Grant Thornton Speaking Engagement

Vibe coding with Generative AI

June 19

Dr. R.M. (Rolando) Gonzales Martinez

Vibe coding with Gen AI in Python

13:30 - 15:00

Introduction

- AI, LLMs, Transformers, AI algorithms and models
- Vibe coding with Python in VS

Vibe coding with Python in Practice

- Automatizing flowchart’s design/coding with AI visual recognition
- Vibe coding with ChatGPT, DeepSeek and CoPilot
- Vibe coding with local LLMs for sensitive information

Discussion

https://github.com/rogon666/GT_training

Grant Thornton Speaking Engagement

Vibe coding with Generative AI

Introduction

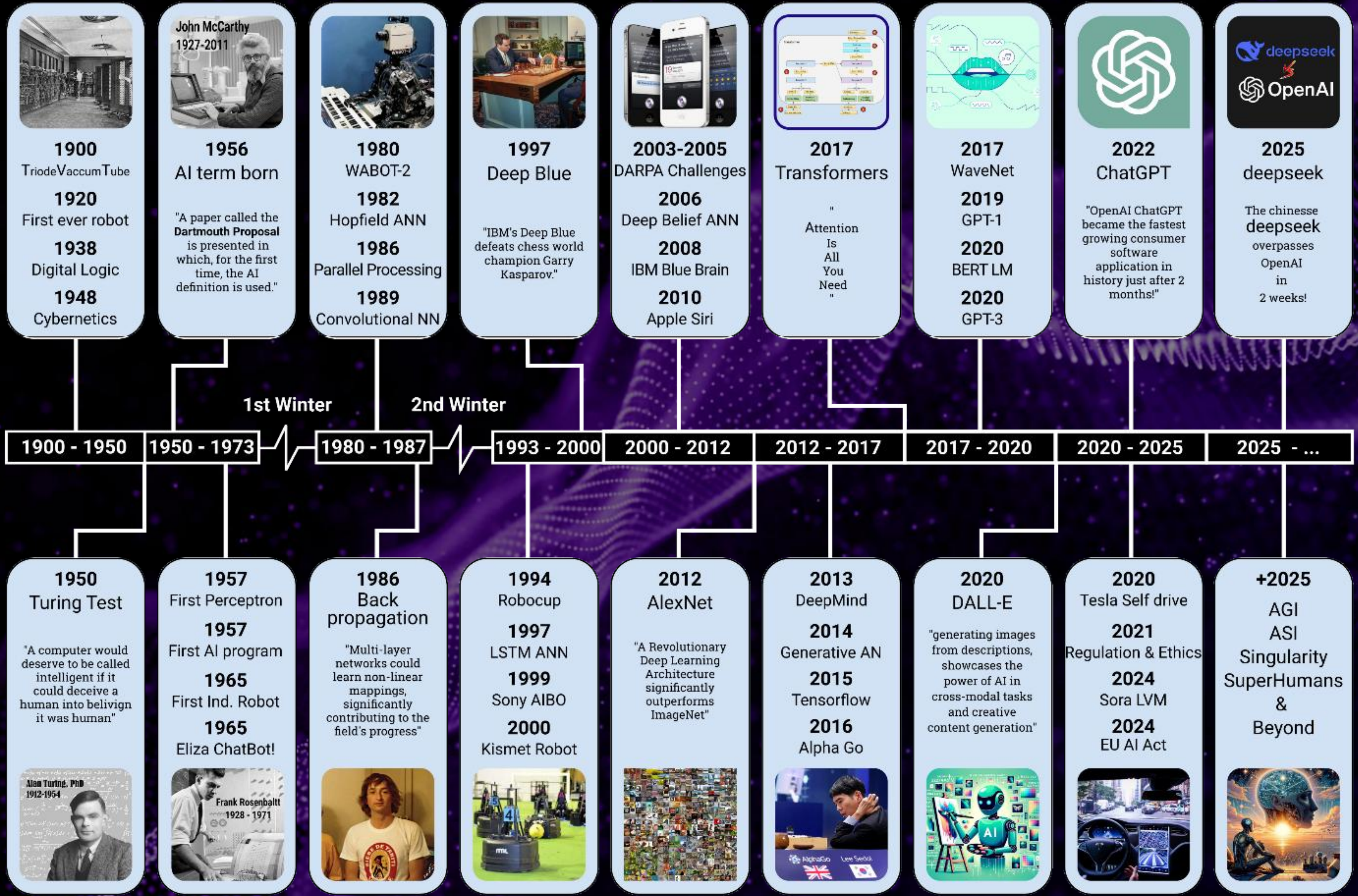
- AI, LLMs, Transformers, AI algorithms and models
- Vibe coding with Python in VS

Vibe coding with Python in Practice

- Automatizing flowchart’s design/coding with AI visual recognition
- Vibe coding with ChatGPT, DeepSeek and CoPilot
- Vibe coding with local LLMs for sensitive information

Discussion

Introduction



1900
TriodeVaccumTube
1920
First ever robot
1938
Digital Logic
1948
Cybernetics

John McCarthy
1927-2011
1956
AI term born
"A paper called the **Dartmouth Proposal** is presented in which, for the first time, the AI definition is used."

1980
WABOT-2
1982
Hopfield ANN
1986
Parallel Processing
1989
Convolutional NN

1997
Deep Blue
"IBM's Deep Blue defeats chess world champion Garry Kasparov."

2003-2005
DARPA Challenges
2006
Deep Belief ANN
2008
IBM Blue Brain
2010
Apple Siri

2017
Transformers
" Attention Is All You Need "

2017
WaveNet
2019
GPT-1
2020
BERT LM
2020
GPT-3

2022
ChatGPT
"OpenAI ChatGPT became the fastest growing consumer software application in history just after 2 months!"

deepseek
OpenAI
2025
deepseek
The chinesse **deepseek** overpasses OpenAI in 2 weeks!

1900 - 1950
1950 - 1973
1st Winter
1980 - 1987
2nd Winter
1993 - 2000
2000 - 2012
2012 - 2017
2017 - 2020
2020 - 2025
2025 - ...

1950
Turing Test
"A computer would deserve to be called intelligent if it could deceive a human into belivign it was human"
Alan Turing, PhD
1912-1954

1957
First Perceptron
1957
First AI program
1965
First Ind. Robot
1965
Eliza ChatBot!
Frank Rosenbaltt
1928 - 1971

1986
Back propagation
"Multi-layer networks could learn non-linear mappings, significantly contributing to the field's progress"

1994
Robocup
1997
LSTM ANN
1999
Sony AIBO
2000
Kismet Robot

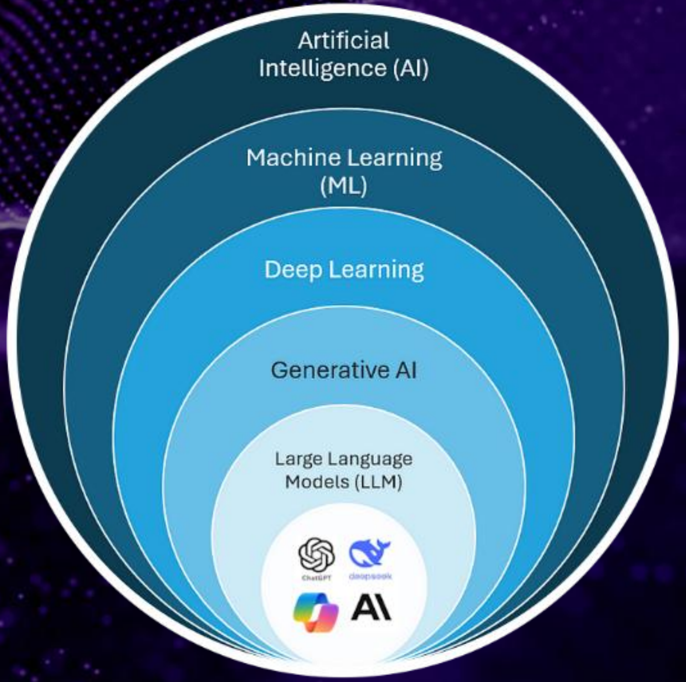
2012
AlexNet
"A Revolutionary Deep Learning Architecture significantly outperforms ImageNet"

2013
DeepMind
2014
Generative AN
2015
Tensorflow
2016
Alpha Go

2020
DALL-E
"generating images from descriptions, showcases the power of AI in cross-modal tasks and creative content generation"

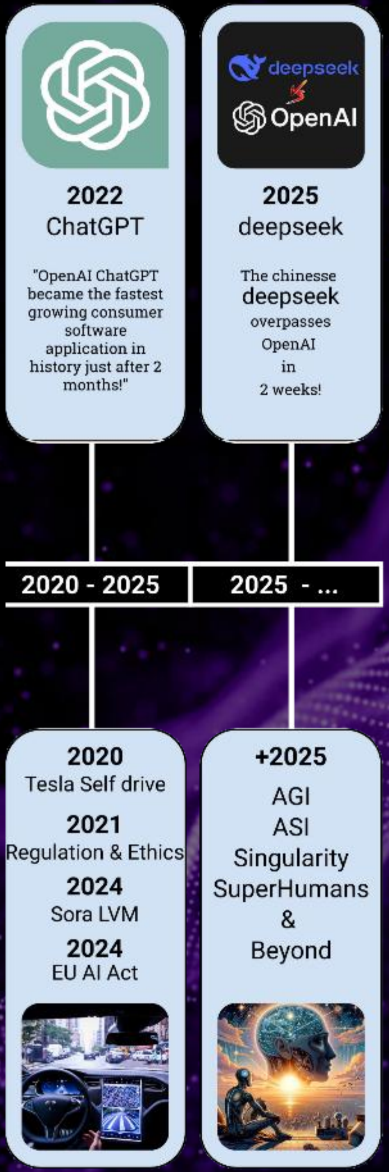
2020
Tesla Self drive
2021
Regulation & Ethics
2024
Sora LVM
2024
EU AI Act

+2025
AGI
ASI
Singularity
SuperHumans
&
Beyond



Artificial Intelligence (AI)
Machine Learning (ML)
Deep Learning
Generative AI
Large Language Models (LLM)

Introduction



2022
ChatGPT

"OpenAI ChatGPT became the fastest growing consumer software application in history just after 2 months!"

2025
deepseek

The chinesse **deepseek** overpasses OpenAI in 2 weeks!

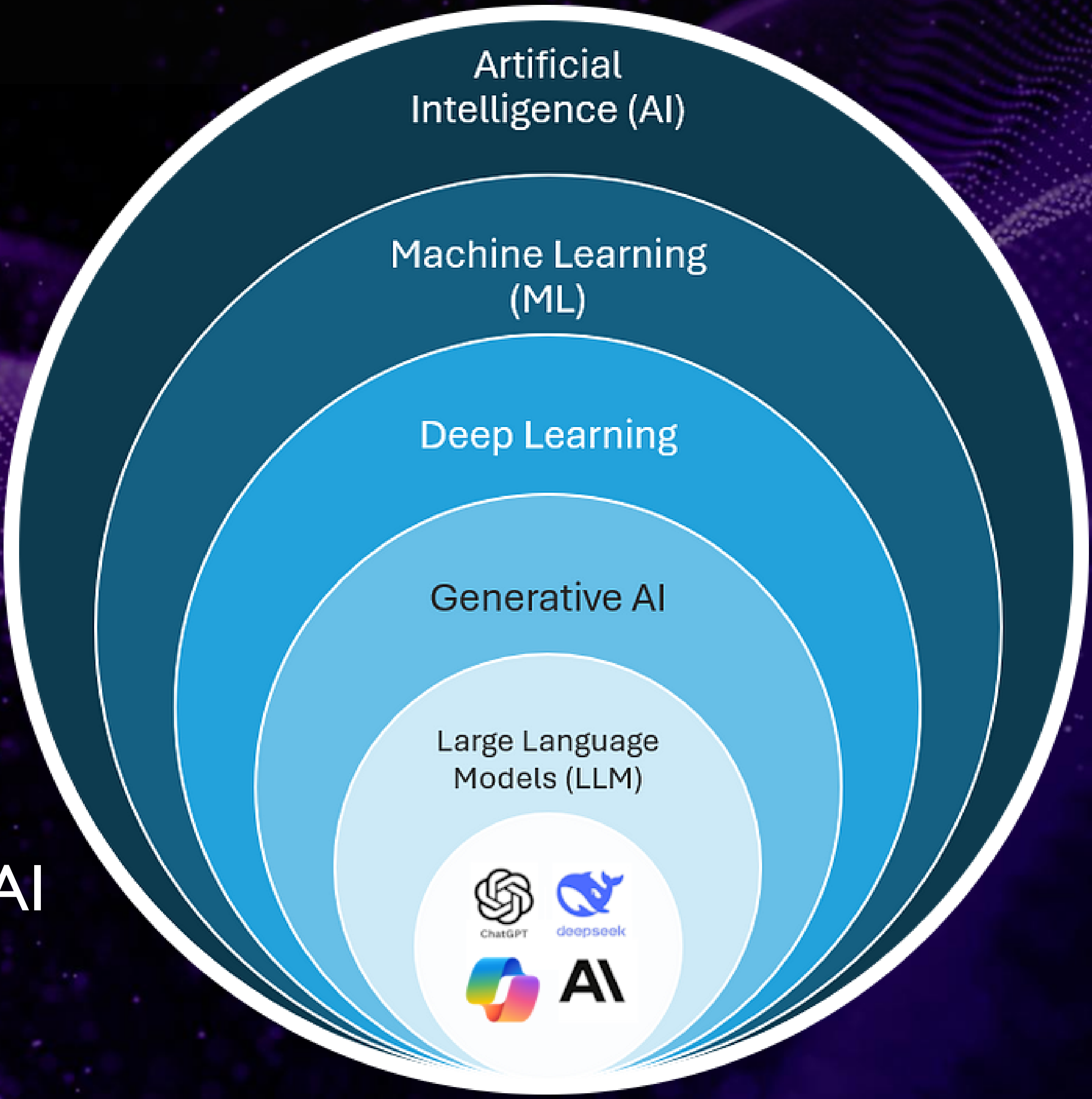
2020 - 2025

2025 - ...

2020
Tesla Self drive
2021
Regulation & Ethics
2024
Sora LVM
2024
EU AI Act

+2025
AGI
ASI
Singularity
SuperHumans
&
Beyond

Narrow AI
vs.
AGI, ASI



Artificial Intelligence (AI)

Machine Learning (ML)

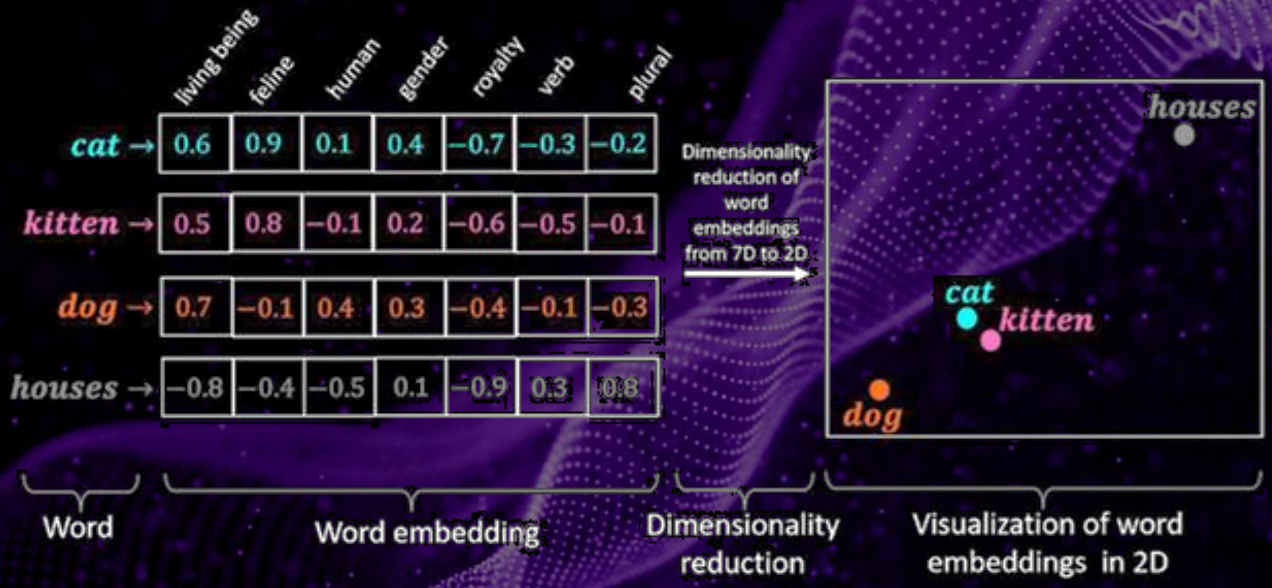
Deep Learning

Generative AI

Large Language Models (LLM)

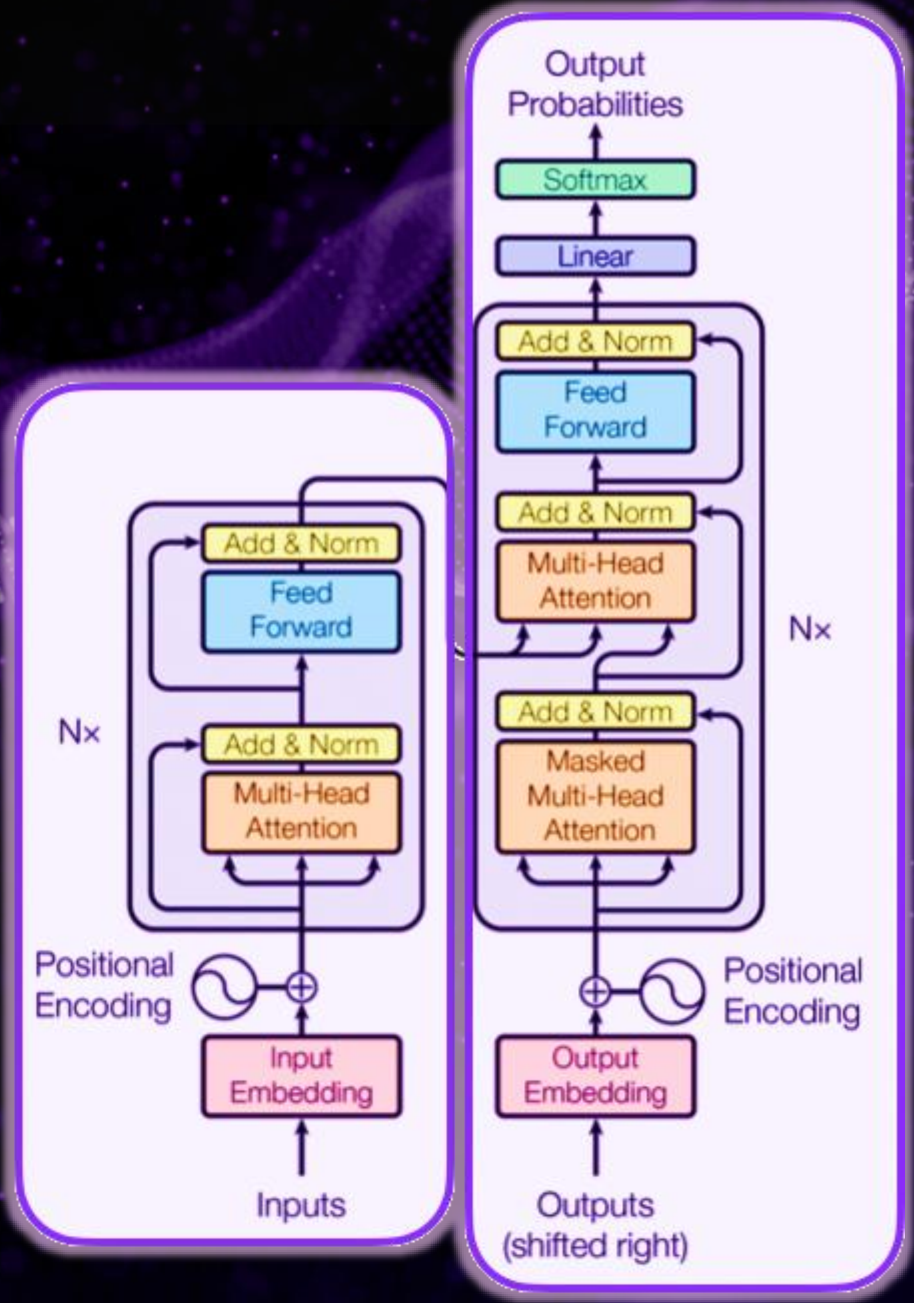
Introduction

LLMs, transformers, NTAs (LNN)



Word	living being	feline	human	gender	royalty	verb	plural
cat	0.6	0.9	0.1	0.4	−0.7	−0.3	−0.2
kitten	0.5	0.8	−0.1	0.2	−0.6	−0.5	−0.1
dog	0.7	−0.1	0.4	0.3	−0.4	−0.1	−0.3
houses	−0.8	−0.4	−0.5	0.1	−0.9	0.3	0.8

Dimensionality reduction of word embeddings from 7D to 2D
houses
cat
kitten
dog
Word
Word embedding
Dimensionality reduction
Visualization of word embeddings in 2D



Introduction

What is vibe coding?

- New paradigm in programming: intentions are expressed in natural language and AI tools generate the corresponding code.
- Coined by Andrej Karpathy in 2025
- Shifts focus from syntax to creative design
- Makes coding more accessible and intuitive

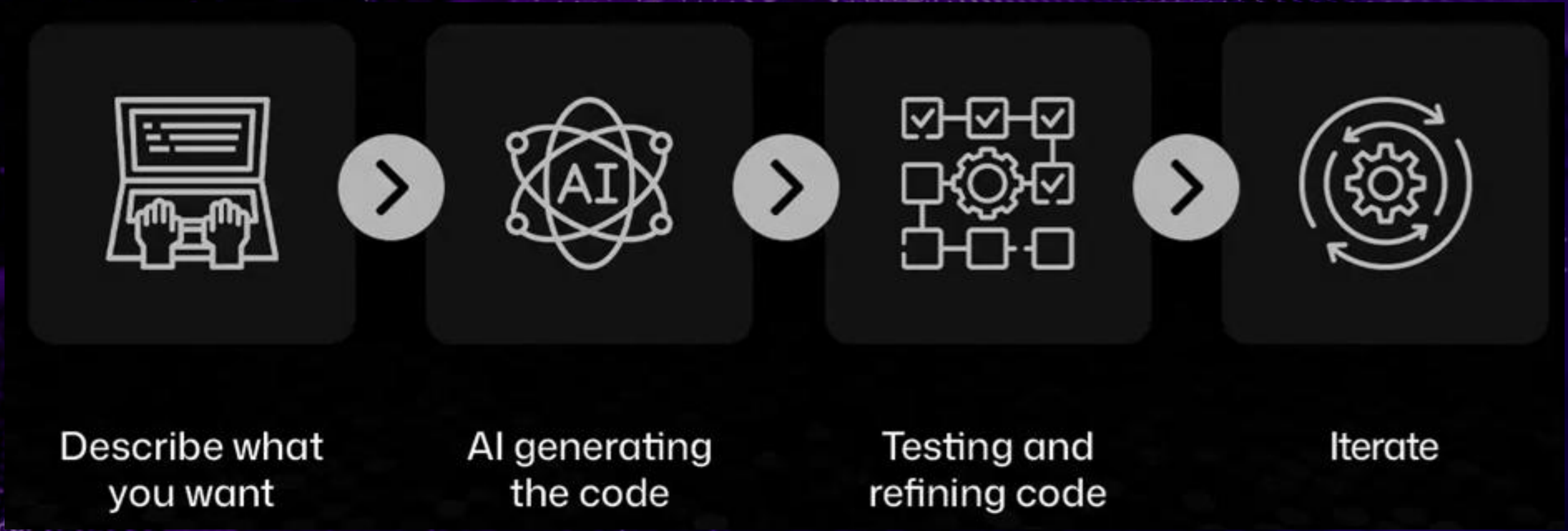
Introduction

Vibe coding emphasizes the creative 'vibe' of development over traditional rules:

- Encourages human-AI collaboration
- Lowers the barrier for non-programmers
- Focuses on high-level logic and design

Introduction

How to Vibe Code in Python?



