Machine Learning Report

Hussain Kaide Johar Manasi

Food Demand Prediction

Problem Statement

In the rapidly evolving landscape of the meal delivery service industry, the ability to accurately forecast meal orders is crucial for ensuring efficient operations and meeting customer expectations. The problem at hand involves developing robust predictive models to forecast the number of meal orders for each meal and fulfillment center combination in the forthcoming weeks. This task is essential for optimizing inventory management, minimizing costs, and enhancing overall service quality.

Objective

The primary objective of this project is to develop predictive models capable of accurately forecasting the number of meal orders for each meal and fulfillment center combination in the upcoming weeks. By leveraging historical data and advanced analytics techniques, the aim is to improve operational efficiency, reduce inventory costs, and enhance customer satisfaction. Ultimately, the goal is to provide actionable insights that enable the meal delivery service to optimize its operations and deliver exceptional service to its customers.

Challenges

The meal delivery service industry faces several challenges that complicate the task of forecasting meal orders:

1. Seasonality and Trends: Demand for meal orders can vary significantly based on seasonal factors, holidays, and emerging trends in consumer preferences. Capturing these variations accurately is essential for effective forecasting.
2. Data Sparsity: Historical data may be incomplete or unevenly distributed across meal categories, fulfillment centers, or time periods. Handling missing or sparse data effectively is crucial for building reliable predictive models.
3. Complex Relationships: The relationship between meal orders and various factors such as meal category, cuisine type, pricing strategies, and fulfillment center location can be intricate and nonlinear. Understanding and modeling these relationships accurately pose significant challenges.
4. Operational Constraints: Practical considerations such as production capacity, delivery logistics, and inventory constraints add another layer of complexity to the forecasting task. Balancing these constraints while meeting customer demand is essential for the success of the meal delivery service.

Dataset Information

Source: https://www.kaggle.com/datasets/kannanaikkal/food-demand-forecasting/data

Meals Dataset

The Meals Dataset serves as a foundational component for our predictive modeling efforts. It contains a comprehensive array of information pertaining to meal categories, cuisine details, and other relevant attributes associated with each meal. This dataset enables us to contextualize meal orders within the broader spectrum of meal types and cuisines offered by the delivery service. By analyzing the characteristics of different meal categories and cuisines, we gain valuable insights into the factors that influence customer preferences and ordering behavior.

Shape: (51, 3)
Columns: ['meal_id', 'category', 'cuisine']

Centers Dataset

The Centers Dataset provides essential insights into the fulfillment centers that serve as the logistical backbone of the meal delivery service. This dataset includes detailed information about each fulfillment center, such as geographical location, operational capacity, and other pertinent attributes. By examining the distribution and characteristics of fulfillment centers, we can better understand the logistical dynamics underlying meal delivery operations. This understanding is crucial for optimizing delivery routes, managing inventory levels, and ensuring timely order fulfillment.

Shape: (77, 5)
Columns: ['center_id', 'city_code', 'region_code', 'center_type', 'op_area']

Sales Dataset

The Sales Dataset constitutes the core source of historical transactional data that forms the basis of our predictive modeling endeavors. This dataset encompasses a wealth of information regarding past meal orders, including the number of orders for each meal from each fulfillment center, as well as associated attributes such as base price, checkout price, and transaction details. By analyzing patterns and trends within this rich dataset, we can uncover valuable insights into past ordering behavior, identify seasonal trends, and discern underlying patterns that drive meal order volumes. This

historical perspective is instrumental in informing our predictive modeling efforts and enhancing the accuracy of our forecasts.

Shape: (456548, 21)
Columns: ['id', 'week', 'center_id', 'meal_id', 'checkout_price', 'base_price', 'emailer_for_promotion', 'homepage_featured', 'num_orders'

Preprocessing

Feature Engineering

Feature engineering is a crucial aspect of the data preprocessing pipeline, aimed at extracting meaningful insights and enhancing the predictive power of our models. In this section, we highlight two key feature engineering techniques employed in our analysis: the creation of discount amount and percentage, and the assignment of quarters to the weeks for improved time series analysis.

1. Creation of Discount Amount and Percentage:

One of the key insights we aim to capture in our predictive models is the impact of discounts on meal orders. To achieve this, we create two new features: discount amount and discount percentage.

