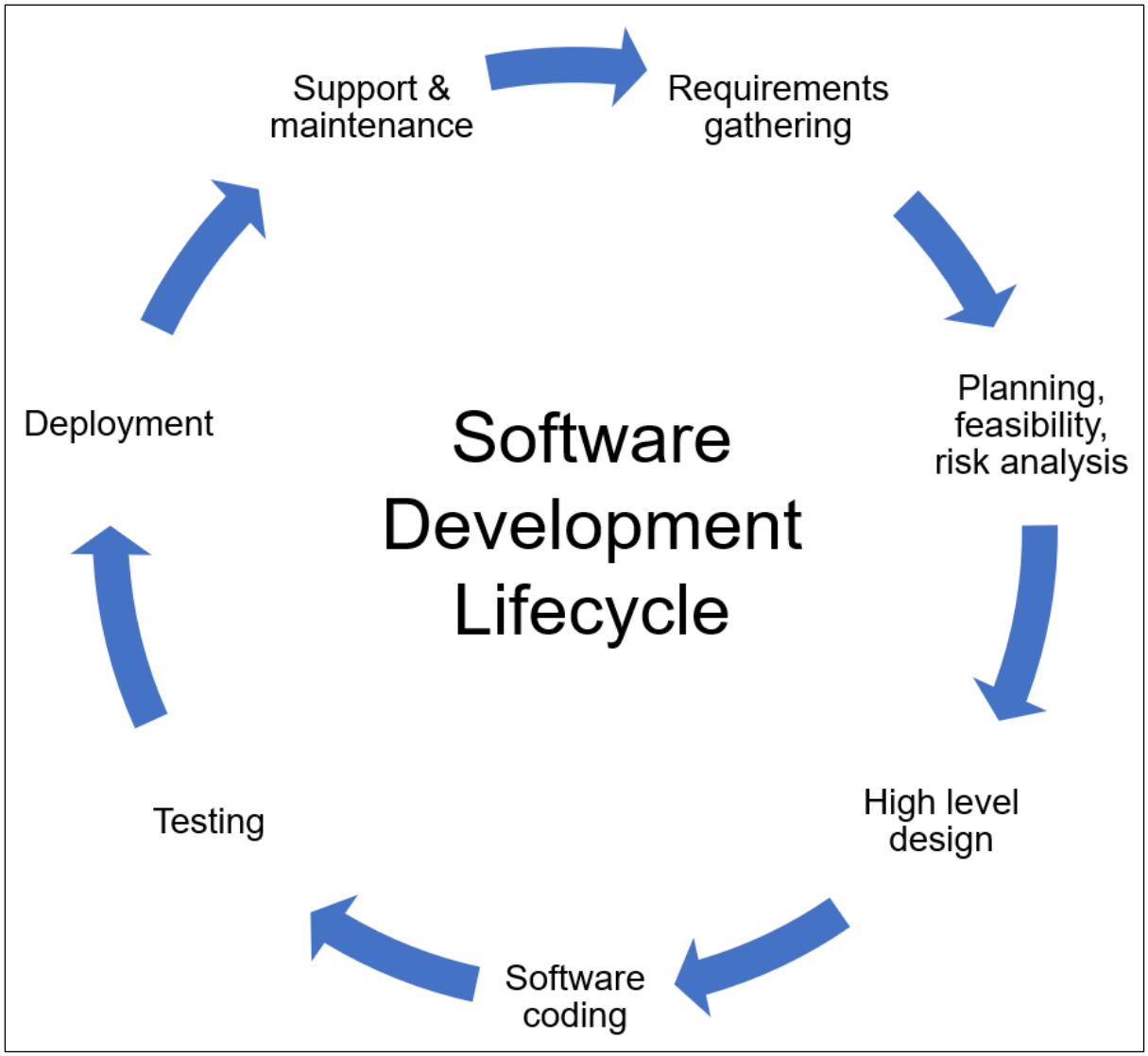
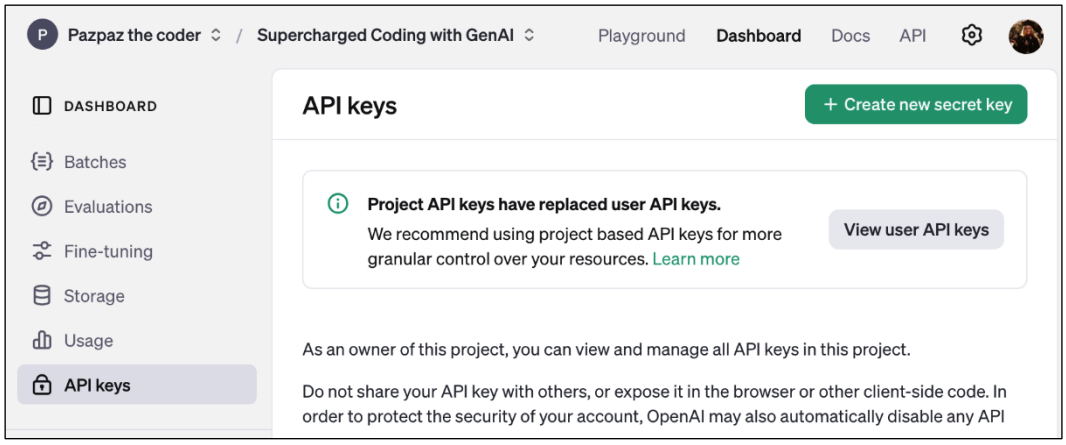
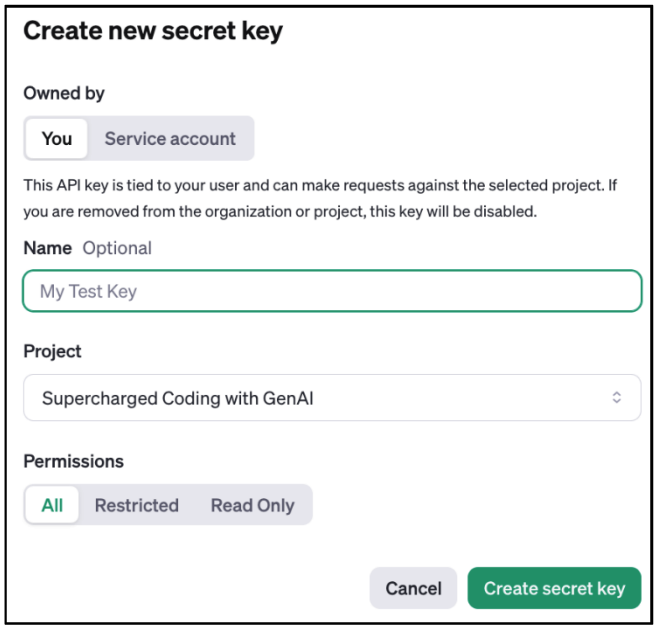
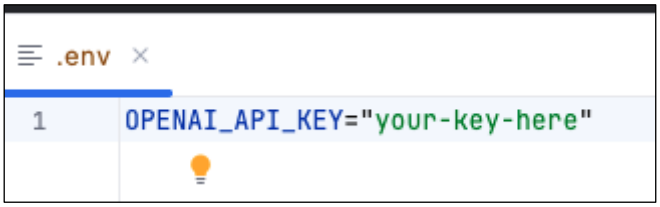
Chapter 1: From Automation to Full Software Development Life Cycle: The Current Opportunity for GenAI

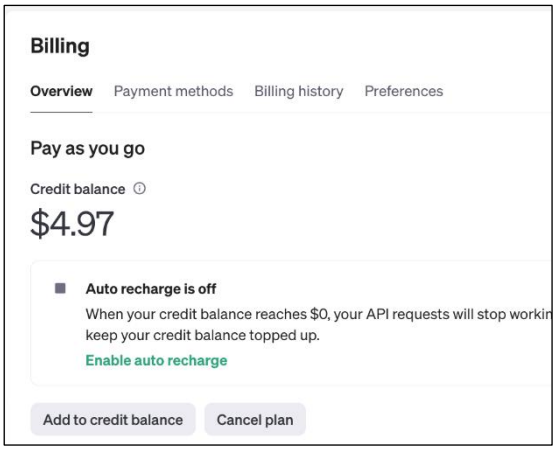


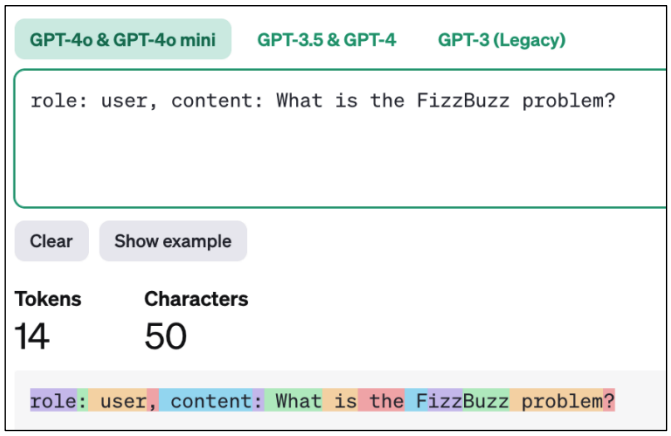
Chapter 2: Your Quick Start Guide to the OpenAI API











GPT-4o

GPT-4o is our most advanced multimodal model that's faster and cheaper than GPT-4 Turbo with stronger vision capabilities. The model has 128K context and an October 2023 knowledge cutoff.

Learn about GPT-4o ↗

Model	Pricing	Pricing with Batch API*
gpt-4o	$2.50 / 1M input tokens	$1.25 / 1M input tokens
	$1.25 / 1M cached** input tokens	
	$10.00 / 1M output tokens	$5.00 / 1M output tokens

Free tier rate limits

This is a high level summary and there are per-model exceptions to these limits (e.g. some legacy models or models with larger context windows have different rate limits). To view the exact rate limits per model for your account, visit the limits section of your account settings.

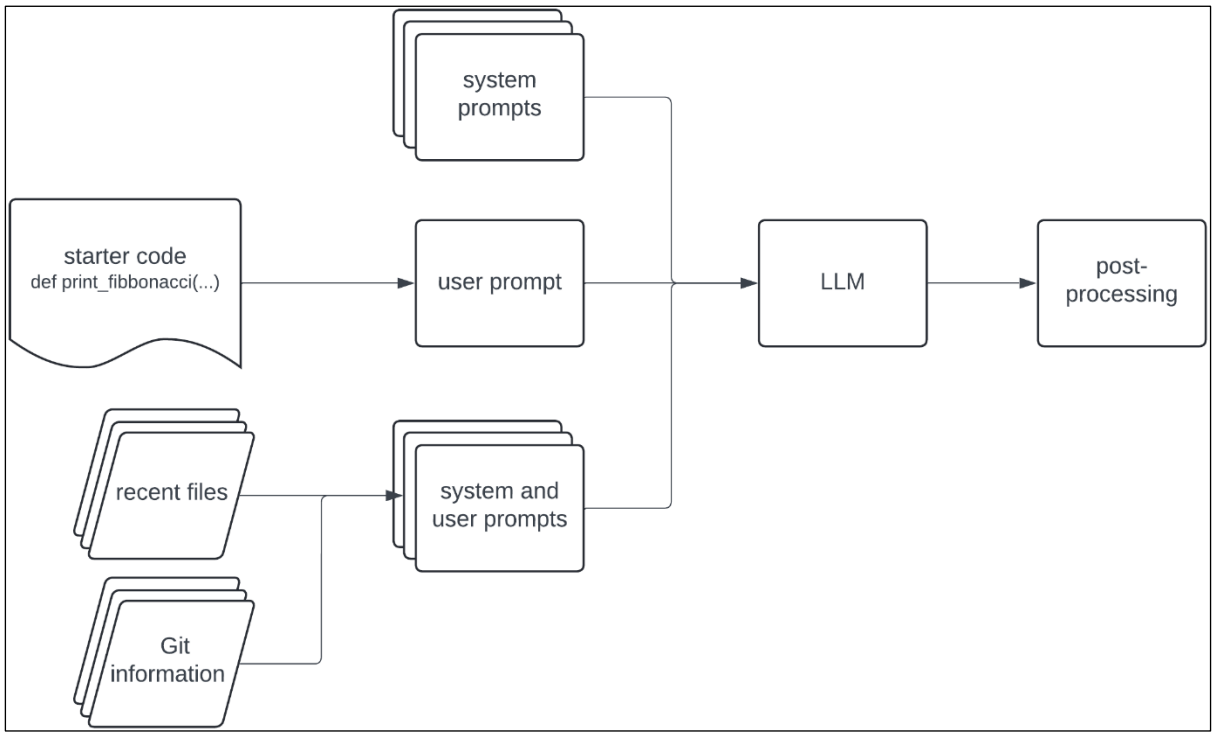
MODEL	RPM	RPD	TPM	BATCH QUEUE LIMIT
gpt-3.5-turbo	3	200	40,000	200,000
text-embedding-3-large	3,000	200	1,000,000	3,000,000
text-embedding-3-small	3,000	200	1,000,000	3,000,000

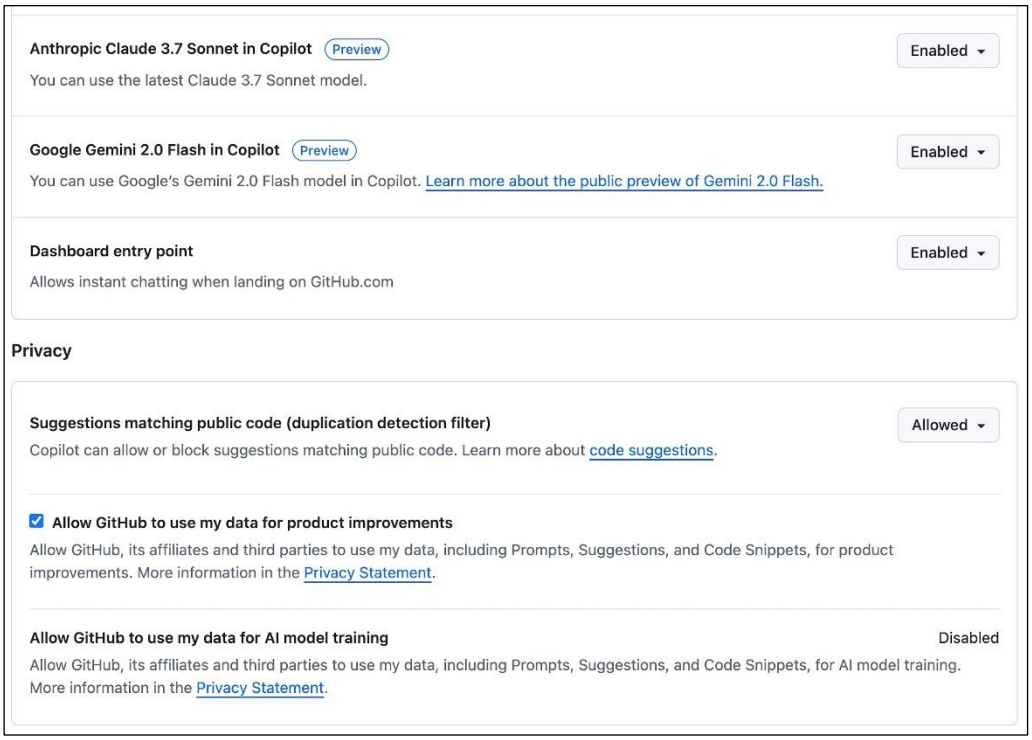
Tier 1 rate limits

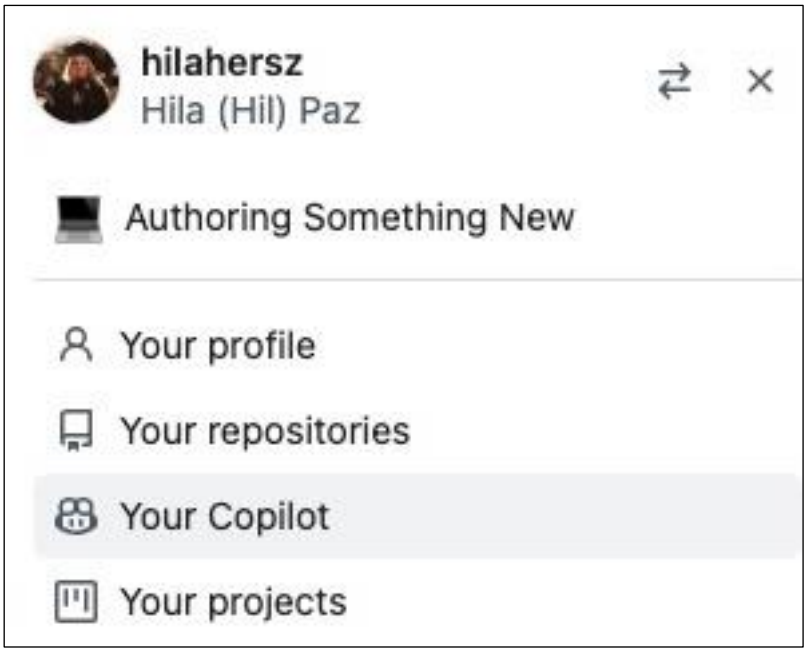
This is a high level summary and there are per-model exceptions to these limits (e.g. some legacy models or models with larger context windows have different rate limits). To view the exact rate limits per model for your account, visit the limits section of your account settings.

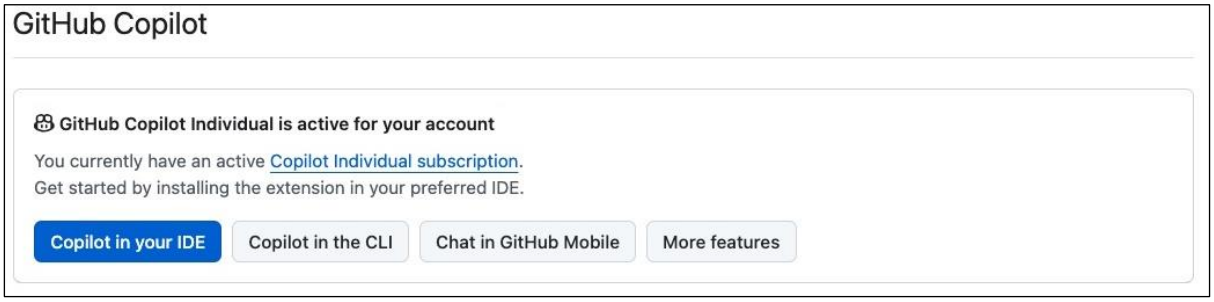
MODEL	RPM	RPD	TPM	BATCH QUEUE LIMIT
gpt-4o	500	-	30,000	90,000
gpt-4o-mini	500	10,000	200,000	2,000,000
gpt-4o-realtime-preview	100	100	20,000	-
gpt-4-turbo	500	-	30,000	90,000

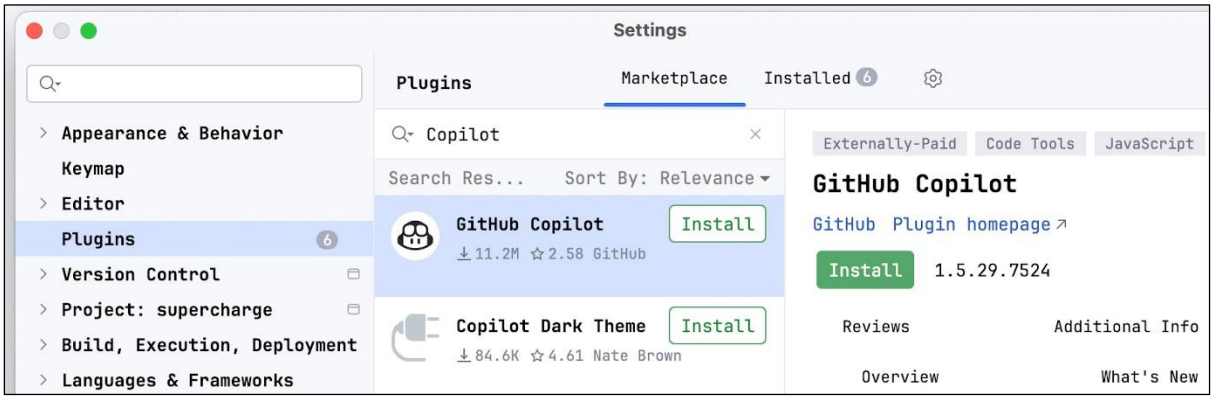
Chapter 3: A Guide to GitHub Copilot with PyCharm, VS Code, and Jupyter Notebook



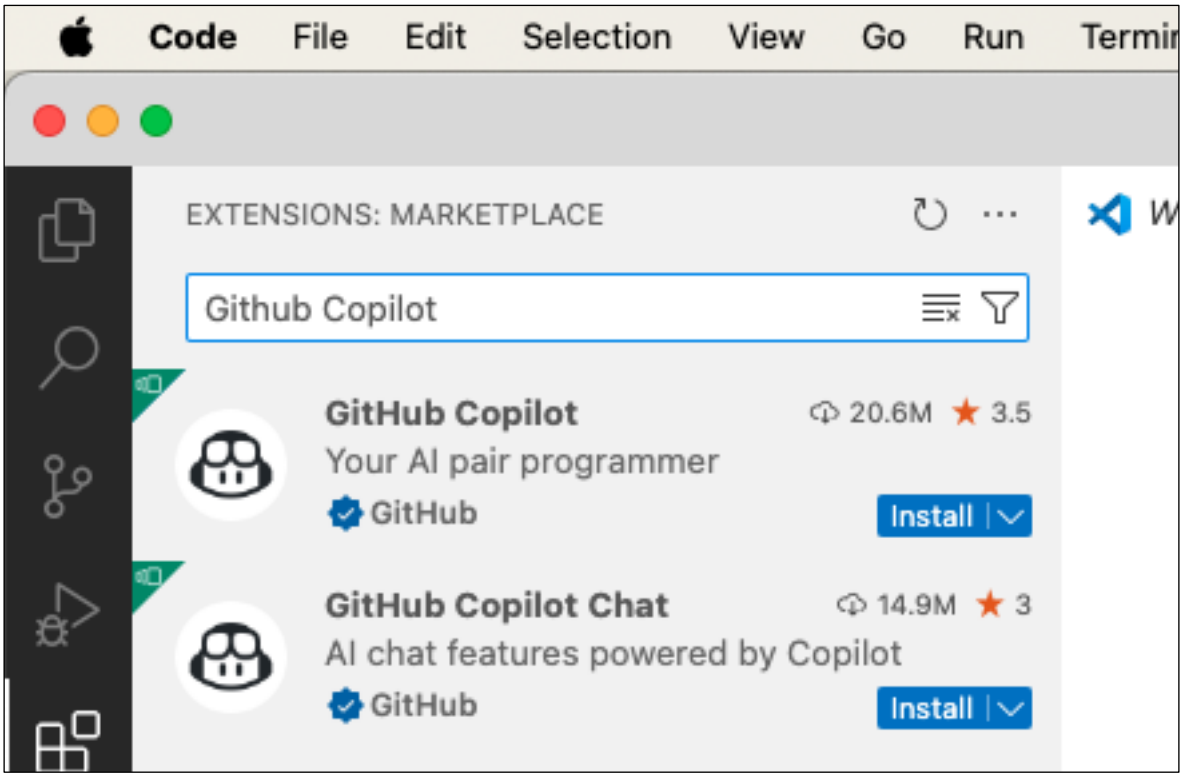


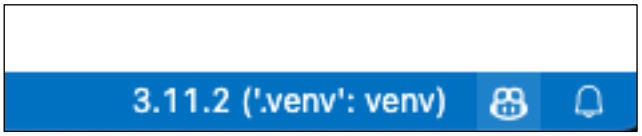

hilahersz
Hila (Hil) Paz
Authoring Something New
Your profile
Your repositories
Your Copilot
Your projects

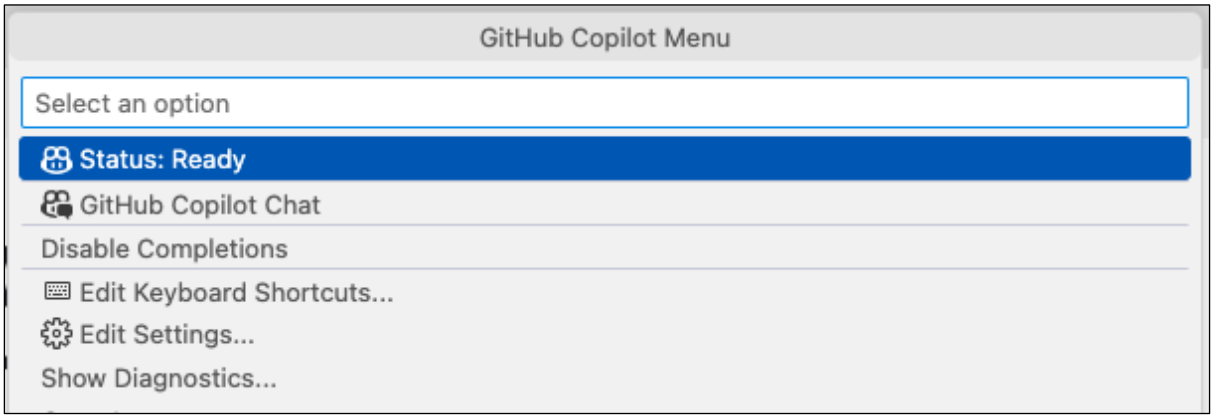

GitHub Copilot
GitHub Copilot Individual is active for your account
You currently have an active Copilot Individual subscription.
Get started by installing the extension in your preferred IDE.
Copilot in your IDE
Copilot in the CLI
Chat in GitHub Mobile
More features

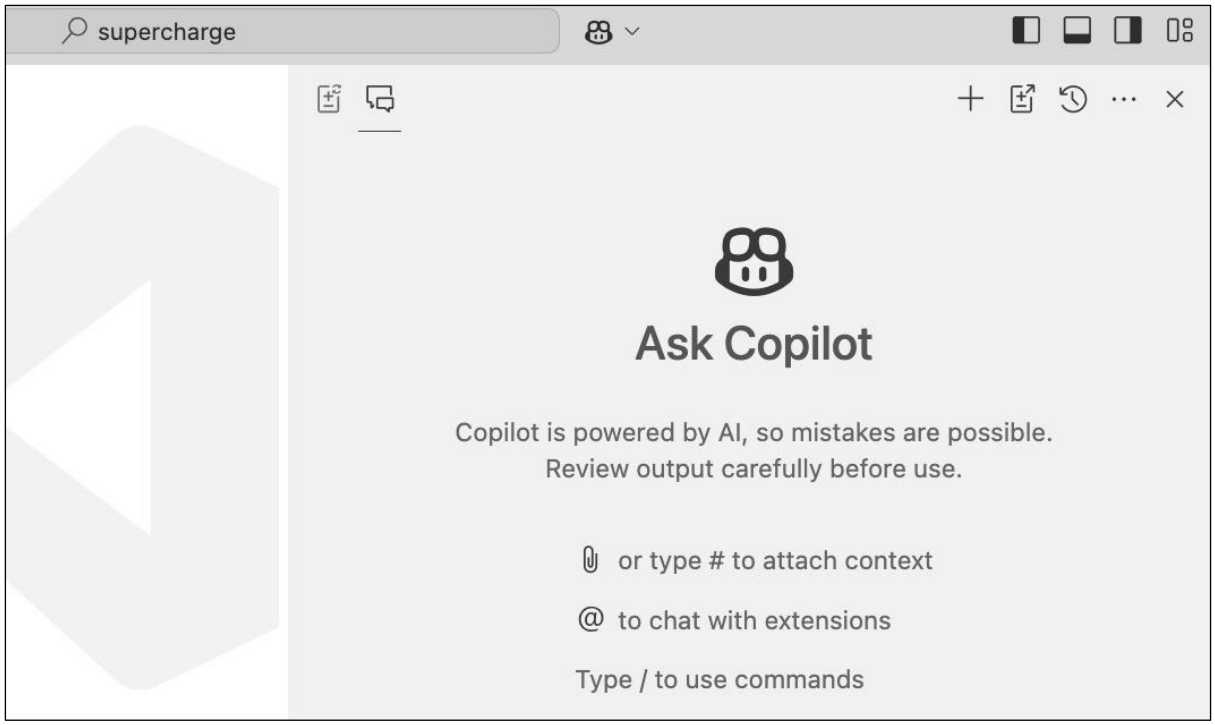

Settings
Plugins
Marketplace
Installed 6
Appearance & Behavior
Keymap
Editor
Plugins 6
Version Control
Project: supercharge
Build, Execution, Deployment
Languages & Frameworks
Copilot
Search Res...
Sort By: Relevance
GitHub Copilot
Install
11.2M 2.58 GitHub
Copilot Dark Theme
Install
84.6K 4.61 Nate Brown
Externally-Paid
Code Tools
JavaScript
GitHub Copilot
GitHub
Plugin homepage
Install
1.5.29.7524
Reviews
Additional Info
Overview
What's New

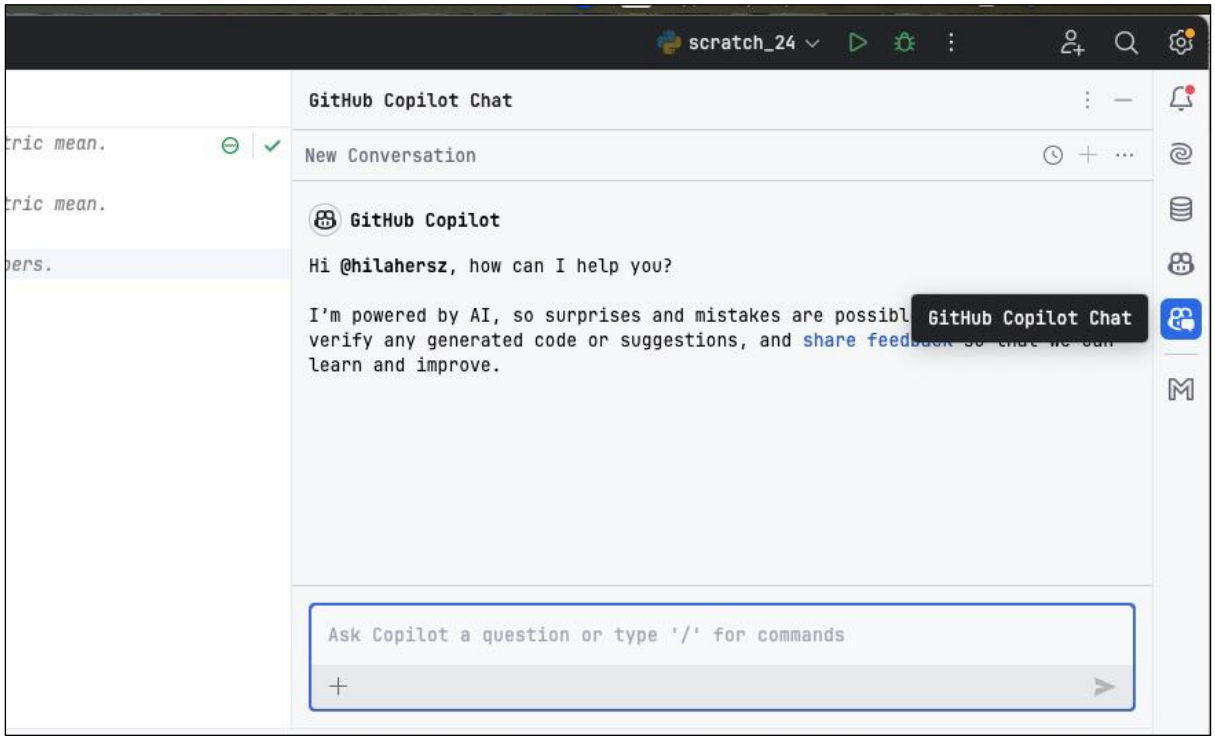

Ready
7:19 (42 chars, 1 line break)
LF
UTF-8
Python 3.11 (su

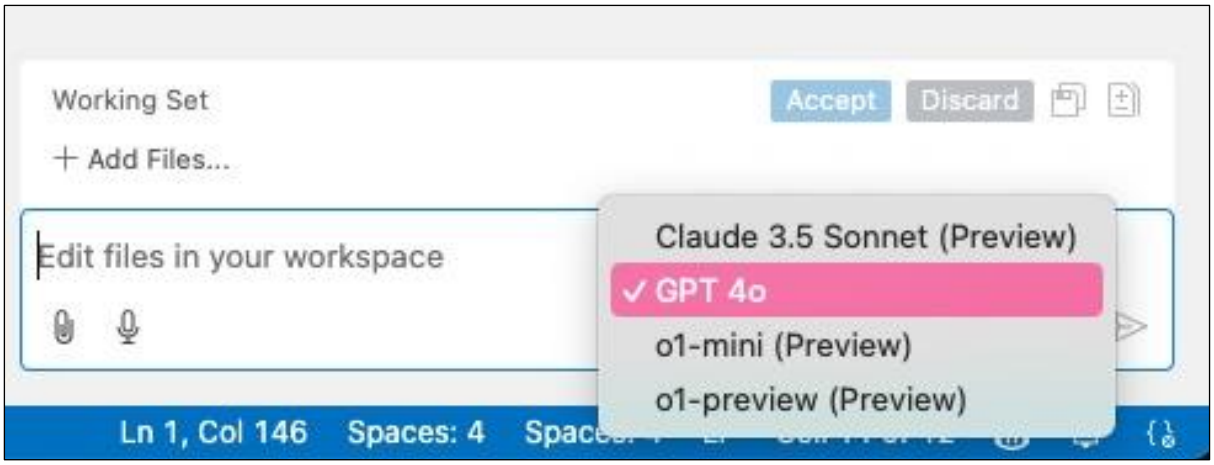

Code File Edit Selection View Go Run Termin
EXTENSIONS: MARKETPLACE
Github Copilot
GitHub Copilot
20.6M 3.5
Your AI pair programmer
GitHub
Install
GitHub Copilot Chat
14.9M 3
AI chat features powered by Copilot
GitHub
Install

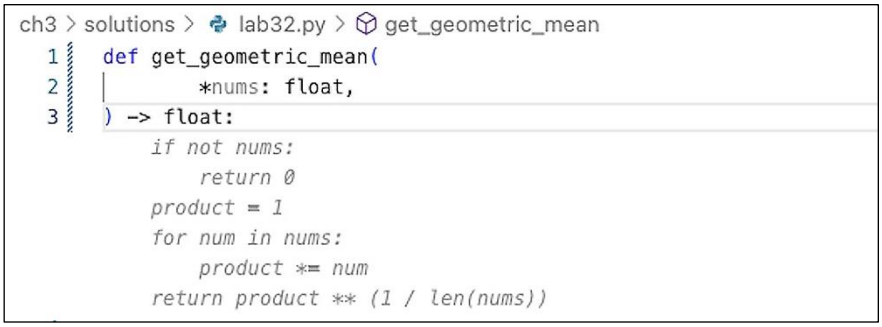

3.11.2 ('.venv': venv)

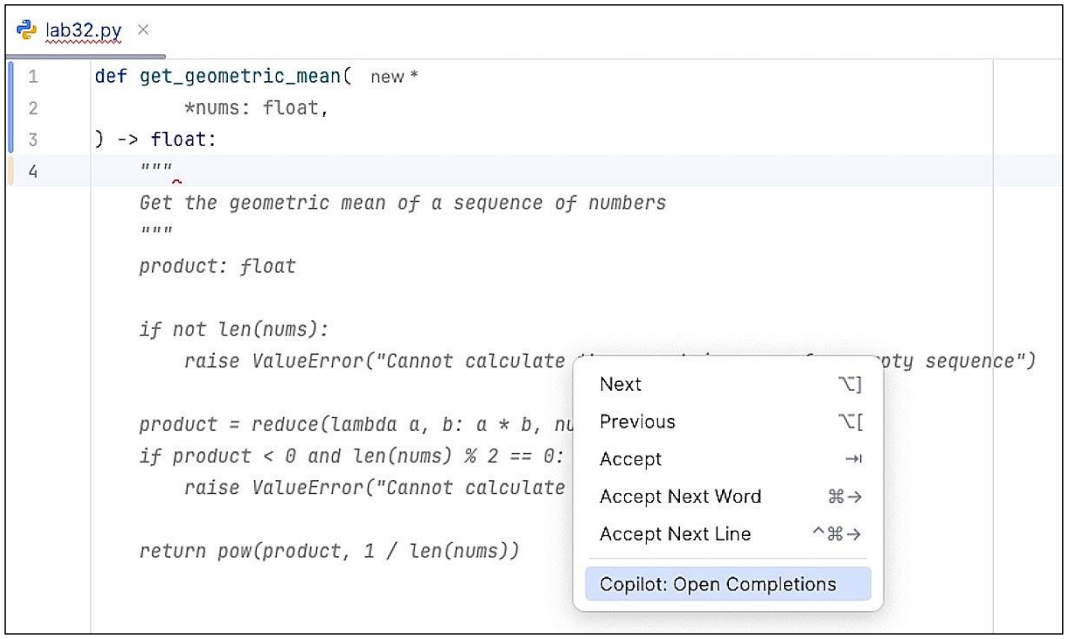

GitHub Copilot Menu
Select an option
Status: Ready
GitHub Copilot Chat
Disable Completions
Edit Keyboard Shortcuts...
Edit Settings...
Show Diagnostics...

