Lab4

1. Team Details

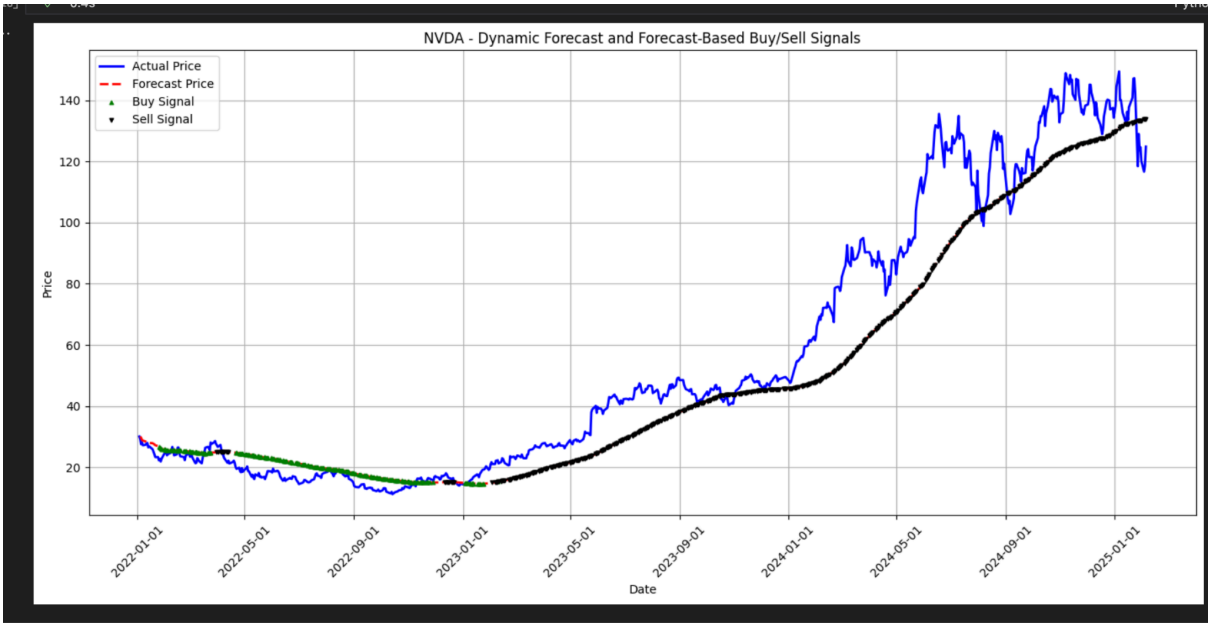
Name	USC ID
Chenxiao Yu	6024079123
Yiqing Hong	4395913002

