import yfinance as yf

import pandas as pd

# Define the tickers for the portfolio and the ETFs

portfolio tickers = ['MSFT', 'AAPL', 'GOOGL', 'AMZN', 'TSLA', 'NVDA', 'META']

etf\_tickers = ['SPY', 'IWM', 'DIA']

# Download the historical data for the last 10 years (for total returns, etc.)

data = Yf. download (portfolio tickers + etf\_tickers, period='10y', adjusted=True)

# Calculate daily returns

Daily reurns= data ['Adj Close'] .pct\_change()

# Portfolio weight (equal weights for all assets)

portfolio weight = 1 / Len (portfolio tickers)

# Calculate Annualized Volatility (Trailing 3 months) # **\***

**\*** Volatility\_3m = daily returns [-63:]. std () \* (252 \*\* 0.5) window for 3 months

# Calculate Beta (Trailing 12 months) against SPY, IWM, DIA

beta\_spy = daily\_returns['MSFT'].cov(daily\_returns['SPY']) / daily\_returns['SPY'].var() beta\_iwm = daily\_returns['MSFT'].cov(daily\_returns['IWM']) / daily\_returns['IWM'].var()

beta\_dia = daily\_returns['MSFT'].cov(daily\_returns['DIA']) / daily\_returns['DIA'].var()

# Calculate the Weekly Drawdown (52-week high and low)

high\_52w = data['Adj Close'].rolling(window=252).max()

low\_52w = data['Adj Close'].rolling(window=252).min()

avg\_weekly\_drawdown = ((high\_52w - low\_52w) / 52).mean() max\_

weekly\_drawdown = (high\_52w - low\_52w) / 52

# Total Return and Annualized Total Return (Trailing 10 years)

total\_return\_10y = (data['Adj Close'][-1] / data['Adj Close'][0]) - 1

annualized\_return\_10y = (data['Adj Close'][-1] / data['Adj Close'][0]) \*\* (1/10) - 1

# Prepare the portfolio metrics table

portfolio\_metrics = pd.DataFrame

({ 'Ticker': portfolio\_tickers,

'Portfolio Weight': portfolio\_weight,

'Annualized Volatility (3M)': volatility\_3m,

'Beta against SPY': beta\_spy,

'Beta against IWM': beta\_iwm,

'Beta against DIA': beta\_dia,

'Avg Weekly Drawdown': avg\_weekly\_drawdown,

'Max Weekly Drawdown': max\_weekly\_drawdown,

'Total Return (10Y)': total\_return\_10y,

'Annualized Total Return (10Y)': annualized\_return\_10y })

print("Portfolio Metrics Table:") print(portfolio\_metrics)

# Calculate Portfolio Returns (Average of the 7 assets)

portfolio\_returns = daily\_returns[portfolio\_tickers].mean(axis=1)

# Compare Portfolio with ETFs (SPY, IWM, DIA)

correlation\_spy = portfolio\_returns.corr(daily\_returns['SPY'])

correlation\_iwm = portfolio\_returns.corr(daily\_returns['IWM'])

correlation\_dia = portfolio\_returns.corr(daily\_returns['DIA'])

cov\_spy\_portfolio = portfolio\_returns.cov(daily\_returns['SPY'])

cov\_iwm\_portfolio = portfolio\_returns.cov(daily\_returns['IWM'])

cov\_dia\_portfolio = portfolio\_returns.cov(daily\_returns['DIA'])

# Calculate Tracking Error (Standard Deviation of Difference in Returns)

tracking\_error\_spy = (portfolio\_returns - daily\_returns['SPY']).std() \* (252 \*\* 0.5)

tracking\_error\_iwm = (portfolio\_returns - daily\_returns['IWM']).std() \* (252 \*\* 0.5)

tracking\_error\_dia = (portfolio\_returns - daily\_returns['DIA']).std() \* (252 \*\* 0.5)

# Sharpe Ratio (Assuming a 2% Risk-Free Rate)

rf\_rate = 0.02

sharpe\_ratio spy = (portfolio\_returns.mean() - rf\_rate) / portfolio\_returns.std() sharpe\_ratio\_iwm = (portfolio\_returns.mean() - rf\_rate) / portfolio\_returns.std() sharpe\_ratio\_dia = (portfolio\_returns.mean() - rf\_rate) / portfolio\_returns.std()

# Volatility of the Portfolio and ETFs

portfolio\_volatility = portfolio\_returns.std() \* (252 \*\* 0.5)

spy\_volatility = daily\_returns['SPY'].std() \* (252 \*\* 0.5)

iwm\_volatility = daily\_returns['IWM'].std() \* (252 \*\* 0.5)

dia\_volatility = daily\_returns['DIA'].std() \* (252 \*\* 0.5)

# Compare the portfolio to the ETFs in a table

comparison\_table = pd.DataFrame({

'ETF Ticker': etf\_tickers,

'Correlation against ETF': [correlation\_spy, correlation\_iwm, correlation\_dia],

'Covariance with Portfolio': [cov\_spy\_portfolio, cov\_iwm\_portfolio, cov\_dia\_portfolio],

'Tracking Error (10Y)': [tracking\_error\_spy, tracking\_error\_iwm, tracking\_error\_dia],

'Sharpe Ratio': [sharpe\_ratio\_spy, sharpe\_ratio\_iwm, sharpe\_ratio\_dia],

'Volatility Spread': [portfolio\_volatility - spy\_volatility, portfolio\_volatility –

iwm\_volatility, portfolio\_volatility - dia\_volatility] })

print("\nPortfolio vs ETF Comparison Table:")

print(comparison\_table) # Create Correlation Matrix between the Portfolio Assets and the ETFs correlation\_matrix = pd.concat([daily\_returns[portfolio\_tickers], daily\_returns[etf\_tickers]], axis=1).corr()