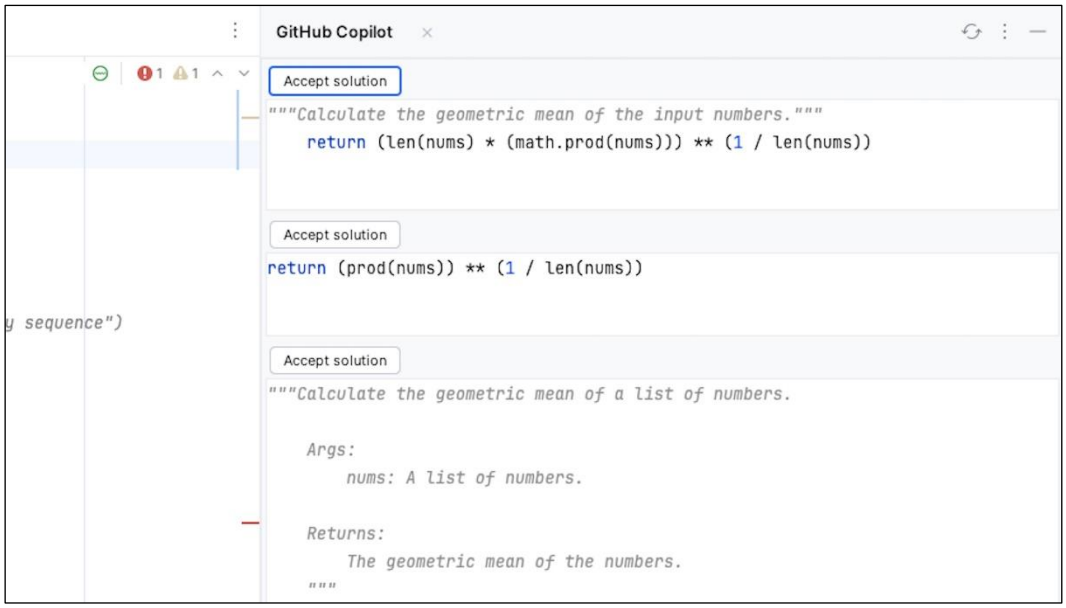

supercharge
Ask Copilot
Copilot is powered by AI, so mistakes are possible.
Review output carefully before use.
or type # to attach context
to chat with extensions
Type / to use commands

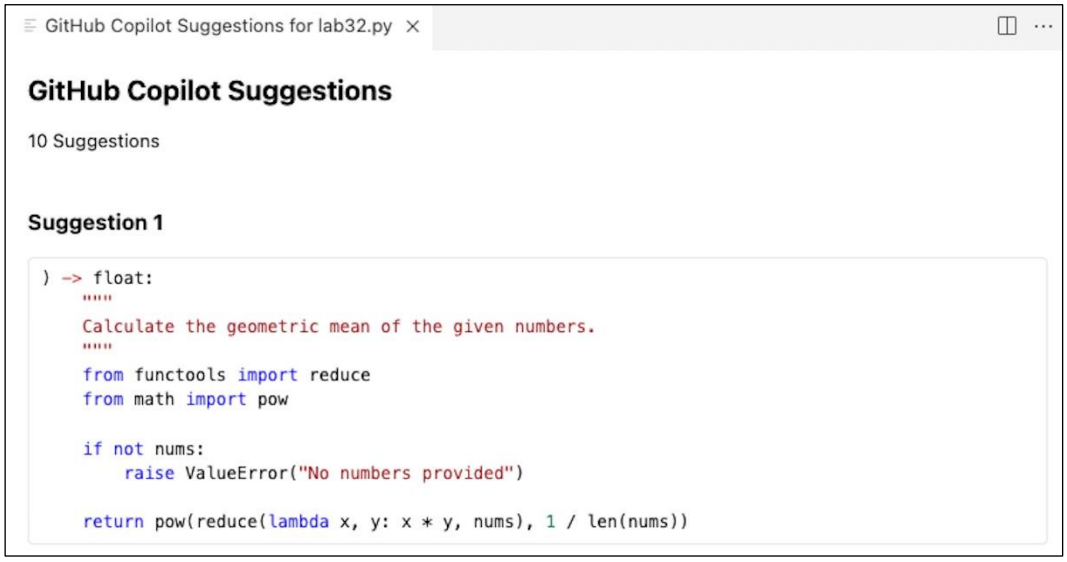

scratch_24
GitHub Copilot Chat
New Conversation
GitHub Copilot
Hi @hilahersz, how can I help you?
I'm powered by AI, so surprises and mistakes are possibl
verify any generated code or suggestions, and share feedback
learn and improve.
GitHub Copilot Chat
Ask Copilot a question or type '/' for commands

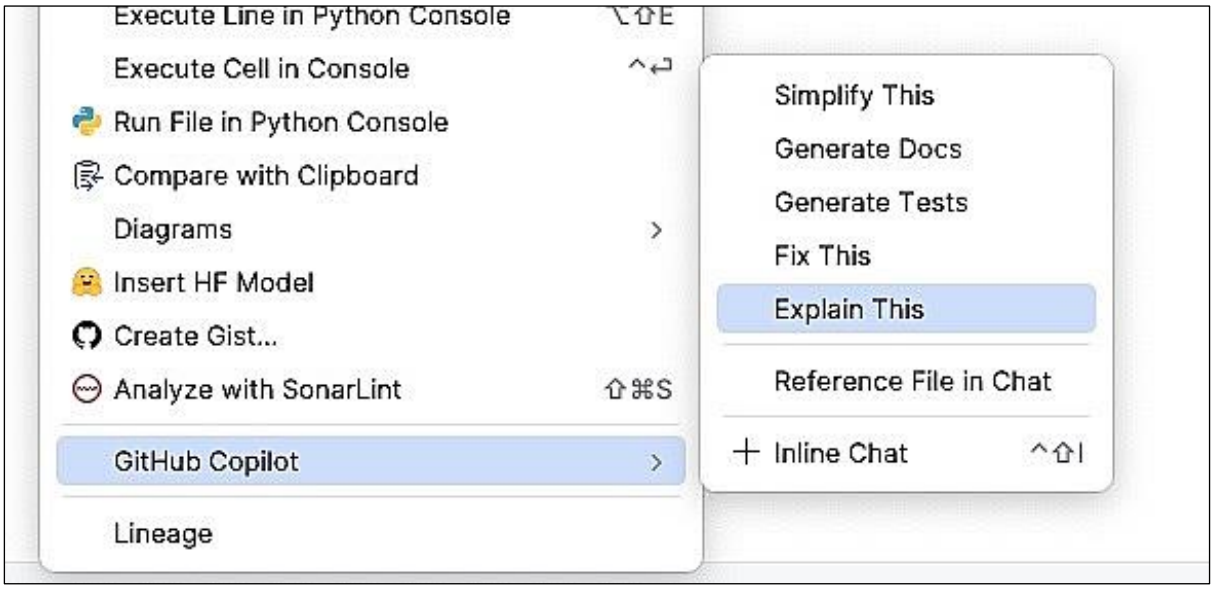

Working Set
Accept
Discard
Add Files...
Edit files in your workspace
Claude 3.5 Sonnet (Preview)
GPT 4o
o1-mini (Preview)
o1-preview (Preview)
Ln 1, Col 146
Spaces: 4

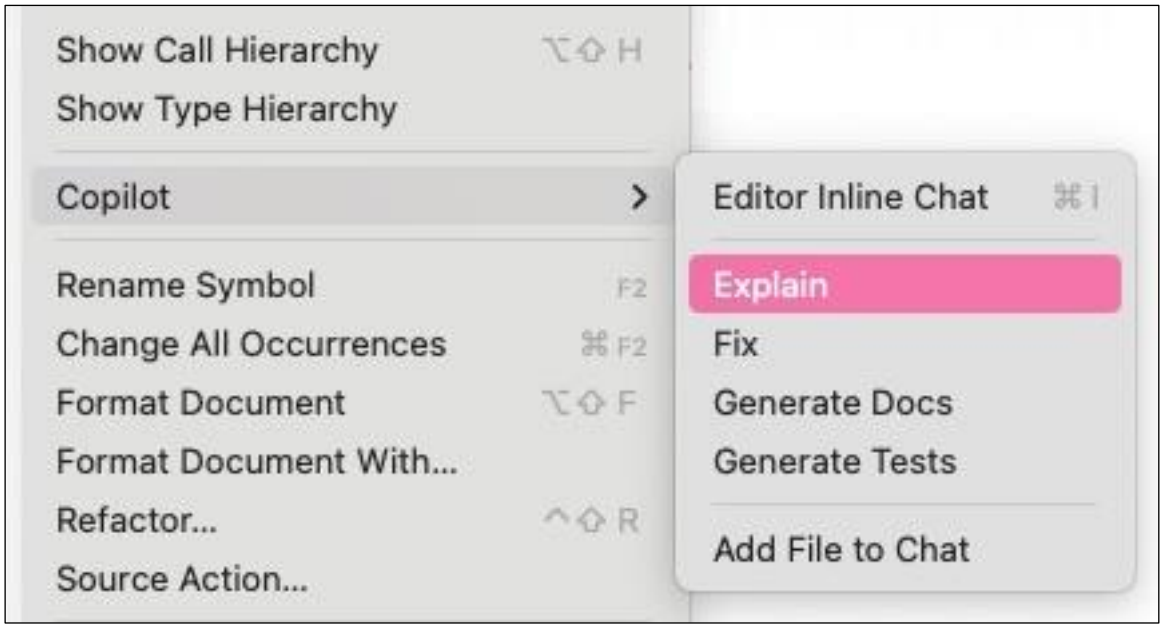

ch3 > solutions > lab32.py > get_geometric_mean
def get_geometric_mean(
*nums: float,
) -> float:
if not nums:
return 0
product = 1
for num in nums:
product *= num
return product ** (1 / len(nums))

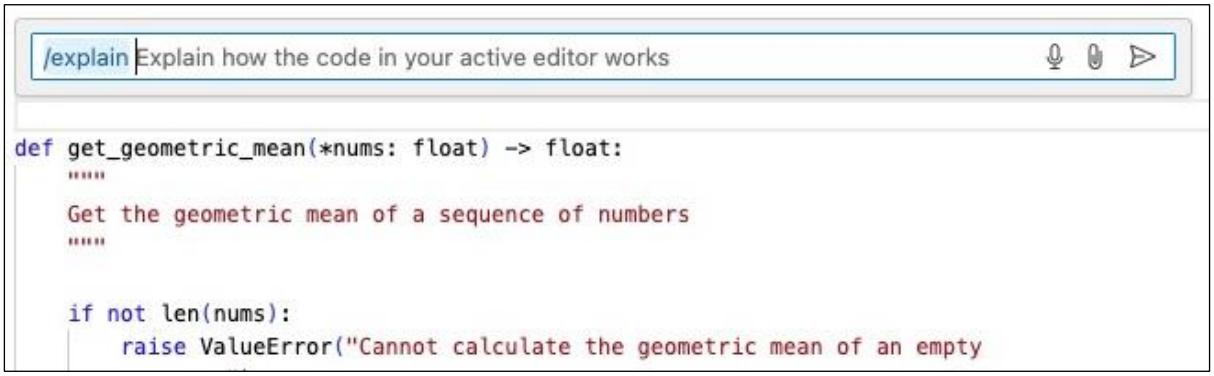

lab32.py
def get_geometric_mean(new *
*nums: float,
) -> float:
"""
Get the geometric mean of a sequence of numbers
"""
product: float
if not len(nums):
raise ValueError("Cannot calculate
product = reduce(lambda a, b: a * b, nu
if product < 0 and len(nums) % 2 == 0:
raise ValueError("Cannot calculate
return pow(product, 1 / len(nums))
Next ⌥]
Previous ⌥[
Accept ⇥
Accept Next Word ⌘→
Accept Next Line ^⌘→
Copilot: Open Completions

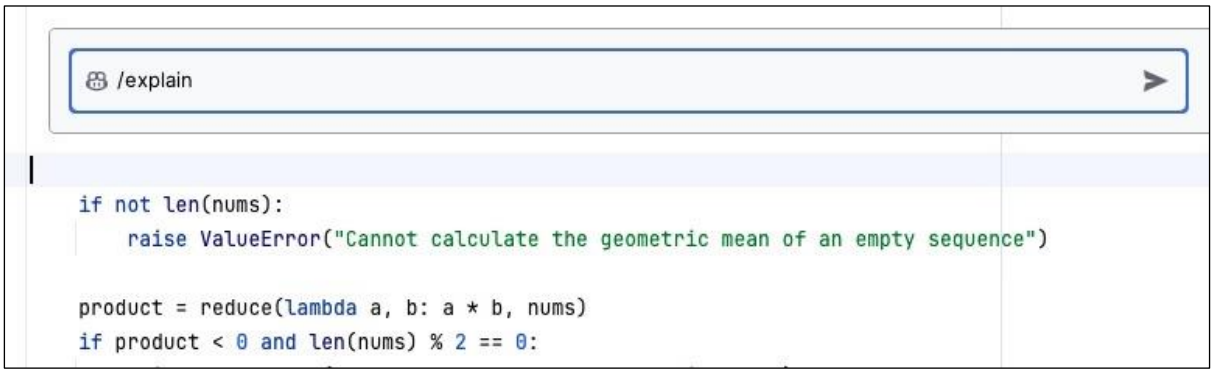

GitHub Copilot
Accept solution
"""Calculate the geometric mean of the input numbers."""
return (len(nums) * (math.prod(nums))) ** (1 / len(nums))
Accept solution
return (prod(nums)) ** (1 / len(nums))
y sequence")
Accept solution
"""Calculate the geometric mean of a list of numbers.
Args:
nums: A list of numbers.
Returns:
The geometric mean of the numbers.
"""

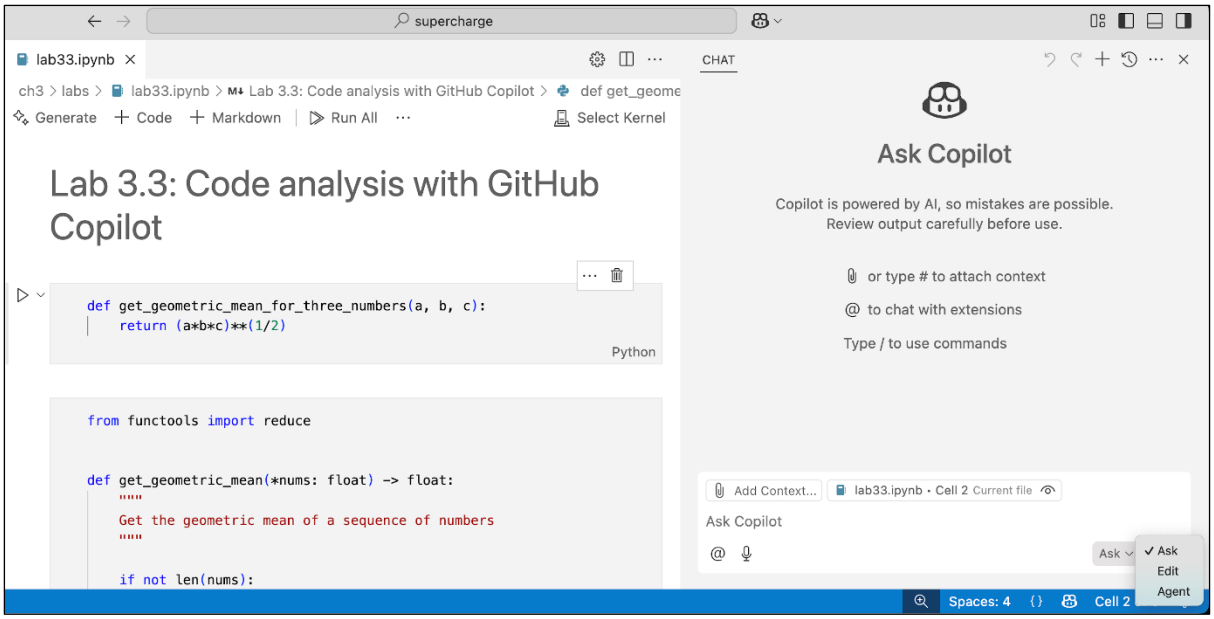

GitHub Copilot Suggestions for lab32.py
GitHub Copilot Suggestions
10 Suggestions
Suggestion 1
) -> float:
"""
Calculate the geometric mean of the given numbers.
"""
from functools import reduce
from math import pow
if not nums:
raise ValueError("No numbers provided")
return pow(reduce(lambda x, y: x * y, nums), 1 / len(nums))

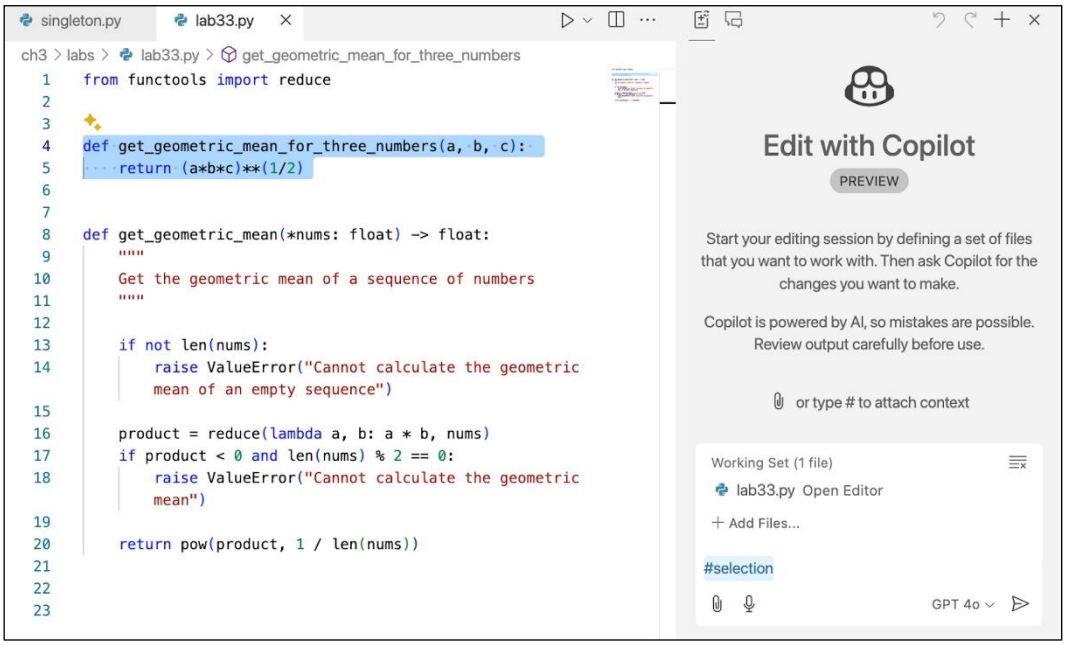

Execute Line in Python Console
Execute Cell in Console
Run File in Python Console
Compare with Clipboard
Diagrams
Insert HF Model
Create Gist...
Analyze with SonarLint
GitHub Copilot
Lineage
Simplify This
Generate Docs
Generate Tests
Fix This
Explain This
Reference File in Chat
Inline Chat

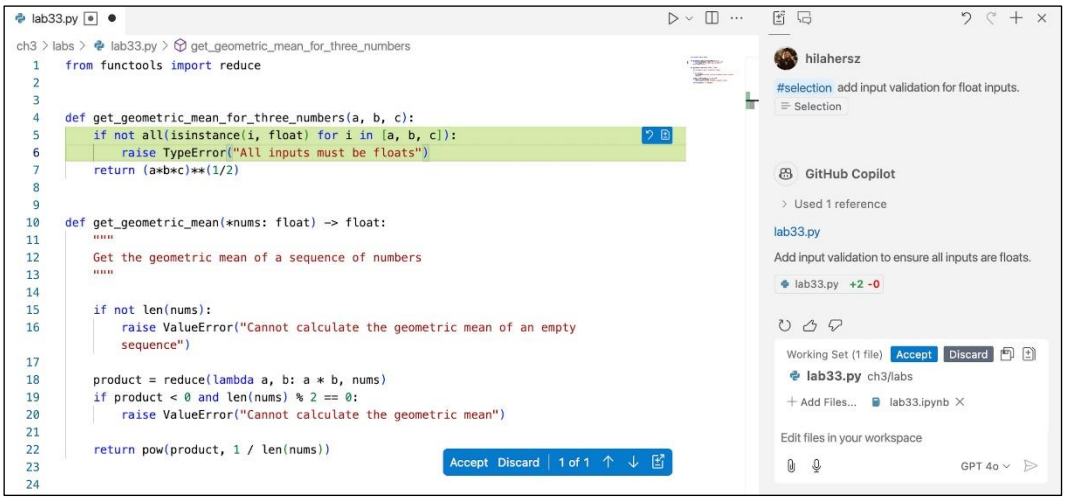

Show Call Hierarchy ⌥⇧H
Show Type Hierarchy
Copilot
Rename Symbol F2
Change All Occurrences ⌘F2
Format Document ⌥⇧F
Format Document With...
Refactor... ^⇧R
Source Action...
Editor Inline Chat ⌘I
Explain
Fix
Generate Docs
Generate Tests
Add File to Chat

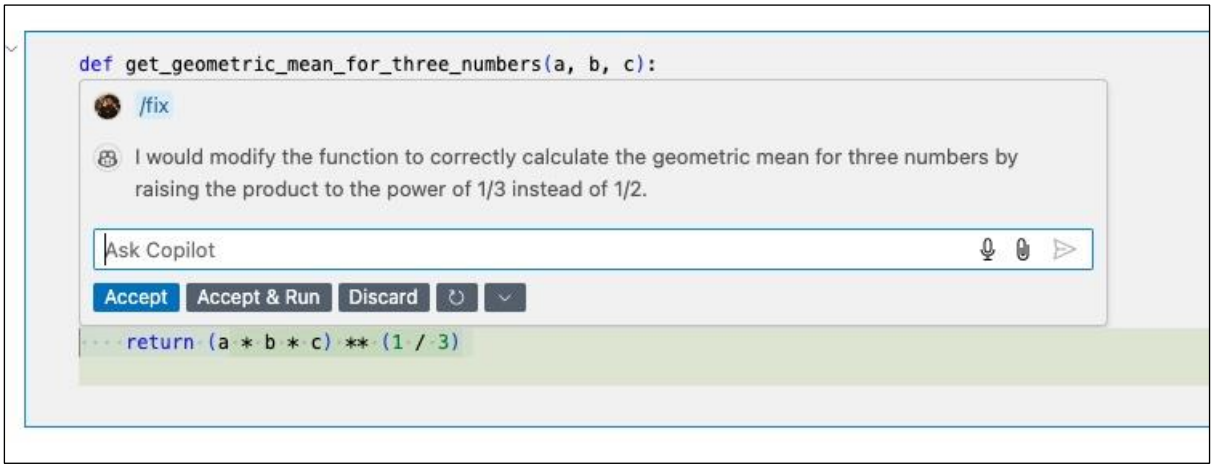

/explain Explain how the code in your active editor works
def get_geometric_mean(*nums: float) -> float:
"""
Get the geometric mean of a sequence of numbers
"""
if not len(nums):
raise ValueError("Cannot calculate the geometric mean of an empty

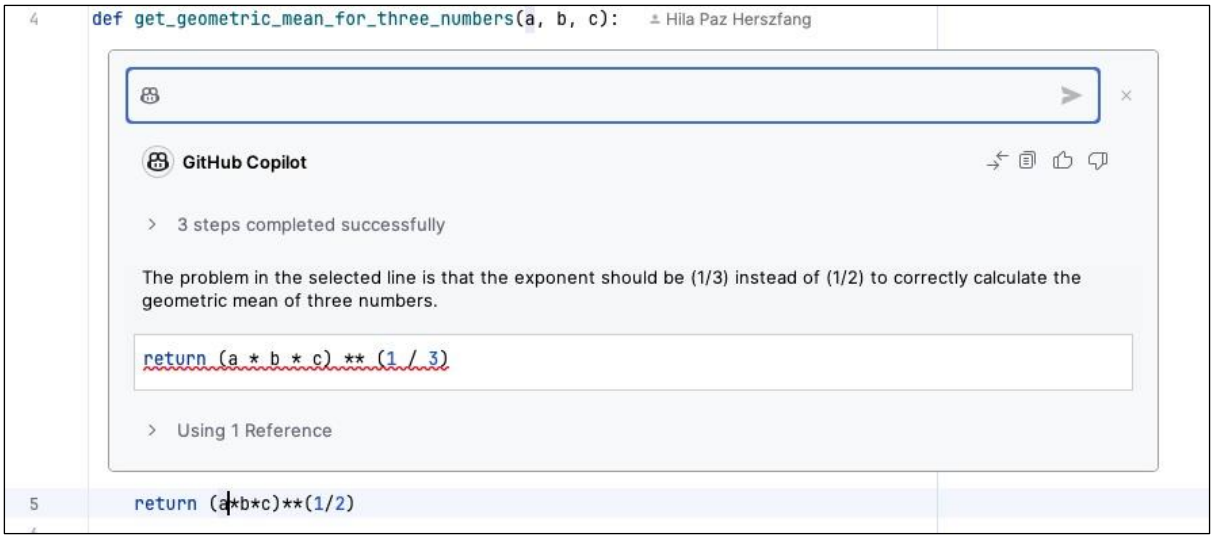

/explain
if not len(nums):
raise ValueError("Cannot calculate the geometric mean of an empty sequence")
product = reduce(lambda a, b: a * b, nums)
if product < 0 and len(nums) % 2 == 0:

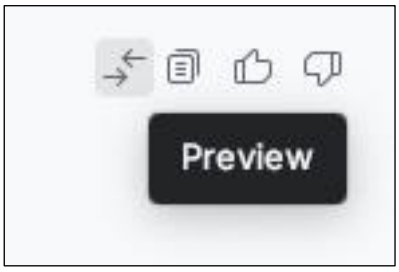

supercharge
lab33.ipynb
ch3 > labs > lab33.ipynb > Lab 3.3: Code analysis with GitHub Copilot > def get_geome
Generate
Code
Markdown
Run All
Select Kernel
Lab 3.3: Code analysis with GitHub Copilot
def get_geometric_mean_for_three_numbers(a, b, c):
return (a*b*c)**(1/2)
Python
from functools import reduce
def get_geometric_mean(*nums: float) -> float:
"""
Get the geometric mean of a sequence of numbers
"""
if not len(nums):
CHAT
Ask Copilot
Copilot is powered by AI, so mistakes are possible.
Review output carefully before use.
or type # to attach context
@ to chat with extensions
Type / to use commands
Add Context...
lab33.ipynb • Cell 2 Current file
Ask Copilot
Ask
Edit
Agent
Spaces: 4
Cell 2

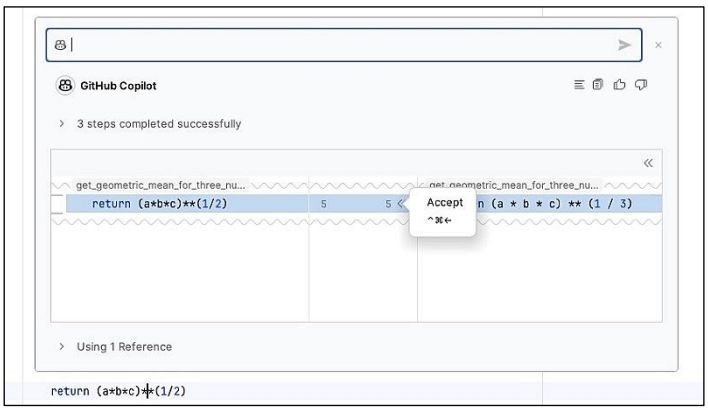

singleton.py
lab33.py
ch3 > labs > lab33.py > get_geometric_mean_for_three_numbers
from functools import reduce
def get_geometric_mean_for_three_numbers(a, b, c):
return (a*b*c)**(1/2)
def get_geometric_mean(*nums: float) -> float:
"""
Get the geometric mean of a sequence of numbers
"""
if not len(nums):
raise ValueError("Cannot calculate the geometric mean of an empty sequence")
product = reduce(lambda a, b: a * b, nums)
if product < 0 and len(nums) % 2 == 0:
raise ValueError("Cannot calculate the geometric mean")
return pow(product, 1 / len(nums))
Edit with Copilot
PREVIEW
Start your editing session by defining a set of files that you want to work with. Then ask Copilot for the changes you want to make.
Copilot is powered by AI, so mistakes are possible.
Review output carefully before use.
or type # to attach context
Working Set (1 file)
lab33.py Open Editor
Add Files...
#selection
GPT 4o


lab33.py
ch3 > labs > lab33.py > get_geometric_mean_for_three_numbers
from functools import reduce
def get_geometric_mean_for_three_numbers(a, b, c):
if not all(isinstance(i, float) for i in [a, b, c]):
raise TypeError("All inputs must be floats")
return (a*b*c)**(1/2)
def get_geometric_mean(*nums: float) -> float:
"""
Get the geometric mean of a sequence of numbers
"""
if not len(nums):
raise ValueError("Cannot calculate the geometric mean of an empty sequence")
product = reduce(lambda a, b: a * b, nums)
if product < 0 and len(nums) % 2 == 0:
raise ValueError("Cannot calculate the geometric mean")
return pow(product, 1 / len(nums))
Accept
Discard
1 of 1
hilahersz
#selection add input validation for float inputs.
Selection
GitHub Copilot
Used 1 reference
lab33.py
Add input validation to ensure all inputs are floats.
lab33.py +2 -0
Working Set (1 file)
Accept
Discard
lab33.py ch3/labs
Add Files...
lab33.ipynb
Edit files in your workspace
GPT 4o

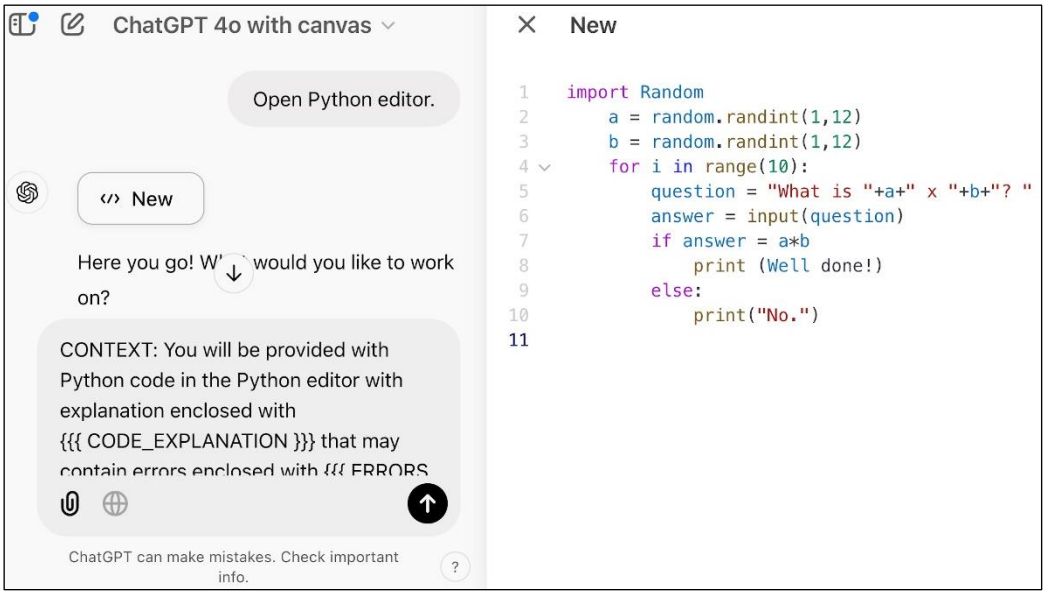

def get_geometric_mean_for_three_numbers(a, b, c):
/fix
I would modify the function to correctly calculate the geometric mean for three numbers by raising the product to the power of 1/3 instead of 1/2.
Ask Copilot
Accept
Accept & Run
Discard
return (a * b * c) ** (1 / 3)

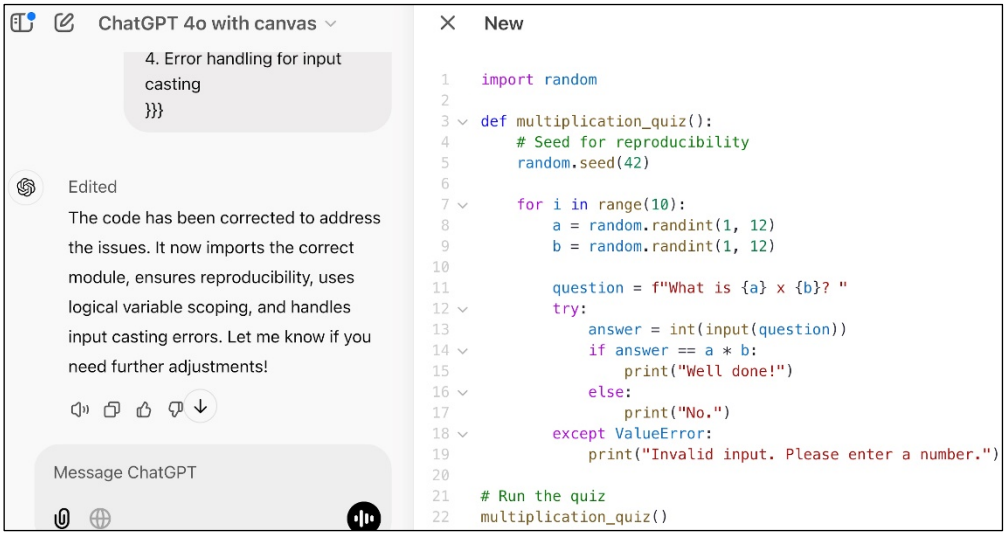

4
def get_geometric_mean_for_three_numbers(a, b, c):
Hila Paz Herszfang
GitHub Copilot
3 steps completed successfully
The problem in the selected line is that the exponent should be (1/3) instead of (1/2) to correctly calculate the geometric mean of three numbers.
return (a * b * c) ** (1 / 3)
Using 1 Reference
5
return (a*b*c)**(1/2)

