Forecasting the Number of Orders for Upcoming 10 Weeks for a Meal Delivery Company

1. Problem statement

A meal delivery company operates in multiple cities and maintains various fulfillment centers across these locations to dispatch meal orders to customers. The company's goal is to forecast orders for the upcoming weeks, enabling these centers to plan their raw material stocks accordingly. Since most raw materials are replenished weekly and are perishable, precise procurement planning is crucial. Accurate demand forecasts are also essential for the effective staffing of the centers.

With 135 weeks of data, the project aims to predict demand (number of orders) for the upcoming 10 weeks (Weeks: 136-145) for the center-meal combinations. To achieve this, we conduct multiple regression models, including Linear Regression, Decision Tree, Random Forest, and XGBoost models. These models are tuned to find the best hyperparameters to train the dataset. The performance of each model on the test dataset is compared using metrics such as R-squared, Mean Absolute Error (MAE), or Root Mean Squared Error (RMSE), and the model with the best performance is selected.

2. Data Wrangling

2.1Data Sources

The data is sourced from kaggle and consists of the following datasets:

- Fulfillment Center Information: Contains a total of 5 features for 77 centers.
- Meal Information: Comprises 3 features detailing information for 52 meals.
- Meal/Center Number of Orders: Encompasses 9 features collected from 423k orders over 135 weeks.

2.2 Data Organization

Upon merging, our dataset is formed, focusing on key features, namely center ID and meal ID. The data is organized by week in ascending order due to its temporal nature. In total, our dataset contains 15 features.

Furthermore, a check for missing values was conducted, and no gaps were found in the dataset.

2.3 Unique Values

Additionally, the number of unique values was determined for certain features:

- Center: 77
- City: 51

- Center Type:3
- Meal: 51
- Meal Category: 14
- Cuisine: 4

3. Exploratory Data Analysis

3.1individual feature distributions

The overall distribution of numeric values is shown in figure 3-1.

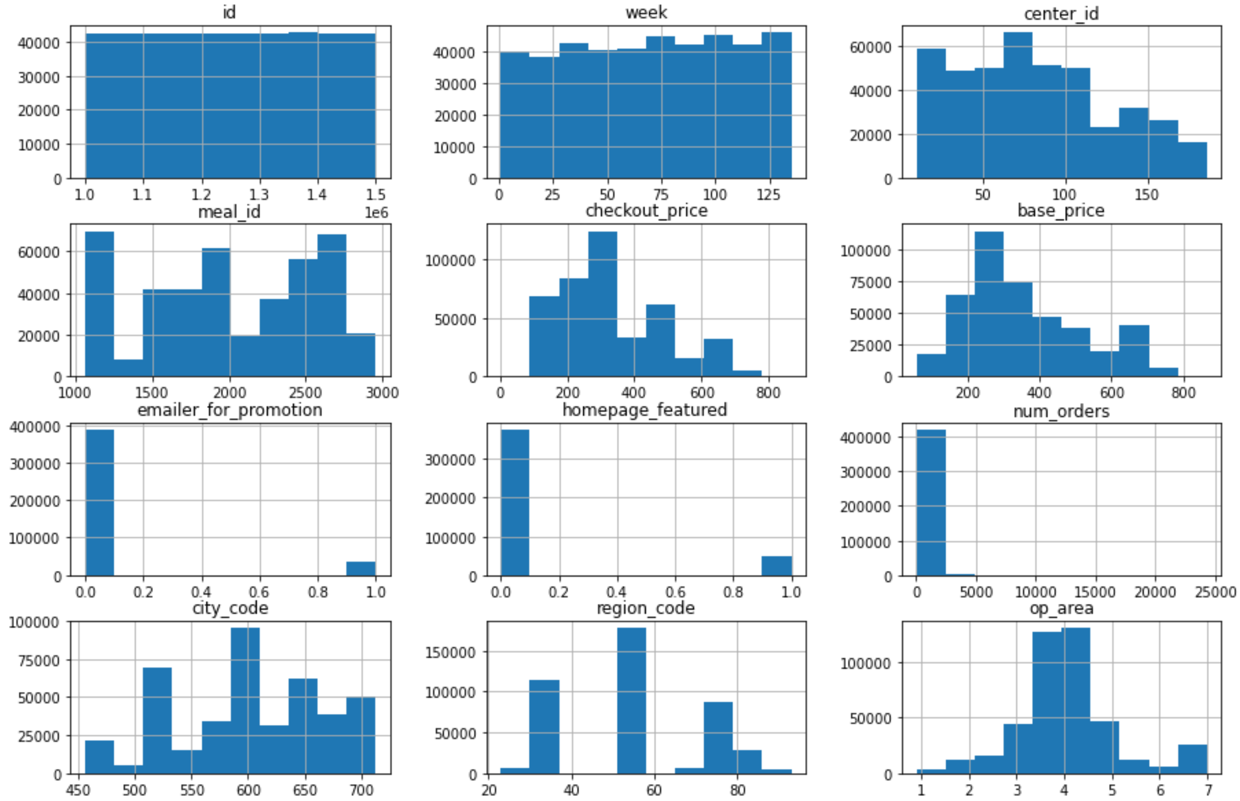


Figure 3-1: Distribution Of Numeric Values

From Figure 3-1 we observed that most features look sensible. The only concern is the 'num_orders' variable, which, despite mostly having low values, exhibits a really high maximum. Further investigation reveals 22 rows with over 10,000 orders, primarily in week 5, mostly for 'Rice Bowl' and some sandwiches. All these high-volume orders are associated with email promotions or homepage features. Additionally, there is a single row with more than 16,000 orders (24,299), also occurring in week 5 and associated with 'Rice Bowl.' No clear evidence of outliers is found, suggesting these occurrences could be linked to special events or promotions during that week.

Moreover, unique values for categorical features are shown in Figure 3-2.

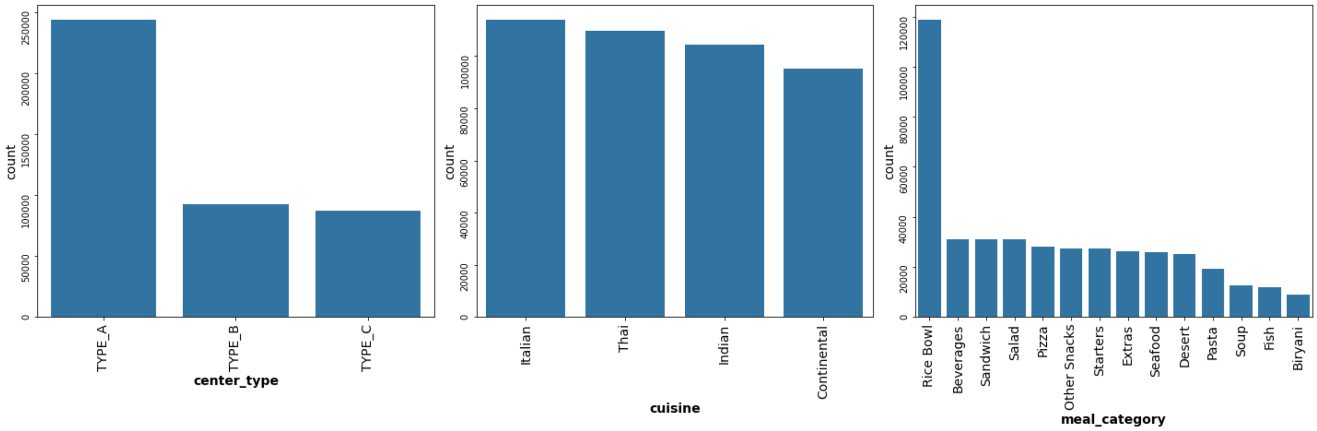


Figure 3-2: Distribution of Unique Values in Categorical Variables

3.2 Exploring 'num_orders' Distribution Across Features

The distribution of num_orders by various features for each of the three years are examined and shown in the following section. The analysis predominantly focuses on the data from years 1 and 2, as these represent complete annual datasets, while year 3 only covers 31 weeks.

