



Stock News Sentiment Analysis

Predicting Stock Movements from News Sentiment



My Little Pony (MLP)

Members



ภักรดنج - ถนนมลีกธิกุล
6633185721



พชร กัลยาณศิริ
6633158821

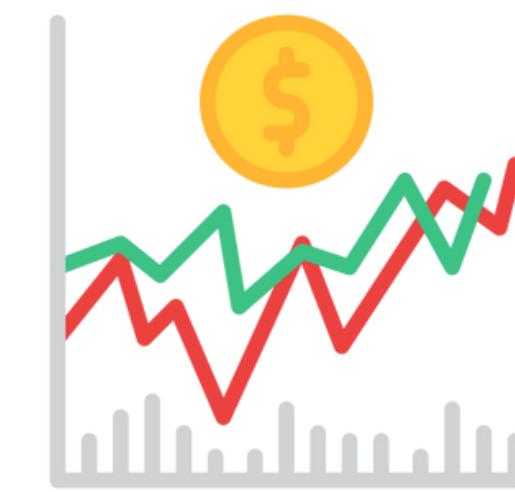
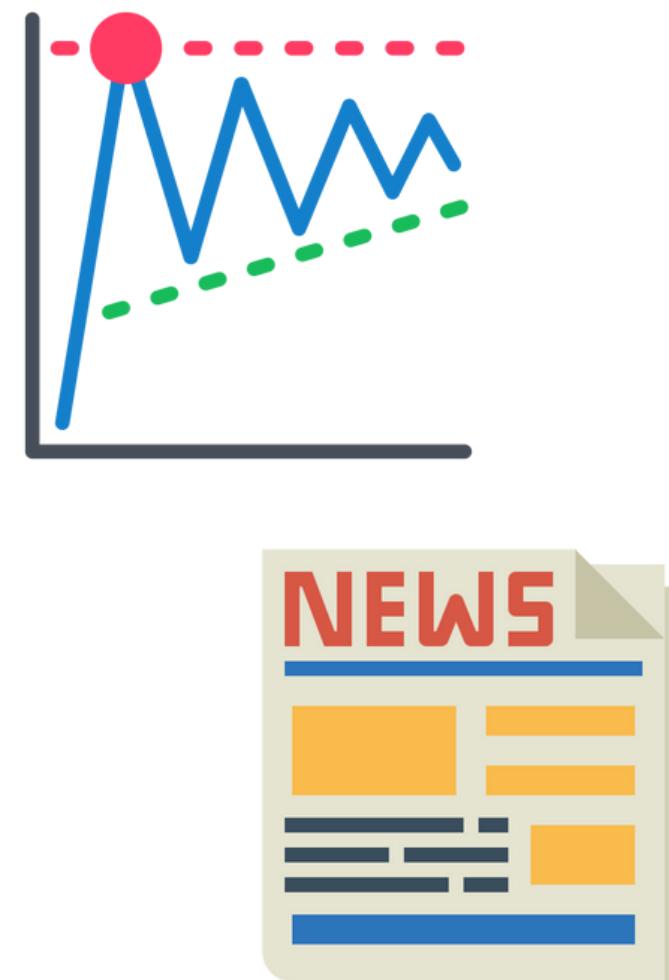


วรรธนา บุญญารัตน์
6633217121

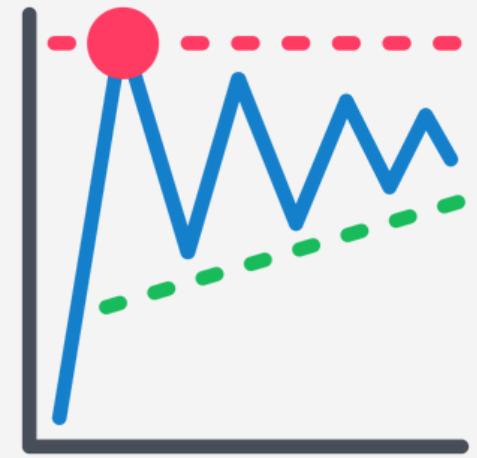


ภาสิศา พิราเวරาร
6633181121

Problem



Existing Methods



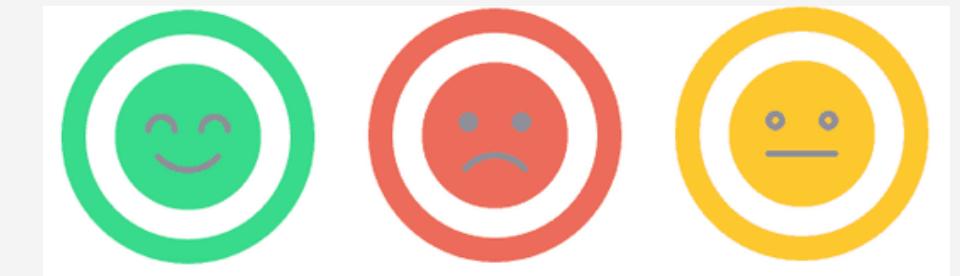
Technical Analysis

MACD, RSI, SMA

Bloomberg

News Sentiment Services

Refinitiv,
Bloomberg

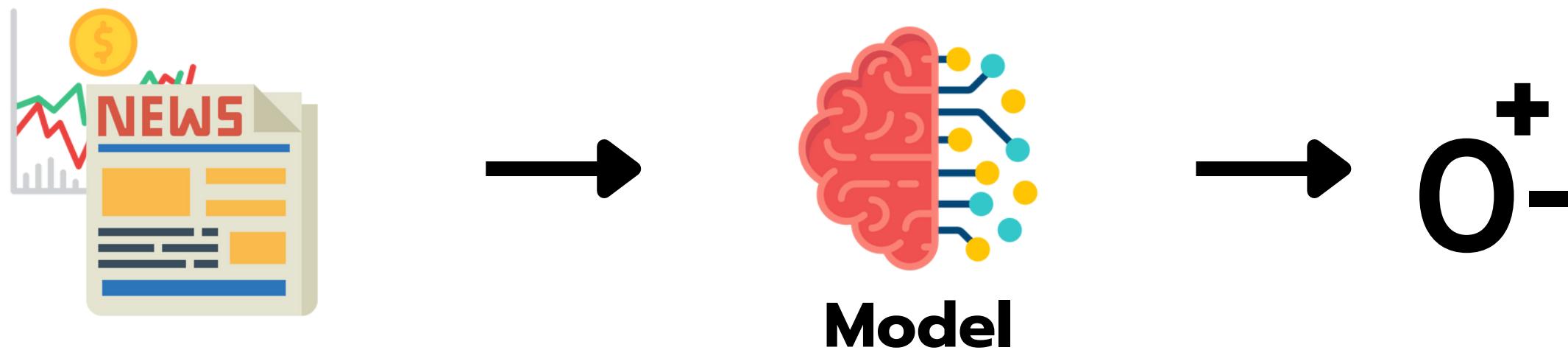


NLP+Time-Series

BERT, FinBERT,
THaLLE

Our Project

The goal is to analyze news articles to classify the sentiment into positive, negative, or neutral, which can potentially reflect market sentiment and help investors make better decisions.



