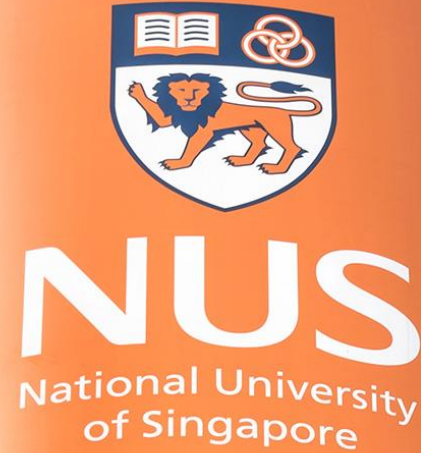


NUS Coding Workshop: Copilot

Yurui :: Teaching Assistant :: Department of Computer and Engineering



National University of Singapore

About me

- Shao Yurui
- Electrical & Computer Engineering
- Freshly graduated from NUS last year 2024
- Full time Teaching Assistant
- Ph.D. Candidate in ML for healthcare



Fun Fact about Me #1

- My parents are living in the northern part of China with a super cute Shiba Inu



Fun Fact about Me #2



- I worked as a Bartender during Year 1 December Break (circuit breaker)



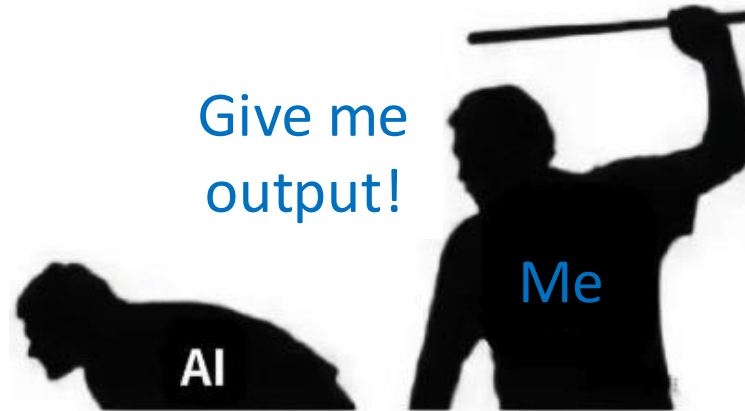
Content Table

- **What is VScode Copilot:**
 - Transitioning from Command line to IDE to Copilot
 - Install Vscode, Anaconda, and register GitHub account
- **Features of copilot:**
 - Generate code, Fix bug, Learn with Copilot
- **NUS AI Policy:**
 - DO and DON'T
- **Roadmap of learning AI:**
 - Courses recommendation
- **Hands on activity:**
 - Create a Python Notebook report

Why learn how to use AI

What you imagine life will be ..

I want to exterminate
humanity.



And the reality:

AI can do all of this.
So what exactly are you here for?

Sorry boss...





What is VScode Copilot

10 mins

File Edit Selection View Go Run Terminal Help

EXPLORER

▼ FOG_PREDICTION

> .vscode

> data

> src

> __pycache__

> arduino_comm

> CNN_Models

> Inn

> LNN_model

> LNN_Models

> Utils

augmentation.ipynb

buzzer_demo.ipynb

data_exploration.ipynb

DataVisualisatio...

fols_preprocess_comp...

Impact_of_Preprocessi...

Inn_split_test.ipynb

Inn_testing-Copy1.ipynb

Inn_testing.ipynb

LNN_Window_target_L...

LNN_Window_target.ip...

main.ipynb

main.py

model_test.ipynb

my_functions.ipynb

notebook.ipynb

pipeline.ipynb

preproce111ss.ipynb

preprocess.ipynb

rnn_model.pth

S121_trial_1_S_L.csv

scanner.ipynb

Serial_com.ipynb

OUTLINE

TIMELINE

PROJECT COMPONENTS

data_exploration.ipynb

DataVisualisation.ipynb

src > DataVisualisation.ipynb > plt.figure(figsize=(12, 4))

Generate + Code + Markdown | Run All | Restart | Clear All Outputs | Jupyter Variables | Outline | base (Python 3.12.7)

```
for file in os.listdir(patient_folder_path):
    if file.endswith('.csv'):
        file_path = os.path.join(patient_folder_path, file)
        # Read the CSV file and append to the list
        tmp_df = pd.read_csv(file_path)
        tmp_df = tmp_df.loc[:, ~tmp_df.columns.str.contains('^Unnamed')]
        dataframes.append(tmp_df)

return dataframes
```

[7] ✓ 0.0s Python

```
my_dataframes = readDataBase()
```

[8] ✓ 1.2s Python

```
plt.figure(figsize=(12, 4))
plt.plot(my_dataframes[0]['Gyro_x_left'])
plt.title('Gyro_x_left from my_dataframes[0]')
plt.xlabel('Index')
plt.ylabel('Gyro_x_left')
plt.show()
```

[36] ✓ 0.1s Python

CHAT

return x

...existing code...

Similar code found with 1 license type - View matches

@workspace plot out my_dataframes[0]['Gyro_x_left']

DataVisualisation.ipynb • Cell 5:1

</> Workspace

> Used 33 references

✓ Collecting workspace information

You can plot my_dataframes[0]['Gyro_x_left'] using matplotlib as follows:

...existing code...
plt.figure(figsize=(12, 4))
plt.plot(my_dataframes[0]['Gyro_x_left'])
plt.title('Gyro_x_left from my_dataframes[0]')
plt.xlabel('Index')
plt.ylabel('Gyro_x_left')
plt.show()
...existing code...