Elliptic envelopes

$$X = \{x_1, x_2, \ldots, x_n\} \subset \mathbb{R}^d$$

$$\mu = \frac{1}{n}\sum_{i=1}^{n} x_i \quad \mu \in \mathbb{R}^d$$

$$\Sigma = \frac{1}{n}\sum_{i=1}^{n}(x_i - \mu)(x_i - \mu)^\top$$

$$\Sigma \in \mathbb{R}^{d\times d}$$

$$D_M(x) = \sqrt{(x-\mu)^\top \Sigma^{-1}(x-\mu)}$$

$$D_M(x)^2 > \chi^2_{d,1-\alpha}$$

$$x = \{x_1, x_2, \ldots, x_n\} \subset \mathbb{R}$$

$$D_M^2(x_i) = \left(\frac{x_i - \mu_{\text{rob}}}{\sigma_{\text{rob}}}\right)^2$$

$$T = \chi^2_{1,1-\alpha}$$

$$D_M^2(x_i) > T$$

Isolation Forests

If $\mathbb{E}[h(x)] \ll c(\psi)$, then $s(x) \approx 1$: strong anomaly

If $\mathbb{E}[h(x)] \approx c(\psi)$, then $s(x) \approx 0.5$: borderline

If $\mathbb{E}[h(x)] \gg c(\psi)$, then $s(x) \approx 0$: normal point

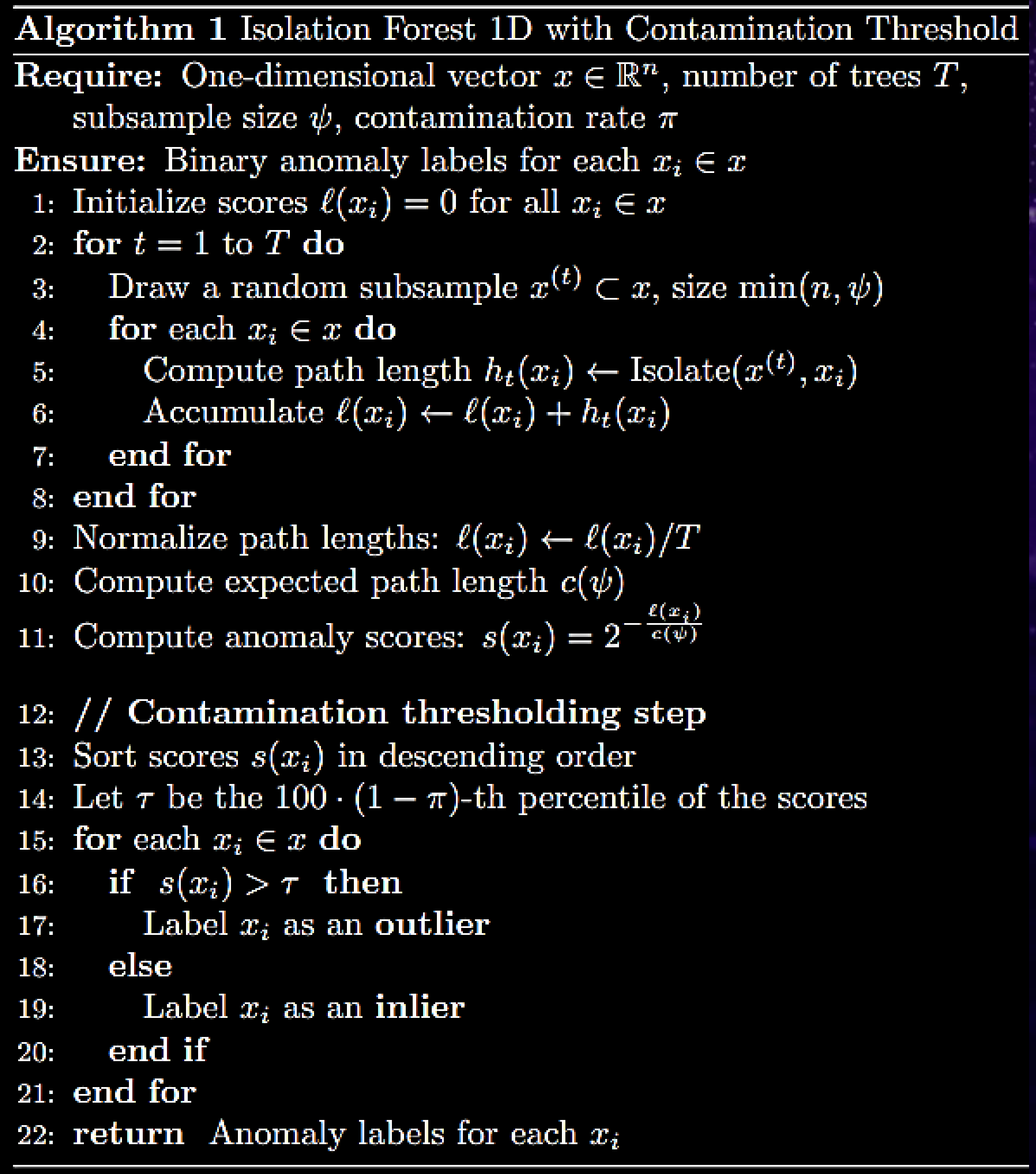
$$s(x) = 2^{-\frac{\mathbb{E}[h(x)]}{c(n)}}$$

$$\mathbb{E}[h(x)] = \frac{1}{T}\sum_{t=1}^{T} h_t(x)$$

$$c(n) = 2H(n-1) - \frac{2(n-1)}{n}$$

$$H(i) \approx \ln(i) + \gamma$$

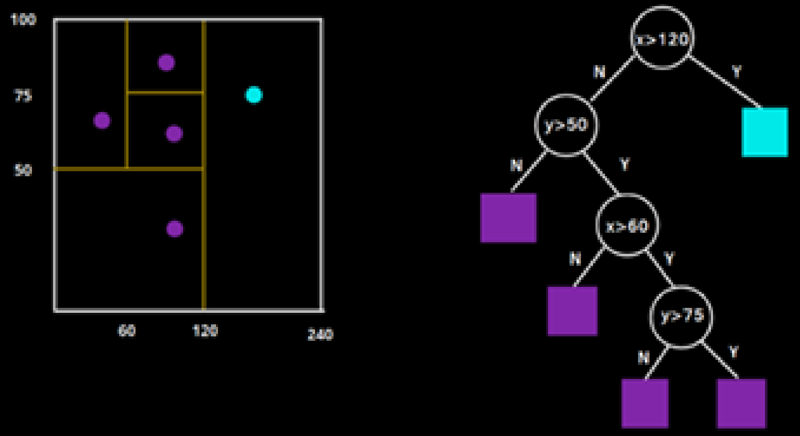
$\gamma \approx 0.5772$ Euler–Mascheroni constant



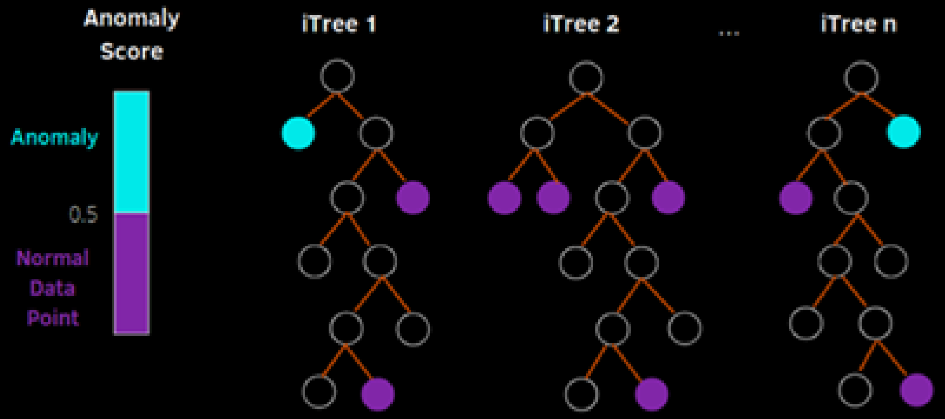
Algorithm 1 Isolation Forest 1D with Contamination Threshold

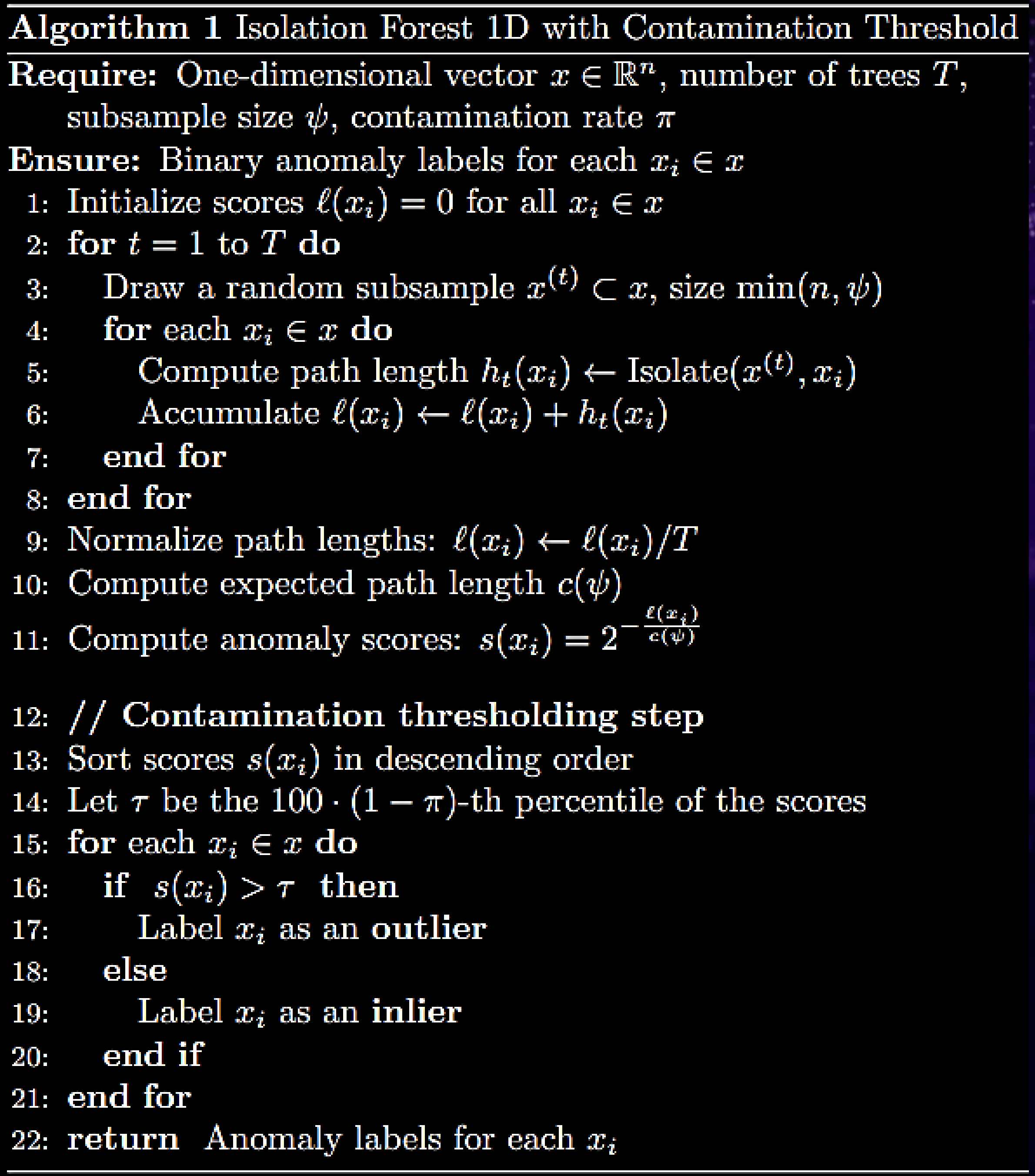
```
Require: One-dimensional vector $x \in \mathbb{R}^n$, number of trees $T$,
    subsample size $\psi$, contamination rate $\pi$
Ensure: Binary anomaly labels for each $x_i \in x$
Initialize scores $\ell(x_i) = 0$ for all $x_i \in x$
for $t = 1$ to $T$ do
    Draw a random subsample $x^{(t)} \subset x$, size $\min(n, \psi)$
    for each $x_i \in x$ do
        Compute path length $h_t(x_i) \leftarrow \text{Isolate}(x^{(t)}, x_i)$
        Accumulate $\ell(x_i) \leftarrow \ell(x_i) + h_t(x_i)$
    end for
end for
Normalize path lengths: $\ell(x_i) \leftarrow \ell(x_i)/T$
Compute expected path length $c(\psi)$
Compute anomaly scores: $s(x_i) = 2^{-\frac{\ell(x_i)}{c(\psi)}}$

// Contamination thresholding step
Sort scores $s(x_i)$ in descending order
Let $\tau$ be the $100 \cdot (1 - \pi)$-th percentile of the scores
for each $x_i \in x$ do
    if $s(x_i) > \tau$ then
        Label $x_i$ as an outlier
    else
        Label $x_i$ as an inlier
    end if
end for
return Anomaly labels for each $x_i$
```

Isolation Forests



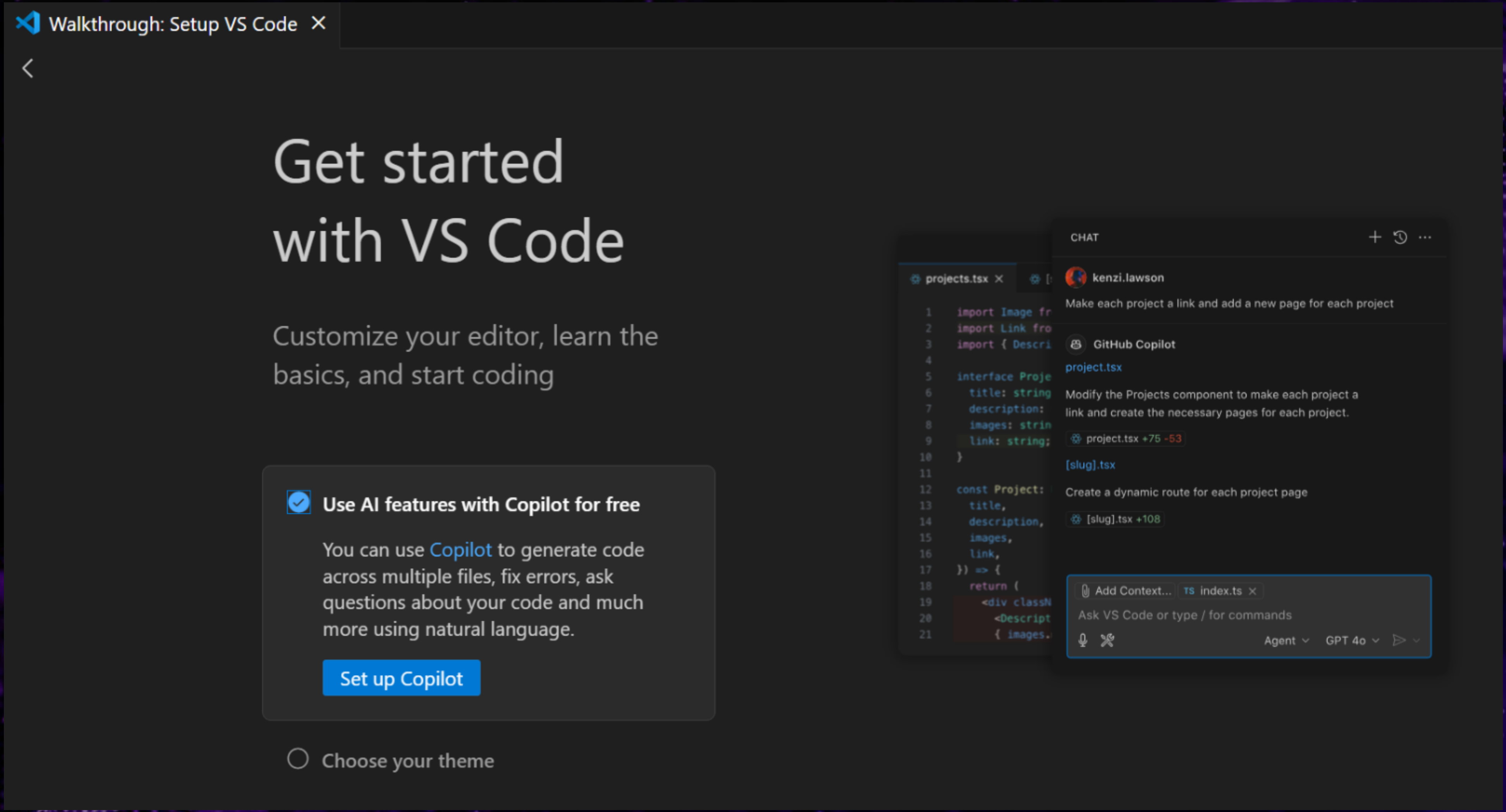
Anomaly Score





```
Algorithm 1 Isolation Forest 1D with Contamination Threshold
Require: One-dimensional vector $x \in \mathbb{R}^n$, number of trees $T$,
    subsample size $\psi$, contamination rate $\pi$
Ensure: Binary anomaly labels for each $x_i \in x$
Initialize scores $\ell(x_i) = 0$ for all $x_i \in x$
for $t = 1$ to $T$ do
    Draw a random subsample $x^{(t)} \subset x$, size $\min(n, \psi)$
    for each $x_i \in x$ do
        Compute path length $h_t(x_i) \leftarrow \text{Isolate}(x^{(t)}, x_i)$
        Accumulate $\ell(x_i) \leftarrow \ell(x_i) + h_t(x_i)$
    end for
end for
Normalize path lengths: $\ell(x_i) \leftarrow \ell(x_i)/T$
Compute expected path length $c(\psi)$
Compute anomaly scores: $s(x_i) = 2^{-\frac{\ell(x_i)}{c(\psi)}}$

// Contamination thresholding step
Sort scores $s(x_i)$ in descending order
Let $\tau$ be the $100 \cdot (1 - \pi)$-th percentile of the scores
for each $x_i \in x$ do
    if $s(x_i) > \tau$ then
        Label $x_i$ as an outlier
    else
        Label $x_i$ as an inlier
    end if
end for
return Anomaly labels for each $x_i$
```

Vibe coding with Python in VS code



Get started with VS Code

Customize your editor, learn the basics, and start coding

☑ **Use AI features with Copilot for free**

You can use Copilot to generate code across multiple files, fix errors, ask questions about your code and much more using natural language.

Set up Copilot

○ Choose your theme

```
import Image fr
import Link fro
import { Descri

interface Proje
  title: string
  description:
  images: strin
  link: string;
}

const Project:
  title,
  description,
  images,
  link,
}) => {
  return (
    <div classN
      <Descript
      { images.
```

kenzi.lawson

Make each project a link and add a new page for each project

GitHub Copilot

project.tsx

Modify the Projects component to make each project a link and create the necessary pages for each project.

project.tsx +75 -53

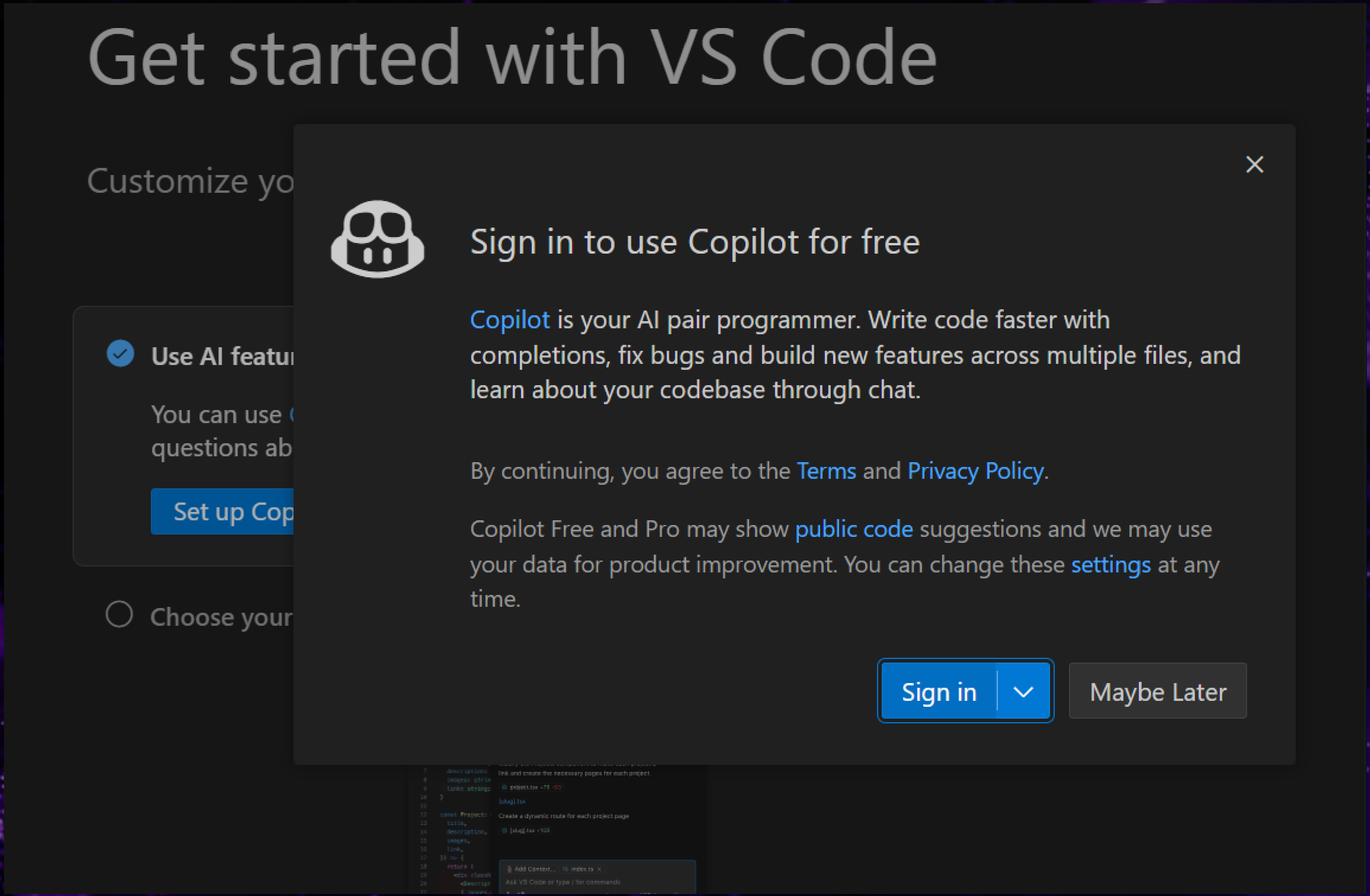
[slug].tsx

Create a dynamic route for each project page

[slug].tsx +108

Ask VS Code or type / for commands

Vibe coding with Python in VS code



Get started with VS Code

Customize yo

Use AI featu

You can use (
questions ab

○ Choose your

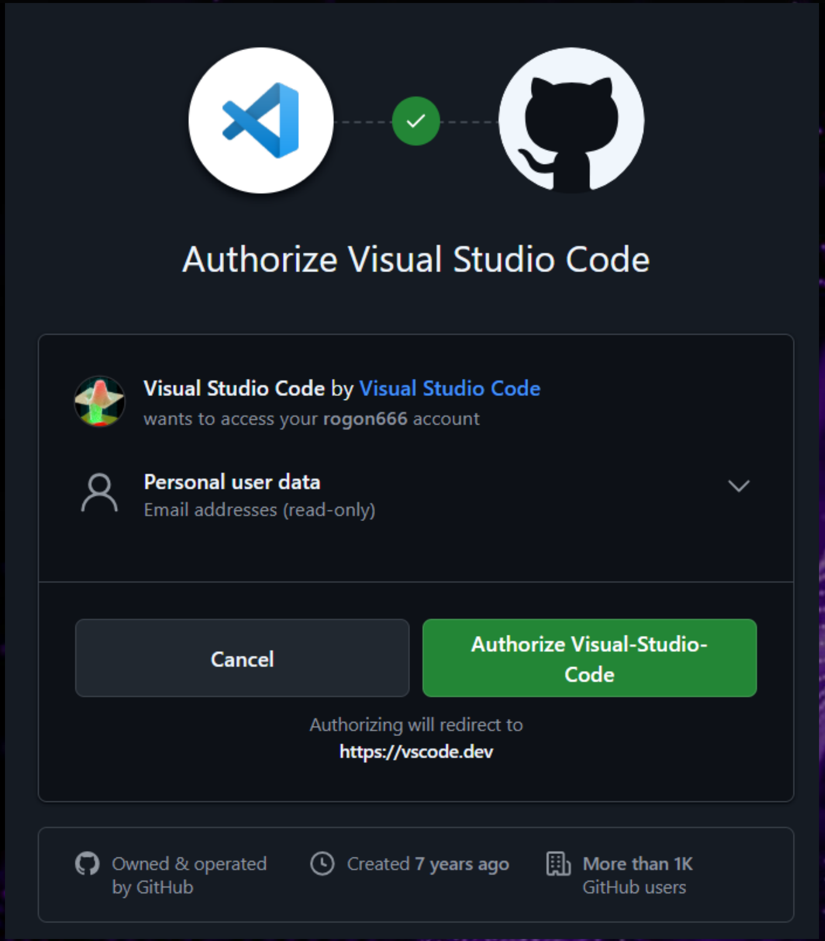
Sign in to use Copilot for free

Copilot is your AI pair programmer. Write code faster with completions, fix bugs and build new features across multiple files, and learn about your codebase through chat.

By continuing, you agree to the Terms and Privacy Policy.

Copilot Free and Pro may show public code suggestions and we may use your data for product improvement. You can change these settings at any time.