1- Meal:

A significant concentration is observed in a specific set of 21 meal IDs (41% of the meals), which collectively contribute to 80% of the total orders. This suggests that these meal IDs play a crucial role in driving a substantial portion of the overall order volume. Meal ID 2290 has the highest number of orders.

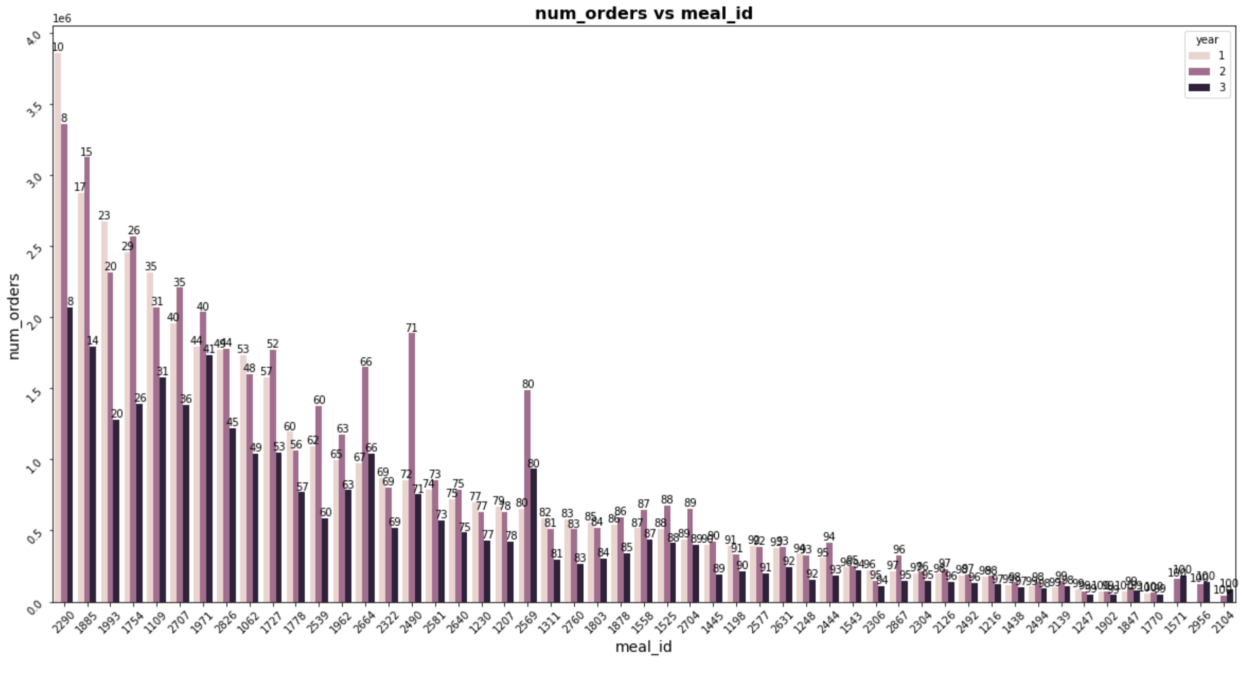


Figure 3-3: Number of order vs Meal Id

2- Center:

The distribution of orders by center for each year reveals a dispersed pattern, with no significant concentration in a small number of centers. Approximately 66% of the centers collectively contribute to 80% of the total number of orders.

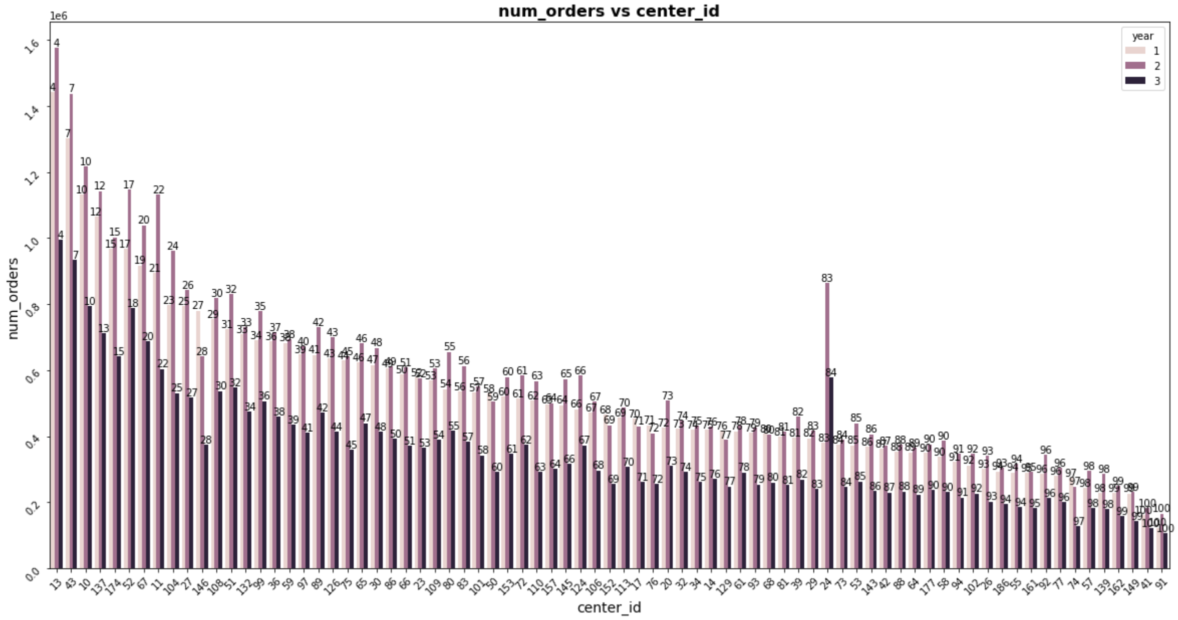


Figure 3-4: Number of order vs Center Id

3- City:

Three cities contribute nearly 30% of the total orders, while 80% of orders are spread across 57% of all cities.