You can add this code to a new cell in

DataVisualisation.ipynb .

🔄 🍏 🗨️ ...

Add Context...

DataVisualisation.ipynb • Cell 5 Current cell 🔍

Ask Copilot

Ask v GPT-4.1 v @ 🗨️ 🔍

Files

Python Notebook

Copilot

© Copyright National University of Singapore. All Rights Reserved.

Install Visual Studio Code (VScode)

- <https://code.visualstudio.com/Download> (Just google VScode)

Visual Studio Code Docs Updates Blog API Extensions FAQ GitHub Copilot MCP

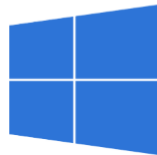
Search Docs

Download

Try [MCP servers](#) to extend agent mode in VS Code!

Download Visual Studio Code

Free and built on open source. Integrated Git, debugging and extensions.



↓ Windows

Windows 10, 11

User Installer [x64](#) [Arm64](#)
System Installer [x64](#) [Arm64](#)
.zip [x64](#) [Arm64](#)



↓ .deb

Debian, Ubuntu

↓ .rpm

Red Hat, Fedora, SUSE

.deb [x64](#) [Arm32](#) [Arm64](#)
.rpm [x64](#) [Arm32](#) [Arm64](#)
.tar.gz [x64](#) [Arm32](#) [Arm64](#)
Snap [Snap Store](#)



↓ Mac

macOS 11.0+

.zip [Intel chip](#) [Apple silicon](#) [Universal](#)
CLI [Intel chip](#) [Apple silicon](#)

Install Anaconda

- <https://www.anaconda.com/download> (Just google Anaconda)



Distribution

Register to get everything you need to get started on your workstation including Cloud Notebooks, Navigator, AI Assistant, Learning and more.

- ✓ Easily search and install thousands of data science, machine learning, and AI packages
- ✓ Manage packages and environments from a desktop application or work from the command line
- ✓ Deploy across hardware and software platforms
- ✓ Distribution installation on Windows, MacOS, or Linux

