import yfinance as yf

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

from matplotlib.backends.backend\_tkagg import FigureCanvasTkAgg

import tkinter as tk

from scipy.stats import norm

# Function to fetch data

def fetch\_data(tickers):

data = yf.download(tickers, start="2021-01-01", end="2023-01-01")

return data['Adj Close']

# Forecast returns using a basic statistical model

def forecast\_returns(returns, days=730):

mean\_returns = returns.mean()

std\_dev\_returns = returns.std()

forecasts = np.random.normal(mean\_returns, std\_dev\_returns, (days, len(returns.columns)))

return pd.DataFrame(forecasts, columns=returns.columns)

# Update the graph and options details

def update(tickers\_entry, root):

tickers = tickers\_entry.get().split(',')

data = fetch\_data(tickers)

returns = data.pct\_change().dropna()

# Forecast future returns and calculate portfolio value

forecasted\_returns = forecast\_returns(returns)

forecasted\_portfolio = (forecasted\_returns + 1).cumprod()

forecasted\_portfolio['Portfolio Value'] = forecasted\_portfolio.dot(np.ones(len(tickers)) / len(tickers))

# Graph

fig, ax = plt.subplots(figsize=(10, 6))

ax.plot(forecasted\_portfolio.index, forecasted\_portfolio['Portfolio Value'], label='Forecasted Portfolio Value')

ax.set\_title('Forecasted Portfolio Performance')

ax.set\_xlabel('Days')

ax.set\_ylabel('Portfolio Value')

ax.legend()

# Clear previous figure

for widget in root.pack\_slaves():

widget.destroy()

# Embedding matplotlib figure in Tkinter

canvas = FigureCanvasTkAgg(fig, master=root)

canvas.draw()

canvas.get\_tk\_widget().pack(side=tk.TOP, fill=tk.BOTH, expand=True)

# Display options needed for hedging

options\_frame = tk.Frame(root)

options\_frame.pack(side=tk.BOTTOM)

tk.Label(options\_frame, text="Options Needed for Hedging:", font=('Arial', 14)).pack(side=tk.TOP)

for ticker in tickers:

# Assuming 1 put option per 100 shares for simplicity

tk.Label(options\_frame, text=f"Buy put options for {ticker}: 100").pack()

# GUI setup

def create\_gui():

root = tk.Tk()

root.title('Portfolio Forecast and Hedging Details')

input\_frame = tk.Frame(root)

input\_frame.pack(side=tk.TOP)

tk.Label(input\_frame, text="Enter stock tickers separated by comma:", font=('Arial', 12)).pack(side=tk.LEFT)

tickers\_entry = tk.Entry(input\_frame)

tickers\_entry.pack(side=tk.LEFT)

submit\_button = tk.Button(input\_frame, text="Submit", command=lambda: update(tickers\_entry, root))

submit\_button.pack(side=tk.LEFT)

root.mainloop()

# Main function to run the application

def main():

create\_gui()

if \_\_name\_\_ == '\_\_main\_\_':

main()