- Discount Amount: This feature represents the monetary value of the discount applied to each meal order. It is calculated by subtracting the checkout price from the base price, providing insight into the magnitude of the discount offered.
- Discount Percentage: This feature quantifies the discount as a percentage of the base price. It is calculated by dividing the discount amount by the base price and multiplying by 100, enabling us to assess the relative impact of discounts across different meal categories and fulfillment centers.

By incorporating these features into our predictive models, we can assess the influence of discounts on meal orders and identify optimal pricing strategies to maximize customer engagement and revenue generation.

2. Assigning Quarters to the Weeks:

Another important aspect of our feature engineering process involves assigning quarters to the weeks in our dataset. This enables us to perform more granular time series analysis and capture seasonal variations in meal orders.

- Quarter Assignment: We divide the 150 weeks in our dataset into four quarters, each comprising approximately 37 weeks. This segmentation allows us to analyze trends and patterns in meal orders at a quarterly level, facilitating more nuanced insights into seasonal fluctuations and long-term trends.
- Temporal Analysis: By organizing the data into quarters, we can conduct time series analysis at a higher level of granularity, identifying seasonal peaks and troughs in meal orders and discerning recurring patterns over time. This information is invaluable for informing strategic decision-making and resource allocation within the meal delivery service.

Encodings

Categorical variables such as cities in the Fulfillment Center dataset, center_id, meal_id, and region_codes were one-hot encoded to prepare them for integration into our machine learning models. This transformation converts categorical variables into binary vectors, where each category is represented by a binary flag indicating its presence or absence. For example, if there are N unique categories in a variable, one-hot encoding creates N binary columns, with each column representing a distinct category. By doing so, we effectively encode categorical information in a format that machine learning algorithms can readily process. This ensures that all relevant categorical attributes contribute meaningfully to our predictive models without introducing biases or inaccuracies.

Normalization

To ensure uniformity in scale and prevent certain features from dominating others during model training, we applied the StandardScaler normalization technique to the relevant numerical attributes in the dataset. StandardScaler transforms numerical features such that they have a mean of zero and a standard deviation of one. By scaling the features to a common scale, StandardScaler facilitates more stable and efficient model training, particularly for algorithms sensitive to variations in feature scales, such as gradient descent-based optimization methods. Normalization also aids in improving the convergence rate of iterative algorithms and enhances the interpretability of model coefficients by making them directly comparable across features.

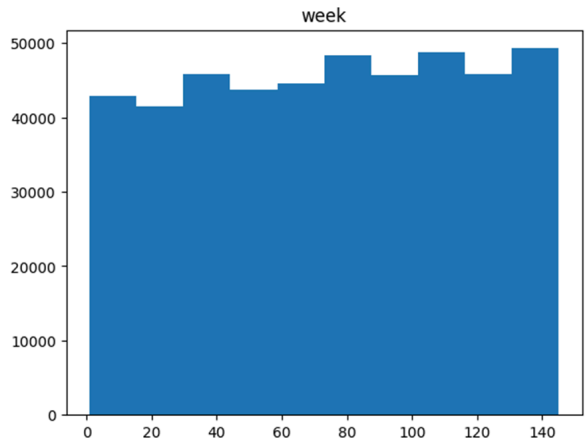
Data Handling

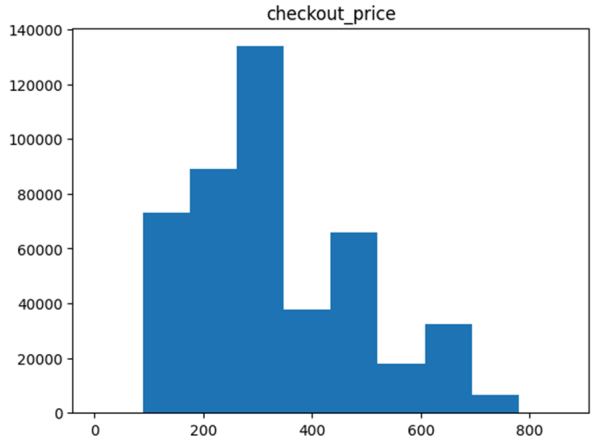
Outliers in numerical features, such as base price and checkout price, were addressed using log transformations to mitigate their impact on model performance. Outliers, or extreme values, can distort statistical analyses and modeling results if left unaddressed. Logarithmic transformations compress the scale of the data, reducing the influence of extreme values while preserving the overall distribution. By applying log transformations to numerical features with skewed distributions or containing outliers, we ensure that our predictive models are more robust and resilient to the effects of anomalous data points. This preprocessing step enhances the stability and accuracy of our models, resulting in more reliable predictions and insights for decision-making purposes.