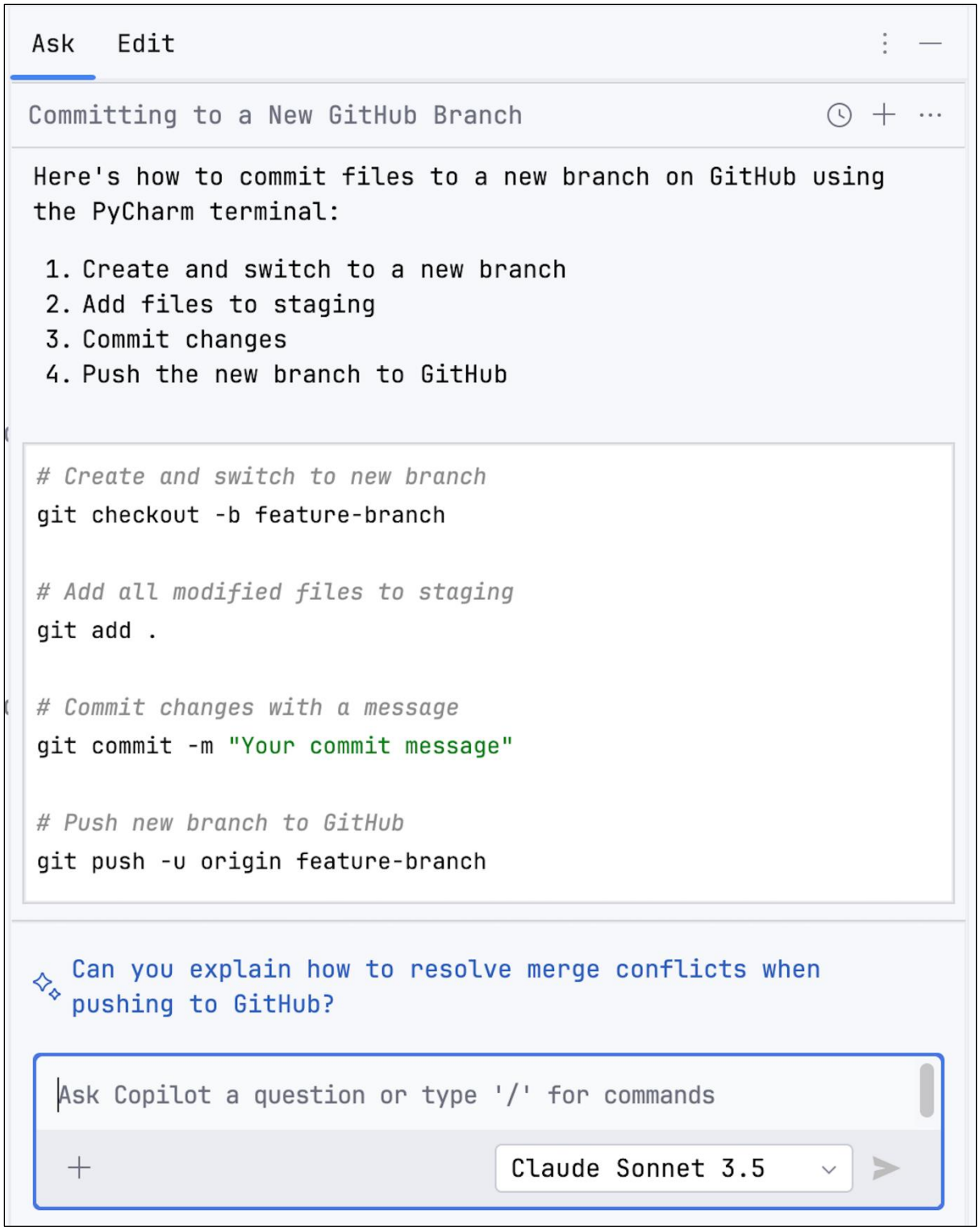

Preview

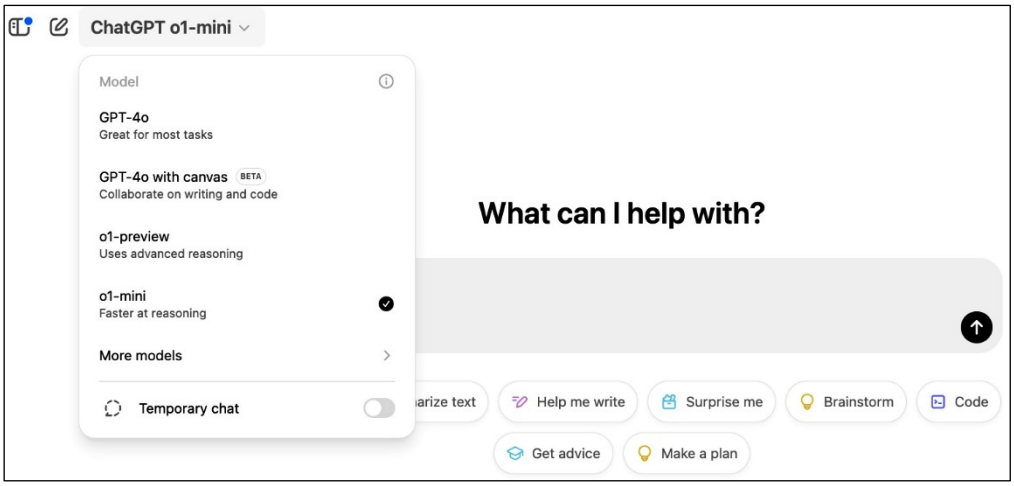

GitHub Copilot
3 steps completed successfully
get_geometric_mean_for_three_nu...
return (a*b*c)**(1/2)
Accept
Using 1 Reference
return (a*b*c)**(1/2)

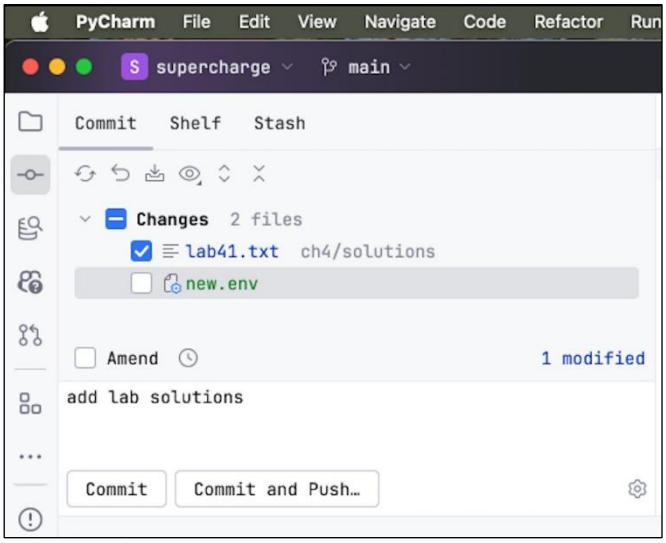
Chapter 4: Best Practices for Prompting with ChatGPT

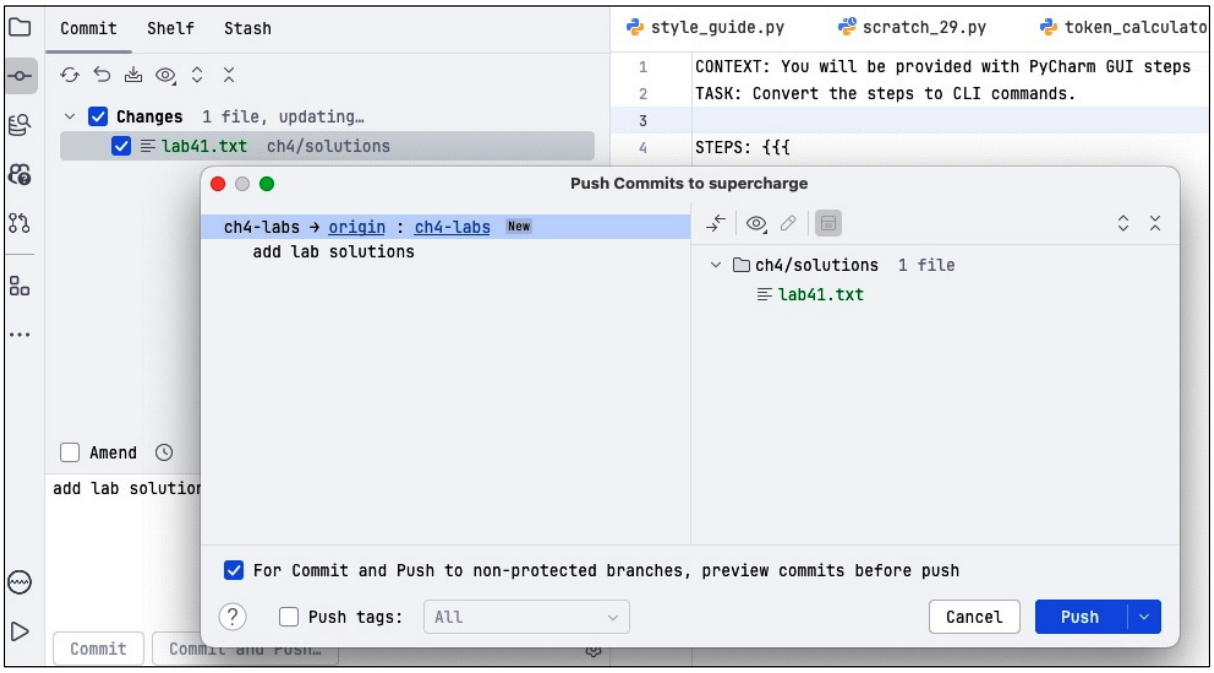





Ask
Edit
Committing to a New GitHub Branch
Here's how to commit files to a new branch on GitHub using the PyCharm terminal:
1. Create and switch to a new branch
2. Add files to staging
3. Commit changes
4. Push the new branch to GitHub
Create and switch to new branch
git checkout -b feature-branch
Add all modified files to staging
git add .
Commit changes with a message
git commit -m "Your commit message"
Push new branch to GitHub
git push -u origin feature-branch
Can you explain how to resolve merge conflicts when pushing to GitHub?
Ask Copilot a question or type '/' for commands
Claude Sonnet 3.5


ChatGPT o1-mini
Model
GPT-4o
Great for most tasks
GPT-4o with canvas
BETA
Collaborate on writing and code
o1-preview
Uses advanced reasoning
o1-mini
Faster at reasoning
More models
Temporary chat
What can I help with?
Help me write
Surprise me
Brainstorm
Code
Get advice
Make a plan


PyCharm
File
Edit
View
Navigate
Code
Refactor
Run
supercharge
main
Commit
Shelf
Stash
Changes 2 files
lab41.txt ch4/solutions
new.env
Amend
1 modified
add lab solutions
Commit
Commit and Push…


Commit
Shelf
Stash
Changes 1 file, updating…
lab41.txt ch4/solutions
style_guide.py
scratch_29.py
token_calculato
1 CONTEXT: You will be provided with PyCharm GUI steps
2 TASK: Convert the steps to CLI commands.
3
4 STEPS: {{{
Push Commits to supercharge
ch4-labs → origin : ch4-labs New
add lab solutions
ch4/solutions 1 file
lab41.txt
Amend
add lab solution
For Commit and Push to non-protected branches, preview commits before push
Push tags:
All
Cancel
Push
Commit

6. Push the changes to the remote branch
}}}

CLI COMMANDS:

bash Copy code

```bash
# Step 1: Review changed files
git status

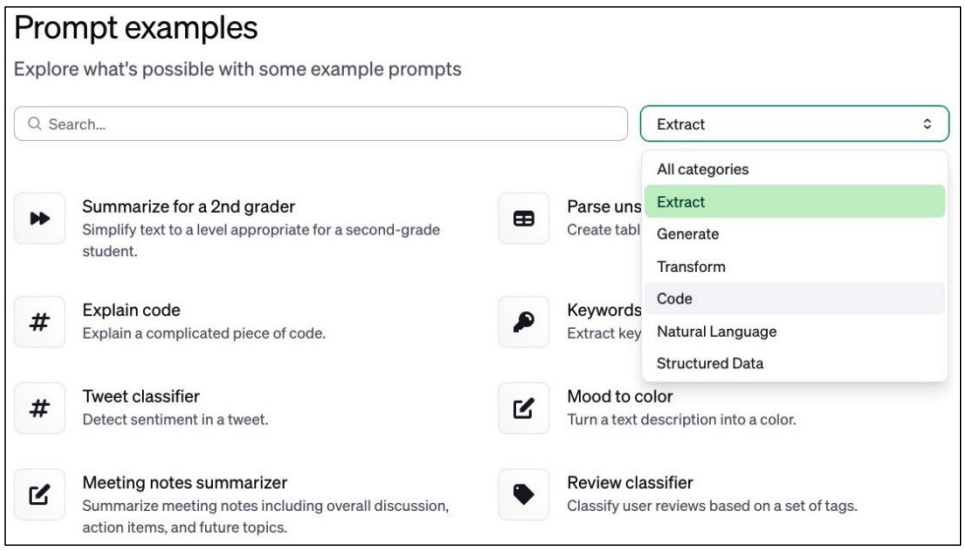
# Step 2: Stage the desired files
git add <file_name>  # Replace <file_name> with the specific file(s) you want to s
# OR to stage all changes
git add .

# Step 3: Add a commit message
git commit -m "Your commit message here"

# Step 4: Commit the files (already combined with Step 3)

# Step 5: Validate branch name
git branch  # This shows the current branch

# Step 6: Push the changes to the remote branch
git push origin <branch_name>  # Replace <branch_name> with your current branch na
```



Improve code efficiency

Transform | Code

Provide ideas for efficiency improvements to Python code.

Prompt

SYSTEM You will be provided with a piece of Python code, and your task is to provide ideas for efficiency improvements.

USER

```
from typing import List


def has_sum_k(nums: List[int], k: int) -> bool:
    """
    Returns True if there are two distinct elements in nums such that their sum
    is equal to k, and otherwise returns False.
    """
    n = len(nums)
    for i in range(n):
        for j in range(i+1, n):
            if nums[i] + nums[j] == k:
                return True
    return False
```

Sample response

Python bug fixer

Transform | Code

Find and fix bugs in source code.

Prompt

SYSTEM You will be provided with a piece of Python code, and your task is to find and fix bugs in it.

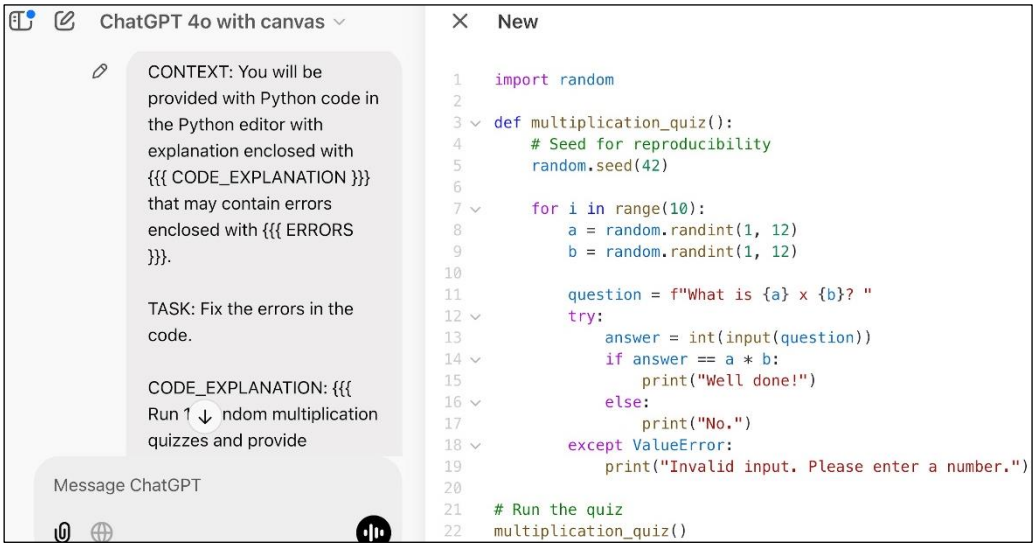
USER

```
import Random
  a = random.randint(1,12)
  b = random.randint(1,12)
  for i in range(10):
    question = "What is "+a+" x "+b+"? "
    answer = input(question)
    if answer = a*b
      print (Well done!)
    else:
      print("No.")
```

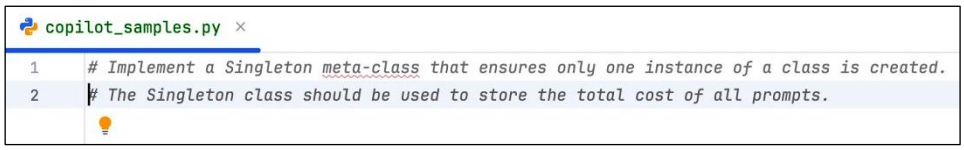
Sample response

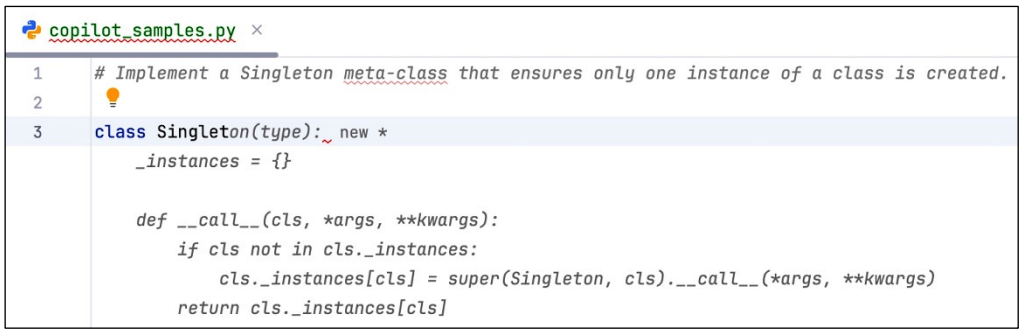
There are several bugs in the provided code:

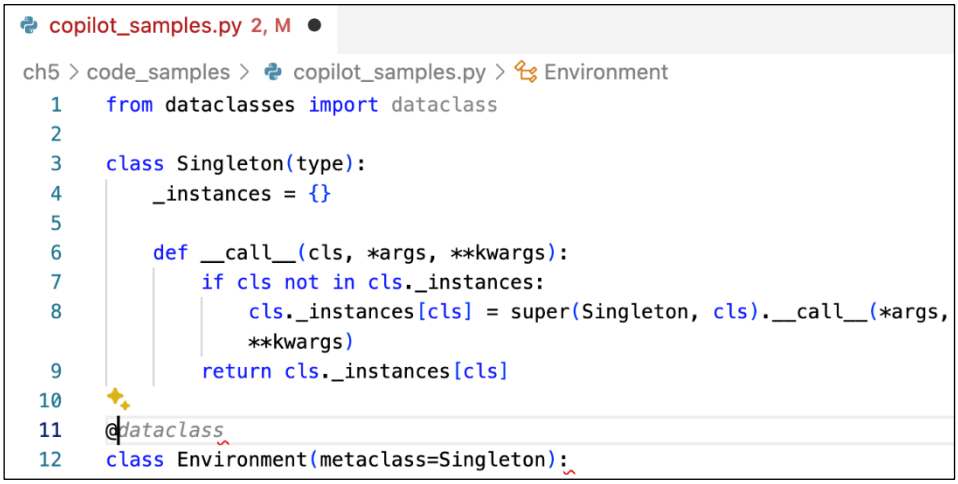
1. The import statement is incorrect. Python is case-sensitive, so "Random" should be "random".
2. Variables `a` and `b` are defined outside the loop, so they will not change for each iteration. They should be inside the loop if you want to generate different questions.



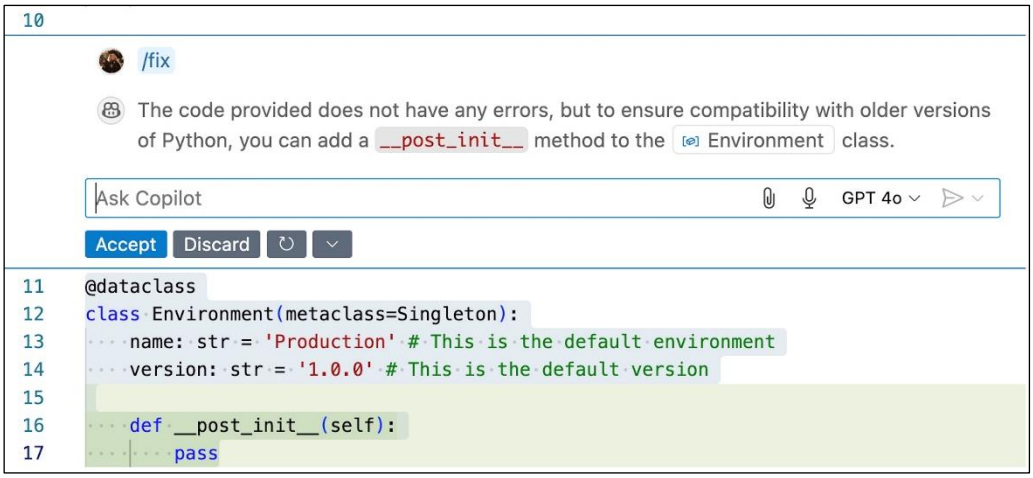
Chapter 5: Best Practices for Prompting with OpenAI API and GitHub Copilot

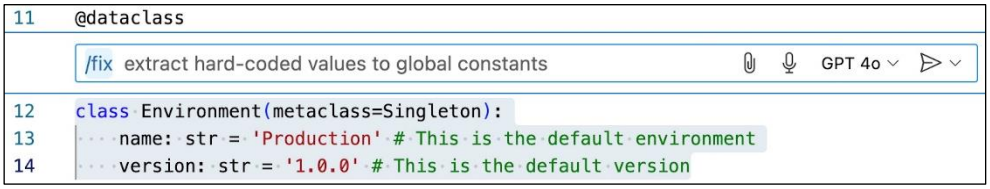


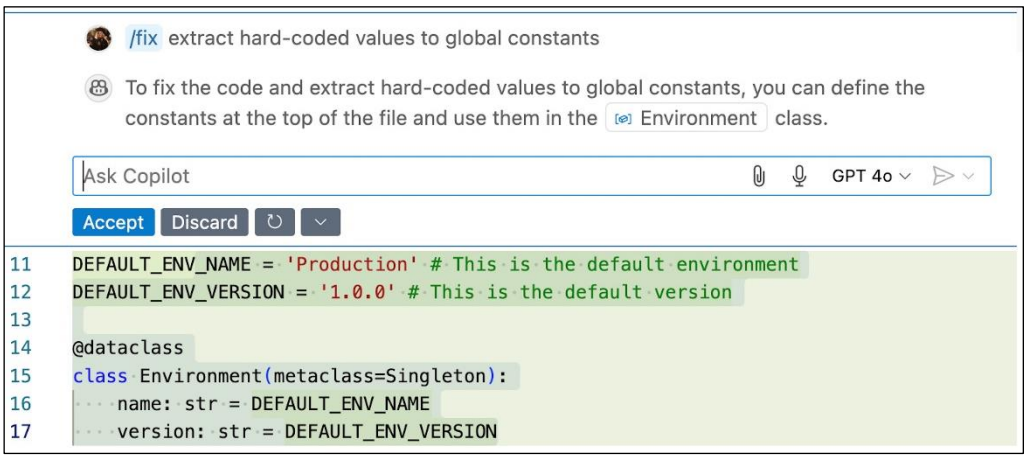


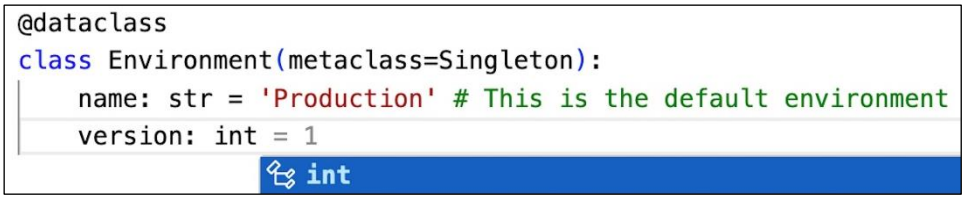








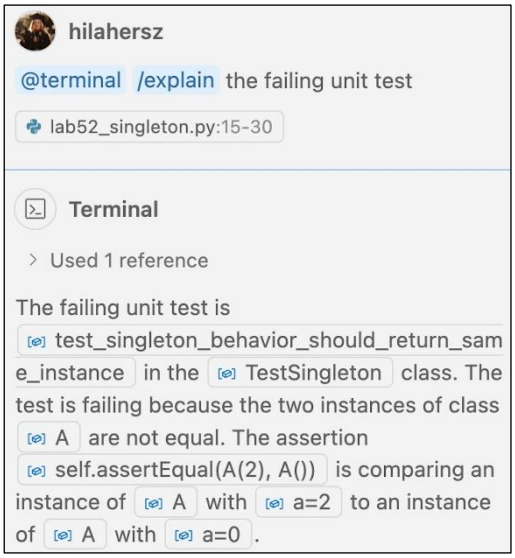


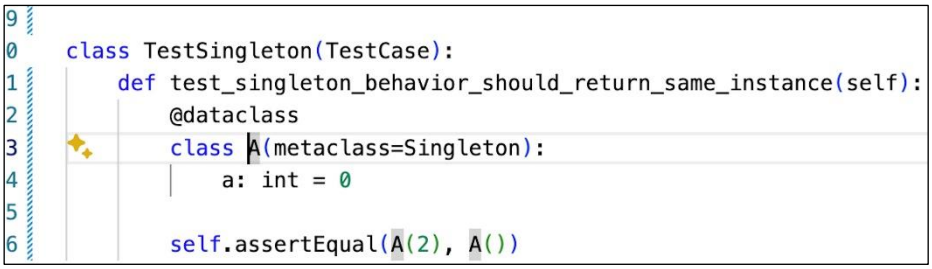


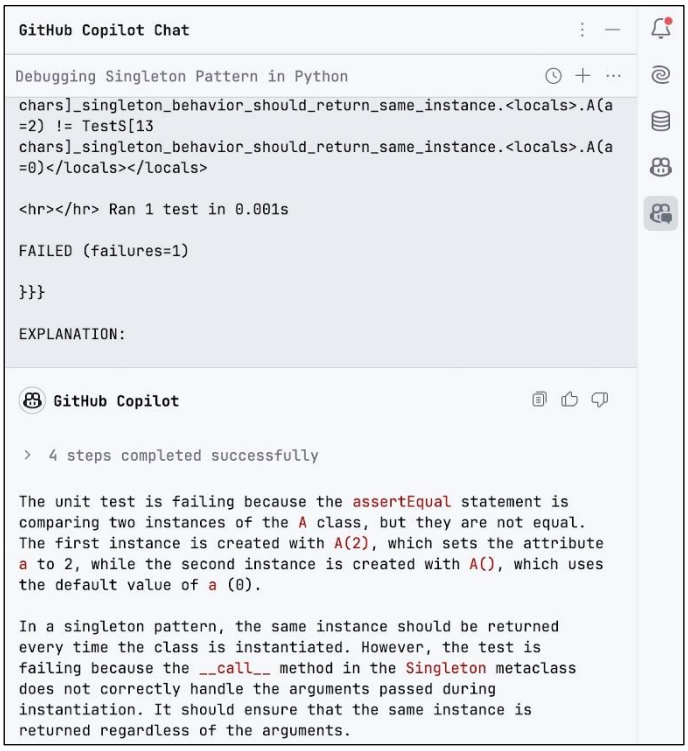

```
.venv➜  supercharge git:(main) ✗ python -m unittest ch5/labs/lab52_singleton.py
F
======================================================================
FAIL: test_singleton_behavior_should_return_same_instance (ch5.labs.lab52_singleton.TestSinglet
test_singleton_behavior_should_return_same_instance)
----------------------------------------------------------------------
Traceback (most recent call last):
  File "/Users/hila/PycharmProjects/private/supercharge/ch5/labs/lab52_singleton.py", line 26,
test_singleton_behavior_should_return_same_instance
    self.assertEqual(A(2), A())
AssertionError: TestS[13 chars]_singleton_behavior_should_return_same_instance.<locals>.A(a=2)
TestS[13 chars]_singleton_behavior_should_return_same_instance.<locals>.A(a=0)

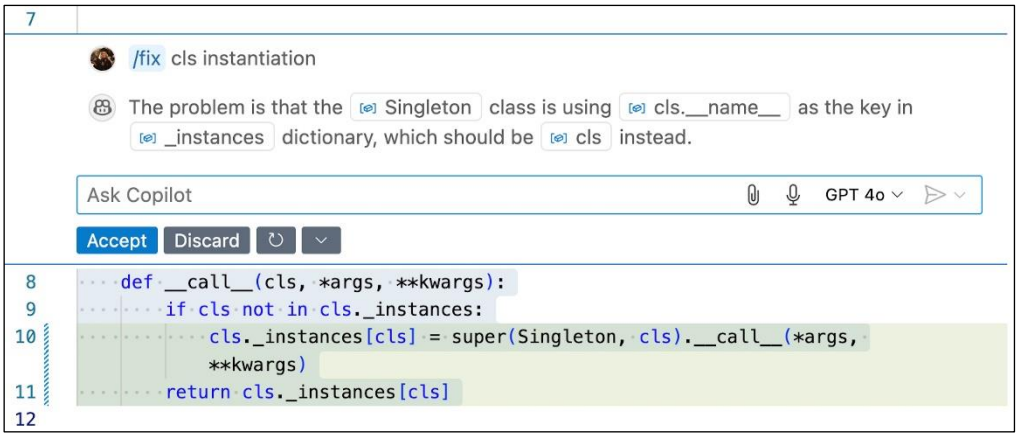
----------------------------------------------------------------------
Ran 1 test in 0.001s

FAILED (failures=1)
```


hilahersz
@terminal /explain the failing unit test
lab52_singleton.py:15-30
Terminal
Used 1 reference
The failing unit test is test_singleton_behavior_should_return_same_instance in the TestSingleton class. The test is failing because the two instances of class A are not equal. The assertion self.assertEqual(A(2), A()) is comparing an instance of A with a=2 to an instance of A with a=0.


class TestSingleton(TestCase):
def test_singleton_behavior_should_return_same_instance(self):
@dataclass
class A(metaclass=Singleton):
a: int = 0
self.assertEqual(A(2), A())

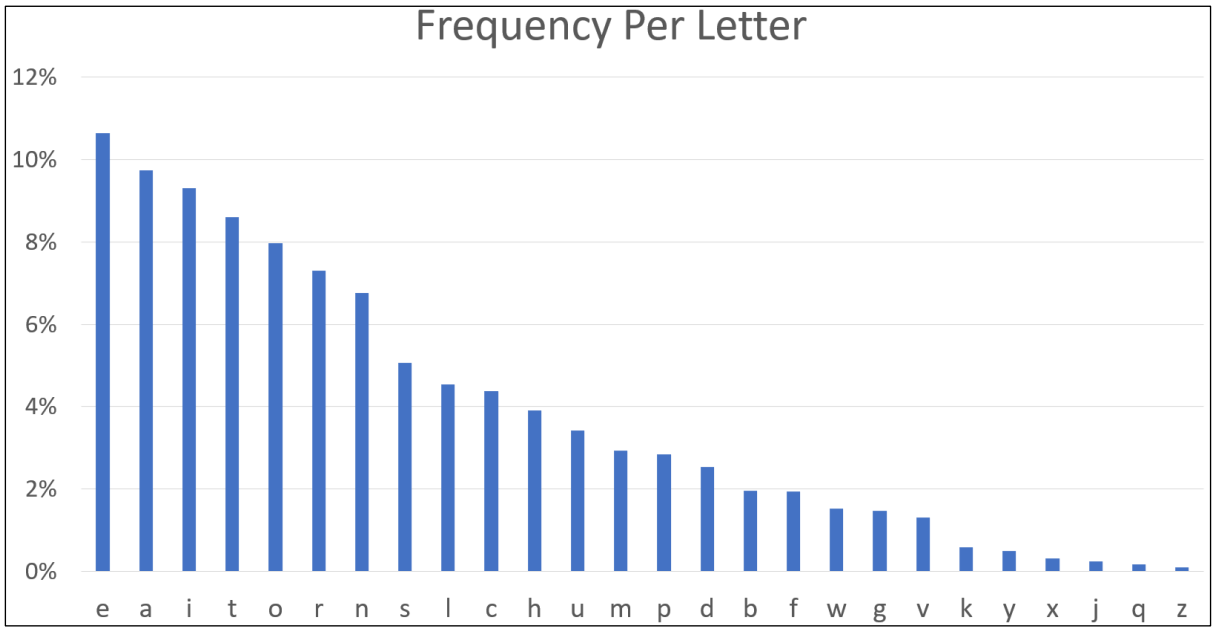

GitHub Copilot Chat
Debugging Singleton Pattern in Python
chars]_singleton_behavior_should_return_same_instance.<locals>.A(a=2) != TestS[13 chars]_singleton_behavior_should_return_same_instance.<locals>.A(a=0)</locals></locals>
<hr></hr> Ran 1 test in 0.001s
FAILED (failures=1)
}}}
EXPLANATION:
GitHub Copilot
4 steps completed successfully
The unit test is failing because the assertEqual statement is comparing two instances of the A class, but they are not equal. The first instance is created with A(2), which sets the attribute a to 2, while the second instance is created with A(), which uses the default value of a (0).
In a singleton pattern, the same instance should be returned every time the class is instantiated. However, the test is failing because the __call__ method in the Singleton metaclass does not correctly handle the arguments passed during instantiation. It should ensure that the same instance is returned regardless of the arguments.

