Assignment ID: **#748544**

*Submissions (part) of Weekly Exercises to*

**Weekly Exercises on AI-Powered Automation for Data Science Using GitHub Copilot in VS Code - May 2025**

***Week Duration:18/05/2025 - 25/05/2025***

Responses are for

Exercise 1, Task 5

Student Name : Arindam Mitra

Registration Code: STU20250430/14045

Email: [arinmitra@gmail.com](mailto:arinmitra@gmail.com)

**Table of Contents**

[5. Use Copilot Throughout & Document It 3](#_Toc201513493)

[6. Analyze Results and Write Summary: After obtaining the outputs, interpret them. Write a short summary of insights in your notebook or document: 8](#_Toc201513494)

[Branch with the highest sales: 8](#_Toc201513495)

[Highest Selling Product: 9](#_Toc201513496)

[Average Spending Comparison: 9](#_Toc201513497)

[Average Customer ratings by branch: 9](#_Toc201513498)

[Patterns in the data: 9](#_Toc201513499)

**List of tables**

[**Table 1. Summary table of VSCode Copilot assistance along with brief descriptions** 3](#_Toc201513524)

**List of figures**

[**Figure 1.** Screenshot of the VS Code IDE interface with GitHub Copilot activated 4](#_Toc201513528)

[**Figure 2.** Moving over to Jupyter Notebook setup in VS Code for rest of the exercise 4](#_Toc201513529)

[**Figure 3.** Further Code addition to Exercise1 task1 with assistance from Copilot 5](#_Toc201513530)

[**Figure 4.** GitHub Copilot suggestions for Exercise 1 task 4 from the comments before acceptance in the Jupyter Notebook 5](#_Toc201513531)

[**Figure 5.** Providing further comments into the code actually helped the AI with a fine-tuned suggestions code forward 6](#_Toc201513532)

[**Figure 6.** Setting the comments properly once, it actually prompted the AI to fine tune the code requirements and necessary Libraries were already invoked to be included 6](#_Toc201513533)

[**Figure 7.** Total sales of branch suggestion by Copilot automatically before acceptance 7](#_Toc201513534)

[**Figure 8.** Plot suggestion by Copilot for the bar chart after accepting the previous suggestion 7](#_Toc201513535)

# 5. Use Copilot Throughout & Document It

The usage of CoPilot was done throughout and supporting screen shots are inserted in this document along with a brief description below.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Serial #** | **Link to the figure in this document** | **Exercise / Task ref** | **Brief Description** | **Remarks** |
| 1. | Figure 1. Screenshot of the VS Code IDE interface with GitHub Copilot activated | Exercise 1, Task 1 | Initial task of the exercise to setup IDE, prepare environment and load the data | Started with .py file to check if the VS Code installation with GitHub Copilot was working correctly or not |
| 2. | Figure 2. Moving over to Jupyter Notebook setup in VS Code for rest of the exercise | Exercise 1, Task 1 | Initial task of the exercise to setup IDE, prepare environment and load the data | Shifted to the Jupyter Notebook in VSCode setup along with the GitHub Copilot working |
| 3. | Figure 3. Further Code addition to Exercise1 task1 with assistance from Copilot | Exercise 1, Task 1 | Adding further code to the existing cell with the code with assistance from Copilot | The assisted newer code was accepted into the working Jupyter Notebook via suitable insertion |
| 4. | Figure 4. GitHub Copilot suggestions for Exrecise 1 task 4 from the comments before acceptance in the Jupyter Notebook | Exercise 1, Task 4 | This screenshot essentially shows that in an assisted Copilot environment, the AI suggests the purpose of the code just by reading the comments | The suggested assistance was accepted and then incorporated into the comments of the actual code |
| 5. | Figure 5. Providing further comments actually helped the AI with a fine-tuned suggestions code forward | Exercise 1, Task 4 | This screenshot shows how by providing the full comments essentially empowers the AI to give the assistance to the code design | The suggestions were accepted as was given by the Copilot to include the necessary libraries in Python for the constructed code to run correctly |
| 6. | Figure 6. Setting the comments properly, actually prompted the AI to fine tune the code requirements and necessary Libraries were already invoked to be included | Exercise 1, Task 4 | This screenshot essentially shows how the full comments helped in automatically identify the required Python Libraries | This is how the code started to look like once the code suggestions were accepted |
| 7. | Figure 7. Total sales of branch suggestion by Copilot automatically before acceptance | Exercise 1, Task 4 | Screenshot to show suggestions by Copilot before accepting the suggestions | **Before** acceptance of suggestion for the code for plotting the bar chart |
| 8. | Figure 8. Plot suggestion by Copilot for the bar chart after accepting the previous suggestion | Exercise 1, Task 4 | Screenshot to show suggestions by Copilot after accepting the suggestions | **After** acceptance of suggestion for the code for plotting the bar chart |

