**#Python script for the** **GPT model evaluation**

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

from sklearn.metrics import accuracy\_score, precision\_score, f1\_score,confusion\_matrix, roc\_auc\_score, recall\_score

from sklearn.metrics import roc\_curve, auc

from sklearn.preprocessing import LabelEncoder

df = pd.read\_excel('/content/chatgpt tm results.xlsx')

# Assuming you have a label\_mapping dictionary defined earlier

label\_mapping = {'positive': 2, 'negative': 1, 'neutral': 0}

# Convert to lowercase, handling NaN values

senti\_pred = [str(label).lower() for label in df['predicted sentiment']]

senti\_target = [label.lower() for label in df['target\_labels2']]

# Mapping target labels

target\_mapped = [label\_mapping[label] for label in senti\_target]

# Mapping predicted labels with handling for empty string

senti\_mapped = [label\_mapping[label] if label in label\_mapping else label\_mapping['neutral'] for label in senti\_pred]

# Display the first few rows of the mapped data

mapped\_df = pd.DataFrame({'senti\_target': target\_mapped, 'senti\_pred': senti\_mapped})

print(mapped\_df.head())

# Calculate accuracy

accuracy = accuracy\_score(target\_mapped, senti\_mapped)

precision = precision\_score(target\_mapped, senti\_mapped, average='macro')

recall = recall\_score(target\_mapped, senti\_mapped, average='macro')

f1 = f1\_score(target\_mapped, senti\_mapped, average='macro')

conf\_matrix = confusion\_matrix(target\_mapped, senti\_mapped)

# Display the results

print(f'Accuracy: {accuracy:.6f}')

print(f'Precision: {precision:.6f}')

print(f'F1 Score: {f1:.6f}')

print('Confusion Matrix:')

print(conf\_matrix)