Datasets

Web scraping from investing.com

- e.g., TSLA, MSFT, NVDA, META etc.
- around 60,000 news (rows)

	ticker	publish_date	publish_time	title	body_text	url
0	TSLA	2024-10-18	06:33	Tesla slips as NHTSA probes FSD collisions in ...	Investing.com -- Tesla shares are down around ...	https://www.investing.com/news/stock-market-news...
1	TSLA	2024-04-04	03:52	Musk says he's raising pay for Tesla AI engine...	Tesla (NASDAQ:) co-founder and CEO Elon Musk ...	https://www.investing.com/news/stock-market-news...
2	TSLA	2023-10-03	11:54	Elon Musk must face fraud lawsuit for disclosi...	By Jonathan Stempel\n\nNEW YORK (Reuters) - El...	https://www.investing.com/news/stock-market-news...
3	TSLA	2024-07-03	07:24	Tesla's Q2 deliveries beat estimates,	Investing.com -- Tesla (NASDAQ:) has reported...	https://www.investing.com/news/stock-market-news...

+ Label

Label Method

- Label with FinBERT
- Label with return_next (of the stock)

Pre-trained

- ProsusAI/finbert
- google-bert/bert-base-uncased

Model

1

**Base: FinBERT +
Label from
FinBERT**

2

**Base: google
BERT + Label
from FinBERT**

3

**Base: FinBERT +
Label from
return_next**

4

**Base: google
BERT + Label
from
return_next**

Using Model

```
from transformers import pipeline

# Load fine-tuned model
model_path = "Photchara/stock_sentiment_Finbert_label"
classifier = pipeline(
    "text-classification",
    model=model_path,
    tokenizer=model_path
)

# Example prediction
text = """intel Corp. plans employment reductions and a tighter return-to-work policy in coming months. Media reports in recent days have suggested Intel could lay off more than 20% of its workforce, which would be the largest such cut in the company's history. The cuts are part of a broader effort by the chip giant to reduce costs and improve efficiency. The company has already announced plans to close several factories and reduce its workforce by about 10% over the next few years. The latest news comes as Intel faces increasing competition from other semiconductor manufacturers, particularly in the mobile phone market. The company is also facing pressure from investors to improve its financial performance. The stock price has been volatile in recent weeks, with some analysts suggesting that it may be overvalued. The company's management team is reportedly considering various options to address these challenges, including further cost reductions and potential acquisitions. The future of Intel remains uncertain, but the company appears to be taking steps to ensure its long-term survival in the highly competitive semiconductor industry.""".strip()
result = classifier(text)
print(result)
```

MagicPython

```
Device set to use cuda:0
[{'label': 'negative', 'score': 0.9999463558197021}]
```

[Hugging Face](#)

Performance

%Accuracy	Return_next	Dataset1	Dataset2	Dataset3	Dataset4
Model1	0.335	0.847	0.314	0.706	0.528
Model2	0.451	0.476	0.569	0.281	0.455
Model3	0.335	0.759	0.317	0.652	0.516
Model4	0.454	0.267	0.423	0.206	0.368
FinBERT	0.329	0.894	0.318	0.715	0.509

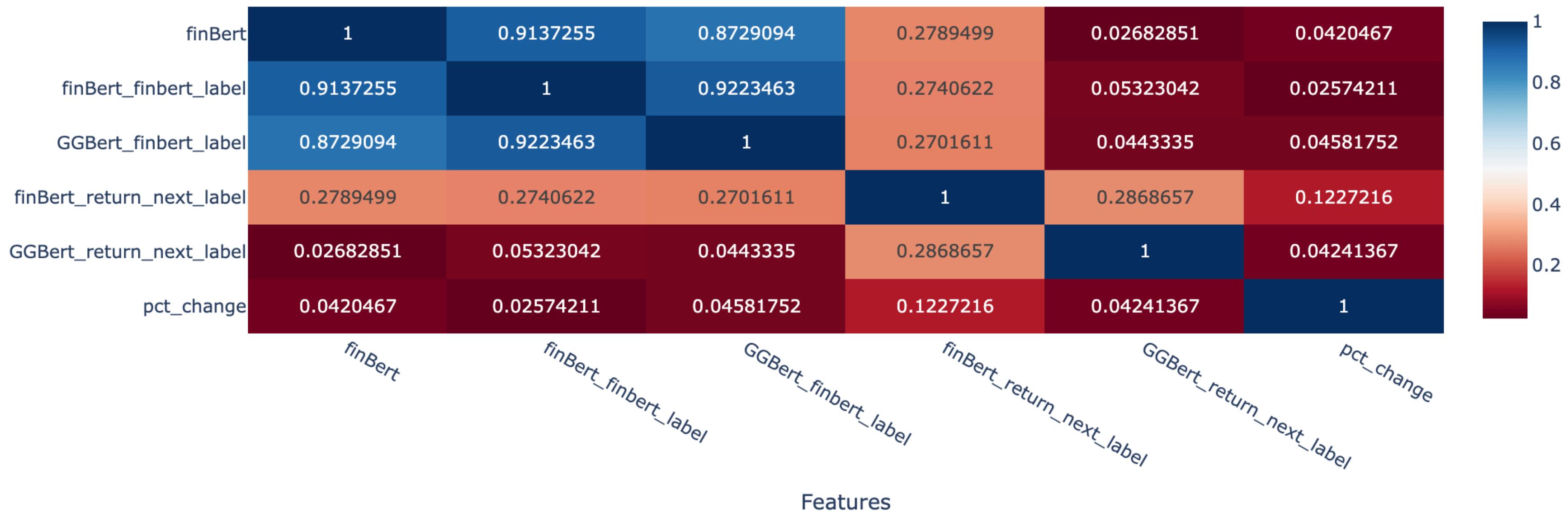
Output

	A date	A finBert	A finBert_finbert_label	A GGBert_finbert_label	A finBert_return_next_label	A
0	2023-11-01	negative	negative	negative	positive	po
1	2023-11-02	positive	positive	positive	positive	ne
2	2023-11-03	negative	negative	negative	positive	po
3	2023-11-06	negative	negative	negative	neutral	ne
4	2023-11-07	positive	positive	positive	positive	po