**Table 1. Summary table of VSCode Copilot assistance along with brief descriptions**

Similarly for the other code modules, the assistance of Copilot was invoked and utilised throughout the exercise.

A screenshot of a computer

AI-generated content may be incorrect.

**Figure 1.** Screenshot of the VS Code IDE interface with GitHub Copilot activated

A screenshot of a computer

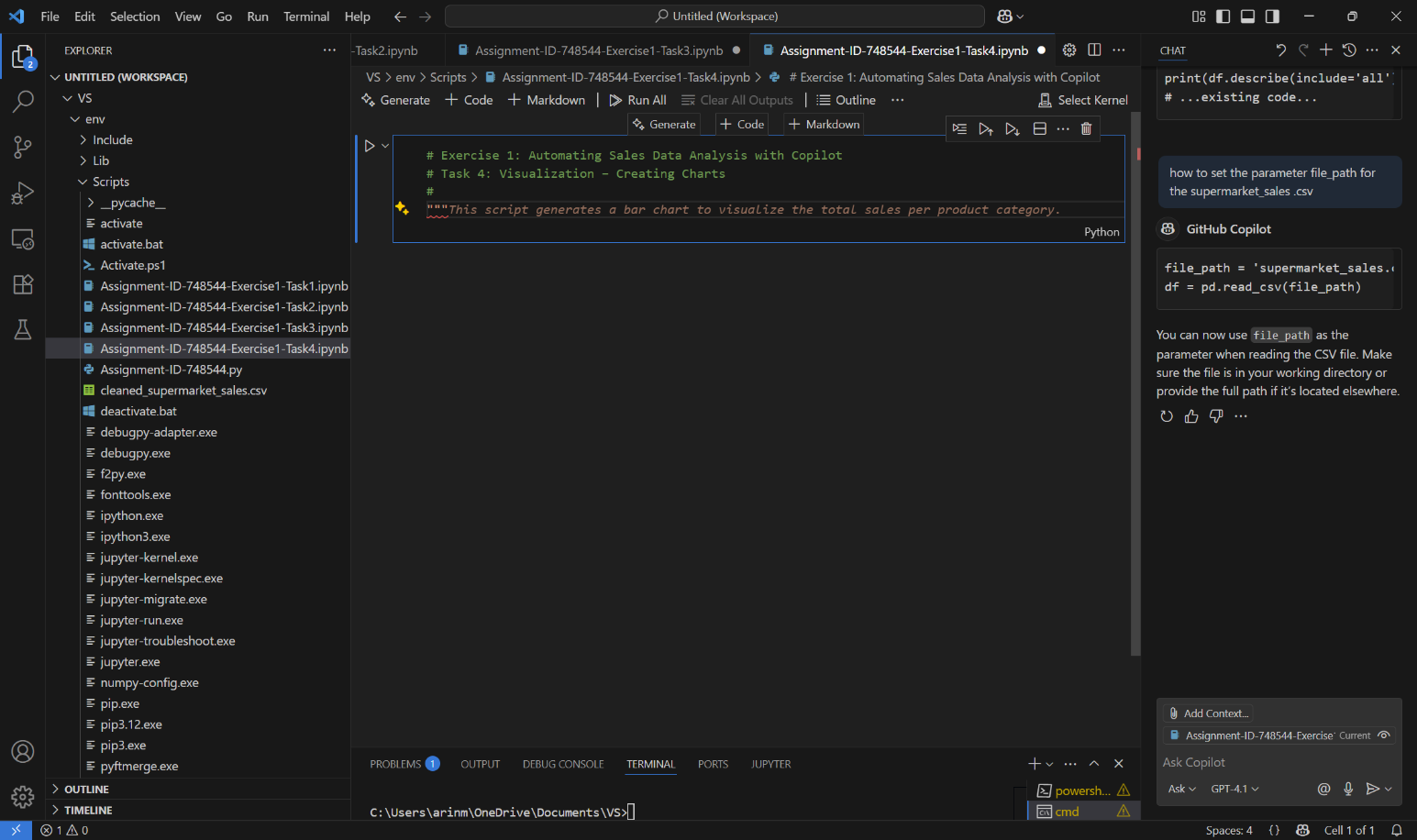
AI-generated content may be incorrect.