Vibe coding with Python



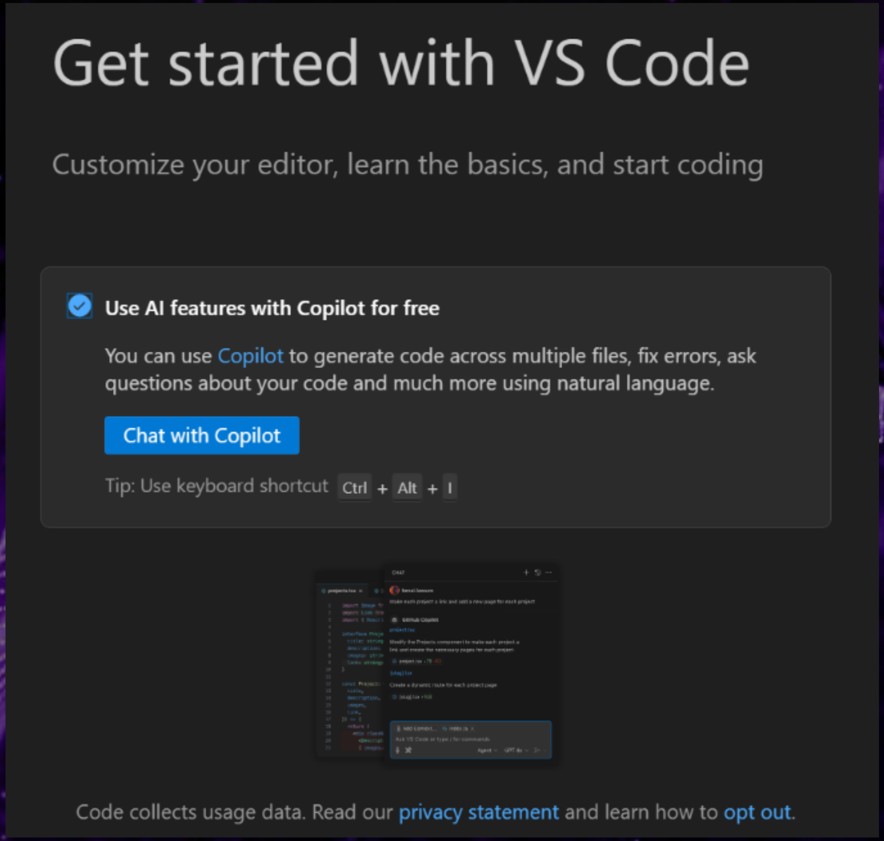
Authorize Visual Studio Code

Visual Studio Code by Visual Studio Code
wants to access your **rogon666** account

Personal user data
Email addresses (read-only)

Authorizing will redirect to
https://vscode.dev

Cancel
Authorize Visual-Studio-Code
Owned & operated by GitHub
Created 7 years ago
More than 1K GitHub users



Get started with VS Code

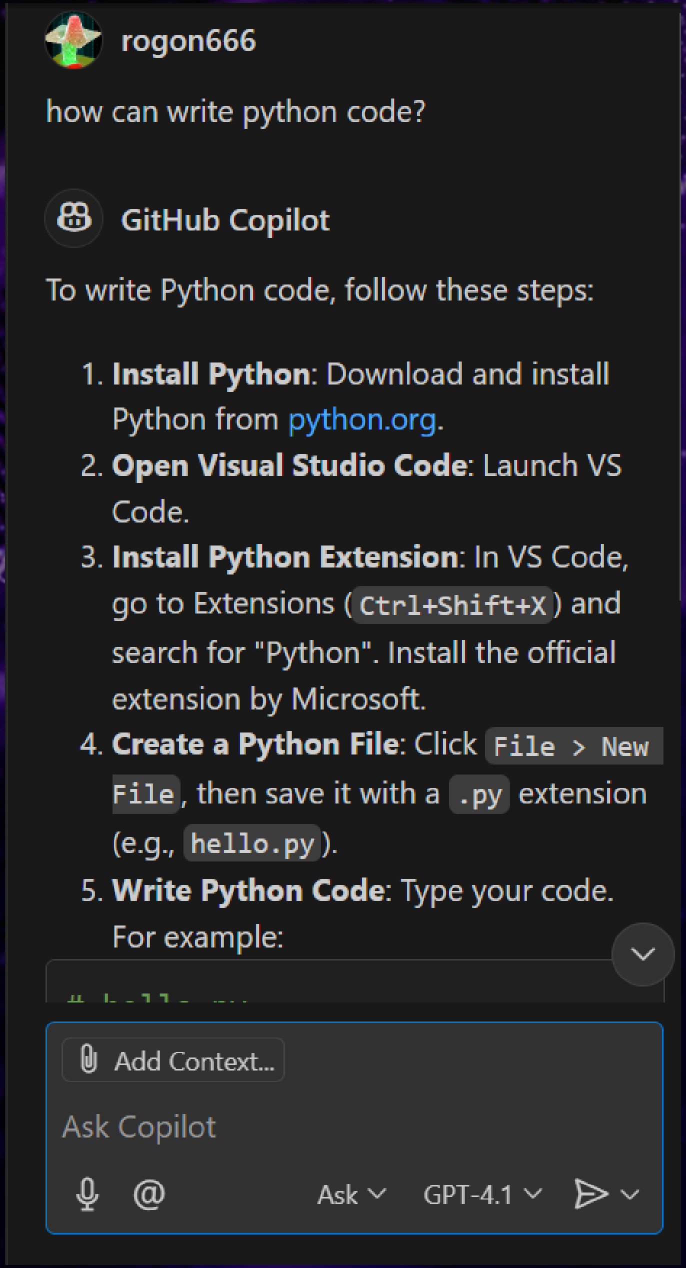
Customize your editor, learn the basics, and start coding

☑ **Use AI features with Copilot for free**

You can use Copilot to generate code across multiple files, fix errors, ask questions about your code and much more using natural language.

Tip: Use keyboard shortcut Ctrl + Alt + I

Code collects usage data. Read our privacy statement and learn how to opt out.



rogon666

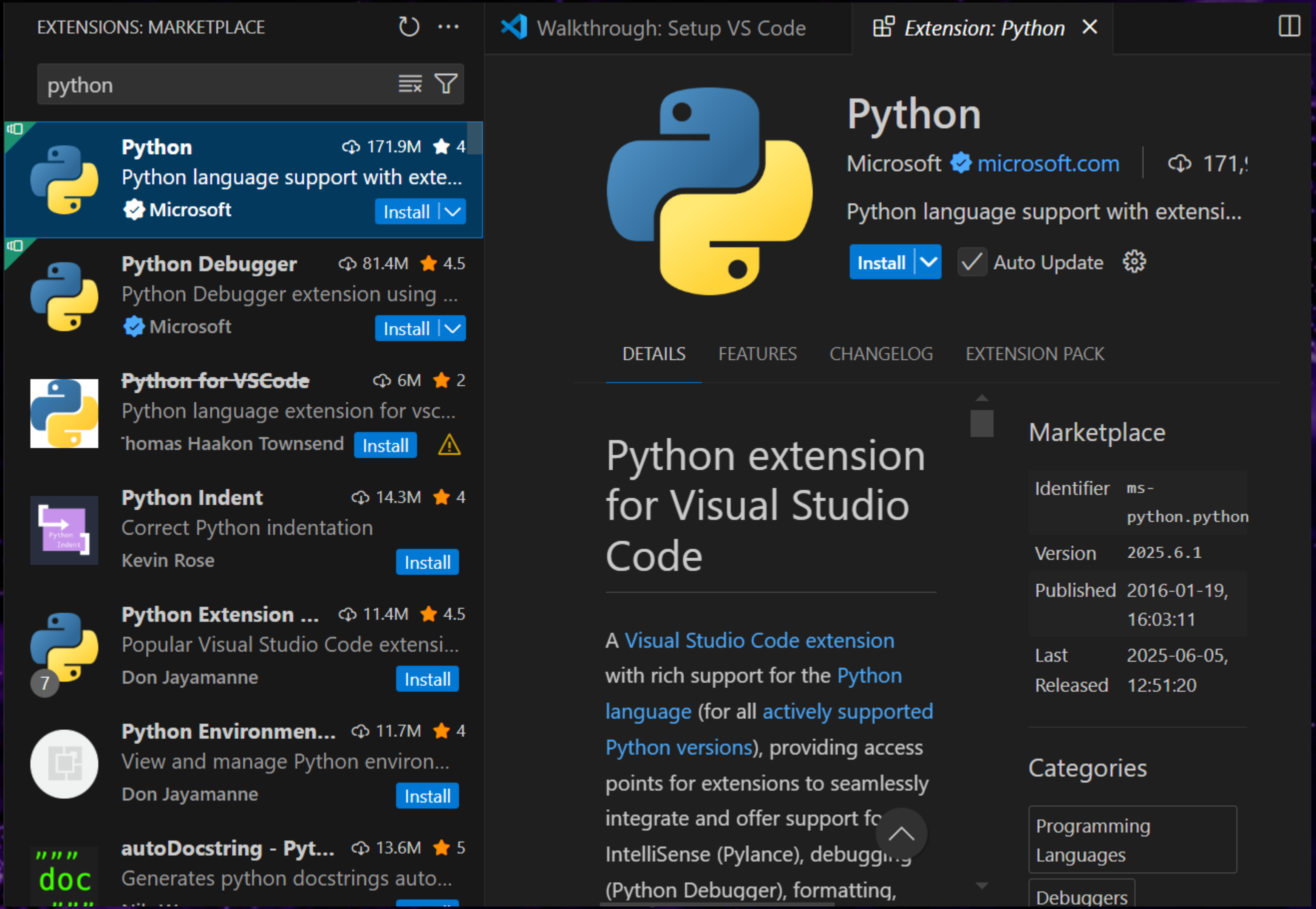
how can write python code?

GitHub Copilot

To write Python code, follow these steps:

1. **Install Python**: Download and install Python from python.org.
2. **Open Visual Studio Code**: Launch VS Code.
3. **Install Python Extension**: In VS Code, go to Extensions (`Ctrl+Shift+X`) and search for "Python". Install the official extension by Microsoft.
4. **Create a Python File**: Click `File > New File`, then save it with a `.py` extension (e.g., `hello.py`).
5. **Write Python Code**: Type your code. For example:

Vibe coding with Python in VS code



Python 171.9M ★ 4
Python language support with exte...
Microsoft Install

Python Debugger 81.4M ★ 4.5
Python Debugger extension using ...
Microsoft Install

~~**Python for VSCode**~~ 6M ★ 2
Python language extension for vsc...
'homas Haakon Townsend Install ⚠

Python Indent 14.3M ★ 4
Correct Python indentation
Kevin Rose Install

Python Extension ... 11.4M ★ 4.5
Popular Visual Studio Code extensi...
Don Jayamanne Install

Python Environmen... 11.7M ★ 4
View and manage Python environ...
Don Jayamanne Install

autoDocstring - Pyt... 13.6M ★ 5
Generates python docstrings auto...

Python

Microsoft microsoft.com | 171,

Python language support with extensi...

Install ☑ Auto Update

Python extension for Visual Studio Code

A Visual Studio Code extension with rich support for the Python language (for all actively supported Python versions), providing access points for extensions to seamlessly integrate and offer support fo IntelliSense (Pylance), debugg... (Python Debugger), formatting,

Marketplace

Identifier	ms-python.python
Version	2025.6.1
Published	2016-01-19, 16:03:11
Last Released	2025-06-05, 12:51:20

Categories

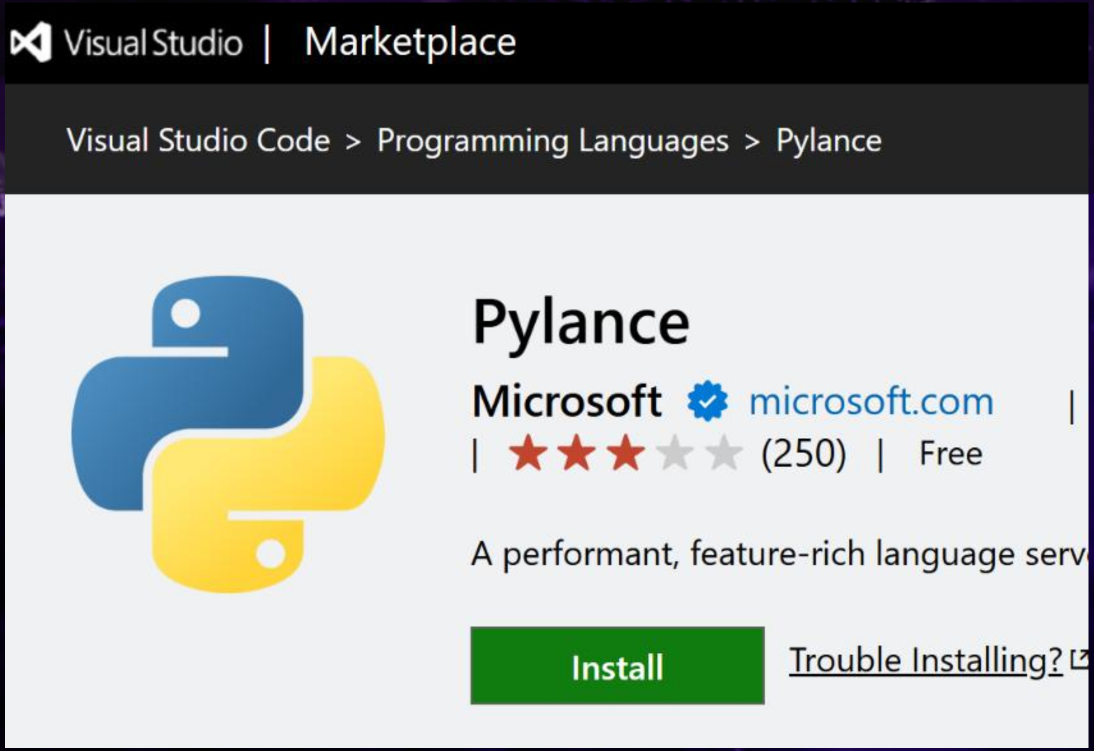
Programming Languages

Debuggers

Vibe coding with Python in VS code

Pylance

Pylance is an extension that **works alongside Python in Visual Studio Code** to provide performant language support



Visual Studio | Marketplace

Visual Studio Code > Programming Languages > Pylance

Pylance

Microsoft microsoft.com |
| ★★★☆☆ (250) | Free

A performant, feature-rich language serv

Grant Thornton Speaking Engagement

Vibe coding with Generative AI

Vibe coding with Gen AI in Python

13:30 - 15:00

Introduction

- AI, LLMs, Transformers, AI algorithms and models
- Vibe coding with Python in VS

Vibe coding with Python in Practice

- Automatizing flowchart’s design/coding with AI visual recognition
- Vibe coding with ChatGPT, DeepSeek and CoPilot
- Vibe coding with local LLMs for sensitive information

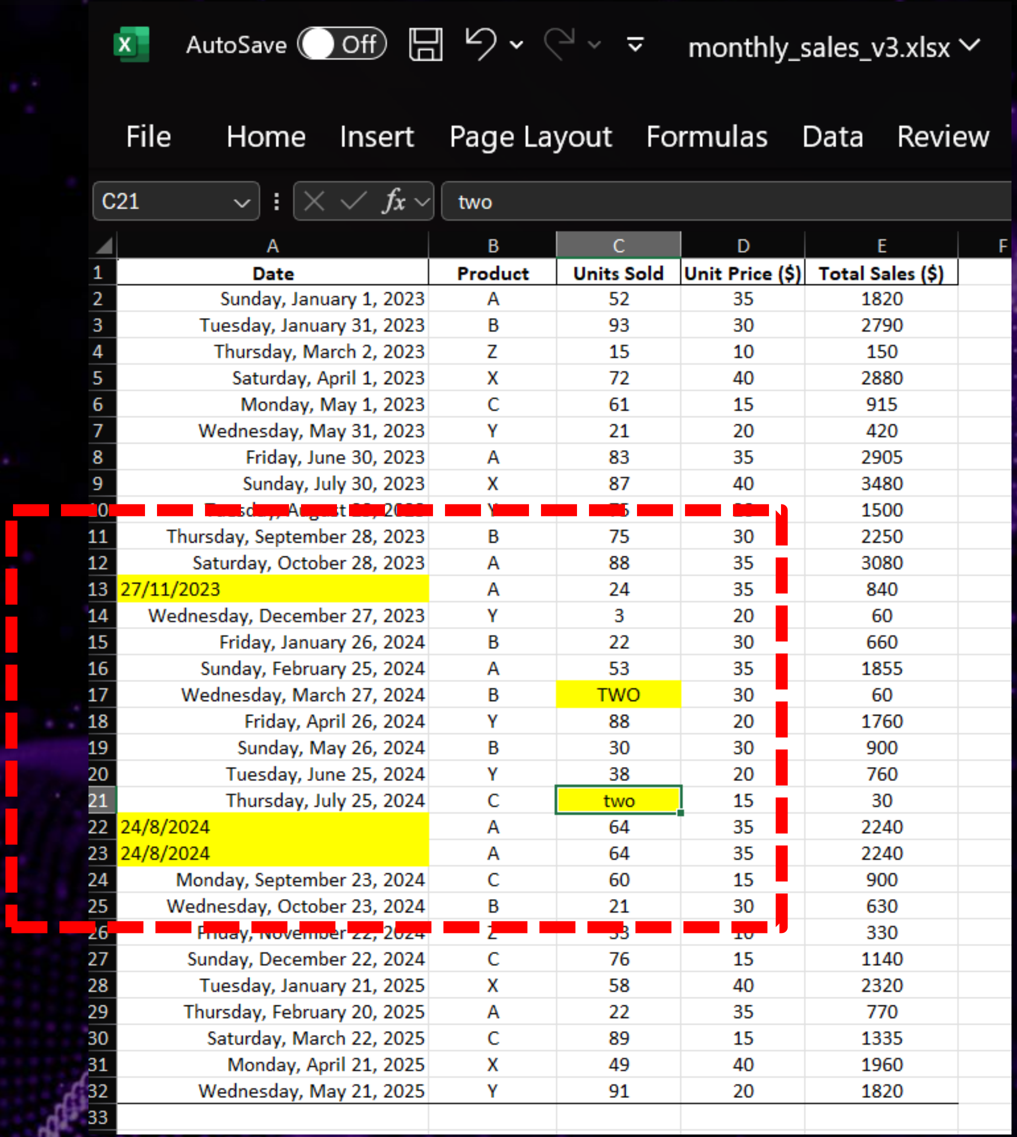
Discussion

Vibe coding with Python in Practice

Goal:

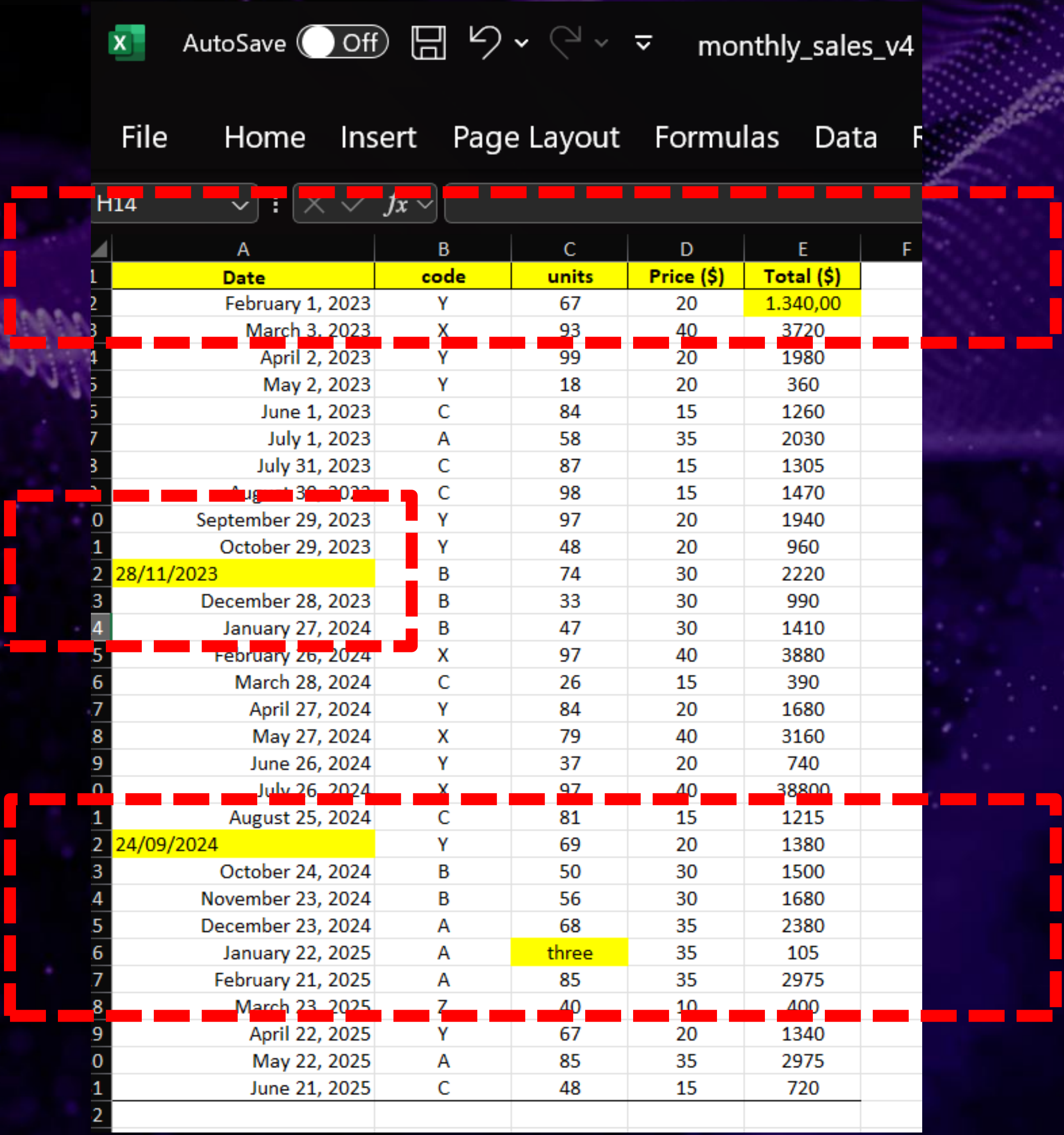
1. Clean and merge multiple Excel files with evident errors in formats and entries (“Excel hell”):
    - monthly_sales_v3.xlsx
    - monthly_sales_v4 - Copy final use this.xlsx
2. Anomaly detection: find not-so-easy-to-spot errors (with isolation forests and elliptical envelopes)
3. Fine tuning the machine learning models

Vibe coding with Python



C21 | two

	A	B	C	D	E
1	Date	Product	Units Sold	Unit Price ($)	Total Sales ($)
2	Sunday, January 1, 2023	A	52	35	1820
3	Tuesday, January 31, 2023	B	93	30	2790
4	Thursday, March 2, 2023	Z	15	10	150
5	Saturday, April 1, 2023	X	72	40	2880
6	Monday, May 1, 2023	C	61	15	915
7	Wednesday, May 31, 2023	Y	21	20	420
8	Friday, June 30, 2023	A	83	35	2905
9	Sunday, July 30, 2023	X	87	40	3480
10	Tuesday, August 29, 2023	Y	75	20	1500
11	Thursday, September 28, 2023	B	75	30	2250
12	Saturday, October 28, 2023	A	88	35	3080
13	27/11/2023	A	24	35	840
14	Wednesday, December 27, 2023	Y	3	20	60
15	Friday, January 26, 2024	B	22	30	660
16	Sunday, February 25, 2024	A	53	35	1855
17	Wednesday, March 27, 2024	B	TWO	30	60
18	Friday, April 26, 2024	Y	88	20	1760
19	Sunday, May 26, 2024	B	30	30	900
20	Tuesday, June 25, 2024	Y	38	20	760
21	Thursday, July 25, 2024	C	two	15	30
22	24/8/2024	A	64	35	2240
23	24/8/2024	A	64	35	2240
24	Monday, September 23, 2024	C	60	15	900
25	Wednesday, October 23, 2024	B	21	30	630
26	Friday, November 22, 2024	Z	33	10	330
27	Sunday, December 22, 2024	C	76	15	1140
28	Tuesday, January 21, 2025	X	58	40	2320
29	Thursday, February 20, 2025	A	22	35	770
30	Saturday, March 22, 2025	C	89	15	1335
31	Monday, April 21, 2025	X	49	40	1960
32	Wednesday, May 21, 2025	Y	91	20	1820
33					



H14

A	B	C	D	E
Date	code	units	Price ($)	Total ($)
February 1, 2023	Y	67	20	1.340,00
March 3, 2023	X	93	40	3720
April 2, 2023	Y	99	20	1980
May 2, 2023	Y	18	20	360
June 1, 2023	C	84	15	1260
July 1, 2023	A	58	35	2030
July 31, 2023	C	87	15	1305
August 30, 2023	C	98	15	1470
September 29, 2023	Y	97	20	1940
October 29, 2023	Y	48	20	960
28/11/2023	B	74	30	2220
December 28, 2023	B	33	30	990
January 27, 2024	B	47	30	1410
February 26, 2024	X	97	40	3880
March 28, 2024	C	26	15	390
April 27, 2024	Y	84	20	1680
May 27, 2024	X	79	40	3160
June 26, 2024	Y	37	20	740
July 26, 2024	X	97	40	38800
August 25, 2024	C	81	15	1215
24/09/2024	Y	69	20	1380
October 24, 2024	B	50	30	1500
November 23, 2024	B	56	30	1680
December 23, 2024	A	68	35	2380
January 22, 2025	A	three	35	105
February 21, 2025	A	85	35	2975
March 23, 2025	Z	40	10	400
April 22, 2025	Y	67	20	1340
May 22, 2025	A	85	35	2975
June 21, 2025	C	48	15	720

Vibe coding with Python in Practice

```
import os
import pandas as pd

new_dir = "C:/~/folder_with_excel_files"
os.chdir(new_dir)
print("New Directory:", os.getcwd())

df1 = pd.read_excel("monthly_sales_v3.xlsx“)
df2 = pd.read_excel("monthly_sales_v4 - Copy final use this.xlsx“)
df1.head()
df2.head()
```

Vibe coding with Python in Practice

Goal:

1. Clean and merge multiple Excel files with evident errors in formats and entries (“Excel hell”):
    - monthly_sales_v3.xlsx
    - monthly_sales_v4 - Copy final use this.xlsx
2. Anomaly detection: find not-so-easy-to-spot errors (with isolation forests and elliptical envelopes)
3. Fine tuning the machine learning models

- Automatizing flowchart’s design and coding with AI visual recognition
- Vibe coding with ChatGPT, DeepSeek and CoPilot
- Vibe coding with local LLMs for sensitive information