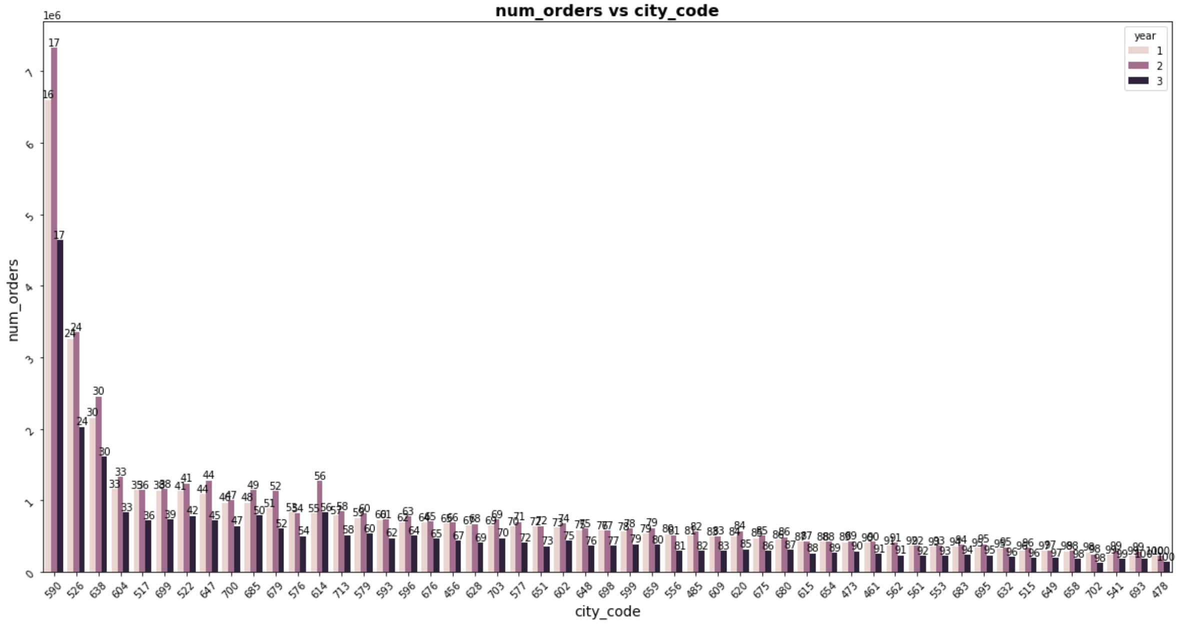


Figure 3-5: Number of order vs City Code

4- Region:

Three centers with codes 56, 34, and 77 contain 88% of the total number of orders, as depicted in the figure. This highlights the significance of these centers compared to others.

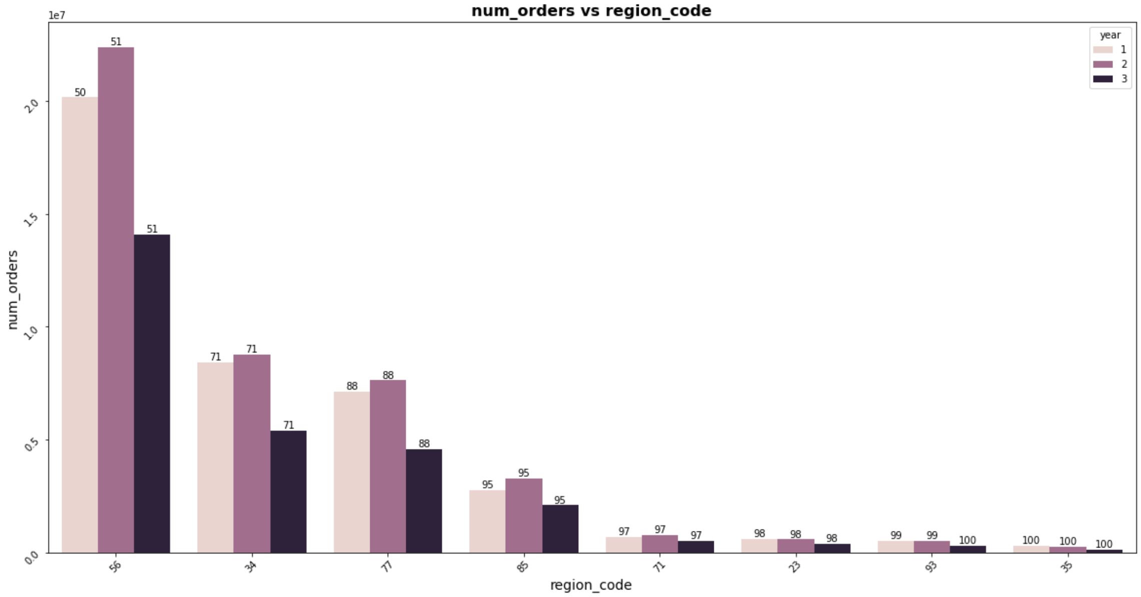


Figure 3-6: Number of order vs Region Code

5- Meal category:

Beverages, Rice Bowl, Sandwich, Salad, and Pizza account for more than 80% of orders among other meal categories, with Beverages having the highest demand of more than 30%.

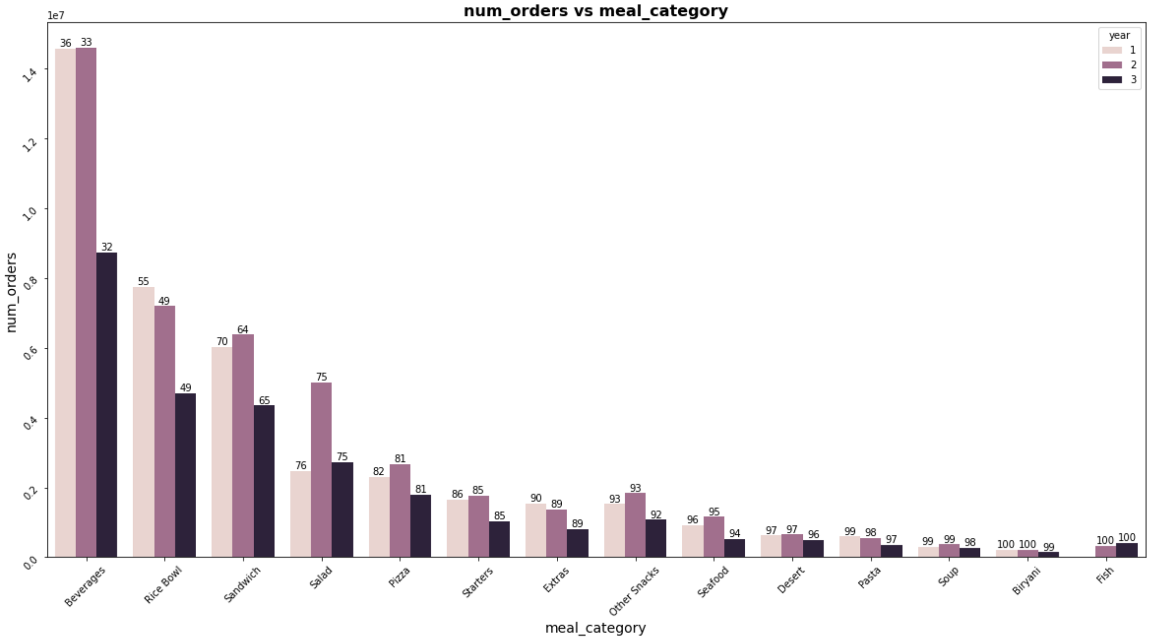


Figure 3-7: Number of order vs Meal Category

6- Center Type:

As shown in the figure 3-8, 58 percent of the number of orders belong to Center Type A.