Free Download

Get access in 30 seconds. Completely free.*

Get Started >

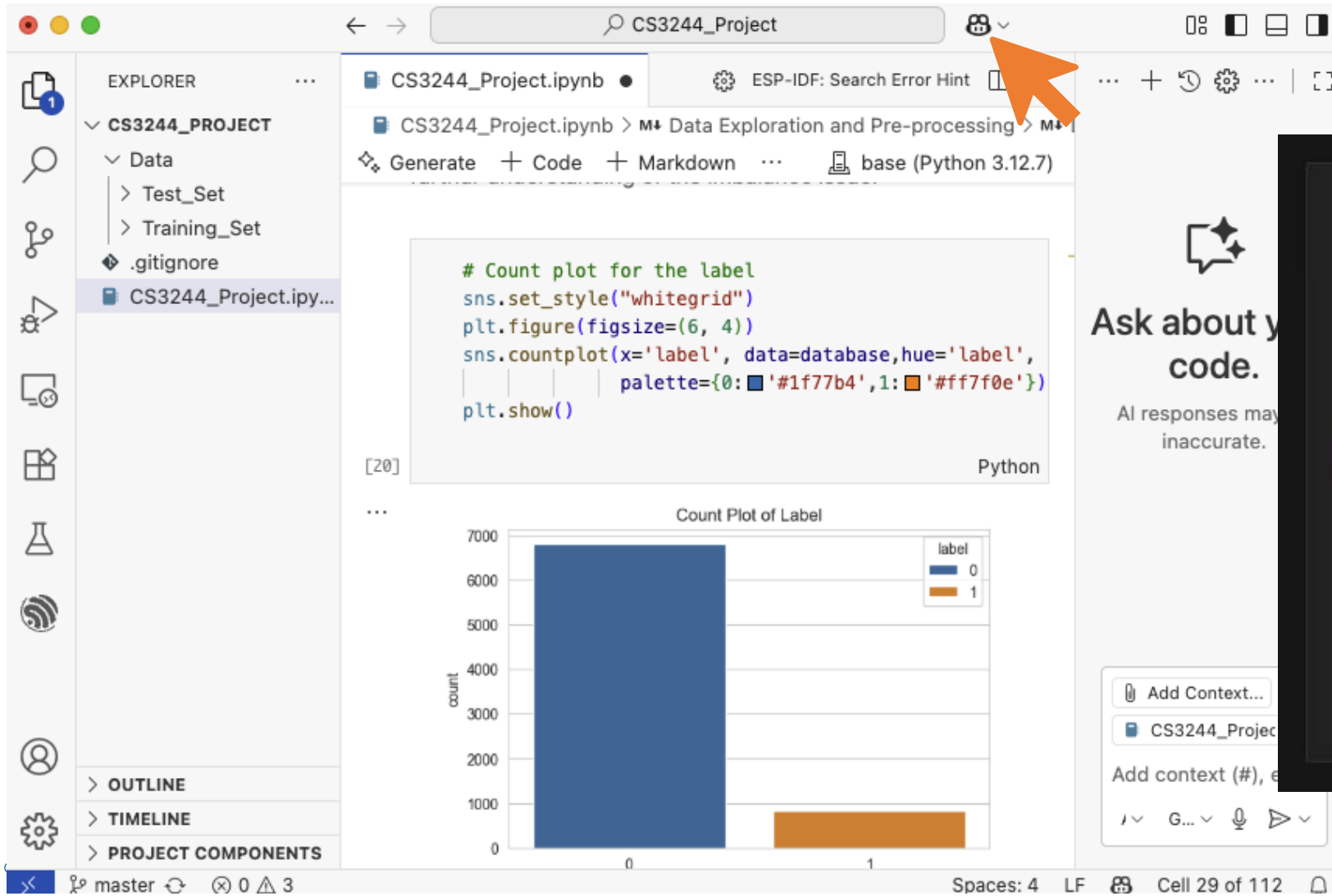
Returning Users >

*Subject to our [Terms of Service](#). Use of Anaconda's offerings at an organization of more than 200 employees/contractors requires a paid business license unless your organization is eligible for discounted or free use. [See Pricing](#).

[Skip registration](#)



Sign up GitHub Account

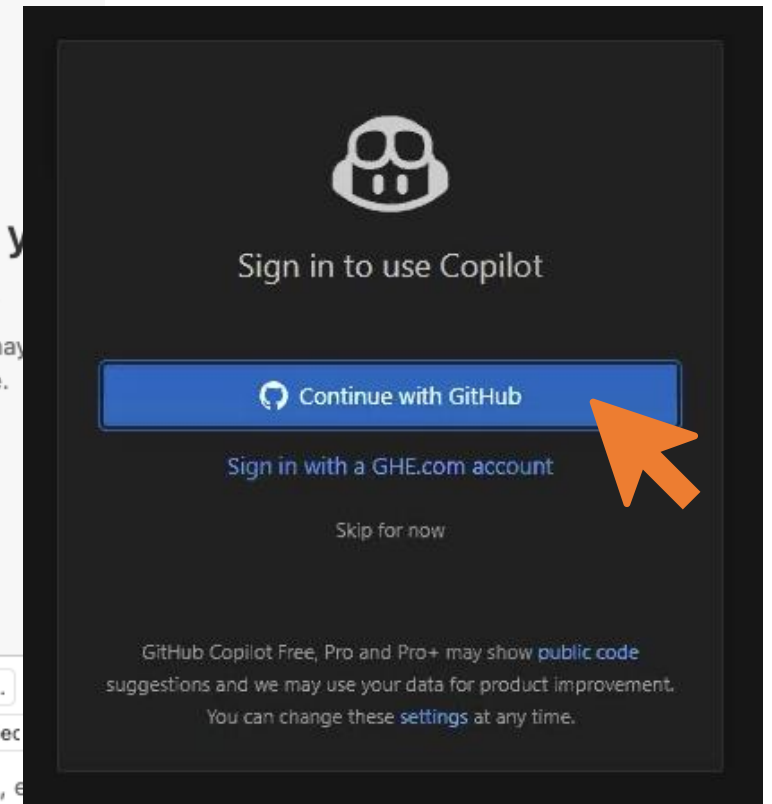


The screenshot shows a Jupyter Notebook interface with a sidebar on the left containing an Explorer view. The main area displays a Python code cell with the following code:

```
# Count plot for the label
sns.set_style("whitegrid")
plt.figure(figsize=(6, 4))
sns.countplot(x='label', data=database, hue='label',
              palette={0: '#1f77b4', 1: '#ff7f0e'})
plt.show()
```

Below the code is a bar chart titled "Count Plot of Label". The x-axis is labeled "label" and has two categories: 0 and 1. The y-axis is labeled "count" and ranges from 0 to 7000. The bar for label 0 is blue and reaches a count of approximately 6800. The bar for label 1 is orange and reaches a count of approximately 800. A legend on the right indicates that label 0 is represented by blue and label 1 by orange.

On the right side of the notebook, there is a sidebar with a GitHub logo and the text "Ask about your code." Below this, there is a button labeled "Continue with GitHub" and a link "Sign in with a GHE.com account". An orange arrow points to the "Continue with GitHub" button.



The overlay shows the GitHub logo at the top. Below it, the text "Sign in to use Copilot" is displayed. A prominent blue button with the GitHub logo and the text "Continue with GitHub" is in the center. Below this button, there is a link "Sign in with a GHE.com account" and the text "Skip for now". At the bottom, there is a small disclaimer: "GitHub Copilot Free, Pro and Pro+ may show public code suggestions and we may use your data for product improvement. You can change these settings at any time." An orange arrow points to the "Continue with GitHub" button.

Sign up GitHub Account

- <https://github.com/signup> (Just google GitHub)

Create your free account

Explore GitHub's core features for individuals and organizations.

See what's included ▾



Sign up to GitHub

Email*

Email

Use school email !!

Password*

.....

Password should be at least 15 characters OR at least 8 characters including a number and a lowercase letter.

Username*

ShaoYurui

Username may only contain alphanumeric characters or single hyphens, and cannot begin or end with a hyphen.

Your Country/Region*

Vietnam ▾

For compliance reasons, we're required to collect country information to send you occasional updates and announcements.