2. Github link:
http://github.com/AiChiMoCha/SP25_DSCI560/tree/main/lab4/scripts

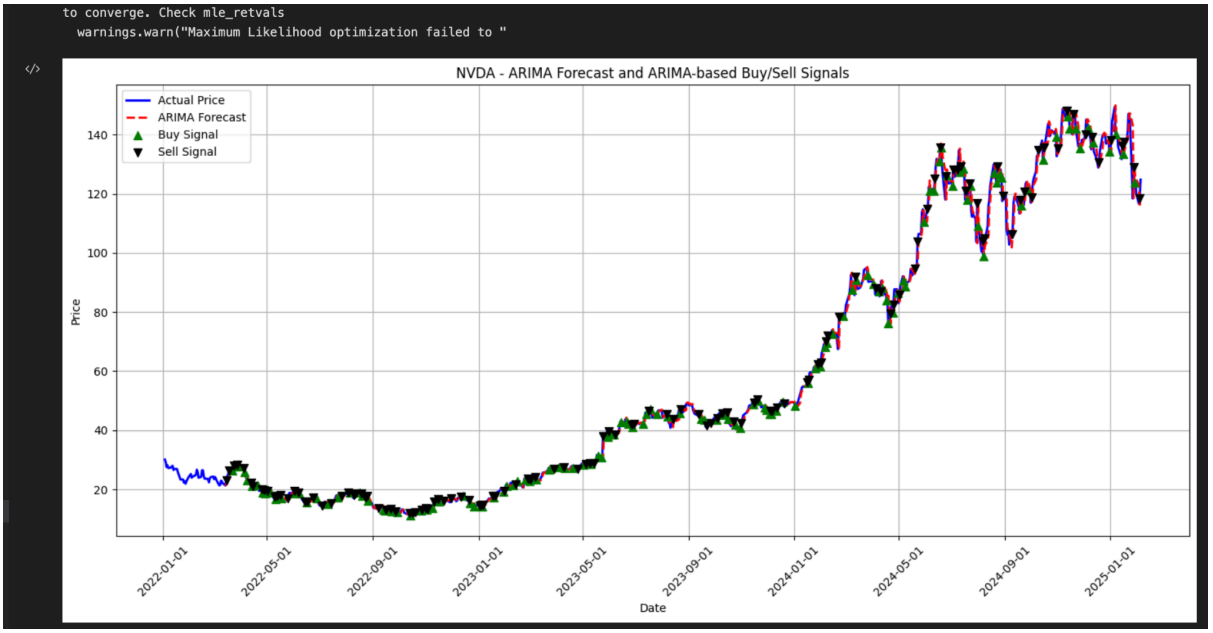
3. Video link:
https://youtu.be/VHzuispfihg

