**あなたの発言**

生成AIをどのように活用したか、プロンプトと回答例を含めて報告書を作ってください。 活用の内容は以下を参考にしてください 2024年度Q4 材料計算科学基礎 講義で使ったLLM 特に書いてなければ、Microsoft 365 Copilot (GPT4モデルとのこと) を使っているた 関連Web チュートリアル： 学生と教員のためのpythonとChatGPT活用法 プロンプト 状況: Visual Studio Codeのpython拡張のJupyter notebookが実行できなかった プロンプト:　VSCodeにipykernelをインストールする方法を教えてください 回答: pip install ipykernel 状況: Jupyter notebookファイルをpythonスクリプトに変換するコマンドがわからなかった プロンプト： VSCodeでJupyter notebookファイルをpythonスクリプトに変換する方法を教えてください 回答: jupyter nbconvert --to script test.ipynb （Jupyter notebook編集画面のツールバーの"エクスポート"メニューでも変換できる） 状況: 最小二乗法のプログラムで使うデータファイルを作成する必要があった プロンプト: y=3xに従い、ノイズを含むデータのCSVフォーマットのテキストを出力してください。xの範囲は0~10で１置きで 回答: 01/data2.csv (download/show) 状況: 2023年のチュートリアルで、ChatGPT3.5に線形最小二乗法のプログラムについて質問した プロンプト: pythonで logN = a + b \* Tinv にフィッティングするプログラムを作ってください 回答: 以下が回答のプログラム # numpyライブラリを 別名 np でimport import numpy as np # numpy.polyfit() の呼び出し。 a, b = np.polyfit(Tinv, logN, 1) 　 問題点: 戻り値の順番を間違えている。チュートリアルでは、Bingで検索した記事を参照して修正する例を紹介 修正プログラム: b, a = np.polyfit(Tinv, logN, 1) 状況: プログラムの自習に便利なプロンプトを紹介 プロンプト: 以下のプログラムが何をしているか、わかりやすく説明してください 01/lsq.py (download/show) # %% infile = "data2.csv" fp = open(infile) xs = [] ys = [] xlabel, ylabel = fp.readline().split(',') print("labels:", xlabel, ylabel) # %% for line in fp: values = line.split(',') x = float(values[0]) y = float(values[1]) xs.append(x) ys.append(y) print("x,y=", x, y) fp.close() import numpy as np a, b = np.polyfit(xs, ys, 1) print("a, b=", a, b) #a,bから回帰データを補間。xの間隔は0~10まで、0.1置き xcal = np.arange(0, 10, 0.1) ycal = a \* xcal + b print("xcal=", xcal) print("ycal=", ycal) # %% import matplotlib.pyplot as plt plt.plot(xs, ys, 'o') plt.plot(xcal, ycal, '-') plt.xlabel(xlabel) plt.ylabel(ylabel) plt.show() copy 回答: 以下、抜粋 このプログラムは、CSVファイルからデータを読み取り、線形回帰を実行し、その結果をプロットするものです。 以下に詳細を説明します： １．CSVファイルの読み込み: 　　・ data2.csvというファイルを開きます。 　　・ 空のリスト xs と ys を初期化して、CSVファイルから読み取る x 値と y 値を保存します。 　　・ファイルの最初の行を読み取り、x 軸と y 軸のラベルを取得して xlabel と ylabel に格納します。 ２． データの処理 　　・ 残りの行をループで読み取ります。 　　・ 各行をカンマで分割し、x 値と y 値を取得します。 　　・ これらの値を浮動小数点数に変換して、それぞれ xs と ys のリストに追加します。 　　・ 各行の x 値と y 値を表示します。 以下、続く 状況: 講義では、一般的なプログラム言語の書き方でpythonプログラムを作ってきたが、pythonにはpython固有の書き方もあり、場合によっては高速化、可読性の向上ができる プロンプト: 次のプログラムを、リスト内包表記、sum,map関数などを使って、pythonらしいプログラムに書き直してください 02/lsq\_general.py (download/show) 回答: 02/lsq\_general\_copilot.py (download/show) 状況: 非線形最小二乗法のプログラムを作成した プロンプト: pythonのcurve\_fit()を使って、非線形最小二乗法を行うプログラムを作成してください 回答:以下は、回答プログラムを修正したもの 02/nlsq\_curvefit.py (download/show) import numpy as np from scipy.optimize import curve\_fit import matplotlib.pyplot as plt # パラメータの初期値 p0=[1.0, 0.0, 1.0] # フィッティングする関数の定義 def func(x, a, x0, w): return a \* np.exp(-(x - x0)\*\*2 / w\*\*2) # データ生成 (例) x\_data = np.linspace(-10, 10, 100) y\_data = func(x\_data, 1.0, 0.0, 2.0) + 0.1 \* np.random.normal(size=x\_data.size) # フィッティング popt, pcov = curve\_fit(func, x\_data, y\_data, p0=p0) # フィッティング結果の表示 print(f"Fitted parameters: a={popt[0]}, x0={popt[1]}, w={popt[2]}") # フィッティングした係数を使って計算した x と y のデータ x\_cal = np.linspace(-10, 10, 100) y\_cal = func(x\_cal, \*popt) # 初期値 p0 を使って計算した y\_ini y\_ini = func(x\_cal, \*p0) # グラフのプロット plt.plot(x\_data, y\_data, 'b-', label='data') plt.plot(x\_cal, y\_ini, 'g--', label='initial fit') plt.plot(x\_cal, y\_cal, 'r-', label='fit') plt.xlabel('x') plt.ylabel('y') plt.legend() plt.show() copy