Exploratory Data Analysis

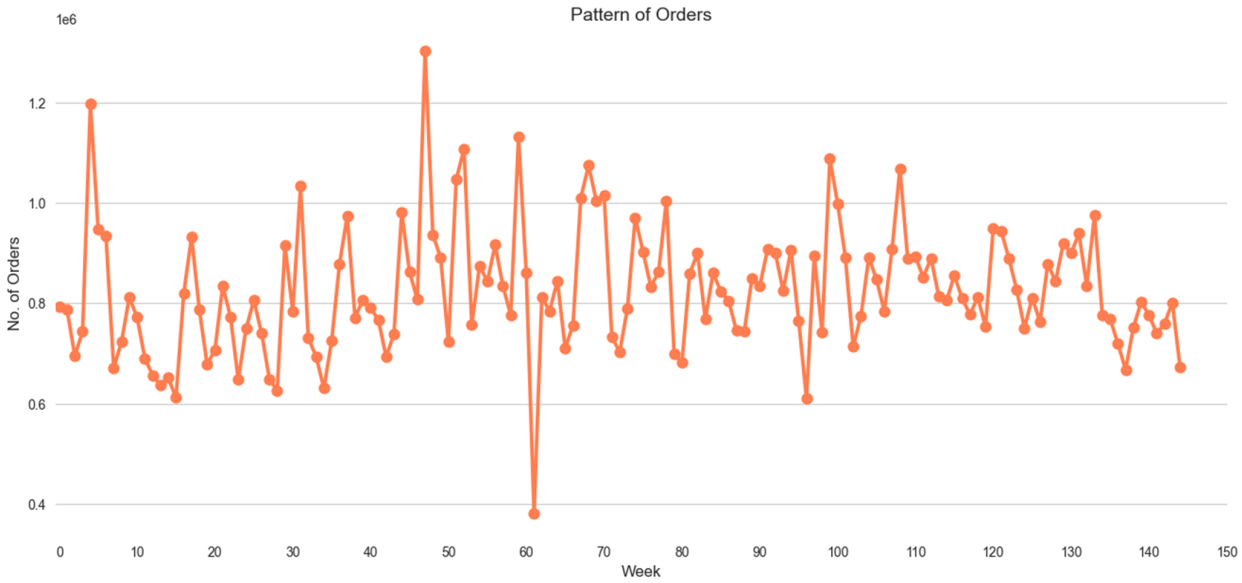
Key Insights:

1. We can see that the order details are evenly distributed across the weeks. And that the orders are mostly priced in between 200 and 400 Rupees. All prices are in Rupees.

