



Assessment Submission Form

Student Number (If this is group work, please include the student numbers of all group participants)	GH1024695 Mahdieh Rajabi
Assessment Title	Using social media tool
Module Code	M608
Module Title	Business project in computer science(RETAKE -SEPTEMBER)
Module Tutor	William Morison
Date Submitted	04.10.2024
<p align="center">Declaration of Authorship</p> <p>I declare that all material in this assessment is my own work except where there is clear acknowledgement and appropriate reference to the work of others.</p> <p>I fully understand that the unacknowledged inclusion of another person's writings or ideas or works in this work may be considered plagiarism and that, should a formal investigation process confirms the allegation, I would be subject to the penalties associated with plagiarism, as per GISMA Business School, University of Applied Sciences' regulations for academic misconduct.</p> <p>Signed.....Mahdieh Rajabi..... Date04.10.2024.....</p>	

Link of GitHub: https://github.com/Mahdiehrajabi/Retake2_Business-project

Table of content

- 1. Introduction**
- 2. Data description**
- 3. Methodology and structure**
 - 3.1. Result**
- 4. Customer behavior**
 - 4.1. Result of customer behavior**
- 5. Market trends**
 - 5.1. Result of market trends**
- 6. Errors and debugging**
 - 6.1. Error1**
 - 6.2. Error2**
 - 6.3. Debug1**
 - 6.4. Debug2**
- 7. Conclusion**
- 8. References**

1. Introduction

Effective product analysis is critical to comprehending consumer preferences, market trends, and corporate performance in today's data-driven environment. Businesses frequently use data to assess the performance of their products, improve consumer satisfaction, and optimize pricing tactics. With the use of Python programming, this project seeks to conduct a thorough analysis of product data. Its main objective is to glean insightful information from a dataset that includes product categories, prices, customer reviews, and ratings.

This report will focus on the explanation about the code which used, codes and results for decision. I used libraries and tools like Pandas, Tkinter, matplotlib and etc. for gain the desire result.

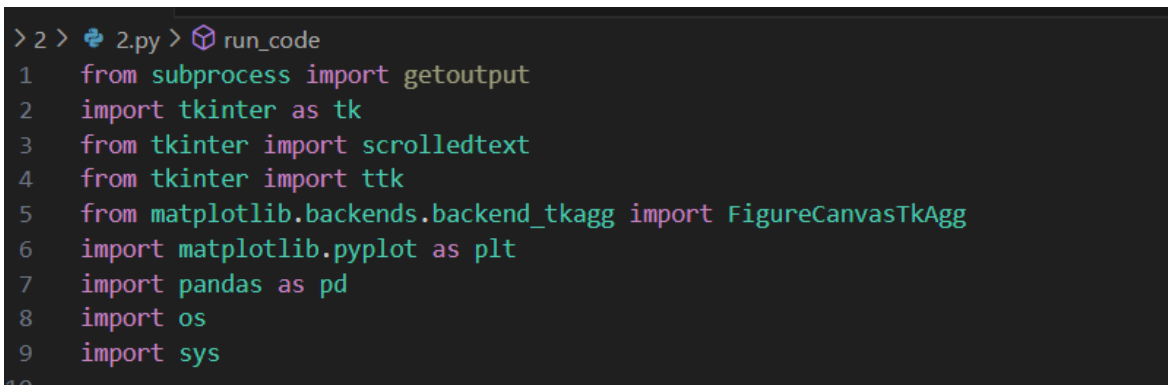
2. Data Description

I got dataset from Kaggle website, and it is related to amazon sales in different categories.

The dataset that was used for this investigation includes facts about a wide range of products, such as information about categories, prices, ratings, and reviews from customers. Understanding dataset helped to have a clear idea to work with.

After considering the dataset, I realized that cleaning data specially in two columns which is related to price are necessary. After cleaning I was able to work and create a tool to have a graphs and visualization data with Python.