Email preferences

☐ Receive occasional product updates and announcements

Create account >



GitHub Education Benefit with NUS email



Yurui (ShaoYurui)

Your personal account

[Go to your personal profile](#)

Public profile

Account

Appearance

Accessibility

Notifications

Access

Billing and licensing ^

Overview

Usage

Budgets and alerts

Licensing

Payment information

Payment history

Additional billing details

Education benefits

Emails

Password and authentication

Sessions

SSH and GPG keys

GitHub Education



Education Benefits

You have a current student coupon applied. Find more information on your benefits [here!](#)

[Start an application](#)



Coupon applied



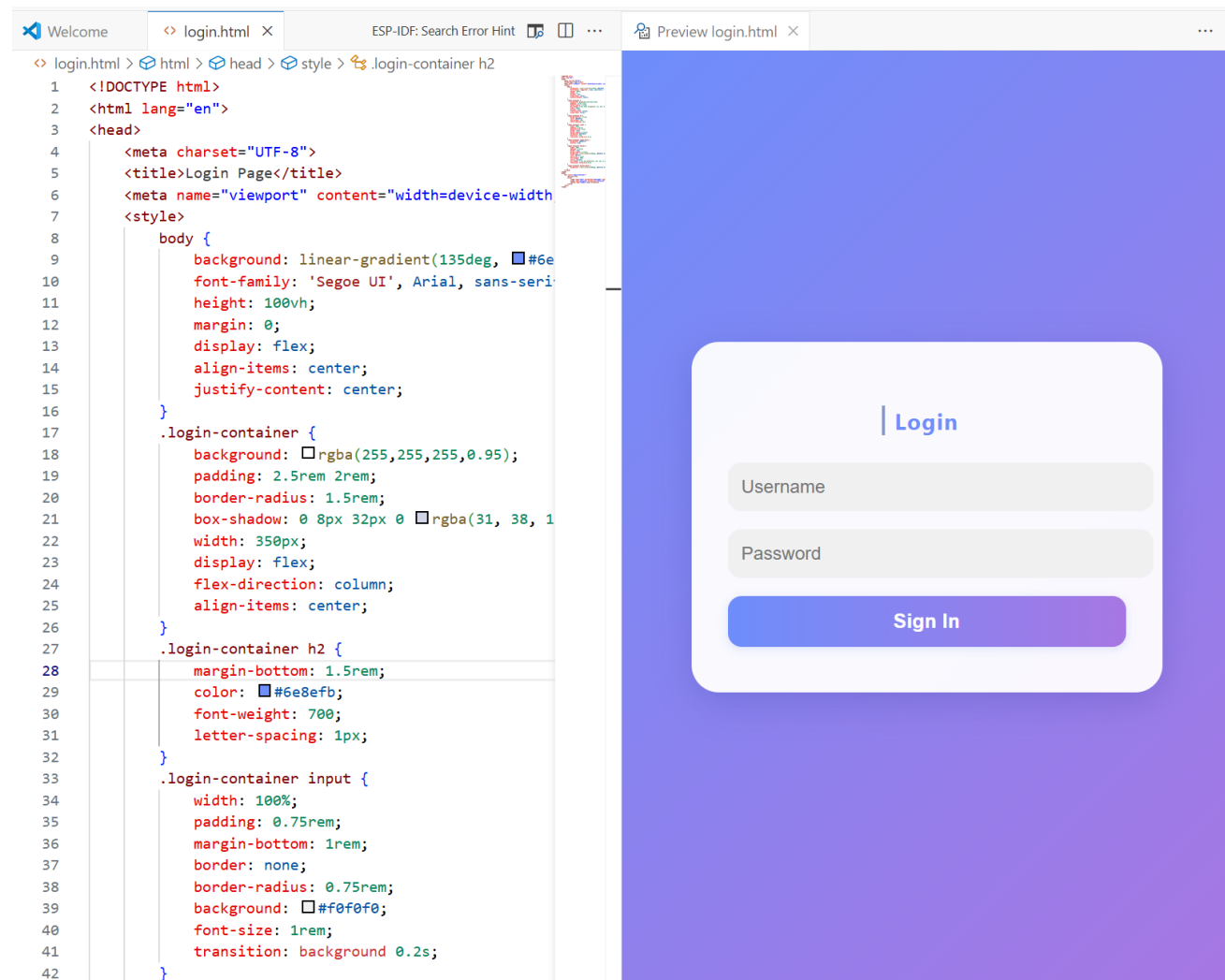
Expires in almost 2 years



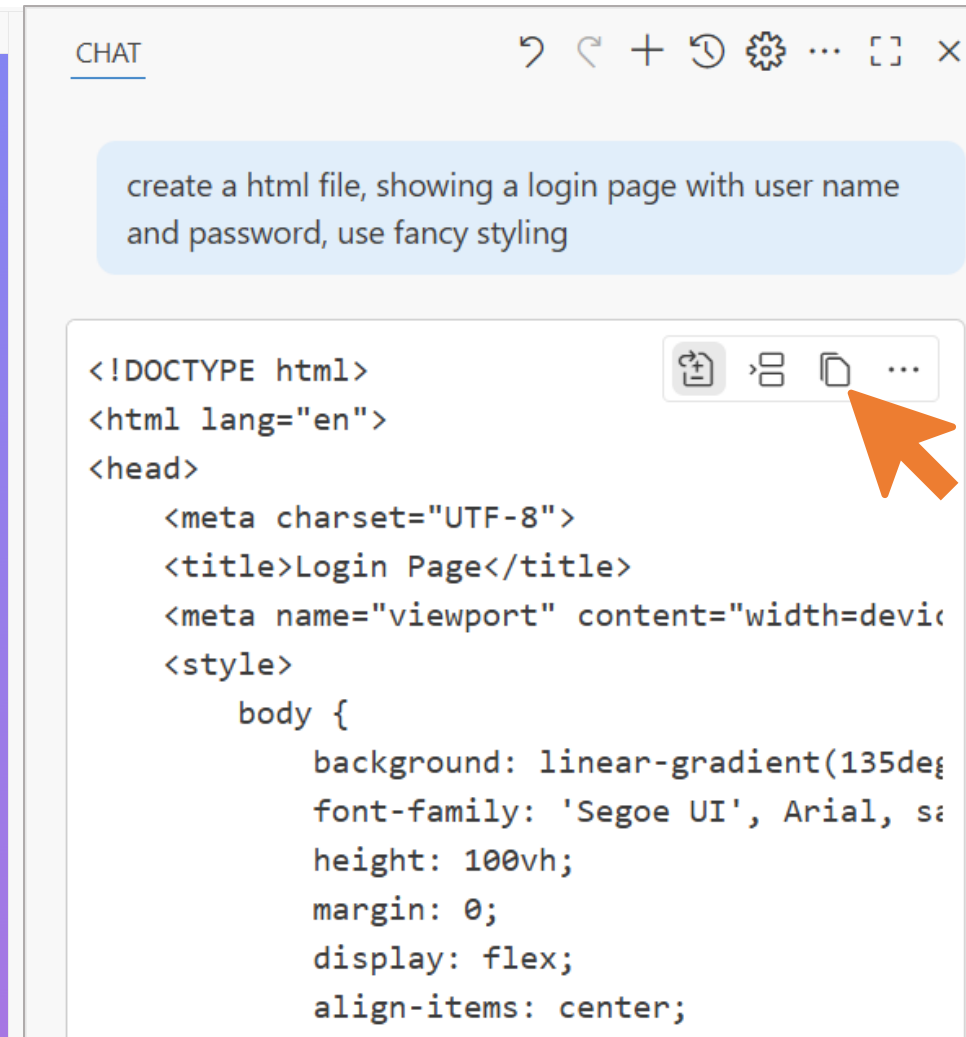
Features of Copilot

15 mins

Generate Code (ASK)



```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Login Page</title>
  <meta name="viewport" content="width=device-width, height=device-height">
  <style>
    body {
      background: linear-gradient(135deg, #6e8efb, #a6c1ee);
      font-family: 'Segoe UI', Arial, sans-serif;
      height: 100vh;
      margin: 0;
      display: flex;
      align-items: center;
      justify-content: center;
    }
    .login-container {
      background: #ffffff;
      padding: 2.5rem 2rem;
      border-radius: 1.5rem;
      box-shadow: 0 8px 32px 0 #313841, 0 0 0 1px #313841;
      width: 350px;
      display: flex;
      flex-direction: column;
      align-items: center;
    }
    .login-container h2 {
      margin-bottom: 1.5rem;
      color: #6e8efb;
      font-weight: 700;
      letter-spacing: 1px;
    }
    .login-container input {
      width: 100%;
      padding: 0.75rem;
      margin-bottom: 1rem;
      border: none;
      border-radius: 0.75rem;
      background: #f0f0f0;
      font-size: 1rem;
      transition: background 0.2s;
    }
    .login-container button {
      width: 100%;
      padding: 0.75rem;
      margin-top: 1rem;
      border: none;
      border-radius: 0.75rem;
