for this analysis.

Since the dataset contains labeled outcomes, supervised learning techniques were applied, 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) and k-nearest neighbors (KNN). 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 and hybrid architectures (CNN-LSTM) 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) and Mean Absolute Error (MAE). The best-performing model was refined and integrated into a workforce optimization plan, ensuring alignment between staffing and projected demand. While the model provides a data-driven foundation, managerial judgement remains crucial for real-world implementation. 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 Analysis

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 from 2023 to 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 as:

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

   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 and Historical Trends

Predictive modeling requires historical data to identify trends and seasonality. The following lag variables were 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

The following features were derived from historical traffic data obtained from Statens vegvesen, covering key high-traffic junctions for inbound and outbound travel in Bergen and 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

Historical variables with predictive potential up to seven days in advance were integrated into the dataset.

Daily records from *Kystverket (n.d.)* track ship arrivals and departures at Bergen and Oslo ports, serving as an indicator of tourist volume fluctuations. The dataset was standardized, and dates were reconstructed as:

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

Arrivals and departures were linked by ship name and date, and each ship's port duration was computed as:

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

To quantify daily port traffic, the total number of docked ships per city and day was calculated as:

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

where $I_{s,t} = 1$ if ship s was in port on day t. The processed dataset was merged with primary data to assess its impact on customer volume.

Additionally, meteorological data from *Norsk Klimaservicesenter*, including temperature, precipitation, rainfall duration, and sunshine, was standardized and integrated for Bergen and Oslo.

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 with different explanatory variables.

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

$$\begin{aligned}
&\text{Integer: } \ \text{Department ID, Year, Month, Day, Hour, Ships in Port}\\
&\text{Floating Point: } \ \text{Revenue, Traffic Volume, Percentage Changes}\\
&\text{Categorical: } \ \text{City}\\
&\text{One-hot encoded: } \ \text{Weekday}
\end{aligned}$$

4. *Feature Normalization:* Except for in the EDA and linear regression, all non-binary numerical features were min-max scaled within departments:

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

   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.

2.2 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.

2.2.1 Revenue, Orders, and Staffing Trends

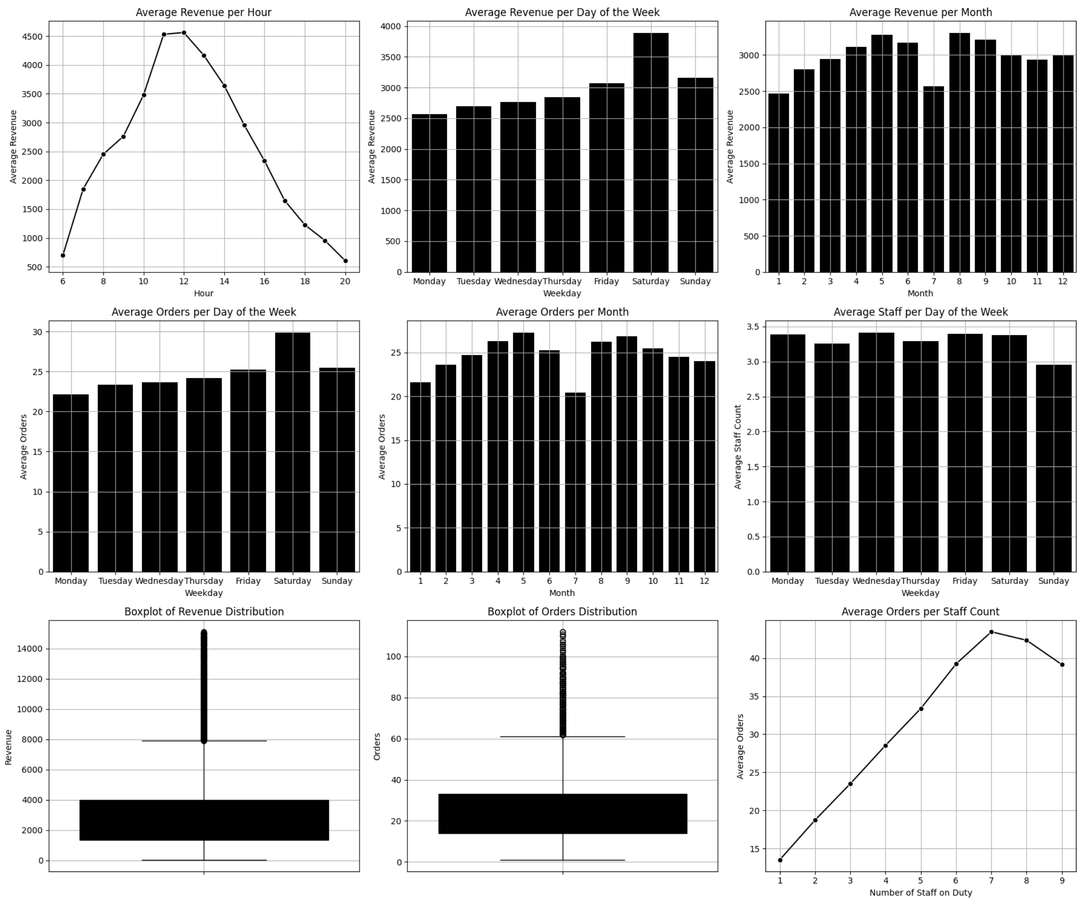


Figure 1: Revenue, orders, and staffing trends.

Revenue follows a predictable daily pattern, peaking at midday. Weekly trends reveal that revenue peaks on Saturdays and reaches its lowest point on Mondays. Monthly revenue fluctuations suggest midyear peaks, likely due to seasonal demand. Order volumes closely align with revenue trends, whereas staffing levels remain steady, with a slight increase 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.

2.2.2 Correlation Analysis and External Influence

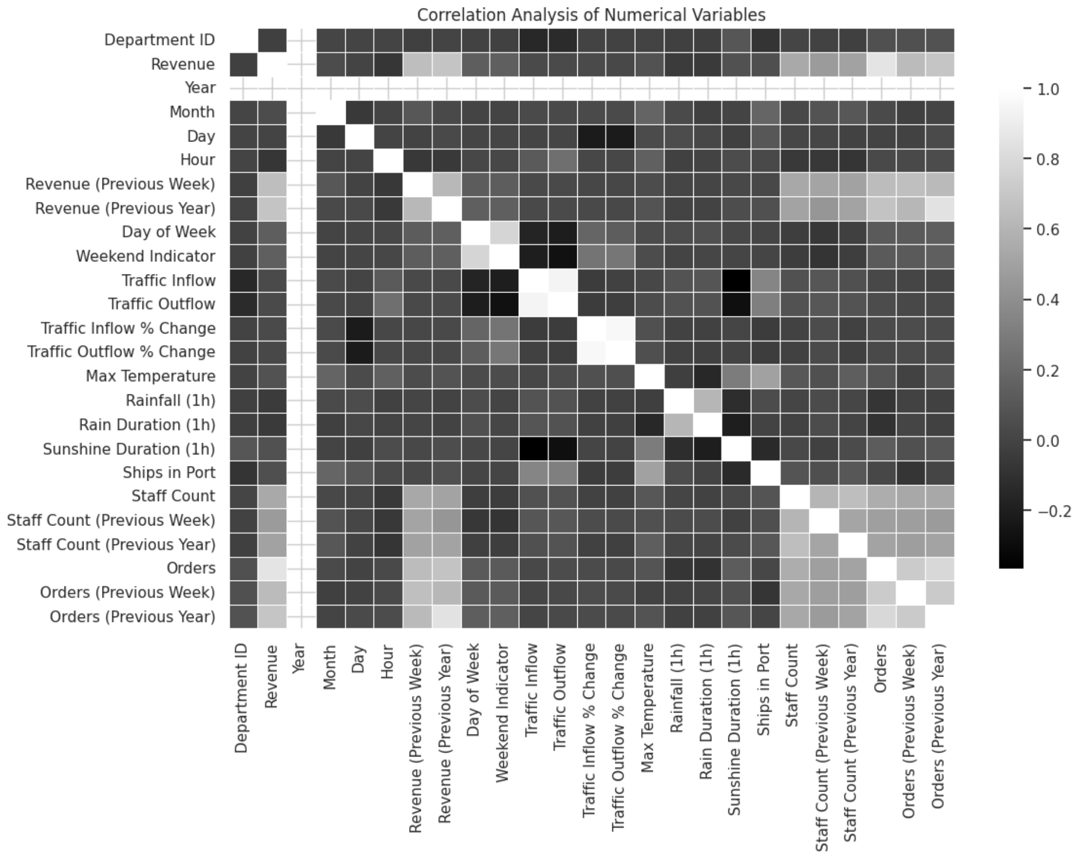


Figure 2: Correlation Analysis of Numerical Variables

The correlation analysis confirms that revenue is primarily driven by order volume. Staffing levels exhibit a moderate correlation with revenue, suggesting that while workforce allocation influences business performance, it is not the sole determining factor. Traffic inflow demonstrates a weak but positive correlation with revenue, indicating a limited impact on sales. Similarly, weather conditions, including temperature and rainfall, show minimal correlation, suggesting that external climatic factors have a negligible effect on business operations.

3 Building AI Models

To ensure a robust framework for model training and evaluation, the dataset was divided into an 80% training set and a 20% test set. The model performance was assessed using MAE and RMSE, which measure the accuracy of predictions by quantifying the average difference between actual and predicted values:

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

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

3.1 Libraries and Tools Overview

The project processed data by using *pandas* and *numpy*, with *os* managing file paths dynamically. Visualization was handled by *matplotlib* and *seaborn*, supporting exploratory analysis, while *statsmodels* facilitated statistical modeling, particularly regression. Machine learning models, including *KNeighborsRegressor*, *Ridge*, *Lasso*, and *RandomForestRegressor*, were implemented with *scikit-learn*, which also provides tools for data splitting, normalization, and validation. Predictive performance was further enhanced by boosting algorithms like *XGBoost* and *LightGBM*. Finally, deep learning models built with *TensorFlow* leveraged neural networks for improved predictions.