4. Algorithm Development

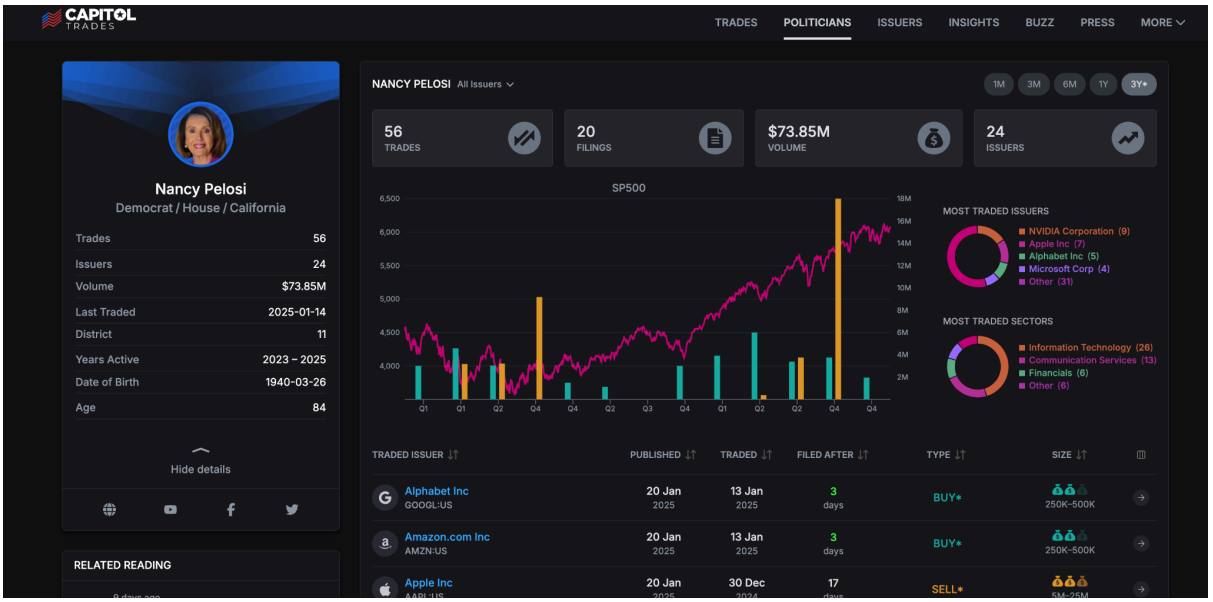
moving average

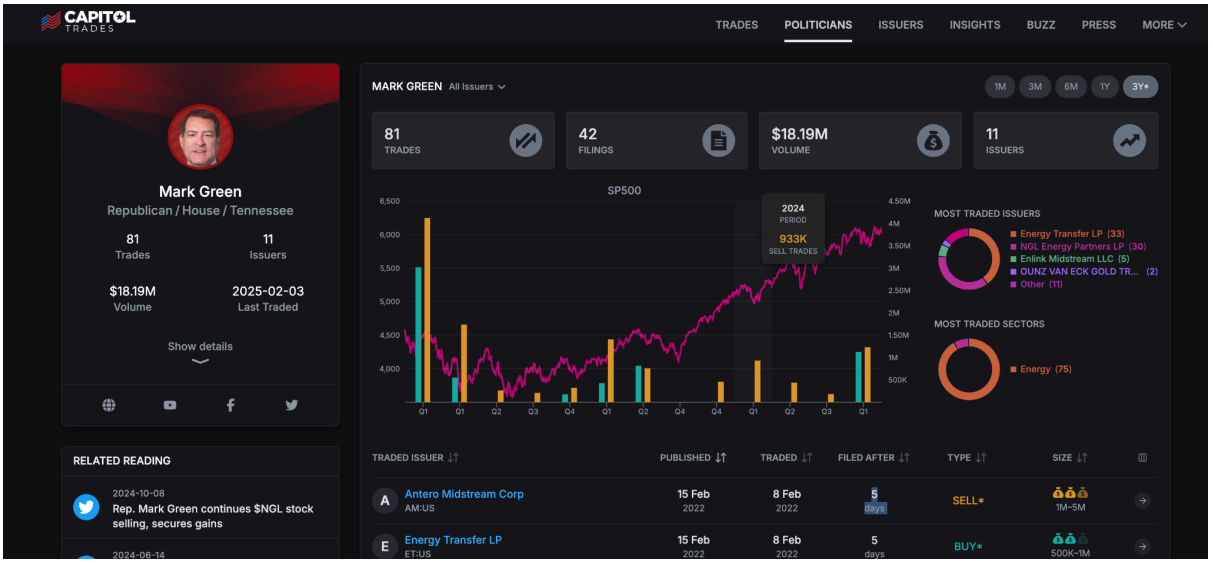


arima

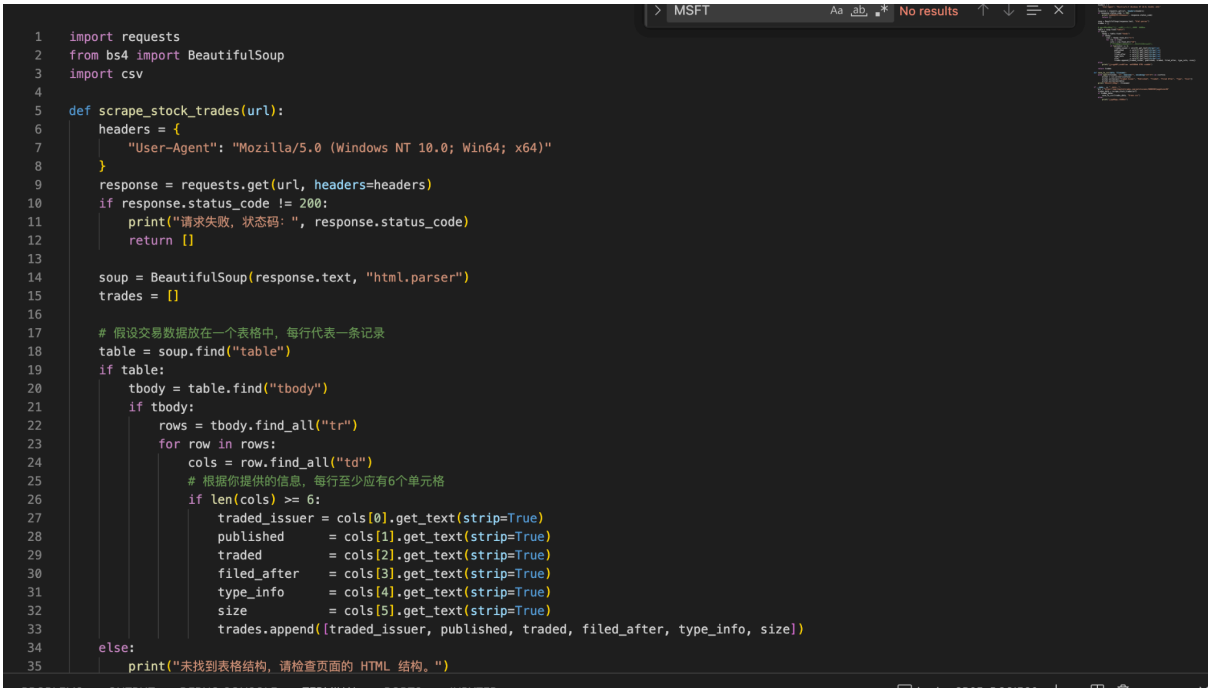


ours- tracking congress member Nancy Pelosi & Mark Greens





our scraper



Our strategy is mainly to buy or sell according to the stock trading records of the congressmen. Since these congressmen make a lot of money on stocks, and people tend to follow these people to buy or sell, their actions reflect or even influence the changes in stock prices to some extent.

For each congressman's transaction record, a signal record is generated based on stock data:

- ref_price: the stock closing price on the trading day (traded_date)
- published_price: the stock closing price on the disclosure day (published_date)
- ref_to_pub_ma: the average closing price from the trading day to the disclosure day (including both ends)
- delay_days: the number of days for disclosure delay
- congress: congressman's name
- signal_valid: whether the signal meets the conditions
- signal_date: signal date (take the trading day corresponding to the disclosure day)
- avg_trade_volume: the average transaction volume after parsing
- signal_strength: the strength set according to the trading volume in the signal

Buy signal:

- Delay <= 10 days: directly valid;
- Delay > 10 days: requires ref_price > published_price or ref_price < ref_to_pub_ma.

Sell signal:

- Delay <= 10 days: directly valid;
- Delay > 10 days: requires ref_price < published_price or ref_price > ref_to_pub_ma.

