**Dominos - Predictive Purchase Order System**

* **Problem Statement:**

Dominos wants to optimize the process of ordering ingredients by predicting future sales and creating a purchase order. By accurately forecasting sales, Dominos can ensure that it has the right amount of ingredients in stock, minimizing waste and preventing stockouts. This project aims to leverage historical sales data and ingredient information to develop a predictive model and generate an efficient purchase order system.

* **Objective:**

to develop an automated Predictive Purchase Order System for Domino's that accurately forecasts weekly pizza sales and calculates ingredient requirements. This system aims to optimize inventory management by generating precise purchase orders based on demand projections, thereby reducing waste, avoiding stockouts, and improving cost efficiency. Using time series forecasting models, the project seeks to align inventory levels with sales demand patterns, streamline the ordering process, and enhance overall operational efficiency for Domino's supply chain.

* **Datasets:**

**The project involves two datasets:** **Pizza Sales** and **Pizza Ingredients**.

* The **Pizza Sales Dataset** contains 48,620 entries which shows detail information about an individual sale. By reading these dataset columns we can get information such as pizza\_id which is a unique identifier for the sale, order\_id that is linking to a specific order, pizza\_name\_id shows an unique identifier for each pizza, quantity gives information about the number of pizzas sold, When the sale occurred is showed by order\_date and order\_time , unit\_price and total\_price as well as pizza\_size and pizza\_category . Basically this dataset contains detail information in view of sales, covering pricing, timing, and pizza characteristics, etc..
* The **Pizza Ingredients Dataset** contains of 518 entries which describe details about the ingredients for various pizzas. It includes pizza\_name\_id ,  pizza\_name ,  pizza\_ingredients , i.e., list of ingredients and Items\_Qty\_In\_Grams, i.e.,the quantity of each ingredient used. This dataset provides information regarding the composition of each pizza and the amounts of ingredients required to make the pizza.
* **Metrics:**

Metrics are critical for evaluating the accuracy and effectiveness of the forecasting models used in this project. The primary metric used is:

* **Mean Absolute Percentage Error (MAPE)**: This measures the accuracy of the forecasts. A lower MAPE indicates more accurate predictions, which is crucial for effective inventory management and minimizing ingredient wastage.
* **Business Use Cases:**

Inventory Management: Ensuring optimal stock levels to meet future demand without overstocking.

Cost Reduction: Minimizing waste and reducing costs associated with expired or excess inventory.

Sales Forecasting: Accurately predicting sales trends to inform business strategies and promotions.

Supply Chain Optimization: Streamlining the ordering process to align with predicted sales and avoid disruptions.

* **Approach:**

The approach involves using time series forecasting and machine learning models to build a reliable and scalable solution for inventory and order management.

1. **Data Preprocessing and Exploration**

Data preprocessing and exploration were done to ensure data quality and prepare features for model training.

**Data Cleaning**

The data cleaning process included:

* **Handling Missing Values**:

Detected missing values, Replaced missing values using mean, median, mode, or placeholders, Removed columns or rows with excessive missing data if necessary.

* **Removing Duplicates**:
* Duplicate rows and redundant columns were removed to prevent data redundancy and ensure data consistency and Fixed inconsistencies, such as standardizing text and correcting typos.
* **Handling Outliers**:

Identified outliers using statistical methods or visualizations, Removed, transformed, or categorized outliers based on their impact.

* **Date Standardization**: Multiple date formats were standardized to improve time-based feature engineering.