3. Methodology and structure



```
> 2 > 2.py > run_code
1  from subprocess import getoutput
2  import tkinter as tk
3  from tkinter import scrolledtext
4  from tkinter import ttk
5  from matplotlib.backends.backend_tkagg import FigureCanvasTkAgg
6  import matplotlib.pyplot as plt
7  import pandas as pd
8  import os
9  import sys
```

Figure1: Import libraries

After importing needed libraries, dataset should be readable and for making sure about path of dataset, I check the directory of file every time.

```

16
17 # Change to the directory containing the CSV file
18 new_directory = r'C:\2'
19 os.chdir(new_directory)
20
21 # Check current working directory
22 print("Current Working Directory:", os.getcwd())
23
24 # Read csv dataset
25 df = pd.read_csv('amazon.csv')
26 df.dropna(inplace=True)
27

```

Figure2: Upload dataset

Then, I wanted to have a function and put details in that function. I called it Run_code() and start to put a code for clearing the previous output for this. It needed to display the result with Tkinter GUI. The line of text_output is about the shape of dataframe and it can bring an overview of how much data is being proceeded. At first glance, I wanted to get info about main top categories and their amount because of that I created a graph to show top 10 categories based on amazon.csv file.

```

29
30 def run_code():
31
32     # Clear previous output
33     text_output.delete('1.0', tk.END)
34
35
36     # Display data shape and numerical properties
37     text_output.insert(tk.END, f" Shape: {df.shape}\n\n")
38
39
40     # Diagram of the main categories
41     df['main_category'] = df['category'].astype(str).str.split('|').str[0]
42     main_category_counts = df['main_category'].value_counts()[:10]
43
44     fig, ax = plt.subplots(figsize=(5, 4))
45     ax.barh(range(len(main_category_counts)), main_category_counts.values)
46     ax.set_xlabel('Number of Products')
47     ax.set_title('Distribution of Products by Main Category (Top 10)')
48     ax.set_yticks(range(len(main_category_counts)))
49     ax.set_yticklabels(main_category_counts.index)
50     ax.tick_params(axis='y', labelsize=4)
51
52     # Display the chart
53     canvas = FigureCanvasTkAgg(fig, master=frame_plot)
54     canvas.draw()
55     canvas.get_tk_widget().pack()
56

```

Figure3: Function and first graph

For showing the plot, the code was this

```
56
57     # Display main category
58     text_output.insert(tk.END, "Top 10 main categories:\n")
59     top_main_categories = pd.DataFrame({
60         'Main Category': main_category_counts.index,
61         'Number of Products': main_category_counts.values
62     })
63     text_output.insert(tk.END, top_main_categories.to_string(index=False) + "\n\n")
64
```

Figure4: Display main categories

Next, I realize that dataset has subcategories, and I classified them as well in new graph. And put a text box to show data as well.

```
4
5     # Display Sub category graph
6     df['sub_category'] = df['category'].astype(str).str.split('|').str[-1]
7     sub_category_counts = df['sub_category'].value_counts()[:10]
8
9     fig2, ax2 = plt.subplots(figsize=(5, 4))
10    ax2.barh(range(len(sub_category_counts)), sub_category_counts.values)
11    ax2.set_xlabel('Number of Products')
12    ax2.set_title('Distribution of Products by Sub Category (Top 10)')
13    ax2.set_yticks(range(len(sub_category_counts)))
14    ax2.set_yticklabels(sub_category_counts.index)
15    ax2.tick_params(axis='y', labelsize=5)
16
17    # Show second graph at same page
18    canvas2 = FigureCanvasTkAgg(fig2, master=frame_plot)
19    canvas2.draw()
20    canvas2.get_tk_widget().pack()
21
22    # Show text
23    text_output.insert(tk.END, "Top 10 sub categories:\n")
24    top_sub_categories = pd.DataFrame({
25        'Sub Category': sub_category_counts.index,
26        'Number of Products': sub_category_counts.values
27    })
28    text_output.insert(tk.END, top_sub_categories.to_string(index=False) + "\n")
29
30
```

Figure5: Second graph

I design and enter the codes for GUI and details of that.

```
89
90
91 # Creat a main window
92 root = tk.Tk()
93 root.title("Product Analysis")
94 root.geometry("900x700")
95
96 # Fram of window
97 frame_plot = tk.Frame(root)
98 frame_plot.pack(side=tk.TOP, fill=tk.BOTH, expand=True)
99
100 # scroll
101 text_output = scrolledtext.ScrolledText(root, width=70, height=10)
102 text_output.pack(padx=10, pady=10)
103
104 # Button for run the code
105 run_button = ttk.Button(root, text="Run Analysis", command=run_code)
106 run_button.pack(pady=30)
107
108 # Main loop
109 root.mainloop()
110
111
```

Figure 6: Making GUI's details

3.1. Result

The outcome of this python file was to have two main graphs that shows basic information and sort them to understand better.