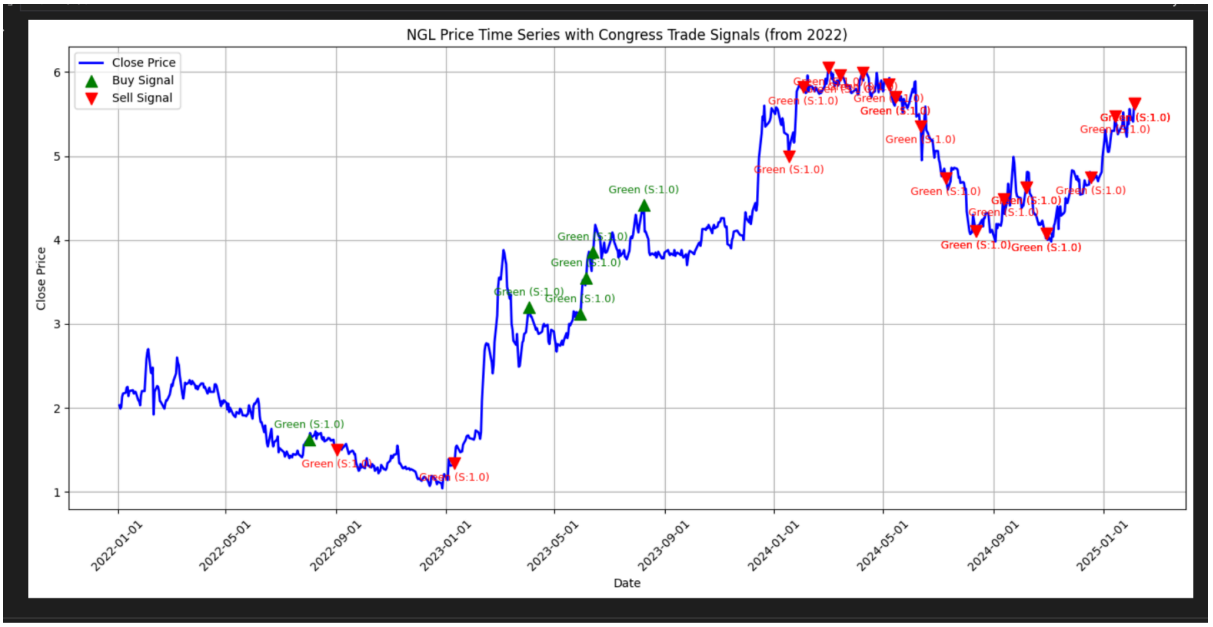
If there is no disclosure date in the record, skip the record.

```python
def generate_signals(trade_df):
    """
    针对每笔议员交易记录，根据股票数据生成信号记录，
    信号记录中包含：
      - ref_price：交易日（traded_date）的股票收盘价
      - published_price：披露日（published_date）的股票收盘价
      - ref_to_pub_ma：从交易日至披露日（包含两端）的收盘价均值
      - delay_days：披露延迟天数
      - congress：议员名称
      - signal_valid：信号是否满足条件
      - signal_date：信号日期（取披露日对应的交易日）
      - avg_trade_volume：解析后的平均交易额
      - signal_strength：根据信号中的交易量设置的强度

    买入信号：
      - 延迟 <= 10 天：直接有效；
      - 延迟 > 10 天：要求 ref_price > published_price 或 ref_price < ref_to_pub_ma。

    卖出信号：
      - 延迟 <= 10 天：直接有效；
      - 延迟 > 10 天：要求 ref_price < published_price 或 ref_price > ref_to_pub_ma。

    如果记录中没有披露日，则跳过该记录。
    """
    signals = []
    stock_data_dict = {}

    for idx, row in trade_df.iterrows():
        # 更新 ticker，确保 ticker 来自 stock_list 中的代码（支持模糊匹配）
        raw_ticker = row['ticker']
```