3.2 Classification Models

3.2.1 Regression Analysis

Model Description

To better understand the explanatory power of the variables in our dataset, we applied linear regression. This method models the relationship between a dependent variable and one or more independent variables by minimizing the sum of squared residuals (James et al., 2021, p. 61). Specifically, we used regression analysis to examine how revenue, orders, and staff count interact across departments, employing the following model:

$$Y_i = \beta_0 + \sum_{j=1}^{n}\beta_j X_{ji} + \epsilon_i \tag{3}$$

where Y_i represents the dependent variable (*Revenue* or *Orders*).

The explanatory power of these regressions varies across departments, as shown in Tables 1 and 2. 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 extended the model by including an interaction term:

$$Revenue_i = \beta_0 + \beta_1 \cdot Staff_i + \beta_2 \cdot Orders_i + \beta_3 \cdot (Staff_i \times Orders_i) + \epsilon_i \quad (4)$$

Here, β_3 captures whether additional staff enhance revenue at higher order volumes ($\beta_3 > 0$) or reduce efficiency when orders are high ($\beta_3 < 0$). This interaction model helps determine if the relationship between staff count and revenue remains consistent across all order levels, or if staffing decisions should be dynamically adjusted based on order fluctuations.

Revenue Prediction

To assess the factors influencing revenue, we examined each variable independently, evaluating its explanatory power. The results are presented in Figure 3 and indicate that orders ($R^2 \approx 0.73$) is the strongest predictor of revenue, aligning with expectations. Historical revenue and order trends also demonstrate relative high 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. This highlights the role of workforce allocation in optimizing revenue.

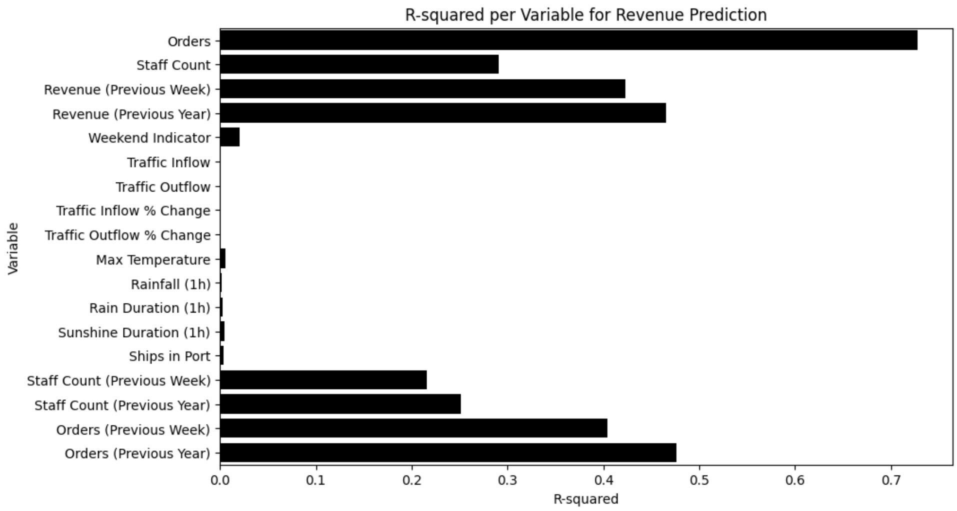


Figure 3: R-squared per Variable for Revenue Prediction

Staff Count as an Explanatory Variable for Revenue and Orders

When examining staff count as a predictor of revenue, the explanatory power varies across departments. Moderate R^2 values suggest staffing influences revenue, while lower values indicate inefficiencies or minimal impact. A data-driven staffing approach could improve performance. 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 reports the regression results for the relationship between staff count and orders, revealing substantial variation in explanatory power.

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

Interaction Analysis

To assess how staff count influences revenue under varying order volumes, we introduced an interaction term in the regression model. This analysis determines whether staff additions have a stronger impact when order volumes are high or low. The results, illustrated in Figure 4, reveal substantial variation across departments. A higher interaction coefficient suggests that additional staff boosts revenue when order volumes are high, while a lower one may indicate diminishing efficiency.

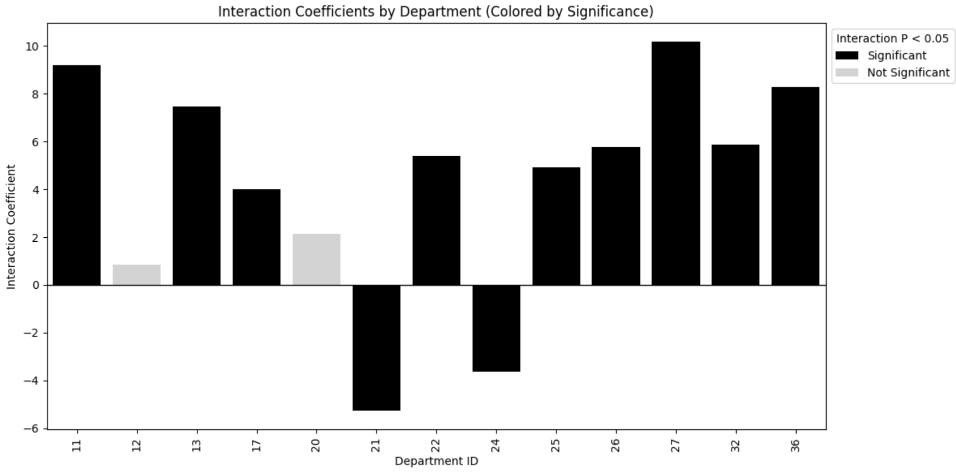


Figure 4: Interaction Coefficients by Department (p-value <0.05 Highlighted).

The analysis highlights key insights into staff count and revenue relationships across departments. Departments 27, 11, 36 and 13 benefit from increased staffing during high order volumes, supporting a flexible workforce strategy. Departments 26, 22, 25, 32 and 17 also show positive effects, though overstaffing in low-demand periods may be inefficient. Conversely, departments 24 and 21 experience reduced efficiency with additional staff at high volumes. Departments 12 and 20 show no significant effects, suggesting staffing changes do not impact revenue. These variations emphasize the importance of tailoring staffing decisions to individual department characteristics.

3.2.2 KNN

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 (James et al., 2021, p.105). 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:

$$\hat{Y}(x^*) = \frac{1}{k} \sum_{i \in \mathcal{N}_k(x^*)} Y_i, \tag{5}$$

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

$$d(X_i, X_j) = \sqrt{\sum_{m=1}^{M} (X_{im} - X_{jm})^2}. \tag{6}$$

Training and Evaluation

The model is trained using a fixed $k = 8$ for all departments. This value is chosen based on cross-validation scores for one k for all departments.

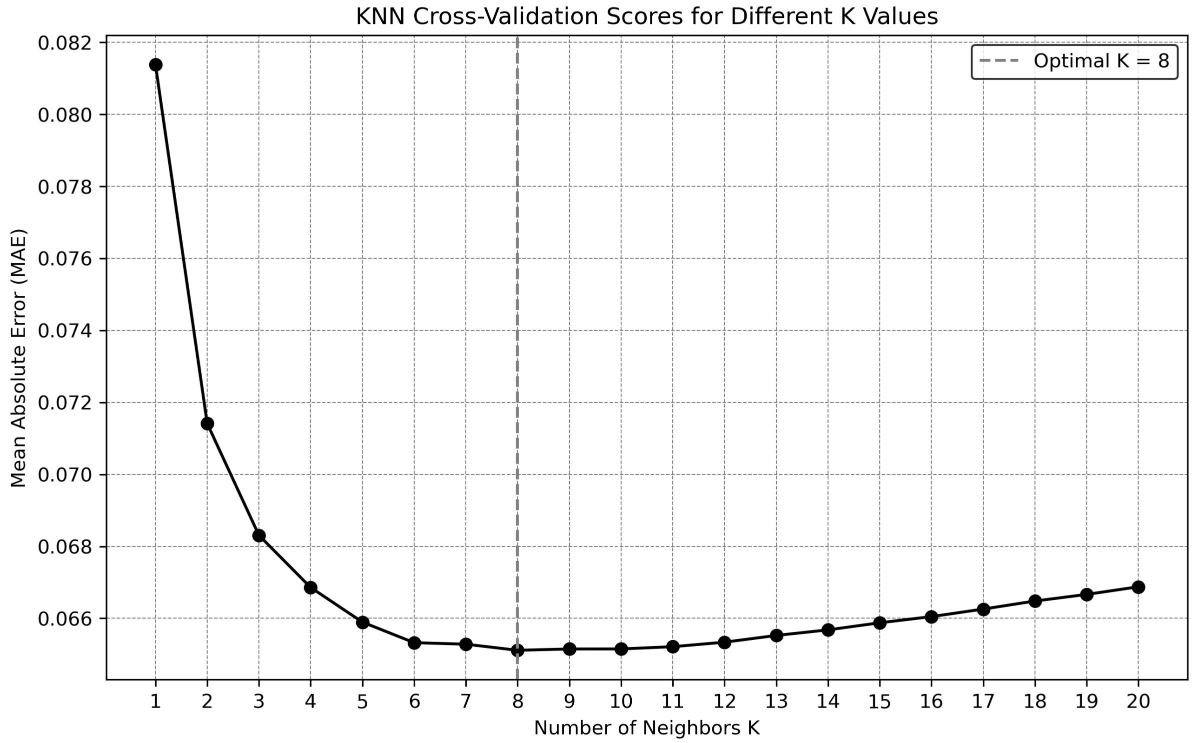


Figure 5: KNN Cross-Validation scores

Department	Fixed K	MAE	RMSE	Avg. Orders	Error Margin (%)
11	8	4.68	6.34	13.73	34.1
12	8	7.62	10.10	31.89	23.9
13	8	6.76	9.54	35.48	19.1
17	8	7.21	9.59	36.47	19.8
20	8	4.92	6.46	19.18	25.6
21	8	6.06	7.98	21.55	28.1
22	8	5.41	6.98	22.07	24.5
24	8	5.08	6.88	20.87	24.3
25	8	6.22	8.45	24.93	24.9
26	8	5.98	8.08	17.92	33.4
27	8	5.58	7.40	18.36	30.4
32	8	7.92	10.57	37.70	21.0
36	8	8.20	11.05	29.97	27.4

Table 3: KNN Prediction Performance per Department with Fixed $k = 8$, Including Avg. Orders and Error Margin

Results

The results show that with a fixed $k = 8$, the model performs consistently across different departments, achieving reasonable MAE and RMSE values.