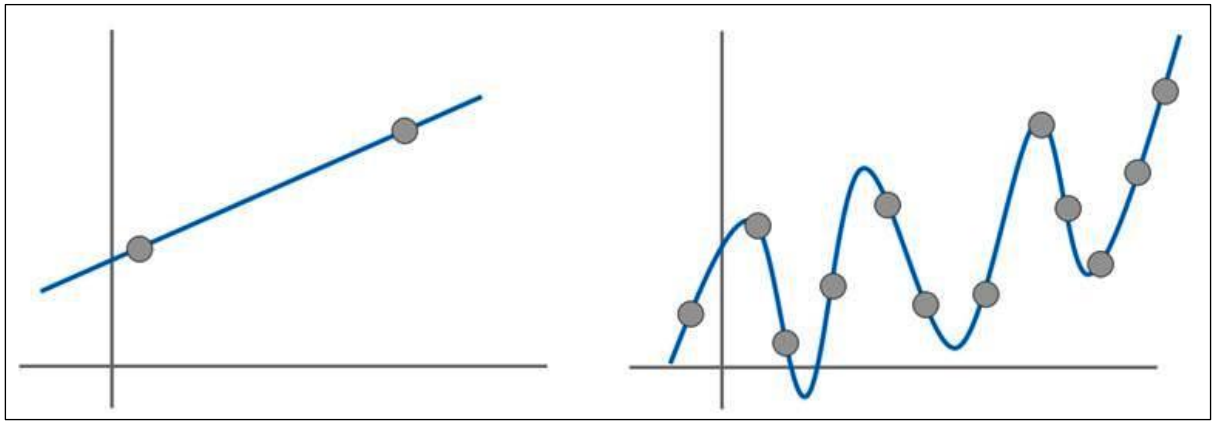



```
.venv➜  supercharge git:(main) python -m unittest ch5/labs/lab52.py
.
----------------------------------------------------------------------
Ran 1 test in 0.001s

OK
```

Chapter 6: Behind the Scenes: How ChatGPT, GitHub Copilot, and Other LLMs Work





1 eelnin ungia wwuetthgsaaeuuoiolifs ttelaroi ic so u dnli ohsaadoiolid
2 t ape ui preirl y ldueeseihenw eioxtincplastmtcteughct ivlrwataartrbi
3 ectee hcmasauisdeb eaeacwinsarsdu enti nuee dba io haqid kh aeeoa lr
4 aelplilvprag el nl einrileadneirearoafr ndstey m iser e hnpatfut hllbi
5 appdsmn rstthylan mvlccirefrrchneahbrgbmefw schnygtof mormde neehc

	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z
a	0.1	15.0	9.2	8.1	6.4	6.0	11.3	20.5	3.4	37.3	5.1	10.4	20.1	6.5	1.1	16.7	0.7	10.2	15.0	6.3	4.8	7.6	16.9	5.6	4.2	19.7
b	1.9	0.6	0.1	0.2	0.9	0.0	0.0	0.0	0.8	0.0	0.3	0.2	3.0	0.1	0.8	0.1	0.0	0.3	0.2	0.0	2.7	0.0	0.1	0.0	1.7	0.8
c	3.5	0.3	1.6	0.1	5.5	0.1	0.1	0.0	6.9	0.1	0.0	0.5	0.4	7.3	2.1	6.5	0.0	2.1	1.8	0.3	5.1	0.0	0.0	19.3	1.0	0.2
d	4.1	0.1	0.1	2.2	11.3	0.1	0.2	0.1	8.5	0.0	0.2	7.2	0.1	16.2	2.4	0.2	0.0	2.5	0.3	0.4	2.8	0.0	0.1	0.0	1.0	0.4
e	0.1	24.0	13.5	34.2	3.7	12.8	26.2	53.7	3.3	14.1	52.2	13.7	24.5	11.3	0.3	15.4	0.0	28.6	16.1	16.8	4.7	56.1	20.2	10.6	44.4	34.4
f	0.9	0.0	0.1	0.0	1.4	10.7	0.0	0.0	1.2	0.0	0.4	0.6	0.2	0.7	11.8	0.0	0.0	0.2	0.2	0.1	0.6	0.0	0.0	0.0	0.3	0.2
g	2.7	0.0	0.1	1.1	1.0	0.1	1.2	0.0	2.3	0.0	0.2	0.1	0.0	14.0	0.6	0.1	0.0	1.6	0.0	0.3	2.7	0.9	0.0	0.0	0.1	0.1
h	0.1	0.1	11.1	0.1	0.3	0.0	10.7	0.0	0.0	0.0	0.9	0.0	0.0	0.1	0.2	1.3	0.0	0.1	7.3	28.8	0.1	0.0	12.5	0.2	0.1	1.1
i	9.0	10.5	6.4	18.5	1.3	21.4	6.6	10.3	0.0	2.0	12.2	13.2	10.9	5.3	1.3	3.1	1.1	9.8	9.3	13.0	3.3	16.9	25.2	7.1	7.1	24.4
j	0.2	0.5	0.0	0.4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.1	0.2	0.0	0.0	0.0	0.0	0.0	0.1	0.0	0.0	0.0	0.0	0.0
k	1.1	0.0	3.3	0.0	0.5	0.0	0.1	0.0	0.3	0.0	0.3	0.5	0.0	2.5	0.8	0.0	0.0	2.2	0.6	0.0	0.1	0.0	0.1	0.0	0.1	0.1
l	10.8	8.1	3.3	10.4	3.9	2.6	1.6	0.3	7.2	0.2	2.0	14.6	9.2	0.5	4.0	7.6	0.0	1.4	1.0	0.8	9.3	0.1	0.4	0.2	1.2	0.9
m	1.9	0.3	0.1	0.8	3.1	0.1	0.6	0.2	2.1	0.0	0.5	0.4	4.9	0.8	8.7	1.0	0.0	2.6	1.0	0.6	3.0	0.0	0.2	0.2	6.2	0.2
n	21.2	0.1	0.0	0.3	11.1	0.0	5.0	0.5	26.9	0.1	1.7	5.9	0.1	1.8	21.7	0.0	0.0	2.6	0.1	0.1	12.0	0.0	4.2	0.2	2.9	0.1
o	0.1	10.6	24.7	6.2	0.3	24.5	9.5	6.9	8.5	22.0	3.3	8.3	10.5	5.5	1.3	13.7	0.0	11.5	5.9	13.8	0.5	3.6	15.0	1.4	7.3	6.9
p	2.3	0.4	0.1	0.2	1.7	0.0	0.1	0.0	0.9	0.1	0.2	0.4	10.3	0.1	3.2	4.3	0.0	1.6	3.0	0.1	4.3	0.0	0.0	39.8	1.1	0.7
q	0.1	0.0	0.7	0.1	0.4	0.0	0.0	0.0	0.1	0.0	0.0	0.0	0.0	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.1
r	13.5	6.1	3.6	2.0	20.2	11.7	15.5	3.2	2.9	0.5	1.1	5.6	0.1	0.1	15.9	22.6	0.0	1.9	0.1	5.7	15.4	0.2	1.2	0.0	0.7	0.1
s	8.1	2.2	0.6	3.9	13.0	0.7	3.8	0.9	8.4	0.0	15.5	3.6	2.3	6.2	4.0	1.0	0.0	8.8	7.3	7.1	10.0	13.6	2.1	0.1	18.2	0.4
t	12.0	2.5	15.6	0.2	4.5	3.6	1.2	2.0	12.6	0.0	0.2	4.3	0.5	15.7	4.2	2.6	7.6	5.7	22.8	1.3	17.5	0.0	1.3	14.0	1.1	0.2
u	1.1	10.3	4.6	8.8	2.4	5.1	4.8	0.7	0.2	23.5	1.2	2.8	2.2	1.7	8.7	3.6	90.3	1.7	6.4	2.0	0.0	0.1	0.0	0.3	0.8	4.9
v	1.6	0.1	0.0	0.7	2.0	0.0	0.0	0.0	2.6	0.0	0.0	0.5	0.0	1.2	2.9	0.0	0.0	1.0	0.0	0.0	0.1	0.0	0.0	0.0	0.0	0.1
w	0.4	0.0	0.0	0.3	1.1	0.0	0.0	0.1	0.1	0.1	0.4	0.1	0.0	0.1	3.4	0.1	0.0	0.3	0.5	1.0	0.1	0.1	0.0	0.2	0.3	1.0
x	0.4	0.0	0.0	0.0	2.8	0.0	0.0	0.0	0.3	0.0	0.0	0.0	0.0	0.0	0.1	0.0	0.0	0.0	0.0	0.0	0.1	0.0	0.0	0.4	0.0	0.0
y	2.5	8.2	1.2	1.3	1.0	0.3	1.5	0.2	0.0	0.0	2.0	7.0	0.6	2.2	0.4	0.1	0.0	3.3	0.8	1.4	0.8	0.5	0.1	0.5	0.0	1.2
z	0.3	0.0	0.0	0.0	0.1	0.0	0.0	0.0	0.4	0.0	0.1	0.0	0.0	0.1	0.0	0.0	0.0	0.0	0.0	0.1	0.1	0.0	0.0	0.0	0.1	1.9

1 wmarerye icose verhe herer duvoucrpa tilllarast itout f cts coles t tere
2 quron ryeurod g tst venk es wedus d it pll inopo antopcol ediomoforen f
3 lzin terop zquntreve re h tha hentwhase ues tabr iningte as ithormarape
4 rhelds sa tonat ainserathaby sard d aran ido rerealdon itino iconderilos
5 bagalintenin bs cutr tha mag temes cr teat owa th ondes brsumesais t g t

1 laa fith any sh wilhipect thenbleraqi alevid can of the sid in majoin sai
2 sciall appoilliv the takems red trichater jappetted siserseparcess feradd
3 ccpcts allansdaidebt cilliculeve remagre yestom the prify she and saiday
4 onhasharce by williedit salichansaid compansucers sankin connotin farkell
5 goo cout ovelucce any annes and a prommit sers consainse theartment ths w

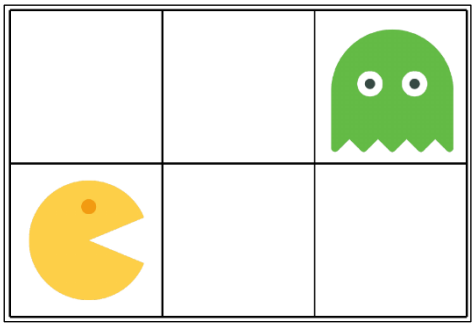
1 axistraders vice resgel ince trial pital pare to company and on mate donal
2 uma in the feeds under heasonal behinance out ints comesterly also for dea
3 mies are it wcashelp included to thanted quota committe of converal insti
4 jave affere in gover see west rainly of apriculates als of the tradio estm
5 tiall bancember shings offection this in chanies anding for tration aprice

1 frn in and corp told keeping an from years to market of about there listerd
2 ow fluctane cited under fight pct on the large share into mazda ways to the
3 orb states all year oper shr longer board if of worldwident effort said und
4 logan outstandar years for that they said the fund dives for the right pres
5 ority and its proving exposures said internative to tights income for throu

1 o at such a president quarter eased prices in london said subsidiary said it
2 res average of a dry eder chemical difficulture deterior quickly export cred
3 ten land ship offer committee changed mexico portugal item for example price
4 men violated at federal of the projects to a strict of francs offer industry
5 f customers approval by their old starting earn that order the governments m

1 aize subsidized by nippon is expects offer of the agreement earlier today to
2 ld start of corporate property owned from the exports of the chairman rich in
3 alty shortly the year from a floor and judicious market will come of the two
4 she city more discharge as its views last year as he said it would add some o
5 ed on industrial country in narrowed to buy up to the italian declined there

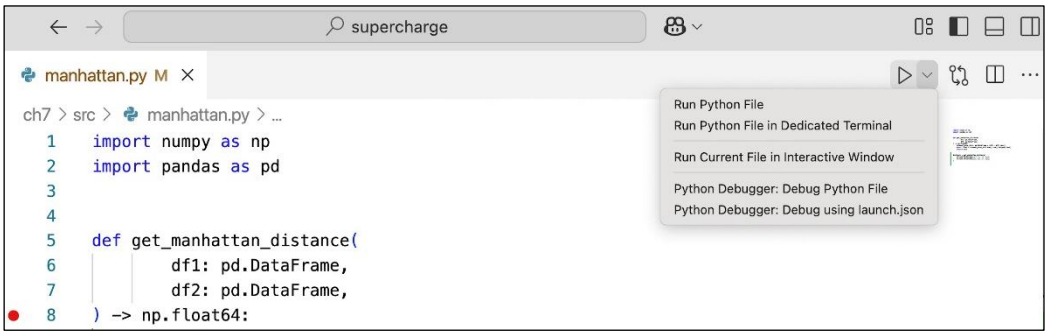
Chapter 7: Reading and Understanding Code Bases with GenAI



```python
def get_manhattan_distance(  7 usages  Hila Paz Herszfang
        df1: pd.DataFrame,
        df2: pd.DataFrame,
) -> np.float64:
    element_wise_dist: pd.DataFrame = (df1 - df2).abs()
    dist: float = element_wise_dist.sum().sum().astype(float)
    return dist


distance = get_manhattan_distance(
    df1=pd.DataFrame([[1, 2], [3, 4]]),
    df2=pd.DataFrame([[5, 6], [7, 8]]),
)
```





manhattan.py

```python
import numpy as np
import pandas as pd


def get_manhattan_distance(  7 usages  Hila Paz Herszfang
        df1: pd.DataFrame,   df1: [0, 1] [0  1  2] [1  3  4]
        df2: pd.DataFrame,   df2: [0, 1] [0  5  6] [1  7  8]
) -> np.float64:
    element_wise_dist: pd.DataFrame = (df1 - df2).abs()   element_wise_dist: [0, 1] [0  4  4] [1  4  4]
    dist: float = element_wise_dist.sum().sum().astype(float)
    return dist

```

construct_from_mng.py

```python
import pandas as pd

coordinates_a : pd.DataFrame = pd.DataFrame({
```

construct_from_mng.py

```python
import pandas as pd

coordinates_a : pd.DataFrame = pd.DataFrame({
    'x': [0, 1, 2],
    'y': [0, 1, 2],
```

```python
coordinates_a : pd.DataFrame = pd.DataFrame({
        "x": [0, 1, 2],
        "y": [0, 1, 2],
        "z": [0, 1, 2]
})

coordinates_a._constructor_from_mgr(
    coordinates_a._mgr,
    coordinates_a.axes,
)

```

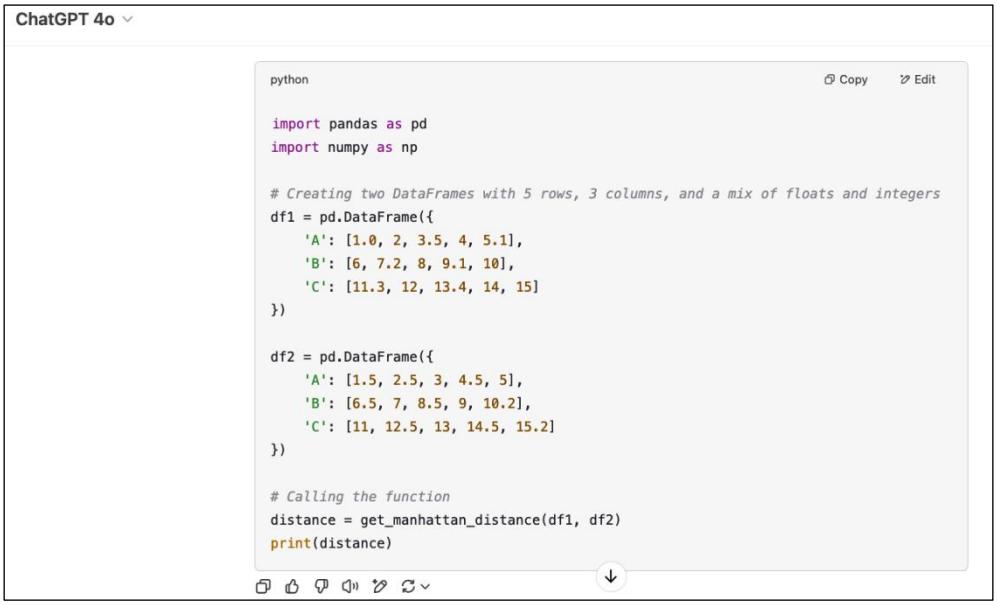
construct_from_mng.py frame.py

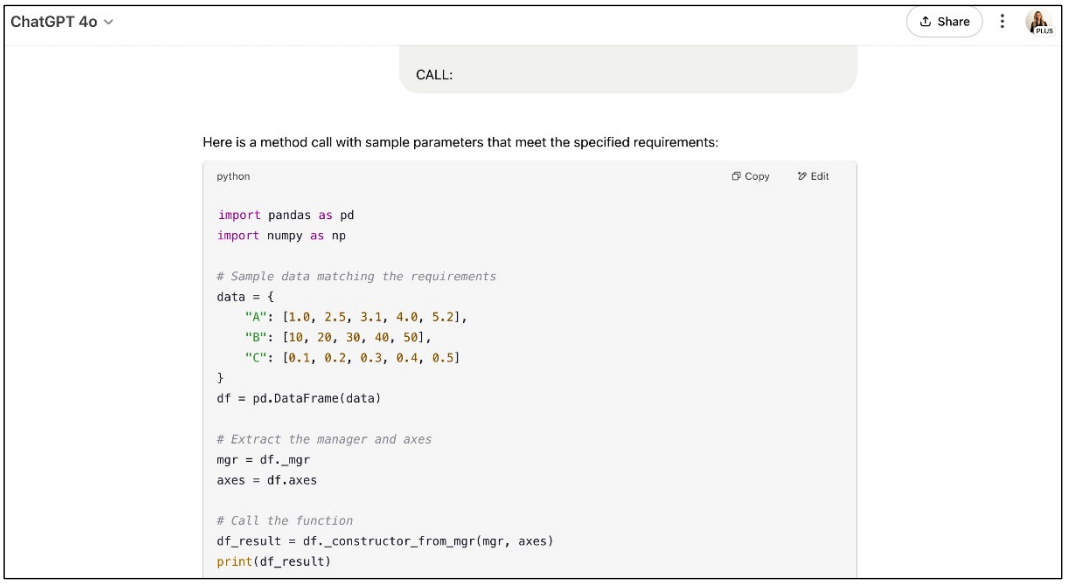
```python
class DataFrame(NDFrame, OpsMixin):


    def _constructor_from_mgr(self, mgr, axes) -> DataFrame:   axes: [RangeIndex(start=0, stop=3, step=1), Index(['x', 'y', 'z'],
        df = DataFrame._from_mgr(mgr, axes=axes)

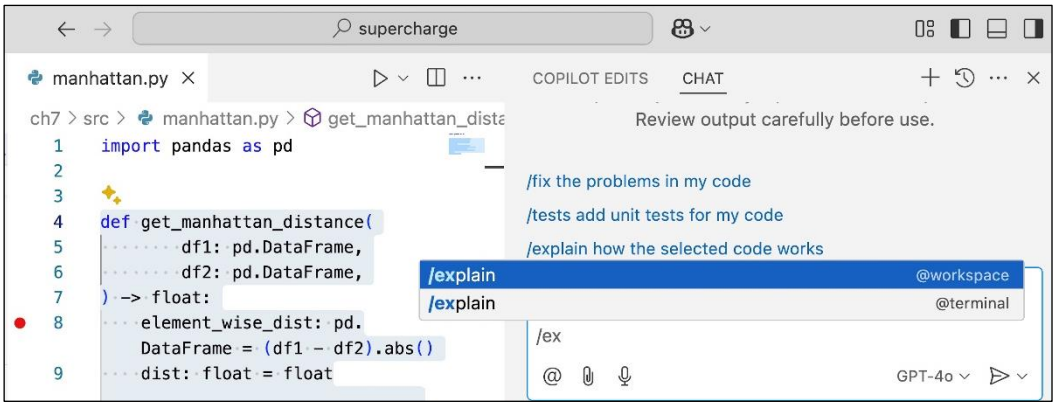
        if type(self) is DataFrame:
            # This would also work `if self._constructor is DataFrame`, but
            #  this check is slightly faster, benefiting the most-common case.
            return df

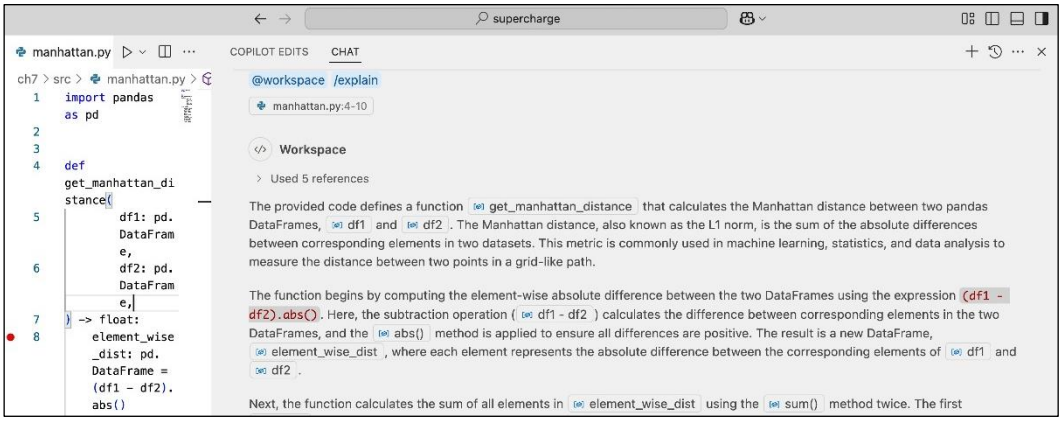
        elif type(self).__name__ == "GeoDataFrame":
            # Shim until geopandas can override their _constructor_from_mgr
            #  bc they have different behavior for Managers than for DataFrames
            return self._constructor(mgr)
```

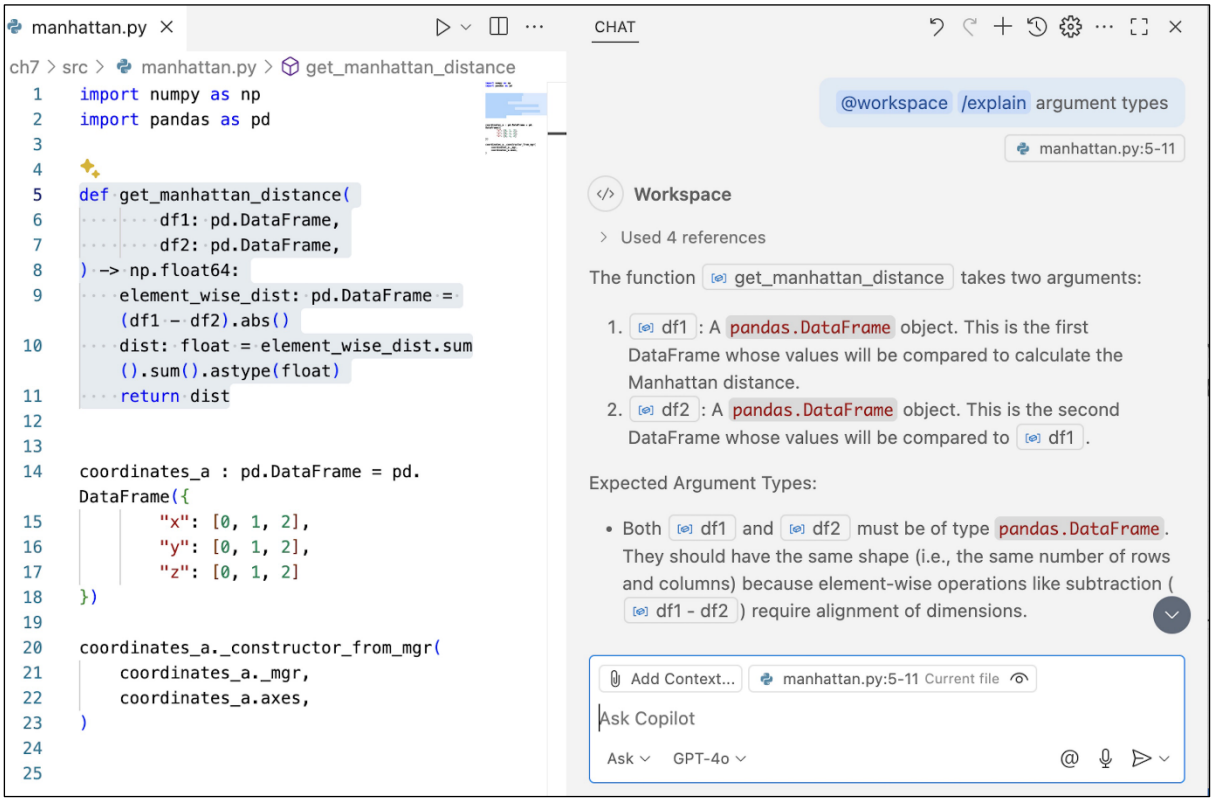

ChatGPT 4o
python
Copy
Edit
import pandas as pd
import numpy as np
# Creating two DataFrames with 5 rows, 3 columns, and a mix of floats and integers
df1 = pd.DataFrame({
'A': [1.0, 2, 3.5, 4, 5.1],
'B': [6, 7.2, 8, 9.1, 10],
'C': [11.3, 12, 13.4, 14, 15]
})
df2 = pd.DataFrame({
'A': [1.5, 2.5, 3, 4.5, 5],
'B': [6.5, 7, 8.5, 9, 10.2],
'C': [11, 12.5, 13, 14.5, 15.2]
})
# Calling the function
distance = get_manhattan_distance(df1, df2)
print(distance)

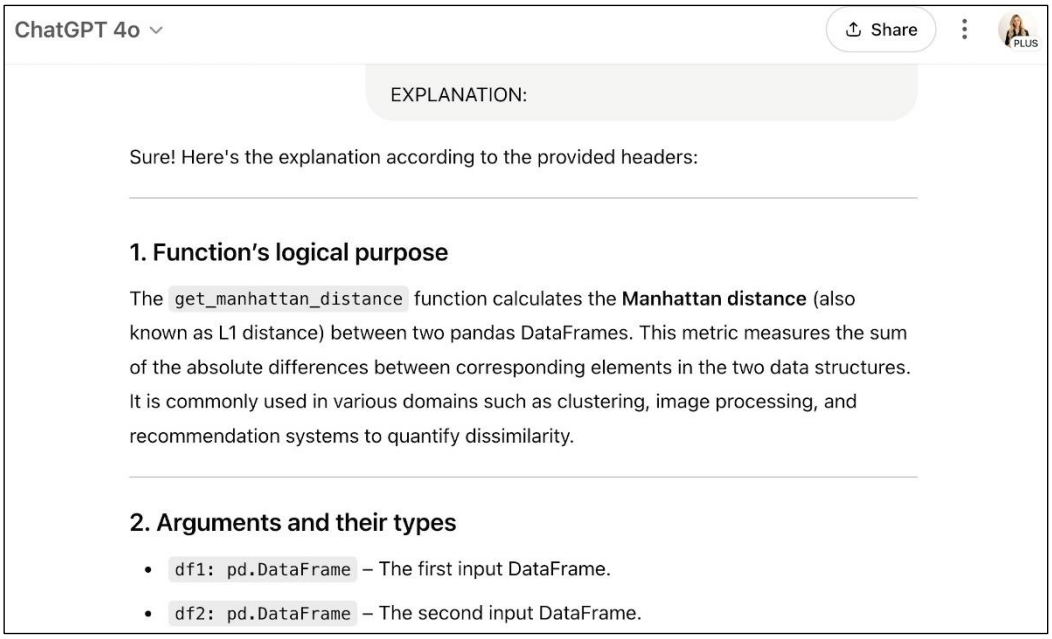

ChatGPT 4o
Share
CALL:
Here is a method call with sample parameters that meet the specified requirements:
python
Copy
Edit
import pandas as pd
import numpy as np
# Sample data matching the requirements
data = {
"A": [1.0, 2.5, 3.1, 4.0, 5.2],
"B": [10, 20, 30, 40, 50],
"C": [0.1, 0.2, 0.3, 0.4, 0.5]
}
df = pd.DataFrame(data)
# Extract the manager and axes
mgr = df._mgr
axes = df.axes
# Call the function
df_result = df._constructor_from_mgr(mgr, axes)
print(df_result)

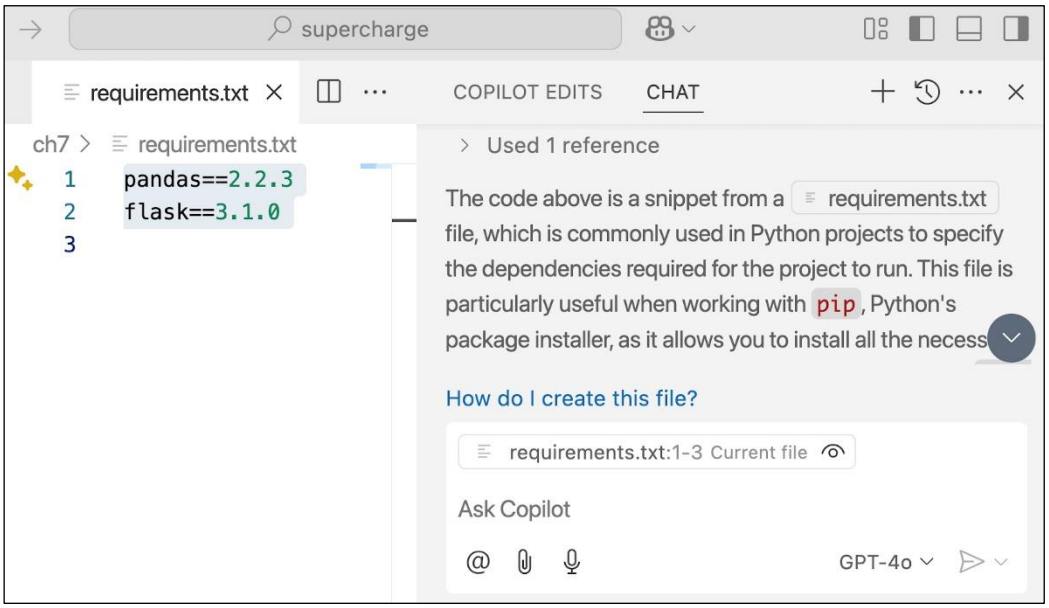

/explain
def get_manhattan_distance(6 usages
df1: pd.DataFrame,
df2: pd.DataFrame,
) -> float:
element_wise_dist: pd.DataFrame = (df1 - df2).abs()
dist: float = float(element_wise_dist.sum().sum())
return dist


supercharge
manhattan.py
COPILOT EDITS
CHAT
ch7 > src > manhattan.py > get_manhattan_dista
import pandas as pd
def get_manhattan_distance(
df1: pd.DataFrame,
df2: pd.DataFrame,
) -> float:
element_wise_dist: pd.
DataFrame = (df1 - df2).abs()
dist: float = float
Review output carefully before use.
/fix the problems in my code
/tests add unit tests for my code
/explain how the selected code works
/explain @workspace
/explain @terminal
/ex
GPT-4o

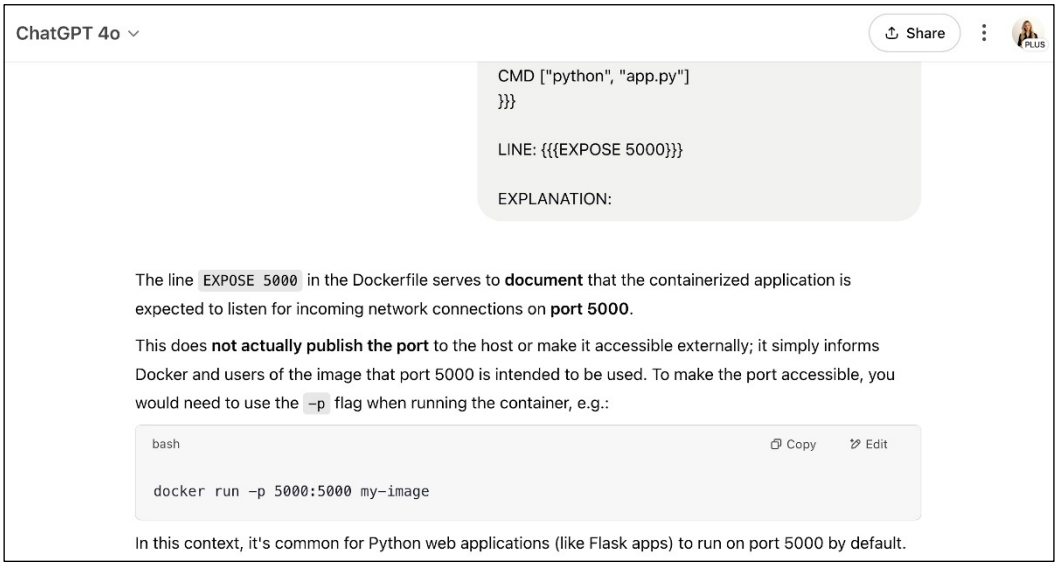

supercharge
manhattan.py
COPILOT EDITS
CHAT
ch7 > src > manhattan.py
@workspace /explain
manhattan.py:4-10
Workspace
Used 5 references
The provided code defines a function get_manhattan_distance that calculates the Manhattan distance between two pandas DataFrames, df1 and df2. The Manhattan distance, also known as the L1 norm, is the sum of the absolute differences between corresponding elements in two datasets. This metric is commonly used in machine learning, statistics, and data analysis to measure the distance between two points in a grid-like path.
The function begins by computing the element-wise absolute difference between the two DataFrames using the expression (df1 - df2).abs(). Here, the subtraction operation (df1 - df2) calculates the difference between corresponding elements in the two DataFrames, and the abs() method is applied to ensure all differences are positive. The result is a new DataFrame, element_wise_dist , where each element represents the absolute difference between the corresponding elements of df1 and df2 .
Next, the function calculates the sum of all elements in element_wise_dist using the sum() method twice. The first


manhattan.py
ch7 > src > manhattan.py > get_manhattan_distance
import numpy as np
import pandas as pd
def get_manhattan_distance(
df1: pd.DataFrame,
df2: pd.DataFrame,
) -> np.float64:
element_wise_dist: pd.DataFrame =
(df1 - df2).abs()
dist: float = element_wise_dist.sum
().sum().astype(float)
return dist
coordinates_a : pd.DataFrame = pd.
DataFrame({
"x": [0, 1, 2],
"y": [0, 1, 2],
"z": [0, 1, 2]
})
coordinates_a._constructor_from_mgr(
coordinates_a._mgr,
coordinates_a.axes,
)
CHAT
@workspace /explain argument types
manhattan.py:5-11
Workspace
Used 4 references
The function get_manhattan_distance takes two arguments:
1. df1 : A pandas.DataFrame object. This is the first DataFrame whose values will be compared to calculate the Manhattan distance.
2. df2 : A pandas.DataFrame object. This is the second DataFrame whose values will be compared to df1 .
Expected Argument Types:
• Both df1 and df2 must be of type pandas.DataFrame . They should have the same shape (i.e., the same number of rows and columns) because element-wise operations like subtraction (df1 - df2) require alignment of dimensions.
Add Context...
manhattan.py:5-11 Current file
Ask Copilot
Ask
GPT-4o


ChatGPT 4o
Share
PLUS
EXPLANATION:
Sure! Here's the explanation according to the provided headers:
1. Function's logical purpose
The get_manhattan_distance function calculates the Manhattan distance (also known as L1 distance) between two pandas DataFrames. This metric measures the sum of the absolute differences between corresponding elements in the two data structures. It is commonly used in various domains such as clustering, image processing, and recommendation systems to quantify dissimilarity.
2. Arguments and their types
df1: pd.DataFrame – The first input DataFrame.
df2: pd.DataFrame – The second input DataFrame.


supercharge
requirements.txt
COPILOT EDITS
CHAT
ch7
requirements.txt
1 pandas==2.2.3
2 flask==3.1.0
3
Used 1 reference
The code above is a snippet from a requirements.txt file, which is commonly used in Python projects to specify the dependencies required for the project to run. This file is particularly useful when working with pip, Python's package installer, as it allows you to install all the necess
How do I create this file?
requirements.txt:1-3 Current file
Ask Copilot
GPT-4o

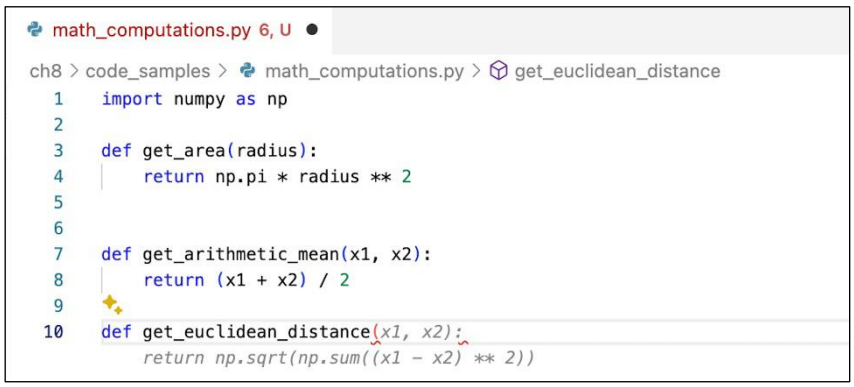

Dockerfile
8 COPY . .
9
GitHub Copilot
> 2 steps completed successfully
The line EXPOSE 5000 in the Dockerfile specifies that the container will listen for network connections on port 5000. This is typically used to indicate the port on which the application inside the container is expected to run. However, it does not actually publish the port; it serves as documentation and a hint for tools like docker run or docker-compose to map the container's port to the host system.
> Using 1 Reference
10 EXPOSE 5000
11
12 CMD ["python", "app.py"]
13


ChatGPT 4o
Share
PLUS
CMD ["python", "app.py"]
}}}
LINE: {{{EXPOSE 5000}}}
EXPLANATION:
The line EXPOSE 5000 in the Dockerfile serves to document that the containerized application is expected to listen for incoming network connections on port 5000.
This does not actually publish the port to the host or make it accessible externally; it simply informs Docker and users of the image that port 5000 is intended to be used. To make the port accessible, you would need to use the -p flag when running the container, e.g.:
bash
Copy
Edit
docker run -p 5000:5000 my-image
In this context, it's common for Python web applications (like Flask apps) to run on port 5000 by default.