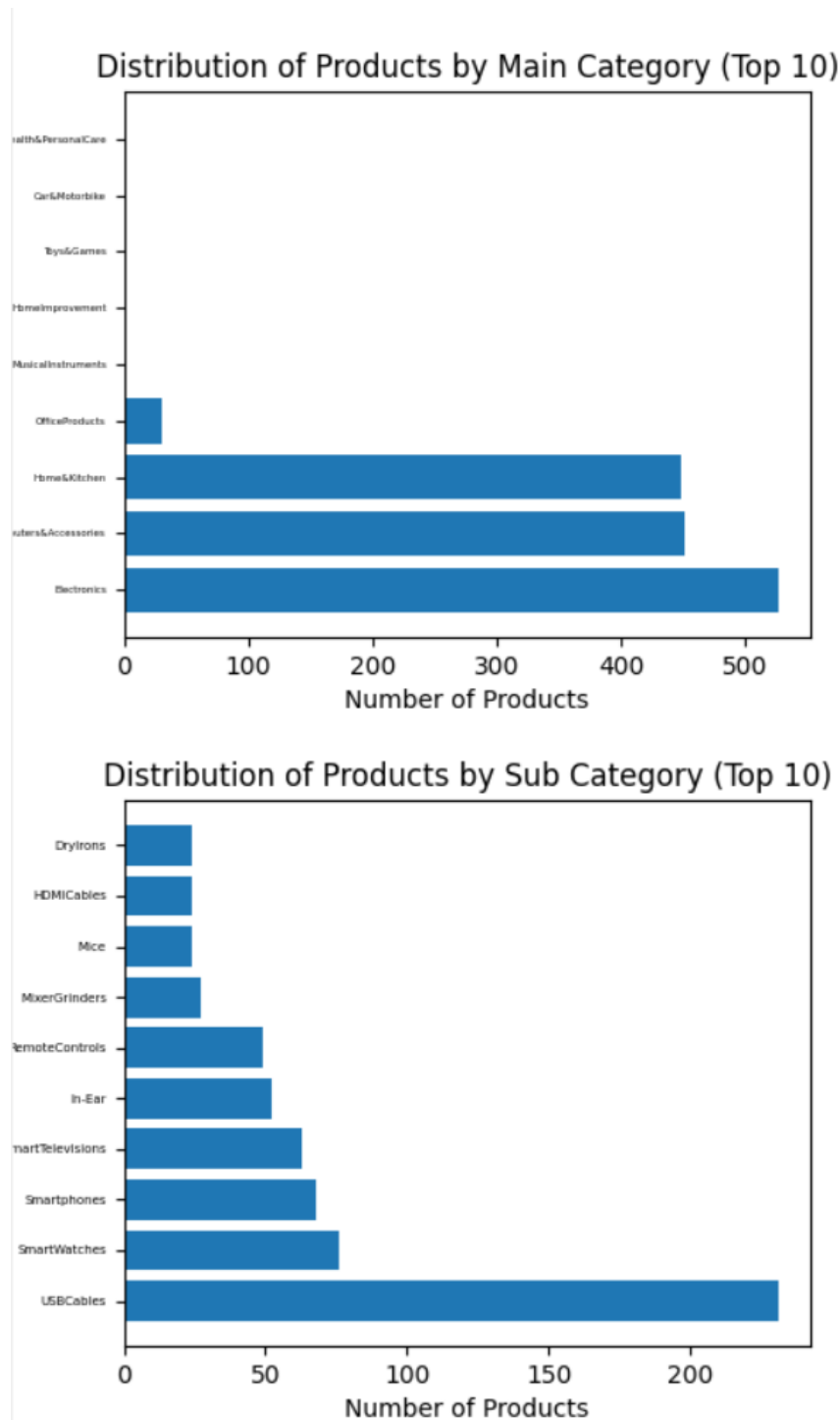


Figure 7: Results of codes

4. Customer behavior

For analyzing customer behavior in amazon, I created one other file that is related to customer behavior. I imported all libraries in this file as well. Next step was to show comparison between category and rating of products in amazon. The range of rate was 0 to 5. As I understood the categories of dataset is a lot, I decided to analyze the rates with those 10 top categories. Before running the code I add the code that is change the rating's data to number.

```
try:

    # Debugging: Print the first few rows of the DataFrame
    print("Loaded DataFrame:")
    print(df.head())

    # Debugging
    print("DataFrame Columns:")
    print(df.columns.tolist())

    # Check the data types
    print("Data Types:")
    print(df.dtypes)

    # Check if the expected columns exist
    expected_columns = ['category', 'rating']
    missing_columns = [col for col in expected_columns if col not in df.columns]

    if missing_columns:
        text_output.insert(tk.END, f"Missing columns: {' '.join(missing_columns)}\n")
        return

    # Extract main categories
    df['main_category'] = df['category'].astype(str).str.split('|').str[0]

    # Get top 10 main categories by count
    top_main_categories = df['main_category'].value_counts().nlargest(10)

    # Display the top 10 main categories in the text output