Evaluation

NVDA

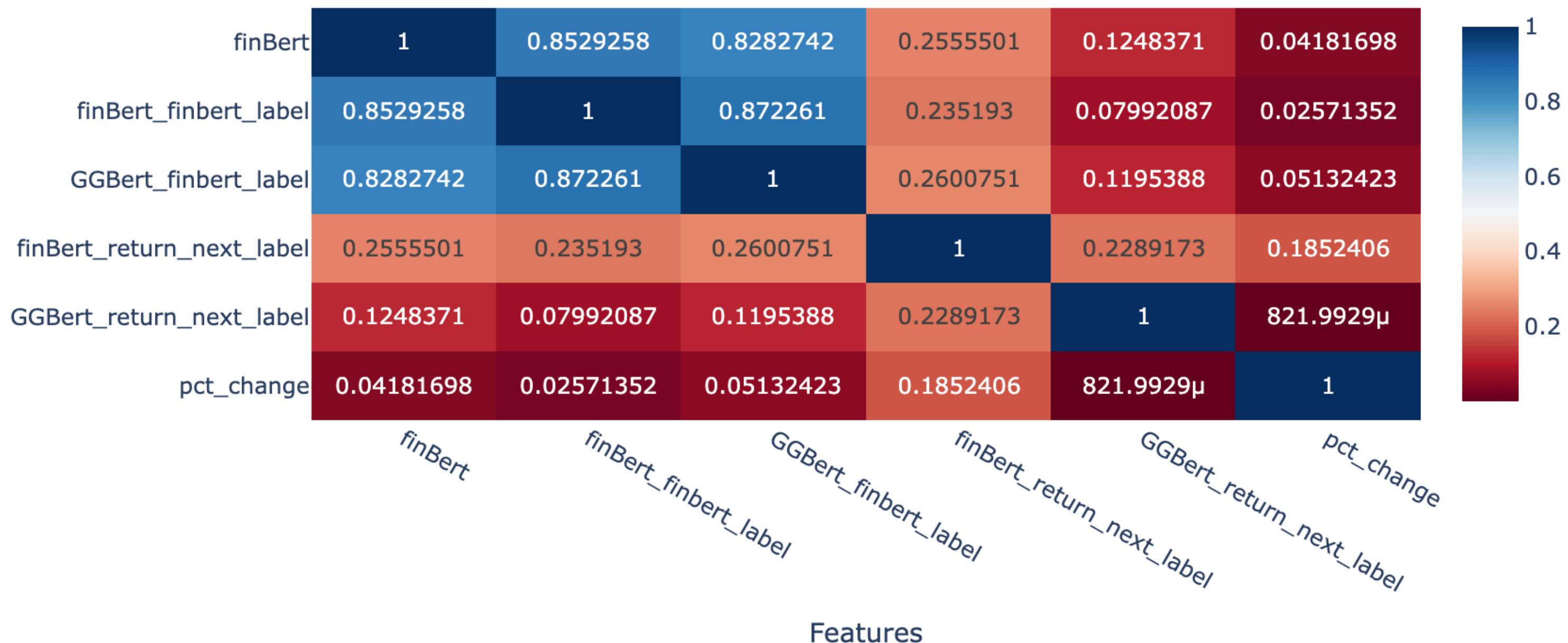
Correlation Matrix



Evaluation