Grant Thornton Speaking Engagement

Vibe coding with AI

Introduction

- AI, LLMs, Transformers, AI algorithms and models
- Vibe coding with Python in VS

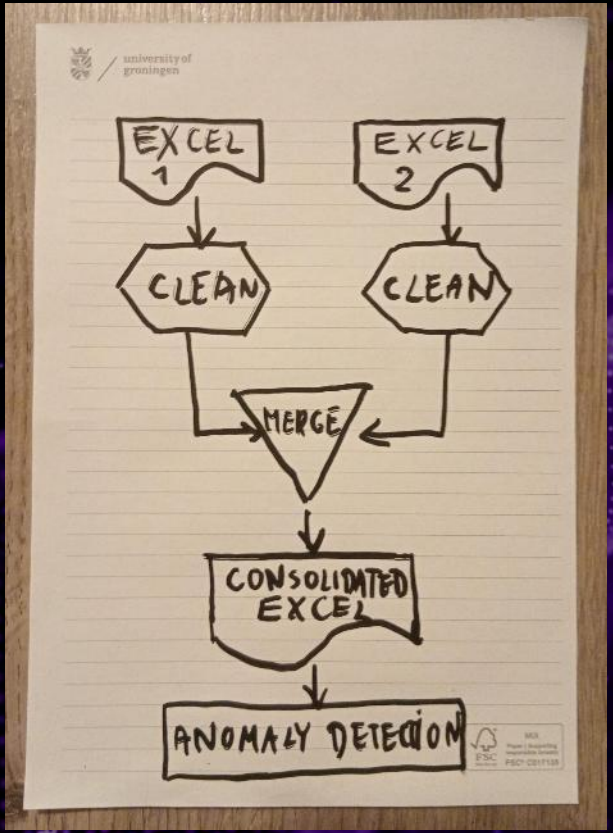
Vibe coding with Python in Practice

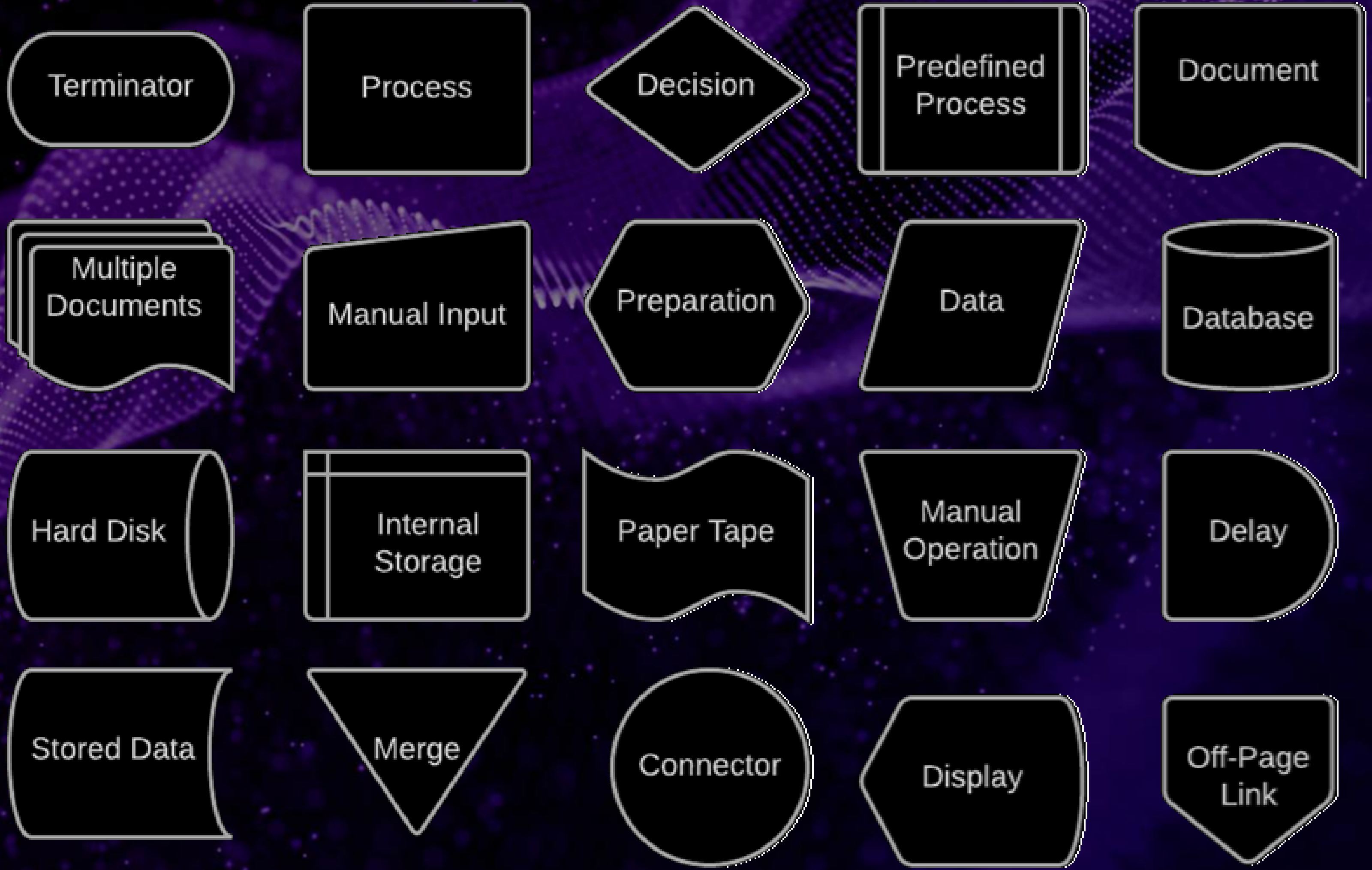
- Automatizing flowchart’s design/coding with AI visual recognition
- Vibe coding with ChatGPT, DeepSeek and CoPilot
- Vibe coding with local LLMs for sensitive information

Discussion

Vibe coding with Python in Practice

Automatizing flowchart's design





Terminator
Process
Decision
Predefined Process
Document

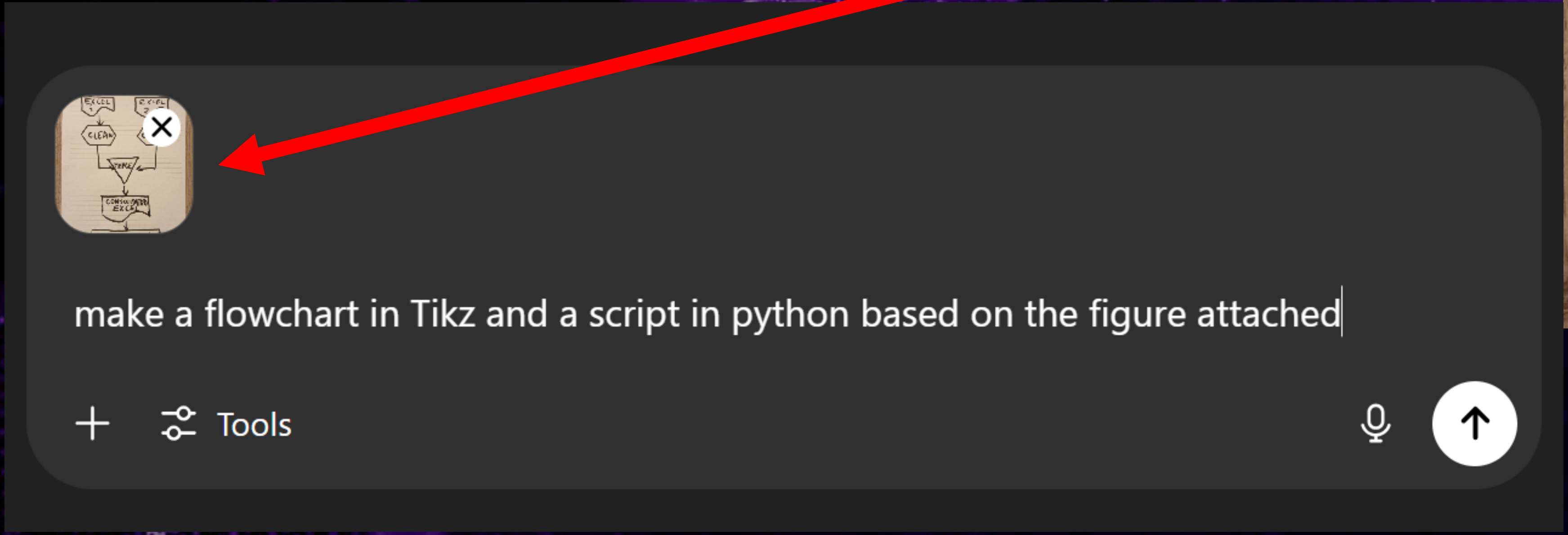
Multiple Documents
Manual Input
Preparation
Data
Database

Hard Disk
Internal Storage
Paper Tape
Manual Operation
Delay

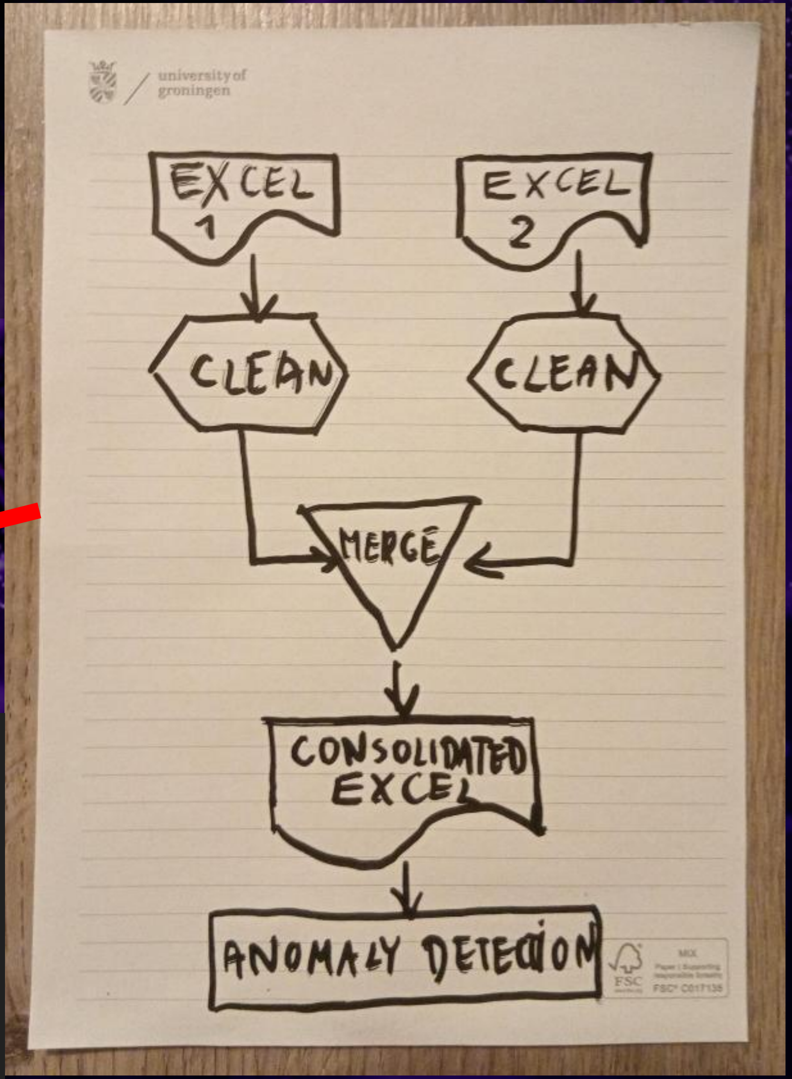
Stored Data
Merge
Connector
Display
Off-Page Link

Vibe coding with Python

Automatizing flowchart's design

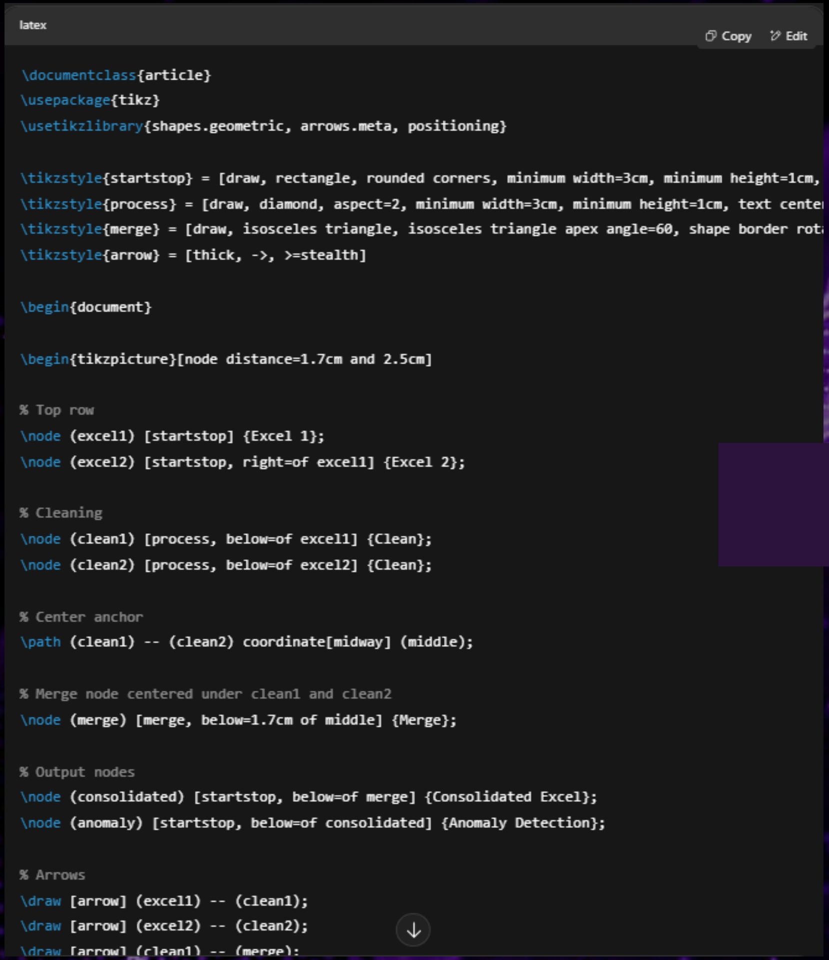


make a flowchart in Tikz and a script in python based on the figure attached

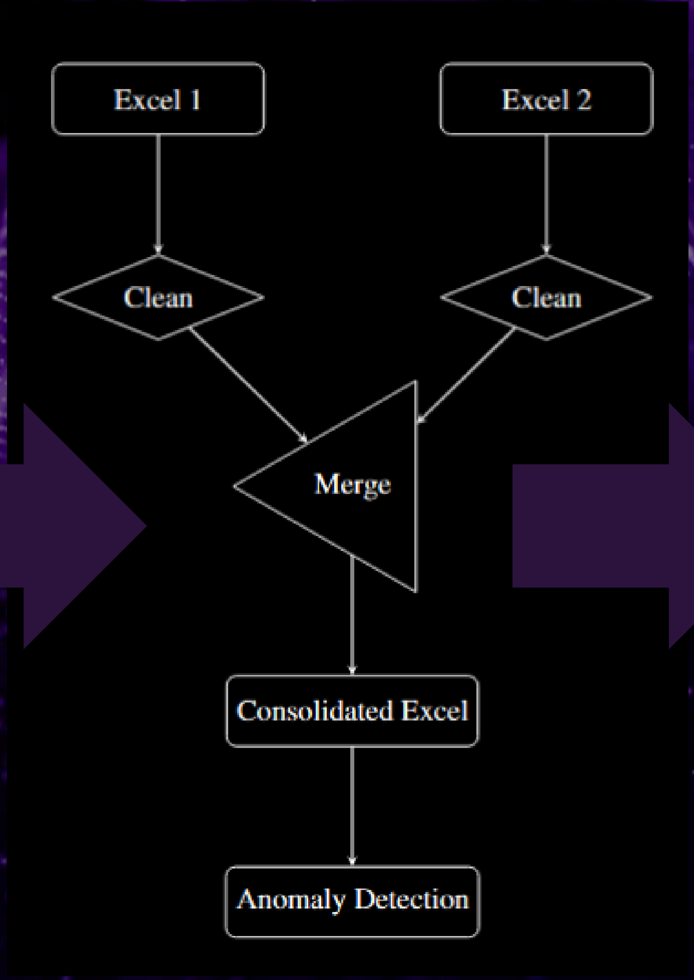


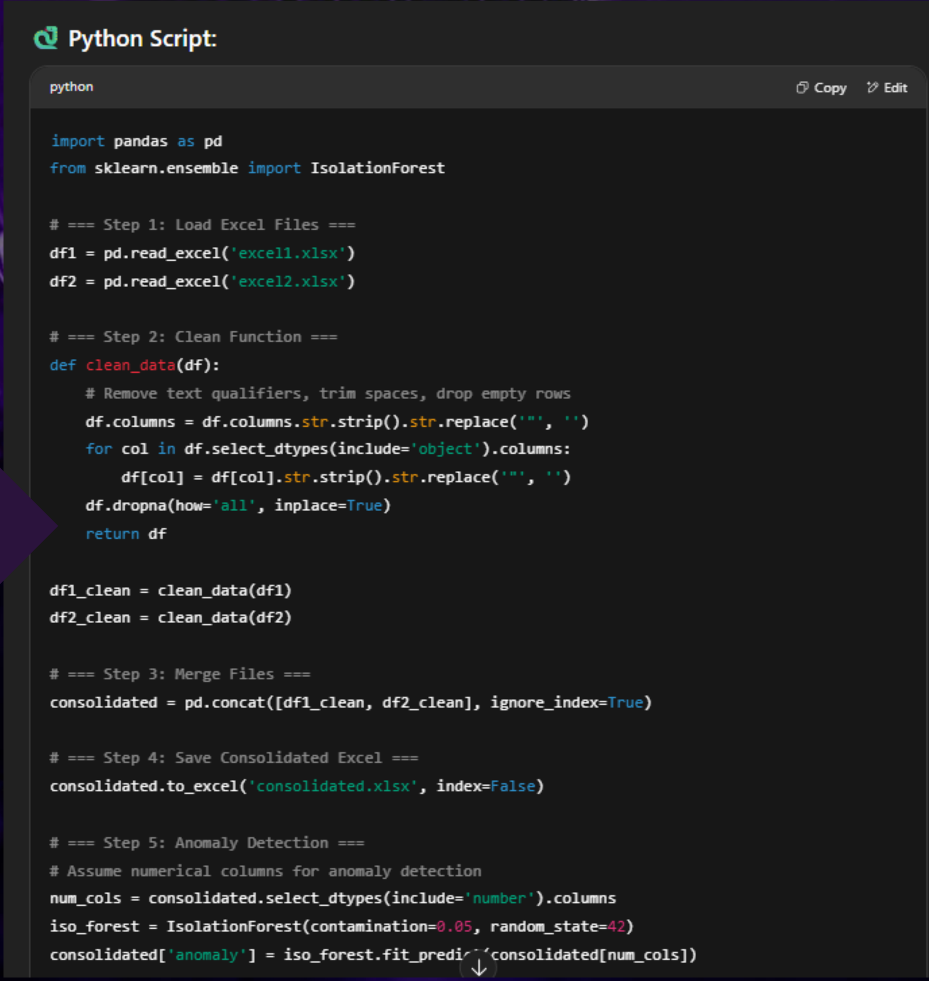
Vibe coding with Python

Automatizing flowchart’s design with ChatGPT



```latex
\documentclass{article}
\usepackage{tikz}
\usetikzlibrary{shapes.geometric, arrows.meta, positioning}

\tikzstyle{startstop} = [draw, rectangle, rounded corners, minimum width=3cm, minimum height=1cm,
\tikzstyle{process} = [draw, diamond, aspect=2, minimum width=3cm, minimum height=1cm, text cente
\tikzstyle{merge} = [draw, isosceles triangle, isosceles triangle apex angle=60, shape border rot
\tikzstyle{arrow} = [thick, ->, >=stealth]

\begin{document}

\begin{tikzpicture}[node distance=1.7cm and 2.5cm]

% Top row
\node (excel1) [startstop] {Excel 1};
\node (excel2) [startstop, right=of excel1] {Excel 2};

% Cleaning
\node (clean1) [process, below=of excel1] {Clean};
\node (clean2) [process, below=of excel2] {Clean};

% Center anchor
\path (clean1) -- (clean2) coordinate[midway] (middle);

% Merge node centered under clean1 and clean2
\node (merge) [merge, below=1.7cm of middle] {Merge};

% Output nodes
\node (consolidated) [startstop, below=of merge] {Consolidated Excel};
\node (anomaly) [startstop, below=of consolidated] {Anomaly Detection};

% Arrows
\draw [arrow] (excel1) -- (clean1);
\draw [arrow] (excel2) -- (clean2);
\draw [arrow] (clean1) -- (merge);
```



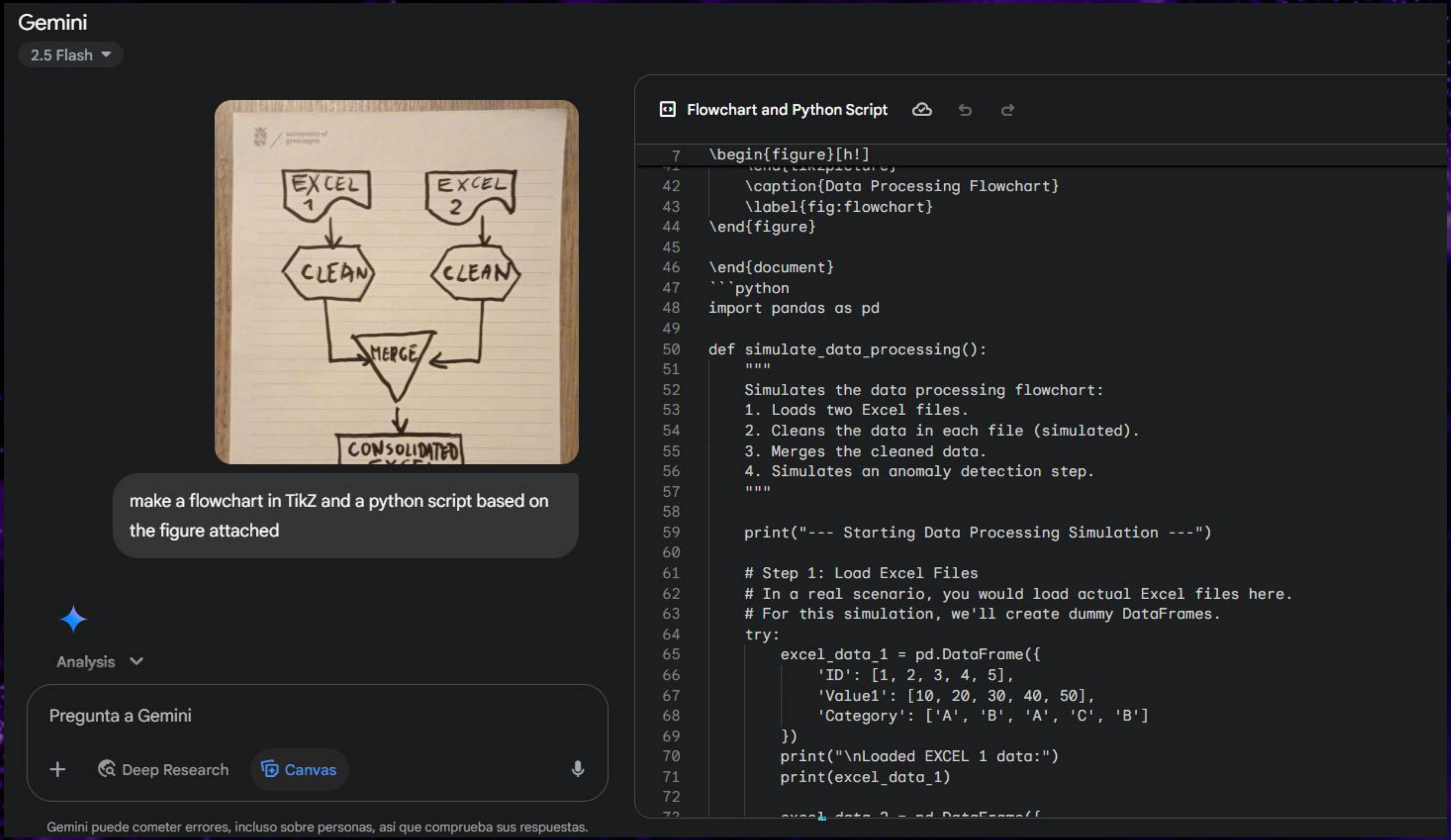


Python Script:

```python
import pandas as pd
from sklearn.ensemble import IsolationForest

# === Step 1: Load Excel Files ===
df1 = pd.read_excel('excel1.xlsx')
df2 = pd.read_excel('excel2.xlsx')

# === Step 2: Clean Function ===
def clean_data(df):
    # Remove text qualifiers, trim spaces, drop empty rows
    df.columns = df.columns.str.strip().str.replace('"', '')
    for col in df.select_dtypes(include='object').columns:
        df[col] = df[col].str.strip().str.replace('"', '')
    df.dropna(how='all', inplace=True)
    return df

df1_clean = clean_data(df1)
df2_clean = clean_data(df2)

# === Step 3: Merge Files ===
consolidated = pd.concat([df1_clean, df2_clean], ignore_index=True)

# === Step 4: Save Consolidated Excel ===
consolidated.to_excel('consolidated.xlsx', index=False)

# === Step 5: Anomaly Detection ===
# Assume numerical columns for anomaly detection
num_cols = consolidated.select_dtypes(include='number').columns
iso_forest = IsolationForest(contamination=0.05, random_state=42)
consolidated['anomaly'] = iso_forest.fit_predi (consolidated[num_cols])
```