    text_output.insert(tk.END, "Top 10 Main Categories by Count:\n")
    text_output.insert(tk.END, top_main_categories.to_string() + "\n\n")
```

Figure 8: Customer data analyzing


```

# Display the top 10 main categories in the text output
text_output.insert(tk.END, "Top 10 Main Categories by Count:\n")
text_output.insert(tk.END, top_main_categories.to_string() + "\n\n")

# Convert the 'rating' column to numeric
df['rating'] = pd.to_numeric(df['rating'], errors='coerce')

# Drop rows with NaN in 'rating' if necessary
df.dropna(subset=['rating'], inplace=True)

# Calculate average ratings by main category
main_category_ratings = df.groupby('main_category')['rating'].mean().sort_values(ascending=False).nlargest(10)

# Debugging: Print average ratings by main category
print("Average Ratings by Main Category:")
print(main_category_ratings) # Print average ratings for debugging

# Create a bar chart for average ratings by main category
fig, ax = plt.subplots(figsize=(8, 5))
main_category_ratings.plot(kind='bar', ax=ax, color='skyblue')
ax.set_title('Average Rating by Main Category (Top 10)', fontsize=16)
ax.set_xlabel('Main Category', fontsize=12)
ax.set_ylabel('Average Rating', fontsize=12)
ax.tick_params(axis='x', rotation=45) # Rotate x-axis labels for better visibility

# Display the bar chart in the GUI
canvas = FigureCanvasTkAgg(fig, master=frame_plot)
canvas.draw()
canvas.get_tk_widget().pack(pady=30)

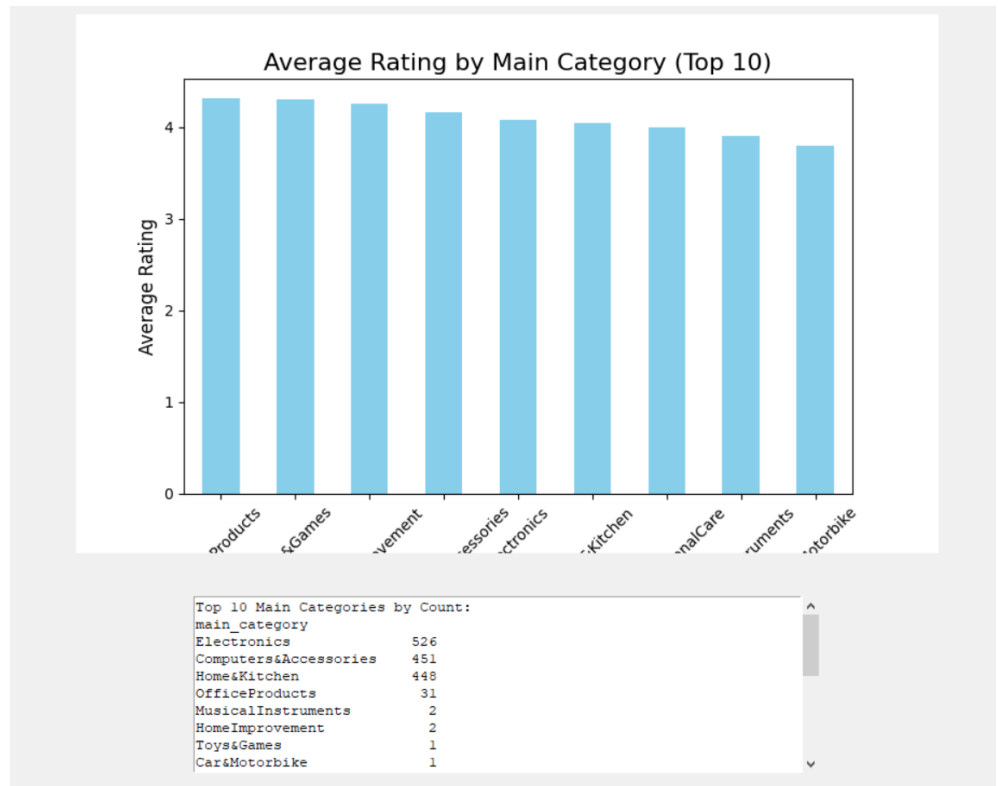
```

