**SOM CODE**

import os

import pandas as pd

from sklearn.preprocessing import MinMaxScaler

from minisom import MiniSom

import matplotlib.pyplot as plt

# Define the file path (with corrected file name)

file\_path = r"C:\Users\HP\Desktop\emc java\Gold  Prices.csv"

# Check if the file exists

if not os.path.exists(file\_path):

    print(f"Error: File not found at {file\_path}.")

    # List all files in the directory to help locate the correct file

    directory\_path = os.path.dirname(file\_path)

    print("Files in directory:", os.listdir(directory\_path))

else:

    try:

        # Load the CSV data

        gold\_data = pd.read\_csv(file\_path)

        print("Data loaded successfully.")

        # Preview the data

        print("Data preview:")

        print(gold\_data.head())

        # Preprocess the data

        data = gold\_data.iloc[:, 1:].values  # Skipping the first column if it's an index or date column

        scaler = MinMaxScaler()

        data\_normalized = scaler.fit\_transform(data)

        # Train the SOM

        som\_shape = (20, 20)  # Define the shape of the SOM grid

        som = MiniSom(som\_shape[0], som\_shape[1], data\_normalized.shape[1], sigma=1.0, learning\_rate=0.5)

        som.random\_weights\_init(data\_normalized)

        som.train\_random(data\_normalized, 100)  # Train the SOM for 100 iterations

        # Visualize the SOM

        plt.figure(figsize=(10, 10))

        plt.title("Self-Organizing Map of Gold Prices Data")

        # Plot each data point on the SOM grid

        for i, x in enumerate(data\_normalized):

            winner = som.winner(x)

            plt.plot(winner[0] + 0.5, winner[1] + 0.5, 'o', markerfacecolor='red', markeredgecolor='k', markersize=8, alpha=0.7)

        # Display the grid with color-coded nodes

        plt.xlim([0, som\_shape[0]])

        plt.ylim([0, som\_shape[1]])

        plt.grid()

        plt.show()

    except Exception as e:

        print("An error occurred while processing the file:", e)