Vibe coding with Python

Automatizing flowchart’s design with Gemini



make a flowchart in TikZ and a python script based on the figure attached

~~~
\begin{figure}[h!]
    \caption{Data Processing Flowchart}
    \label{fig:flowchart}
\end{figure}

\end{document}
```python
import pandas as pd

def simulate_data_processing():
    """
    Simulates the data processing flowchart:
    1. Loads two Excel files.
    2. Cleans the data in each file (simulated).
    3. Merges the cleaned data.
    4. Simulates an anomaly detection step.
    """

    print("--- Starting Data Processing Simulation ---")

    # Step 1: Load Excel Files
    # In a real scenario, you would load actual Excel files here.
    # For this simulation, we'll create dummy DataFrames.
    try:
        excel_data_1 = pd.DataFrame({
            'ID': [1, 2, 3, 4, 5],
            'Value1': [10, 20, 30, 40, 50],
            'Category': ['A', 'B', 'A', 'C', 'B']
        })
        print("\nLoaded EXCEL 1 data:")
        print(excel_data_1)
~~~

Gemini puede cometer errores, incluso sobre personas, así que comprueba sus respuestas.

Grant Thornton Speaking Engagement

Vibe coding with Generative AI

Introduction

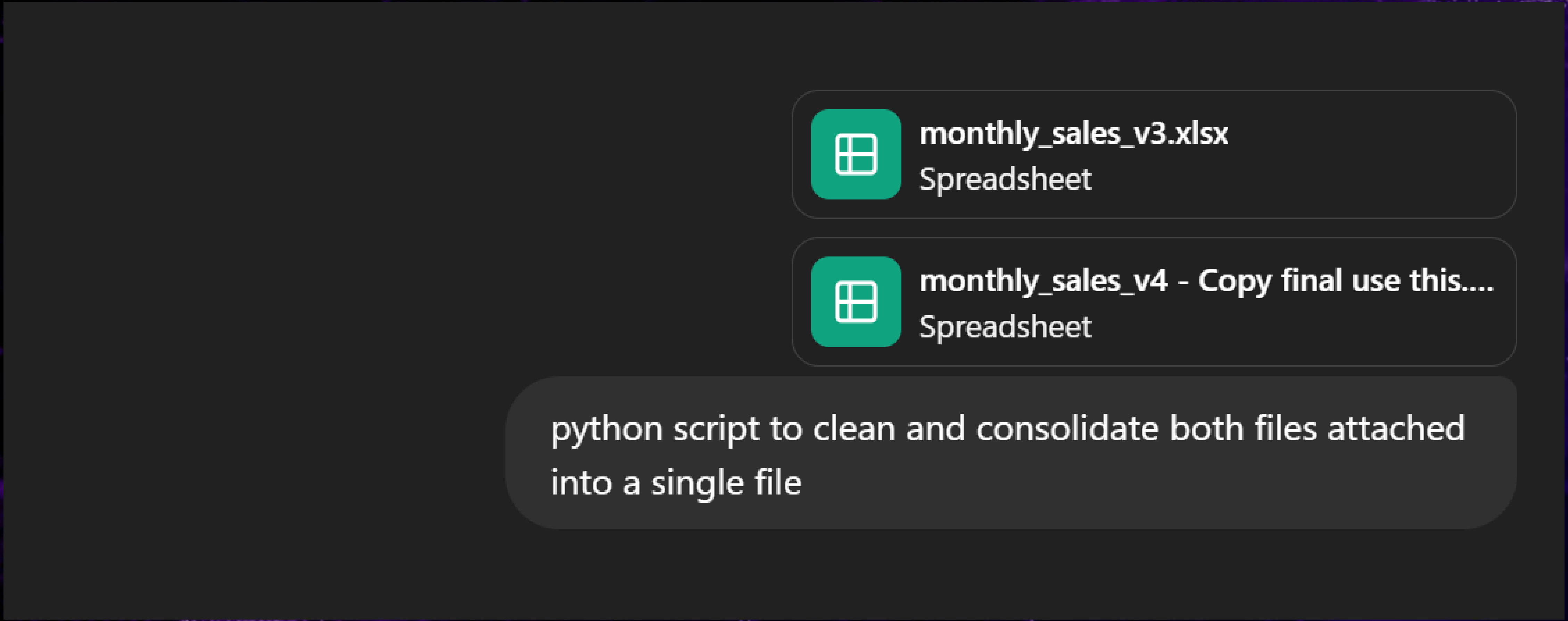
- AI, LLMs, Transformers, AI algorithms and models
- Vibe coding with Python in VS

Vibe coding with Python in Practice

- Automatizing flowchart's design/coding with AI visual recognition
- Vibe coding with ChatGPT, DeepSeek and CoPilot
- Vibe coding with local LLMs for sensitive information

Discussion

Vibe coding with Python and ChatGPT

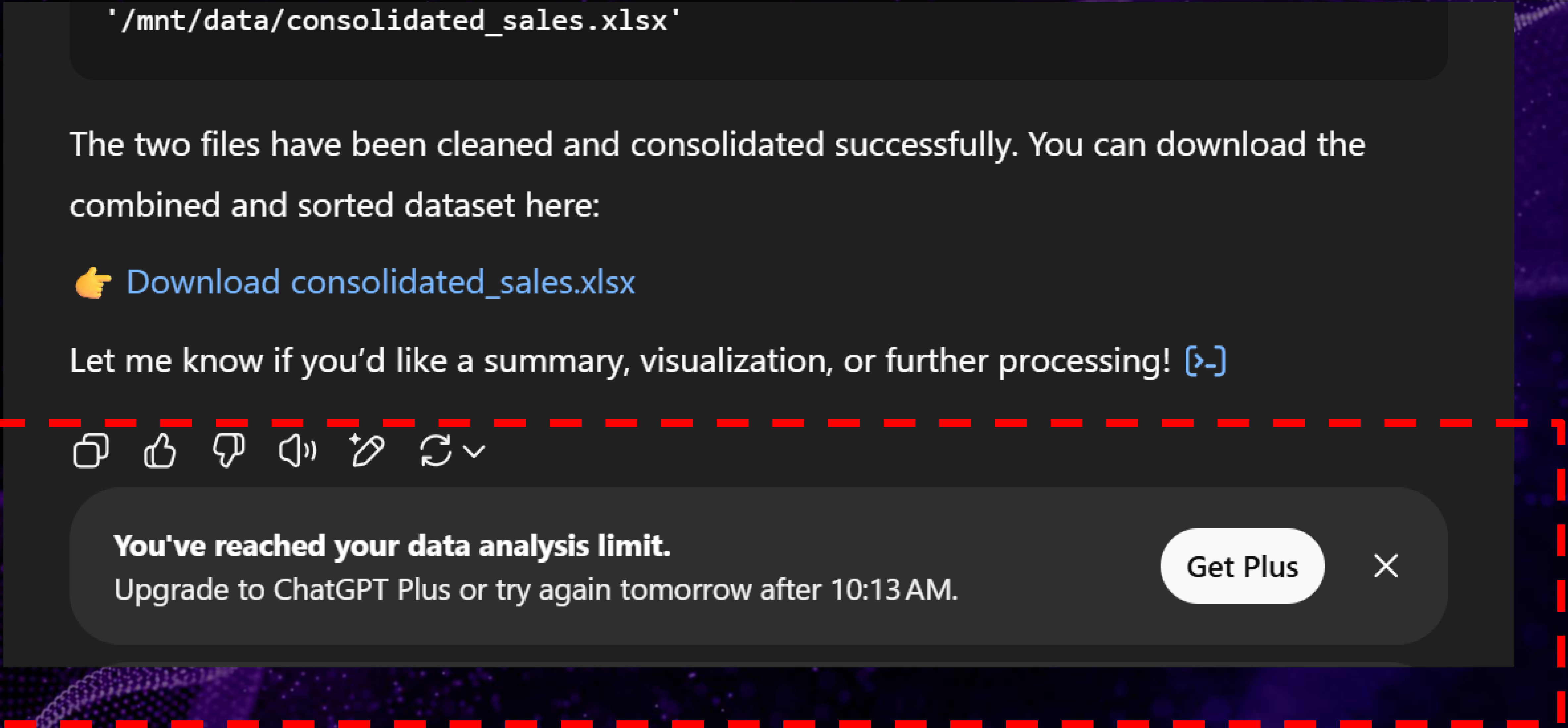


monthly_sales_v3.xlsx
Spreadsheet

monthly_sales_v4 - Copy final use this....
Spreadsheet

python script to clean and consolidate both files attached into a single file

Vibe coding with Python and ChatGPT



```
'/mnt/data/consolidated_sales.xlsx'
```

The two files have been cleaned and consolidated successfully. You can download the combined and sorted dataset here:

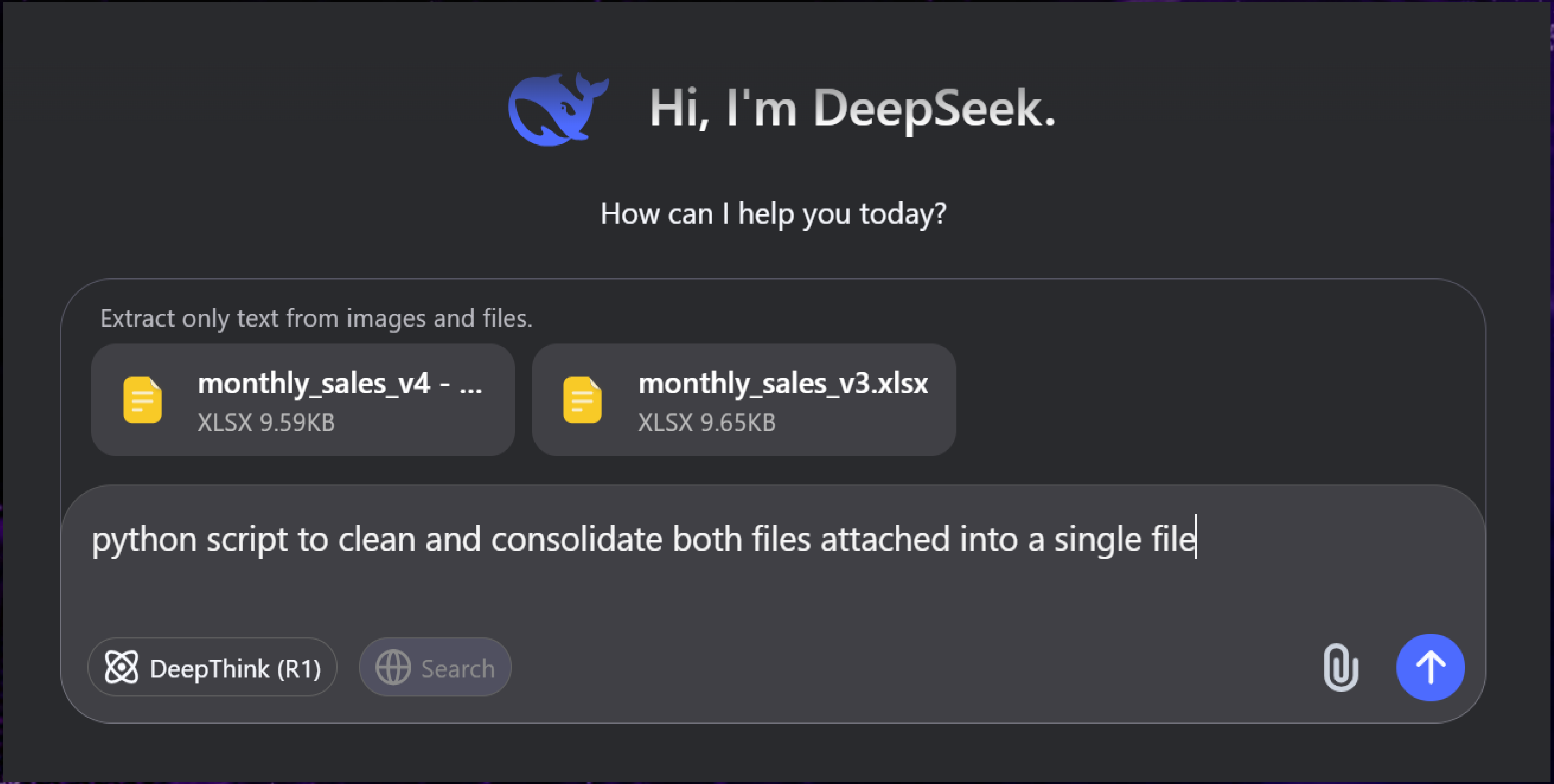
👉 Download consolidated_sales.xlsx

Let me know if you'd like a summary, visualization, or further processing!

You've reached your data analysis limit.

Upgrade to ChatGPT Plus or try again tomorrow after 10:13 AM.

Vibe coding with Python and DeepSeek



Hi, I'm DeepSeek.

How can I help you today?

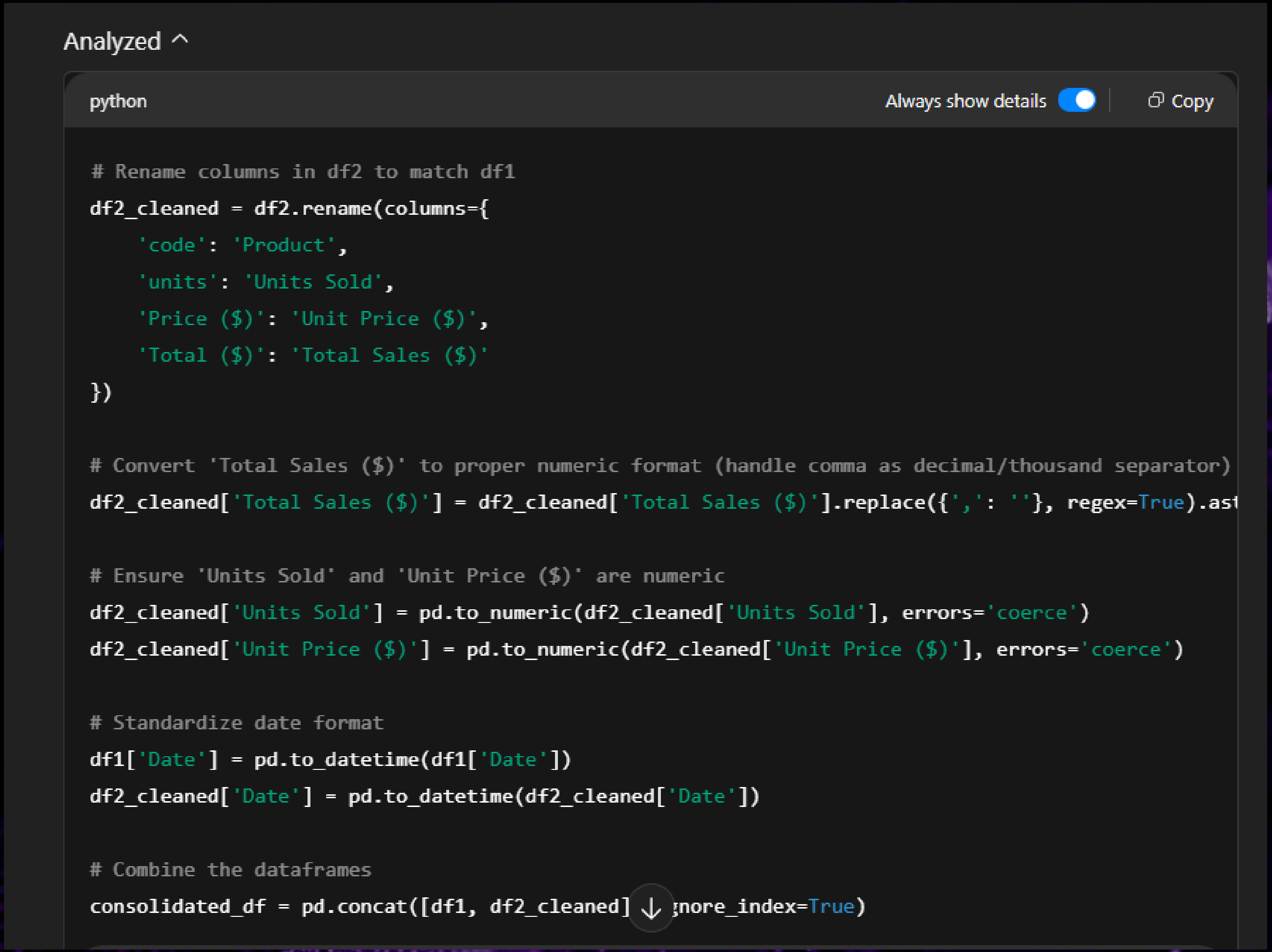
Extract only text from images and files.

monthly_sales_v4 - ...
XLSX 9.59KB

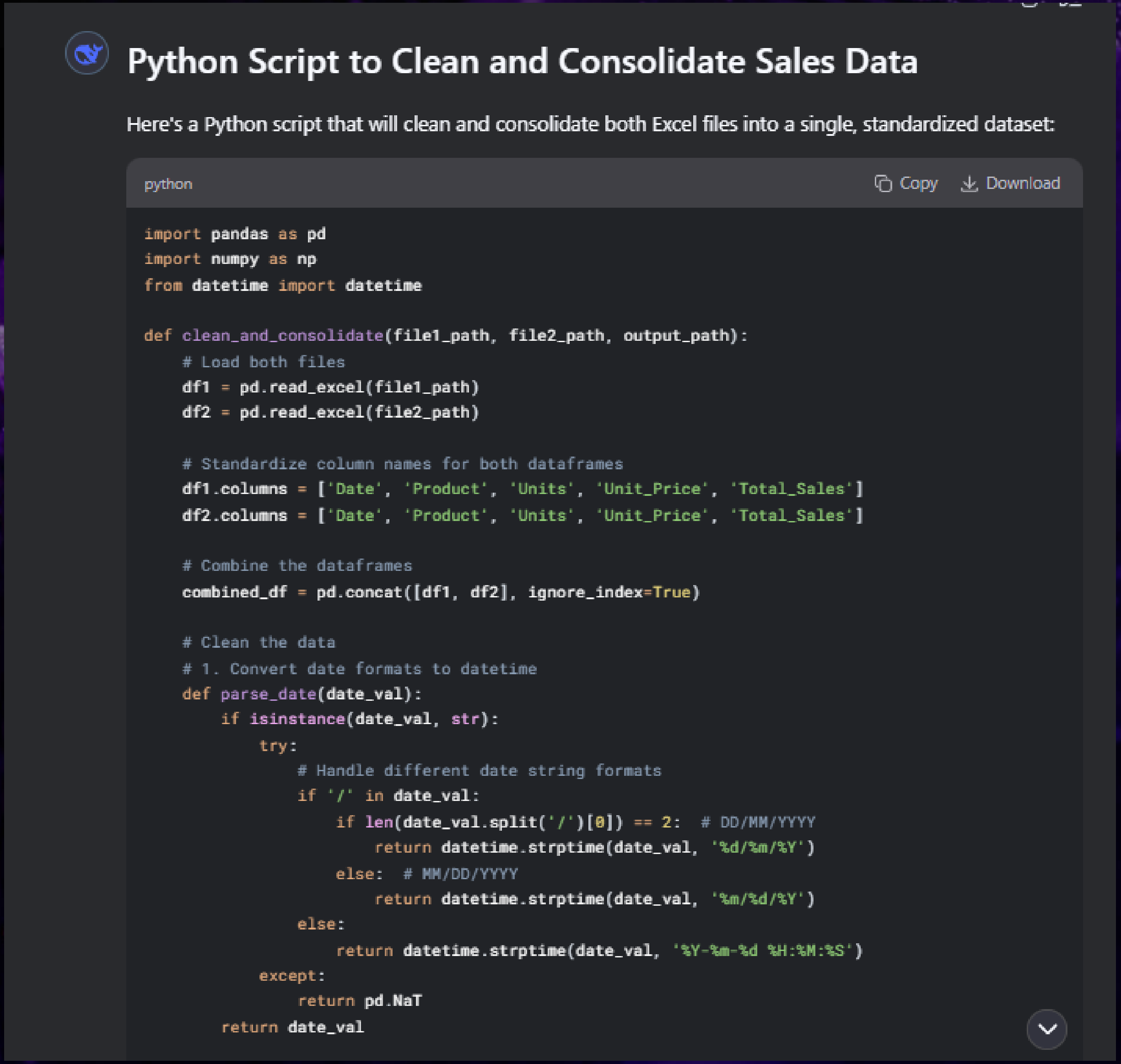
monthly_sales_v3.xlsx
XLSX 9.65KB

python script to clean and consolidate both files attached into a single file

Vibe coding with Python



```python
# Rename columns in df2 to match df1
df2_cleaned = df2.rename(columns={
    'code': 'Product',
    'units': 'Units Sold',
    'Price ($)': 'Unit Price ($)',
    'Total ($)': 'Total Sales ($)'
})

# Convert 'Total Sales ($)' to proper numeric format (handle comma as decimal/thousand separator)
df2_cleaned['Total Sales ($)'] = df2_cleaned['Total Sales ($)'].replace({',': ''}, regex=True).ast

# Ensure 'Units Sold' and 'Unit Price ($)' are numeric
df2_cleaned['Units Sold'] = pd.to_numeric(df2_cleaned['Units Sold'], errors='coerce')
df2_cleaned['Unit Price ($)'] = pd.to_numeric(df2_cleaned['Unit Price ($)'], errors='coerce')

# Standardize date format
df1['Date'] = pd.to_datetime(df1['Date'])
df2_cleaned['Date'] = pd.to_datetime(df2_cleaned['Date'])