1. **Exploratory Data Analysis (EDA)**

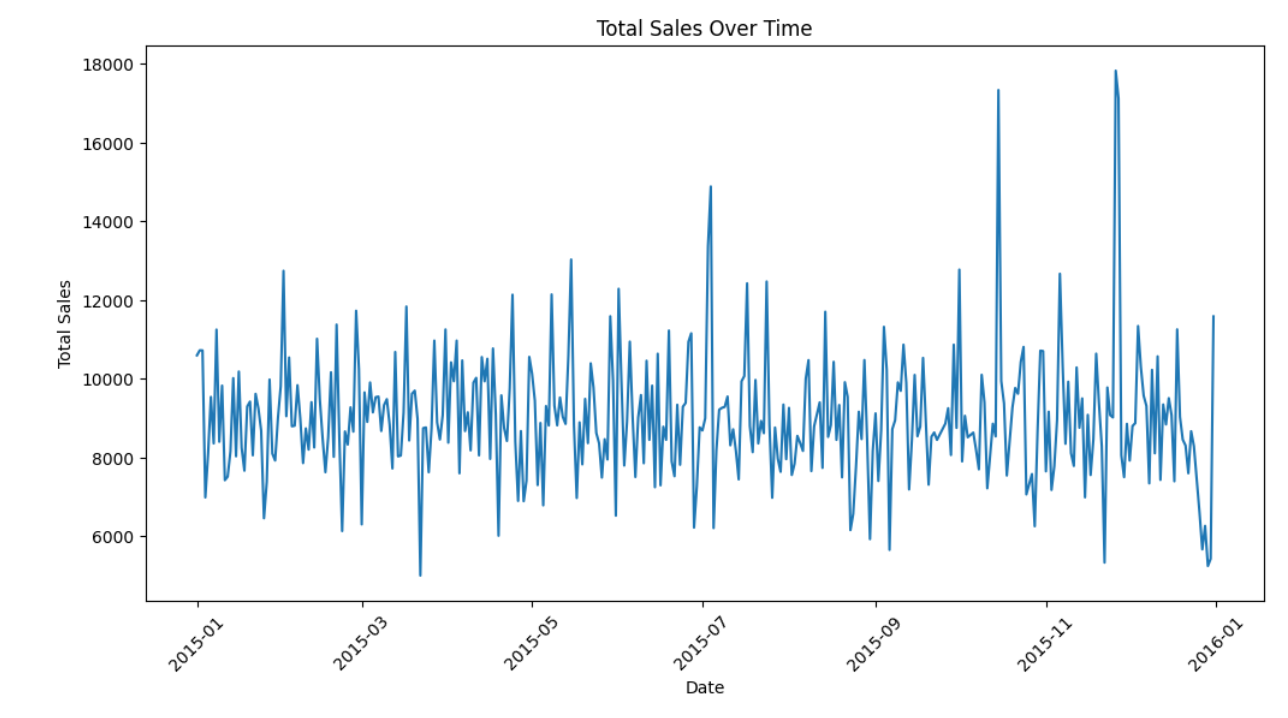
The EDA conducted for Domino's Predictive Purchase Order System focuses on understanding the sales trends, identifying seasonal patterns, and creating new features to enhance forecasting accuracy. This report outlines the main findings from the EDA, particularly focusing on feature engineering to capture demand drivers such as weekly patterns, promotions, and holiday effects.

EDA was performed to gain insights into sales patterns and ingredient usage:

* **Time Series Visualization**: Plotted daily and weekly sales trends to identify seasonality and growth patterns.
* **Category Analysis**: Examined pizza categories to determine popular types and peak sales times.
* **Correlation Analysis**: Investigated relationships between features such as quantity, total\_price, and seasonal factors (e.g., weekends, holidays).

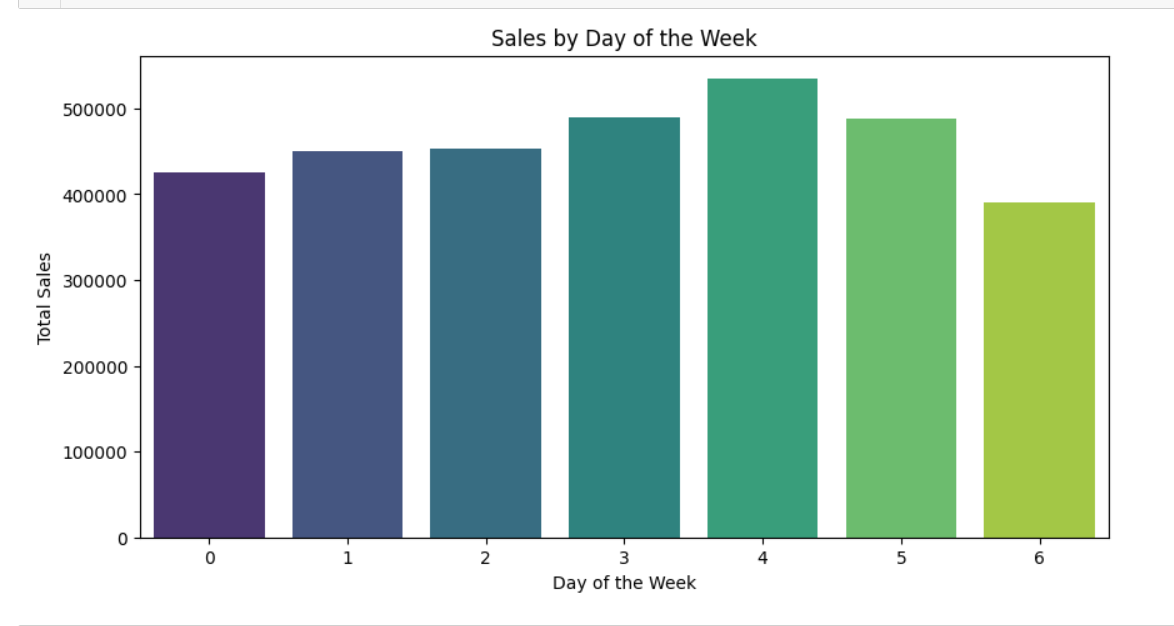
**Visualization of the data to identify significant features:**