Figure 9: Create a second graph

4.1. Result of customer behavior

The result is the graph that display rates and categories

Figure 10:
Customer behavior's
result



5. Market trends

Analyzing this part help to industry to understand about the people's demand and observe performance and also apply strategies if it needs.

I created another python file to be separate from another and work with this closely.

I imported libraries and start to write a code that it can create graph for my purpose. Then, I selected the actual price column and discounted column comparison to main categories.

For this part I had to clean the prices columns to numeric.

```
try:

    # Convert relevant columns to numeric (Make sure that all are number)
    df['discounted_price'] = pd.to_numeric(df['discounted_price'], errors='coerce')
    df['actual_price'] = pd.to_numeric(df['actual_price'], errors='coerce')
    df.dropna(subset=['discounted_price', 'actual_price'], inplace=True)

    # Define main categories
    df['main_category'] = df['category'].astype(str).str.split('|').str[0]

    # Calculate average prices and counts for main categories
    main_category_counts = df['main_category'].value_counts()[:10]
    avg_discounted_price_by_category = df.groupby('main_category')['discounted_price'].mean().loc[main_category_counts.index]
    avg_actual_price_by_category = df.groupby('main_category')['actual_price'].mean().loc[main_category_counts.index]

    # Combine both averages into a single DataFrame
    comparison_df = pd.DataFrame({
        'Average Discounted Price': avg_discounted_price_by_category,
        'Average Actual Price': avg_actual_price_by_category
    }).reset_index()

    # Set the figure for the bar plot
    fig, ax = plt.subplots(figsize=(12, 6))

    # Set the bar width
    bar_width = 0.25
    x = range(len(comparison_df))

    # Plot bars for average discounted prices
    ax.bar(x, comparison_df['Average Discounted Price'], width=bar_width, label='Average Discounted Price', color='purple')

    # Plot bars for average actual prices
    ax.bar([p + bar_width for p in x], comparison_df['Average Actual Price'], width=bar_width, label='Average Actual Price', color='lightgreen')
```

Figure 11: Market trends code

5.1. Result of Market trends

By anticipating which products will be in greater demand, inventory levels can be optimized to prevent outages and overstocking.

