Reference
This API documentation is created based on the instruction:
https://stoplight.io/api-documentation-guide
Database resource from Kaggle:
https://www.kaggle.com/datasets/shilongzhuang/pizza-sales?datasetId=2566526&sortBy=commentCount

Functionality

*Function#1: convert_data
Input:
Read the input .csv file in the function parameter
Output:
Return the cleaned data.
Store a new .csv file in the local file after data cleaning named "Cleaned.csv".
Sample output:
{..., Month:1...12, week_info:..., Hour:.., Minute:..., Second:..., Meal_time: ...}
With 18 named columns and 48621 rows.
For more details, see the file "Cleaned.csv"
How it works:
This is a data cleaning and preparation function that converts the time-based data(order_date, order_time) to different timing categories, like Month, week_info, Hour, Minute, and Second. Also, based on the Hour column, it sorts each pizza order into four categories for Meal time, namely lunch, afternoon tea, dinner, and late-night meal.
This provides the foundation for further analysis.

*Function#2: meal_hist
Input:
Cleaned data obtained after the data cleaning process. (output from convert_data)
Output:
Histogram graph for different meal times for pizza orders.
Stored in a local file named "meal_hist.jpg".
How it works:
It generates a histogram that shows the frequency of the pizza order at different meal times, ranging from lunch, afternoon tea, and dinner to late-night meals.
The x-axis represents four types of meal time, and the y-axis represents the order frequency.

*Function#3: week_hist
Input:
Cleaned data obtained after the data cleaning process. (output from convert_data)
Output:
Histogram graph for weekly information for a pizza order.
Stored in a local file named "week_hist.jpg".
How it works:

Rearrange the days of the week using the map function.
Plot a histogram to display the frequency of the pizza orders on a weekly basis, ranging from Sunday to Saturday.
The x-axis represents the days of the week, and the y-axis represents the order frequency of the pizza.

*Function#4: month_hist
Input:
Cleaned data obtained after the data cleaning process. (output from convert_data)
Output:
Histogram graph for monthly information for pizza orders.
Stored in a local file named "month_hist.jpg".
How it works:
Map the numerical indexes to the respective months.
It plots a histogram that shows the frequency of the pizza order on a monthly basis, ranging from January to December.
The x-axis represents 12 months of the year, and the y-axis represents the order frequency of the pizza to the corresponding month.

*Function#5: category_hist
Input:
Cleaned data obtained after the data cleaning process. (output from convert_data)
Output:
Histogram graph for different categories among all sales.
Stored in a local file named "category_hist.jpg".
How it works:
Create the labels for each pizza category and generate a histogram that shows the frequency of the pizza orders for each of the four categories.
The x-axis represents the pizza type and the y-axis represents the frequency of pizza sales for each category.

*Function#6: category_pie
Input:
Cleaned data obtained after the data cleaning process. (output from convert_data)
Output:
Pie chart to display the percentage of pizza sales by each pizza category.
Stored in a local file named "category_pie.jpg"
How it works:
Extract labels of pizza categories and their respective counts.
Plot a pie chart to display the percentage distribution of pizza category.

*Function#7: size_hist
Input:
Cleaned data obtained after the data cleaning process. (output from convert_data)

Output:
Histogram graph to display distribution of pizza sales according to different pizza sizes
Stored in a local file named "size_hist.jpg".
How it works:
Create the labels for five sizes and generate a histogram that shows the frequency of the pizza sale for each size of pizza.
The x-axis represents the pizza size and the y-axis represents the frequency of sales.

*Function#8: size_pie
Input:
Cleaned data obtained after the data cleaning process. (output from convert_data)
Output:
Pie chart to display the percentage of pizza sales by each pizza size.
Stored in a local file named "size_pie.jpg".
How it works:
Extract labels of pizza sizes and their respective size counts.
Plot a pie chart to display the percentage distribution of pizza size.

*Function#9: summary_data
Input:
Cleaned data obtained after the data cleaning process. (output from convert_data)
Output:
Dictionary named pizza_sales_dict stores all statistical information with key, and value matched in pairs.
{'Lunch': 263266.05, 'Afternoon_Tea': 182249.1, 'Dinner': 306378.6, 'Late_Night_Meal': 65966.3, 'Max': ['Dinner', 306378.6], 'Min': ['Late_Night_Meal', 65966.3],
'Size sale':
L 18526
M 15385
S 14137
XL 544
XXL 28
Name: pizza_size, dtype: int64, 'Weekday': 595474.15, 'Weekend': 222385.9, 'Most_sold': 'The Classic Deluxe Pizza'}
How it works:
This is a data summary function that gives us all important statistical pizza sales information in an empty dictionary.
Firstly, find all sales information in four meal times and store them in the dictionary respectively.
Second, find the maximum and minimum sales period from the new dictionary.
Third, find the frequency of pizza sales based on different sizes.
Fourth, create two new dictionaries to find the weekday and weekend sales and store them in the initial empty dictionary.
Lastly, find the most well-sold pizza and store it in the dictionary.

*Function#10: linear_reg_month
Input:
Cleaned data obtained after the data cleaning process. (output from convert_data)
Output:
A linear regression model to predict monthly sales based on the monthly quantity of pizzas and the respective monthly index
How it works:
Sum up the monthly sales based on the total price from each order and sum up the monthly pizza sales amount based on quantity for each order.
Then create a monthly index(e.g:1,2...12) and merge it into the data frame to find monthly pizza amount sales.
The dependent variable is total monthly sales and the independent variable is a monthly index and monthly sales.
Next, split data for machine learning and generate a linear regression model.