* 1. **Sales Trends Over Time**



* **Daily and Weekly Aggregations:** Sales data was aggregated by day and by week, revealing clear weekly seasonality.
* **Visualization:** Line plots of daily and weekly sales indicated consistent peaks and valleys, with notable increases in sales on weekends and during holiday periods.

**B. Sales by Day of the Week**



* **Bar Chart Analysis:** Sales were consistently higher on Saturdays and Sundays, with lower sales typically on Mondays and Tuesdays.
* **Implication:** Including "day of the week" as a feature in the forecasting model could help capture this weekly pattern, improving forecast accuracy.

**C. Monthly Sales Patterns**

A graph of sales by month

Description automatically generated

* **Box Plot by Month:** Box plots showing sales distribution per month highlighted higher demand in December and early fall months.
* **Insight:** Monthly seasonality could be incorporated to better anticipate monthly variations in sales.

**D. Effect of Promotional Periods**

A graph of sales during promotional periods

Description automatically generated

* **Sales Boost:** Comparison of average sales during promotional vs. non-promotional periods showed a noticeable uplift in sales during promotions.
* **Insight:** Promotions are a strong driver of sales, and including this feature in the model can help in capturing these spikes more effectively.

**E. Impact of Holidays**

A graph of sales on holidays

Description automatically generated

* **Holiday Effect Visualization**: Holidays and pre-holiday periods showed significant increases in sales, as seen in spikes around Christmas and New Year.
* **Implication:** A holiday indicator feature is valuable for the model to predict increased demand around key holidays.

**3. Feature Engineering**

In order to improve the predictive performance of the time series forecasting models, several new features were engineered from the existing sales data. These features capture temporal patterns and external factors that may influence pizza sales.

**A. Day of the Week**

**Purpose**: To account for weekly demand patterns, since certain days may see higher or lower sales (e.g., weekends vs. weekdays).

* **Method**: Extracted the day of the week from the order\_date column, converting each date to a categorical variable representing Monday, Tuesday, etc.
* **Observations**: Sales were generally higher on weekends (Saturday and Sunday), suggesting higher demand during these periods.

**B. Month**

* **Purpose**: To capture potential monthly trends or seasonality that may influence demand, such as end-of-month paydays or monthly promotions.
* **Method**: Extracted the month from each order\_date, creating a categorical variable for each month (January, February, etc.).
* **Observations**: December showed an increase in sales, likely due to holiday season demand, while summer months showed more consistent but moderate demand.

**C. Promotional Periods**

* **Purpose**: To evaluate the impact of promotional campaigns on sales volume, as discounts or special offers can significantly increase demand.
* **Method**: Created a binary indicator for known promotional periods in the dataset, marking dates where special promotions or discounts were active.
* **Observations**: Sales volumes were notably higher during promotional periods, indicating a strong effect of promotions on customer purchasing behavior.

**D. Holiday Effects**

* **Purpose**: To account for fluctuations in demand due to national holidays, special events, or festive periods (e.g., New Year’s, Christmas).
* **Method**: Introduced a binary holiday flag for dates that coincide with major holidays. Additionally, a feature was created to identify the days leading up to holidays.
* **Observations**: Significant increases in sales were observed before and during major holidays, particularly around Christmas and New Year's Eve. This suggests that customers purchase more frequently or in higher quantities during holiday seasons.

**E. Week of the Year**

* **Purpose**: To capture any seasonal weekly patterns that repeat annually, which could help in modeling demand fluctuations throughout the year.
* **Method**: Derived a week\_of\_year feature from the order\_date, representing each week from 1 to 52.
* **Observations**: Peaks in certain weeks aligned with the start of school holidays and festive seasons, helping to identify cyclical weekly patterns.
  + 1. **Model Selection:**

The goal of model selection is to choose a time series forecasting model that accurately predicts weekly sales of different pizza categories for Domino's, which will ultimately drive the automated purchase order system. Each model type has distinct strengths, and selecting the right model involves balancing accuracy, interpretability, and responsiveness to seasonality, trend, and holiday effects.