      background: #6e8efb;
      color: white;
      font-weight: 700;
      letter-spacing: 1px;
    }
  </style>
</head>
<body>
  <div>
    <h2>Login</h2>
    <div>
      <input type="text" value="Username" />
      <input type="password" value="Password" />
      <button>Sign In</button>
    </div>
  </div>
</body>
</html>
```



CHAT

create a html file, showing a login page with user name and password, use fancy styling

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Login Page</title>
  <meta name="viewport" content="width=device-width, height=device-height">
  <style>
    body {
      background: linear-gradient(135deg, #6e8efb, #a6c1ee);
      font-family: 'Segoe UI', Arial, sans-serif;
      height: 100vh;
      margin: 0;
      display: flex;
      align-items: center;
      justify-content: center;
    }
    .login-container {
      background: #ffffff;
      padding: 2.5rem 2rem;
      border-radius: 1.5rem;
      box-shadow: 0 8px 32px 0 #313841, 0 0 0 1px #313841;
      width: 350px;
      display: flex;
      flex-direction: column;
      align-items: center;
    }
    .login-container h2 {
      margin-bottom: 1.5rem;
      color: #6e8efb;
      font-weight: 700;
      letter-spacing: 1px;
    }
    .login-container input {
      width: 100%;
      padding: 0.75rem;
      margin-bottom: 1rem;
      border: none;
      border-radius: 0.75rem;
      background: #f0f0f0;
      font-size: 1rem;
      transition: background 0.2s;
    }
    .login-container button {
      width: 100%;
      padding: 0.75rem;
      margin-top: 1rem;
      border: none;
      border-radius: 0.75rem;
      background: #6e8efb;
      color: white;
      font-weight: 700;
      letter-spacing: 1px;
    }
  </style>
</head>
<body>
  <div>
    <h2>Login</h2>
    <div>
      <input type="text" value="Username" />
      <input type="password" value="Password" />
      <button>Sign In</button>
    </div>
  </div>
</body>
</html>
```

Generate Code (AGENT)

The screenshot shows a code editor with the following HTML and CSS code:

```
<> html > html > head > style > button
1 <!-- You can save this as login.html and open in your browser -->
2 <!DOCTYPE html>
3 <html lang="en">
4 <head>
5   <meta charset="UTF-8">
6   <title>Login Page</title>
7   <link href="https://fonts.googleapis.com/css?family=Montserrat:"
8   <style>
9     body { font-family: Arial, sans-serif; background:
10     body {
11       font-family: 'Montserrat', Arial, sans-serif;
12       background: linear-gradient(135deg, #74ebd5 0%, #AC
13       min-height: 100vh;
14       margin: 0;
15       display: flex;
16       align-items: center;
17       justify-content: center;
18     }
19     .login-container {
20       width: 300px;
21       margin: 100px auto;
22       background: #fff;
23       padding: 30px;
24       border-radius: 8px;
25       box-shadow: 0 0 10px rgba(0,0,0,0.1);
26     }
27     width: 350px;
28     background: rgba(255,255,255,0.95);
29     padding: 40px 30px 30px 30px;
30     border-radius: 18px;
```

The right-hand panel shows a preview of the login page with the text "use fancy styling". Below the preview, there is a chat interface with the following text:

The login page now uses fancy styling with gradients, rounded corners, modern input effects, and a stylish button. Open the file in your browser to see the enhanced design. If you need further customization or features, let me know!

Below the chat interface, there is a status bar showing "1 file changed" and "Keep Undo" buttons. The chat interface also shows "Add Context..." and "html +" buttons. The status bar at the bottom shows "Col 25 Spaces: 4 UTF-8 {} HTML" and a bell icon.

Fix Bugs

```
Welcome | login.html | fix demo.py [icon] X
fix demo.py > ...
1  import tkinter as tk
2
3  # Create the main window
4  root = tk.Tk()
5  root.title("Python Graphic Demo")
6  root.geometry("300x200")
7
8  # Add a label
9  label = tk.Label(root, text="Hello, World!", font=("Segoe UI",
10 label.pack(pady=40)
11
12 # Add a button
13 def on_click():
14     label.config(text="Button Clicked!")
15
16 button = tk.Button(root, text="Click Me", commaand=on_click, fo
17 button = tk.Button(root, text="Click Me", command=on_click, for
18 button.pack(pady=10)
19
20 # Start the main event loop
21 root.mainloop()
```

@workspace /fix

fix demo.py:1-20

Workspace

> Used 1 reference

Pseudocode Plan:

1. Import tkinter.
2. Create main window, set title and geometry.
3. Add a label.
4. Define button click handler.
5. Add a button (fix typo in 'command' argument).

Code:

```
import tkinter as tk

# Create the main window
root = tk.Tk()
```




“Context”: What is known to Copilot

```
demo_workspace.py U x ESP-IDF: Search Error Hint ...  
demo_workspace.py  
1 from hello import print_hello  
2  
3 print_hello()
```

```
/demo_workspace.py  
ASJDBIyf98a7s6dyriugasdf7i843
```

```
hello.py U x ...  
hello.py > print_hello  
1 def print_hello():  
2     print("ASJDBIyf98a7s6dyriugasdf7i843")
```

@workspace /fix print hello()
demo_workspace.py

 **Workspace**
> Used 1 reference

Context (#)

demo_workspace.py U × ESP-IDF: Search Error Hint

demo_workspace.py

```
1 from hello import print_hello
2
3 print_hello()
```

CHAT

#file:demo_workspace.py #file:demo_fix.py fix print hello

demo_fix.py

demo_workspace.py

demo_workspace.py

> Used 2 references

To fix the `print_hello` function, you need to ensure that the `hello.py` module exists and defines a `print_hello` function. Here's how you can create or update `hello.py`:

```
def print_hello():
    print("Hello, World!")
```

Now, your `demo_workspace.py` will work as expected:

Add Context...

demo_workspace.py ×

Add context (#), extensions (@), commands (/)

Ask GPT-4.1



Context (highlight)

The image shows a code editor interface with a file named `demo_fix.py` open. The code is a Python script using Tkinter to create a simple GUI. A red rectangle highlights lines 12-14, which define a function `on_click` that updates the label text when a button is clicked. To the right, a chat window is open, displaying the same code snippet. A red rectangle in the chat window highlights the context selection dropdown, which shows `demo_fix.py:12-14` as the selected context.

```
demo_fix.py > ...
1  import tkinter as tk
2
3  # Create the main window
4  root = tk.Tk()
5  root.title("Python Graphic Demo")
6  root.geometry("300x200")
7
8  # Add a label
9  label = tk.Label(root, text="Hello, World!", font=("Segoe
10 label.pack(pady=40)
11
12 # Add a button
13 def on_click():
14     label.config(text="Button Clicked!")
15
16 button = tk.Button(root, text="Click Me", command=on_click)
17 button.pack(pady=10)
18
19 # Start the main event loop
20 root.mainloop()
```

CHAT

```
root.geometry("300x200")

# Add a label
label = tk.Label(root, text="Hello, World
label.pack(pady=40)

# Add a button
def on_click():
    label.config(text="Button Clicked!")

button = tk.Button(root, text="Click Me",
button.pack(pady=10)

# Start the main event loop
root.mainloop()
```

Add Context ... demo_fix.py:12-14

Add context (#), extensions (@), commands (/)

Ask GPT-4.1

Context

"Dear Professors,

Good day, hope you are doing well ! I would like to inform you that I will be on leave from 18 Jul to 31 Jul. Thank you so much for the continued support!

Thank you for your time and support.

Sincerely

Yurui"

This is the context you provided

Improve this email |



Context (highlight)

Introducing GPT-5

ChatGPT now has our smartest, fastest, most useful model yet, with thinking built in — so you get the best answer, every time.