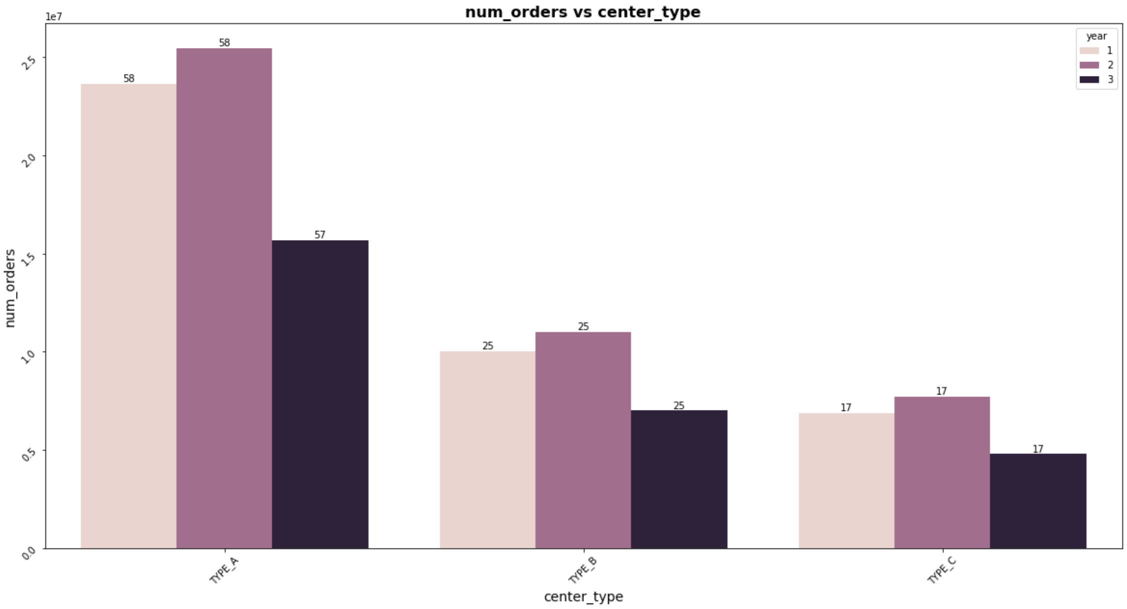


Figure 3-8: Number of order vs Center Type

7- Cuisine:

The distribution of the number of orders for cuisines is shown in the figure 3-9. Italian cuisine with 35%, 38%, and 39% has the highest number of orders in the first, second, and third years, respectively.

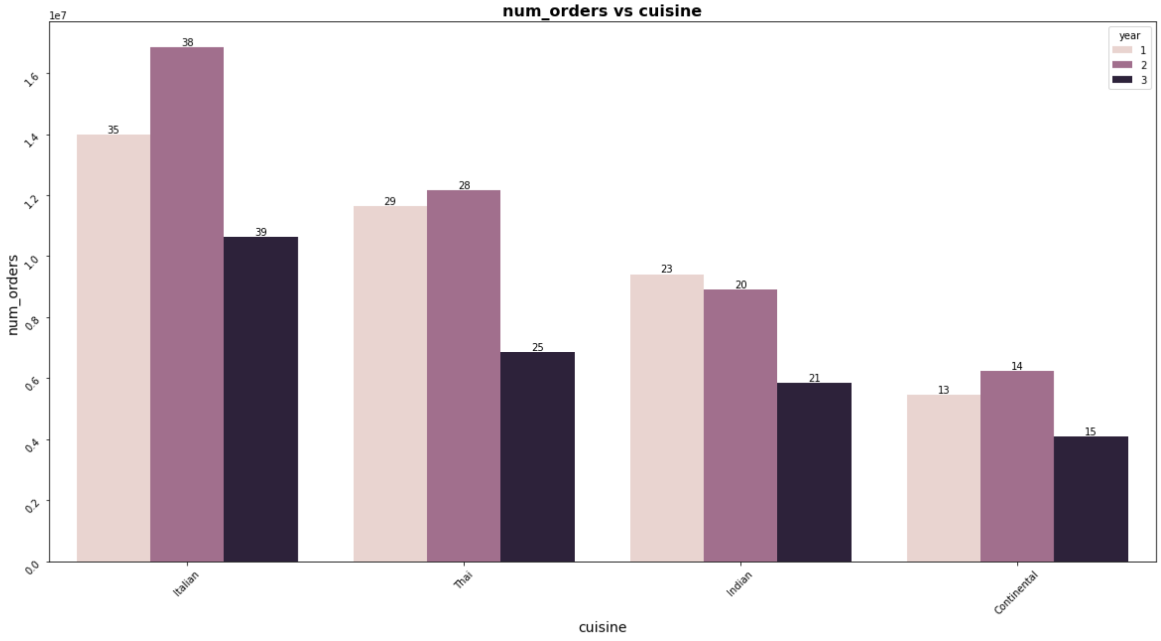


Figure 3-9: Number of order vs Cuisine

8- Week:

Weeks 48, 5, 32 in the first year, weeks 8, 1 in the second year, and week 5 in the third year have the highest number of orders, each contributing 3 to 4 percent of orders.