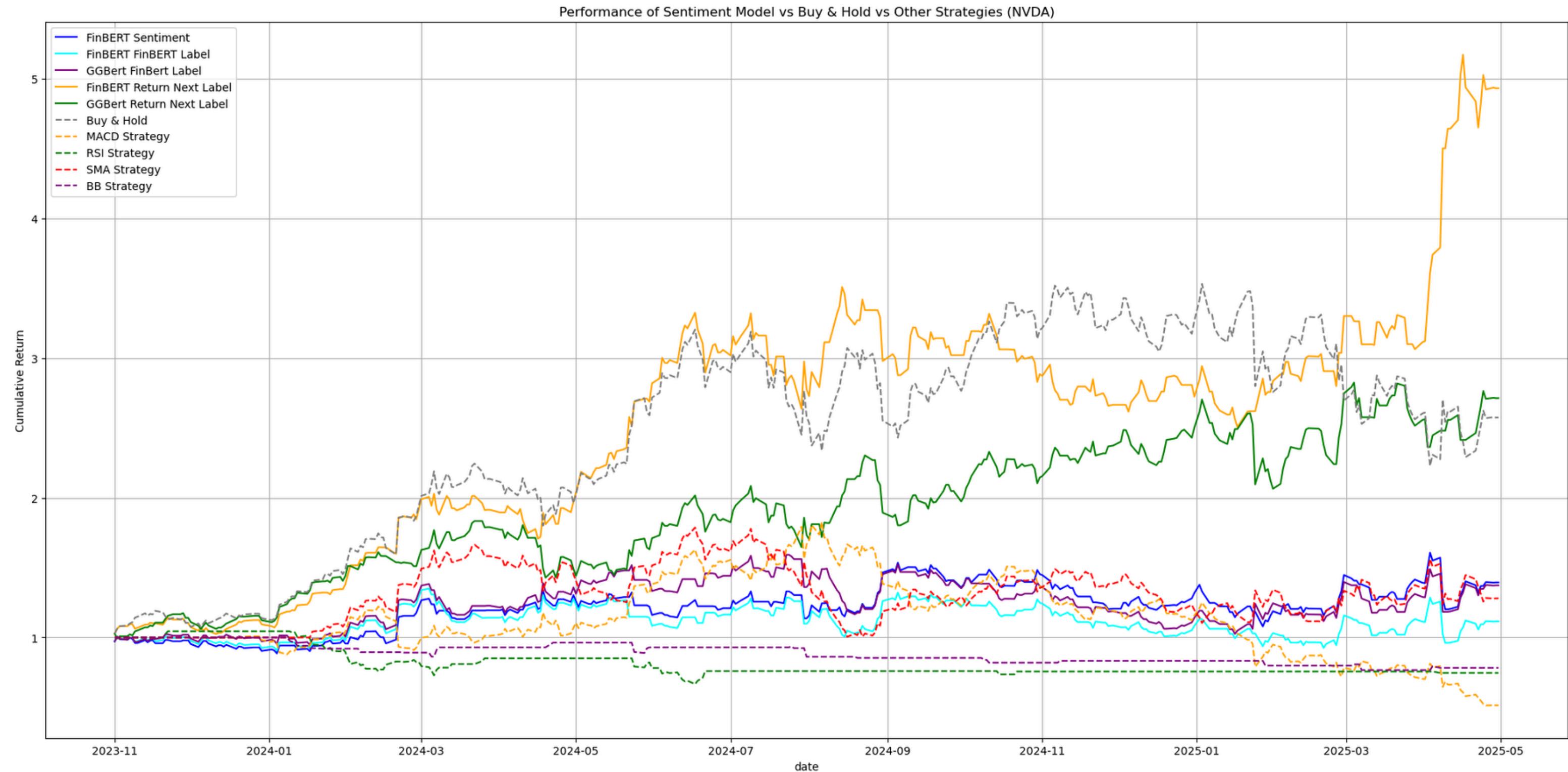
^SPX

Correlation Matrix



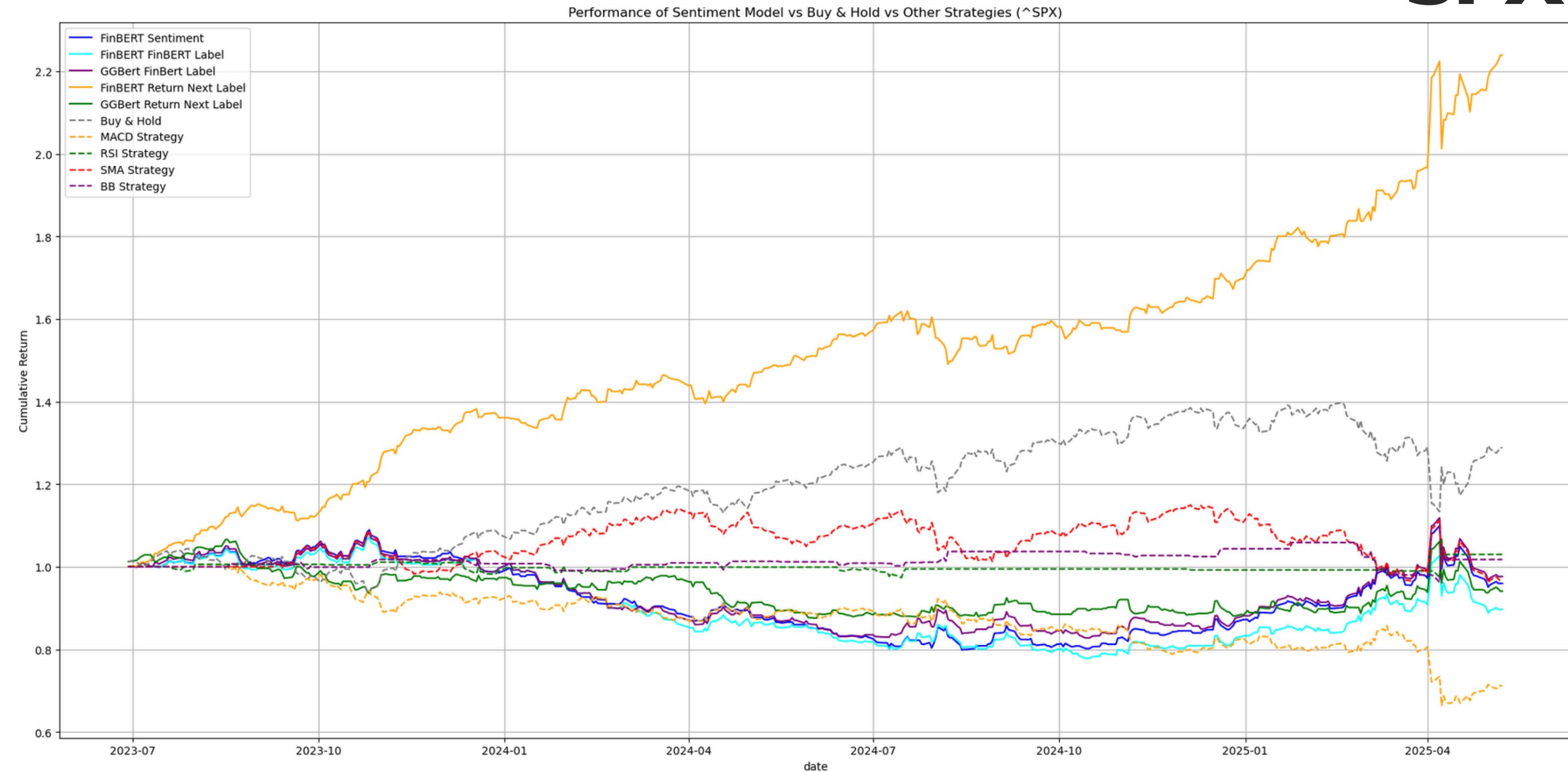
Evaluation