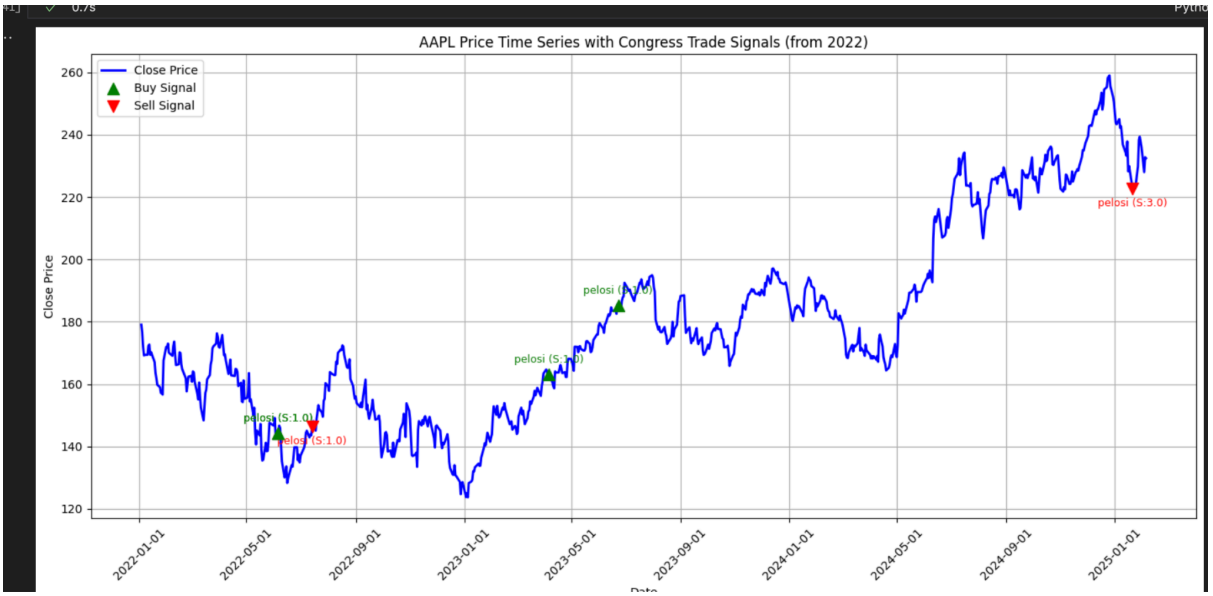
```python
        continue

      traded_date = pd.Timestamp(row['traded_date'])
      published_date = pd.Timestamp(row['published_date'])

      # 获取交易日的收盘价作为 ref_price
      nearest_traded_date = stock_df.index[np.argmin(np.abs(stock_df.index - traded_date))]
      ref_price = stock_df.loc[nearest_traded_date, 'close_price']

      # 获取披露日的收盘价作为 published_price
      nearest_published_date = stock_df.index[np.argmin(np.abs(stock_df.index - published_date))]
      published_price = stock_df.loc[nearest_published_date, 'close_price']

      # 计算交易日至披露日之间的均价（ref_to_pub_ma）
      if traded_date > published_date:
          ref_to_pub_ma = ref_price
      else:
          period_data = stock_df.loc[traded_date:published_date]
          if not period_data.empty:
              ref_to_pub_ma = period_data['close_price'].mean()
          else:
              ref_to_pub_ma = np.nan

      # 判断信号是否有效
      signal_valid = False
      if trade_type == "BUY":
          if delay <= 10:
              signal_valid = True
          else:
              if published_price is not None and (ref_price > published_price or ref_price < ref_to_pub_ma):
                  signal_valid = True
      elif trade_type == "SELL":
          if delay <= 10:
              signal_valid = True
          else:
```

Examples of following the tracking strategy

NGL



APPL



our ongoing plan: Combine with other data to optimize strategy

We will combine the trading data of congressmen with other market data to build a multi-factor stock selection strategy to improve the trading success rate.

(1) Combine with fundamental data

If the stocks bought by congressmen also have good fundamentals (such as high growth and low valuation), the success rate will be higher.

When selecting stocks, we will combine:

- PE (price-to-earnings ratio), PB (price-to-book ratio)
- ROE (return on equity)
- Revenue growth rate

Strategy:

Only buy stocks with excellent fundamentals of congressmen's transactions. For example, among the companies bought by Pelosi, select targets with ROE > 15% and revenue growth > 10%.
(2) Combine with sentiment analysis
The buying and selling behavior of congressmen may be related to market sentiment. We may combine social media (such as X/Twitter) and news sentiment analysis:

When the congressmen buy stocks, is there a positive discussion on social media?
Is there policy news support?

Strategy:
If the positive sentiment of the stock on social media increases after the congressmen buy, increase the position. On the other hand, if news shows that the company faces regulatory risks, hold it with caution.