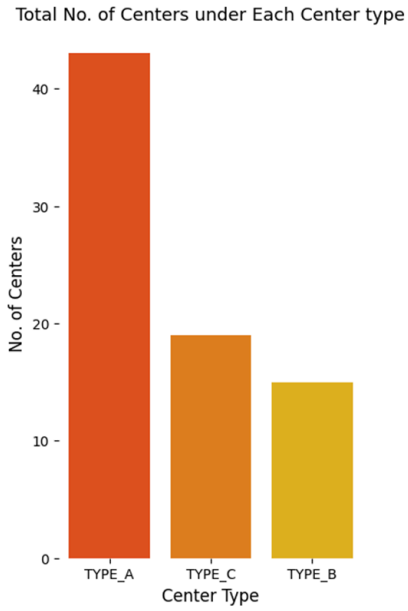


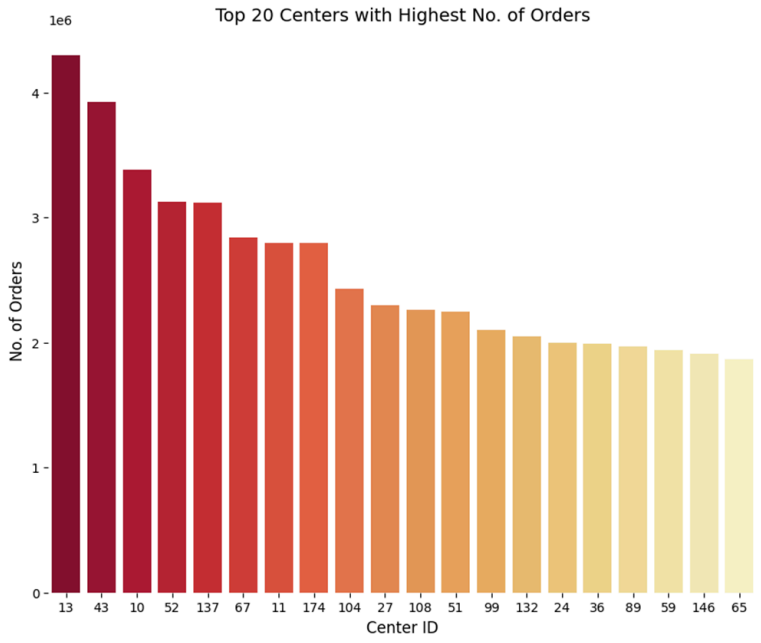


2. The line plot shows the ups and downs of number of orders over the weeks. We can see that there is no trend, nor seasonality.

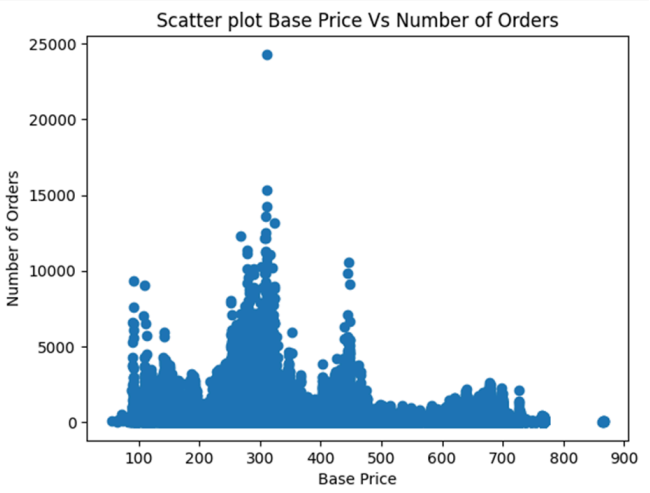


3. The below bar graphs illustrate the details of Centers. Specifically the type of centers, and the number of orders to the top centers. We can see that the most common type of center is “TYPE_A”, and the highest number of orders are entertained by center 13. Note that high number of orders does not indicate higher revenue and has no indication towards pricing.