Chapter 8: An Introduction to Prompt Engineering







math_computations.py 1, U ●

ch8 > code_samples > math_computations.py > get_eucleadian_distance

```
3   def get_area(
6       """
7       Compute the area of a circle given its radius.
8       """
9       area: float = np.pi * radius ** 2
10      return area
11
12
13  def get_arithmetic_mean(
14          x1: float,
15          x2: float,
16          ) -> float:
17      """
18      Compute the arithmetic mean of two numbers.
19      """
20      arithmetic_mean: float = (x1 + x2) / 2
21      return arithmetic_mean
22
23
24  def get_eucleadian_distance(
            x1: float,
            y1: float,
            x2: float,
            y2: float,
            ) -> float:
        """
        Compute the Euclidean distance between two points.
        """
        eucleadian_distance: float = np.sqrt((x2 - x1) ** 2 + (y2 - y1) ** 2)
        return eucleadian_distance
25
```

style_guide.py M

ch8 > code_samples > style_guide.py > ...

```python
import numpy as np

def get_area(
        radius: float,
        ) -> float:
    area: float = np.pi * radius ** 2
    return area


def get_arithmetic_mean(
        x1: float,
        x2: float,
        ) -> float:
    arithmetic_mean: float = (x1 + x2) / 2
    return arithmetic_mean

```

style_guide.py M | distances.py 1, U

ch8 > code_samples > distances.py > get_eucleadian_distance

```python
def get_eucleadian_distance(
        x1: float,
        x2: float,
        y1: float,
        y2: float,
        ) -> float:
    distance: float = ((x2 - x1) ** 2 + (y2 - y1) ** 2) ** 0.5
    return distance
```

chain_of_thought.py 3, R

ch8 > code_samples > chain_of_thought.py > ...

```python
import ast


def validate_no_bugs_in_source_code(sc: str) -> None:
    validate_syntax(sc)
    validate_compilation(sc)
    validate_reoroducibility_across_runs(sc)


def validate_syntax(sc: str) -> None:
    try:
        ast.parse(sc)
    except SyntaxError:
        raise ValueError("Syntax error in source code")

```

Chapter 9: Advanced Prompt Engineering for Coding-Related Tasks

get_average_return.py

```python
from typing import Dict

def get_geometric_mean(  new *
        net_returns: Dict[str, float],
) -> float:
    product: float = 1
    for key in net_returns:
        product *= net_returns[key]
    geometric_mean: float = product ** (1 / len(net_returns))
    return geometric_mean
```

ChatGPT 4o

Share

FIXED CODE:

Here's the fixed implementation of the code that will pass all the provided unit tests:

python

Copy Edit

```python
import math
import cmath  # Import cmath for handling complex numbers

def get_geometric_mean(a, b):
    if a * b < 0:  # If one value is negative, return a complex re
        return cmath.sqrt(a * b)
    elif a < 0 and b < 0:  # If both values are negative, convert
        a, b = abs(a), abs(b)
    return math.sqrt(a * b)
```

Ask anything

cot_copilot.py

```python
# What is the next number in the sequence? 17, 34, 14, 28, 8, 16, 4
```

cot_copilot.py

```
# What is the next number in the sequence?

# Number 1: 17
# Number 2: 34. Relationship to Number 1: 2
# Number 3: 14. Relationship to Number 2: -20
# Number 4: 28. Relationship to Number 3: 2
# Number 5: 8. Relationship to Number 4: -20
# Number 6: 16. Relationship to Number 5: 2
# Number 7: 6. Relationship to Number 6: -20
```

What is the next number in the sequence? 17, 34, 14, 28, 8, 16,

Thought about number sequence pattern for 47 seconds

A concise way to see the pattern is that the terms alternate between "double the previous term" and "subtract 20 from the previous term." Concretely:

1. $17 \xrightarrow{\times 2} 34$
2. $34 \xrightarrow{-20} 14$
3. $14 \xrightarrow{\times 2} 28$
4. $28 \xrightarrow{-20} 8$
5. $8 \xrightarrow{\times 2} 16$
6. $16 \xrightarrow{-20} -4$

So the next number in the sequence is $\boxed{-4}$.

ChatGPT o1

What is the next number in the sequence? 17, 34, 14, 28, 8, 16,

Analyzing the sequence

Our latest models

Reasoning models for complex, multi-step problems

OpenAI o1

Frontier reasoning model that supports tools, Structured Outputs, and vision | 200k context length

Price

Input:
$15.00 / 1M tokens

Cached input:
$7.50 / 1M tokens

Output:
$60.00 / 1M tokens

OpenAI o3-mini

Small cost-efficient reasoning model that's optimized for coding, math, and science, and supports tools and Structured Outputs | 200k context length

Price

Input:
$1.10 / 1M tokens

Cached input:
$0.55 / 1M tokens

Output:
$4.40 / 1M tokens

GPT models for everyday tasks

GPT-4.5

Largest GPT model designed for creative tasks and agentic planning, currently available in a research preview | 128k context length

Price

Input:
$75.00 / 1M tokens

Cached input:
$37.50 / 1M tokens

Output:
$150.00 / 1M tokens

GPT-4o

High-intelligence model for complex tasks | 128k context length

Price

Input:
$2.50 / 1M tokens

Cached input:
$1.25 / 1M tokens

Output:
$10.00 / 1M tokens

GPT-4o mini

Affordable small model for fast, everyday tasks | 128k context length

Price

Input:
$0.150 / 1M tokens

Cached input:
$0.075 / 1M tokens

Output:
$0.600 / 1M tokens

```
5
6   def get_average_return(  Hila Paz Herszfang *
7           net_returns: Dict[str, float],
8   ) -> float:
9       gross_returns: np.ndarray = get_gross_returns(net_returns)
10      gross_average: float = get_geometric_mean(gross_returns)
11      net_average: float = get_net_average(gross_average)
12      return net_average
13
```

```python
from typing import Dict

import numpy as np


def get_average_return(  Hila Paz Herszfang *
        net_returns: Dict[str, float],
) -> float:
    gross_returns: np.ndarray = get_gross_returns(net_returns)
    gross_average: float = get_geometric_mean(gross_returns)
    net_average: float = get_net_average(gross_average)
    return net_average


def get_gross_returns( new *
        net_returns: Dict[str, float],
) -> np.ndarray:
    gross_returns: np.ndarray = np.array(list(net_returns.values())) + 1
    return gross_returns
```

```python
def get_average_return(  Hila Paz Herszfang *
        net_returns: Dict[str, float],
) -> float:
    gross_returns: np.ndarray = get_gross_returns(net_returns)
    gross_average: float = get_geometric_mean(gross_returns)
    net_average: float = get_net_average(gross_average)
    return net_average


def get_gross_returns(  1 usage  Hila Paz Herszfang *
        net_returns: Dict[str, float],
) -> np.ndarray:
    gross_returns: np.ndarray = np.array(list(net_returns.values())) + 1
    return gross_returns


def get_geometric_mean(
```

ChatGPT 4o

Share

Add type hints to all variables.

Sure! Here's the updated version with **explicit type hints** for all variables inside the functions:

python Copy Edit

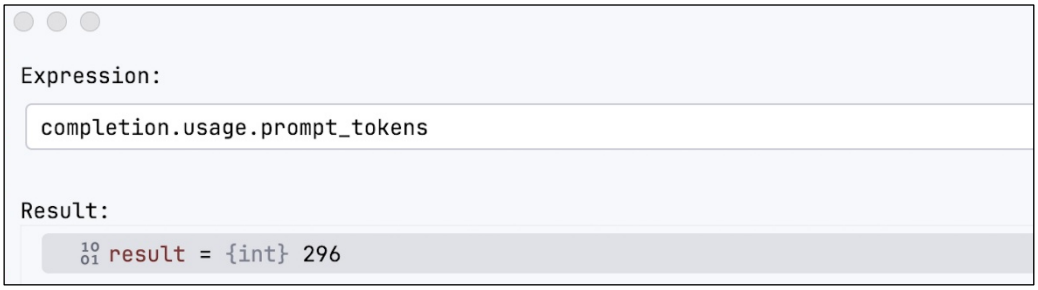
```python
import numpy as np
from typing import Dict

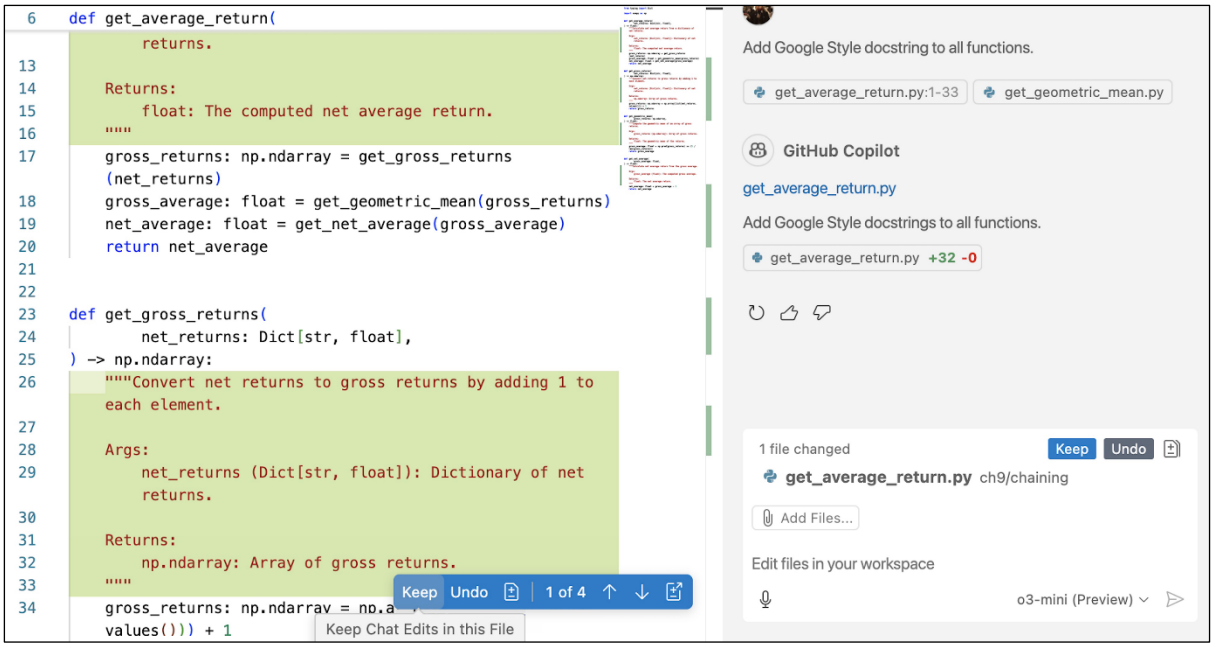
def get_gross_returns(net_returns: Dict[str, float]) -> np.ndarray:
    """
    Convert net returns (e.g., 0.05 for 5%) to gross returns (e.g., 1.05).
    """
    gross_returns: np.ndarray = np.array([1 + r for r in net_returns.values()])
    return gross_returns

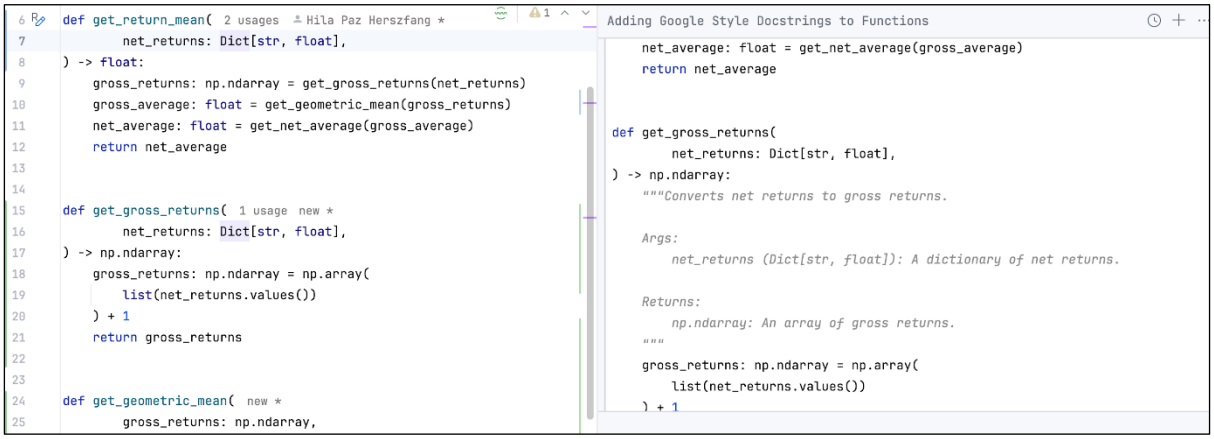
def get_geometric_mean(gross_returns: np.ndarray) -> float:
    """
    Compute the geometric mean of an array of gross returns.
    """
    product: float = np.prod(gross_returns)
```


ChatGPT 4o
Share
include Google Style docstring.
Sure! Here's the version with full **Google Style docstrings** and all **variable type hints**:
python
Copy
Edit
import numpy as np
from typing import Dict
def get_gross_returns(net_returns: Dict[str, float]) -> np.ndarray:
"""Converts net returns to gross returns.
Args:
net_returns (Dict[str, float]): A dictionary of net returns,
where the key is typically a time period and the value is the net return
Returns:
np.ndarray: An array of gross returns (e.g., 1.05 for a 5% net return).
"""
gross_returns: np.ndarray = np.array([1 + r for r in net_returns.values()])

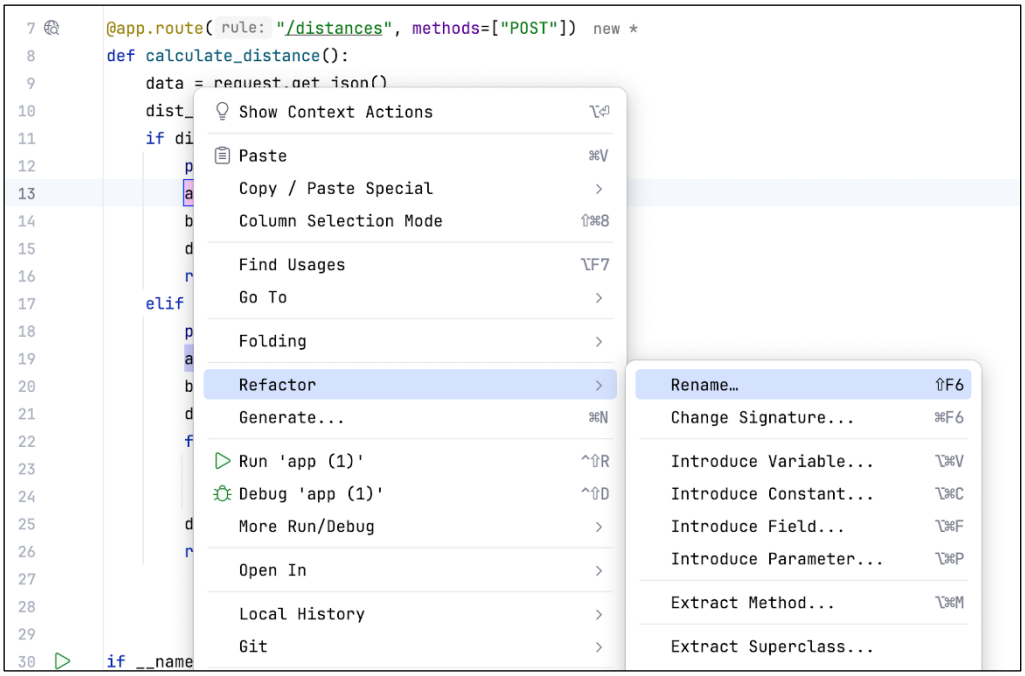

output: str = completion.choices[0].message.content output: "Here's the implementation of the functions
messages.append({"role": "assistant", "content": output})
print(output)
Evaluate
Expression:
completion.usage.prompt_tokens
Result:
result = {int} 1253
Threads & Va


Expression:
completion.usage.prompt_tokens
Result:
result = {int} 296

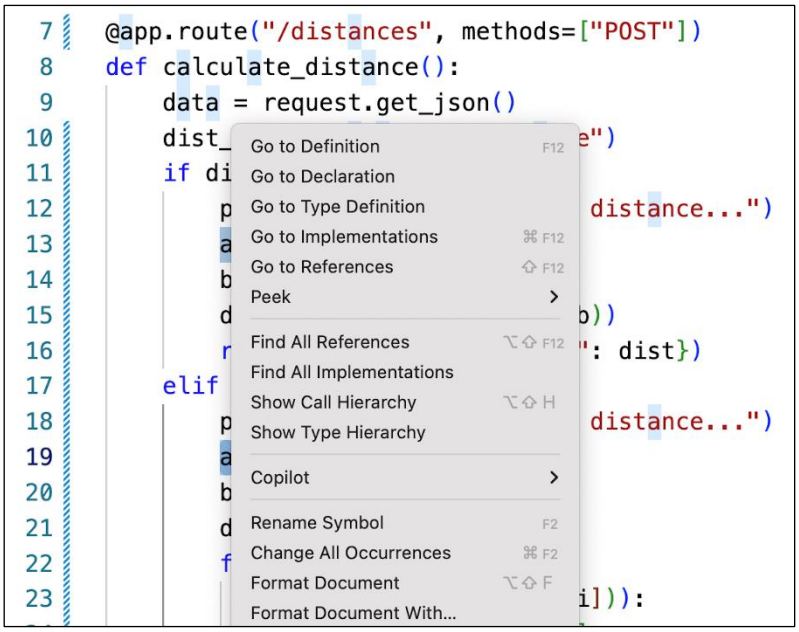

def get_average_return(
returns.
Returns:
float: The computed net average return.
"""
gross_returns: np.ndarray = get_gross_returns
(net_returns)
gross_average: float = get_geometric_mean(gross_returns)
net_average: float = get_net_average(gross_average)
return net_average
def get_gross_returns(
net_returns: Dict[str, float],
) -> np.ndarray:
"""Convert net returns to gross returns by adding 1 to
each element.
Args:
net_returns (Dict[str, float]): Dictionary of net
returns.
Returns:
np.ndarray: Array of gross returns.
"""
values())) + 1
Keep
Undo
1 of 4
Keep Chat Edits in this File
Add Google Style docstring to all functions.
get_average_return.py:1-33
get_geometric_mean.py
GitHub Copilot
get_average_return.py
Add Google Style docstrings to all functions.
get_average_return.py +32 -0
1 file changed
Keep
Undo
get_average_return.py ch9/chaining
Add Files...
Edit files in your workspace
o3-mini (Preview)

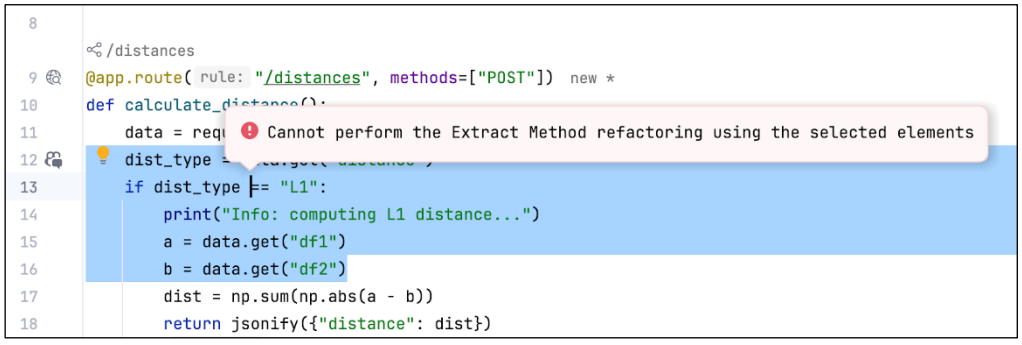

def get_return_mean(2 usages Hila Paz Herszfang *
net_returns: Dict[str, float],
) -> float:
gross_returns: np.ndarray = get_gross_returns(net_returns)
gross_average: float = get_geometric_mean(gross_returns)
net_average: float = get_net_average(gross_average)
return net_average
def get_gross_returns(1 usage new *
net_returns: Dict[str, float],
) -> np.ndarray:
gross_returns: np.ndarray = np.array(
list(net_returns.values())
) + 1
return gross_returns
def get_geometric_mean(new *
gross_returns: np.ndarray,
Adding Google Style Docstrings to Functions
net_average: float = get_net_average(gross_average)
return net_average
def get_gross_returns(
net_returns: Dict[str, float],
) -> np.ndarray:
"""Converts net returns to gross returns.
Args:
net_returns (Dict[str, float]): A dictionary of net returns.
Returns:
np.ndarray: An array of gross returns.
"""
gross_returns: np.ndarray = np.array(
list(net_returns.values())
) + 1

Chapter 10: Refactoring Code with GenAI






@app.route("/distances", methods=["POST"])
def calculate_distance():
data = request.get_json()
Go to Definition F12
Go to Declaration
Go to Type Definition
Go to Implementations ⌘ F12
Go to References ⇧ F12
Peek
Find All References ⌥⇧ F12
Find All Implementations
Show Call Hierarchy ⌥⇧ H
Show Type Hierarchy
Copilot
Rename Symbol F2
Change All Occurrences ⌘ F2
Format Document ⌥⇧ F
Format Document With...


/distances
@app.route(rule: "/distances", methods=["POST"]) new *
def calculate_distance():
Cannot perform the Extract Method refactoring using the selected elements
if dist_type == "L1":
print("Info: computing L1 distance...")
a = data.get("df1")
b = data.get("df2")
dist = np.sum(np.abs(a - b))
return jsonify({"distance": dist})

```
7   @app.route("/distances", methods=["POST"])
8   def calculate_distance():
9       data = request.get_json()
        dist_type = data.get("distance")
        if dist_type == "L1":
            print("Info: computing L1 distance...")
            a = np.array(data.get("df1"))
            b = np.array(data.get("df2"))
            dist = np.sum(np.abs(a - b))
            return jsonify({"distance": dist})
        elif dist_type == "L2":
            print("Info: computing L2 distance...")
            a = np.array(data.get("df1"))
            b = np.array(data.get("df2"))
            dist_2 = 0
            for i in range(len(a)):
                for j in range(len(a[i])):
                    dist_2 += (a[i][j] - b[i][j]) ** 2
            dist = np.sqrt(dist_2)
            return jsonify({"distance": dist})
10
11
12  @app.route("/distances", methods=["POST"])
13  def calculate_distance():
```

```
7   @app.route("/distances", methods=["POST"])
8   def calculate_distance():
9       a, b, dist_type = parse_request_parameters(request)
10      dist_func = {"L1": get_manhattan_dist, "L2": get_euclidean_dist}.get
        (dist_type)
11      dist = dist_func(a, b)
12      return jsonify({"distance": dist})
13
14
15  def parse_request_parameters(request):
```

```python
@app.route("/distances", methods=["POST"])
def calculate_distance():
    a, b, dist_type = parse_request_parameters(request)
    dist_func = {"L1": get_manhattan_dist, "L2": get_euclidean_dist}.get
    (dist_type)
    dist = dist_func(a, b)
    return jsonify({"distance": dist})

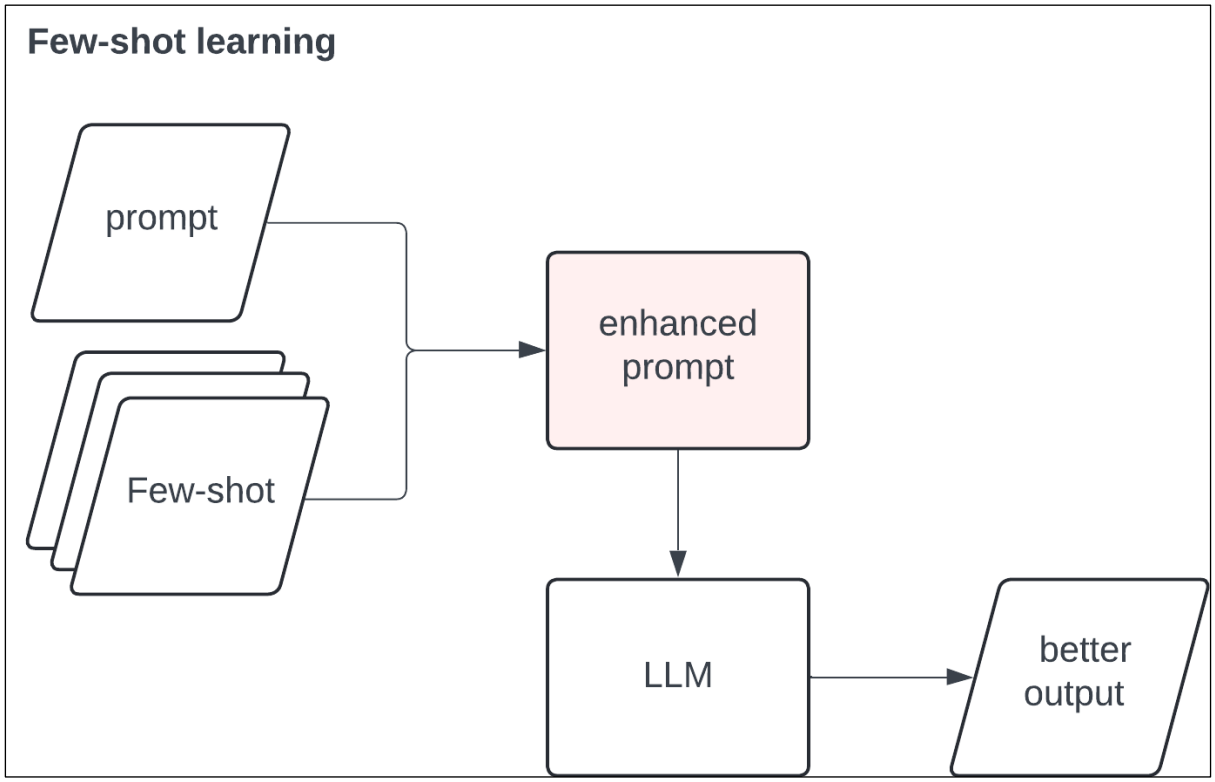

def parse_request_parameters(request):
    data = request.get_json()
    a = np.array(data.get("df1"))
    b = np.array(data.get("df2"))
    dist_type = data.get("distance")
    return a, b, dist_type
```

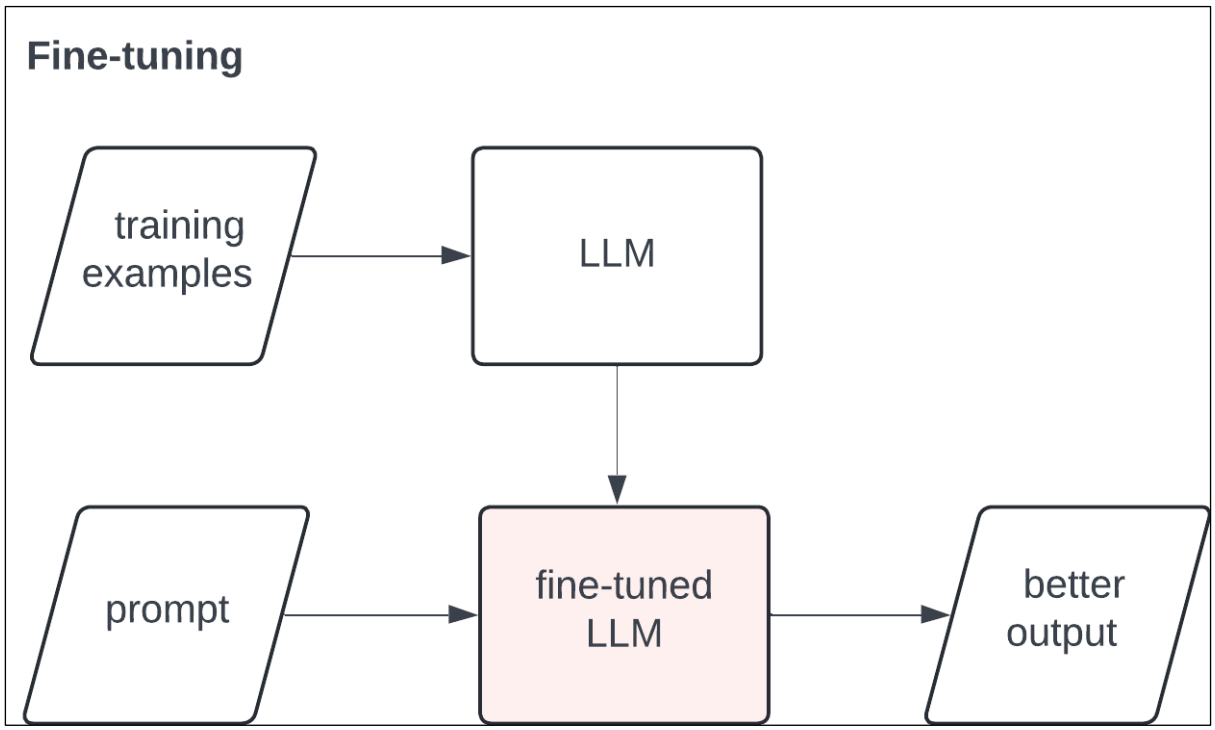


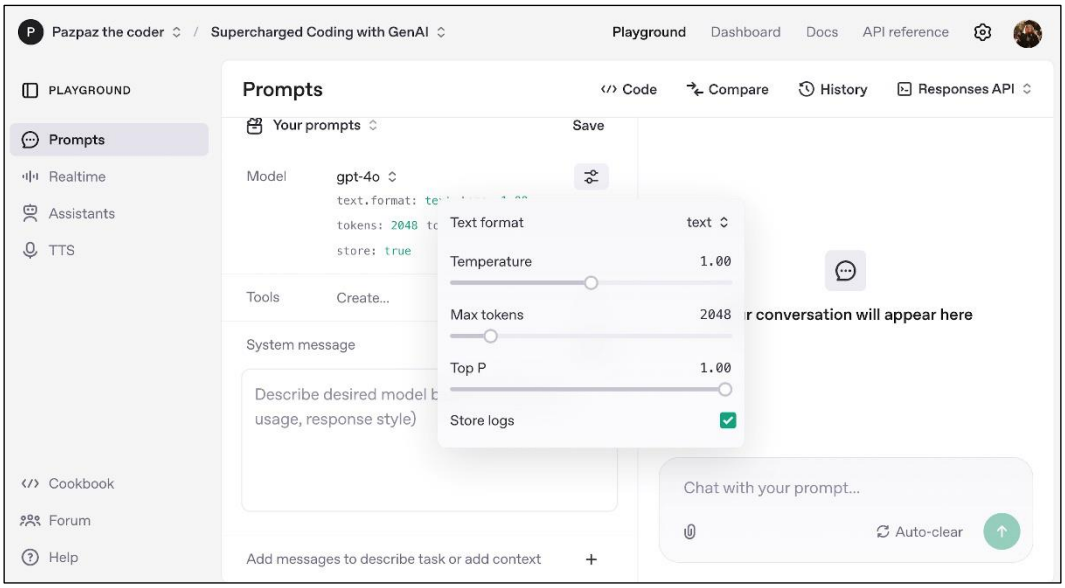
```
28      def get_euclidean_dist(a, b):  1 usage  new *
29          print("Info: computing L2 distance...")
30          dist_2 = np.sum((a - b) ** 2)
31
32
33          dist_2 = 0
34          for i in range(len(a)):
35              for j in range(len(a[i])):
36                  dist_2 += (a[i][j] - b[i][j]) ** 2
37          return np.sqrt(dist_2)
```

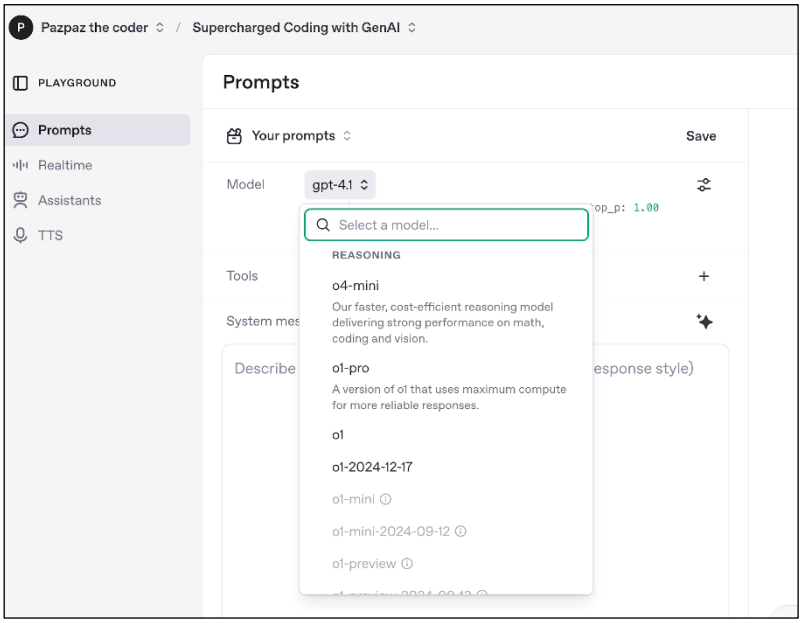
```
28      def get_euclidean_dist(a, b):  1 usage  new *
29          print("Info: computing L2 distance...")
30          dist_2 = np.sum((a - b) ** 2)
31          dist = np.sqrt(dist_2)
32
33
34          dist_2 = 0
35          for i in range(len(a)):
36              for j in range(len(a[i])):
37                  dist_2 += (a[i][j] - b[i][j]) ** 2
38          return np.sqrt(dist_2)
```

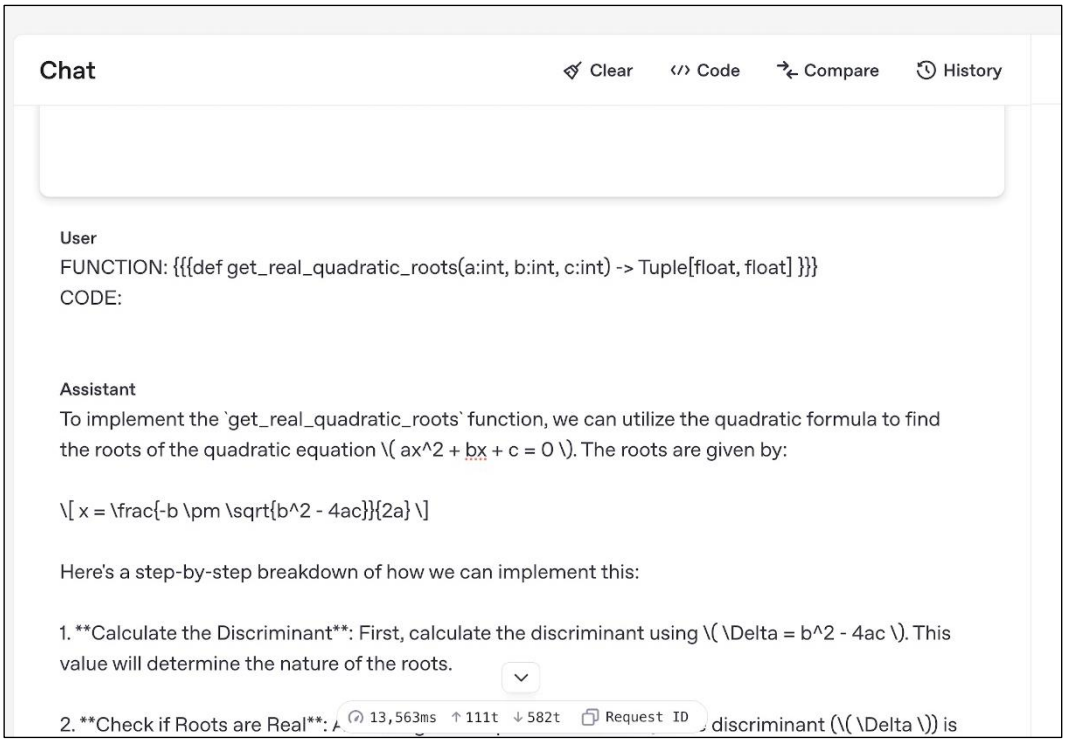
Chapter 11: Fine-Tuning Models with OpenAI






Pazpaz the coder / Supercharged Coding with GenAI
Playground
Dashboard
Docs
API reference
PLAYGROUND
Prompts
Realtime
Assistants
TTS
Cookbook
Forum
Help
Prompts
Code
Compare
History
Responses API
Your prompts
Save
Model
gpt-4o
text.format: te
tokens: 2048 to
store: true
Text format
text
Temperature
1.00
Max tokens
2048
Top P
1.00
Store logs
Tools
Create...
System message
Describe desired model b
usage, response style)
r conversation will appear here
Chat with your prompt...
Auto-clear
Add messages to describe task or add context


Pazpaz the coder / Supercharged Coding with GenAI
PLAYGROUND
Prompts
Realtime
Assistants
TTS
Prompts
Your prompts
Save
Model
gpt-4.1
op_p: 1.00
Select a model...
REASONING
o4-mini
Our faster, cost-efficient reasoning model delivering strong performance on math, coding and vision.
o1-pro
A version of o1 that uses maximum compute for more reliable responses.
o1
o1-2024-12-17
o1-mini
o1-mini-2024-09-12
o1-preview
Tools
System mes
Describe
esponse style)



Fine-tuning our models

Customize our models to get even higher performance for your specific use cases.