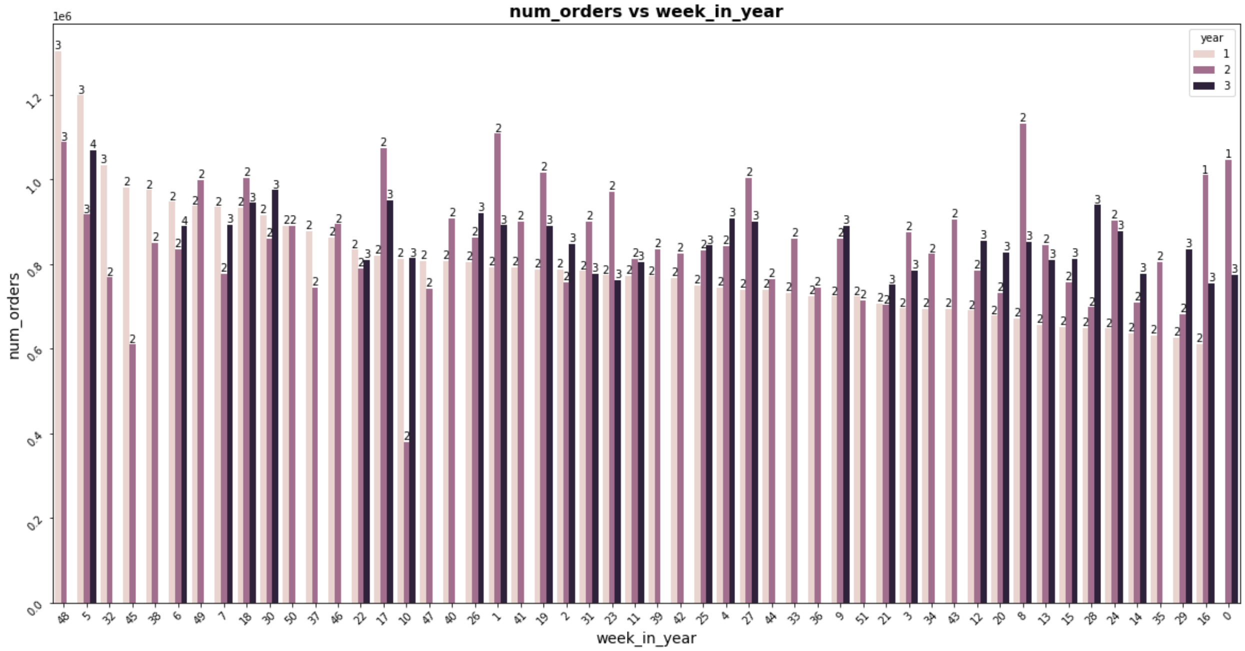
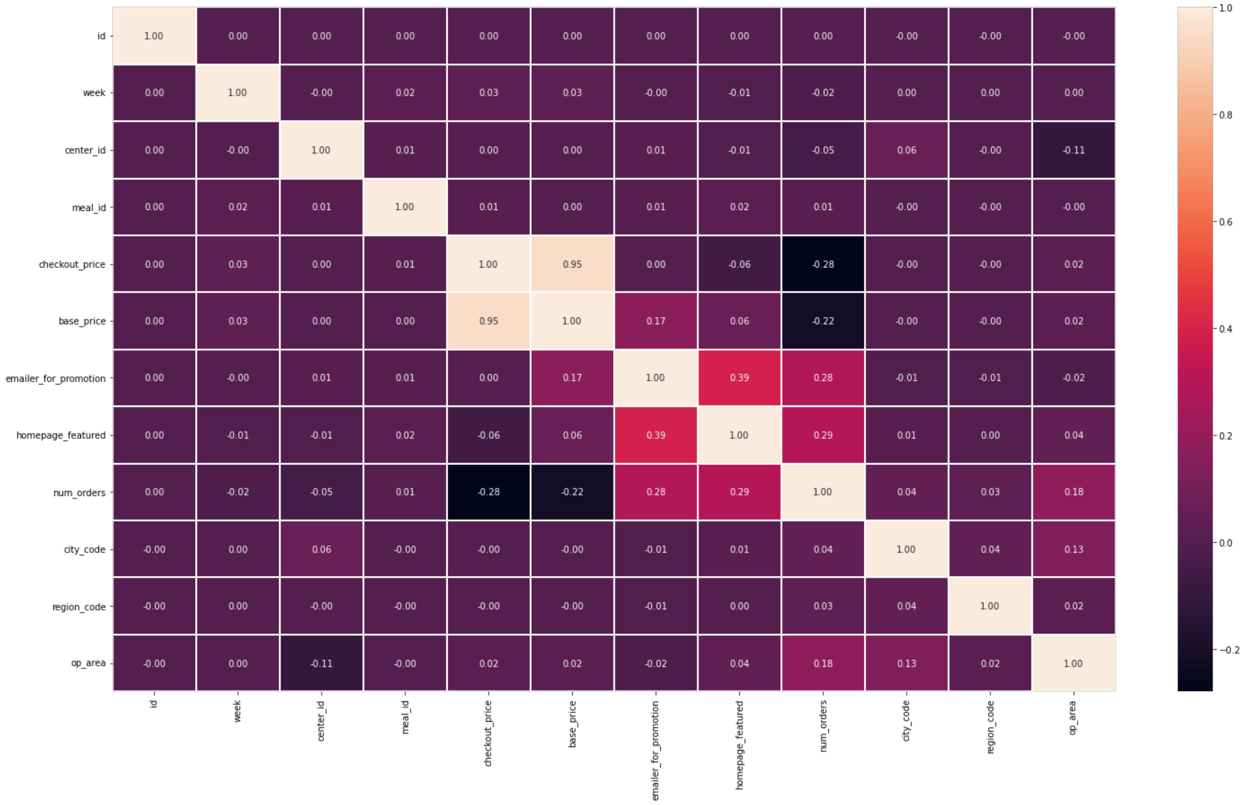


Figure 3-10: Number of order vs Week

The total number of orders for each year was explored, with a focus on specific features such as meals, centers, and cuisines. By closely examining these elements, we can pinpoint those that consistently contribute the most significant share of orders. This insight is crucial for refining our models and allocating resources effectively, as it directs our attention to key features that have a substantial impact on the overall order distribution.

3.3Exploring Feature Relationships

The correlation heatmap is plotted to illustrate the relationship between variables.



	id	week	center_id	meal_id	checkout_price	base_price	emailer_for_promotion	homepage_featured	num_orders	city_code	region_code	op_area
id	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	-0.00	-0.00	-0.00
week	0.00	1.00	-0.00	0.02	0.03	0.03	-0.00	-0.01	-0.02	0.00	0.00	0.00
center_id	0.00	-0.00	1.00	0.01	0.00	0.00	0.01	-0.01	-0.05	0.06	-0.00	-0.11
meal_id	0.00	0.02	0.01	1.00	0.01	0.00	0.01	0.02	0.01	-0.00	-0.00	-0.00
checkout_price	0.00	0.03	0.00	0.01	1.00	0.95	0.00	-0.06	-0.28	-0.00	-0.00	0.02
base_price	0.00	0.03	0.00	0.00	0.95	1.00	0.17	0.06	-0.22	-0.00	-0.00	0.02
emailer_for_promotion	0.00	-0.00	0.01	0.01	0.00	0.17	1.00	0.39	0.28	-0.01	-0.01	-0.02
homepage_featured	0.00	-0.01	-0.01	0.02	-0.06	0.06	0.39	1.00	0.29	0.01	0.00	0.04
num_orders	0.00	-0.02	-0.05	0.01	-0.28	-0.22	0.28	0.29	1.00	0.04	0.03	0.18
city_code	-0.00	0.00	0.06	-0.00	-0.00	-0.00	-0.01	0.01	0.04	1.00	0.04	0.13
region_code	-0.00	0.00	-0.00	-0.00	-0.00	-0.00	-0.01	0.00	0.03	0.04	1.00	0.02
op_area	-0.00	0.00	-0.11	-0.00	0.02	0.02	-0.02	0.04	0.18	0.13	0.02	1.00

Figure 3-11: Correlation Heatmap Between Variables

The correlation matrix provides valuable insights into the relationships between various variables. A strong positive correlation of 0.95 is observed between base price and checkout price, indicating a significant association. Additionally, moderate to mild correlations are identified between email promotion and homepage promotion (0.39) and between the number of orders and op_area (0.18). On the other hand, moderate negative correlations are noted between the number of orders and checkout price (0.28) as well as between the number of orders and base price (0.22). These findings highlight key connections that offering valuable insights for decision-making and strategy.

4. Modeling

4.1Data Preprocessing:

The process included generating dummy features from categorical variables (center type, meal category, and cuisine), doubling the number of features from 15 to 30. After removing "id" and "num_orders," 28 features were retained. The data underwent a sequential split for training and test sets, maintaining the temporal order of observations in time-series data. This approach prevents the improper utilization of future information to predict past events, a crucial consideration for time-series data. A 30% portion of the data was allocated to the test set to ensure proper evaluation.

4.2Model selection

We studied the performance of four regression models: Linear Regression, Decision Tree, Random Forest, and XGBoost. The summary of the model results is shown in table 4.1.

Table 4-1 Model Performance Comparison

Models	R-Squared	Mean Squared Error	Mean Absolute Error	Computation Time
Linear Regression	0.41	90,254	169	1.23
Decision Tree	0.47	81,155	126	1.81
Random forest	0.71	45,037	99	116.30
XGboost	0.65	53,446	119	1.64
Tunned XGBoost	0.75	39,369	96	245.36

1- Linear Regression

The analysis begins with a Linear Regression model, valued for its simplicity and interpretability, serving as a baseline for performance. Subsequently, feature selection is introduced to mitigate overfitting, with a focus on improving model performance, interpretability, and computational efficiency. However, results indicate a slight decline in the model's performance, emphasizing the challenges of discarding important features during the selection process. Hyperparameter tuning follows, utilizing GridSearchCV with time series cross-validation, leading to the selection of the best hyperparameter (k=27), revealing that the feature selection technique did not significantly enhance the Linear Regression model.