**1. ARIMA (Auto-Regressive Integrated Moving Average)**

* **Purpose:** ARIMA is a widely used model for time series forecasting that captures trend and seasonality by combining autoregressive (AR), differencing (I), and moving average (MA) components.
* **Strengths:** Best for stationary time series data (no strong seasonality); can provide accurate forecasts for data with short-term dependencies.
* **Limitations:** ARIMA may struggle with longer-term seasonality or complex trends unless explicitly modeled (SARIMA is more suitable for seasonality).

**2. SARIMA (Seasonal ARIMA)**

* **Purpose:** SARIMA extends ARIMA by incorporating seasonal components, making it more suitable for data with regular, predictable patterns, such as weekly or monthly demand.
* **Strengths:** Captures both seasonal and non-seasonal patterns, making it useful for the weekly demand variations observed in Domino’s sales.
* **Limitations:** Requires parameter tuning for each seasonal component, which can make it computationally intensive on larger datasets.

**3. Prophet**

* **Purpose:** Developed by Facebook, Prophet is designed to handle time series data that displays strong seasonality and holiday effects. It also works well with missing data and outliers.
* **Strengths:** Automatically detects seasonality and holiday effects, which aligns with Domino’s promotional and holiday-driven demand spikes. It’s also flexible in handling irregular data.
* **Limitations:** Prophet is less effective if the data doesn’t exhibit clear seasonality or holiday effects. It may also be slightly less interpretable compared to traditional models.

**4. LSTM (Long Short-Term Memory)**

* **Purpose:** LSTM is a type of recurrent neural network (RNN) that is particularly useful for sequential data. It captures complex, long-term dependencies, making it well-suited for time series forecasting.
* **Strengths:** Effective for datasets with intricate patterns, such as demand influenced by multiple temporal factors. It learns relationships in sequences, accommodating complex seasonal patterns and trends.
* **Limitations**: LSTMs require substantial computational resources and larger datasets to perform well. Tuning can be complex, and LSTMs may lack interpretability compared to simpler models.

**5. Regression Models (e.g., Linear or Ridge Regression)**

* **Purpose:** Traditional regression models can be used in time series forecasting by incorporating lagged features or engineered features from the data (e.g., day of the week, month, promotions).
* **Strengths:** Straightforward to interpret and highly customizable, allowing for the inclusion of external features like promotional periods or holiday effects.
* **Limitations:** Typically, regression models are limited in capturing temporal dependencies compared to specialized time series models. They may underperform if there are complex seasonal patterns.
  + 1. **Model Training:**

The model training process involves using historical sales data to build a predictive model capable of accurately forecasting future pizza demand. By analyzing trends, seasonality, and demand patterns in historical data, the model learns to capture the underlying dynamics that influence sales volumes. In this project, the model training steps are as follows:

1. **Data Preparation:** Preprocess the historical sales data to ensure it’s clean, consistent, and formatted correctly for time series analysis. This includes handling missing values, creating any necessary features (e.g., day of the week, holiday periods), and scaling the data if required.
2. **Train-Test Split:** Split the data into training and testing sets. Typically, the most recent portion of the data is reserved as the test set, ensuring that the model’s ability to predict future values is validated. For time series, it’s essential to avoid random splits, preserving the temporal order.
3. **Model Fitting:** Using the training data, the chosen model (e.g., SARIMA, Prophet, or LSTM) is fit on the historical sales data. This step involves optimizing model parameters to minimize prediction error on the training set, balancing the model’s accuracy with its ability to generalize to unseen data.
4. **Parameter Tuning:** Models such as SARIMA and LSTM often require parameter tuning (e.g., p, d, q values for SARIMA; number of neurons and layers for LSTM). Hyperparameters are adjusted through trial and error or cross-validation, aiming to achieve the best performance on the training data without overfitting.

**Model Evaluation**

**Pizza Sales by Week**

* This process begins by aggregating pizza sales on a weekly basis, converting the order\_date to a datetime format for accurate grouping.
* The data is then split into training (80%) and testing (20%) sets to prepare for model evaluation.
* The Mean Absolute Percentage Error (MAPE) function is defined to assess model performance by comparing actual and predicted values.

**ARIMA Model Tuning**

* The ARIMA model is tuned using a grid search over specified values of p, d, and q parameters to find the optimal configuration.
* The model forecasts sales for the test set, and the best MAPE score and corresponding parameters are printed.
* The predicted values are formatted for display, allowing for easy comparison with actual sales.
* Finally, a line plot visualizes the actual vs. predicted weekly sales, helping to evaluate the ARIMA model's forecasting performance.