GPT-4o

Fine-tuning price

Input:
$3.750 / 1M tokens

Cached input:
$1.875 / 1M tokens

Output:
$15.000 / 1M tokens

Training:
$25.000 / 1M tokens

GPT-4o mini

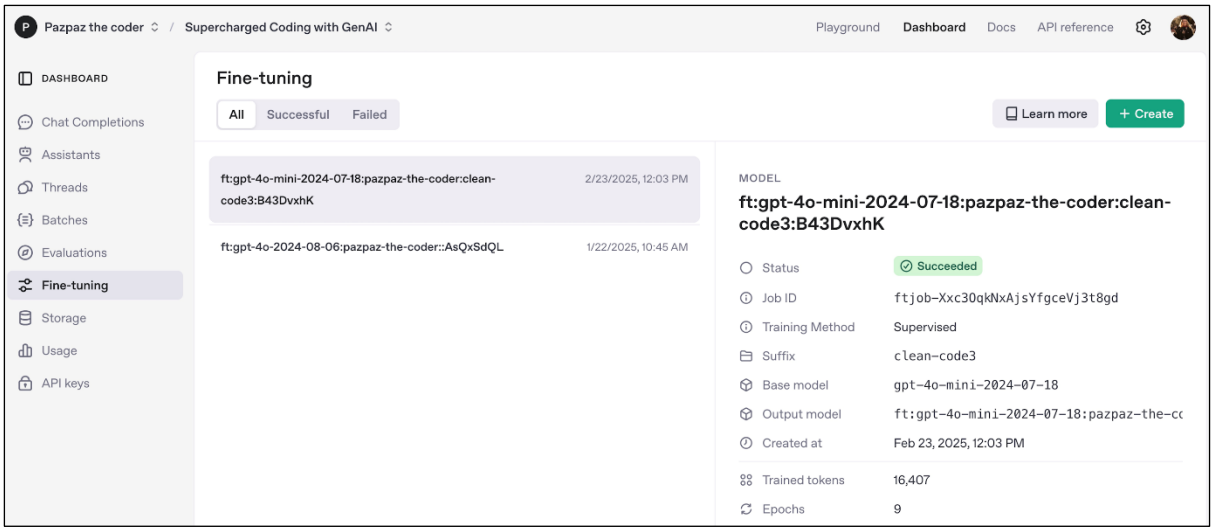
Fine-tuning price

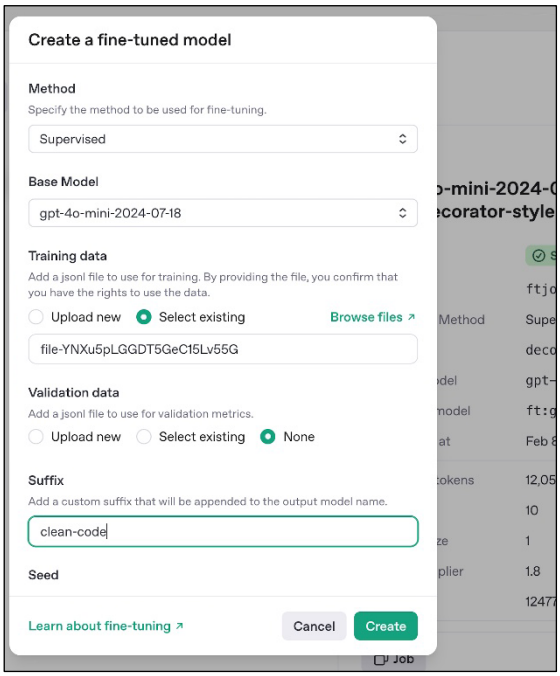
Input:
$0.30 / 1M tokens

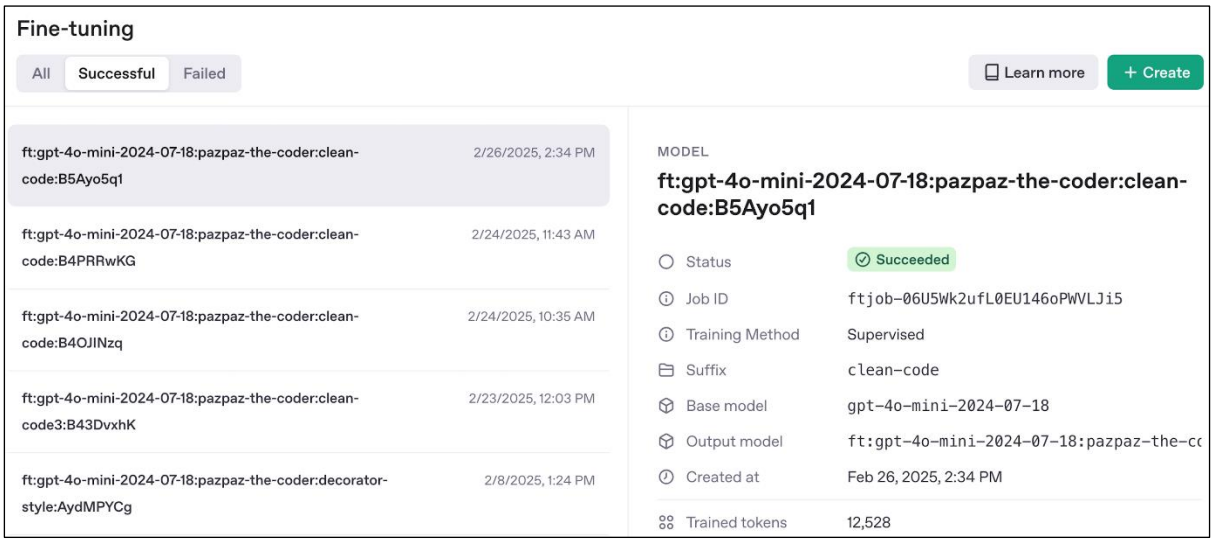
Cached input:
$0.15 / 1M tokens

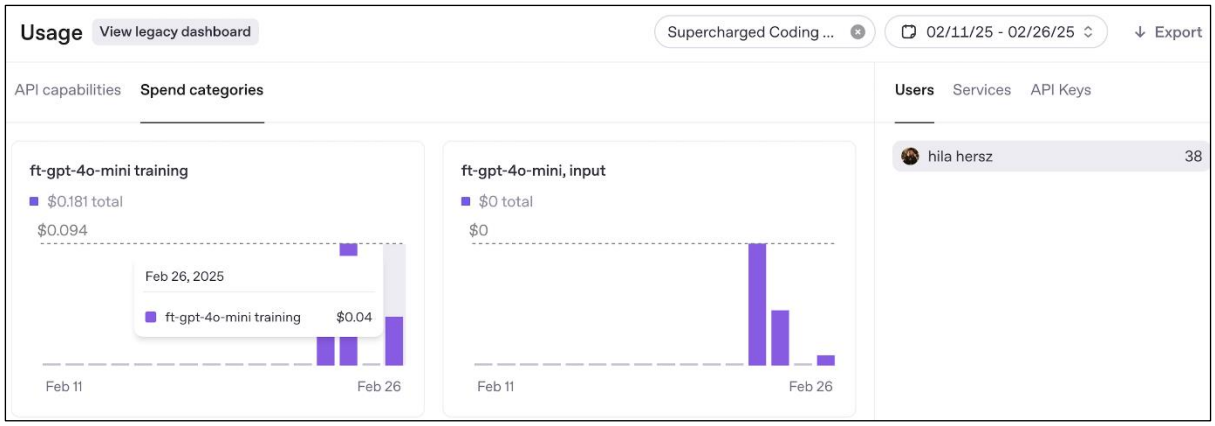
Output:
$1.20 / 1M tokens

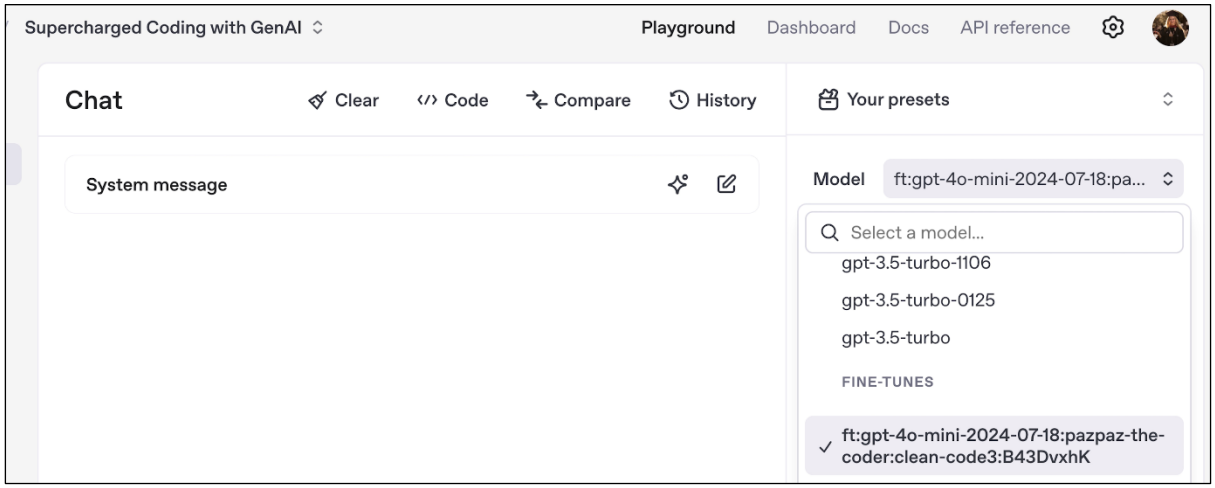
Training:
$3.00 / 1M tokens

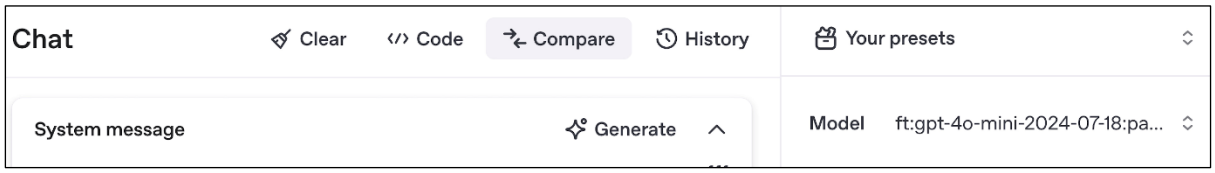

Pazpaz the coder
Supercharged Coding with GenAI
Playground
Dashboard
Docs
API reference
DASHBOARD
Chat Completions
Assistants
Threads
Batches
Evaluations
Fine-tuning
Storage
Usage
API keys
Fine-tuning
All
Successful
Failed
Learn more
+ Create
ft:gpt-4o-mini-2024-07-18:pazpaz-the-coder:clean-code3:B43DvxhK
2/23/2025, 12:03 PM
ft:gpt-4o-2024-08-06:pazpaz-the-coder::AsQxSdQL
1/22/2025, 10:45 AM
MODEL
ft:gpt-4o-mini-2024-07-18:pazpaz-the-coder:clean-code3:B43DvxhK
Status
Succeeded
Job ID
ftjob-Xxc3OqkNxAjsYfgceVj3t8gd
Training Method
Supervised
Suffix
clean-code3
Base model
gpt-4o-mini-2024-07-18
Output model
ft:gpt-4o-mini-2024-07-18:pazpaz-the-cc
Created at
Feb 23, 2025, 12:03 PM
Trained tokens
16,407
Epochs
9

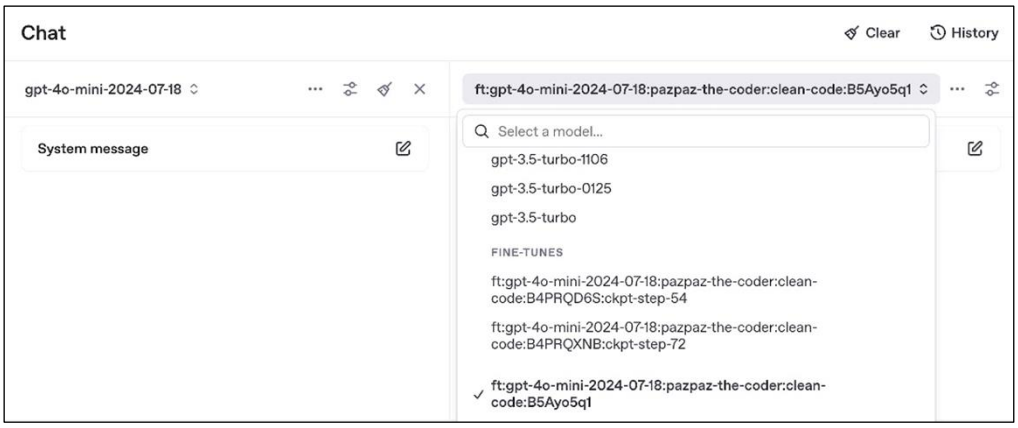

Create a fine-tuned model
Method
Specify the method to be used for fine-tuning.
Supervised
Base Model
gpt-4o-mini-2024-07-18
Training data
Add a jsonl file to use for training. By providing the file, you confirm that you have the rights to use the data.
Upload new
Select existing
Browse files
file-YNXu5pLGGDT5GeC15Lv55G
Validation data
Add a jsonl file to use for validation metrics.
Upload new
Select existing
None
Suffix
Add a custom suffix that will be appended to the output model name.
clean-code
Seed
Learn about fine-tuning
Cancel
Create

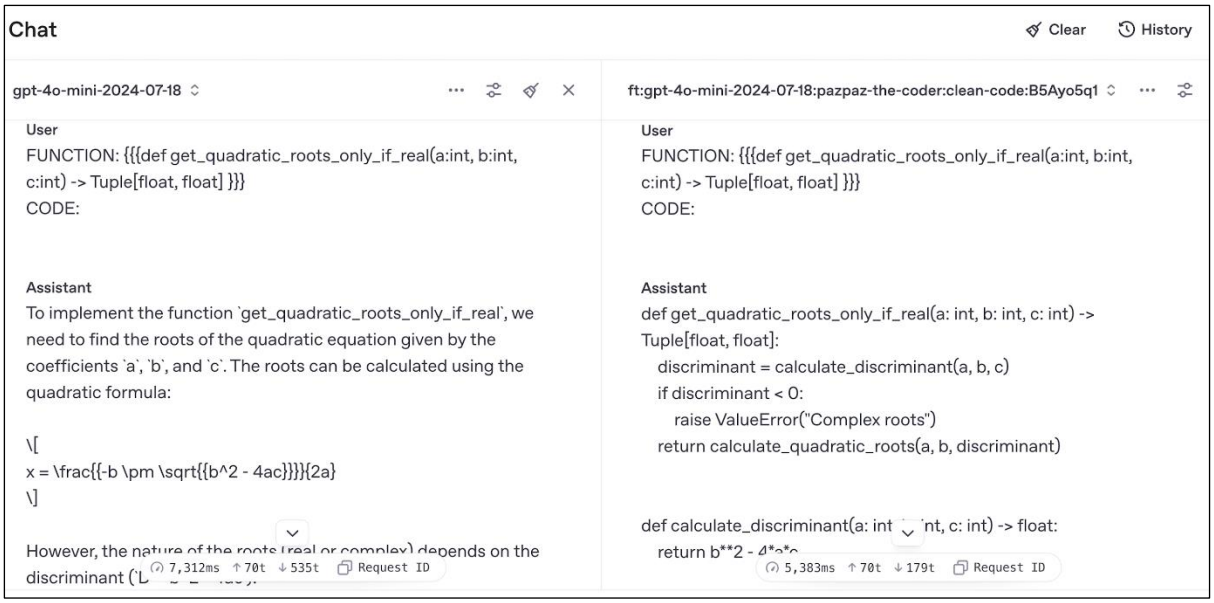

Fine-tuning
All
Successful
Failed
Learn more
+ Create
ft:gpt-4o-mini-2024-07-18:pazpaz-the-coder:clean-code:B5Ayo5q1
2/26/2025, 2:34 PM
ft:gpt-4o-mini-2024-07-18:pazpaz-the-coder:clean-code:B4PRRwKG
2/24/2025, 11:43 AM
ft:gpt-4o-mini-2024-07-18:pazpaz-the-coder:clean-code:B4OJINzq
2/24/2025, 10:35 AM
ft:gpt-4o-mini-2024-07-18:pazpaz-the-coder:clean-code3:B43DvxhK
2/23/2025, 12:03 PM
ft:gpt-4o-mini-2024-07-18:pazpaz-the-coder:decorator-style:AydMPYCg
2/8/2025, 1:24 PM
MODEL
ft:gpt-4o-mini-2024-07-18:pazpaz-the-coder:clean-code:B5Ayo5q1
Status
Succeeded
Job ID
ftjob-06U5Wk2ufL0EU146oPWVLJi5
Training Method
Supervised
Suffix
clean-code
Base model
gpt-4o-mini-2024-07-18
Output model
ft:gpt-4o-mini-2024-07-18:pazpaz-the-cc
Created at
Feb 26, 2025, 2:34 PM
Trained tokens
12,528


Usage
View legacy dashboard
Supercharged Coding ...
02/11/25 - 02/26/25
Export
API capabilities
Spend categories
Users
Services
API Keys
ft-gpt-4o-mini training
$0.181 total
$0.094
Feb 26, 2025
ft-gpt-4o-mini training $0.04
Feb 11
Feb 26
ft-gpt-4o-mini, input
$0 total
$0
Feb 11
Feb 26
hila hersz
38

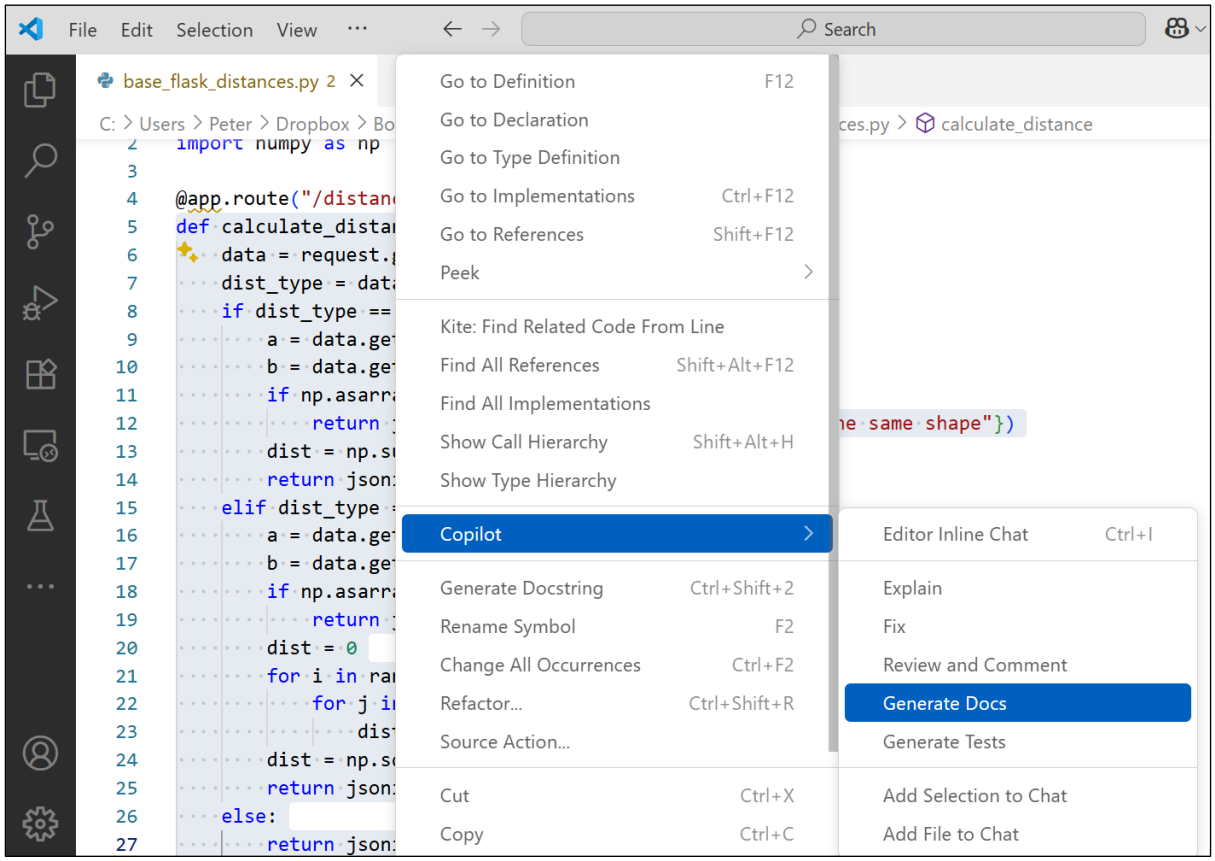

Supercharged Coding with GenAI
Playground
Dashboard
Docs
API reference
Chat
Clear
Code
Compare
History
Your presets
System message
Model
ft:gpt-4o-mini-2024-07-18:pa...
Select a model...
gpt-3.5-turbo-1106
gpt-3.5-turbo-0125
gpt-3.5-turbo
FINE-TUNES
ft:gpt-4o-mini-2024-07-18:pazpaz-the-coder:clean-code3:B43DvxhK


Chat
Clear
Code
Compare
History
Your presets
System message
Generate
Model
ft:gpt-4o-mini-2024-07-18:pa...


Chat
Clear
History
gpt-4o-mini-2024-07-18
ft:gpt-4o-mini-2024-07-18:pazpaz-the-coder:clean-code:B5Ayo5q1
System message
Select a model...
gpt-3.5-turbo-1106
gpt-3.5-turbo-0125
gpt-3.5-turbo
FINE-TUNES
ft:gpt-4o-mini-2024-07-18:pazpaz-the-coder:clean-code:B4PRQD6S:ckpt-step-54
ft:gpt-4o-mini-2024-07-18:pazpaz-the-coder:clean-code:B4PRQXNB:ckpt-step-72
ft:gpt-4o-mini-2024-07-18:pazpaz-the-coder:clean-code:B5Ayo5q1


Chat
Clear
History
gpt-4o-mini-2024-07-18
User
FUNCTION: {{{def get_quadratic_roots_only_if_real(a:int, b:int, c:int) -> Tuple[float, float] }}}
CODE:
Assistant
To implement the function `get_quadratic_roots_only_if_real`, we need to find the roots of the quadratic equation given by the coefficients `a`, `b`, and `c`. The roots can be calculated using the quadratic formula:
\[
x = \frac{{-b \pm \sqrt{{b^2 - 4ac}}}}{2a}
\]
However, the nature of the roots (real or complex) depends on the
7,312ms ↑70t ↓535t Request ID
ft:gpt-4o-mini-2024-07-18:pazpaz-the-coder:clean-code:B5Ayo5q1
User
FUNCTION: {{{def get_quadratic_roots_only_if_real(a:int, b:int, c:int) -> Tuple[float, float] }}}
CODE:
Assistant
def get_quadratic_roots_only_if_real(a: int, b: int, c: int) -> Tuple[float, float]:
discriminant = calculate_discriminant(a, b, c)
if discriminant < 0:
raise ValueError("Complex roots")
return calculate_quadratic_roots(a, b, discriminant)
def calculate_discriminant(a: int
nt, c: int) -> float:
return b**2 - 4*a*c
5,383ms ↑70t ↓179t Request ID

Chapter 12: Documenting Code with GenAI



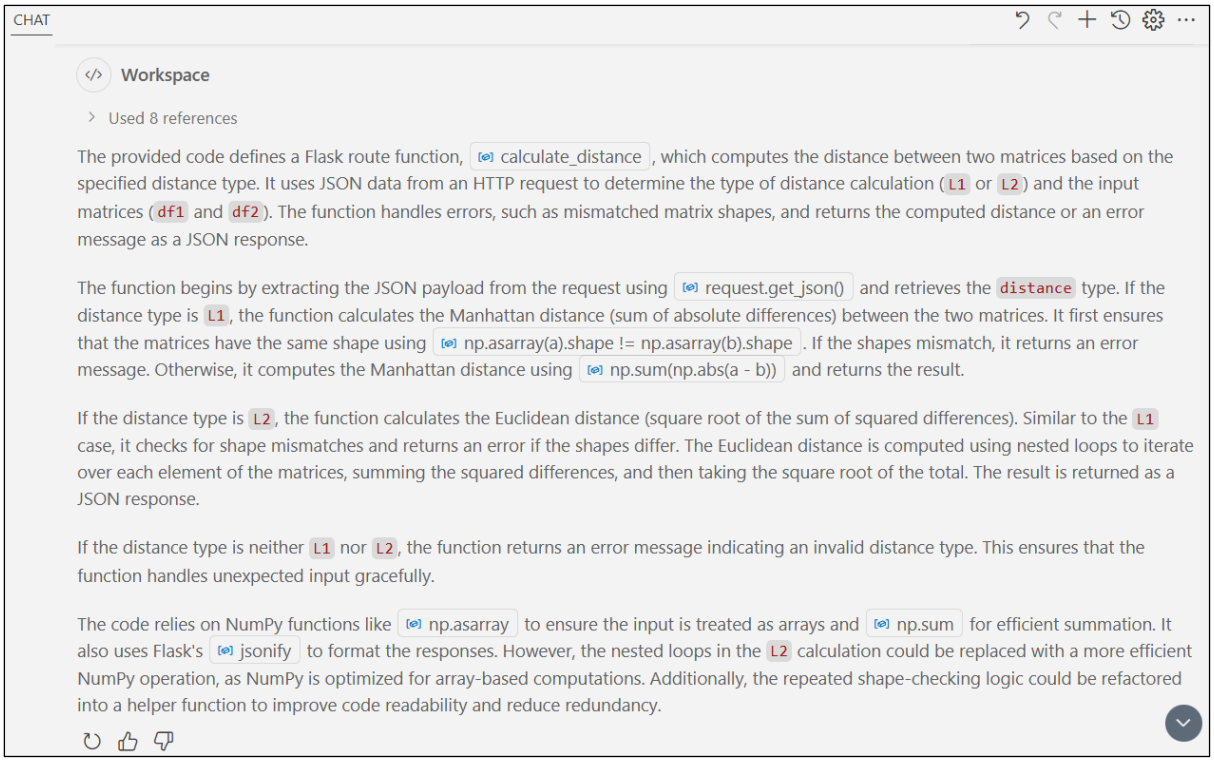

```python
@app.route("/distances", methods=["POST"])
def calculate_distance():
    """
    Calculate the distance between two matrices based on the specified distance type.

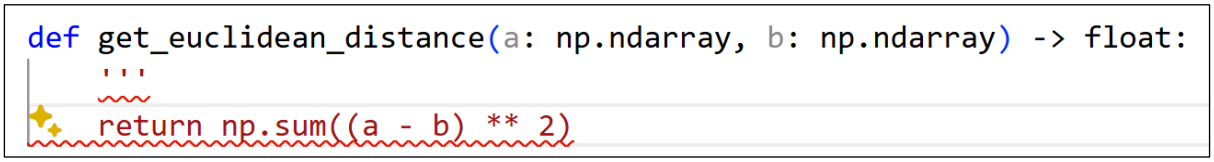
    This function expects a JSON payload with the following structure:
    {
        "distance": "L1" or "L2",  # Type of distance metric to use
        "df1": [[...], [...]],     # First matrix (list of lists)
        "df2": [[...], [...]]      # Second matrix (list of lists)
    }

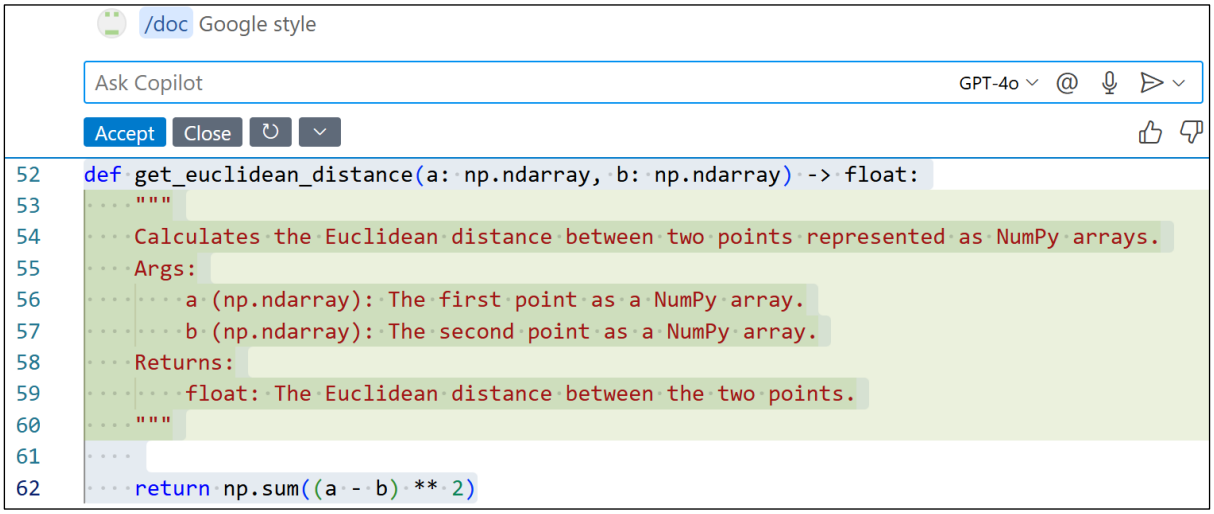
    Returns:
        - A JSON response containing the calculated distance if the input is valid.
        - A JSON error message if the matrices have different shapes or if the distance type is invalid.

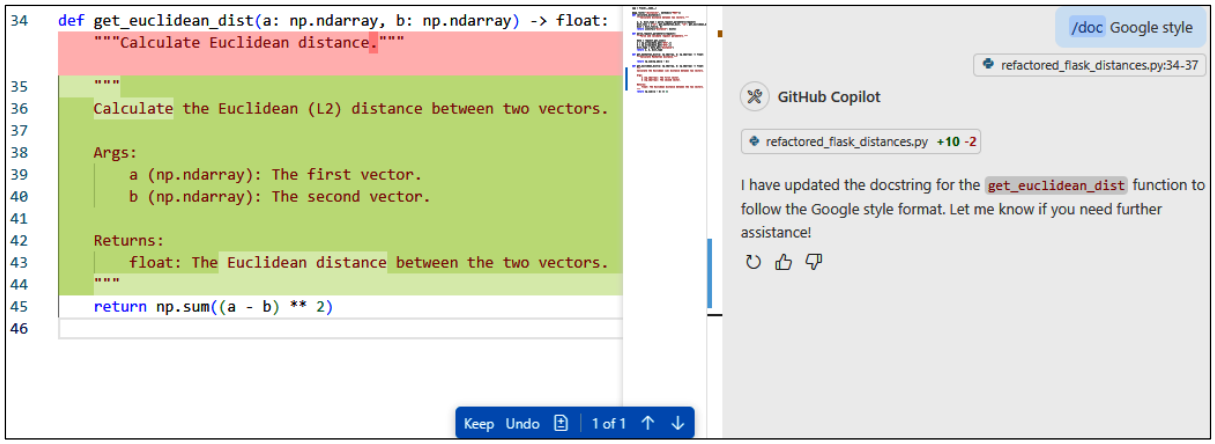
    Distance types:
        - "L1": Manhattan distance (sum of absolute differences).
        - "L2": Euclidean distance (square root of the sum of squared differences).

    Error cases:
        - If the matrices `df1` and `df2` do not have the same shape, an error message is returned.
        - If the `distance` type is not "L1" or "L2", an error message is returned.
    """
```