NVDA



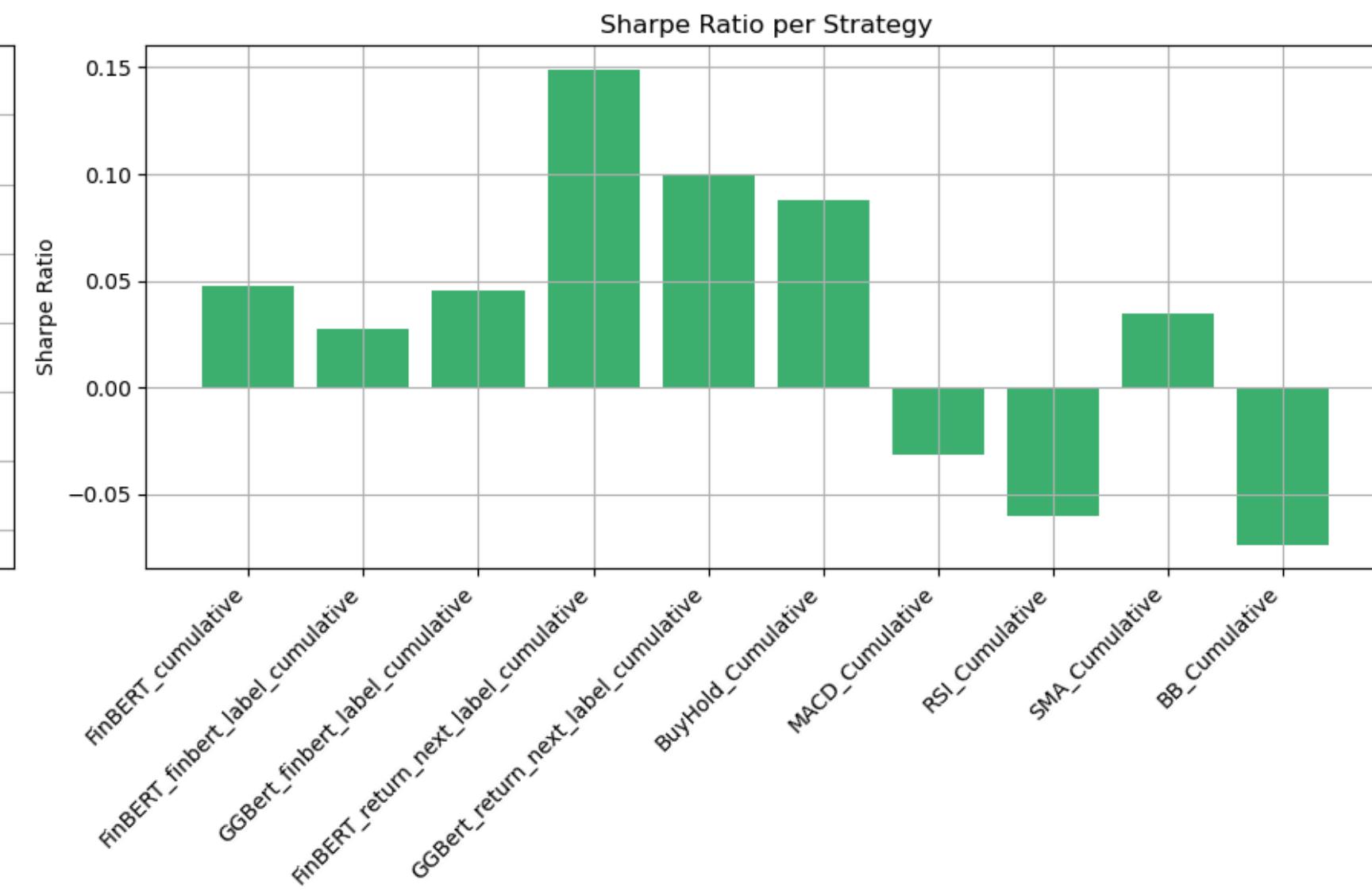
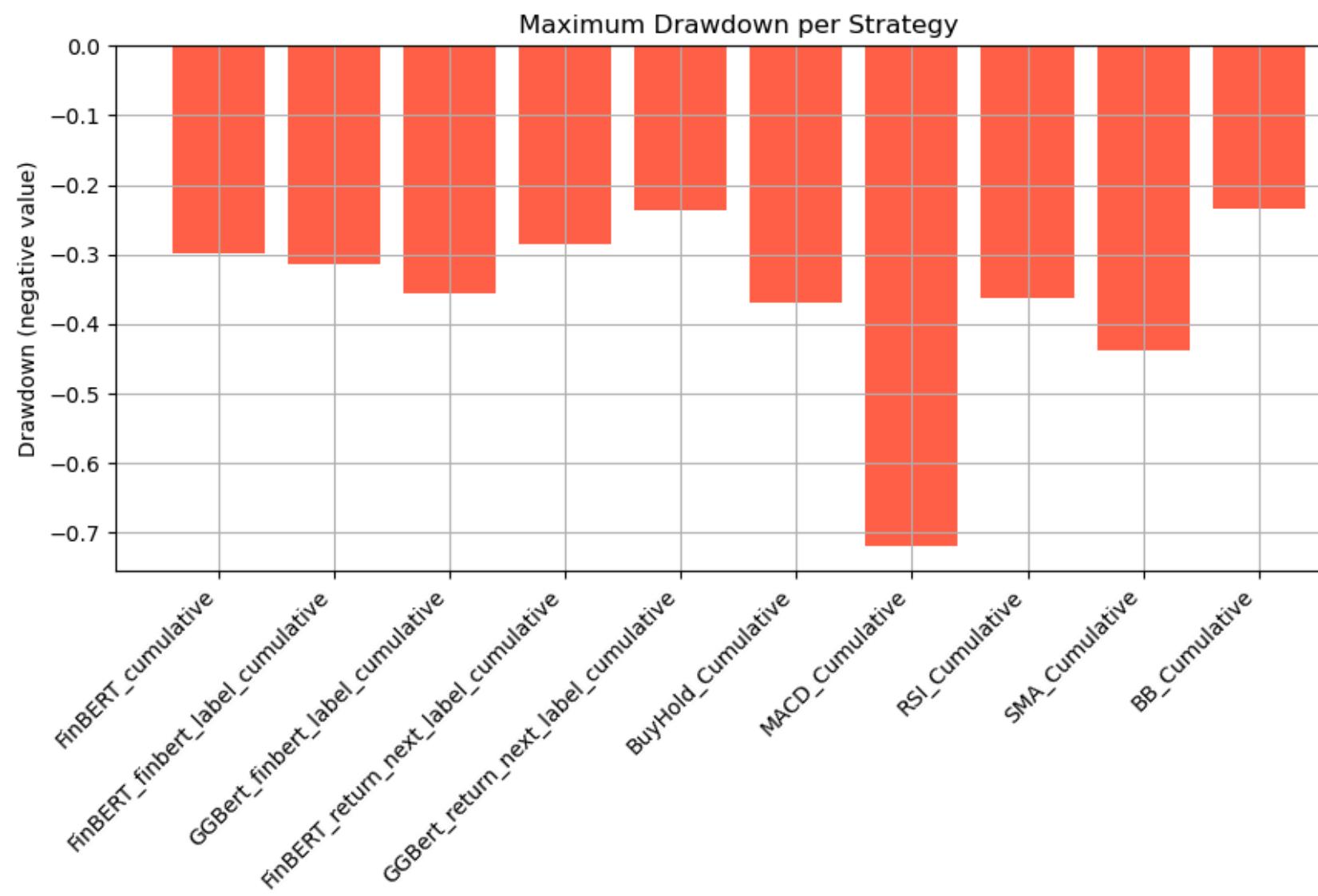
Evaluation