The exploration proceeds to regularization techniques, specifically Ridge and Lasso Regression. Despite adjusting hyperparameters, both techniques show limited impact on improving Mean Squared Error (MSE) or R2 score. This prompts a critical evaluation of the suitability of Linear Regression for accurately predicting the 'Number of Orders.' The lack of success with linear regression prompts the consideration of more advanced modeling techniques, starting with a Decision Tree Regressor in the following step.

2- Decision Tree Regressor

The Decision Tree Regressor is introduced to capture non-linear relationships between features and the number of orders, potentially outperforming linear regression. The model exhibits better performance with lower Mean Squared Error (MSE) and Mean Absolute Error (MAE), along with a slightly higher R2 score compared to linear regression. This improvement is attributed to the decision tree's ability to handle complex, non-linear relationships. Hyperparameter tuning is performed resulting in improved performance, reduced overfitting, and better generalization compared to the default model. The success of random search prompts the exploration of ensemble methods like Random Forest in the subsequent step to potentially enhance model performance further.

3- Random Forest model

The Random Forest model which is an ensemble method, outperforms the Decision Tree model by capturing more complex relationships in the data. It demonstrates lower Mean Squared Error (MSE) and Mean Absolute Error (MAE), along with a higher R2 score, indicating better predictive accuracy and model generalization. Despite the increased computation time, the superior performance justifies the use of Random Forest.

Hyperparameter tuning is also explored to further enhance model performance. The Random Forest model with RandomSearch hyperparameter tuning produced slightly better results compared to the default Random Forest model. However, this improvement comes at the cost of increased computation time, which more than quadrupled compared to the default model.

Next, the XGBoost model is examined to assess potential improvements in performance.

4- XGBoost model

XGBoost, a gradient boosting algorithm, utilizes an ensemble of weak learners, typically decision trees, to sequentially correct errors and build a robust predictive model. Incorporating regularization techniques and tree pruning, XGBoost prevents overfitting and enhances efficiency through parallel and distributed computing. Despite Random Forest's superior performance, XGBoost stands out for its computational efficiency.

Hyperparameter tuning for XGBoost involves parameters like 'n_estimators,' 'learning_rate,' 'max_depth,' 'min_child_weight,' 'gamma,' and 'colsample_bytree.' Randomized Search optimizes these hyperparameters, yielding a tuned XGBoost model with improved predictive accuracy. The best hyperparameters result in a lower Mean Squared Error, Mean Absolute Error, and a higher R2 score, justifying the increased computation time. The tuned XGBoost model presents a compelling choice for accurate and efficient predictions.

4.3Final Model Selection

In summary, the modeling process encompassed the exploration of diverse algorithms, such as linear regression, decision tree, random forest, and XGBoost. Each model underwent hyperparameter tuning to optimize its performance. The linear regression model, while simple, may struggle to capture complex data relationships. The decision tree model exhibited improved predictive accuracy compared to linear regression. However, both random forest and XGBoost outperformed the individual decision tree model. While random forest demonstrated strong predictive accuracy, XGBoost, particularly with hyperparameter tuning, surpassed all models, presenting a lower Mean Squared Error, Mean Absolute Error, and a higher R2 score.

The choice of the best model hinges on specific problem requirements. If computational efficiency is a priority, the random forest model with default settings might be preferred due to its respectable performance and relatively lower computation time. However, if maximizing predictive accuracy is paramount and computational resources permit, the tuned XGBoost model emerges as the top performer among the considered algorithms.

Consequently, we will employ the tuned XGBoost model as our preferred choice.

4.4Refit the Model on the Entire Training Dataset

In this step, we aim to refit the model using the entire training set and subsequently utilize the tunned XGBoost model to predict the number of orders for the test set.

The model seems to generalize well to unseen data, indicated by the lower MSE (22,870) and MAE (91) on the larger dataset. However, the slight reduction in the R2 score (0.70) suggests that the model's explanatory power is marginally compromised when applied to the larger dataset. It's important to acknowledge that estimates of model performance are susceptible to the noise and uncertainty inherent in the data. Figure 4-1 displays the actual vs. predicted number of orders for each week, center ID, and meal ID.

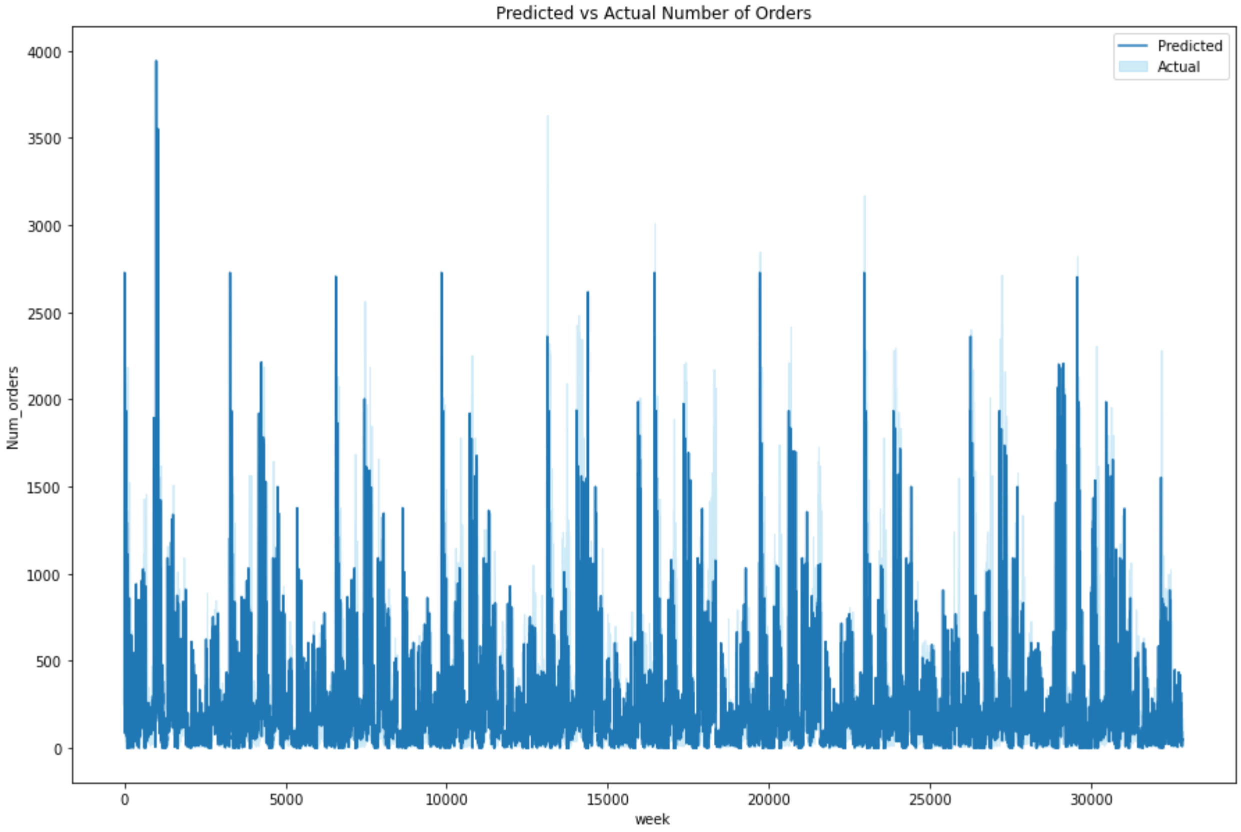


Figure 4-1: Actual vs Predicted Number of Orders

Figure 4-2 to 4-4 show the total actual vs. predicted number of orders for weeks, center IDs, and meal IDs, respectively.