The figure below illustrates actual vs. predicted orders for Department 13 in Week 37.

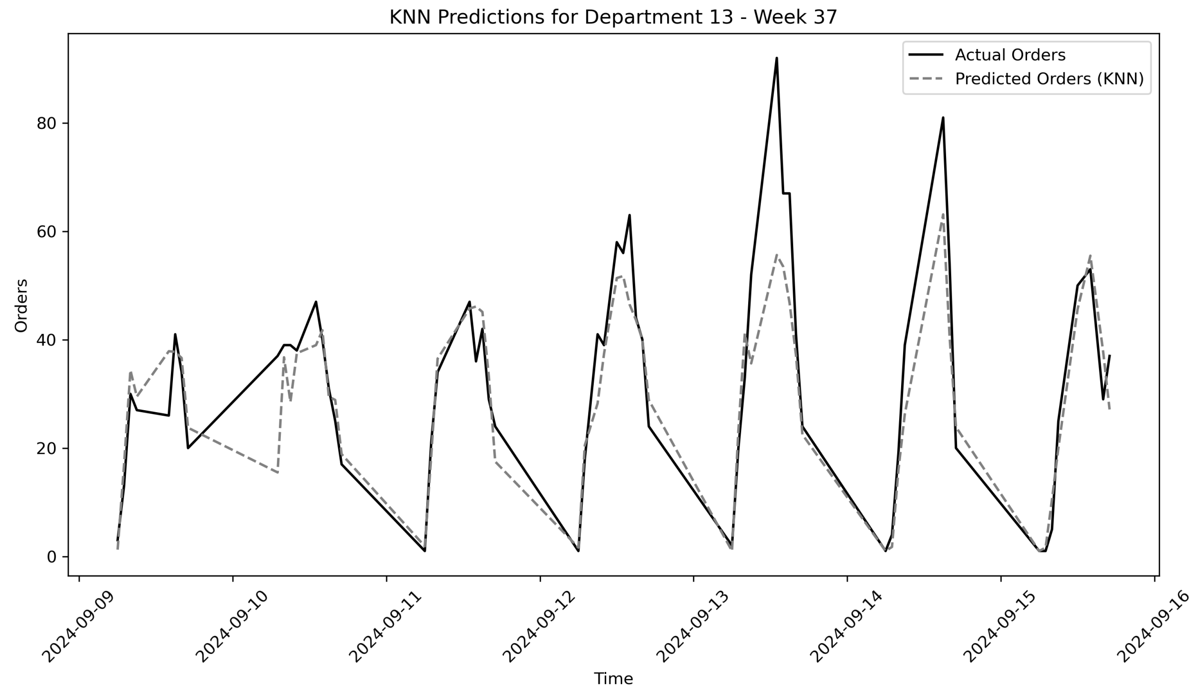


Figure 6: KNN Prediction Performance per Department with Fixed k = 8, Including Avg. Orders and Error Margin

3.2.3 Regularized Regression (Lasso, Ridge)

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. Ridge and Lasso regression mitigate multicollinearity by reducing variance in correlated predictors, enhancing model stability over ordinary least squares, as demonstrated by Yang and Wen (2018).

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.

Results

Department	Ridge MAE / RMSE	Lasso MAE / RMSE	Avg. Orders	Ridge Error Margin (%)	Lasso Error Margin (%)
11	4.85 / 6.28	4.76 / 6.20	13.73	35.3%	34.7%
12	7.36 / 9.39	7.31 / 9.35	31.89	23.1%	22.9%
13	8.00 / 10.23	7.86 / 10.10	35.48	22.5%	22.2%
17	7.22 / 9.53	7.20 / 9.51	36.47	19.8%	19.7%
20	5.24 / 6.90	5.20 / 6.86	19.18	27.3%	27.1%
21	6.12 / 7.92	6.09 / 7.88	21.55	28.4%	28.3%
22	5.52 / 7.00	5.49 / 6.96	22.07	25.0%	24.9%
24	5.37 / 6.99	5.33 / 6.94	20.87	25.7%	25.5%
25	6.50 / 8.62	6.47 / 8.58	24.93	26.1%	25.9%
26	6.59 / 8.59	6.50 / 8.51	17.92	36.8%	36.3%
27	6.06 / 7.80	6.01 / 7.75	18.36	33.0%	32.7%
32	8.20 / 10.55	8.13 / 10.48	37.70	21.8%	21.6%
36	9.16 / 11.61	9.03 / 11.49	29.97	30.6%	30.1%

Table 4: Performance of Ridge and Lasso models across departments with individual error margins

The Lasso and Ridge models exhibited similar performance across departments, with minor variations in MAE and RMSE. Lasso generally showed slightly lower RMSE values, indicating better generalization.