# Combine the dataframes
consolidated_df = pd.concat([df1, df2_cleaned]  gnore_index=True)
```

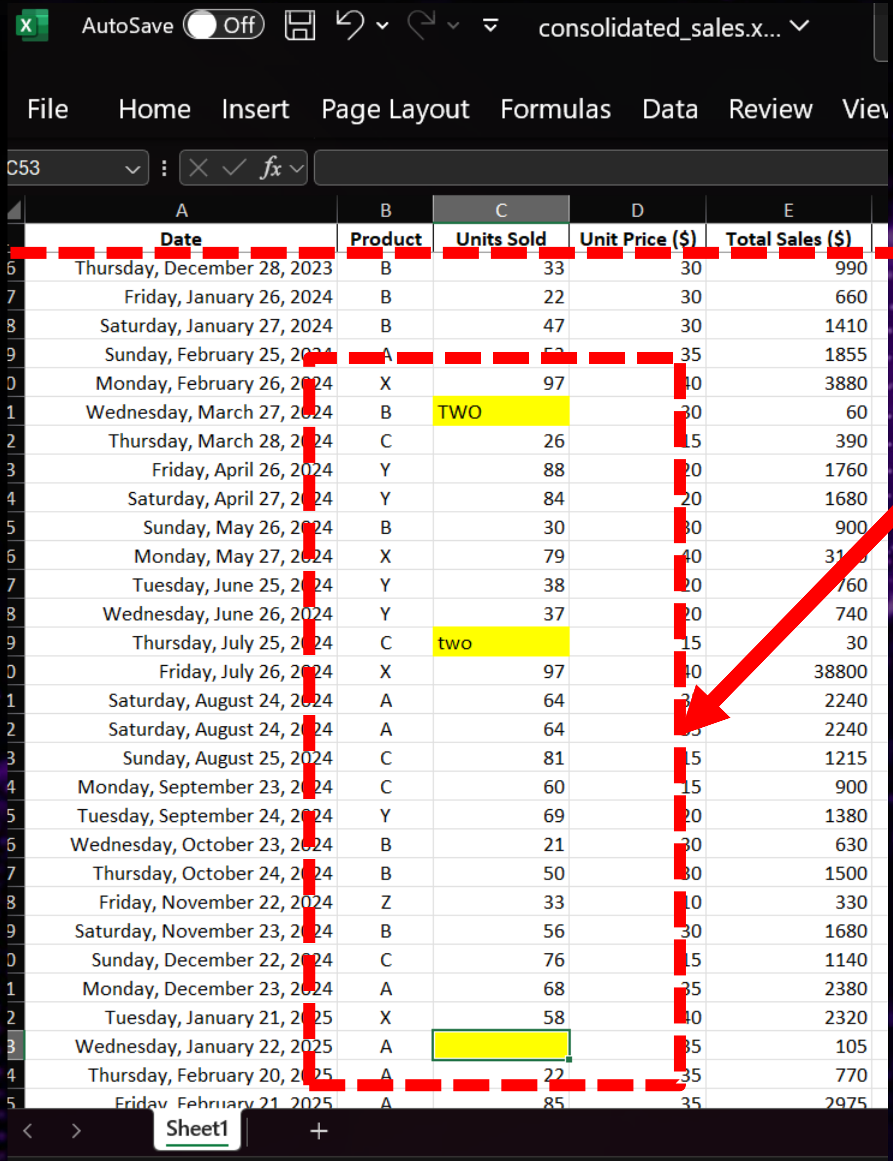


Python Script to Clean and Consolidate Sales Data

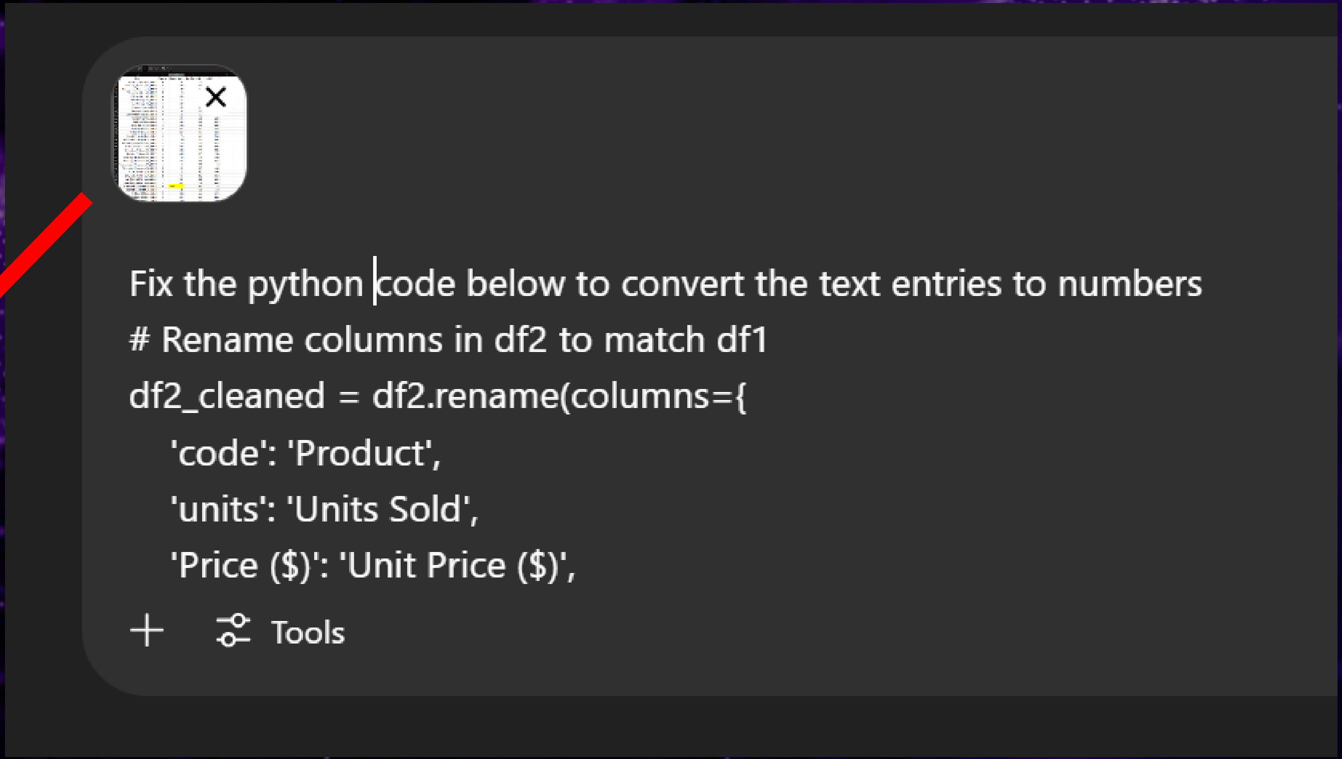
Here's a Python script that will clean and consolidate both Excel files into a single, standardized dataset:

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

def clean_and_consolidate(file1_path, file2_path, output_path):
    # Load both files
    df1 = pd.read_excel(file1_path)
    df2 = pd.read_excel(file2_path)

    # Standardize column names for both dataframes
    df1.columns = ['Date', 'Product', 'Units', 'Unit_Price', 'Total_Sales']
    df2.columns = ['Date', 'Product', 'Units', 'Unit_Price', 'Total_Sales']

    # Combine the dataframes
    combined_df = pd.concat([df1, df2], ignore_index=True)

    # Clean the data
    # 1. Convert date formats to datetime
    def parse_date(date_val):
        if isinstance(date_val, str):
            try:
                # Handle different date string formats
                if '/' in date_val:
                    if len(date_val.split('/')[0]) == 2:  # DD/MM/YYYY
                        return datetime.strptime(date_val, '%d/%m/%Y')
                    else:  # MM/DD/YYYY
                        return datetime.strptime(date_val, '%m/%d/%Y')
                else:
                    return datetime.strptime(date_val, '%Y-%m-%d %H:%M:%S')
            except:
                return pd.NaT
        return date_val
```

Vibe coding with Python and ChatGPT



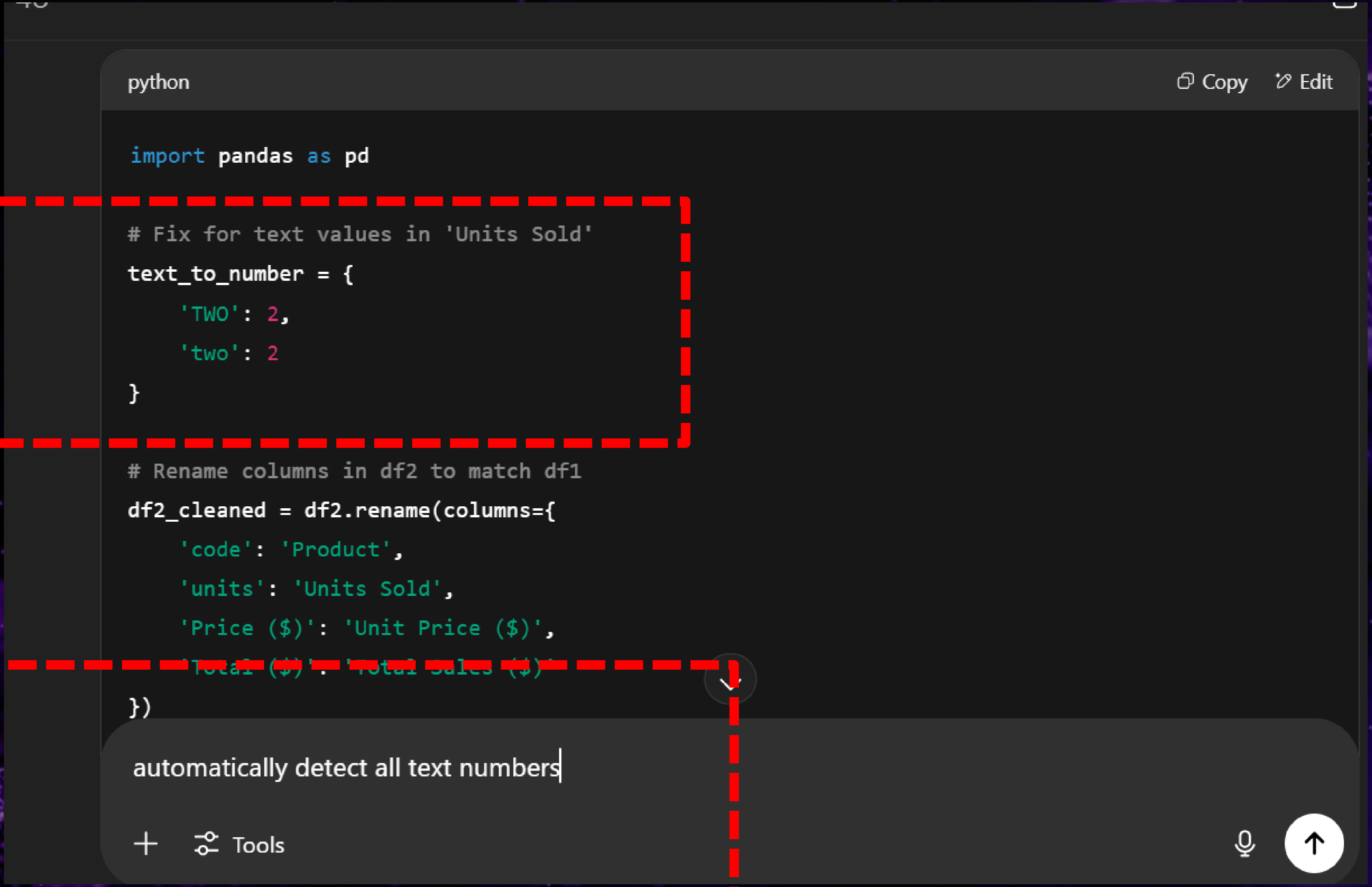
Date	Product	Units Sold	Unit Price ($)	Total Sales ($)
Thursday, December 28, 2023	B	33	30	990
Friday, January 26, 2024	B	22	30	660
Saturday, January 27, 2024	B	47	30	1410
Sunday, February 25, 2024	A	53	35	1855
Monday, February 26, 2024	X	97	40	3880
Wednesday, March 27, 2024	B	TWO	30	60
Thursday, March 28, 2024	C	26	15	390
Friday, April 26, 2024	Y	88	20	1760
Saturday, April 27, 2024	Y	84	20	1680
Sunday, May 26, 2024	B	30	30	900
Monday, May 27, 2024	X	79	40	[illegible]
Tuesday, June 25, 2024	Y	38	20	760
Wednesday, June 26, 2024	Y	37	20	740
Thursday, July 25, 2024	C	two	15	30
Friday, July 26, 2024	X	97	40	38800
Saturday, August 24, 2024	A	64	35	2240
Saturday, August 24, 2024	A	64	35	2240
Sunday, August 25, 2024	C	81	15	1215
Monday, September 23, 2024	C	60	15	900
Tuesday, September 24, 2024	Y	69	20	1380
Wednesday, October 23, 2024	B	21	30	630
Thursday, October 24, 2024	B	50	30	1500
Friday, November 22, 2024	Z	33	10	330
Saturday, November 23, 2024	B	56	30	1680
Sunday, December 22, 2024	C	76	15	1140
Monday, December 23, 2024	A	68	35	2380
Tuesday, January 21, 2025	X	58	40	2320
Wednesday, January 22, 2025	A		35	105
Thursday, February 20, 2025	A	22	35	770
Friday, February 21, 2025	A	85	35	2975



Fix the python code below to convert the text entries to numbers

```
# Rename columns in df2 to match df1
df2_cleaned = df2.rename(columns={
    'code': 'Product',
    'units': 'Units Sold',
    'Price ($)': 'Unit Price ($)',
```

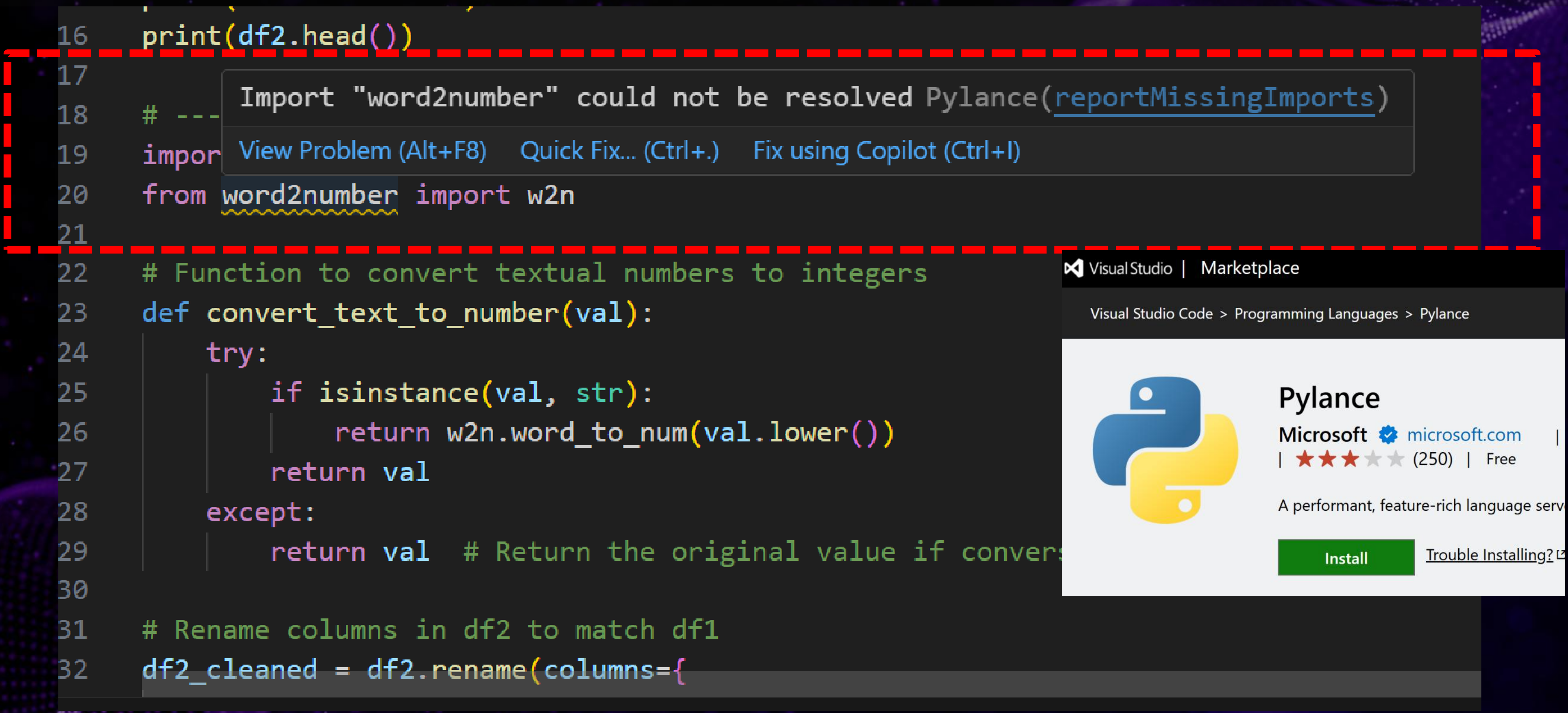
Vibe coding with Python and ChatGPT



```python
import pandas as pd

# Fix for text values in 'Units Sold'
text_to_number = {
    'TWO': 2,
    'two': 2
}

# Rename columns in df2 to match df1
df2_cleaned = df2.rename(columns={
    'code': 'Product',
    'units': 'Units Sold',
    'Price ($)': 'Unit Price ($)',
    'Total ($)': 'Total Sales ($)'
})
```

automatically detect all text numbers

Vibe coding with Python and ChatGPT



```
print(df2.head())

# ---
impor
from word2number import w2n

# Function to convert textual numbers to integers
def convert_text_to_number(val):
    try:
        if isinstance(val, str):
            return w2n.word_to_num(val.lower())
        return val
    except:
        return val  # Return the original value if convers

# Rename columns in df2 to match df1
df2_cleaned = df2.rename(columns={
```

Import "word2number" could not be resolved Pylance(reportMissingImports)

Visual Studio Code > Programming Languages > Pylance

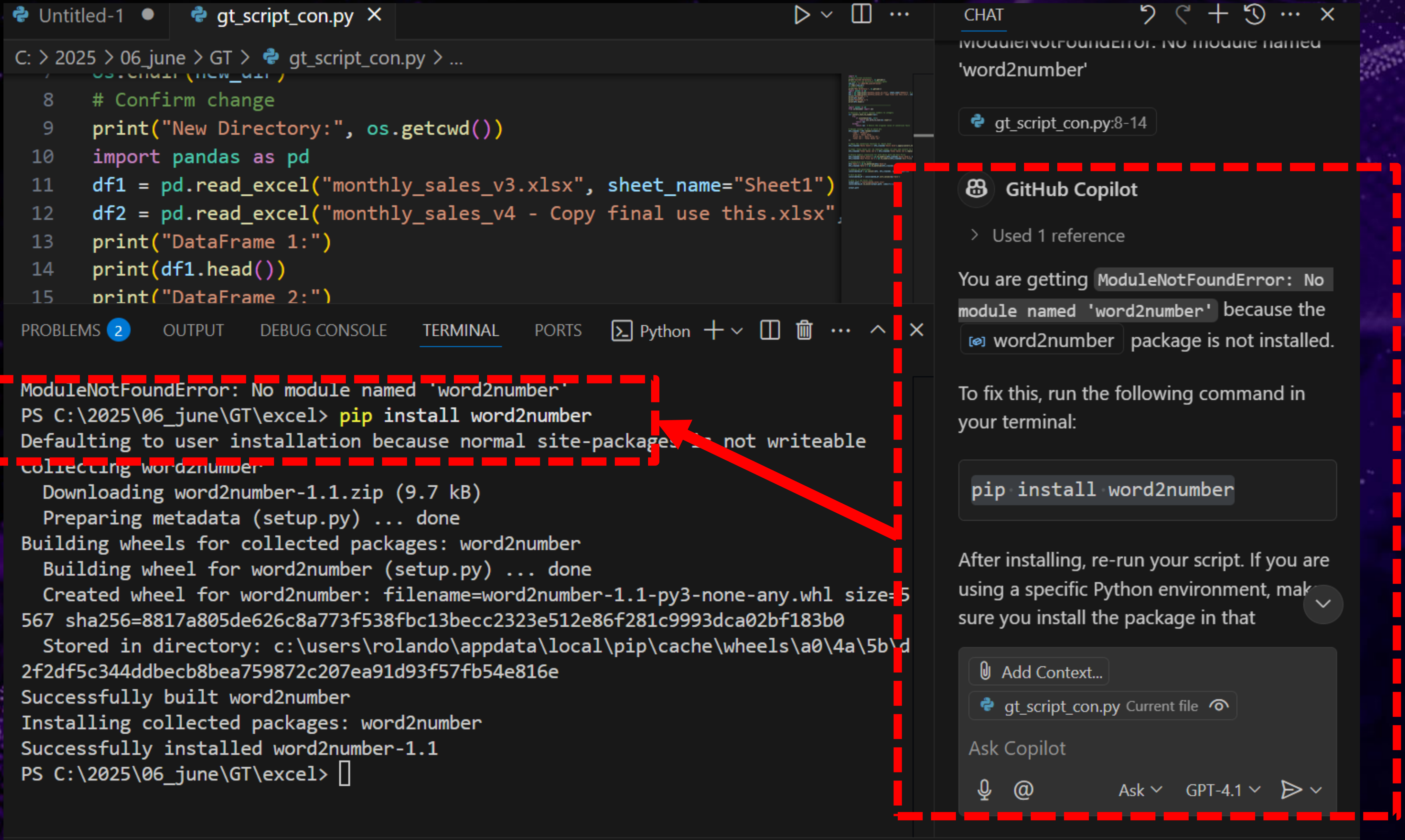
Pylance

Microsoft microsoft.com |

| ★★★☆☆ (250) | Free

A performant, feature-rich language serv

Vibe coding with Python and Copilot



```
# Confirm change
print("New Directory:", os.getcwd())
import pandas as pd
df1 = pd.read_excel("monthly_sales_v3.xlsx", sheet_name="Sheet1")
df2 = pd.read_excel("monthly_sales_v4 - Copy final use this.xlsx",
print("DataFrame 1:")
print(df1.head())
print("DataFrame 2:")
```

```
ModuleNotFoundError: No module named 'word2number'
PS C:\2025\06_june\GT\excel> pip install word2number
Defaulting to user installation because normal site-packages is not writeable
Collecting word2number
  Downloading word2number-1.1.zip (9.7 kB)
  Preparing metadata (setup.py) ... done
Building wheels for collected packages: word2number
  Building wheel for word2number (setup.py) ... done
  Created wheel for word2number: filename=word2number-1.1-py3-none-any.whl size=5
567 sha256=8817a805de626c8a773f538fbc13becc2323e512e86f281c9993dca02bf183b0
  Stored in directory: c:\users\rolando\appdata\local\pip\cache\wheels\a0\4a\5b\d
2f2df5c344ddbecb8bea759872c207ea91d93f57fb54e816e
Successfully built word2number
Installing collected packages: word2number
Successfully installed word2number-1.1
PS C:\2025\06_june\GT\excel> 
```

ModuleNotFoundError: No module named 'word2number'

gt_script_con.py:8-14

GitHub Copilot

Used 1 reference

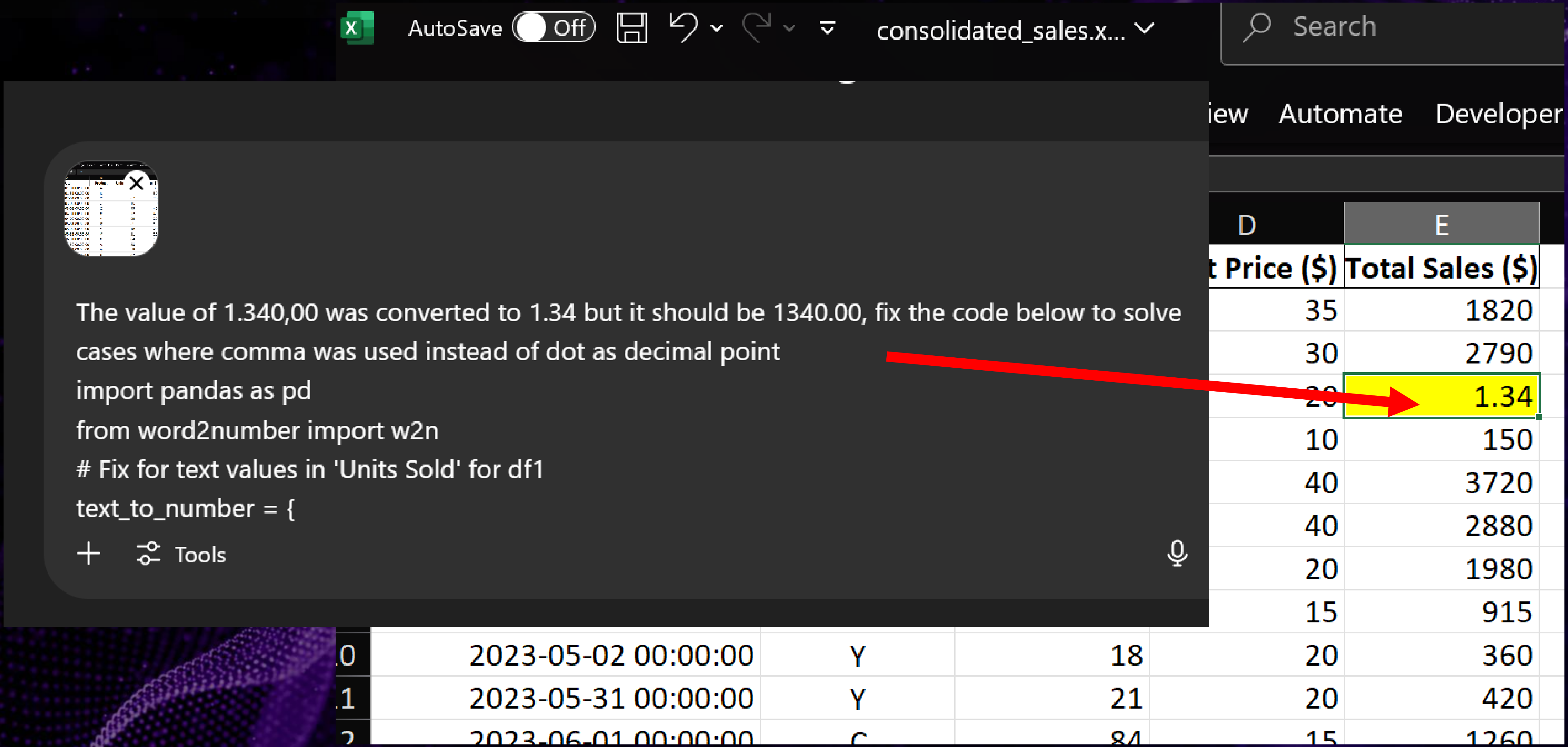
You are getting `ModuleNotFoundError: No module named 'word2number'` because the word2number package is not installed.