^SPX



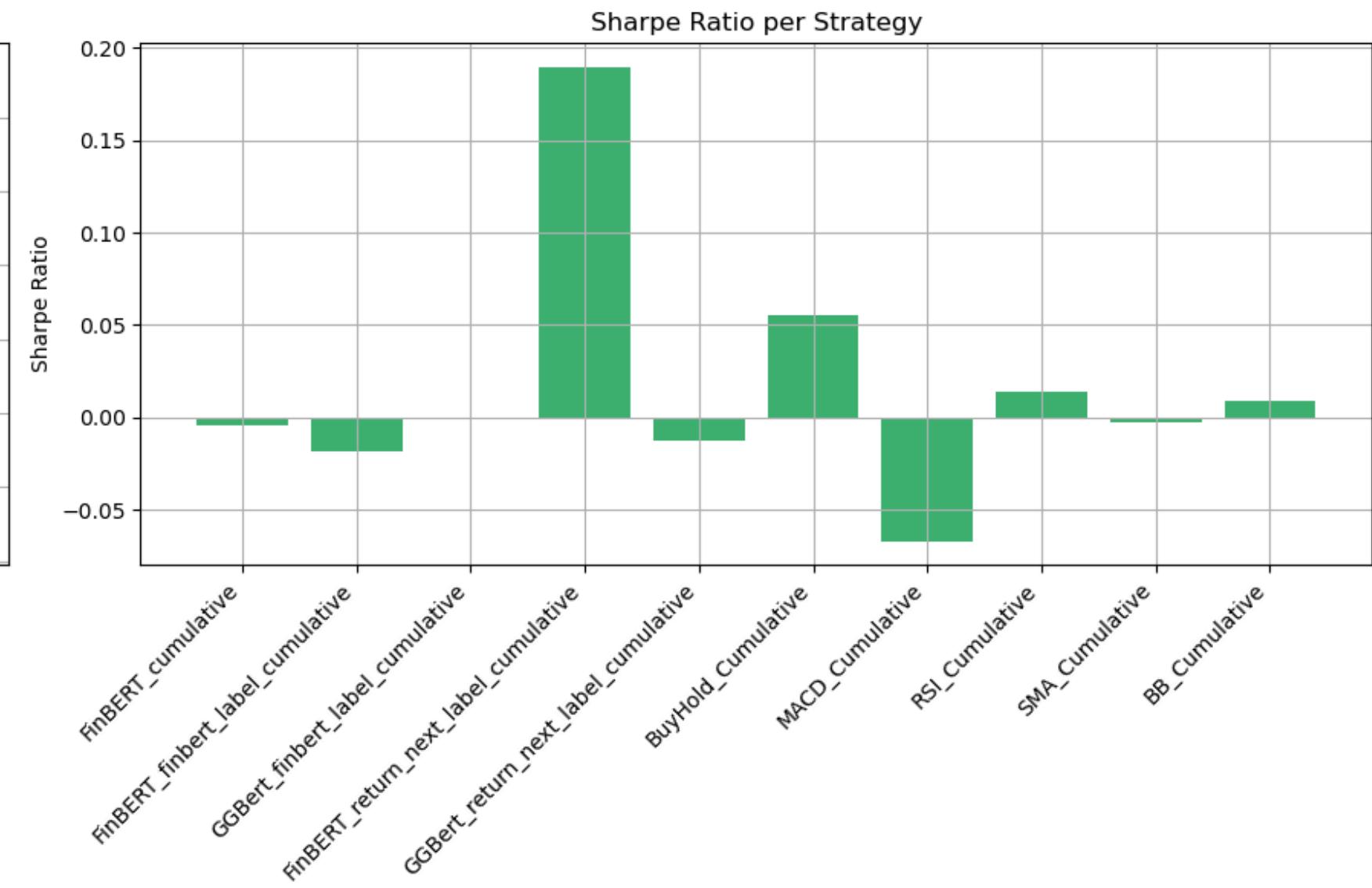
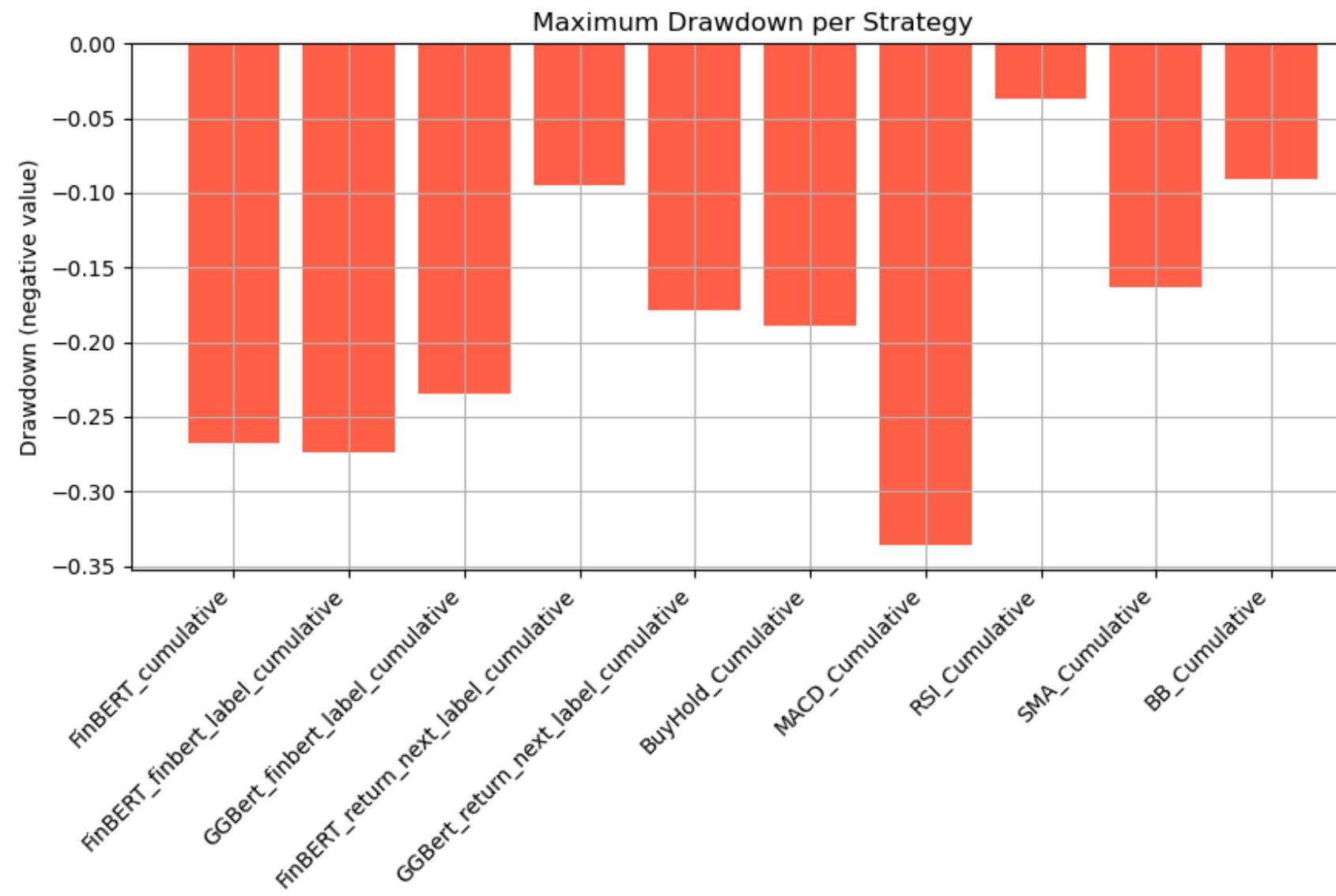
Evaluation