A graph with blue and orange lines

Description automatically generated

**Best SARIMA Model Training and Output**

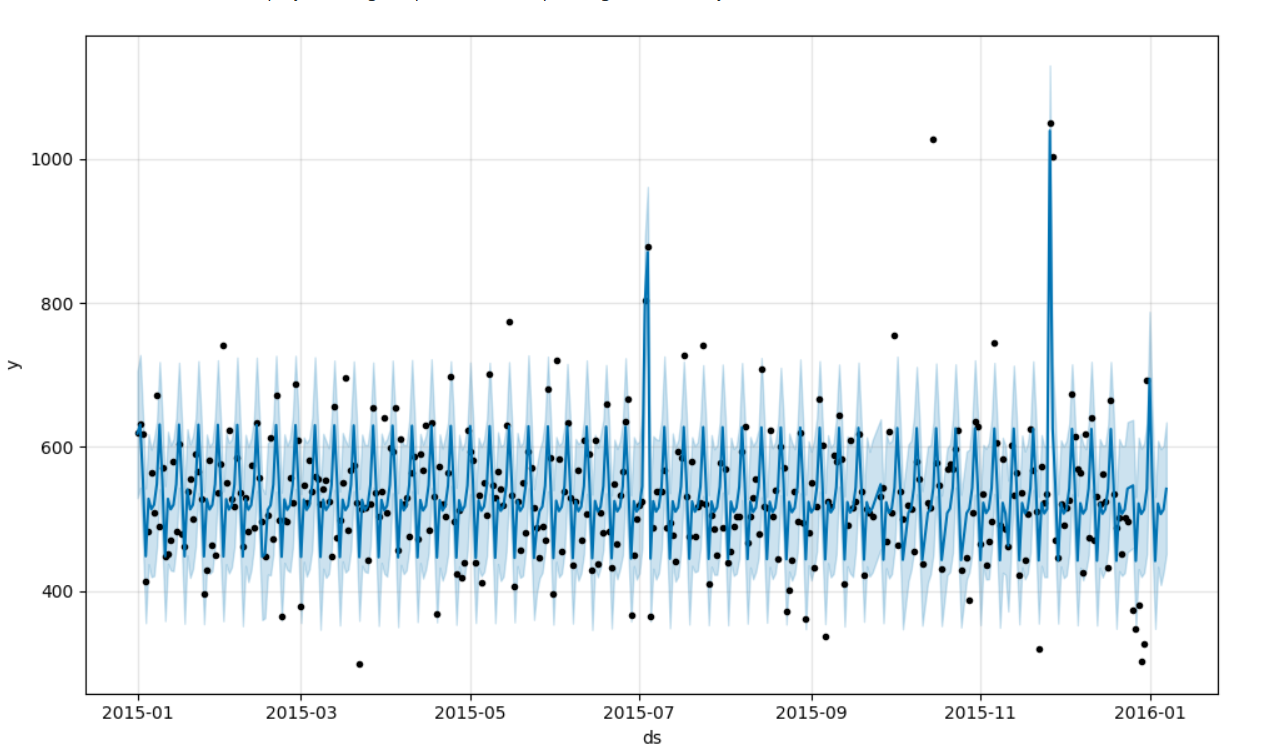
* The SARIMA model is trained with orders (1, 1, 1) for ARIMA and seasonal components.
* It forecasts sales for the test set, calculating the Mean Absolute Percentage Error (MAPE) for accuracy assessment.
* The best MAPE score is printed to evaluate model performance.
* Predictions are formatted for easy comparison with actual sales.
* A line plot visualizes actual vs. predicted weekly sales, aiding in performance evaluation.

A graph with blue and orange lines

Description automatically generated

**Prophet Model Forecasting**

* The order\_date column is converted to datetime format and renamed to 'ds' for dates and 'y' for target values.
* The Prophet model is fitted to the prepared data, with US country holidays included for enhanced accuracy.
* Future dates for the next 7 days are generated to predict sales.
* The forecast results are displayed using Prophet's built-in plotting functionality.