**Copilot の発言**

理解しました。それでは、提供いただいた内容をもとに、生成AIの活用についての報告書を作成いたします。

**生成AI活用報告書**

**1. はじめに**

この報告書では、2024年度Q4の「材料計算科学基礎」の講義において、生成AIをどのように活用したかについて説明します。特に、Microsoft 365 Copilot (GPT-4モデル) を使用した具体的なプロンプトと回答例を示します。

**2. 活用例**

**2.1 Visual Studio Codeの拡張機能設定**

**状況**

Visual Studio CodeのPython拡張のJupyter Notebookが実行できなかった。

**プロンプト:** 「VSCodeにipykernelをインストールする方法を教えてください。」

**回答:**

コピー

pip install ipykernel

**2.2 Jupyter Notebookファイルの変換**

**状況**

Jupyter NotebookファイルをPythonスクリプトに変換するコマンドがわからなかった。

**プロンプト:** 「VSCodeでJupyter NotebookファイルをPythonスクリプトに変換する方法を教えてください。」

**回答:**

コピー

jupyter nbconvert --to script test.ipynb

（Jupyter Notebook編集画面のツールバーの"エクスポート"メニューでも変換できる）

**2.3 データファイルの作成**

**状況**

最小二乗法のプログラムで使用するデータファイルを作成する必要があった。

**プロンプト:** 「y=3xに従い、ノイズを含むデータのCSVフォーマットのテキストを出力してください。xの範囲は0~10で１置きで。」

**回答:** 以下のようなデータを含むCSVファイル data2.csv を生成しました。

**2.4 線形最小二乗法のプログラムについての質問**

**状況**

ChatGPT-3.5に線形最小二乗法のプログラムについて質問した。

**プロンプト:** 「Pythonで logN = a + b \* Tinv にフィッティングするプログラムを作ってください。」

**回答:**

python

コピー

import numpy as np

a, b = np.polyfit(Tinv, logN, 1)

しかし、戻り値の順番が間違っていたため、以下のように修正しました。

python

コピー

b, a = np.polyfit(Tinv, logN, 1)

**2.5 プログラムの説明**

**状況**

プログラムの自習に便利なプロンプトを紹介。

**プロンプト:** 「以下のプログラムが何をしているか、わかりやすく説明してください」

python

コピー

# CSVファイルからデータを読み取り、線形回帰を実行し、その結果をプロットするプログラム

import numpy as np

import matplotlib.pyplot as plt

infile = "data2.csv"

fp = open(infile)

xs = []

ys = []

xlabel, ylabel = fp.readline().split(',')

print("labels:", xlabel, ylabel)

for line in fp:

values = line.split(',')

x = float(values[0])

y = float(values[1])

xs.append(x)

ys.append(y)

print("x,y=", x, y)

fp.close()

a, b = np.polyfit(xs, ys, 1)

print("a, b=", a, b)

xcal = np.arange(0, 10, 0.1)

ycal = a \* xcal + b

print("xcal=", xcal)

print("ycal=", ycal)

plt.plot(xs, ys, 'o')

plt.plot(xcal, ycal, '-')

plt.xlabel(xlabel)

plt.ylabel(ylabel)

plt.show()

**回答:** このプログラムは、CSVファイルからデータを読み取り、線形回帰を実行し、その結果をプロットするものです。

**2.6 プログラムの高速化と可読性の向上**

**状況**

一般的なプログラム言語の書き方でPythonプログラムを作成したが、Python固有の書き方もあり、高速化や可読性の向上ができる。

**プロンプト:** 「次のプログラムを、リスト内包表記、sum, map関数などを使って、Pythonらしいプログラムに書き直してください。」

**回答:** lsq\_general.pyを以下のように書き直しました。

**2.7 非線形最小二乗法のプログラム**

**状況**

非線形最小二乗法のプログラムを作成した。

**プロンプト:** 「Pythonのcurve\_fit()を使って、非線形最小二乗法を行うプログラムを作成してください。」

**回答:**

python

コピー

import numpy as np

from scipy.optimize import curve\_fit

import matplotlib.pyplot as plt

# パラメータの初期値

p0 = [1.0, 0.0, 1.0]

# フィッティングする関数の定義

def func(x, a, x0, w):

return a \* np.exp(-(x - x0)\*\*2 / w\*\*2)

# データ生成 (例)

x\_data = np.linspace(-10, 10, 100)

y\_data = func(x\_data, 1.0, 0.0, 2.0) + 0.1 \* np.random.normal(size=x\_data.size)

# フィッティング

popt, pcov = curve\_fit(func, x\_data, y\_data, p0=p0)

# フィッティング結果の表示

print(f"Fitted parameters: a={popt[0]}, x0={popt[1]}, w={popt[2]}")

# フィッティングした係数を使って計算した x と y のデータ

x\_cal = np.linspace(-10, 10, 100)

y\_cal = func(x\_cal, \*popt)

# 初期値 p0 を使って計算した y\_ini

y\_ini = func(x\_cal, \*p0)

# グラフのプロット

plt.plot(x\_data, y\_data, 'b-', label='data')

plt.plot(x\_cal, y\_ini, 'g--', label='initial fit')

plt.plot(x\_cal, y\_cal, 'r-', label='fit')

plt.xlabel('x')

plt.ylabel('y')

plt.legend()

plt.show()

**3. 結論**

以上のように、生成AIを活用することで、Visual Studio Codeの設定からデータ解析、プログラムの効率化まで幅広いサポートが得られました。今後もこの技術を活用し、さらなる教育の充実を目指します。

このような形式で、生成AIの活用報告書を作成いたしました。必要に応じて内容を追加・修正することも可能ですので、お知らせください。