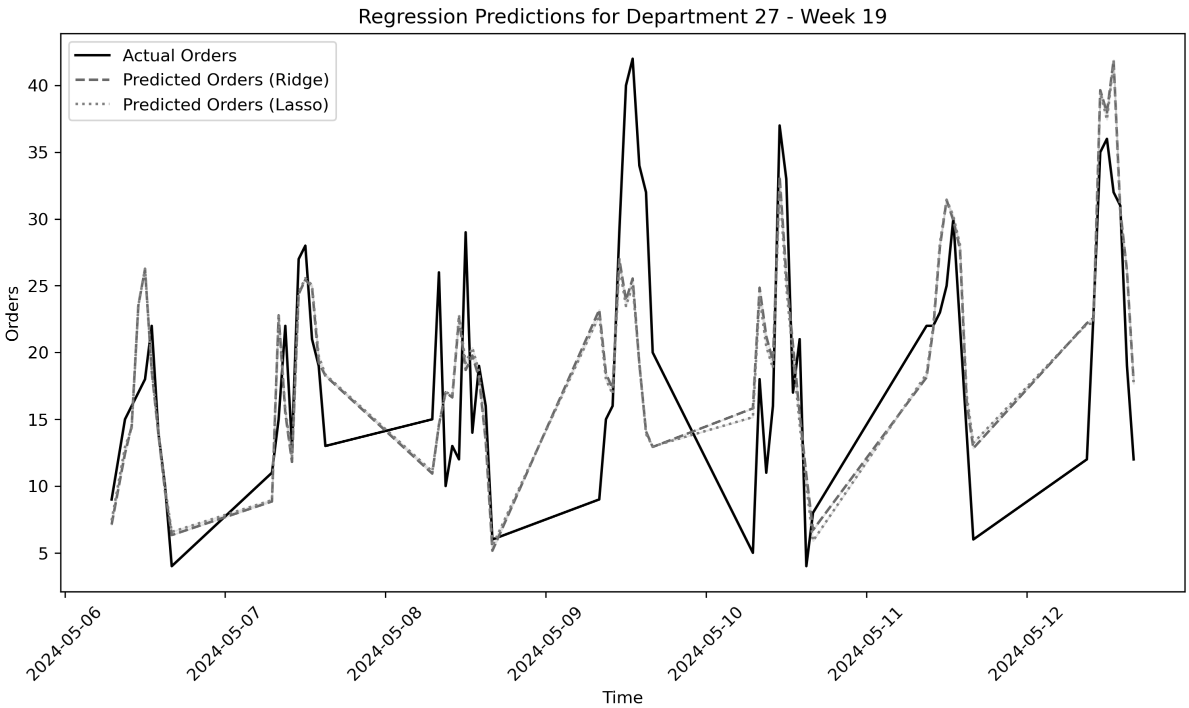


Figure 7: Comparison of Ridge and Lasso regression models

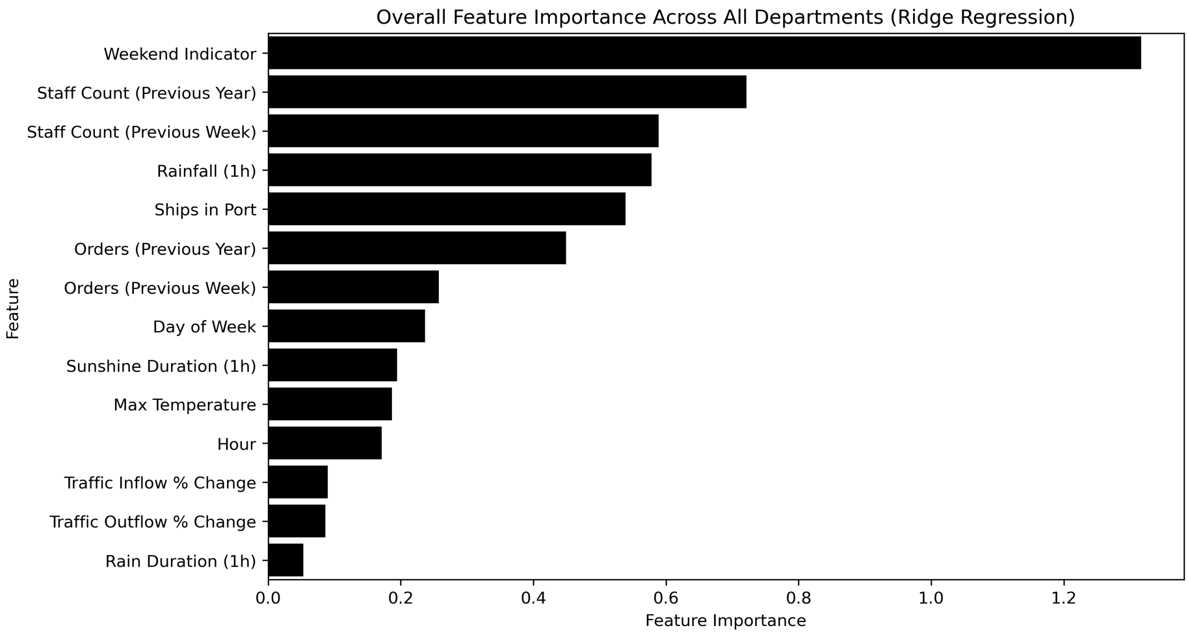


Figure 8: Overall feature importance for Ridge regression model

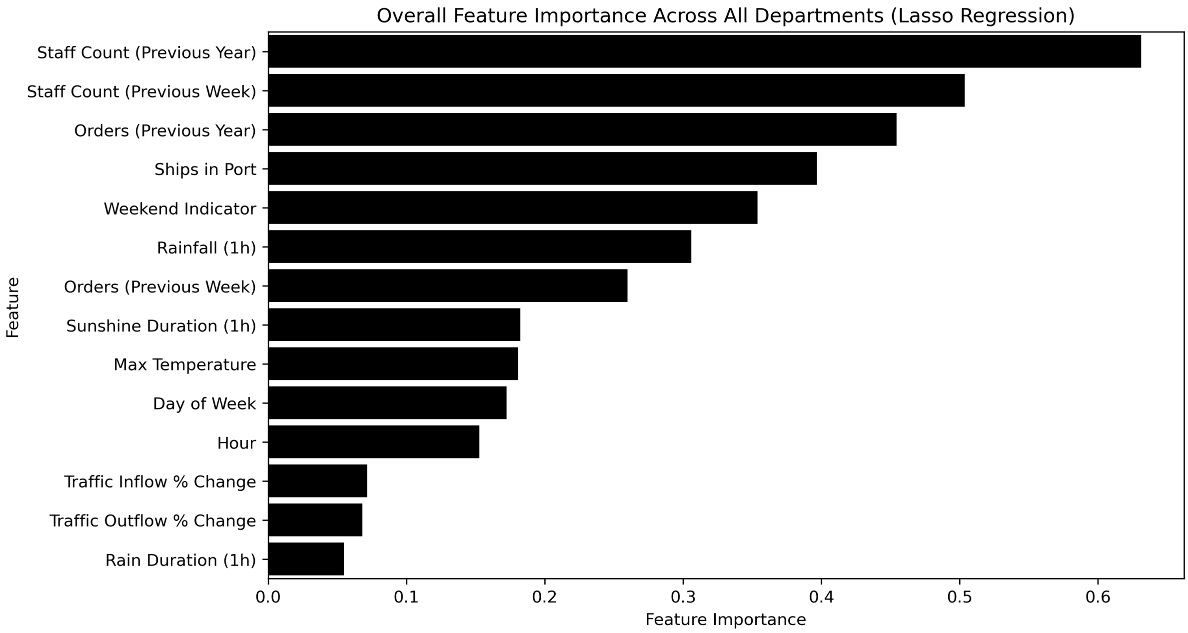


Figure 9: Overall feature importance for LASSO model

The analysis indicates that both models effectively captured overall trends, though residual patterns revealed some variation across departments.

3.3 Building Basic Models

3.3.1 Random Forest Model for Order Prediction

Model Overview

To forecast hourly order volume for each department, we employed a Random Forest Regressor, an ensemble learning approach leveraging multiple decision trees. This method has been widely used in staffing prediction models and has demonstrated its effectiveness in optimizing workforce allocation (Meller et al., 2020). The model aggregates predictions from individual trees, formulated as:

$$\hat{y} = \frac{1}{N} \sum_{i=1}^{N} T_i(X) \tag{9}$$

where $T_i(X)$ represents the output of each decision tree, and N denotes the total number of trees in the ensemble. Random Forest reduces variance by averaging multiple independent trees, improving stability and predictive accuracy compared to a single decision tree (James et al., 2021, p. 344.).

Results

Table 5 presents the MAE and RMSE for each department.

Table 5: Random Forest Prediction Performance per Department

Department	MAE	RMSE	Avg. Orders	Error Margin (%)
11	4.83	6.65	13.73	35.2%
12	7.00	9.16	31.89	21.9%
13	6.65	9.26	35.48	18.8%
17	7.17	9.61	36.47	19.7%
20	5.00	6.59	19.18	26.1%
21	6.16	8.08	21.55	28.6%
22	5.49	7.15	22.07	24.9%
24	5.07	6.74	20.87	24.3%
25	6.33	8.49	24.93	25.4%
26	6.46	8.49	17.92	36.0%
27	5.70	7.54	18.36	31.0%
32	7.99	10.42	37.70	21.2%
36	7.80	10.39	29.97	26.0%

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

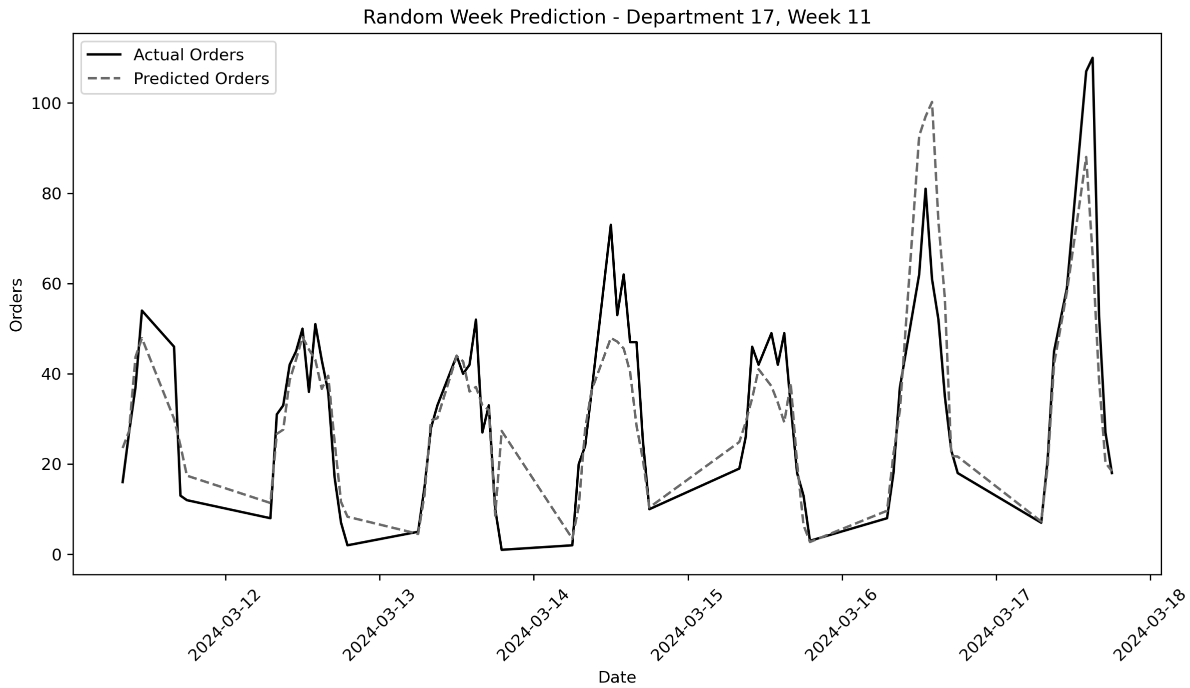


Figure 10: Random Forest predictions vs. actual orders for department 17, week 11

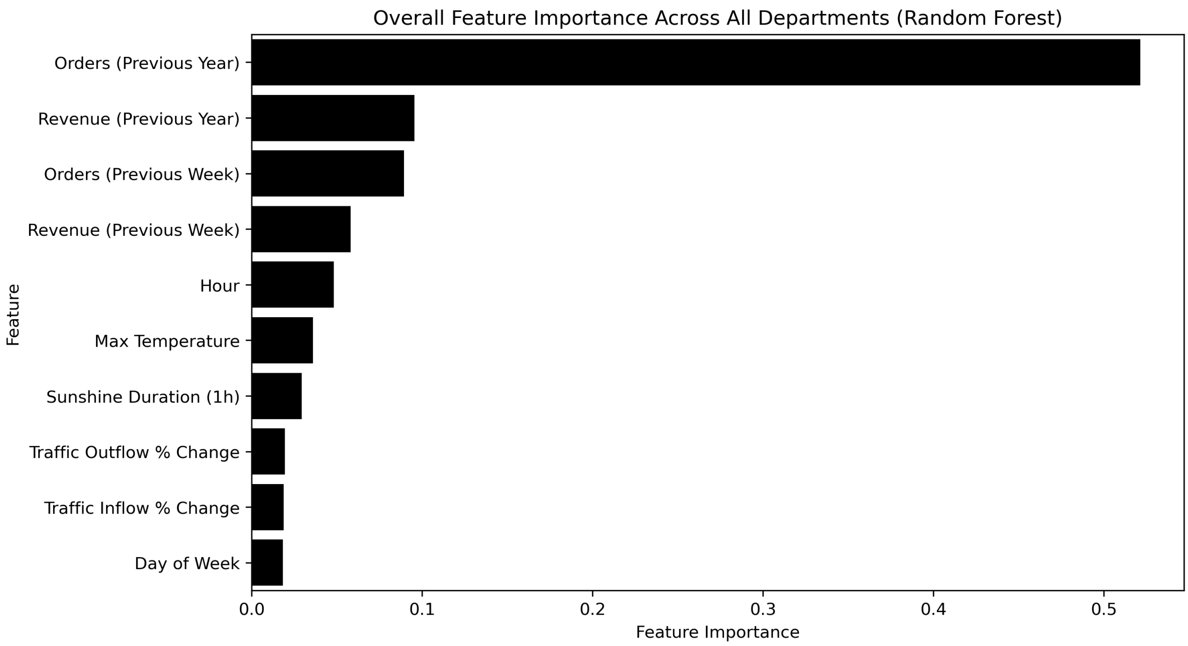


Figure 11: Overall feature importance across all departments

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

3.3.2 Boosting Models: XGBoost and LightGBM

Model Overview

Boosting models such as XGBoost and LightGBM iteratively enhance weak learners by minimizing a predefined loss function $L(y, \hat{y})$. By sequentially correcting errors from previous models, boosting effectively captures complex patterns in the data (James et al., 2021, p. 346). The objective function is expressed as:

$$\mathcal{L}(\theta) = \sum_{i=1}^{n} L(y_i, f(x_i; \theta)) + \Omega(f) \tag{10}$$

where:

- $L(y_i, f(x_i; \theta))$ denotes the loss function.
- $\Omega(f)$ a regularization term
- $f(x_i; \theta)$ the predictive function

Each new boosting iteration refines previous residuals as follows:

$$h_t(x) = -\gamma \frac{\partial L(y, \hat{y})}{\partial \hat{y}} \tag{11}$$

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

Results

The models' performance metrics across different departments are summarized in Table 6.