**Figure 2.** Moving over to Jupyter Notebook setup in VS Code for rest of the exercise

A screenshot of a computer program

AI-generated content may be incorrect.

**Figure 3.** Further Code addition to Exercise1 task1 with assistance from Copilot



**Figure 4.** GitHub Copilot suggestions for Exercise 1 task 4 from the comments before acceptance in the Jupyter Notebook

A computer screen shot of a program

AI-generated content may be incorrect.

**Figure 5.** Providing further comments into the code actually helped the AI with a fine-tuned suggestions code forward

A computer screen shot of a program

AI-generated content may be incorrect.

**Figure 6.** Setting the comments properly once, it actually prompted the AI to fine tune the code requirements and necessary Libraries were already invoked to be included

A screenshot of a computer program

AI-generated content may be incorrect.

**Figure 7.** Total sales of branch suggestion by Copilot automatically before acceptance

A screenshot of a computer screen

AI-generated content may be incorrect.

**Figure 8.** Plot suggestion by Copilot for the bar chart after accepting the previous suggestion

# 6. Analyze Results and Write Summary: After obtaining the outputs, interpret them. Write a short summary of insights in your notebook or document:

* Which branch had the highest sales? How much higher was it than others roughly (e.g. 10% higher)?
* Which product line is the top seller? Does it significantly outsell the others?
* Do members spend more than normal customers on average? Quantify the difference.
* (If analyzed) Which branch has the highest customer satisfaction rating? Any notable differences?
* What patterns or surprises do you see in the data? Write these observations in a few sentences or bullet points. This summary will be useful for your report and portfolio. You can include it as markdown text in a Jupyter notebook or as comments in your .py file (and later transfer to your submission write-up).

After analysing the results, the summary of such analysis is as given below:

## Branch with the highest sales:

Branch C is the top-performing location with ₹110,568.71 in total sales, signalling strong operational effectiveness and customer engagement. It sets a benchmark for other branches and presents an opportunity to replicate its success model across the network. It was marginally higher than the other two branches by approximately about (~5%) on the totals by the store from the graph

A graph of a graph with a bar chart

AI-generated content may be incorrect.

## Highest Selling Product:

1. **Top product line by total sales:** *Food and Beverages*
2. **Total sales value:** ₹56,144.84

Observations:

* *Food and Beverages* is the **highest revenue-generating category**, suggesting strong customer demand and high turnover in this product line.

Food and Beverages is the top-selling product line, contributing ₹56,144.84 in total sales. This highlights it as a key revenue driver and a strategic area for continued investment and growth initiatives.

## Average Spending Comparison:

1. **Members** spend an average of **₹327.79**.
2. **Normal customers** spend an average of **₹318.12**.

Observations:

* **Members spend approximately ₹9.67 more** on average than normal customers.
* This **indicates higher customer value** among members, possibly due to loyalty programs, personalized offers, or perceived benefits of membership.
* While the difference is modest (~3%), over a large customer base, this **can translate into significant revenue gains**.

Members show slightly higher average spending than normal customers, suggesting that membership programs may be effective in encouraging higher spending behavior. This insight can inform strategies for customer retention and upselling.

## Average Customer ratings by branch:

Branch A – 7.027059; Branch B – 6.818072 and Branch C – 7.072866

Branch C leads in customer satisfaction with an average rating of 7.07, followed closely by Branch A. Branch B lags slightly, indicating potential areas for improvement in customer experience and service delivery.

## Patterns in the data:

The overall performance is showing a regular trend of good performance with respect to sales, service delivery or overall customer satisfactions. However, there are scope for improving upon the current performances and expand the business by doing further analysis on the data.