NVDA



Evaluation

^SPX



Evaluation

- **initial_balance = 1,000,000**

```
def simulate_trading_strategy(x):
    initial_balance = 1000000
    capital = initial_balance
    position = 0

    for i in range(len(df)):
        if df[x][i] == 1 and position == 0:
            position = capital / df["Close"][i]
            capital = 0
            #print(f"Buying at {df['Close'][i]} on {df['date'][i]}")
        elif df[x][i] == -1 and position > 0:
            capital = position * df["Close"][i]
            position = 0
            #print(f"Selling at {df['Close'][i]} on {df['date'][i]}")
        elif df[x][i] == 0:
            #print(f"Holding at {df['Close'][i]} on {df['date'][i]}")
            pass

    final_value = capital if capital > 0 else position * df["Close"].iloc[-1]
    returns = (final_value - initial_balance) / initial_balance * 100
    print(f"Strategy from: {x}")
    print(f"Profit/Loss: {final_value - initial_balance:.2f}")
    print(f"Portfolio Return: {returns:.2f}% from {initial_balance} to {final_value:.2f} over {len(df)} days\n")

✓ 0.0s
```

Charging fee is not calculated

Strategy from: finBert
Profit/Loss: 1278578.06
Portfolio Return: 127.86% from 1000000 to 2278578.06 over 372 days

Strategy from: finBert_finbert_label
Profit/Loss: 943403.54
Portfolio Return: 94.34% from 1000000 to 1943403.54 over 372 days

Strategy from: GGBert_finbert_label
Profit/Loss: 1176250.69
Portfolio Return: 117.63% from 1000000 to 2176250.69 over 372 days

Strategy from: finBert_return_next_label
Profit/Loss: 2177838.49
Portfolio Return: 217.78% from 1000000 to 3177838.49 over 372 days

Strategy from: GGBert_return_next_label
Profit/Loss: 1592840.48
Portfolio Return: 159.28% from 1000000 to 2592840.48 over 372 days

Strategy from: sma_signal
Profit/Loss: 1194513.04
Portfolio Return: 119.45% from 1000000 to 2194513.04 over 372 days

Strategy from: rsi_signal
Profit/Loss: 154914.65
Portfolio Return: 15.49% from 1000000 to 1154914.65 over 372 days

Strategy from: macd_signal_line
Profit/Loss: 335331.35
Portfolio Return: 33.53% from 1000000 to 1335331.35 over 372 days

Strategy from: bb_signal
Profit/Loss: 425764.75
Portfolio Return: 42.58% from 1000000 to 1425764.75 over 372 days



Evaluation

- initial_balance = 1,000,000

```
def simulate_trading_strategy(x):
    initial_balance = 1000000
    capital = initial_balance
    position = 0

    for i in range(len(df)):
        if df[x][i] == 1 and position == 0:
            position = capital / df["Close"][i]
            capital = 0
            #print(f"Buying at {df['Close'][i]} on {df['date'][i]}")
        elif df[x][i] == -1 and position > 0:
            capital = position * df["Close"][i]
            position = 0
            #print(f"Selling at {df['Close'][i]} on {df['date'][i]}")
        elif df[x][i] == 0:
            #print(f"Holding at {df['Close'][i]} on {df['date'][i]}")
            pass

    final_value = capital if capital > 0 else position * df["Close"].iloc[-1]
    returns = (final_value - initial_balance) / initial_balance * 100
    print(f"Strategy from: {x}")
    print(f"Profit/Loss: {final_value - initial_balance:.2f}")
    print(f"Portfolio Return: {returns:.2f}% from {initial_balance} to {final_value:.2f} over {len(df)} days\n")

✓ 0.0s
```

Charging fee is not calculated

Strategy from: finBert
Profit/Loss: 65886.89
Portfolio Return: 6.59% from 1000000 to 1065886.89 over 467 days

Strategy from: finBert_finbert_label
Profit/Loss: 51174.35
Portfolio Return: 5.12% from 1000000 to 1051174.35 over 467 days

Strategy from: GGBert_finbert_label
Profit/Loss: 33489.56
Portfolio Return: 3.35% from 1000000 to 1033489.56 over 467 days

Strategy from: finBert_return_next_label
Profit/Loss: 689235.53
Portfolio Return: 68.92% from 1000000 to 1689235.53 over 467 days

Strategy from: GGBert_return_next_label
Profit/Loss: 63500.13
Portfolio Return: 6.35% from 1000000 to 1063500.13 over 467 days

Strategy from: sma_signal
Profit/Loss: 132303.73
Portfolio Return: 13.23% from 1000000 to 1132303.73 over 467 days

Strategy from: rsi_signal
Profit/Loss: 84626.54
Portfolio Return: 8.46% from 1000000 to 1084626.54 over 467 days

Strategy from: macd_signal_line
Profit/Loss: 80230.54
Portfolio Return: 8.02% from 1000000 to 1080230.54 over 467 days

Strategy from: bb_signal
Profit/Loss: 196422.37
Portfolio Return: 19.64% from 1000000 to 1196422.37 over 467 days