5. Mock trading environment
Core input: Initial capital, stocks, buy_date, sell_date, action, shares
Output: transaction records and performance metrics

```
[**********************100%**********************]  1 of 1 completed

1 Failed download:
['AAPL']: YFPricesMissingError('$%ticker%: possibly delisted; no price data found  (1d 2025-02-09
 00:54:18.093297 -> 2025-02-10 00:54:18.093297) (Yahoo error = "Data doesn\'t exist for startDate
 = 1739080458, endDate = 1739166858")')
No data found for AAPL on 2025-02-09 00:54:18.093297.
No data for AAPL on 2025-02-09 00:54:18.093297. Skipping.
[**********************100%**********************]  1 of 1 completed

1 Failed download:
['GOOGL']: YFPricesMissingError('$%ticker%: possibly delisted; no price data found  (1d 2025-02-0
9 00:54:19.096305 -> 2025-02-10 00:54:19.096305) (Yahoo error = "Data doesn\'t exist for startDat
e = 1739080459, endDate = 1739166859")')
No data found for GOOGL on 2025-02-09 00:54:19.096305.
No data for GOOGL on 2025-02-09 00:54:19.096305. Skipping.

Remaining cash: $6832.53
Total spent: $3167.47
Portfolio after trades: {'AAPL': {'buy_date': datetime.date(2025, 1, 21), 'buy_price': 222.639999
38964844, 'shares': 8}, 'GOOGL': {'buy_date': datetime.date(2025, 1, 21), 'buy_price': 198.050003
0517578, 'shares': 7}}
Total profit: $0.00
Portfolio return: 0.00%
Total portfolio value (cash + stocks): $6832.53
```

The basic idea is to get the signal from our algorithm which decides at which date add which amount of stocks into the portfolio or sell out which amount of stocks. And we use yfinance api to get the close price of the stock at that date, and finally use different performance metrics to measure the performance of the portfolio.
Performance metric:
Portfolio return

The total return of a portfolio over a given period:

$$\text{Portfolio Return} = \frac{\text{Final Portfolio Value} - \text{Initial Portfolio Value}}{\text{Initial Portfolio Value}}$$

If you have multiple stocks, the portfolio return can be calculated as the weighted sum of individual stock returns:

$$\text{Portfolio Return} = \sum_{i=1}^{n} w_i R_i$$

Where:

- w_i = Weight of stock i in the portfolio
- R_i = Return of stock i

Annualized return

Annualized return normalizes returns over different time periods, allowing comparison between different investments:

$$\text{Annualized Return} = (1 + R_{\text{total}})^{\frac{252}{\text{Days}}} - 1$$

or for multiple years:

$$\text{Annualized Return} = \left(\frac{\text{Final Portfolio Value}}{\text{Initial Portfolio Value}}\right)^{\frac{1}{T}} - 1$$

Where:

- R_{total} = Total return of the portfolio
- T = Number of years

Sharpe ratio

The Sharpe Ratio measures risk-adjusted return:

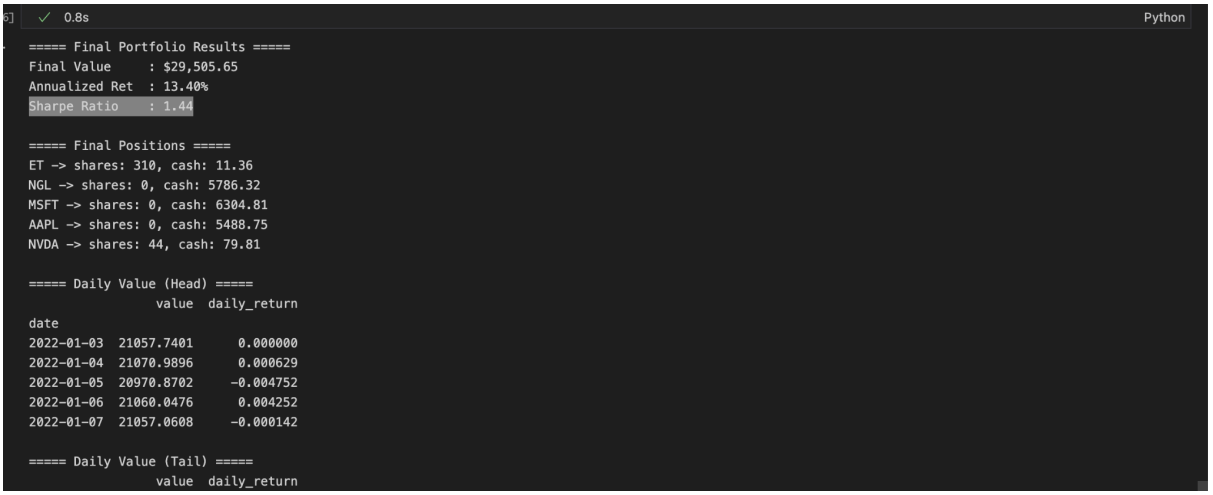
$$\text{Sharpe Ratio} = \frac{R_p - R_f}{\sigma_p}$$