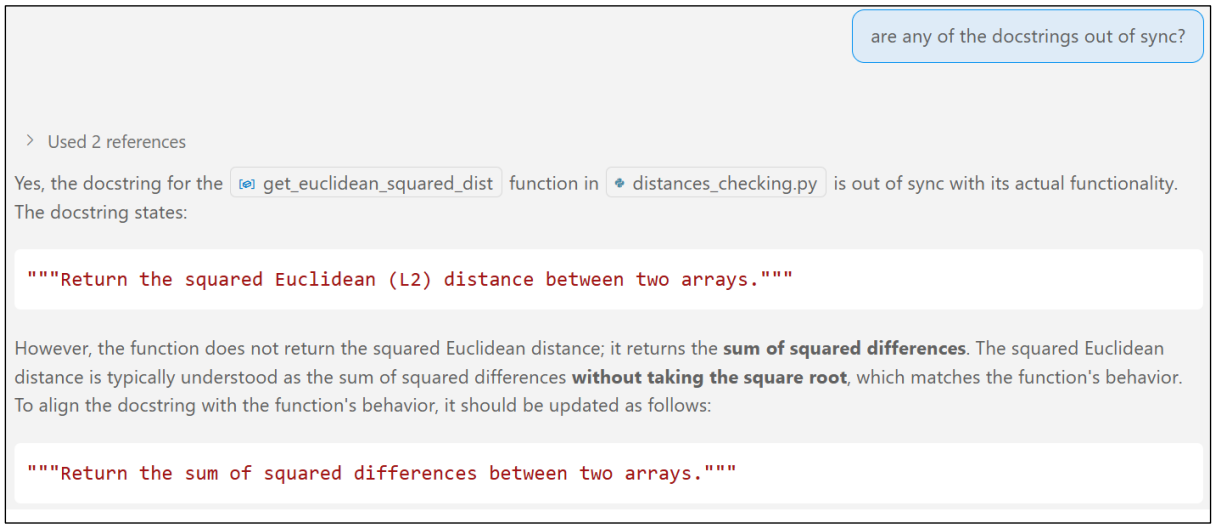

CHAT
Workspace
Used 8 references
The provided code defines a Flask route function, calculate_distance, which computes the distance between two matrices based on the specified distance type. It uses JSON data from an HTTP request to determine the type of distance calculation (L1 or L2) and the input matrices (df1 and df2). The function handles errors, such as mismatched matrix shapes, and returns the computed distance or an error message as a JSON response.
The function begins by extracting the JSON payload from the request using request.get_json() and retrieves the distance type. If the distance type is L1, the function calculates the Manhattan distance (sum of absolute differences) between the two matrices. It first ensures that the matrices have the same shape using np.asarray(a).shape != np.asarray(b).shape. If the shapes mismatch, it returns an error message. Otherwise, it computes the Manhattan distance using np.sum(np.abs(a - b)) and returns the result.
If the distance type is L2, the function calculates the Euclidean distance (square root of the sum of squared differences). Similar to the L1 case, it checks for shape mismatches and returns an error if the shapes differ. The Euclidean distance is computed using nested loops to iterate over each element of the matrices, summing the squared differences, and then taking the square root of the total. The result is returned as a JSON response.
If the distance type is neither L1 nor L2, the function returns an error message indicating an invalid distance type. This ensures that the function handles unexpected input gracefully.
The code relies on NumPy functions like np.asarray to ensure the input is treated as arrays and np.sum for efficient summation. It also uses Flask's jsonify to format the responses. However, the nested loops in the L2 calculation could be replaced with a more efficient NumPy operation, as NumPy is optimized for array-based computations. Additionally, the repeated shape-checking logic could be refactored into a helper function to improve code readability and reduce redundancy.


def get_euclidean_distance(a: np.ndarray, b: np.ndarray) -> float:
'''
return np.sum((a - b) ** 2)

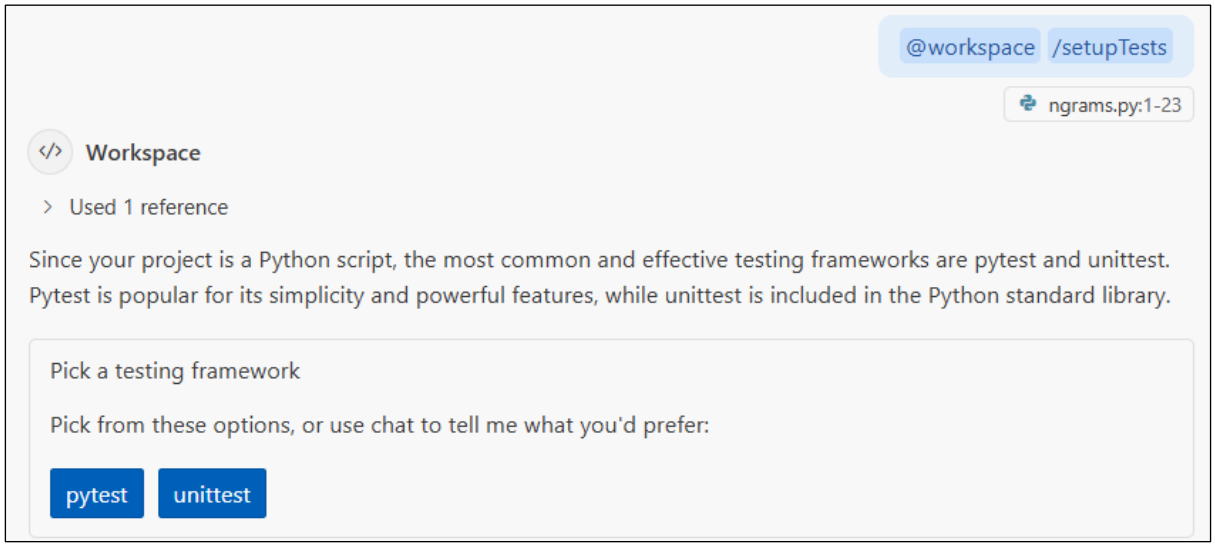

/doc Google style
Ask Copilot
GPT-4o
Accept
Close
52 def get_euclidean_distance(a: np.ndarray, b: np.ndarray) -> float:
53 """
54 Calculates the Euclidean distance between two points represented as NumPy arrays.
55 Args:
56 a (np.ndarray): The first point as a NumPy array.
57 b (np.ndarray): The second point as a NumPy array.
58 Returns:
59 float: The Euclidean distance between the two points.
60 """
61
62 return np.sum((a - b) ** 2)

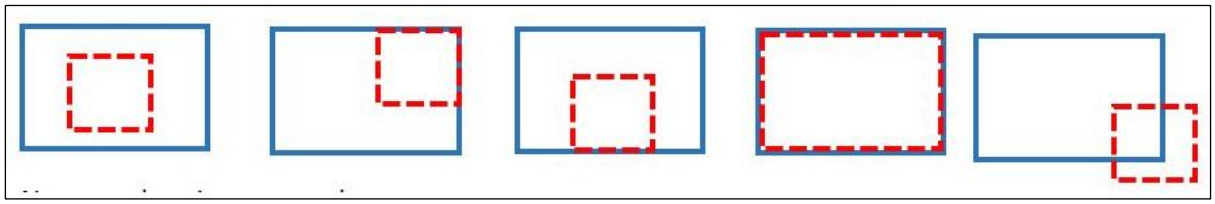

34 def get_euclidean_dist(a: np.ndarray, b: np.ndarray) -> float:
"""Calculate Euclidean distance."""
"""
Calculate the Euclidean (L2) distance between two vectors.
Args:
a (np.ndarray): The first vector.
b (np.ndarray): The second vector.
Returns:
float: The Euclidean distance between the two vectors.
"""
return np.sum((a - b) ** 2)
Keep Undo 1 of 1
/doc Google style
refactored_flask_distances.py:34-37
GitHub Copilot
refactored_flask_distances.py +10 -2
I have updated the docstring for the get_euclidean_dist function to follow the Google style format. Let me know if you need further assistance!

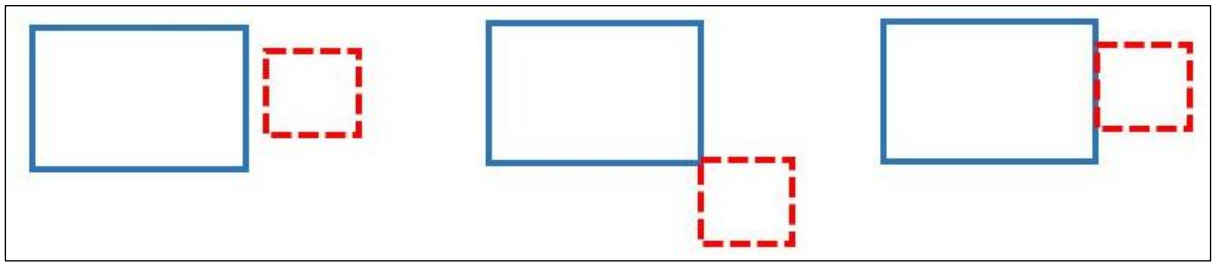

write all docstrings for my current file
GitHub Copilot
Please provide the code from your current file, or upload the file, so I can write docstrings for all the functions and methods it contains.

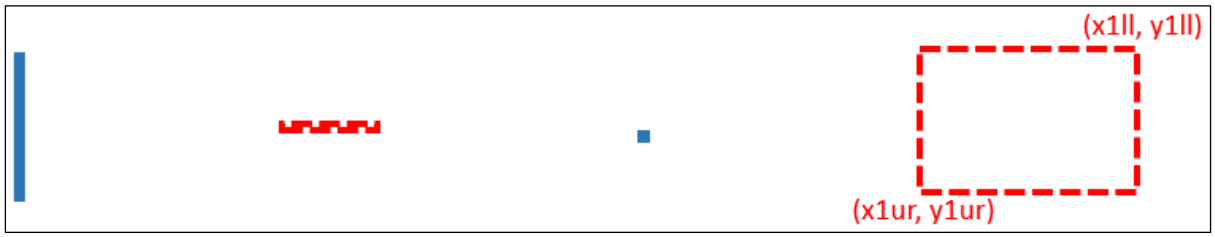

are any of the docstrings out of sync?
Used 2 references
Yes, the docstring for the get_euclidean_squared_dist function in distances_checking.py is out of sync with its actual functionality. The docstring states:
"""Return the squared Euclidean (L2) distance between two arrays."""
However, the function does not return the squared Euclidean distance; it returns the sum of squared differences. The squared Euclidean distance is typically understood as the sum of squared differences without taking the square root, which matches the function's behavior. To align the docstring with the function's behavior, it should be updated as follows:
"""Return the sum of squared differences between two arrays."""

Chapter 13: Writing and Maintaining Unit Tests









GitHub Copilot

ID	First Name	Last Name	Address	City	Country	Disease	Age
10234	Alice	Smith	123 Maple St.	Boston	USA	Influenza	34
20458	Brian	Johnson	456 Oak Dr.	Toronto	Canada	Diabetes	45
31567	Carla	Lee	789 Pine Ave.	London	UK	Hypertension	52
42689	David	Kim	321 Cedar St.	Sydney	Australia	Asthma	28
53790	Emma	Patel	654 Birch Dr.	Mumbai	India	Malaria	41
64821	Frank	Müller	987 Elm Ave.	Berlin	Germany	Arthritis	60
75932	Grace	Rossi	159 Spruce St.	Rome	Italy	COVID-19	37
86043	Henry	Dubois	753 Willow Dr.	Paris	France	Tuberculosis	50
97154	Irene	Chen	852 Aspen Ave.	Beijing	China	Hepatitis	29
18265	Jack	Lopez	951 Walnut St.	Madrid	Spain	Dengue	33

Here's the FASTA-formatted file with 3 randomly generated DNA sequences, each 15–25 nucleotides long:

shell Copy Edit

```
>Sequence_1
GAAATTACCGACGGCCGGCC
>Sequence_2
TGAAAGGACCCGCTCCTTA
>Sequence_3
TTCAGACGGTCTACTT
```

Let me know if you'd like to download this as a `.fasta` file or generate more sequences.

create sample inputs for get_euclidian_distance()

GitHub Copilot

Here are some sample inputs for `get_euclidean_dist()`:

```python
import numpy as np

# Example 1: 1D arrays
a = np.array([1, 2, 3])
b = np.array([4, 5, 6])

# Example 2: 2D arrays
a2 = np.array([[1, 2], [3, 4]])
b2 = np.array([[5, 6], [7, 8]])

# Example 3: Negative values
a3 = np.array([-1, -2, -3])
b3 = np.array([1, 2, 3])
```

You can call `get_euclidean_dist(a, b)`, `get_euclidean_dist(a2, b2)`, or `get_euclidean_dist(a3, b3)` with these inputs.

@workspace /tests generate unit tests for lowercase_remove_punct_numbers

ngrams.py:1-19

Workspace

Used 1 reference

Here is the unit test file for the lowercase_remove_punct_numbers function using the `unittest` framework:

Plan:

1. Import the lowercase_remove_punct_numbers function using an absolute import.
2. Write test cases to cover:
   - Text with punctuation, numbers, and uppercase letters.
   - Text with only lowercase letters.
   - Text with spaces and special characters.
   - Empty string input.

/tests add a test for a 10-digit phone number

Ask Copilot GPT-4o @

Accept Close

```python
    def test_with_phone_number(self):
        text = "Call me at 123-456-7890"
        expected = "call me at "
        self.assertEqual(lowercase_remove_punct_numbers(text), expected)
```

```python
# test create_ngrams with 10-digit phone numbers including area codes
    def test_phone_number(self):
        self.assertEqual(
            create_ngrams("(123) 456-7890", 2),
            ["12", "23", "34", "45", "56", "67", "78", "89", "90"]
        )
```

/tests add a new test for a zip code of the form of five digits - 4 digits

Ask Copilot GPT-4.1

Accept Close

```python
    def test_zip_code(self):
        # Should remove all digits and punctuation, leaving only spaces (if any)
        self.assertEqual(
            create_ngrams("12345-6789", 2),
            []
        )

    def test_longer_ngram(self):
        self.assertEqual(
            create_ngrams("abcdefg", 5),
            ["abcde", "bcdef", "cdefg"]
        )

    def test_ngram_with_spaces(self):
        self.assertEqual(
            create_ngrams("a b c", 3),
            ["a b", " b ", "b c"]
        )
```

```python
def test_lowercase_remove_punct_numbers_with_symbols(self):
    self.assertEqual(
        lowercase_remove_punct_numbers("Te$ting #punctu@ation &numbers 456"),
        "testing punctuation numbers "
    )
```

PROBLEMS 2 OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
....F..
======================================================================
FAIL: test_lowercase_remove_punct_numbers_with_symbols (tests.test_ngrams_chatgpt.TestTextUtils.test_lowerc
ase_remove_punct_numbers_with_symbols)
----------------------------------------------------------------------
Traceback (most recent call last):
  File "C:\Users\PeterH\Documents\Euro 2025\Book Writing\testing chapter\src\tests\test_ngrams_chatgpt.py",
 line 13, in test_lowercase_remove_punct_numbers_with_symbols
    self.assertEqual(
AssertionError: 'teting punctuation numbers ' != 'testing punctuation numbers '
- teting punctuation numbers
+ testing punctuation numbers
?   +
```

Summary of Suggested Additions

Function	Suggested New Tests
`lowercase_remove_punct_numbers`	Empty string, all punctuation, all alpha, mixed whitespace
`multiple_to_single_spaces`	Empty, only whitespace, already spaced
`create_ngrams`	Empty, small/large `n`, edge alignment cases

❌ Failed Test 1: `test_lowercase_remove_punct_numbers_whitespace_mix`

Why it failed:

The function `lowercase_remove_punct_numbers` preserves tabs (`\t`) and newlines (`\n`) as whitespace, but your test expected them to be replaced with regular spaces.

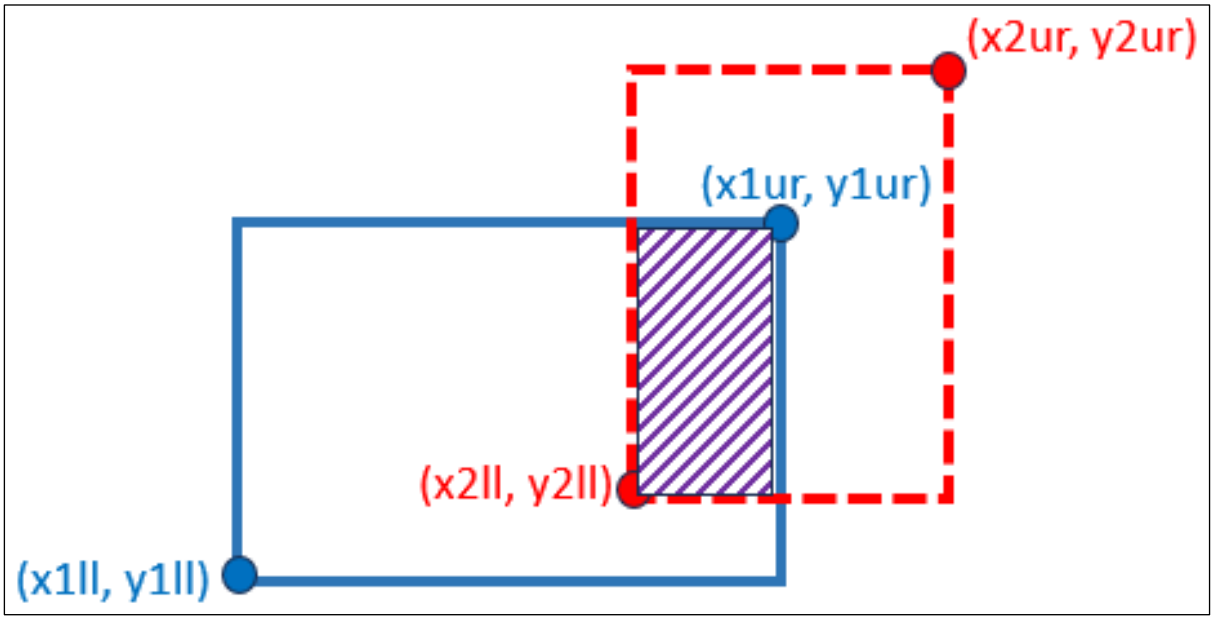
Fix:

The `re.sub(r'[^a-z\s]', '', text.lower())` pattern retains all whitespace characters (`\s` includes tab, newline, space). So the actual return includes `\t` and `\n`.

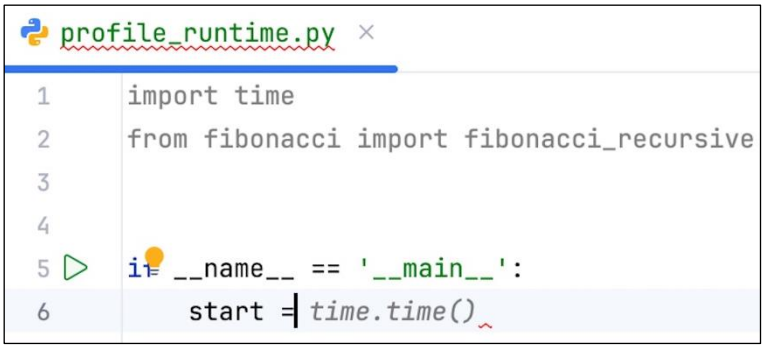
✅ **Update the test to reflect actual behavior** or modify the function to normalize whitespace. Here's the adjusted test:

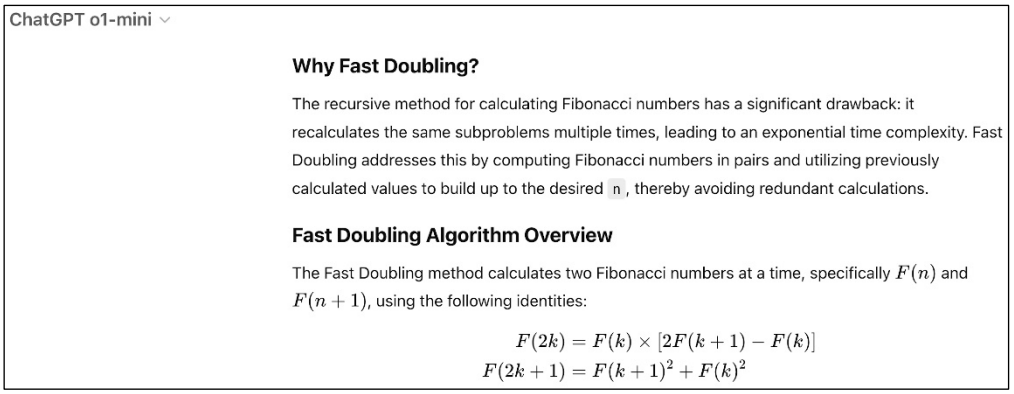
python Copy Edit

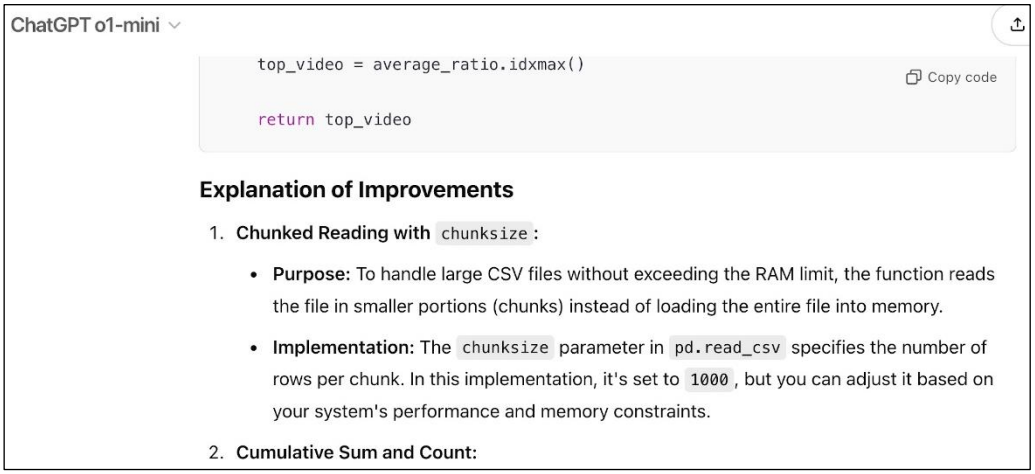
```python
def test_lowercase_remove_punct_numbers_whitespace_mix(self):
    self.assertEqual(lowercase_remove_punct_numbers("ABC\tDEF\nGHI"), "abc\tdef\nghi")
```



Chapter 14: GenAI for Runtime and Memory Management







```
Line #    Mem usage    Increment  Occurrences   Line Contents
=============================================================
     5     93.1 MiB     93.1 MiB           1   @profile
     6                                         def find_top_watch_ratio_video(path):
     7     93.1 MiB      0.0 MiB           1       cumulative_sum = None
     8     93.1 MiB      0.0 MiB           1       cumulative_count = None
     9
    10     93.1 MiB      0.0 MiB           1       chunksize = 1000
    11
    12     93.1 MiB      0.0 MiB           1       i=0
    13   4326.1 MiB -25998.3 MiB         101       for chunk in pd.read_csv(path, index_col=0, chunksize=chunksize):
    14   4326.1 MiB  -9607.8 MiB         100           print(f"chunk {i}")
    15   4326.1 MiB  -9607.9 MiB         100           i+=1
    16   4420.9 MiB    162.9 MiB         100           chunk_sum = chunk.sum(skipna=True)
    17   4515.5 MiB    131.7 MiB         100           chunk_count = chunk.count()
    18
    19   4515.5 MiB  -9323.6 MiB         100           if cumulative_sum is None:
    20   2221.3 MiB      0.0 MiB           1               cumulative_sum = chunk_sum
    21   2221.3 MiB      0.0 MiB           1               cumulative_count = chunk_count
    22                                                 else:
    23   4515.5 MiB  -9303.6 MiB          99               cumulative_sum += chunk_sum
    24   4515.5 MiB  -9307.2 MiB          99               cumulative_count += chunk_count
    25
    26   2657.0 MiB  -1669.2 MiB           1       if cumulative_sum is None or cumulative_count is None:
    27                                                 raise ValueError("The provided CSV file is empty or only contains NaN values.")
    28
    29   2657.0 MiB      0.0 MiB           1       average_ratio = cumulative_sum / cumulative_count
    30
    31   2657.0 MiB      0.1 MiB           1       top_video = average_ratio.idxmax()
    32
    33   2657.0 MiB      0.0 MiB           1       return top_video
```

hilahersz

print the runtime of fibonacci_recursive for n=35

GitHub Copilot

> 4 steps completed successfully

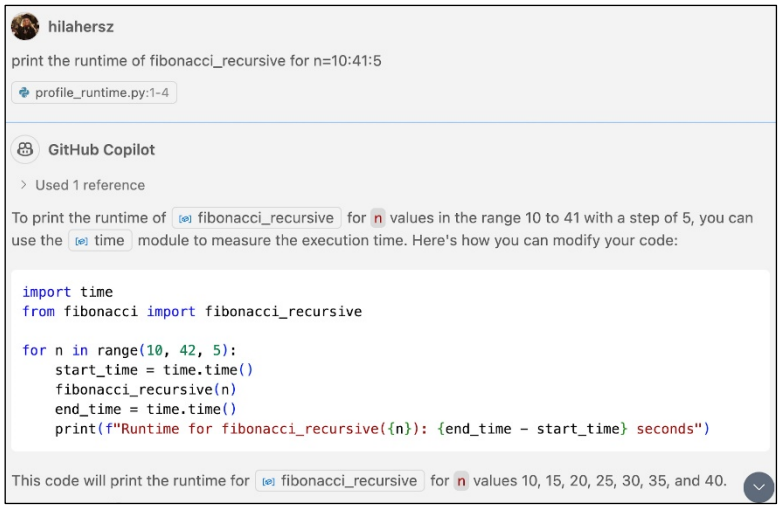
To print the runtime of `fibonacci_recursive` for `n=35`, you can use the `time` module to measure the start and end times of the function call. Here is the complete code:

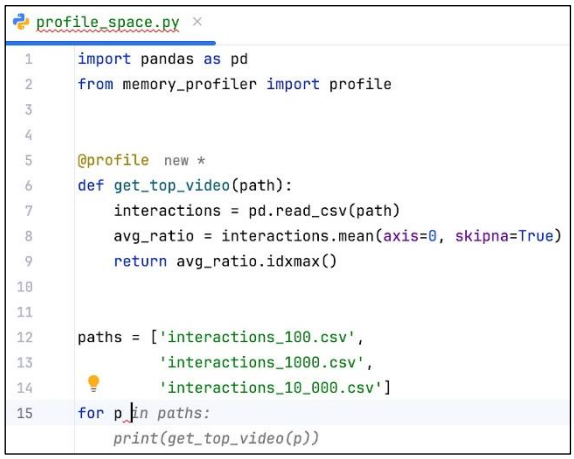
```python
import time
from fibonacci import fibonacci_recursive

n = 35
start_time = time.time()
result = fibonacci_recursive(n)
end_time = time.time()

print(f"Result: {result}")
print(f"Runtime: {end_time - start_time} seconds")
```

```
Line #    Mem usage    Increment  Occurrences   Line Contents
=============================================================
     5    130.7 MiB    130.7 MiB           1   @profile
     6                                         def get_top_video(path):
     7   1020.4 MiB    889.7 MiB           1       interactions = pd.read_csv(path)
     8   1113.4 MiB     93.0 MiB           1       avg_ratio = interactions.mean(axis=0, skipna=True)
     9   1113.4 MiB      0.0 MiB           1       return avg_ratio.idxmax()
```






```
Filename: /Users/hila/PycharmProjects/private/supercharge/ch14/profile_space.py

Line #    Mem usage    Increment  Occurrences   Line Contents
=============================================================
     5     93.2 MiB     93.2 MiB           1   @profile
     6                                         def get_top_video(path):
     7     93.8 MiB      0.6 MiB           1       interactions = pd.read_csv(path)
     8     94.0 MiB      0.2 MiB           1       avg_ratio = interactions.mean(axis=0, skipna=True)
     9     94.0 MiB      0.0 MiB           1       return avg_ratio.idxmax()


top video:  video_82
Filename: /Users/hila/PycharmProjects/private/supercharge/ch14/profile_space.py

Line #    Mem usage    Increment  Occurrences   Line Contents
=============================================================
     5     94.0 MiB     94.0 MiB           1   @profile
     6                                         def get_top_video(path):
     7    124.0 MiB     29.9 MiB           1       interactions = pd.read_csv(path)
     8    131.6 MiB      7.6 MiB           1       avg_ratio = interactions.mean(axis=0, skipna=True)
     9    131.6 MiB      0.0 MiB           1       return avg_ratio.idxmax()


top video:  video_629
Filename: /Users/hila/PycharmProjects/private/supercharge/ch14/profile_space.py

Line #    Mem usage    Increment  Occurrences   Line Contents
=============================================================
     5    130.7 MiB    130.7 MiB           1   @profile
     6                                         def get_top_video(path):
     7   1020.4 MiB    889.7 MiB           1       interactions = pd.read_csv(path)
     8   1113.4 MiB     93.0 MiB           1       avg_ratio = interactions.mean(axis=0, skipna=True)
     9   1113.4 MiB      0.0 MiB           1       return avg_ratio.idxmax()


top video:  video_7238
```

Conclusion

- **Maximal Input (`n_max`): 43**
- **Runtime at `n=43`**: Approximately **78.5 seconds** (within the 100-second limit)
- **Runtime at `n=44`**: Approximately **123 seconds** (exceeds the limit)

Therefore, the maximal input `n` that the `fibonacci_recursive` function can handle within **100 seconds** is **43**.

Maximal Input: 43



ChatGPT o1-mini

Share

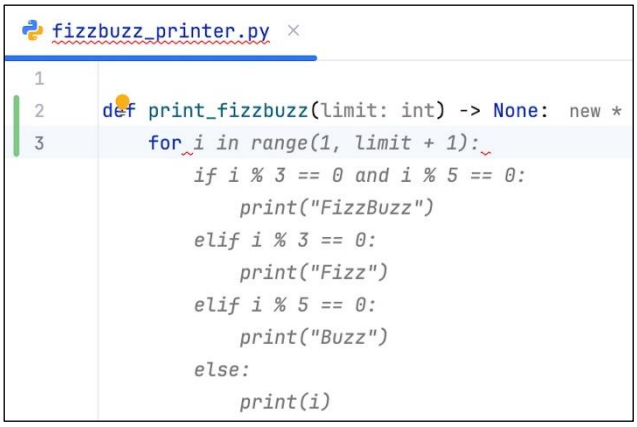
Conclusion

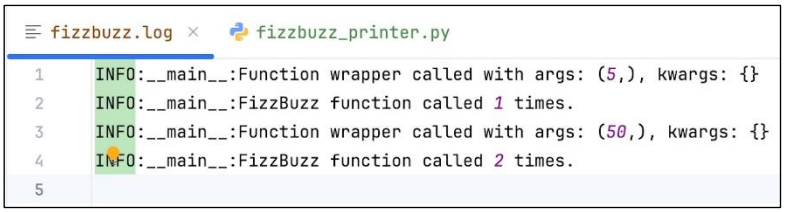
The maximal input size that the `get_top_video` function can handle within a 5 GB RAM limit is approximately 22,000 rows by 22,000 columns.

Note: This estimation assumes a square matrix and linear scaling based on the provided benchmarks. Actual memory usage may vary slightly due to factors like data types, overhead, and system-specific memory management. To ensure stability, it's advisable to use input sizes slightly below the estimated maximum, such as **20,000 rows x 20,000 columns**.