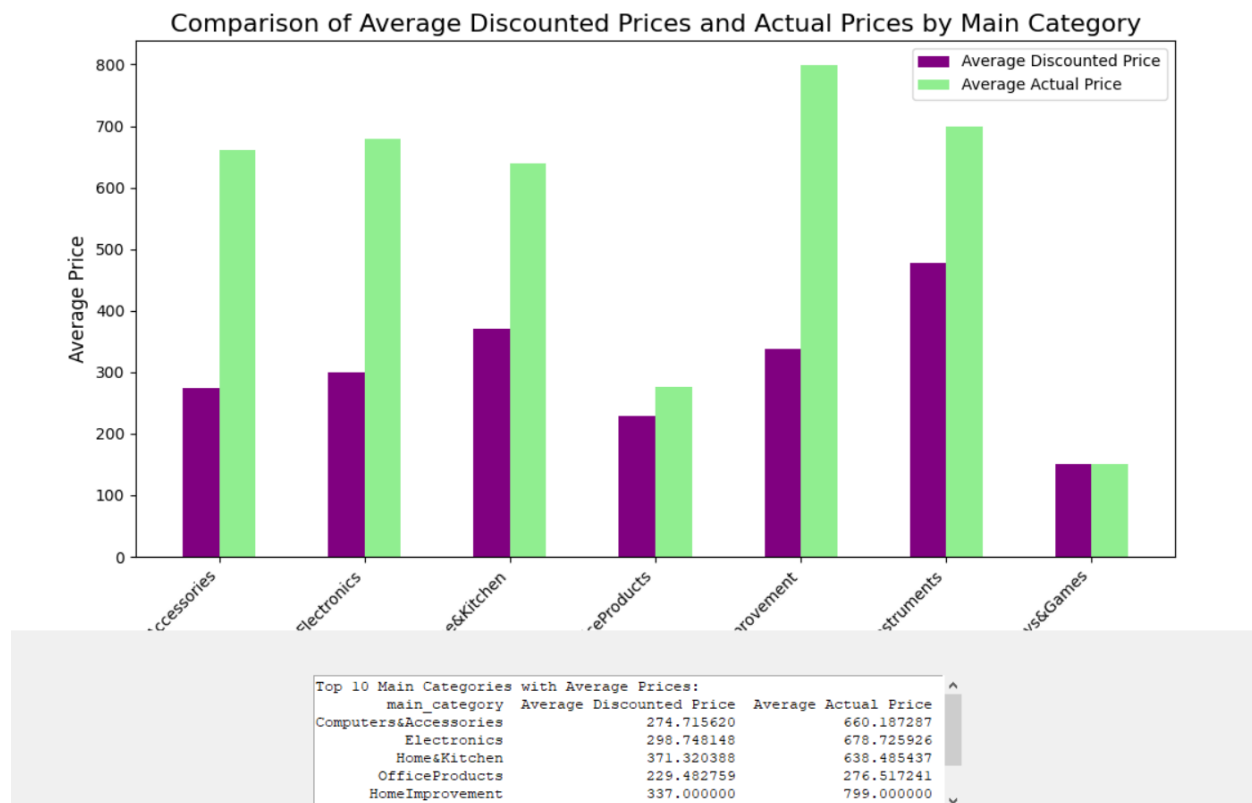


Figure 12: market trend's result

6. Errors and debugging

6.1. Error1: File Not Found

Before starting to load the CSV file, the software makes sure it is present. A user-friendly notice alerting the user to the problem is presented in the GUI if the file cannot be located.

6.2. Error2: Invalid Data Types

In order to guarantee that the data is in the right format for processing, the application incorporates type conversion stages. For example, non-numeric items are forced to NaN and the rating column is transformed to numeric values.

6.3. Debug 1: Statements to Print

Print statements were added to the code to output important details about the Data Frame at different processing phases. This involved printing the column names, displaying the data types for each column, and displaying the first few rows of the loaded Data Frame.

6.4. Debug 2: Frequent Testing

Iteratively, the code was created and tested. The program was tested to verify that all the important features implemented correctly and to find any problems right once.

7. Conclusion

I thoroughly analyzed a product dataset for this assignment, paying particular attention to ratings and price comparisons among different categories. The results of the investigation showed that a small number of people dominated the top 10 major categories, suggesting that some categories are noticeably more popular than others. In addition, the average ratings for each of these categories showed variations in the levels of consumer satisfaction, with some categories scoring noticeably higher than others.

A comparison of the discounted and actual costs revealed important trends in pricing tactics and the ways in which discounts influence consumer perception and purchase behavior. The research made clear how important data-driven insights are to Knowing customer preferences and market trends.

8. References

Codemy.com (2019). *Positioning With Tkinter's Grid System - Python Tkinter GUI Tutorial #2*. [online] YouTube. Available at: <https://www.youtube.com/watch?v=BSfbjrqIw20&list=PLCC34OHNcOtoC6GglhF3ncJ5rLwQrLGnV> [Accessed 4 Oct. 2024].

nSiS (2024). *How to plot a graph onto a tkinter canvas*. [online] Stack Overflow. Available at: <https://stackoverflow.com/questions/72542210/how-to-plot-a-graph-onto-a-tkinter-canvas> [Accessed 4 Oct. 2024].

pythonprogramming.net. (n.d.). *Python Programming Tutorials*. [online] Available at: <https://pythonprogramming.net/how-to-embed-matplotlib-graph-tkinter-gui/>.

sentdex (2014). *Tkinter tutorial Python 3.4, creating a full scale Program GUI part 1*. [online] YouTube. Available at: <https://www.youtube.com/watch?v=HjNHATw6XgY&list=PLQVvva0QuDclKx-QpC9wntnURXVJqLyk> [Accessed 4 Oct. 2024].