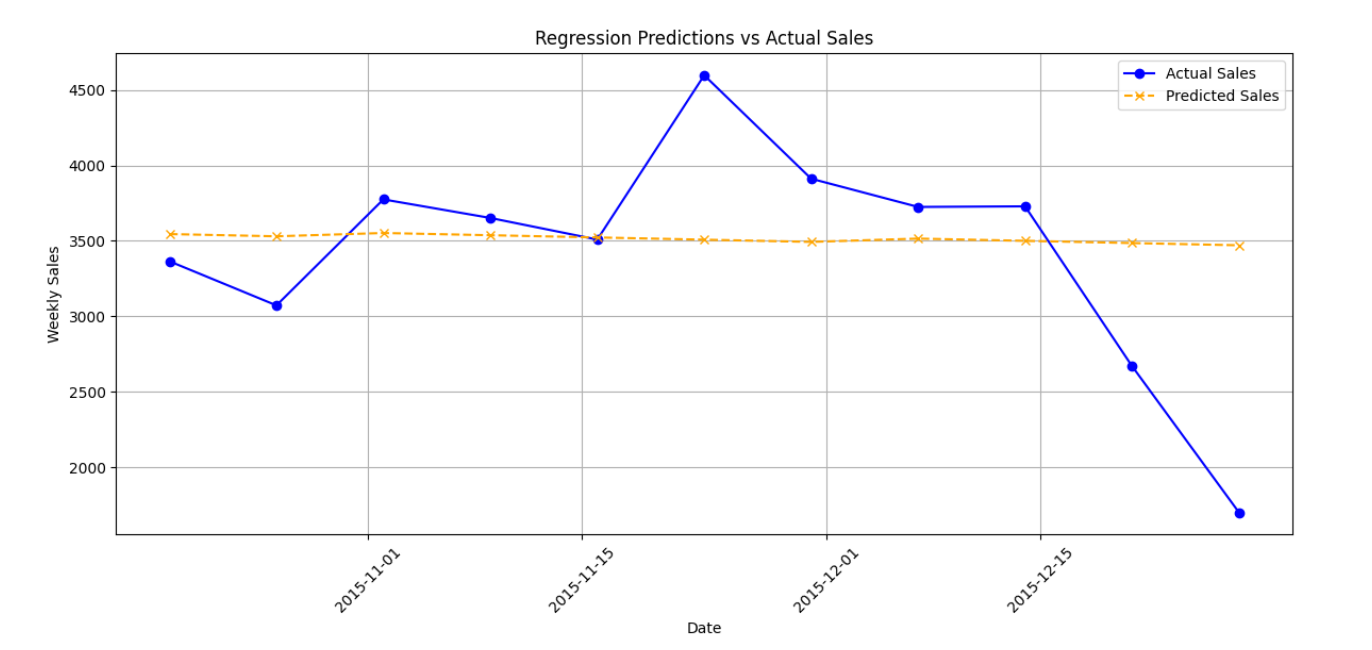


A graph with blue lines and orange dots

Description automatically generated

**Regression Model**

* Data Preparation: Converts order\_date to datetime and aggregates weekly sales.
* Feature Engineering: Creates features: week of the year, day of the week, month, and year.
* Train-Test Split: Divides data into 80% training and 20% testing sets.
* Model Training: Trains a linear regression model on the training set.
* Model Evaluation: Calculates and prints MAPE for accuracy assessment.
* Visualization: Plots actual vs. predicted weekly sales for performance evaluation.



**LSTM Model for Weekly Sales**

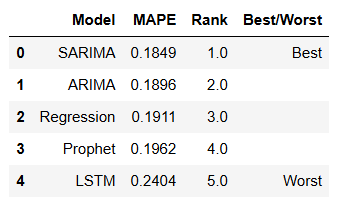
* Weekly sales data is aggregated and split into training (80%) and test sets, then normalized using MinMaxScaler.
* Sequences are created for LSTM input, and an LSTM model is trained to predict sales.
* The Mean Absolute Percentage Error (MAPE) is calculated to evaluate the model's accuracy.
* A plot compares actual vs. predicted weekly sales, assessing the LSTM model's performance.

