Dominos - Predictive Purchase Order System

Submitted by:

Yamini Devi

Table of contents	
1.	Project overview
2.	Data cleaning and preprocessing • Check null values, duplicates, unique and nunique • Remove duplicates if occur • Clean date column • To fill null values using df.loc • Standardize spelling • Statistics Analysis • Remove outliers • Feature Engineering • Exploratory Data Analysis
3.	Weekly sales prediction
4.	Time Series Forecasting Using Prophet
5.	MAPE (Mean Absolute Percentage Error) Calculation
6.	Future Sales Predictions
7.	Conclusion and Summary
8.	References

1. Project Overview

The project focuses on analyzing and forecasting pizza sales using two datasets: Sales and Ingredients. The key objectives are to clean and preprocess the data, conduct exploratory data analysis (EDA), create new features, and then apply time series forecasting techniques to predict future pizza sales. Additionally, model performance is evaluated using statistical measures such as MAPE (Mean Absolute Percentage Error).

2. Data Preprocessing & Cleaning

Sales Dataset:

1. **Check for Null Values, Duplicates, Unique, and nunique:**
    - **Null Values**: It's important to identify any missing values in the dataset as they can skew analysis or disrupt modeling processes. We inspect the dataset to determine where missing data occurs.
    - **Duplicates**: Duplicate rows can introduce bias or inflate metrics like sales or revenue, so we check for and remove any duplicate entries.
    - **Unique and nunique**: Understanding the unique values in each column helps identify potential inconsistencies in categorical data. For example, analyzing if product names are spelled correctly or standardized.
2. **Remove Duplicated Values:**
    - If duplicate rows are found in the dataset, they need to be removed to ensure that our analysis isn't skewed by redundant records. Duplication often occurs in data collection due to system issues or repeated entries.
3. **Cleaning Date Column:**
    - The date column is particularly important for time series analysis. It is essential to ensure that all dates are in the correct format (e.g., YYYY-MM-DD) and that invalid dates (e.g., typos or incorrect entries) are corrected. This step ensures that future trend analysis and time-based aggregations are accurate.
4. **Filling Null Values Using df.loc:**
    - Null or missing values can be problematic, especially if they exist in key columns. Using logical conditions, such as replacing missing sales data with the average sales for that day or week, helps avoid skewed results while maintaining data integrity.
5. **Standardizing the Spelling:**
    - In many datasets, categorical columns (e.g., pizza types or ingredients) may have inconsistencies in spelling or capitalization. For example, 'Pepperoni' could appear as 'pepperoni', 'PEPPERONI', or 'Peperoni'. Standardizing the spelling ensures consistency, improves readability, and avoids errors during analysis or modeling.

6. **Statistical Analysis:**
    - **Mean**: The average value of a particular column, e.g., average sales or revenue, which provides a sense of the general performance.
    - **Median**: The middle value in the data distribution, useful for identifying typical behavior while ignoring outliers.
    - **Mode**: The most frequent value, which can be relevant for categorical data, such as the most popular pizza sold.
    - **Standard Deviation (SD)**: Measures how spread out the data is from the mean, indicating volatility in sales or revenue.
    - **Variance**: The square of the standard deviation, which also reflects the dispersion in the data.
7. **Remove Outliers:**
    - Outliers are extreme values that fall far outside the general data trend. These may represent incorrect data entries or extraordinary events (like a sales spike during a promotion). Removing outliers prevents them from distorting statistical analysis and machine learning models. Techniques like the Z-score or interquartile range (IQR) help detect these outliers.
8. **Feature Engineering:**
    - This step involves creating new columns from existing data to provide additional insights. For example:
        - **Day of the week**: We can extract the day (Monday, Tuesday, etc.) from the date column, which can help determine which days have the highest pizza sales.
        - **Month**: Sales can also vary by season or month, so adding a month column can help track sales trends over time.
        - **Weekend Indicator**: Creating a new feature that indicates whether a sale happened on the weekend or a weekday could show different buying patterns.
9. **Exploratory Data Analysis (EDA):**
    - **Visualization**: Data is visualized using plots to uncover trends, patterns, or anomalies. For example, plotting sales over time can reveal seasonality (i.e., sales peaks during weekends, holidays, or specific months).
    - **Relationships**: Scatter plots or correlation matrices can help identify relationships between variables, such as whether larger pizzas are sold more often on weekends or if certain types of pizzas drive higher revenue.
    - **Distribution**: Histograms are used to analyze the distribution of key variables like revenue or sales quantity, helping to identify normal behavior and outliers.