```
Line #    Mem usage    Increment  Occurrences   Line Contents
=============================================================
     5     93.1 MiB     93.1 MiB           1   @profile
     6                                         def get_top_video(path):
     7   3562.4 MiB   3469.3 MiB           1       interactions = pd.read_csv(path)
     8   4643.5 MiB   1081.1 MiB           1       avg_ratio = interactions.mean(axis=0, skipna=True)
     9   4643.7 MiB      0.2 MiB           1       return avg_ratio.idxmax()
```

Chapter 15: Going Live with GenAI: Logging, Monitoring, and Errors

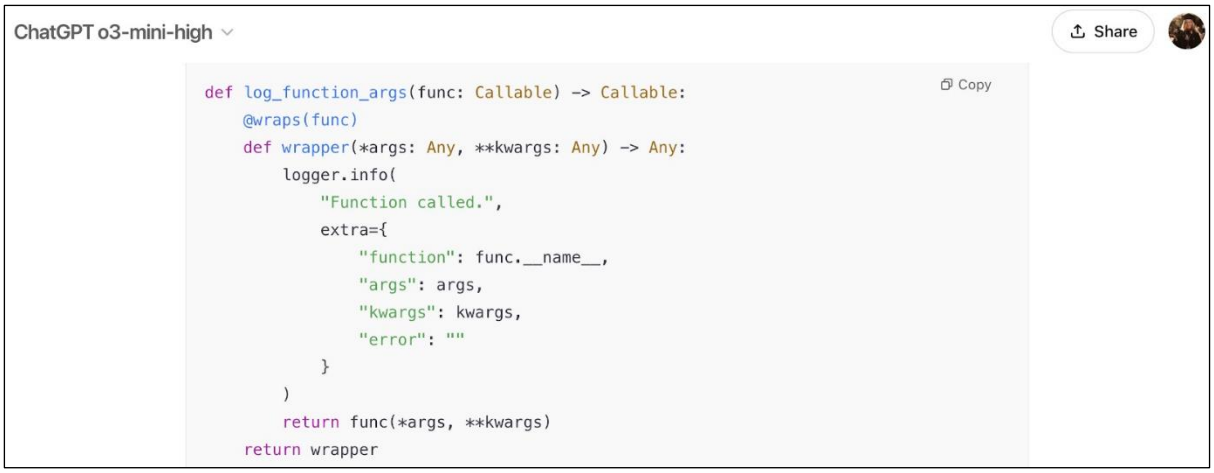


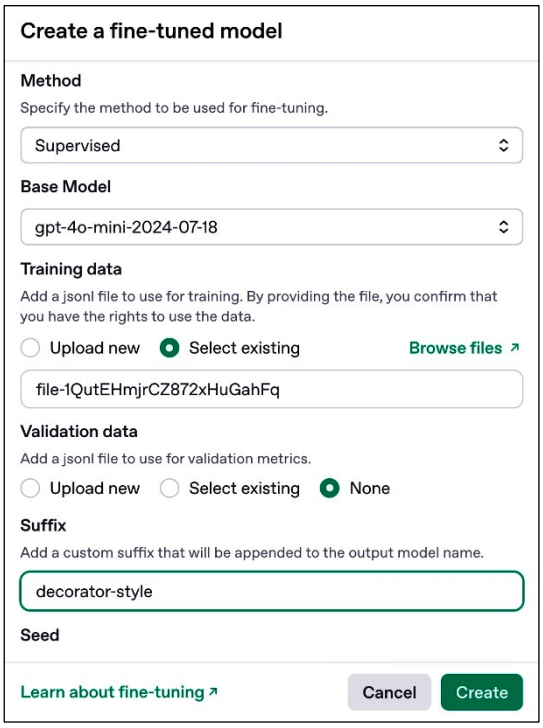


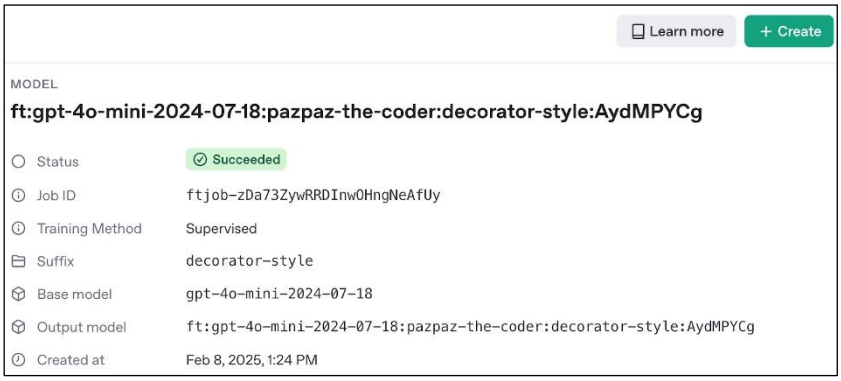

```python
def log_function_args(func: callable) -> callable: new *
    def wrapper(*args, **kwargs):
        message: str = "Function called."
        logger.info(message,
                    extra={"function": func.__name__,
                           "args": args,
                           "kwargs": kwargs,
                           "error": "",
                           "timing": ""})
        return func(*args, **kwargs)

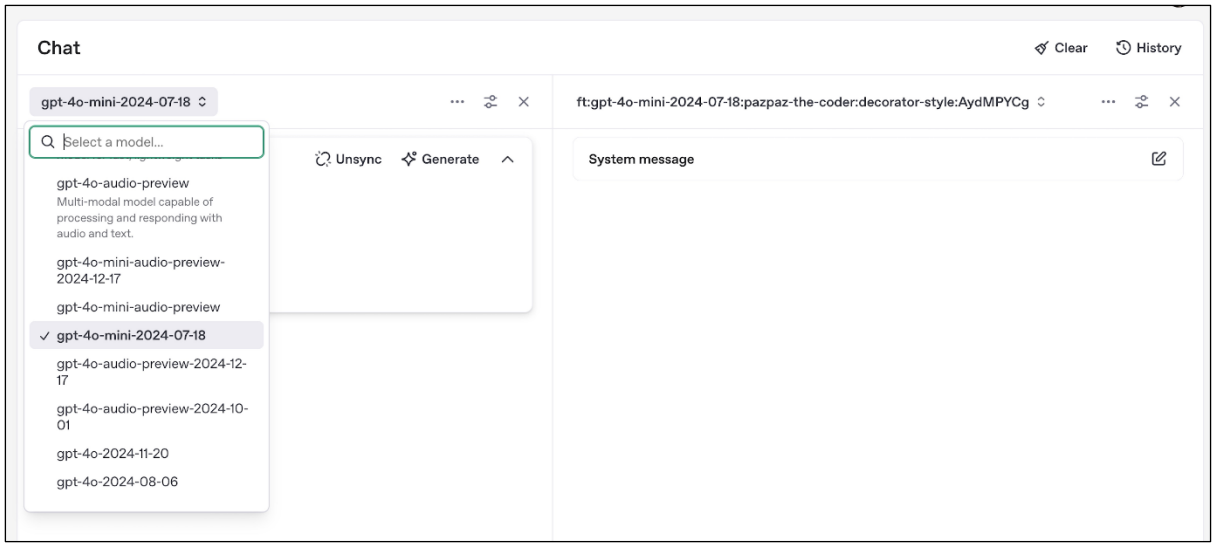
    return wrapper

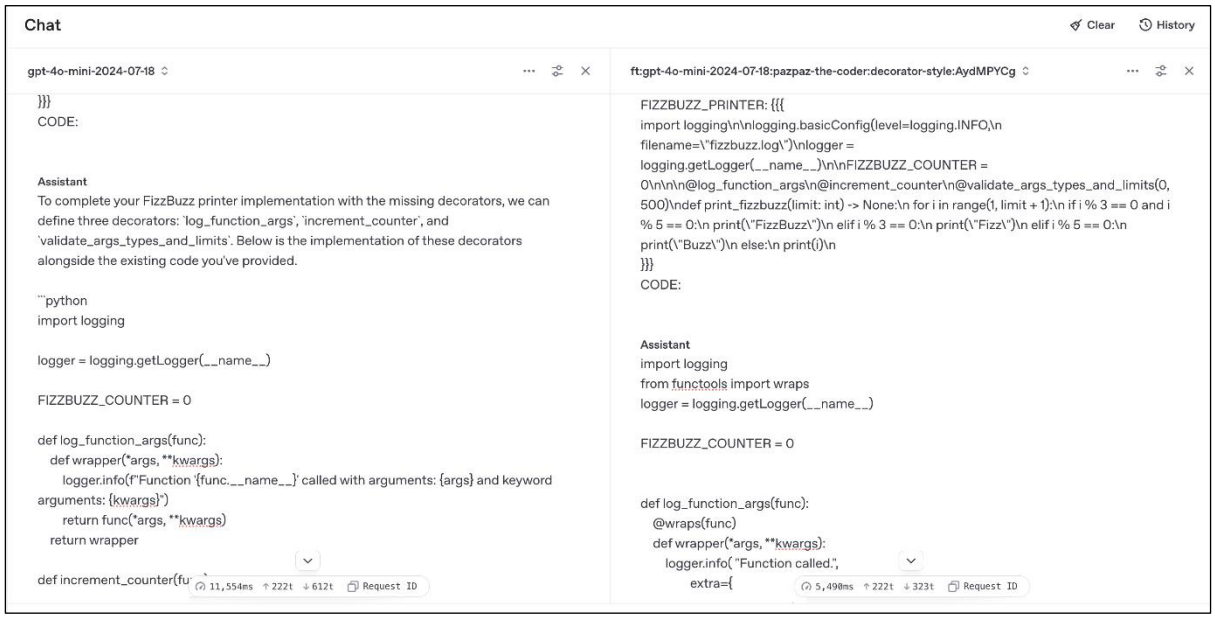

@log_function_args  2 usages  new *
@increment_counter
@validate_args_types_and_limits(0, 500)
def print_fizzbuzz(limit: int) -> None:
```

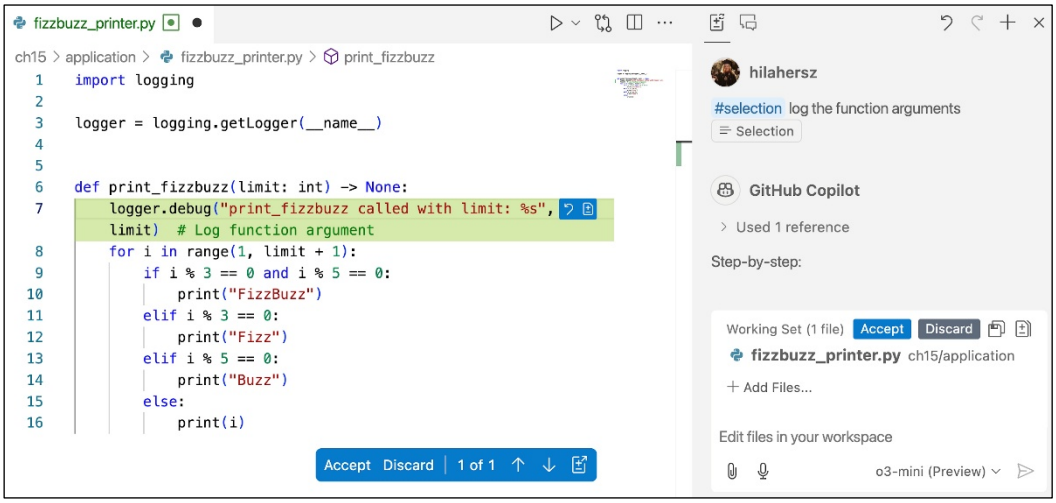

ChatGPT o3-mini-high
Share
Copy
def log_function_args(func: Callable) -> Callable:
@wraps(func)
def wrapper(*args: Any, **kwargs: Any) -> Any:
logger.info(
"Function called.",
extra={
"function": func.__name__,
"args": args,
"kwargs": kwargs,
"error": ""
}
)
return func(*args, **kwargs)
return wrapper

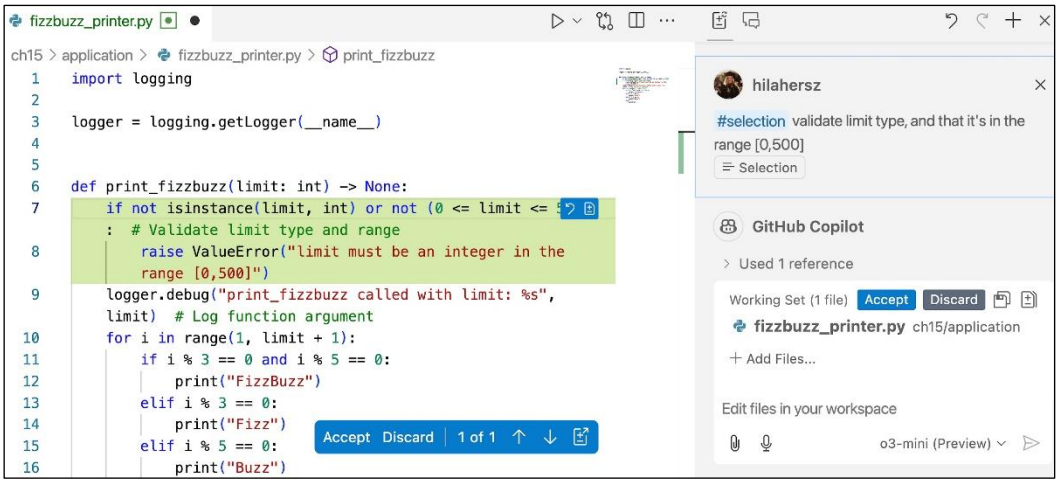

Create a fine-tuned model
Method
Specify the method to be used for fine-tuning.
Supervised
Base Model
gpt-4o-mini-2024-07-18
Training data
Add a jsonl file to use for training. By providing the file, you confirm that you have the rights to use the data.
Upload new
Select existing
Browse files
file-1QutEHmjrCZ872xHuGahFq
Validation data
Add a jsonl file to use for validation metrics.
Upload new
Select existing
None
Suffix
Add a custom suffix that will be appended to the output model name.
decorator-style
Seed
Learn about fine-tuning
Cancel
Create

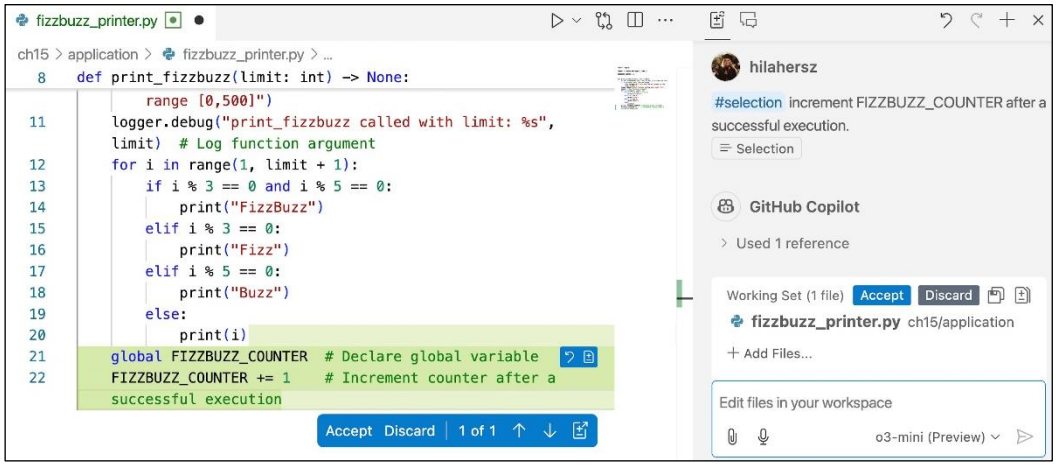

Learn more
+ Create
MODEL
ft:gpt-4o-mini-2024-07-18:pazpaz-the-coder:decorator-style:AydMPYCg
Status
Succeeded
Job ID
ftjob-zDa73ZywRRDInwOHngNeAfUy
Training Method
Supervised
Suffix
decorator-style
Base model
gpt-4o-mini-2024-07-18
Output model
ft:gpt-4o-mini-2024-07-18:pazpaz-the-coder:decorator-style:AydMPYCg
Created at
Feb 8, 2025, 1:24 PM


Chat
Clear
History
gpt-4o-mini-2024-07-18
Select a model...
gpt-4o-audio-preview
Multi-modal model capable of processing and responding with audio and text.
gpt-4o-mini-audio-preview-2024-12-17
gpt-4o-mini-audio-preview
gpt-4o-mini-2024-07-18
gpt-4o-audio-preview-2024-12-17
gpt-4o-audio-preview-2024-10-01
gpt-4o-2024-11-20
gpt-4o-2024-08-06
Unsync
Generate
ft:gpt-4o-mini-2024-07-18:pazpaz-the-coder:decorator-style:AydMPYCg
System message


Chat
Clear
History
gpt-4o-mini-2024-07-18
}}}
CODE:
Assistant
To complete your FizzBuzz printer implementation with the missing decorators, we can define three decorators: `log_function_args`, `increment_counter`, and `validate_args_types_and_limits`. Below is the implementation of these decorators alongside the existing code you've provided.
```python
import logging
logger = logging.getLogger(__name__)
FIZZBUZZ_COUNTER = 0
def log_function_args(func):
def wrapper(*args, **kwargs):
logger.info(f"Function '{func.__name__}' called with arguments: {args} and keyword arguments: {kwargs}")
return func(*args, **kwargs)
return wrapper
def increment_counter(fu
11,554ms ↑222t ↓612t Request ID
ft:gpt-4o-mini-2024-07-18:pazpaz-the-coder:decorator-style:AydMPYCg
FIZZBUZZ_PRINTER: {{{
import logging\n\nlogging.basicConfig(level=logging.INFO,\n filename=\"fizzbuzz.log\")\nlogger = logging.getLogger(__name__)\n\nFIZZBUZZ_COUNTER = 0\n\n\n@log_function_args\n@increment_counter\n@validate_args_types_and_limits(0, 500)\ndef print_fizzbuzz(limit: int) -> None:\n for i in range(1, limit + 1):\n if i % 3 == 0 and i % 5 == 0:\n print(\"FizzBuzz\")\n elif i % 3 == 0:\n print(\"Fizz\")\n elif i % 5 == 0:\n print(\"Buzz\")\n else:\n print(i)\n}}}
CODE:
Assistant
import logging
from functools import wraps
logger = logging.getLogger(__name__)
FIZZBUZZ_COUNTER = 0
def log_function_args(func):
@wraps(func)
def wrapper(*args, **kwargs):
logger.info( "Function called.",
extra={
5,490ms ↑222t ↓323t Request ID


fizzbuzz_printer.py
ch15 > application > fizzbuzz_printer.py > print_fizzbuzz
1 import logging
2
3 logger = logging.getLogger(__name__)
4
5
6 def print_fizzbuzz(limit: int) -> None:
7 logger.debug("print_fizzbuzz called with limit: %s", limit) # Log function argument
8 for i in range(1, limit + 1):
9 if i % 3 == 0 and i % 5 == 0:
10 print("FizzBuzz")
11 elif i % 3 == 0:
12 print("Fizz")
13 elif i % 5 == 0:
14 print("Buzz")
15 else:
16 print(i)
Accept Discard | 1 of 1
hilahersz
#selection log the function arguments
Selection
GitHub Copilot
Used 1 reference
Step-by-step:
Working Set (1 file) Accept Discard
fizzbuzz_printer.py ch15/application
Add Files...
Edit files in your workspace
o3-mini (Preview)
```





get_geometric_mean.py

```python
from typing import Union
import cmath


def get_geometric_mean(*args: float) -> Union[float, complex]:  new *
    product = get_product(*args)
    nth_root = len(args)
    is_positive = is_product_positive(product)
    is_nth_root_odd = is_odd(nth_root)
    is_mean_real = is_geometric_mean_real(is_nth_root_odd, is_positive)
    mean = get_mean(product, nth_root, is_mean_real)

    return mean
```

fizzbuzz_printer.py

```python
import logging

logger = logging.getLogger(__name__)

FIZZBUZZ_COUNTER = 0


@log_function_args  new *
@increment_counter
@validate_args_types_and_limits(0, 500)
def print_fizzbuzz(limit: int) -> None:
```

fizzbuzz_printer.py

```python
import logging

logger = logging.getLogger(__name__)

FIZZBUZZ_COUNTER = 0


def log_function_args(func):  new *
    def wrapper(*args, **kwargs):
        logger.info(f"Function {func.__name__} called with args: {args}, kwargs: {kwargs}")
        return func(*args, **kwargs)

    return wrapper


@log_function_args  new *
@increment_counter
@validate_args_types_and_limits(0, 500)
def print_fizzbuzz(limit: int) -> None:
    for i in range(1, limit + 1):
```

fizzbuzz_printer.py

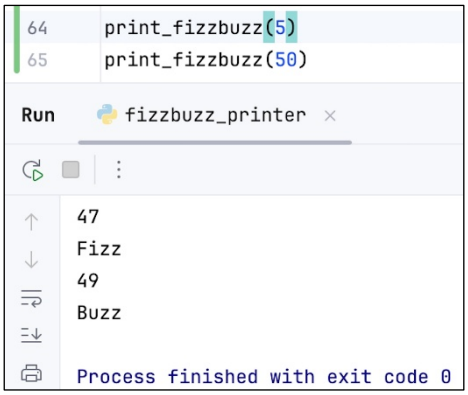
```python
FIZZBUZZ_COUNTER = 0


def log_function_args(func):  1 usage  new *
    def wrapper(*args, **kwargs):  new *
        logger.info(f"Function {func.__name__} called with args: {args}, kwargs: {kwargs}")
        return func(*args, **kwargs)

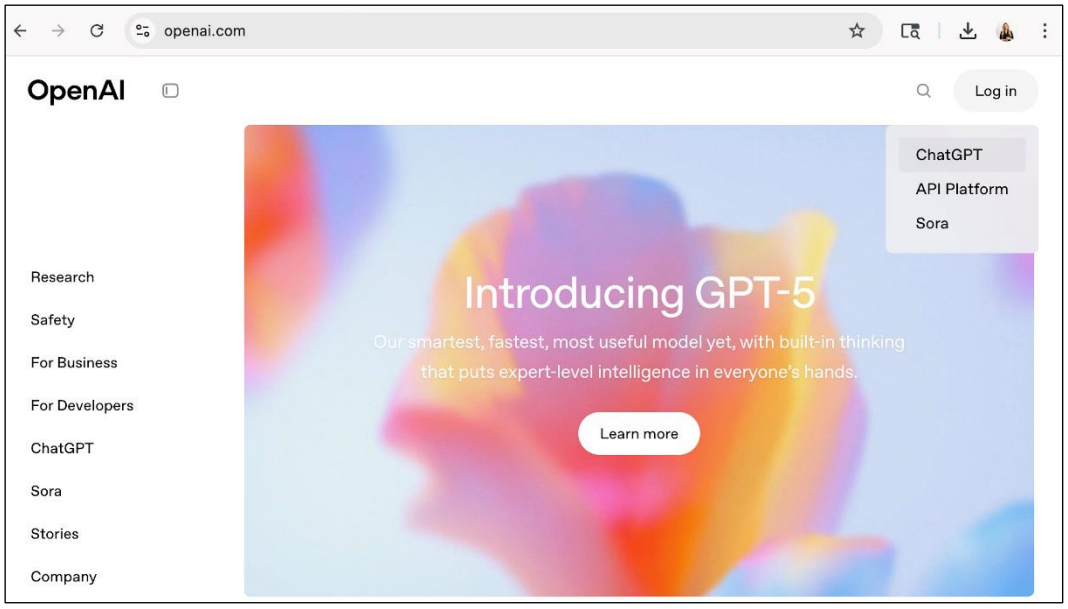
    return wrapper

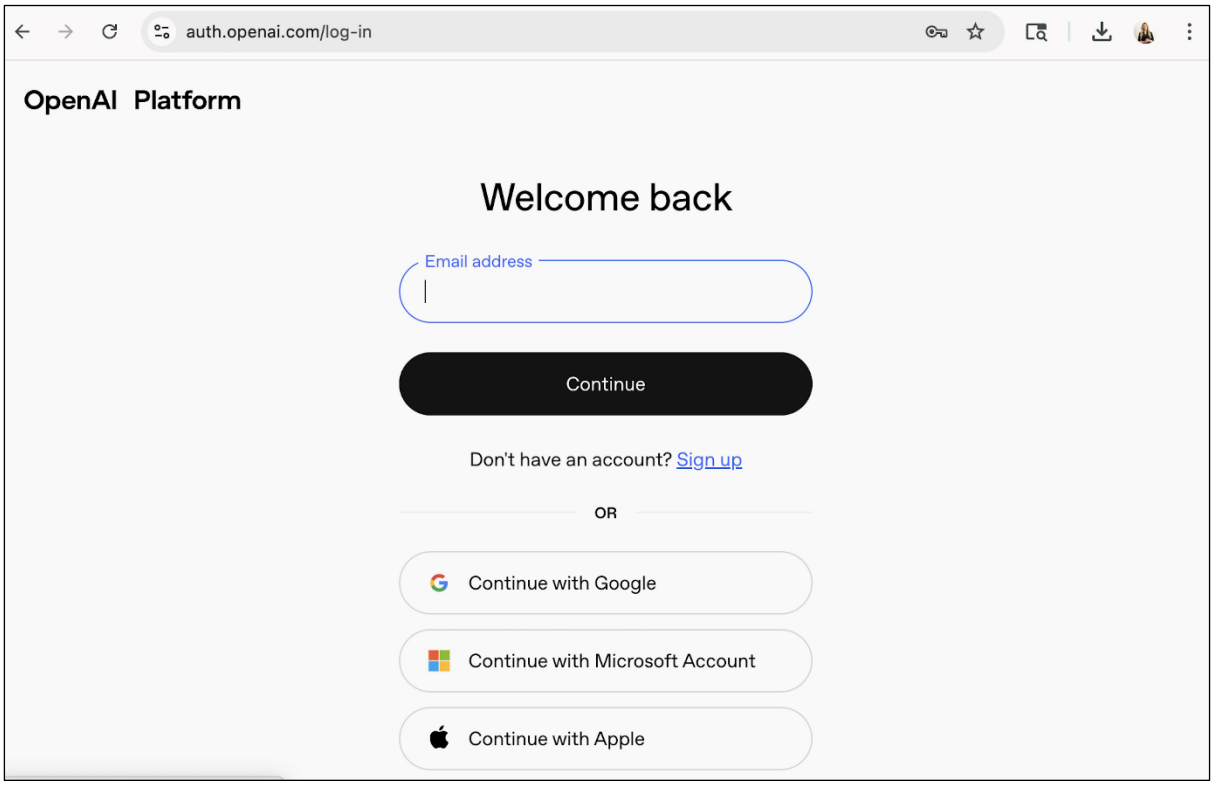

def increment_counter(func):


@log_function_args  new *
@increment_counter
@validate_args_types_and_limits(0, 500)
def print_fizzbuzz(limit: int) -> None:
    for i in range(1, limit + 1):
```


64 print_fizzbuzz(5)
65 print_fizzbuzz(50)
Run
fizzbuzz_printer
47
Fizz
49
Buzz
Process finished with exit code 0

Chapter 17: Appendix



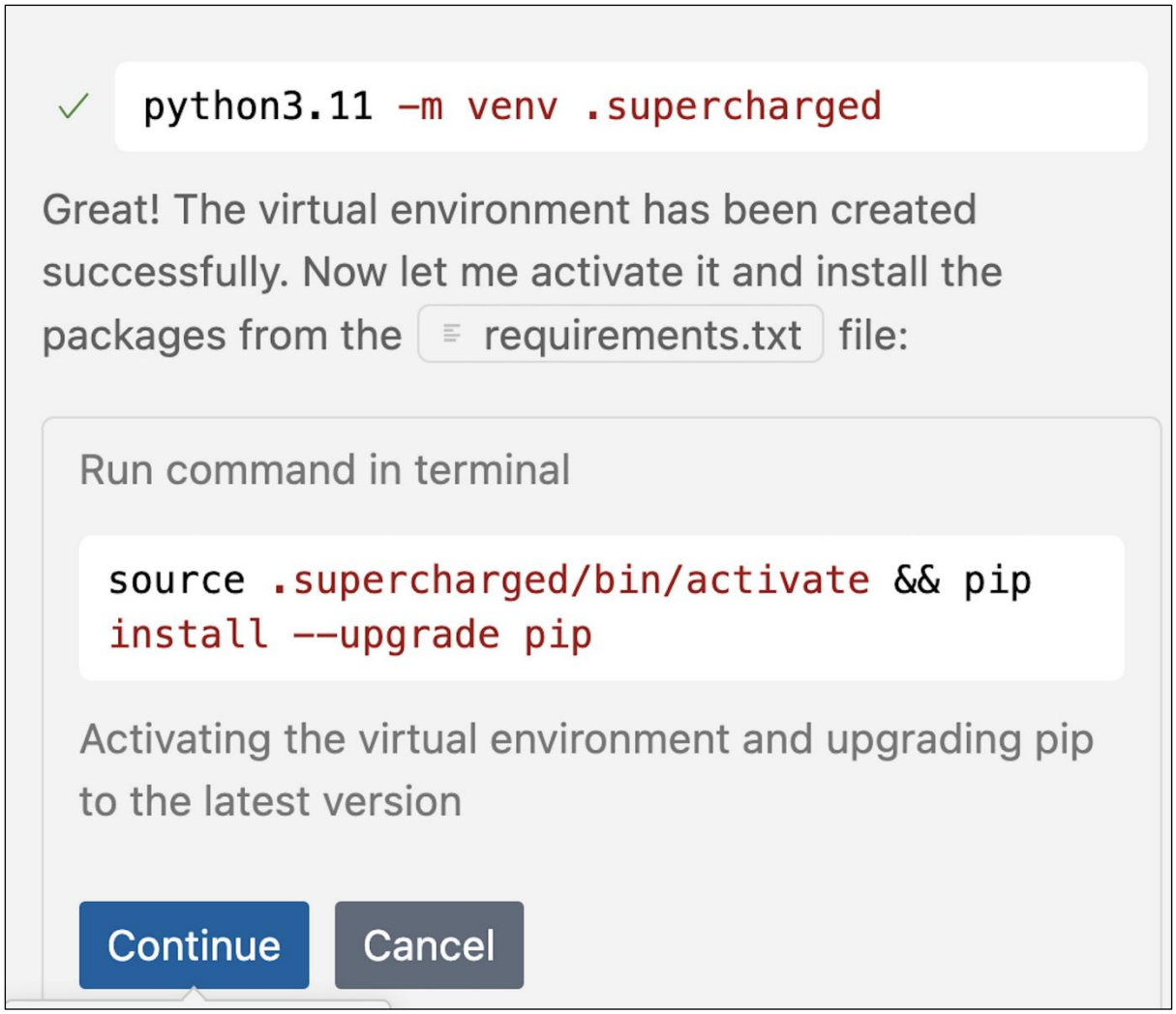


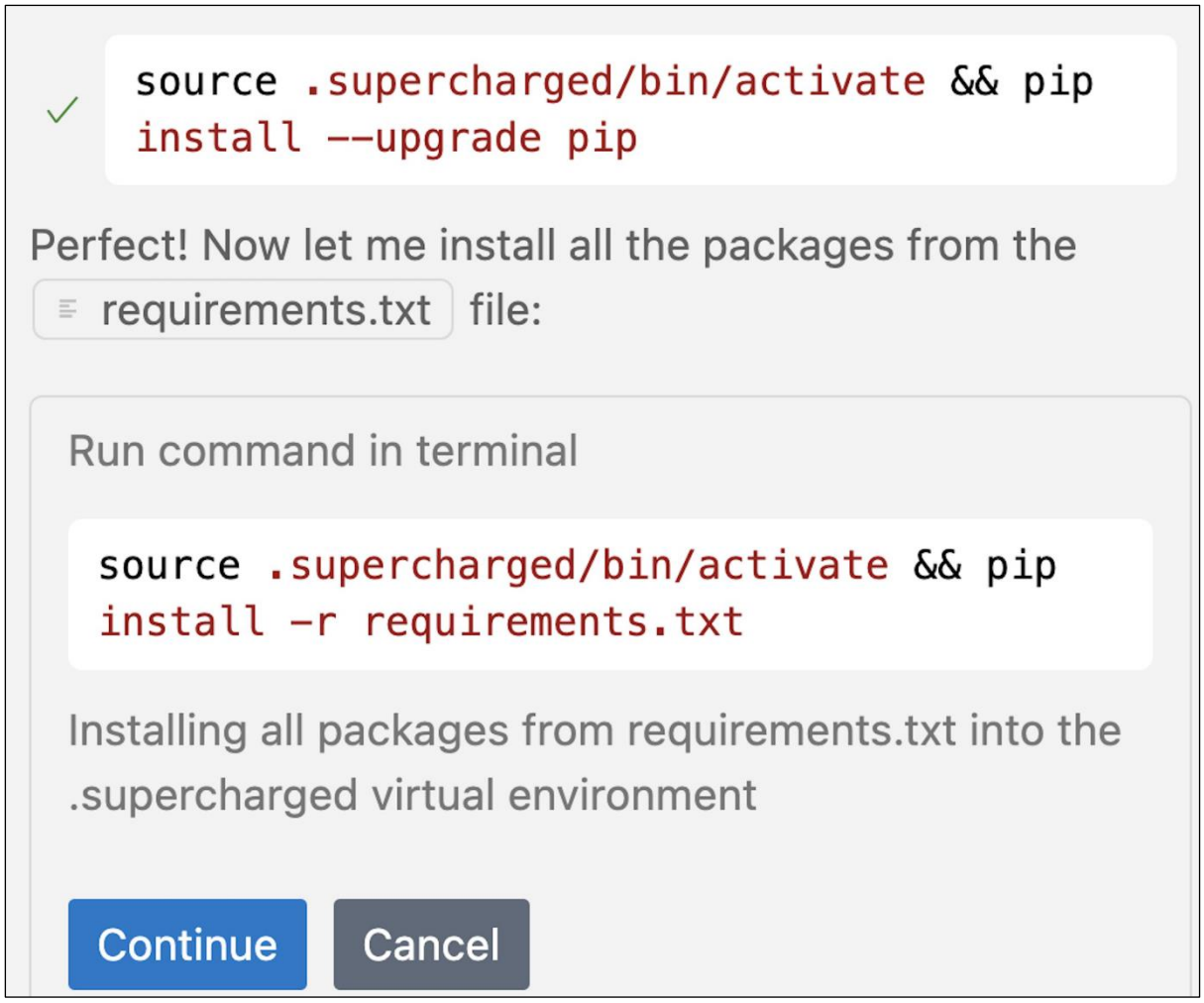
requirements.txt

```
pandas
numpy
openai
matplotlib
jupyter
memory-profiler
line_profiler
flask
pytest
```

requirements.txt

```
pandas==2.2.3
flask==3.1.0
```


python3.11 -m venv .supercharged
Great! The virtual environment has been created successfully. Now let me activate it and install the packages from the requirements.txt file:
Run command in terminal
source .supercharged/bin/activate && pip install --upgrade pip
Activating the virtual environment and upgrading pip to the latest version
Continue
Cancel


source .supercharged/bin/activate && pip install --upgrade pip
Perfect! Now let me install all the packages from the requirements.txt file:
Run command in terminal
source .supercharged/bin/activate && pip install -r requirements.txt
Installing all packages from requirements.txt into the .supercharged virtual environment
Continue
Cancel

CHAT

Virtual Environment Setup Complete! 🎉

I have successfully set up the virtual environment for your project with the following specifications:

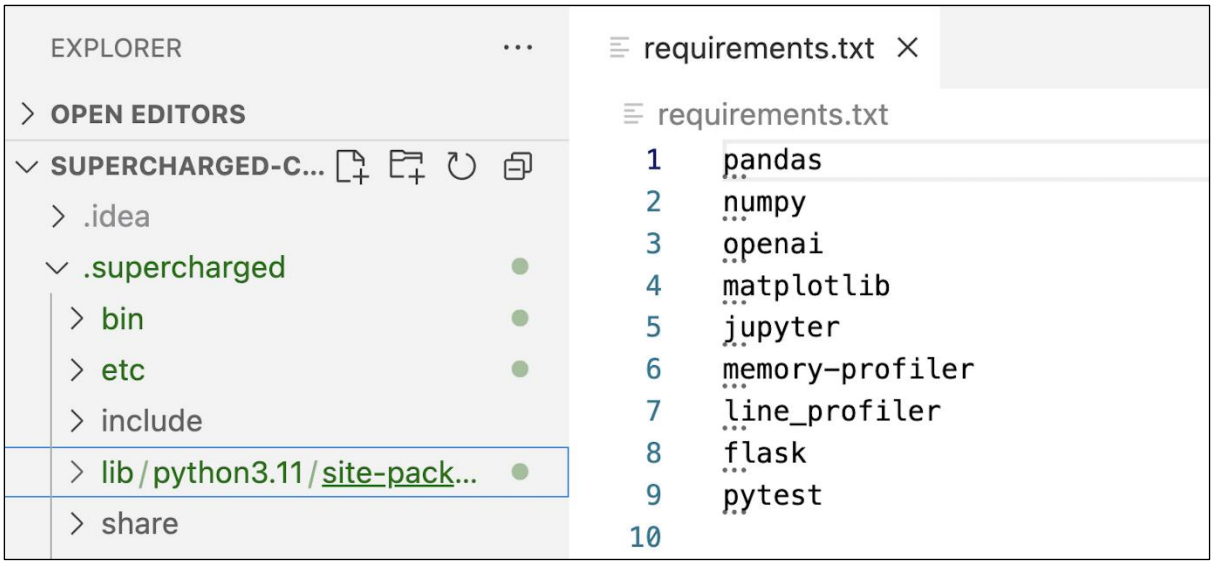
✅ Setup Summary:

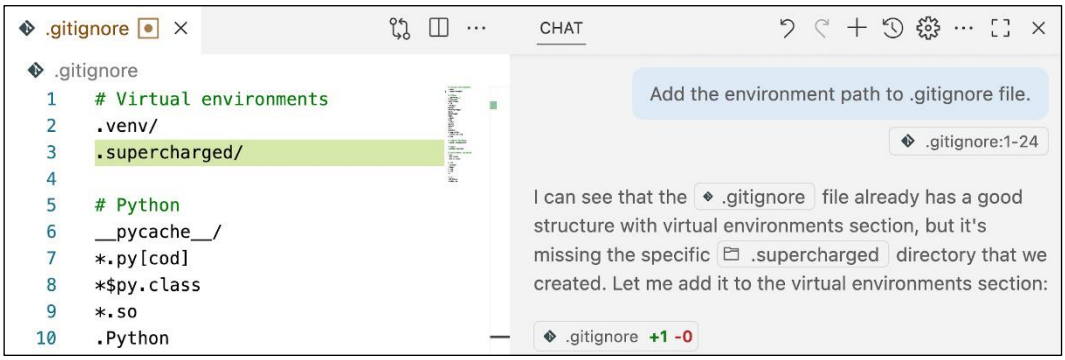
- **Environment Name:** `.supercharged`
- **Location:** Project root (`.supercharged`)
- **Python Version:** Python 3.11.2
- **Tool Used:** `venv`

✅ Installed Packages:

All packages from `requirements.txt` have been successfully installed:

- **pandas** (2.3.1)
- **numpy** (2.3.2)
- **openai** (1.99.6)
- **matplotlib** (3.10.5)
- **jupyter** (1.1.1) - Full Jupyter suite including JupyterLab
- **memory-profiler** (0.61.0)


EXPLORER
OPEN EDITORS
SUPERCHARGED-C...
.idea
.supercharged
bin
etc
include
lib / python3.11 / site-pack...
share
requirements.txt
requirements.txt
1 pandas
2 numpy
3 openai
4 matplotlib
5 jupyter
6 memory-profiler
7 line_profiler
8 flask
9 pytest
10


.gitignore
.gitignore
1 # Virtual environments
2 .venv/
3 .supercharged/
4
5 # Python
6 __pycache__/
7 *.py[cod]
8 *$py.class
9 *.so
10 .Python
CHAT
Add the environment path to .gitignore file.
.gitignore:1-24
I can see that the .gitignore file already has a good structure with virtual environments section, but it's missing the specific .supercharged directory that we created. Let me add it to the virtual environments section:
.gitignore +1 -0