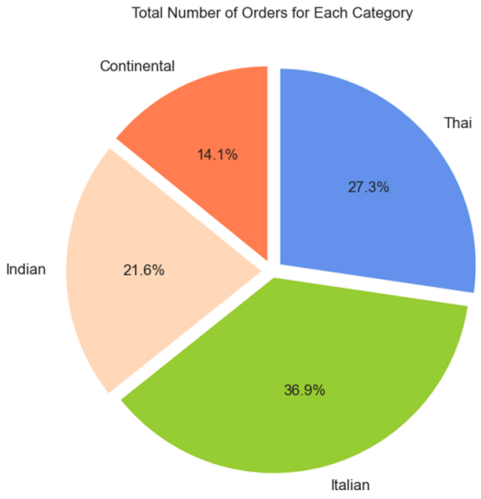


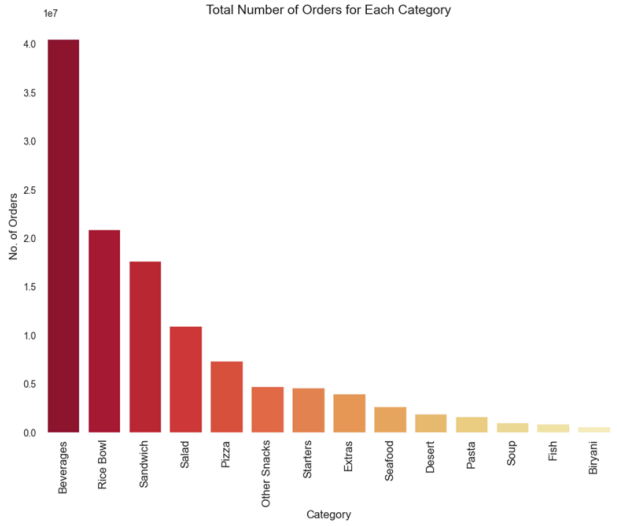


4. We can see that most of the orders are priced between 250 and 350 Rupees, with significant price fluctuations. Additionally, it seems that most orders receive a discount upto 150 Rupees, with discount frequency going down after that.

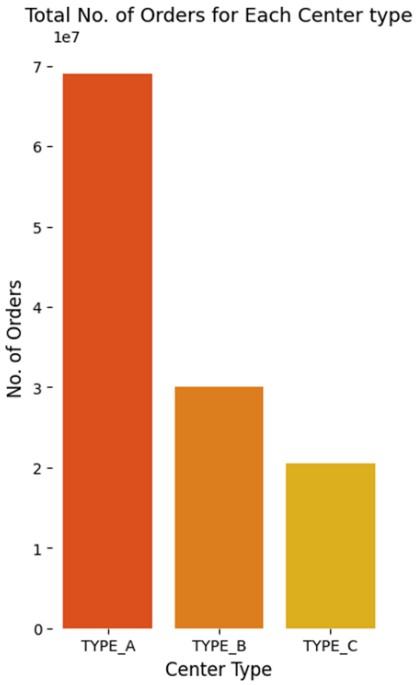


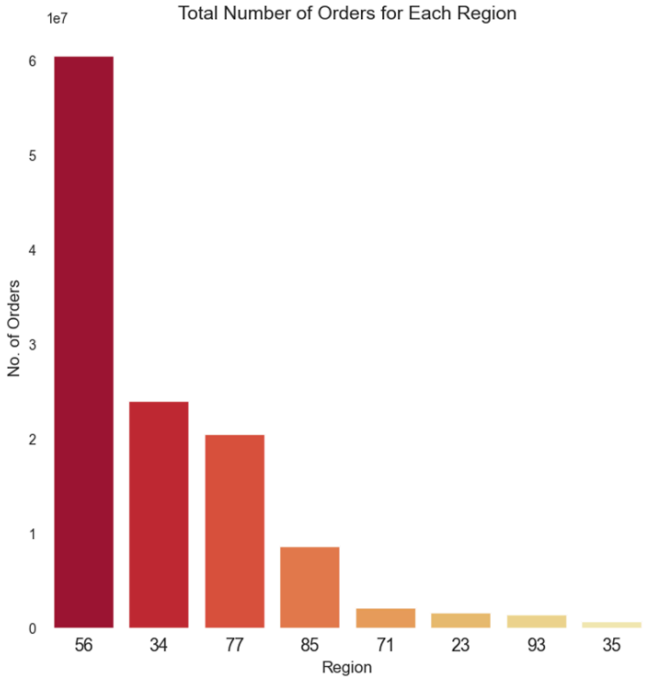
5. While most of the meals are Italian, we can also see a high number of Thai and Indian cuisines in the meals that are ordered. It is interesting to note that most of the orders are beverages, followed by mixed categories of Indian, Thai, and Italian options.





6. In addition to being most common center, TYPE_A is also the center with the most orders taken. Also, region code 56 seems to be ordering the most meals.





Models Used

In our use case, predictive modeling serves as the foundation for improving order forecasting accuracy. By exploring various modeling techniques, we aim to identify the most effective approaches for predicting meal orders based on factors such as meal categories, cuisines, and fulfillment center locations. Ultimately, our goal is to leverage predictive modeling to optimize resource allocation, minimize costs, and ensure timely delivery to customers. Through this process, we aim to extract actionable insights that drive operational efficiency and deliver tangible value to the meal delivery service.