A graph with blue and orange lines

Description automatically generated

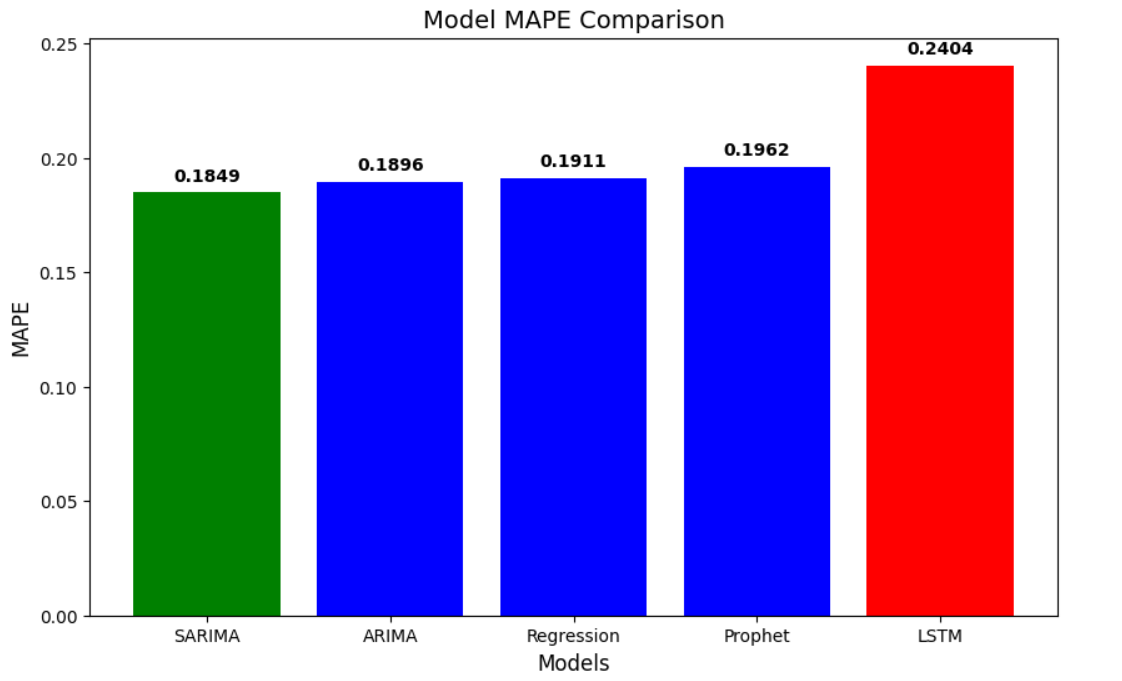
**Model Comparison: MAPE Scores**

Performance Overview: The table below summarizes the Mean Absolute Percentage Error (MAPE) scores of different forecasting models, highlighting their ranking and performance.



**Scores Visualization**

* Generated bar charts to compare MAPE scores across different models for quick performance assessment.



**Conclusion**

The SARIMA model outperformed other models, providing the most accurate sales predictions, while LSTM yielded the least accurate results. This project lays the groundwork for improving inventory management through data-driven forecasting techniques.

* 1. **Sales Forecasting**

1. **Forecast Horizon:**

The forecast horizon is set to one week, predicting sales for each day within that week. This weekly horizon aligns with typical inventory cycles and allows for responsive adjustments based on observed sales patterns.

1. **Forecasting Process:**

**Input Data:** Using recent sales data, the model is provided with the necessary inputs, such as day of the week, promotional periods, and holiday indicators, to simulate future sales. This input ensures that the model can account for patterns like weekly seasonality and external factors influencing demand.

**Model Prediction:** The trained model then generates a forecast for daily pizza sales for each category (e.g., BBQ Chicken, Veggie Veg) for the next seven days. The forecasted values provide insight into expected demand, allowing for proactive stock adjustments.

1. **Seasonality and Trends:**

By incorporating weekly seasonality and historical trends, the model can capture expected variations in demand, such as peak sales on weekends or during promotional events. This makes the forecasts more reliable, particularly in anticipating high-demand periods.