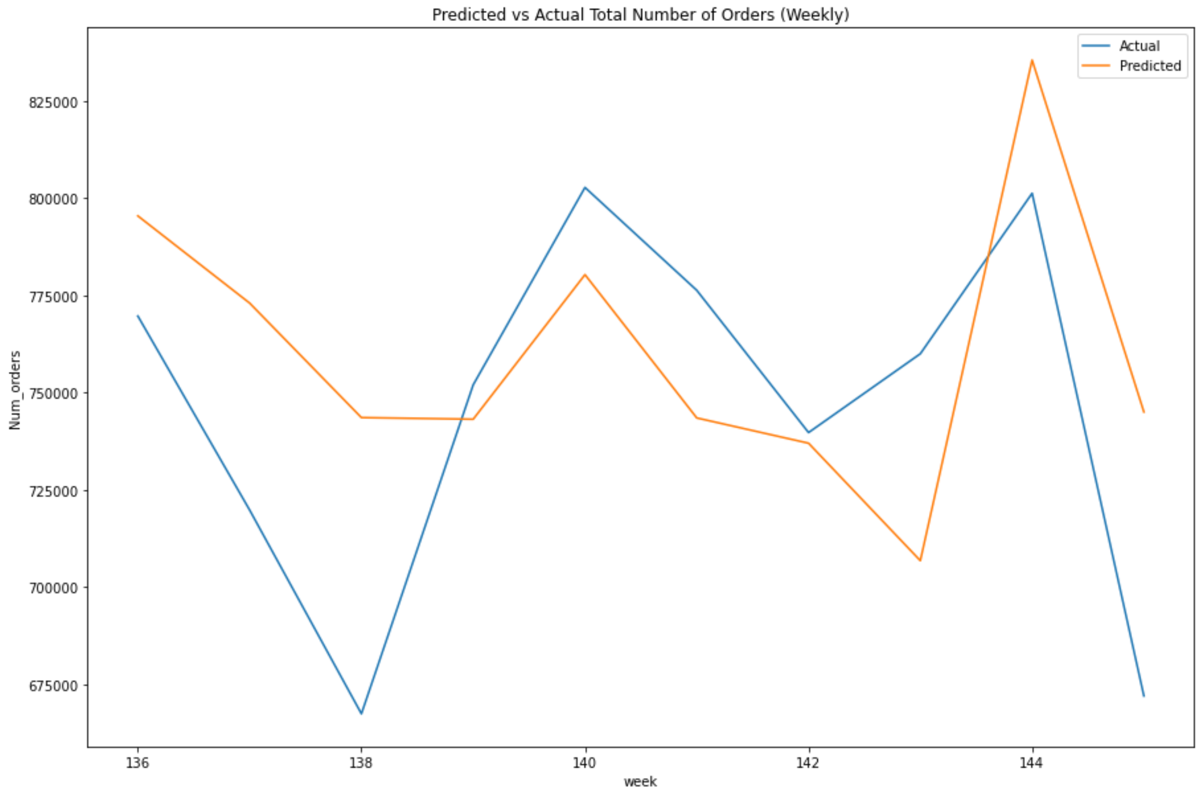


Figure 4-2: Weekly Total Actual vs Predicted Number of Orders

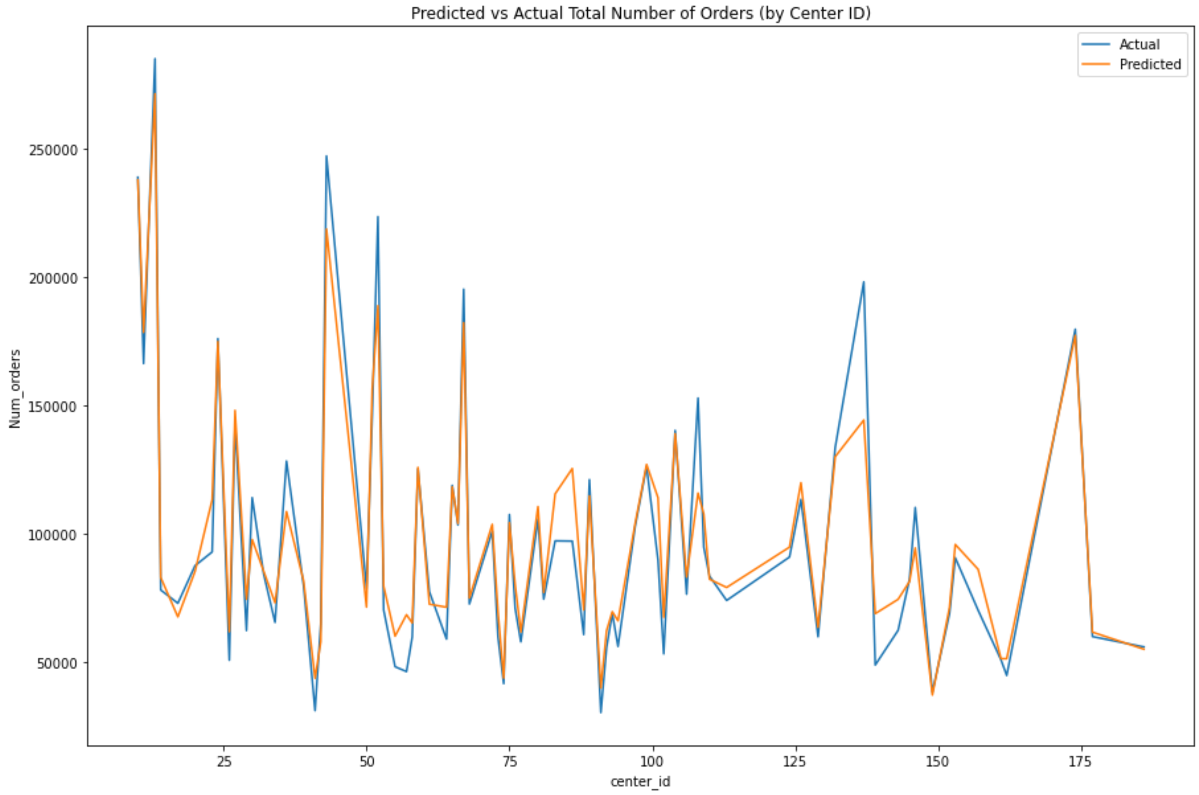


Figure 4-3: Total Actual vs Predicted Number of Orders by Center Ids

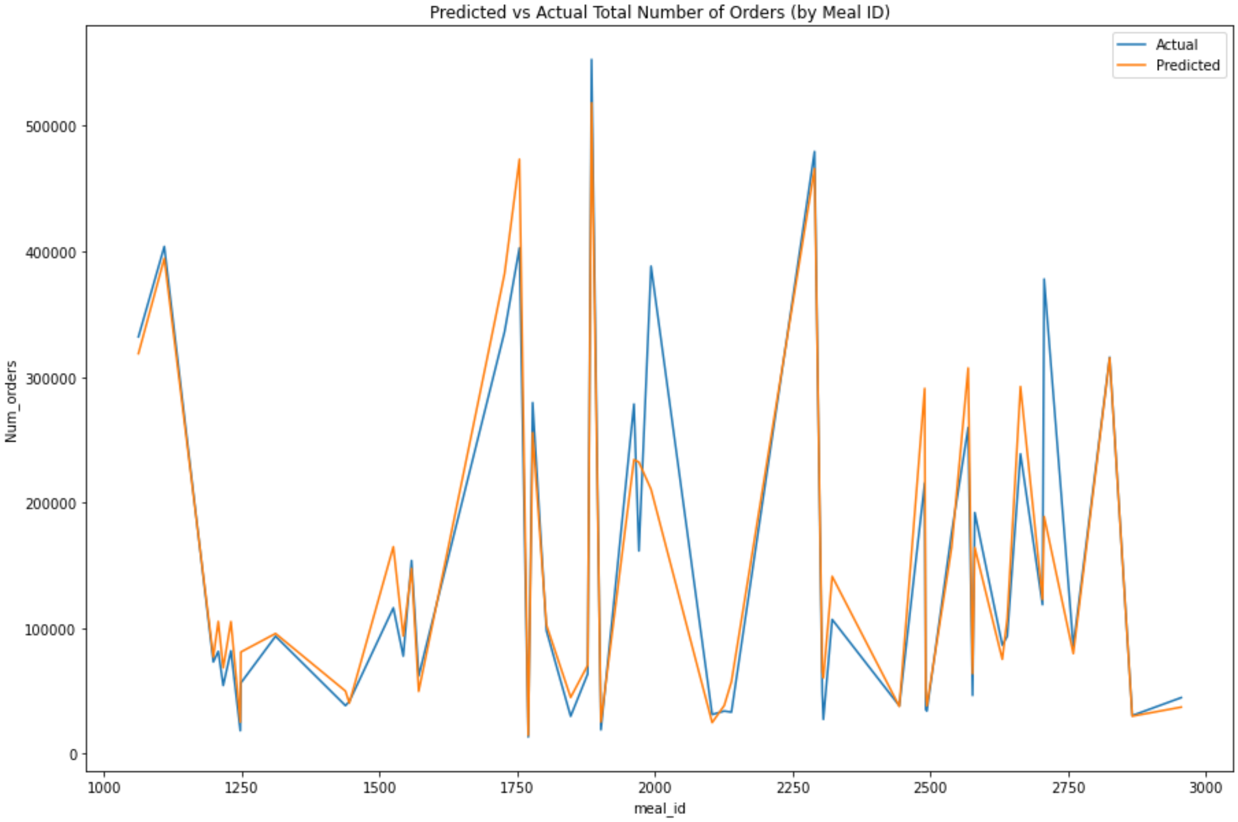


Figure 4-4: Total Actual vs Predicted Number of Orders by Meal Ids

The above figures represent the actual vs. predicted number of orders for weeks, center IDs, and meal IDs. The model seems to capture the overall behavior of the data, and we observe a quite good match between the predicted and actual values.

4.5Feature Importance

Variable importance in XGBoost allows us to understand the contribution of each feature to the model's predictions effectively. The model calculates importance scores based on how often a feature is used to split the data across all trees in the ensemble and how much each split improves the model's performance. Higher importance scores suggest that a feature plays a more critical role in making accurate predictions. Figure 4-5 shows the variable importance of our model.

- meal_category and cuisine: features like C_Rice_Bowl, and C_Thai, among others, indicate that their presence significantly influences the predictions.