Table 6: Model performance across departments with average orders and error margins

Department	XGBoost MAE	XGBoost RMSE	LightGBM MAE	LightGBM RMSE	Avg. Orders	XGBoost Error Margin (%)	LightGBM Error Margin (%)
11	4.87	6.76	4.90	6.61	13.73	35.5%	35.7%
12	7.20	9.44	7.10	9.27	31.89	22.6%	22.2%
13	6.82	9.44	6.82	9.31	35.48	19.2%	19.2%
17	7.35	9.87	7.35	9.85	36.47	20.2%	20.2%
20	5.04	6.64	5.06	6.61	19.18	26.3%	26.4%
21	6.31	8.24	6.15	8.04	21.55	29.3%	28.5%
22	5.53	7.24	5.34	7.00	22.07	25.1%	24.2%
24	5.30	7.04	5.28	6.97	20.87	25.4%	25.3%
25	6.36	8.57	6.40	8.54	24.93	25.5%	25.7%
26	6.39	8.39	6.40	8.38	17.92	35.7%	35.7%
27	5.75	7.60	5.64	7.39	18.36	31.3%	30.7%
32	8.12	10.71	8.00	10.49	37.70	21.5%	21.2%
36	7.94	10.49	7.93	10.51	29.97	26.5%	26.5%

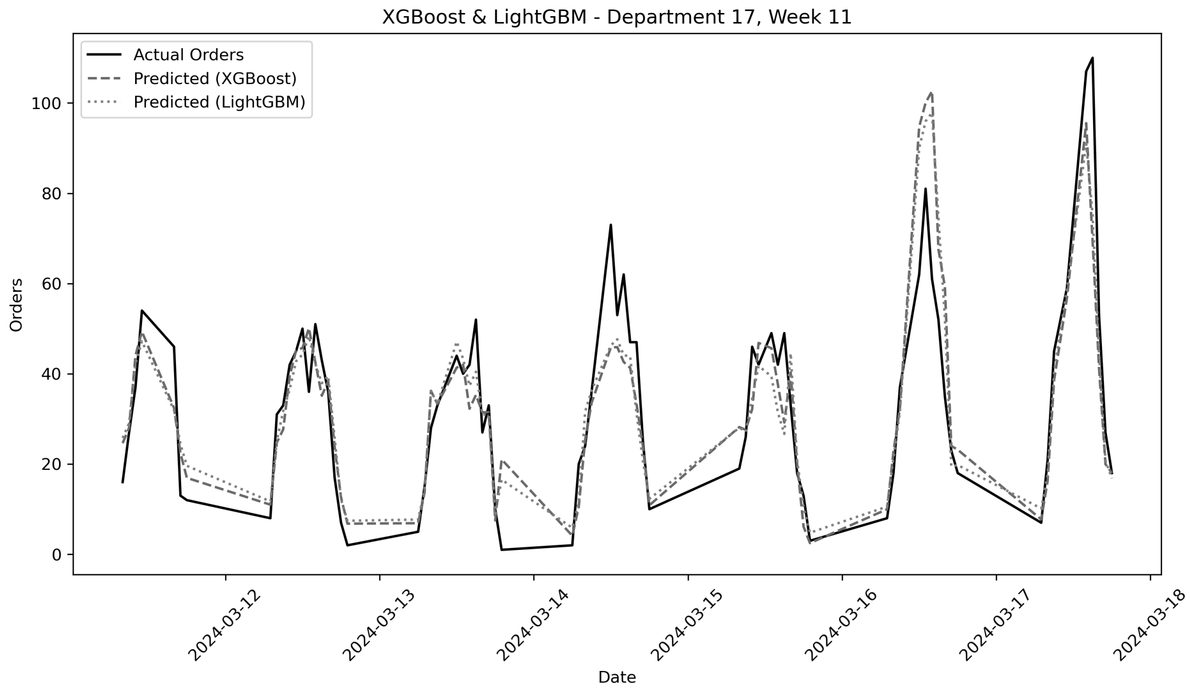


Figure 12: Predictions vs. actual orders for XGBoost and LightGBM models. Department 17, week 11

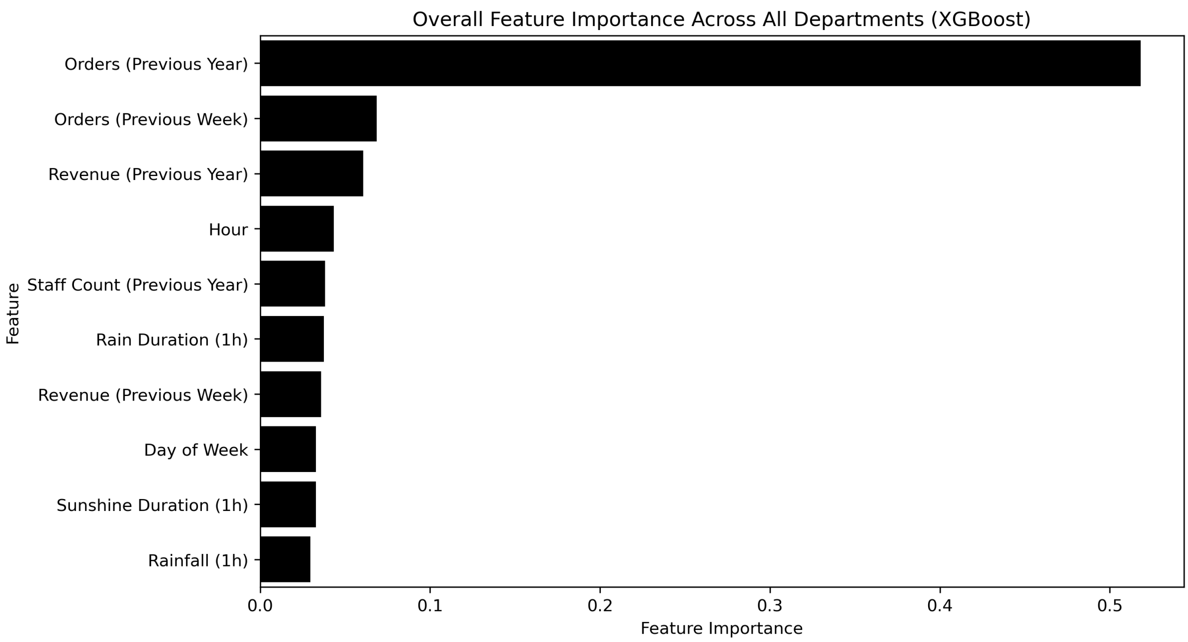


Figure 13: Feature importance analysis for XGBoost model

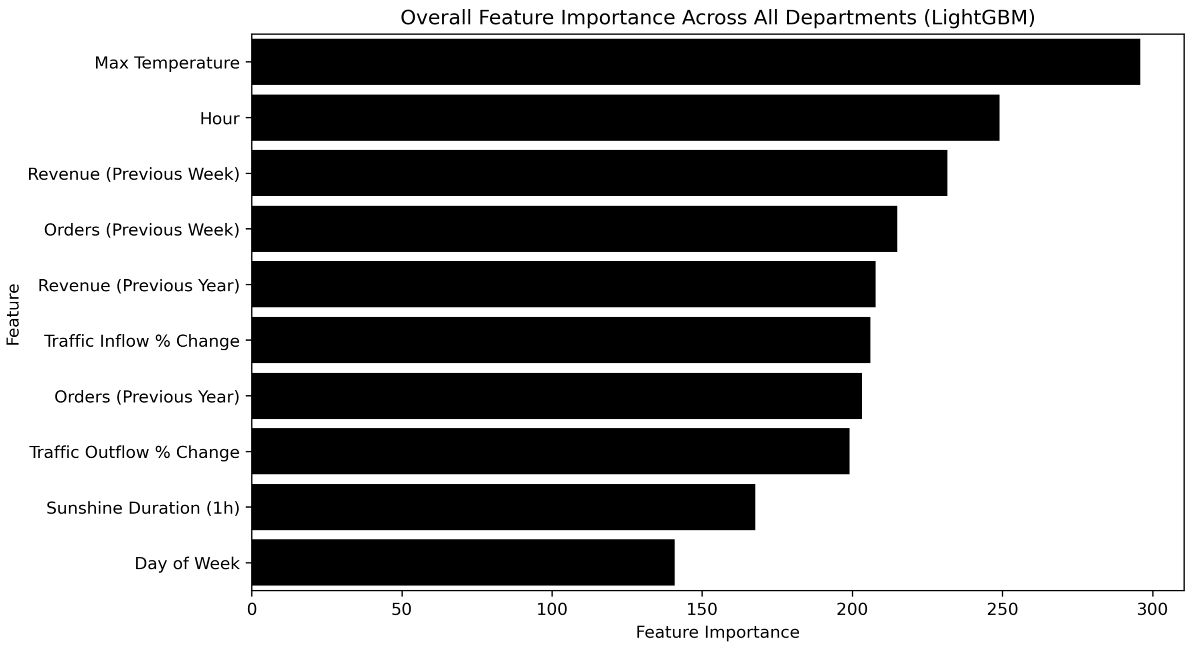


Figure 14: Feature importance analysis for LightGBM model

3.4 Building Deep Learning Models

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

$$X' = \frac{X - X_{\min}}{X_{\max} - X_{\min}} \tag{12}$$

$$y' = \frac{y - y_{\min}}{y_{\max} - y_{\min}} \tag{13}$$

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:

$$\hat{y} = y'(y_{\max} - y_{\min}) + y_{\min} \tag{14}$$

3.4.1 Neural Network (ReLU)

Model Description

The neural network model is designed to predict the number of orders per department using multiple input features. Neural networks can learn complex patterns in data, making them effective for modeling sequential dependencies in order predictions (James et al., 2021, p. 427). 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:

$$\hat{y} = f(W_3 \cdot \max(0, W_2 \cdot \max(0, W_1 X + b_1) + b_2) + b_3) \tag{15}$$

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

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

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

Results

The results for each department are summarized in Table 7.