```
"# Add a button  
def on_click():  
    label.config(text="Button Clicked!")"
```

This is the context you provided

Explain this code



Learning with Copilot

demo.py > ...

```
1 # Simple Flask server for login
2 from flask import Flask, render_template_string, request
3
4 app = Flask(__name__)
5
6 LOGIN_HTML = """
7 <!DOCTYPE html>
8 <html>
9 <body>
10     <form method="POST">
11         <input name="username" placeholder="Username" required>
12         <input name="password" type="password" placeholder="Password" required>
13         <button type="submit">Login</button>
14     </form>
15     {% if message %}
16     <p>{{ message }}</p>
17     {% endif %}
18 </body>
19 </html>
20 """
21
22 USERS = {"user1": "pass1", "user2": "pass2"}
23
24 @app.route("/", methods=["GET", "POST"])
25 def login():
26     message = ""
27     if request.method == "POST":
```

I want to learn about how to set up a server and handle basic login and password, give me some exercises and example to learn

<> login.html:1-42

> Used 1 reference

Here are some beginner exercises and examples to help you learn about setting up a server and handling basic login/password:

Exercises

1. Set up a simple HTTP server

- Use Python's `http.server` or Node.js `express` to serve your HTML login page.

2. Handle form submission

- Modify your login page to POST username and password to the server.
- On the server, print the received data.



Summary

- **Mode**
 - Ask
 - Agent
- **Slash Command**
 - \fix
 - \tests
- **Context**
 - Cursor Highlight
 - # Files
 - @workspace



Summary

Additional Resources

- Get Started with GitHub Copilot in VS Code (2025)
<https://www.youtube.com/watch?v=vdBxfFVXnc0&t=625s>
- Copilot Best Practices (What Not To Do)
<https://www.youtube.com/watch?v=2q0BoioYSxQ>



NUS AI Policy

5 mins



NUS AI Policy

- Policy for Use of AI in Teaching & Learning (2024)

<https://ctlit.nus.edu.sg/ai-community-of-practice/policies/>

- The NUS AI Guidelines for Students can be found as follows:

https://libguides.nus.edu.sg/new2nus/ai_guidelines_infographics



NUS AI Policy:

- Guidelines on the Use of AI Tools for Academic Work
 - **Don't Use AI to Plagiarize**
 - Generating an output and presenting it as your own work or idea without attribution.
 - **Proper uses of AI tools**
 - Gathering information and looking up explanations for basic concepts.