To fix this, run the following command in your terminal:

```
pip install word2number
```

After installing, re-run your script. If you are using a specific Python environment, make sure you install the package in that

Vibe coding with Python

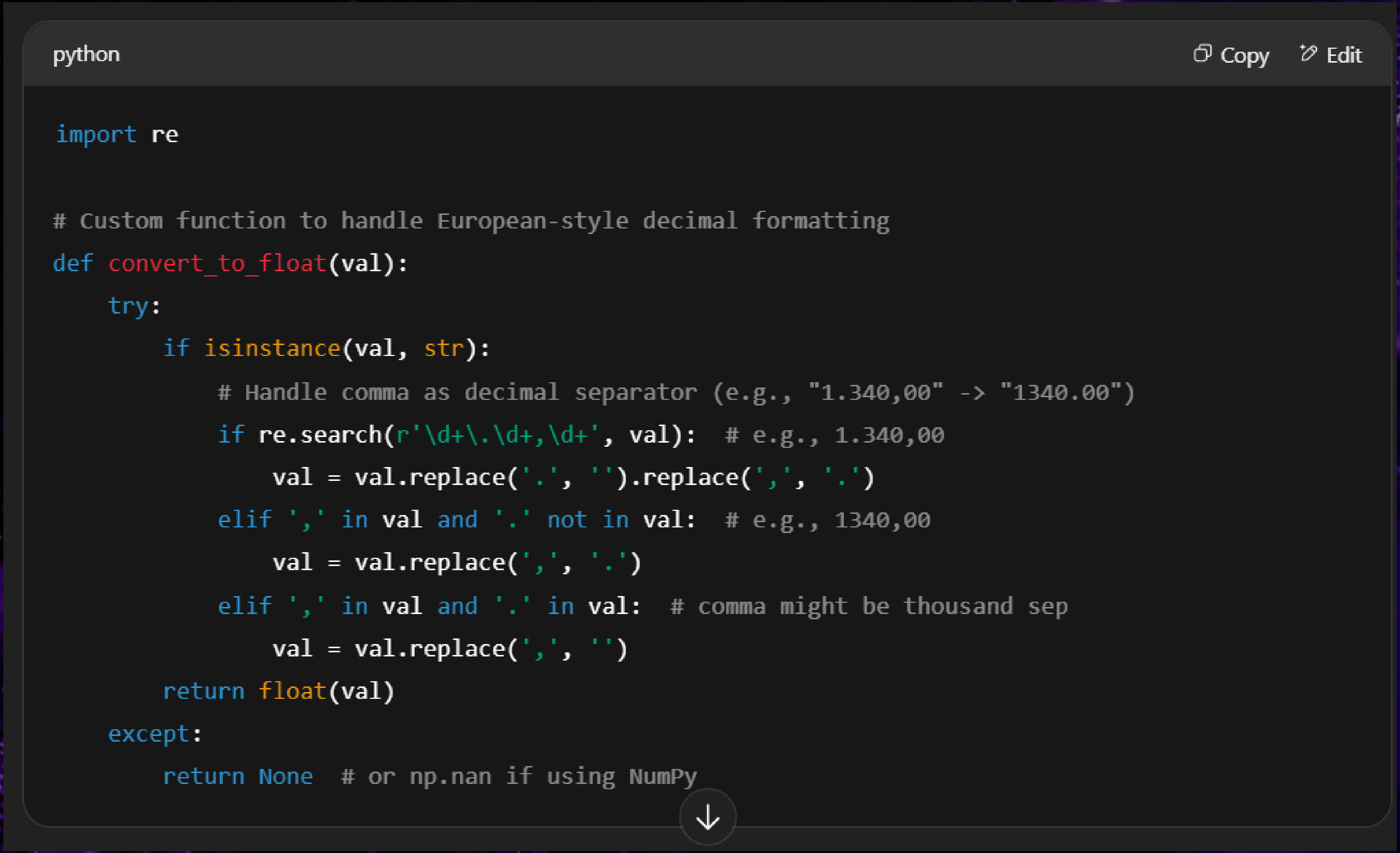


The value of 1.340,00 was converted to 1.34 but it should be 1340.00, fix the code below to solve cases where comma was used instead of dot as decimal point

```
import pandas as pd
from word2number import w2n
# Fix for text values in 'Units Sold' for df1
text_to_number = {
```

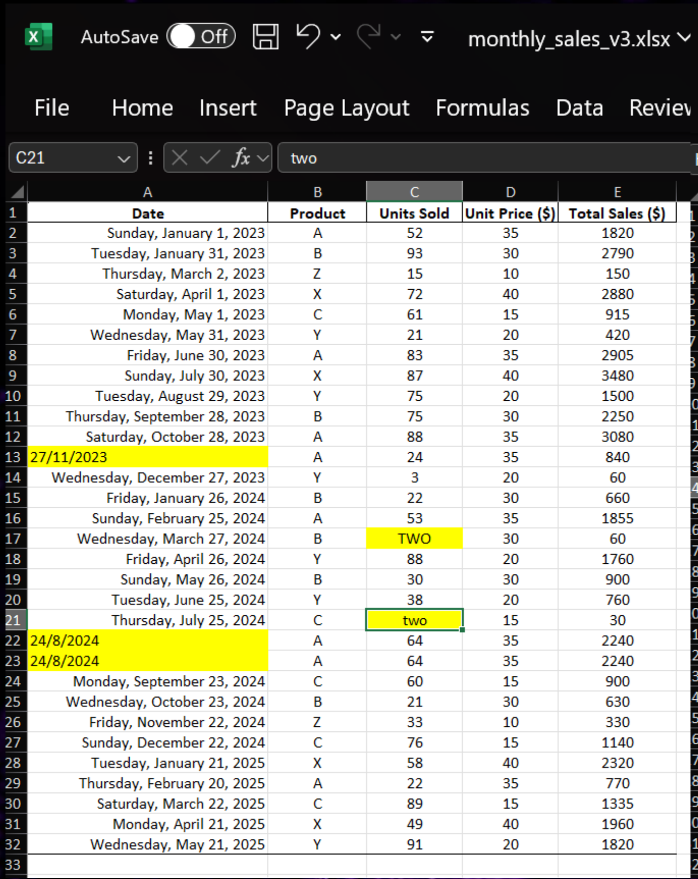
				D	E
				t Price ($)	Total Sales ($)
				35	1820
				30	2790
				20	1.34
				10	150
				40	3720
				40	2880
				20	1980
				15	915
0	2023-05-02 00:00:00	Y	18	20	360
1	2023-05-31 00:00:00	Y	21	20	420
2	2023-06-01 00:00:00	C	84	15	1260

Vibe coding with Python

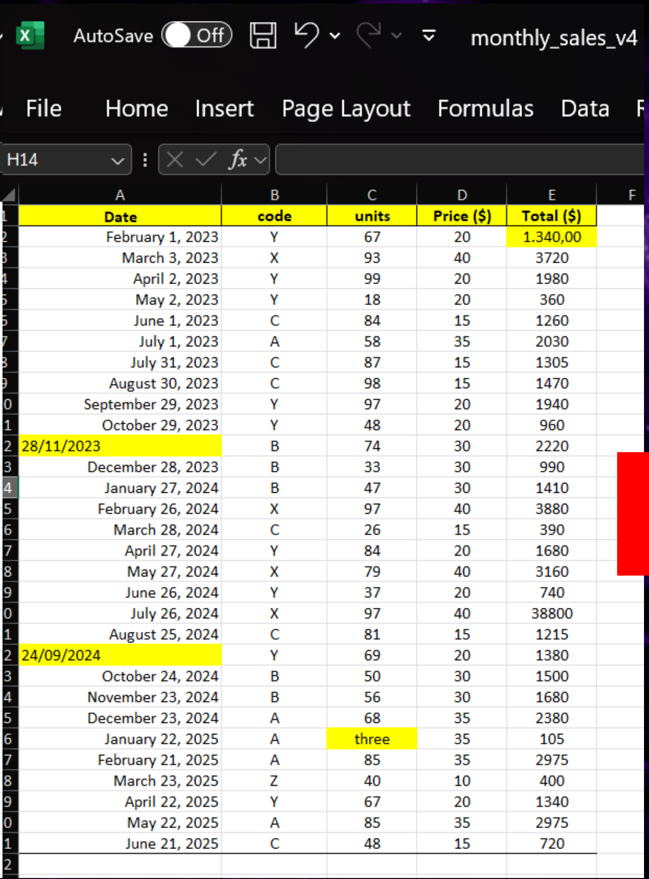


```python
import re

# Custom function to handle European-style decimal formatting
def convert_to_float(val):
    try:
        if isinstance(val, str):
            # Handle comma as decimal separator (e.g., "1.340,00" -> "1340.00")
            if re.search(r'\d+\.\d+,\d+', val):  # e.g., 1.340,00
                val = val.replace('.', '').replace(',', '.')
            elif ',' in val and '.' not in val:  # e.g., 1340,00
                val = val.replace(',', '.')
            elif ',' in val and '.' in val:  # comma might be thousand sep
                val = val.replace(',', '')
        return float(val)
    except:
        return None  # or np.nan if using NumPy
```

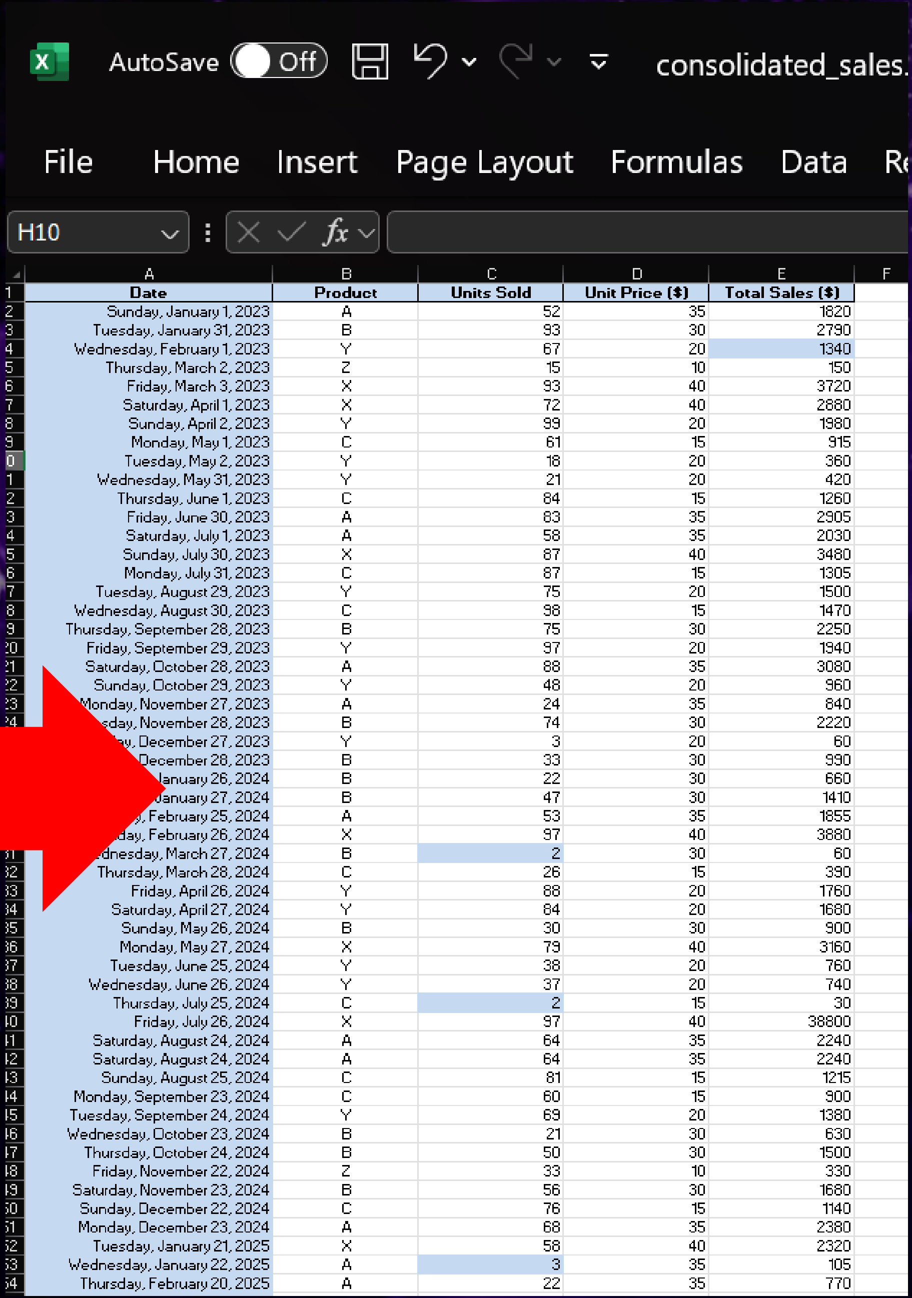
Vibe coding with Python



Date	Product	Units Sold	Unit Price ($)	Total Sales ($)
Sunday, January 1, 2023	A	52	35	1820
Tuesday, January 31, 2023	B	93	30	2790
Thursday, March 2, 2023	Z	15	10	150
Saturday, April 1, 2023	X	72	40	2880
Monday, May 1, 2023	C	61	15	915
Wednesday, May 31, 2023	Y	21	20	420
Friday, June 30, 2023	A	83	35	2905
Sunday, July 30, 2023	X	87	40	3480
Tuesday, August 29, 2023	Y	75	20	1500
Thursday, September 28, 2023	B	75	30	2250
Saturday, October 28, 2023	A	88	35	3080
27/11/2023	A	24	35	840
Wednesday, December 27, 2023	Y	3	20	60
Friday, January 26, 2024	B	22	30	660
Sunday, February 25, 2024	A	53	35	1855
Wednesday, March 27, 2024	B	TWO	30	60
Friday, April 26, 2024	Y	88	20	1760
Sunday, May 26, 2024	B	30	30	900
Tuesday, June 25, 2024	Y	38	20	760
Thursday, July 25, 2024	C	two	15	30
24/8/2024	A	64	35	2240
24/8/2024	A	64	35	2240
Monday, September 23, 2024	C	60	15	900
Wednesday, October 23, 2024	B	21	30	630
Friday, November 22, 2024	Z	33	10	330
Sunday, December 22, 2024	C	76	15	1140
Tuesday, January 21, 2025	X	58	40	2320
Thursday, February 20, 2025	A	22	35	770
Saturday, March 22, 2025	C	89	15	1335
Monday, April 21, 2025	X	49	40	1960
Wednesday, May 21, 2025	Y	91	20	1820

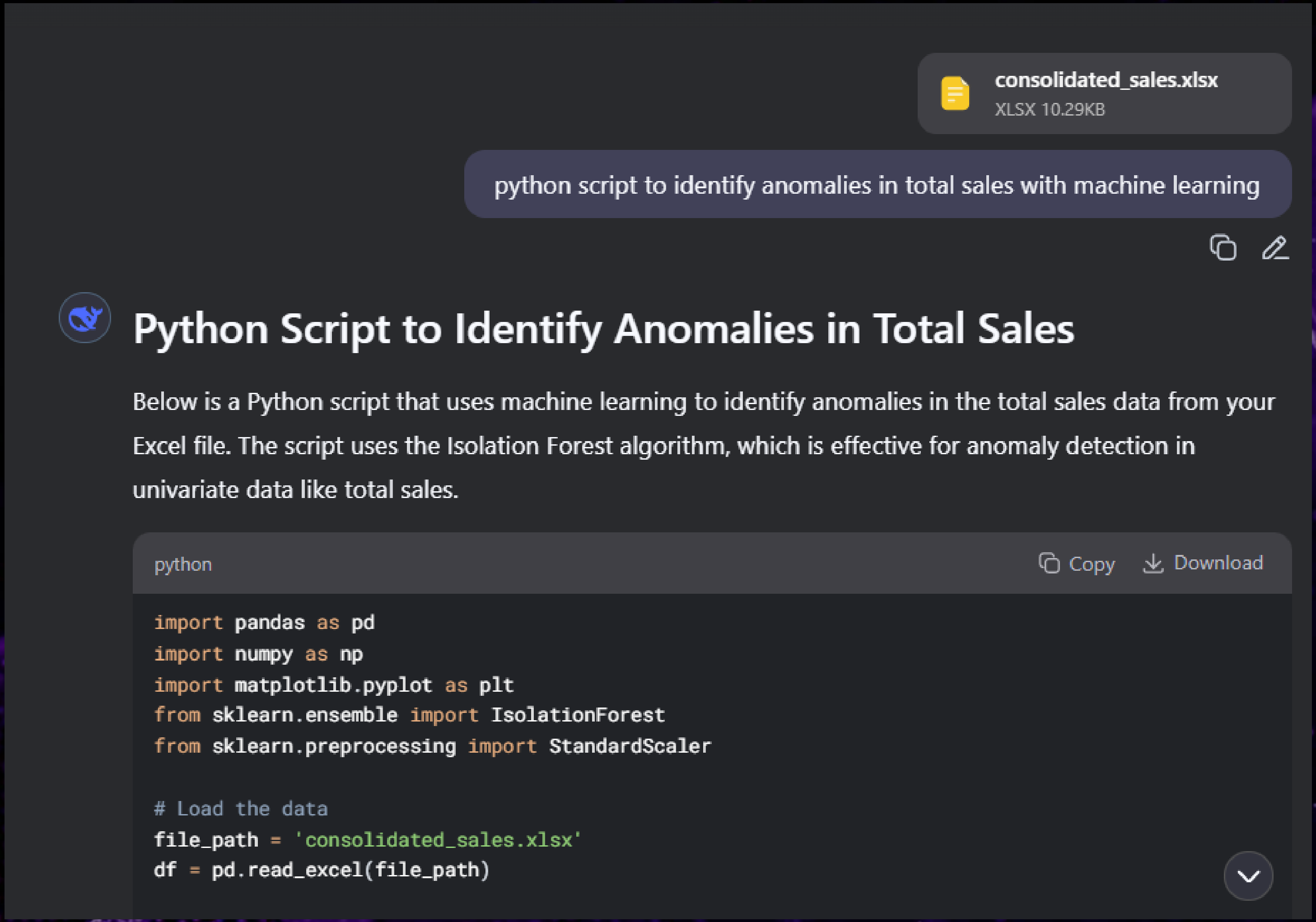


Date	code	units	Price ($)	Total ($)
February 1, 2023	Y	67	20	1.340,00
March 3, 2023	X	93	40	3720
April 2, 2023	Y	99	20	1980
May 2, 2023	Y	18	20	360
June 1, 2023	C	84	15	1260
July 1, 2023	A	58	35	2030
July 31, 2023	C	87	15	1305
August 30, 2023	C	98	15	1470
September 29, 2023	Y	97	20	1940
October 29, 2023	Y	48	20	960
28/11/2023	B	74	30	2220
December 28, 2023	B	33	30	990
January 27, 2024	B	47	30	1410
February 26, 2024	X	97	40	3880
March 28, 2024	C	26	15	390
April 27, 2024	Y	84	20	1680
May 27, 2024	X	79	40	3160
June 26, 2024	Y	37	20	740
July 26, 2024	X	97	40	38800
August 25, 2024	C	81	15	1215
24/09/2024	Y	69	20	1380
October 24, 2024	B	50	30	1500
November 23, 2024	B	56	30	1680
December 23, 2024	A	68	35	2380
January 22, 2025	A	three	35	105
February 21, 2025	A	85	35	2975
March 23, 2025	Z	40	10	400
April 22, 2025	Y	67	20	1340
May 22, 2025	A	85	35	2975
June 21, 2025	C	48	15	720



Date	Product	Units Sold	Unit Price ($)	Total Sales ($)
Sunday, January 1, 2023	A	52	35	1820
Tuesday, January 31, 2023	B	93	30	2790
Wednesday, February 1, 2023	Y	67	20	1340
Thursday, March 2, 2023	Z	15	10	150
Friday, March 3, 2023	X	93	40	3720
Saturday, April 1, 2023	X	72	40	2880
Sunday, April 2, 2023	Y	99	20	1980
Monday, May 1, 2023	C	61	15	915
Tuesday, May 2, 2023	Y	18	20	360
Wednesday, May 31, 2023	Y	21	20	420
Thursday, June 1, 2023	C	84	15	1260
Friday, June 30, 2023	A	83	35	2905
Saturday, July 1, 2023	A	58	35	2030
Sunday, July 30, 2023	X	87	40	3480
Monday, July 31, 2023	C	87	15	1305
Tuesday, August 29, 2023	Y	75	20	1500
Wednesday, August 30, 2023	C	98	15	1470
Thursday, September 28, 2023	B	75	30	2250
Friday, September 29, 2023	Y	97	20	1940
Saturday, October 28, 2023	A	88	35	3080
Sunday, October 29, 2023	Y	48	20	960
Monday, November 27, 2023	A	24	35	840
sday, November 28, 2023	B	74	30	2220
ay, December 27, 2023	Y	3	20	60
December 28, 2023	B	33	30	990
January 26, 2024	B	22	30	660
January 27, 2024	B	47	30	1410
, February 25, 2024	A	53	35	1855
day, February 26, 2024	X	97	40	3880
dnesday, March 27, 2024	B	2	30	60
Thursday, March 28, 2024	C	26	15	390
Friday, April 26, 2024	Y	88	20	1760
Saturday, April 27, 2024	Y	84	20	1680
Sunday, May 26, 2024	B	30	30	900
Monday, May 27, 2024	X	79	40	3160
Tuesday, June 25, 2024	Y	38	20	760
Wednesday, June 26, 2024	Y	37	20	740
Thursday, July 25, 2024	C	2	15	30
Friday, July 26, 2024	X	97	40	38800
Saturday, August 24, 2024	A	64	35	2240
Saturday, August 24, 2024	A	64	35	2240
Sunday, August 25, 2024	C	81	15	1215
Monday, September 23, 2024	C	60	15	900
Tuesday, September 24, 2024	Y	69	20	1380
Wednesday, October 23, 2024	B	21	30	630
Thursday, October 24, 2024	B	50	30	1500
Friday, November 22, 2024	Z	33	10	330
Saturday, November 23, 2024	B	56	30	1680
Sunday, December 22, 2024	C	76	15	1140
Monday, December 23, 2024	A	68	35	2380
Tuesday, January 21, 2025	X	58	40	2320
Wednesday, January 22, 2025	A	3	35	105
Thursday, February 20, 2025	A	22	35	770

Vibe coding with Python

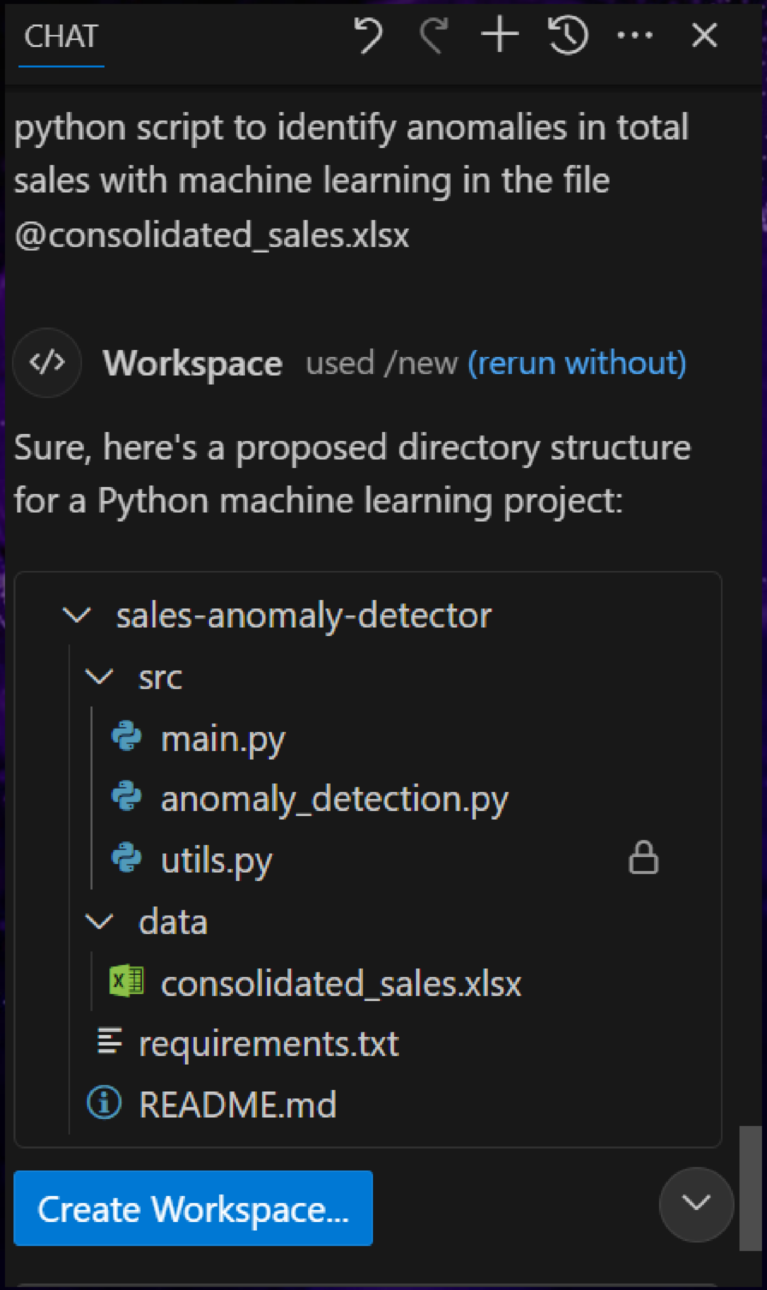


python script to identify anomalies in total sales with machine learning

Python Script to Identify Anomalies in Total Sales

Below is a Python script that uses machine learning to identify anomalies in the total sales data from your Excel file. The script uses the Isolation Forest algorithm, which is effective for anomaly detection in univariate data like total sales.

```python
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.ensemble import IsolationForest
from sklearn.preprocessing import StandardScaler

# Load the data
file_path = 'consolidated_sales.xlsx'
df = pd.read_excel(file_path)
```



python script to identify anomalies in total sales with machine learning in the file @consolidated_sales.xlsx

Workspace used /new (rerun without)

Sure, here's a proposed directory structure for a Python machine learning project:

- sales-anomaly-detector
  - src
    - main.py
    - anomaly_detection.py
    - utils.py
  - data
    - consolidated_sales.xlsx
  - requirements.txt
  - README.md