Table 7: Neural Network Prediction Performance per Department with Avg. Orders and Error Margin

Department	MAE	RMSE	Avg. Orders	Error Margin (%)
11	5.33	7.02	13.73	38.8
12	7.49	9.60	31.89	23.5
13	8.18	10.78	35.48	23.1
17	7.38	9.64	36.47	20.2
20	5.74	7.38	19.18	29.9
21	6.51	8.35	21.55	30.2
22	5.81	7.40	22.07	26.3
24	5.51	7.13	20.87	26.4
25	6.35	8.40	24.93	25.5
26	6.62	8.77	17.92	36.9
27	6.12	7.86	18.36	33.3
32	8.31	10.75	37.70	22.0
36	8.35	10.90	25.10	33.3

The training process is visualized by plotting the loss function over epochs. Figure 16 shows the training and validation loss for a randomly selected department.

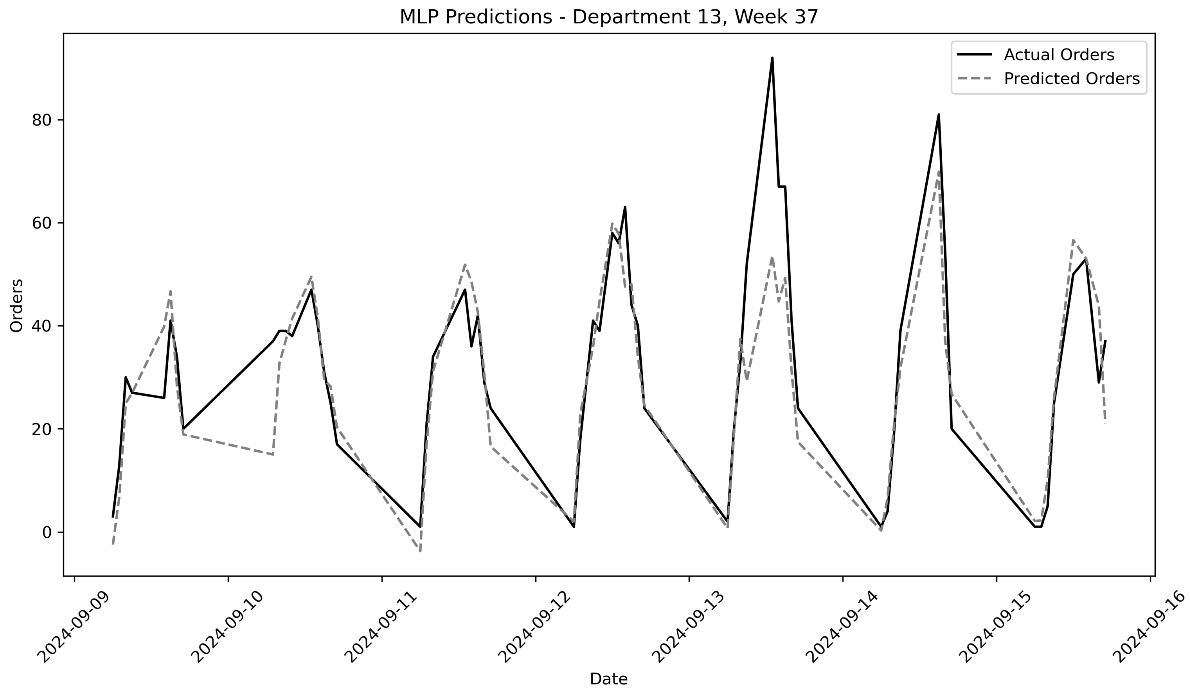


Figure 15: ReLu model predictions for department 13, week 37

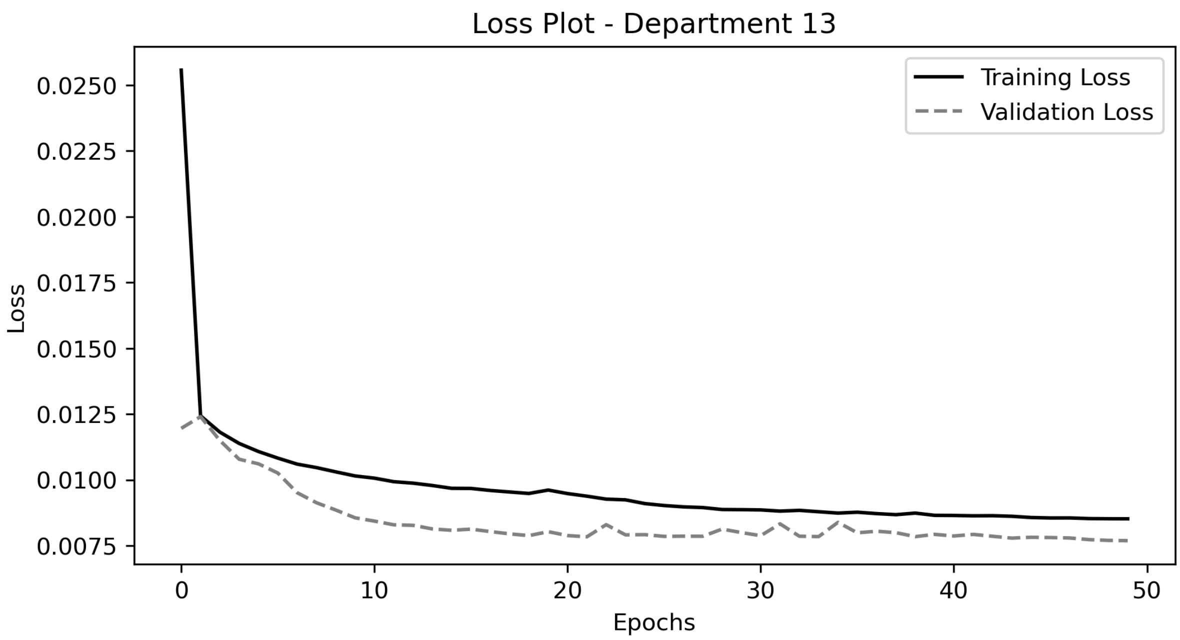


Figure 16: Loss plot for the randomly selected department 13

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.

3.4.2 Hybrid Models - (CNN-LSTM)

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. Similar methods have improved accuracy in food delivery forecasting (Crivellari et al., 2022) and retail sales predictions (Ahmed et al., 2024).

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:

$$h_t = \sigma(W_f F_t + W_h h_{t-1} + b) \tag{17}$$

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.

Results

Table 8 presents the performance metrics for each department.

Department	MAE	RMSE	Avg. Orders	Error Margin (%)
11	4.93	6.64	13.73	35.9%
12	8.17	10.92	31.89	25.6%
13	9.75	12.83	35.48	27.5%
17	8.09	10.87	36.47	22.2%
20	5.14	6.71	19.18	26.8%
21	6.81	8.79	21.55	31.6%
22	6.39	8.13	22.07	28.9%
24	6.10	7.98	20.87	29.2%
25	6.60	8.89	24.93	26.5%
26	7.38	9.61	17.92	41.2%
27	6.35	8.39	18.36	34.6%
32	9.62	12.60	37.70	25.5%
36	8.68	11.49	29.97	29.0%

Table 8: CNN-LSTM model performance across different departments.