Ingredients Dataset:

1. **Check for Null Values, Duplicates, Unique, and Nunique:**
    - Similar to the Sales dataset, we begin by identifying any missing values, duplicate entries, and counting unique values in key columns (e.g., ingredient names). This ensures we have a clean and reliable dataset for analysis.
2. **Remove Duplicated Values:**
    - As in the Sales dataset, we remove duplicate entries to avoid over-representing any ingredient in our analysis. Duplicates could arise if the same ingredient is listed multiple times by mistake.
3. **Filling Null Values Using df.loc:**
    - For missing ingredient data, we need to fill in the gaps. This might involve replacing missing values with the most common ingredient or imputing the value based on related information.
4. **Standardizing the Spelling:**
    - Ensuring that ingredient names are spelled consistently is essential to avoid errors in analysis. For example, 'cheddar' and 'CHEDDAR' should both be standardized to 'cheddar'.

3. Weekly Pizza Sales Analysis

- **Objective**: The goal here is to analyze sales performance on a weekly basis to understand the broader trends. Aggregating the data by week allows us to see how sales fluctuate throughout the year, accounting for factors like weekends, holidays, or seasonal events.
- **Method**: By summing up daily sales for each week, we can calculate total weekly sales. Visualizing these weekly trends using line charts can highlight peak sales periods, seasonal variations, and any declines or growth patterns. This insight is particularly useful for planning inventory and staffing.

4. Time Series Forecasting Using Prophet

- **Objective**: We use Facebook's Prophet model to forecast future pizza sales. Prophet is designed to handle time series data, especially when the data has clear patterns like daily or weekly seasonality.
- **Process**:
    1. **Prepare Data**: The dataset needs to be formatted with two columns: one for dates and one for sales figures. Prophet requires this specific structure to process the data correctly.
    2. **Training**: We train the model on the historical data (i.e., past pizza sales) to identify patterns, trends, and seasonality.

3. **Prediction**: Once the model is trained, we can generate future sales predictions for the desired time horizon (e.g., the next year). These predictions will account for the trends and seasonality observed in the historical data.

5. MAPE (Mean Absolute Percentage Error) Calculation

- **Objective**: To evaluate the accuracy of the sales predictions. MAPE measures the accuracy of a forecast by comparing the predicted values to the actual observed values.
- **Interpretation**:
    - A lower MAPE indicates higher forecast accuracy, while a higher MAPE suggests larger errors in the predictions.
    - For example, a MAPE of 5% means the forecast error is, on average, 5% of the actual sales figures.
- **Usage**: After generating sales predictions using the Prophet model, we compare these predictions to the actual sales figures (for the same time periods) and calculate MAPE. This metric gives us an idea of how well the model is performing.

6. Future Sales Predictions

- **Objective**: Once the time series model is trained and validated, it can be used to forecast future sales. These predictions provide valuable insights for decision-making, such as inventory management, staffing, and marketing.
- **Steps**:
    1. **Long-Term Forecasts**: With the trained Prophet model, we can project sales for future periods (e.g., 3 months or 1 year ahead). This helps in planning ahead for peak seasons or potential slowdowns.
    2. **Visualization**: The forecast is usually visualized with confidence intervals to show the range of expected sales. This allows decision-makers to plan for best and worst-case scenarios.

7. Conclusion and Summary

In conclusion, this project involved cleaning and processing the Sales and Ingredients datasets, followed by exploratory analysis and forecasting. Key steps included:

- **Data Preprocessing**: Handling missing values, removing duplicates, and standardizing inconsistent data entries.
- **Exploratory Data Analysis**: Uncovering trends, seasonality, and patterns in pizza sales.
- **Feature Engineering**: Creating new time-related features to enhance analysis.
- **Forecasting**: Using Prophet to generate future sales predictions and evaluate accuracy using MAPE.

The outcome provides valuable insights into pizza sales behavior, helping businesses make data-driven decisions for inventory, staffing, and marketing strategies.

8. References

- Pandas documentation
- Data visualization
  - Seaborn documentation
  - Matplotlib documentation
  - Plotly express documentation
- Prophet documentation