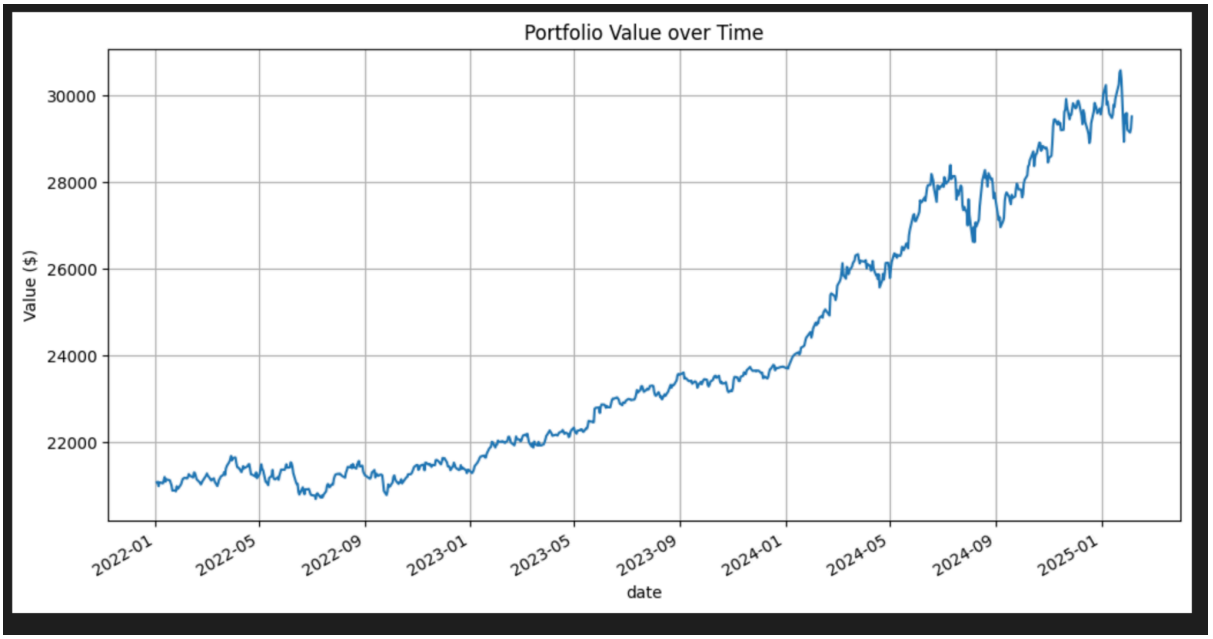
Where:

- R_p = Portfolio return (annualized)
- R_f = Risk-free rate (e.g., U.S. Treasury yield)
- σ_p = Portfolio standard deviation (volatility)

A higher Sharpe Ratio indicates better risk-adjusted performance.

results here




Portfolio Value over Time
Value ($)
30000
28000
26000
24000
22000
2022-01
2022-05
2022-09
2023-01
2023-05
2023-09
2024-01
2024-05
2024-09
2025-01
date