 - Generating output for critique and analysis, for self-learning, or to compare against one's work for self-evaluation and improvement.
 - Help with proof reading and editing writing
 - **Acknowledging your use of AI**

Acknowledging your use of AI

You can give this acknowledgement through a note or “methods section” at the end of the assignment explaining, e.g., which AI tools were used, in which parts of the process they were used, what were the prompts used to generate results, and what you did with the outputs to add value. One way this can be done is in a tabular form as shown below:

AI Tool Used	Prompt and output	How the output is used in the assignment

- Policy for Use of AI in Teaching & Learning (2024)
<https://ctlit.nus.edu.sg/ai-community-of-practice/policies/>
- Page 7

Example: Acknowledging your use of AI

AI Tool Used	Prompt and output	How the output is used in the assignment
ChatGPT	<p>Input:</p> <p>read csv in python</p> <p>output:</p> <p>In Python, the most common way to read a CSV file is with pandas.</p> <p>Here's the basic example:</p> <pre>import pandas as pd</pre> <p># Read a CSV file</p> <pre>df = pd.read_csv("file.csv")</pre> <p># Display the first 5 rows</p> <pre>print(df.head())</pre>	Paste and modify to read csv file

NUS AI Policy:

**Never dump the course materials
into ChatGPT (notes, tutorials ...)**

ChatGPT ▾

What are you working on?

+ solve this:|



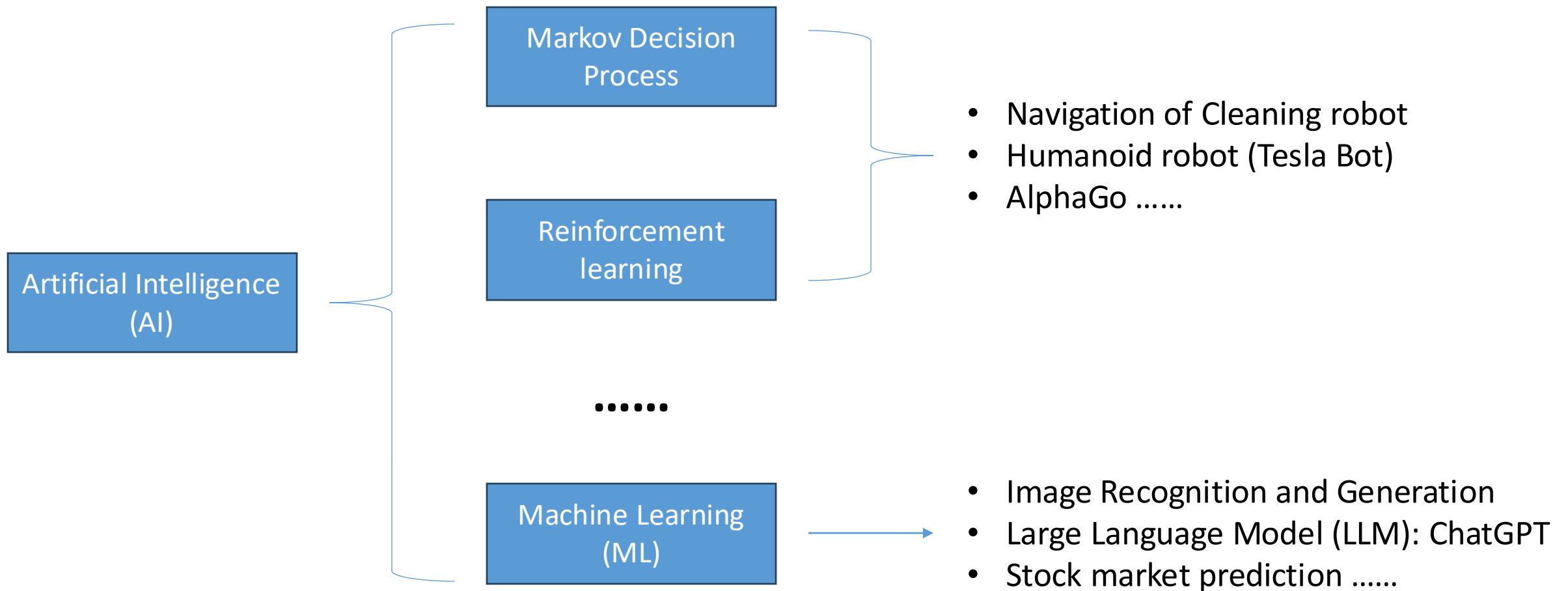
EE2026_Tutorial-
5 (wk7)_2024



Roadmap of Learning AI

5 mins

Development of AI



Development of ML

Machine Learning
(ML)

1960s

Classical
Machine Learning

1967

Neural Network
Multilayer Perceptron

1989

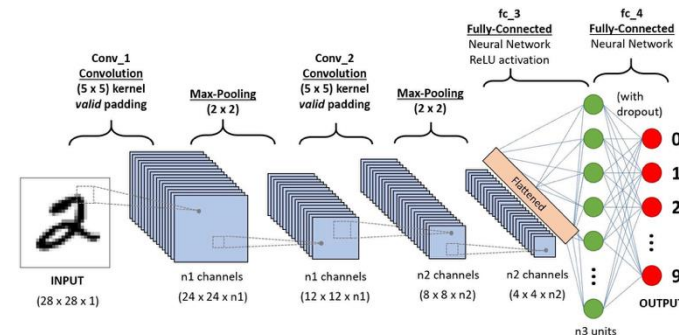
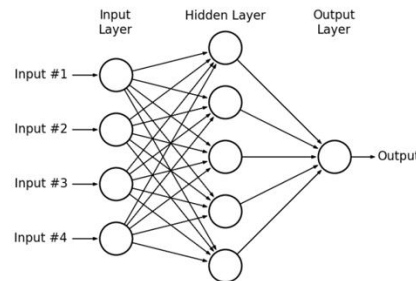
Convolutional Neural
Networks

2017

Transformer

Mathematical Model:

- Support Vector Machine
- Random Forest



EE2211

Introduction to Machine Learning

EE3703

Machine Learning with Applications

EE4212

Computer Vision

CS4248

Natural Language Processing



Hands on Activity

15 mins

DIY

Create a Python Notebook, write these down in the first cell.

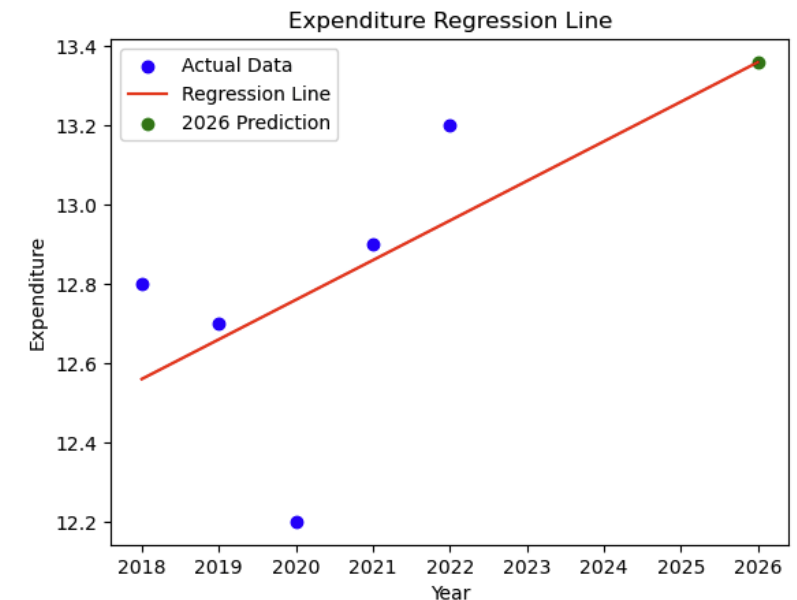
```
years = [2018, 2019, 2020, 2021, 2022]
expenditure = [12.8, 12.7, 12.2, 12.9, 13.2]
```

[4] ✓ 0.0s

Python

TODO:

1. Get a regression line from the past year data
2. Predict the value for 2026
3. Plot out the graph



Feedback



- <https://forms.office.com/r/FkaRnTcmm4?origin=IprLink>



Thank YOU