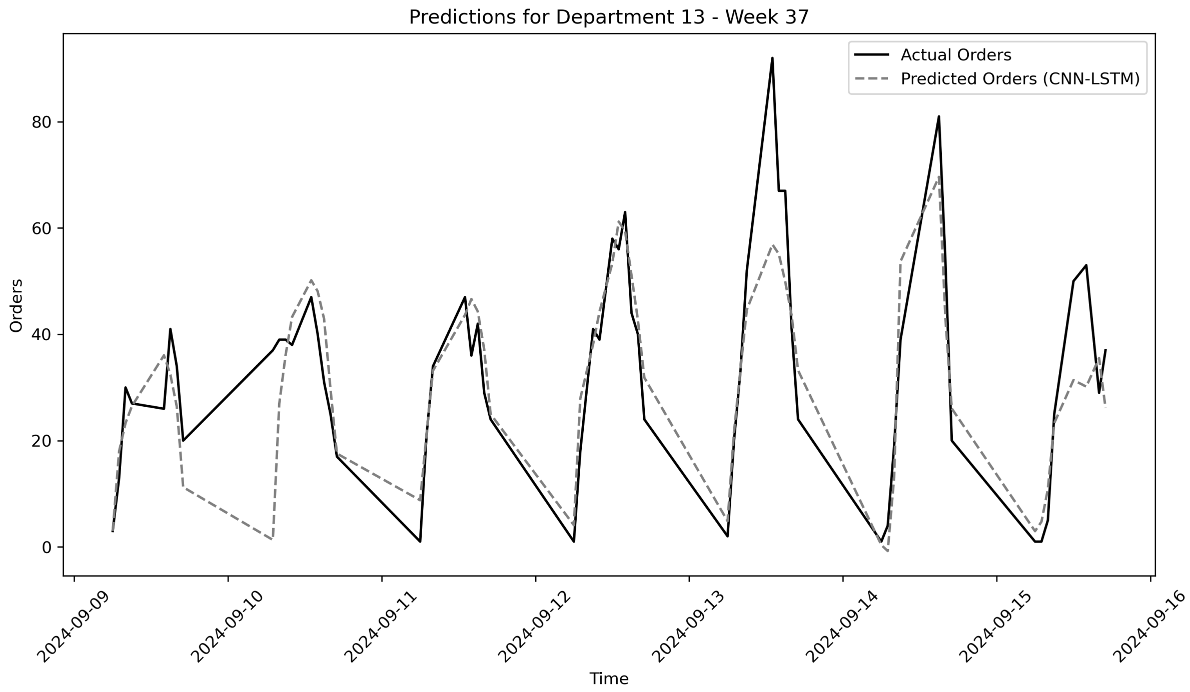


Figure 17: Predicted vs. actual order volumes for department 13, week 37

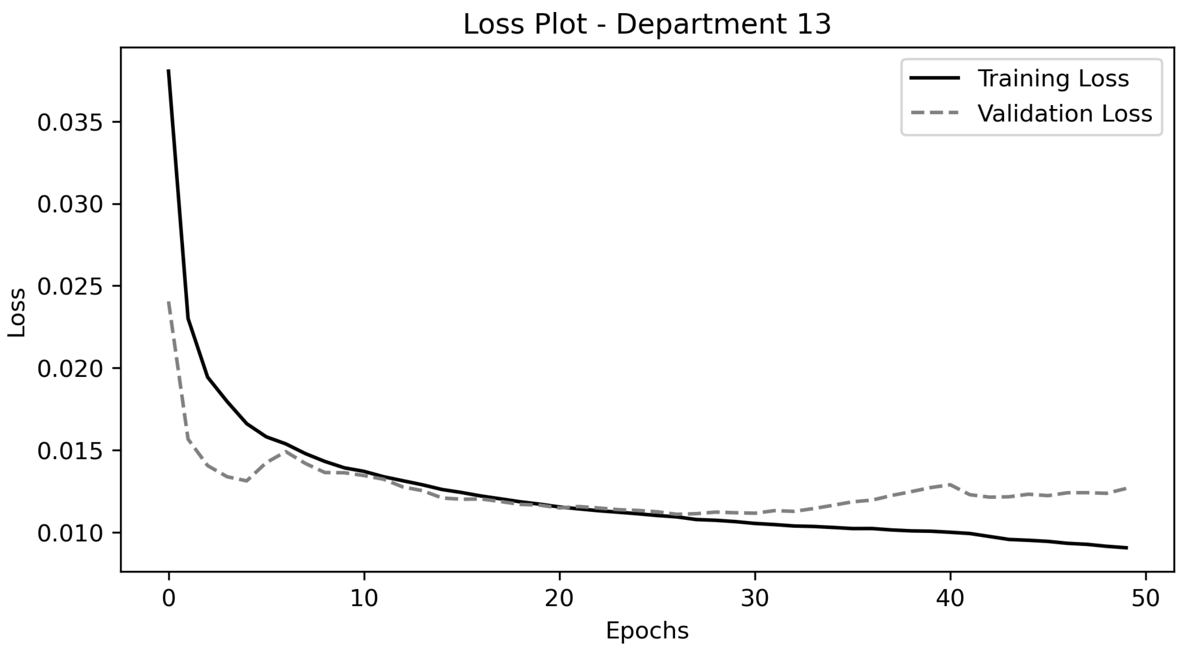


Figure 18: Training and validation loss over epochs for CNN-LSTM

4 Evaluating AI Models

In this section, we compare the performance of the models introduced in earlier sections, focusing on their predictive accuracy across different departments. We assess the models using MAE and RMSE.

4.1 Performance Metrics

4.1.1 Mean Absolute Error (MAE)

D	KNN	Ridge	Lasso	Random Forest	XGBoost	LightGBM	ReLU	CNN-LSTM
11	4.68	4.85	4.76	4.83	4.87	4.90	5.33	4.93
12	7.62	7.36	7.31	7.00	7.20	7.10	7.49	8.17
13	6.76	7.99	7.86	6.65	6.82	6.82	8.18	9.75
17	7.21	7.22	7.20	7.17	7.35	7.35	7.38	8.09
20	4.92	5.24	5.20	5.00	5.04	5.06	5.74	5.14
21	6.06	6.12	6.09	6.16	6.31	6.15	6.51	6.81
22	5.41	5.52	5.49	5.49	5.53	5.34	5.81	6.39
24	5.08	5.37	5.33	5.07	5.30	5.28	5.51	6.10
25	6.22	6.50	6.47	6.33	6.36	6.40	6.35	6.60
26	5.98	6.59	6.50	6.46	6.39	6.40	6.62	7.38
27	5.58	6.06	6.01	5.70	5.75	5.64	6.12	6.35
32	7.92	8.20	8.13	7.99	8.12	8.00	8.31	9.62
36	8.20	9.16	9.03	7.80	7.94	7.93	8.35	8.68
Total	81.64	86.18	85.38	81.65	83.88	82.37	87.70	94.01
Average	6.28	6.63	6.57	6.28	6.45	6.34	6.75	7.23

Table 9: Overall MAE Comparison for All Models by Department

4.1.2 Root Mean Squared Error (RMSE)

D	KNN	Ridge	Lasso	Random Forest	XGBoost	LightGBM	ReLU	CNN-LSTM
11	6.34	6.28	6.20	6.65	6.76	6.61	7.02	6.64
12	10.10	9.39	9.35	9.16	9.44	9.27	9.60	10.92
13	9.54	10.23	10.10	9.26	9.44	9.31	10.78	12.83
17	9.59	9.53	9.51	9.61	9.87	9.85	9.64	10.87
20	6.46	6.90	6.86	6.59	6.64	6.61	7.38	6.71
21	7.98	7.92	7.88	8.08	8.24	8.04	8.35	8.79
22	6.98	7.00	6.96	7.15	7.24	6.99	7.40	8.13
24	6.88	6.99	6.94	6.74	7.04	6.97	7.13	7.98
25	8.45	8.62	8.58	8.49	8.57	8.54	8.40	8.89
26	8.08	8.59	8.51	8.49	8.39	8.38	8.77	9.61
27	7.40	7.80	7.75	7.54	7.60	7.39	7.86	8.39
32	10.57	10.55	10.48	10.42	10.71	10.49	10.75	12.60
36	11.05	11.61	11.49	10.39	10.49	10.51	10.90	11.49
Total	109.42	110.41	110.11	108.57	110.43	107.96	114.98	124.85
Average	8.42	8.49	8.47	8.35	8.49	8.30	8.84	9.60

Table 10: Overall RMSE Comparison for All Models by Department

4.2 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 KNN, Ridge Regression, Lasso Regression, XGBoost, LightGBM, Neural Networks, and CNN-LSTM.

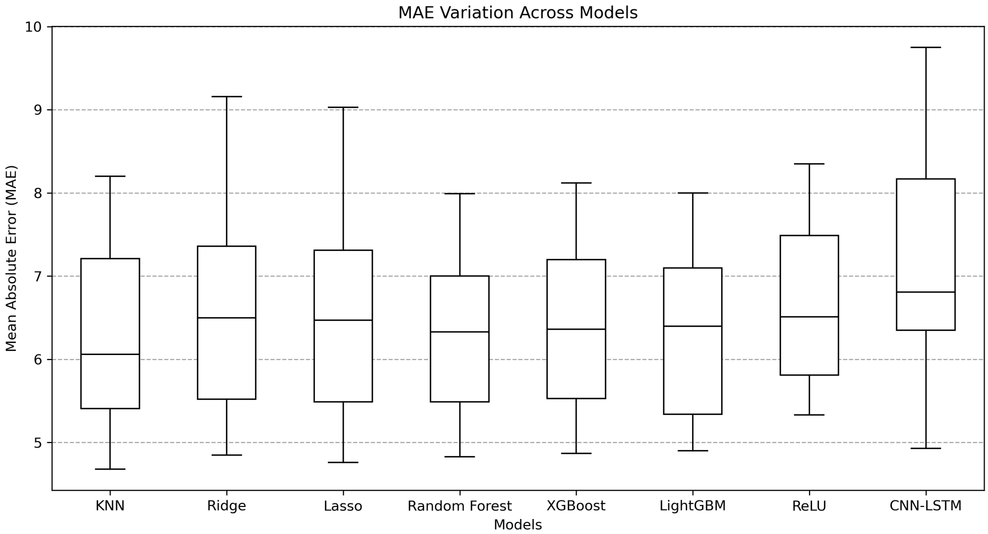


Figure 19: MAE variation across models

To quantify performance, the mean MAE's (μ_{MAE}) and standard deviations (σ_{MAE}) were calculated for each model:

$$\mu_{MAE} = \frac{1}{n}\sum_{i=1}^{n} MAE_i \tag{18}$$

$$\sigma_{MAE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(MAE_i - \mu_{MAE})^2} \tag{19}$$

where n represents the number of departments analyzed.

Model	Mean MAE	Std Dev
KNN	6.280	1.127
Random Forest	6.281	0.996
LightGBM	6.336	1.018
XGBoost	6.383	1.024
Lasso	6.568	1.225
Ridge	6.629	1.244
ReLU	6.746	1.040
CNN-LSTM	7.232	1.488

Table 11: Model Performance Analysis across n departments.

Additionally, a complexity factor (C) was 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:

$$R = \mu_{MAE} + \sigma_{MAE} + 0.5C \quad (20)$$

The final ranking metric (R) is a weighted sum of the mean MAE, the standard deviation of MAE, and the complexity factor, where a lower overall score indicates a more favorable model. This approach balances accuracy with practical feasibility, making it a suitable method for assessing model performance in real-world applications.

Model	Mean MAE	Std Dev	Complexity	Overall Score
Random Forest	6.281	0.996	2	8.277
Lasso	6.568	1.225	1	8.293
LightGBM	6.336	1.018	2	8.354
Ridge	6.629	1.244	1	8.373
KNN	6.280	1.127	2	8.407
XGBoost	6.383	1.024	2	8.407
ReLU	6.746	1.040	3	9.287
CNN-LSTM	7.232	1.488	3	10.220

Table 12: Model Performance Analysis including Complexity.

5 Summary of Findings

5.1 Discussion of Results

5.1.1 Performance Comparison

The Random Forest model demonstrated the best performance for workforce planning, achieving the lowest overall score of 8.277, with strong predictive accuracy (MAE = 6.281) and stable results. This combination of accuracy and consistency makes Random Forest the most suitable model for practical deployment in workforce planning. KNN showed similar predictive accuracy (MAE = 6.280) but exhibited significantly higher variability ($\sigma = 1.127$), resulting in a higher overall score of 8.407. The increased variability undermines KNN's stability, making it less reliable in contexts where consistency is crucial.