- homepage_featured and emailer_for_promotion: They seem to be significant contributors to the model, indicating that whether a meal is featured on the homepage or promoted via email strongly influences the predictions.
- op_area, and checkout_price: These numerical features have a moderate impact on predictions.

Other Features like region_code, center_type, city_code, week, and base_price contribute to the model's predictions but are not among the important features.

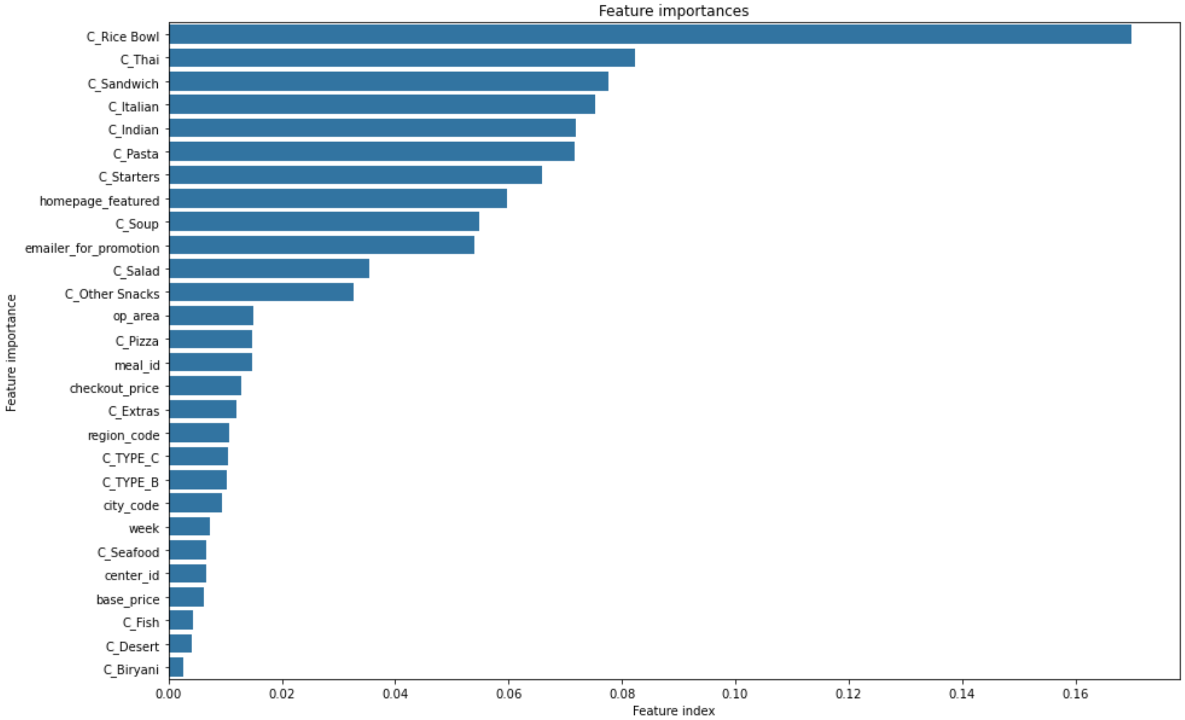


Figure 4-5: Feature Importance

5. Future direction:

While the presented models strike a balance between performance and computation time, it's essential to consider avenues for further improvement:

- Utilizing more data: Currently, we have data spanning three years. Exploring additional data from another year or two can enhance the model's predictive capabilities.
- Incorporating feature engineering: Employing advanced feature engineering techniques can contribute to refining the model's performance.
- Exploring other ensemble models: Experimenting with alternative ensemble models, such as the Adaboost model, could provide valuable insights and potentially lead to improved results.