*Function#11: linear_reg_eve
Input:
Data after the data cleaning process (output from convert_data)
Output:
A linear regression model to predict the sales on Dec 24, 2015, at different meal times.
How it works:
Firstly, count the total sales and quantity based on different meal times on Dec 24th, and merge sales, quantity, and meal time into a data frame.
Convert four meal times into categorical number:1,2,3,4. The dependent variable is total sales on Dec. 24th, and the independent variable is the meal time category index and meal time sales.
Next, split data for machine learning and generate a linear regression model.

*Function#12: linear_reg_unit_price
Input:
Data after the data cleaning process (output from convert_data), the numerical value for pizza size, and the numerical value for pizza category.
Output:
The predicted unit price from the Linear regression model is based on the input pizza size and category.
Sample output:
Predicted pizza unit price is $ [15.85613274]
How it works:
Convert the categorical variables- pizza size and pizza category into the numerical variables(1-5 and 1-4).
Declare the Independent variables as pizza size and pizza category and the dependent variables as the unit price of the pizza.
Perform train/test split of the dataset and fit the multiple regression model.

A multiple linear regression model is generated to predict the unit price of pizza based on the input pizza size and pizza category.

*Function#13: print_plot
Input:
Numerical value user input.
Output:
Secondary level menu for instructions on checking data visualization information.
Correct input - Store graphs in a local file, such as a histogram or pie chart.
Wrong input - Please provide the correct name
Special information: 8 - Back to the main menu
Sample Output:
Enter the number want to check for Graphical Plots below:

1. Histogram graph for Different Meal Time
2. Histogram graph for Weekly Information
3. Histogram graph for Month Information
4. Histogram graph for Pizza Category
5. Histogram graph for Pizza Size
6. Pie chart for Pizza Category
7. Pie chart for Pizza Size
8. Back to the previous section

>> 3
Check your local file document for graph month_hist.jpg
How it works:
This is the function under the graphical information and asks the user to provide which plot they want to check and store in a local file.
The user input must be a number ranging from 1 to 8. The function will present the respective diagram to the user.
However, if the input is 8, it will back to the menu section.
Input outside 1 to 8 will ask the user for correction.
The desired graphical plots will be stored in the local file.

*Function#14: print_stat
Input:
Numerical value user input.
Output:
Secondary level menu for instructions on checking statistical information for data.
Correct input - Based on the number provided, a print number for related information.
Wrong input - Please provide the correct number.
Special information: 11 - Back to the main menu
Sample Output:
Enter the number want to check for statistics below:

1. Lunchtime pizza sales for the whole year of 2015
2. Afternoon Tea time pizza sales

3. Dinner-time pizza sales
4. Late Night Meal time pizza sales
5. Maximum pizza sales meal time
6. Minimum pizza sales meal time
7. Pizza sale by size
8. Total Weekday Pizza Sale Information
9. Total Weekend Pizza Sale Information
10. Most sold Pizza
11. Back to the previous section

>> 9
A total weekend sale is 222385.9
How it works:
Based on user inputs, find the index for specific information by calling the key name of the dictionary from the function summary_data.
The desired statistical information will be displayed.

*Function#15: print_predict
Input:
Numerical value user input.
Output:
Secondary level menu displaying instructions for printing model prediction result.
Correct input - Prediction values for each selected regression model.
Wrong input - Please provide the correct number.
Special information: 4 - Back to the main menu
Sample Output:
Enter the number want to check for Graphical Plots below:

1. Predict Monthly Sales
2. Predict Sales for Meal Time on Christmas Eve
3. Predict Pizza Prices based on size and category
4. Back to the previous section

>> 2
Please provide the number for different meal times:

1. Lunch(Smaller or equal to 1 pm)
2. Afternoon Tea(In between 2 pm to 4 pm)
3. Dinner(In between 5 pm to 8 pm)
4. Late Night Meal(Later than 8 pm)

>> 4
Please provide the pizza amount sold in that period on Christmas Eve:
>> 233
The predicted sale for Christmas Eve is $ [3798.89705882]
How it works:
The print_predict function calls the respective multiple regression model functions by passing the necessary parameters in the function call.

*Function#16: main
Input:
CSV file - "Data Model - Pizza Sales.csv"
Numerical value user input
Output:
Based on the input value, the respectively mapped sub-function is called.
Special information: 4 - End the checking information process

Sample Output 1:
Type number for checking pizza sales information:
1. Graphical Information
2. Statistical Summary Information
3. Prediction Information
4. Quit the checking process
>> t
Please provide the correct number

Sample Output 2:
Type number for checking pizza sales information:
1. Graphical Information
2. Statistical Summary Information
3. Prediction Information
4. Quit the checking process
>> 1
Enter the number want to check for Graphical Plots below:
1. Histogram graph for Different Meal Time
2. Histogram graph for Weekly Information
3. Histogram graph for Month Information
4. Histogram graph for Pizza Category
5. Histogram graph for Pizza Size
6. Pie chart for Pizza Category
7. Pie chart for Pizza Size
8. Back to the previous section
>> 3
Check your local file document for graph month_hist.jpg

How it works:
Data pre-processing before displaying the main menu.
While loop map user input to respective sub-function.
Try-except to handle user input exceptions.
==========
==========