Gradient Boosting models, such as LightGBM and XGBoost, underperformed relative to Random Forest, with overall scores of 8.354 and 8.407, respectively. Despite their ability to handle complex datasets, their predictive accuracy was marginally worse than Random Forest, rendering them less efficient for real-world deployment. Simpler models, such as Lasso and Ridge regression, exhibited higher MAEs (6.568 and 6.629) and struggled to capture complex patterns, as they are unable to model non-linear relationships effectively.

Deep Learning models, including ReLU and CNN-LSTM, showed the worst performance, with overall scores of 9.287 and 10.220, respectively. Despite their ability to model non-linear relationships, their predictive accuracy (MAEs of 6.746 and 7.232) was significantly lower than that of the other models. The high computational cost and inferior performance render them unsuitable for this application, where both efficiency and accuracy is essential.

5.1.2 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.

5.1.3 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 Random Forest 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.

5.1.4 Model Limitations and Future Research

While predictive models enhance workforce scheduling by aligning staffing with demand, their implementation necessitates managerial oversight to account for operational contingencies, such as unexpected promotions or macroeconomic fluctuations. Prior research (Döring et al., 2024) highlights the predictive value of macroeconomic indicators, including GDP growth and consumer confidence. Integrating such factors, alongside business-specific variables like marketing initiatives and competitive dynamics, could further refine the model's adaptability and predictive accuracy.

Model performance also exhibited variation across departments, suggesting that a uniform modeling approach may not be optimal for all locations. Future research could explore ensemble methods that dynamically adjust model selection based on department-specific characteristics, enhancing both precision and robustness.

5.2 Implementation and Deployment

Business Perspective

This AI-driven forecasting approach offers a scalable, data-driven solution for workforce planning, similar to recommender systems optimizing customer choices. By analyzing historical and real-time data, the model dynamically adjusts staffing to match demand, reducing costs while maintaining service quality.

Like recommender systems mitigating choice overload, AI-powered workforce planning predicts order volumes and suggests optimal staffing. A real-time dashboard visualizes these insights, allowing managers to make informed adjustments, enhancing efficiency in demand-driven industries such as cafés, restaurants, and hospitality.

Power BI Dashboard for Decision Support

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.

Using PySpark for Big Data Processing

Given IT architecture and cost considerations, Azure ML Notebook is recommended for scalable machine learning. For demonstration, we implemented a PySpark notebook in Microsoft Fabric, leveraging its efficiency for large-scale data processing.

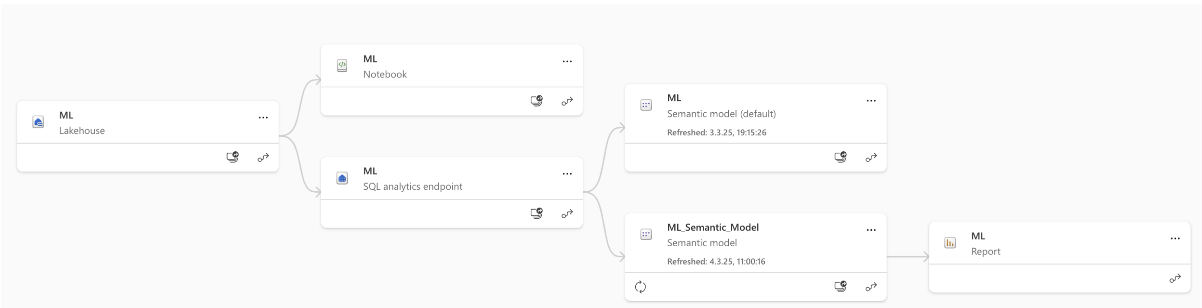


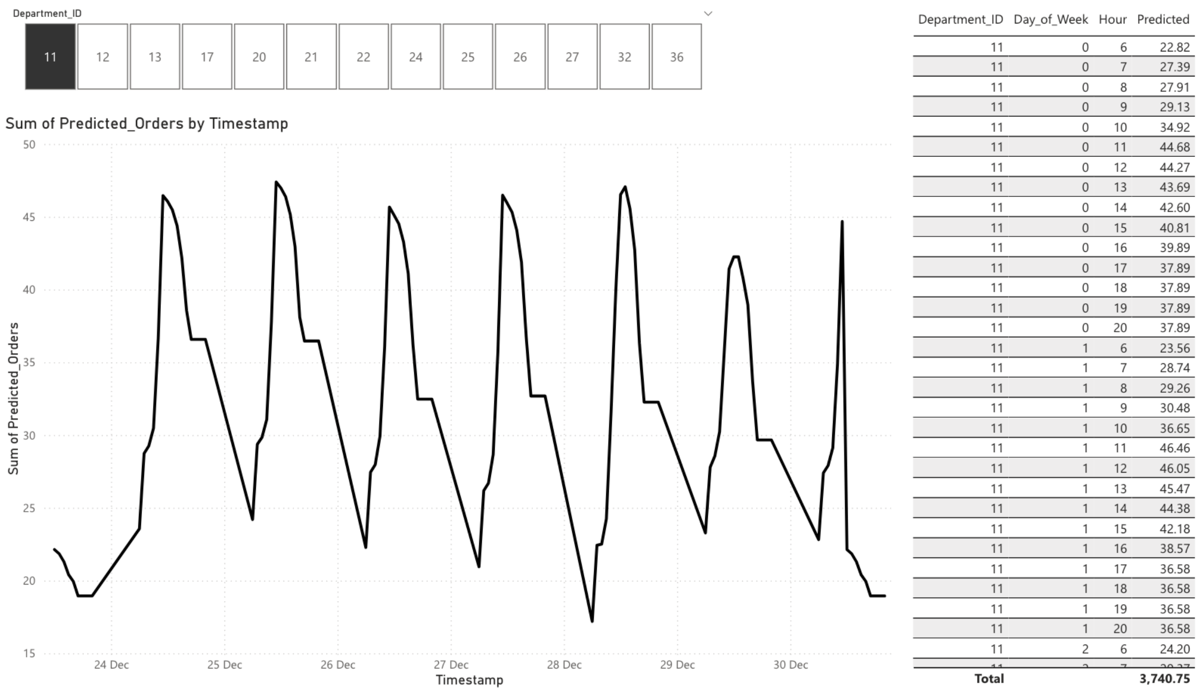
Figure 20: Fabric Lakehouse and Power BI Integration

Scheduling and Model Execution

The PySpark notebook runs weekly, ensuring the staffing plan stays updated with the latest order trends. For simplicity, only internal historical data was used, as external factors like weather and traffic had low overall feature importance.

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.



Department_ID	Day_of_Week	Hour	Predicted
11	0	6	22.82
11	0	7	27.39
11	0	8	27.91
11	0	9	29.13
11	0	10	34.92
11	0	11	44.68
11	0	12	44.27
11	0	13	43.69
11	0	14	42.60
11	0	15	40.81
11	0	16	39.89
11	0	17	37.89
11	0	18	37.89
11	0	19	37.89
11	0	20	37.89
11	1	6	23.56
11	1	7	28.74
11	1	8	29.26
11	1	9	30.48
11	1	10	36.65
11	1	11	46.46
11	1	12	46.05
11	1	13	45.47
11	1	14	44.38
11	1	15	42.18
11	1	16	38.57
11	1	17	36.58
11	1	18	36.58
11	1	19	36.58
11	1	20	36.58
11	2	6	24.20
Total			**3,740.75**

Figure 21: 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

Ahmed, R. S., Hasnain, M., Mahmood, M. H., & Mehmood, M. A. (2024). *Comparison of deep learning algorithms for retail sales forecasting.* IECE Transactions on Intelligent Systematics, *1*(3), 112–126. Retrieved from
https://www.iece.org/article/abs/tis.2024.300700

Crivellari, A., Beinat, E., Caetano, S., Seydoux, A., & Cardoso, T. (2022). Multi-target CNN-LSTM regressor for predicting urban distribution of short-term food delivery demand. *Journal of Business Research, 145*, 613–625.
https://doi.org/10.1016/j.jbusres.2022.02.039

Döring, L., Grumbach, F., & Reusch, P. (2024). *Optimizing sales forecasts through automated integration of market indicators.* ResearchGate.
https://www.researchgate.net/publication/381373126_Optimizing_Sales_Forecasts_through_Automated_Integration_of_Market_Indicators

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2021). *An introduction to statistical learning: With applications in R* (2nd ed.). Springer.

Kystverket. (n.d.). *Hendelses- og analyseinformasjonsystem (HAIS).* Retrieved March 5, 2025, from https://hais.kystverket.no/https://hais.kystverket.no/

Meller, J. (2020). *Data-driven operations management: Combining machine learning and optimization for improved decision-making.* Retrieved from https://www.researchgate.net/publication/342379797_Data-driven_Operations_Management_Combining_Machine_Learning_and_Optimization_for_Improved_Decision-making

Shehzadi, T. (2025). *Time series analysis with machine learning: A comprehensive review and future directions.* ResearchGate. Retrieved from
https://www.researchgate.net/publication/388836992_Time_Series_Analysis_with_Machine_Learning_A_Comprehensive_Review_and_Future_Directions

Yang, X., & Wen, W. (2018). *Ridge and Lasso regression models for cross-version defect prediction.* IEEE Transactions on Reliability, *67*(3), 885–896.
https://doi.org/10.1109/TR.2018.2847353

Appendix

This appendix provides a short description of the supplementary files submitted along with this report.

- **Models.ipynb** – Implements and evaluates the models.
- **EDA.ipynb** – Code for exploratory data analysis.
- **Fact_table.ipynb** – Processes features from internal data source.
- **Merged_Main.ipynb** – Merges internal and external datasets and finalizes preprocessing for modeling.
- **Staffing.ipynb** – Cleans and prepares raw staffing data.
- **Fabric_Notebook.py** – PySpark notebook used for demonstrating deployment in Microsoft Fabric.
- **Traffic_Data.csv** – External traffic data by city

- **Weather_Data.xlsx** – External weather data by city.
- **Cruise_Data.xlsx** – External data tracking cruise ship traffic by city.