Grant Thornton Speaking Engagement

Vibe coding with Generative AI

Introduction

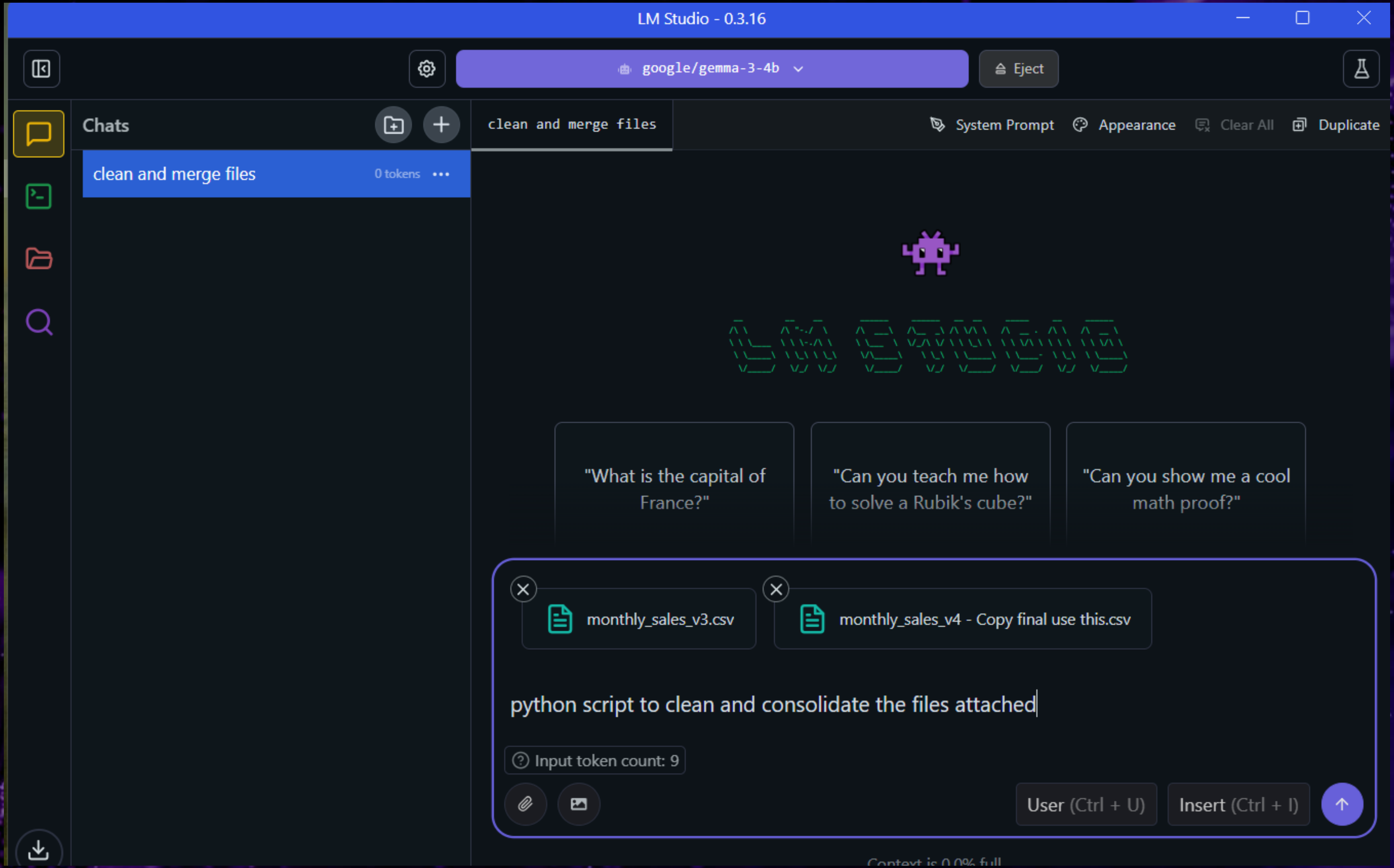
- AI, LLMs, Transformers, AI algorithms and models
- Vibe coding with Python in VS

Vibe coding with Python in Practice

- Automatizing flowchart’s design/coding with AI visual recognition
- Vibe coding with ChatGPT, DeepSeek and CoPilot
- Vibe coding with local LLMs for sensitive information

Discussion

Vibe coding with Python

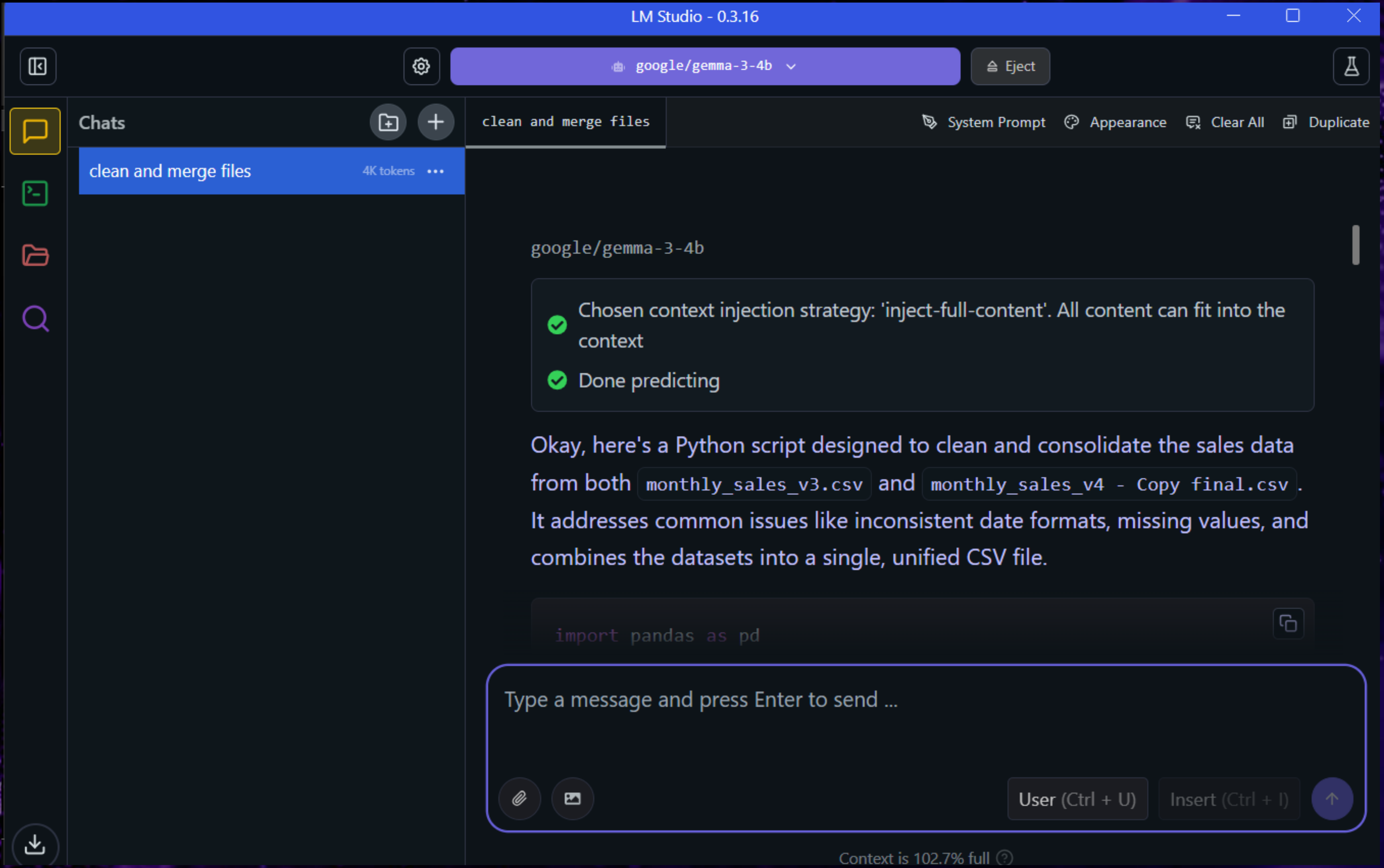


monthly_sales_v3.csv

monthly_sales_v4 - Copy final use this.csv

python script to clean and consolidate the files attached

Vibe coding with Python



google/gemma-3-4b

- ✓ Chosen context injection strategy: 'inject-full-content'. All content can fit into the context
- ✓ Done predicting

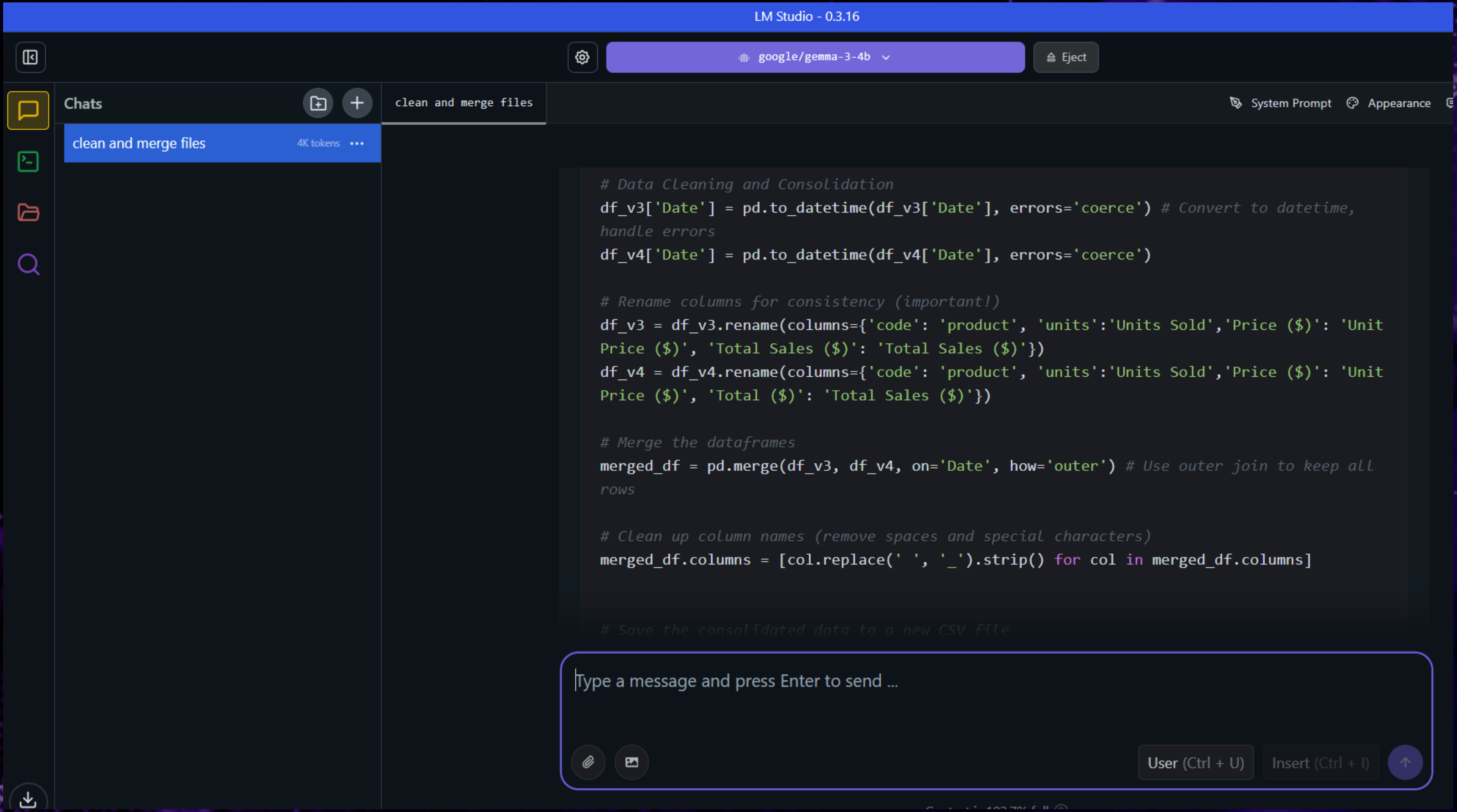
Okay, here's a Python script designed to clean and consolidate the sales data from both `monthly_sales_v3.csv` and `monthly_sales_v4 - Copy final.csv`. It addresses common issues like inconsistent date formats, missing values, and combines the datasets into a single, unified CSV file.

```
import pandas as pd
```

Type a message and press Enter to send ...

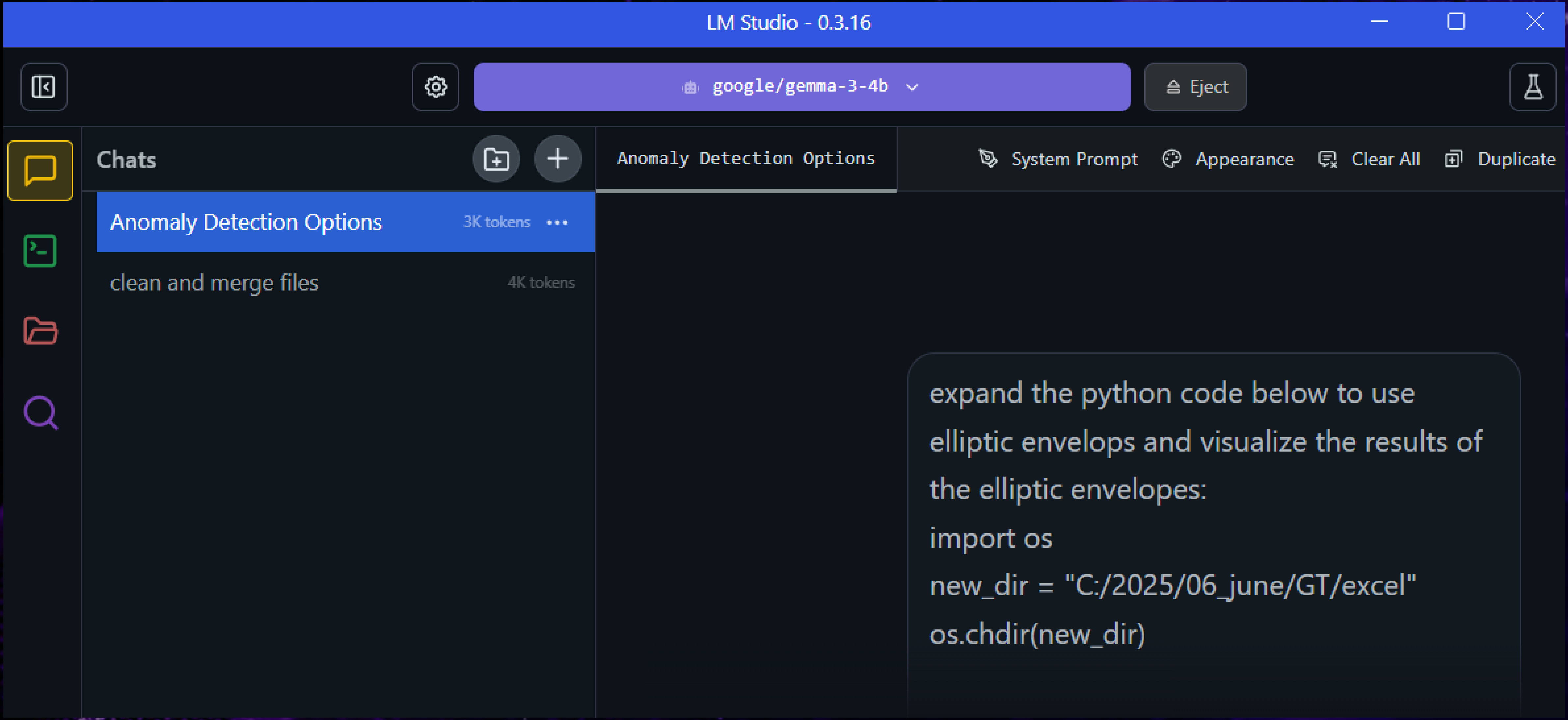
Context is 102.7% full

Vibe coding with Python



```python
# Data Cleaning and Consolidation
df_v3['Date'] = pd.to_datetime(df_v3['Date'], errors='coerce') # Convert to datetime, handle errors
df_v4['Date'] = pd.to_datetime(df_v4['Date'], errors='coerce')

# Rename columns for consistency (important!)
df_v3 = df_v3.rename(columns={'code': 'product', 'units':'Units Sold','Price ($)': 'Unit Price ($)', 'Total Sales ($)': 'Total Sales ($)'})
df_v4 = df_v4.rename(columns={'code': 'product', 'units':'Units Sold','Price ($)': 'Unit Price ($)', 'Total ($)': 'Total Sales ($)'})

# Merge the dataframes
merged_df = pd.merge(df_v3, df_v4, on='Date', how='outer') # Use outer join to keep all rows

# Clean up column names (remove spaces and special characters)
merged_df.columns = [col.replace(' ', '_').strip() for col in merged_df.columns]


# Save the consolidated data to a new CSV file
```

Vibe coding with Python

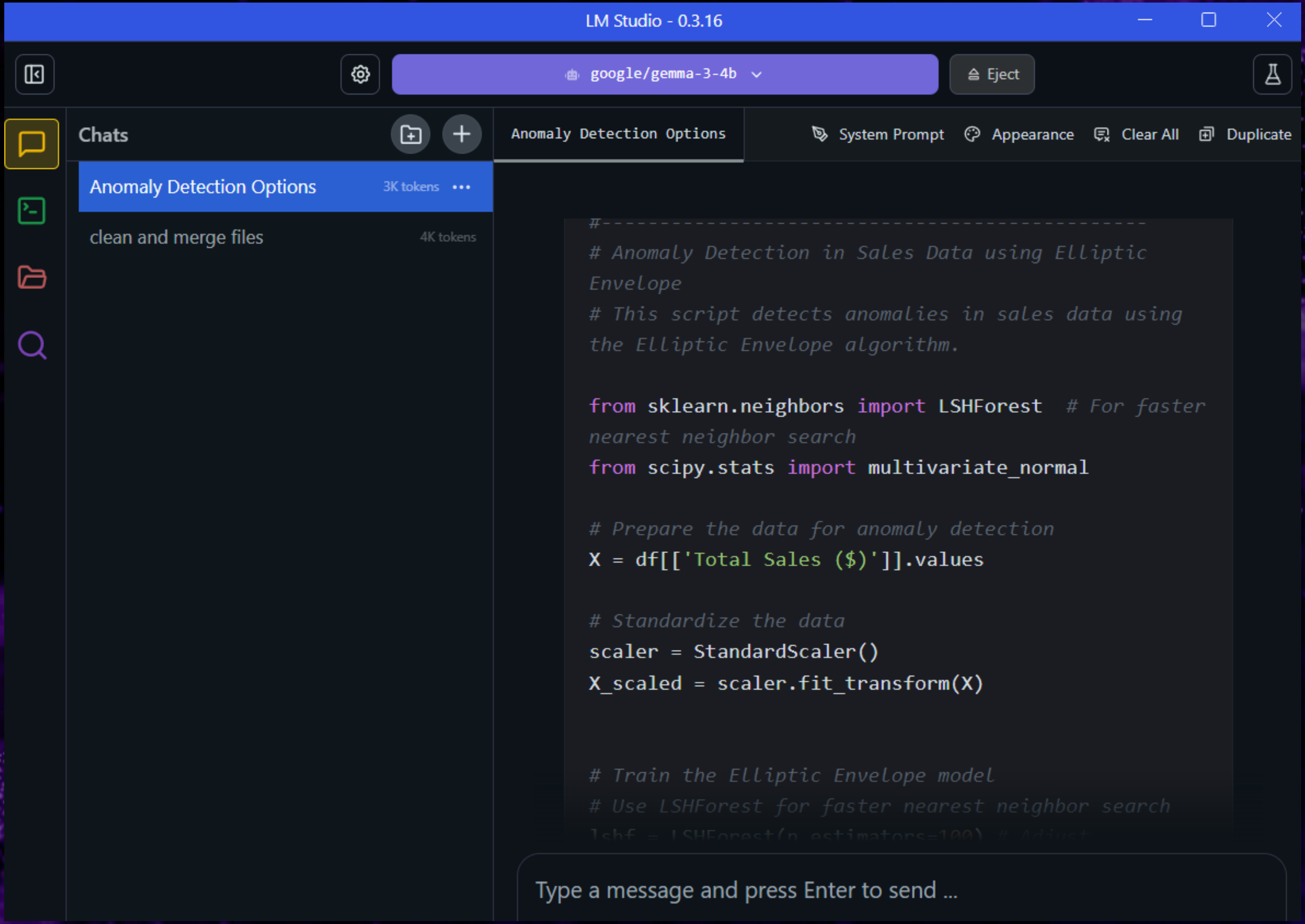


LM Studio - 0.3.16
google/gemma-3-4b
Eject
Chats
Anomaly Detection Options 3K tokens
clean and merge files 4K tokens
Anomaly Detection Options
System Prompt
Appearance
Clear All
Duplicate

expand the python code below to use elliptic envelops and visualize the results of the elliptic envelopes:

```
import os
new_dir = "C:/2025/06_june/GT/excel"
os.chdir(new_dir)
```

Vibe coding with Python


LM Studio - 0.3.16
google/gemma-3-4b
Eject
Chats
Anomaly Detection Options 3K tokens
clean and merge files 4K tokens
Anomaly Detection Options
System Prompt
Appearance
Clear All
Duplicate
Type a message and press Enter to send ...

```python
#-----------------------------------------------
# Anomaly Detection in Sales Data using Elliptic
Envelope
# This script detects anomalies in sales data using
the Elliptic Envelope algorithm.

from sklearn.neighbors import LSHForest  # For faster
nearest neighbor search
from scipy.stats import multivariate_normal

# Prepare the data for anomaly detection
X = df[['Total Sales ($)']].values

# Standardize the data
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)


# Train the Elliptic Envelope model
# Use LSHForest for faster nearest neighbor search
lshf = LSHForest(n_estimators=100) # Adjust
```

Grant Thornton Speaking Engagement

Vibe coding with Generative AI

Introduction

- AI, LLMs, Transformers, AI algorithms and models
- Vibe coding with Python in VS

Vibe coding with Python in Practice

- Automatizing flowchart's design/coding with AI visual recognition
- Vibe coding with ChatGPT, DeepSeek and CoPilot
- Vibe coding with local LLMs for sensitive information

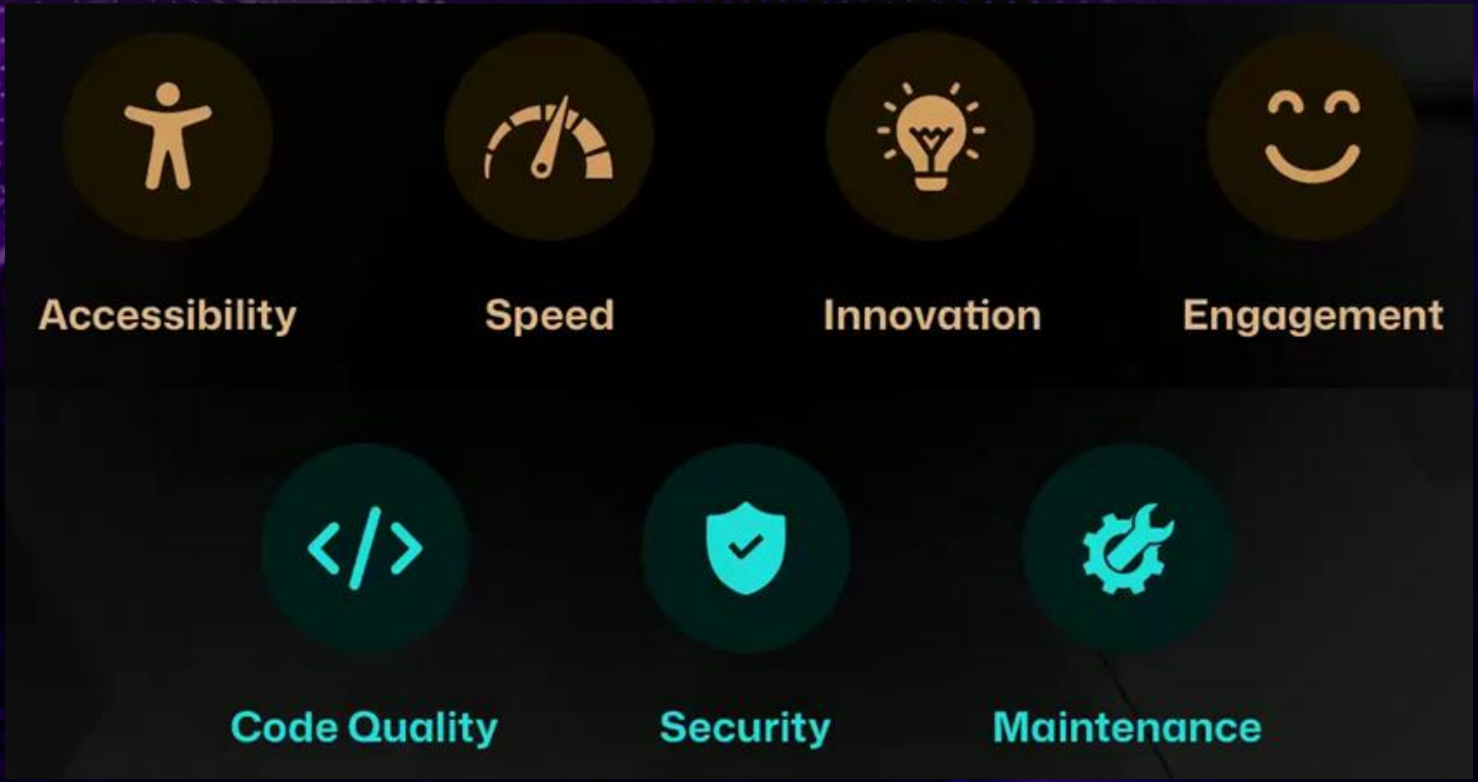
Discussion

Discussion

Vibe coding:
Pros and cons

✓ Fast development and prototyping
✓ Lower entry barrier for beginners
✓ Focus on ideas over syntax

⚠ Requires testing and review (debugging)
⚠ Potential over-dependence on AI
⚠ Security and code quality concerns



Discussion

- Running local LLMs of Gen AI can be used in cases of privacy or security concerns
- Machine learning algorithms/models can now easily be implemented to perform more “advanced” black-box data analysis, but for fine tuning it is necessary to properly know the algorithms/models
- …

Grant Thornton Speaking Engagement

Vibe coding with Generative AI

June 19

Dr. R.M. (Rolando) Gonzales Martinez

Vibe coding with Gen AI in Python

13:30 - 15:00

Introduction

- AI, LLMs, Transformers, AI algorithms and models
- Vibe coding with Python in VS

Vibe coding with Python in Practice

- Automatizing flowchart's design/coding with AI visual recognition
- Vibe coding with ChatGPT, DeepSeek and CoPilot
- Vibe coding with local LLMs for sensitive information

Discussion

https://github.com/rogon666/GT_training