Linear Regression

Linear regression is a fundamental statistical technique used for modeling the relationship between a dependent variable and one or more independent variables. In the context of our meal order forecasting task, linear regression models seek to establish a linear relationship between predictor variables (such as meal category, cuisine details, and pricing information) and the target variable (number of meal orders). While simple and interpretable, linear regression assumes a linear relationship between variables, which may not always hold true for complex real-world data.

Decision Tree

Decision trees are versatile machine learning models that recursively partition the feature space into subsets based on simple decision rules. Each internal node of the tree represents a decision based on a feature, leading to a splitting of the data into distinct branches. In the context of meal order forecasting, decision trees can capture nonlinear relationships between predictors and target variables, making them well-suited for handling complex data patterns. However, decision trees are prone to overfitting and may struggle with generalization on unseen data without proper regularization.

KNN Regressor

The K-nearest neighbors (KNN) algorithm is a non-parametric method used for regression tasks. In KNN regression, predictions are made by averaging the target values of the K nearest neighbors in the feature space. KNN is intuitive and flexible, as it does not assume any underlying functional form for the data. However, it can be computationally expensive, especially with large datasets, and may not perform well in high-dimensional spaces.

Ridge and Lasso Regressions

Ridge and Lasso regressions are regularization techniques used to address multicollinearity and overfitting in linear regression models. Ridge regression adds a penalty term to the least squares objective function, while Lasso regression imposes a penalty based on the absolute value of the coefficients. These techniques help prevent overfitting by shrinking the coefficients towards zero, effectively simplifying the model and improving its generalization performance.

Random Forest Regressor

Random forests are ensemble learning methods that combine multiple decision trees to improve predictive performance. Each tree in the forest is trained on a bootstrap sample of the data and makes independent predictions. The final prediction is obtained by averaging or taking the mode of the predictions from individual trees. Random forests are robust, scalable, and capable of capturing complex interactions between features. They are less prone to overfitting compared to individual decision trees and can handle high-dimensional data effectively.

CatBoost

CatBoost is a gradient boosting algorithm specifically designed for categorical feature handling and efficient training on large-scale datasets. It utilizes a novel algorithm for handling categorical variables, eliminating the need for manual preprocessing such as one-hot encoding. CatBoost automatically handles categorical variables internally and provides robust performance with minimal hyperparameter tuning. It is particularly well-suited for our meal order forecasting task, where categorical features such as meal categories and fulfillment center locations play a significant role.

XGBoost Regressor

XGBoost (Extreme Gradient Boosting) is a popular gradient boosting algorithm known for its efficiency, scalability, and high predictive accuracy. It builds a series of decision trees sequentially, where each subsequent tree corrects the errors of the previous ones. XGBoost employs advanced regularization techniques to prevent overfitting and achieve superior performance on a wide range of predictive modeling tasks. With its ability to handle missing values, categorical features, and large datasets efficiently, XGBoost is a versatile choice for meal order forecasting.

LightGBM

LightGBM is another gradient boosting algorithm known for its speed and efficiency in handling large-scale datasets. It adopts a novel technique called gradient-based one-side sampling (GOSS) to reduce memory usage and speed up training. LightGBM uses a leaf-wise tree growth strategy and employs histogram-based algorithms for binning continuous features, resulting in faster computation and improved accuracy. With its ability to handle categorical features, missing values, and large datasets effectively, LightGBM is well-suited for our meal order forecasting task.

Results and Evaluation

In the evaluation of predictive models for meal order forecasting, we adopt a structured approach that encompasses benchmarking against linear regression, leveraging base models such as KNN and decision trees, exploring regularization techniques with Ridge and Lasso Regression, and harnessing the predictive power of ensemble methods including Random Forests, CatBoost, XGBoost, and LightGBM.

Benchmarking with Linear Regression:

Linear regression serves as the benchmark model against which the performance of more complex algorithms is compared. Its simplicity and interpretability make it an ideal starting point for analysis. By fitting a linear relationship between predictor variables and the target variable (meal orders), linear regression provides a baseline for assessing the predictive capabilities of more sophisticated models. Despite its simplicity, linear regression offers valuable insights into the linear relationships present in the data.

