**Dominos - Predictive Purchase Order System**

## **Problem Statement:**

This project focuses on optimizing Domino's inventory management by building a predictive system that forecasts pizza sales and generates purchase orders for ingredients. By leveraging historical sales data, the goal is to develop a model that accurately predicts future sales, allowing Domino's to order the right amount of ingredients, minimizing waste, and preventing stockouts.

## **Objective:**

* To Develop a predictive model to forecast pizza sales.
* To Create a purchase order system that calculates the required quantities of ingredients based on the sales forecast.

## Dataset Overview

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

* The **Pizza Sales Dataset** comprises 48,620 entries, each detailing an individual sale. This includes information such as pizza\_id (a unique identifier for the sale), order\_id (linking to a specific order), pizza\_name\_id (a unique identifier for each pizza), quantity (number of pizzas sold), order\_date and order\_time (when the sale occurred), unit\_price and total\_price (pricing details), as well as pizza\_size and pizza\_category (size and type of pizza). This dataset offers a thorough view of sales, covering pricing, timing, and pizza characteristics.
* The **Pizza Ingredients Dataset** consists of 518 entries that describe the ingredients for various pizzas. It includes pizza\_name\_id (a unique identifier for each pizza), pizza\_name (name of the pizza), pizza\_ingredients (list of ingredients), and Items\_Qty\_In\_Grams (the quantity of each ingredient used). This dataset provides detailed insights into the composition of each pizza and the amounts of ingredients required.

You can download the datasets from the following links:

[Download pizza\_sales Dataset](https://docs.google.com/spreadsheets/d/1gB2O_G9G_ym41h-Wra3k4VDHHYVONh3jfkEBLzAOuUI/edit?usp=sharing)

[Download pizza\_ingredients Dataset](https://docs.google.com/spreadsheets/d/13h0bdBCgcnf7kduth_5ci1pMytzXc02H57Ms30Xg5AA/edit?usp=drivesdk)

## **Metrics**

* [**Mean Absolute Percentage Error**](https://scikit-learn.org/stable/modules/generated/sklearn.metrics.mean_absolute_percentage_error.html): Used to evaluate the accuracy of forecasting models. It measures the average absolute percentage error between the predicted values and the actual values.

## **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**

### I. Data Preprocessing

**Data Cleaning** ensures the dataset's accuracy and consistency through:

* **Handling Missing Data**:
  + Detected missing values.
  + Replaced missing values using mean, median, mode, or placeholders.
  + Removed columns or rows with excessive missing data if necessary.
* **Removing Inconsistent Data**:
  + Checked for format consistency and valid ranges.
  + 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.

### **II. Exploratory Data Analysis (EDA)**

**Exploratory Data Analysis (EDA)** discovers patterns, relationships, and anomalies in the data.

### i) Top-Selling Pizzas

* This visualization highlights the top 10 most popular pizzas based on total sales.
* It helps identify which pizzas are in highest demand among customers.
* The y-axis represents different pizza names, while the x-axis shows the quantity sold.

#### **ii) Distribution of Pizza Categories:**

* This visualization displays the distribution of pizza orders across various categories (e.g., vegetarian, meat-lovers).
* It provides insights into customer preferences and trends by category.
* The y-axis represents different pizza categories, while the x-axis shows the number of orders for each category.

#### **iii) Sales Trends Over Time:**

* This visualization shows daily pizza sales over time by converting the order\_date column into a datetime format.
* It helps identify trends, seasonality, and sales spikes throughout the observed period.
* The line plot illustrates the quantity of pizzas sold each day.

#### **iv) Sales by Day of the Week**

* This visualization aggregates total sales by day of the week to identify which days generate the most revenue.
* The data is grouped by day\_of\_week, summing the total\_price for each day.
* The x-axis represents the days of the week (ordered from Monday to Sunday), while the y-axis shows the total sales for each day.
* The bar plot helps pinpoint trends in customer purchasing behavior throughout the week.

### III. Sales Prediction

Sales Prediction involves **Time Series Forecasting**, a technique used to predict future values based on historical data collected over time. The process includes the following steps:

#### i) Feature Engineering

Created new variables from the raw sales data to improve the model’s performance, such as:

* **Day of the Week**: Extracted the day of the week from the sales date to capture weekly variations.
* **Month**: Extracted the month from the sales date to account for monthly trends and seasonal patterns.
* **Holiday Effects**: Identified and included features for holidays or special events that can impact sales patterns.

#### ii) Model Selection

Model Selection involves choosing the most suitable forecasting model for our sales data:

* [**ARIMA (AutoRegressive Integrated Moving Average)**](https://www.statsmodels.org/stable/generated/statsmodels.tsa.arima.model.ARIMA.html): Captures trends and autocorrelations in non-seasonal data.
* [**SARIMA (Seasonal ARIMA)**](https://www.statsmodels.org/stable/generated/statsmodels.tsa.statespace.sarimax.SARIMAX.html): Extends ARIMA to handle seasonality.

#### iii) Model Training

Model Training involves fitting the chosen model to historical sales data:

* Split the data into training and test sets to evaluate model performance.
* Trained the model on the training set by adjusting parameters to minimize prediction errors.
* Optimized model performance by tuning hyperparameters using techniques like cross-validation or grid search.

### iv) 📊 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.

#### 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.

#### **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.

#### **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.

### 🧩 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.

| **Model** | **MAPE** | **Rank** | **Best/Worst** |
| --- | --- | --- | --- |
| SARIMA | 0.1849 | 1 | Best |
| ARIMA | 0.1896 | 2 |  |
| Regression | 0.1911 | 3 |  |
| Prophet | 0.1962 | 4 |  |
| LSTM | 0.2404 | 5 | Worst |

## **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.

### **SARIMA Model Forecasting**

This section demonstrates how 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.

### Predicted Ingredient Quantities

This section calculates 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.

## **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.