1. **Using Forecasts for Inventory Planning:**

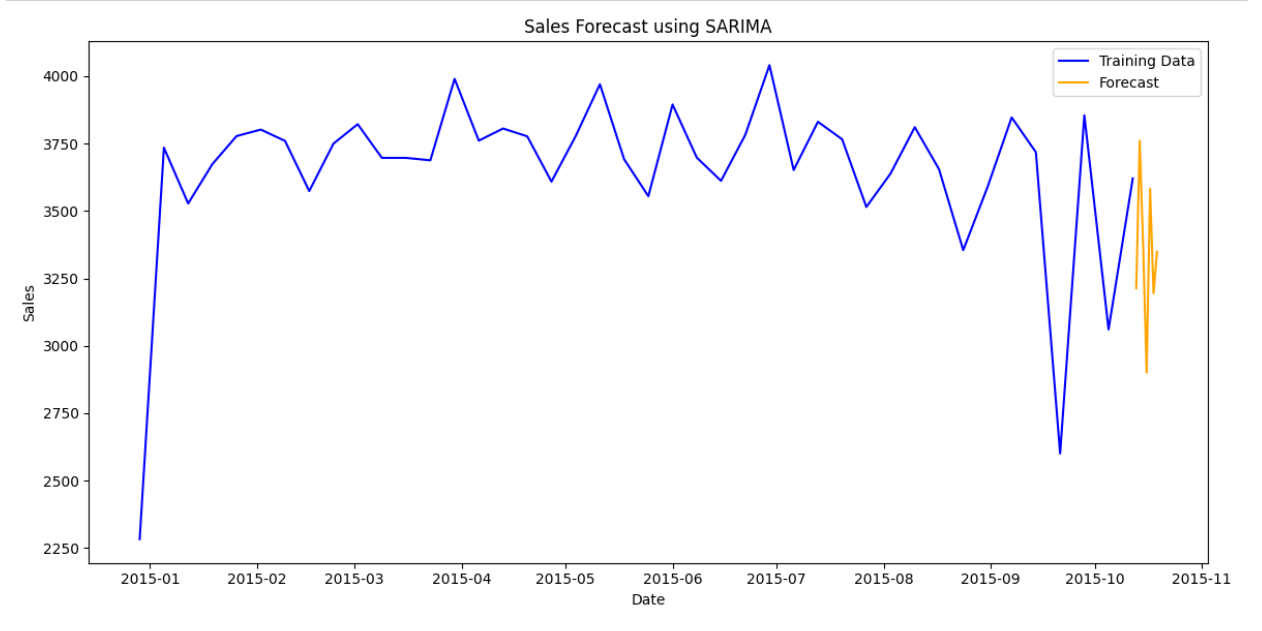
The generated sales forecast feeds into the inventory management system. By knowing projected demand for specific pizza categories, Domino's can adjust its ingredient orders accordingly, ensuring adequate stock levels while minimizing excess inventory.

1. **Adjustments Based on New Data:**

To maintain accuracy, the model can be retrained periodically with recent data, ensuring forecasts remain aligned with evolving sales trends. This is particularly useful during changing conditions, such as new promotions or seasonal holidays, which may impact demand patterns.

**To load the best-performing SARIMA model and use it to forecast future sales:**

1. **Model Loading:** The SARIMA model, previously trained and saved as best\_sarima\_model.pkl, is loaded for use in forecasting.
2. **Forecasting:** The model predicts sales for the next 7 days (n\_forecast = 7).
3. **Visualization:** A plot is generated to compare the training data with the forecasted sales, clearly illustrating expected future sales trends. The forecast is displayed in orange, while the training data is shown in blue.



* 1. **Ingredient Calculation:**

Calculation of the total quantity of ingredients needed based on predicted pizza sales for the upcoming week.

* 1. Mapping Predicted Sales: The Ingredients\_dataset maps predicted sales to corresponding ingredients from the next\_week\_pizza\_sales\_forecasts\_arima.
  2. Calculating Total Quantity: The total quantity for each ingredient is computed by multiplying the grams needed per item (Items\_Qty\_In\_Grams) by the predicted quantity of pizzas sold.
  3. Summarizing Totals: The summed total quantities for each ingredient are displayed as a dictionary, giving a clear overview of the ingredient requirements for the upcoming week.
  4. Visualizing Quantities: A bar chart visualizes the top 10 predicted ingredients, illustrating the total quantity (in grams) needed for the next week.
  5. Saving Results: The ingredient totals are saved to a CSV file, predicted\_ingredient\_totals.csv, for future reference and easy sharing.

A screenshot of a computer

Description automatically generated

A graph of food ingredients

Description automatically generated

A screenshot of a recipe

Description automatically generated

* 1. **Purchase Order Creation:**
* **Objective**: Generate a detailed purchase order listing the exact quantities of each ingredient needed for the forecasted sales period.
* **Order Format**: The purchase order includes:
  + **Ingredient Name**: Lists each ingredient required (e.g., cheese, tomato sauce, dough, toppings).
  + **Quantity**: Specifies the total quantity needed for the forecasted period, derived from the ingredient calculations.
  + **Unit of Measure**: Each ingredient's unit (e.g., grams, kilograms, liters) is listed to ensure clarity.
  + **Supplier Information**: Optional fields may include supplier details for easier procurement.
* **Automated Generation**: Once all quantities are calculated, the system compiles them into a structured purchase order format, which can be automatically sent to suppliers or used for internal inventory planning.
* **Results:**

The project delivers highly accurate pizza sales forecasts, providing precise predictions for the upcoming week. These forecasts enable better planning and optimized inventory management. Based on the predicted sales, a comprehensive purchase order is generated, detailing the exact quantities of each ingredient required. This ensures that the necessary ingredients are stocked efficiently, minimizing waste and avoiding shortages. The result supports seamless operations by aligning supply with demand, improving both supply chain efficiency and overall business performance.