Base Models: KNN and Decision Trees:

K-nearest neighbors (KNN) and decision trees are employed as base models to capture nonlinear relationships and complex interactions within the data. KNN makes predictions based on the similarity of input data points, while decision trees recursively partition the feature space to make predictions. These base models serve as foundational components for more advanced techniques, providing a starting point for model building and evaluation.

Regularization Techniques: Ridge and Lasso Regression:

Building on linear regression, we explore regularization techniques such as Ridge and Lasso Regression to address multicollinearity and overfitting. Ridge regression adds a penalty term to the least squares objective function, while Lasso regression imposes a penalty based on the absolute value of the coefficients. By controlling the complexity of the model and shrinking the coefficients towards zero, Ridge and Lasso Regression enhance generalization performance and improve predictive accuracy.

Ensemble Methods: Random Forests, CatBoost, XGBoost, and LightGBM:

To further improve predictive performance, ensemble methods are employed, leveraging the collective wisdom of multiple decision trees. Random Forests aggregate predictions from an ensemble of decision trees to produce more robust and accurate forecasts. CatBoost, XGBoost, and LightGBM are gradient boosting algorithms that sequentially build decision trees, where each subsequent tree corrects the errors of the previous ones. By combining the predictions of multiple weak learners, ensemble methods enhance predictive accuracy and robustness.

Parameter Limitations:

To streamline execution time and computational resources, model parameters are constrained. Specifically, the number of estimators is limited to 100, the maximum depth of decision trees is set to 10, and a learning rate of 0.5 is enforced. These

parameter constraints strike a balance between model complexity and computational efficiency, ensuring timely model training and evaluation without compromising predictive performance.

Model	MAE	MSE	R2
Linear Regression	0.494	0.390	0.704
Decision Tree	0.534	0.462	0.649
KNN Regressor	0.428	0.314	0.762
Ridge and Lasso Regression	0.494 0.562	0.390 / 0.498	0.704 / 0.622
Random Forest Regressor	0.522	0.438	0.668
CatBoost	0.381	0.247	0.812
XGBoost Regressor	0.393	0.260	0.802
LightGBM	0.392	0.257	0.804

Conclusions

Takeaways

1. Model Performance: The comparative analysis of CatBoost, XGBoost Regressor, and LightGBM reveals that all three models yield competitive results in terms of Mean Absolute Error (MAE), Mean Squared Error (MSE), and R-squared (R2) scores. These metrics indicate the accuracy and predictive power of the models in forecasting meal orders for the meal delivery service.

2. Model Robustness: Despite minor differences in performance metrics, the consistency in performance across CatBoost, XGBoost Regressor, and LightGBM underscores the robustness of these models. Their ability to consistently deliver accurate predictions across different evaluation metrics reaffirms their suitability for addressing the forecasting challenges inherent in the meal delivery service industry.
3. Feature Importance: An additional takeaway involves analyzing feature importance across the models to identify the most influential factors driving meal orders. By examining the relative importance of features such as meal category, cuisine details, pricing information, and fulfillment center locations, the meal delivery service can gain insights into customer preferences and behavior, enabling targeted marketing strategies and product offerings.

Business Impact

1. Optimized Inventory Management: Accurate forecasting of meal orders enables the meal delivery service to optimize inventory management processes. By anticipating demand fluctuations with high precision, the service can minimize stockouts, reduce excess inventory, and ensure adequate supply levels to meet customer demand. This results in cost savings and operational efficiencies by minimizing wastage and storage costs associated with excess inventory.
2. Informed Decision-Making: The insights derived from predictive modeling not only inform operational decisions related to inventory management and logistics but also guide strategic planning and business development initiatives. By leveraging predictive analytics, the meal delivery service can identify emerging trends, anticipate market demand shifts, and capitalize on growth opportunities proactively. Informed decision-making based on data-driven insights enables the service to stay ahead of the competition and adapt to evolving customer preferences and market dynamics effectively.

In conclusion, the successful application of CatBoost, XGBoost Regressor, and LightGBM in forecasting meal orders underscores their significance in driving operational efficiency, enhancing customer satisfaction, and informing strategic decision-making within the meal delivery service industry. By harnessing the power of predictive analytics, the service can unlock actionable insights that drive tangible business impact, positioning it for sustained success and growth in a competitive market landscape.