Conclusion

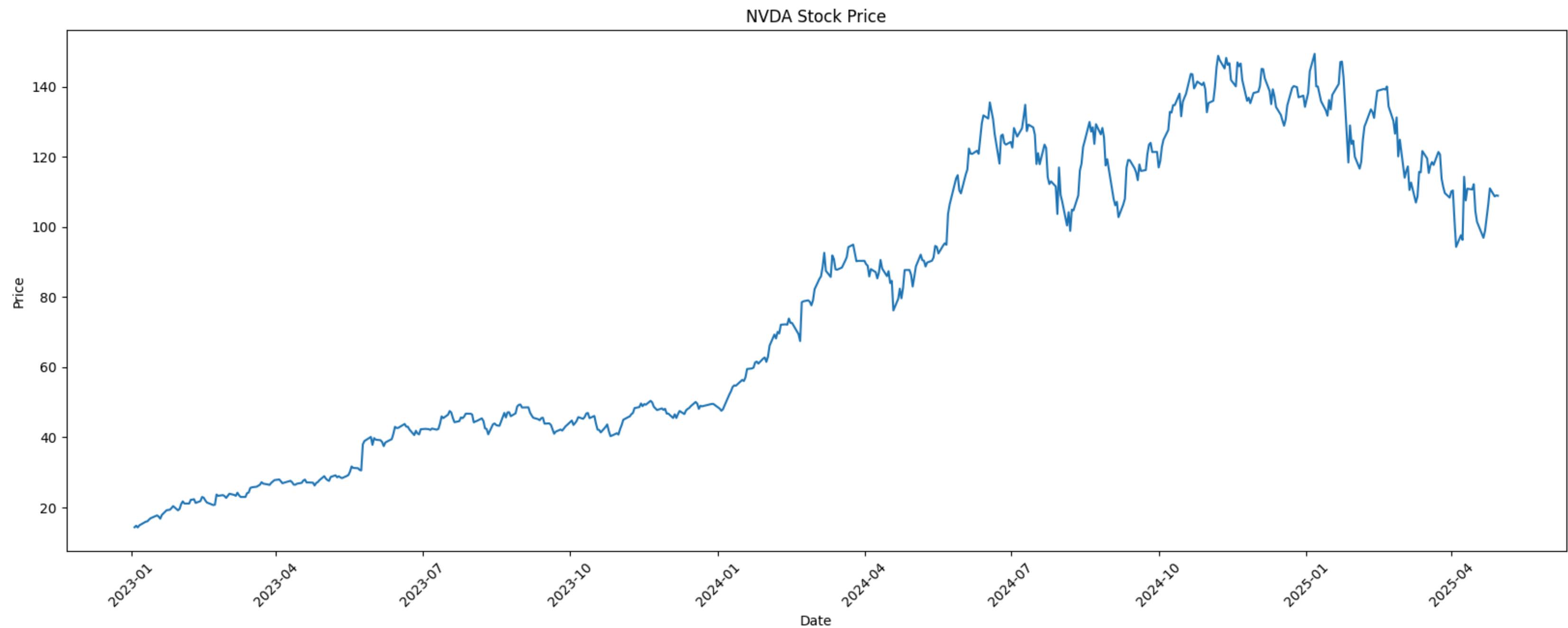
- **NLP model can be used for strategy trading**
- **Using return_next as labeler then fine-tuning has a better result**
- **Sentiment from news has potential to predict the signal**
- **We can fine-tune our model with another feature to enhance its performance**

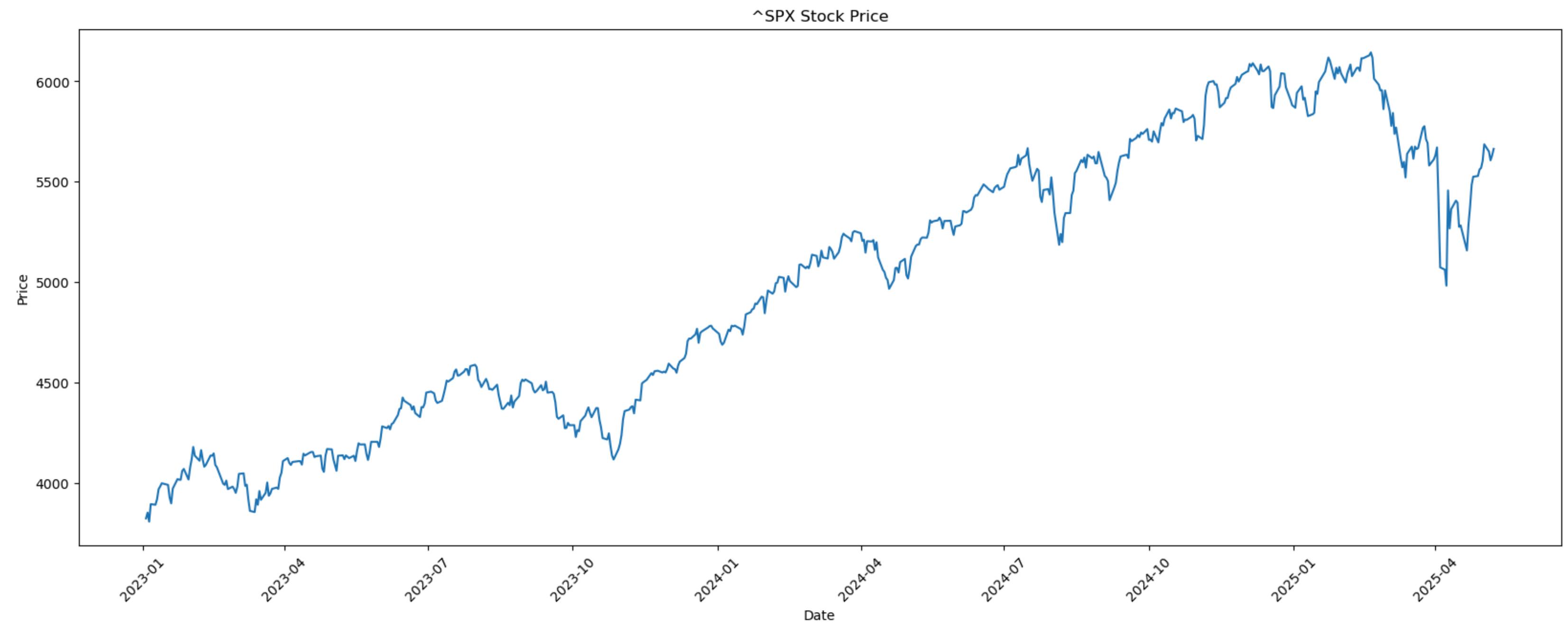
Thank you

[Github](#)

Test sets

- [Sentiment Analysis for Financial News](#)
- [News Sentiment Analysis](#)
- [Twitter Financial News Sentiment Dataset](#)
-  [Stock Market Tweet | Sentiment Analysis lexicon](#)





ตัวอย่างการ Fine-tune FinBERT ในงานวิจัยและโปรเจกต์

shekolla/finbert-financial-sentiment

- แนวคิด:
 - ใช้ FinBERT และ DistilRoBERTa ที่ fine-tune สำหรับข่าวการเงิน
 - รองรับการวิเคราะห์ sentiment และ emotional tone (เช่น joy, fear, anger)

ตัวอย่างการ Fine-tune FinBERT ในงานวิจัยและโปรเจกต์

LikithMeruvu/FinBert-Finetuning-for-Stock-Sentiment

- Dataset: ใช้ข่าวหุ้นจาก Kaggle ที่มีการ preprocess แล้ว
- Label: ข่าวถูกจัดกลุ่มเป็น Positive, Negative และ Neutral โดยใช้ข้อมูลที่มีอยู่
- ผลลัพธ์: หลังจากเทรนประมาณ 5 ชั่วโมงบน GPU ได้ accuracy ประมาณ 81–82%

ตัวอย่างการ Fine-tune FinBERT ในงานวิจัยและโปรเจกต์

kdave/FineTuned_Finbert

- บริบท: ไม่เดลถูก fine-tune สำหรับข่าวหุ้นในตลาดอินเดียโดยเฉพาะ
- วิธีการ:
 - Label: ใช้ข่าวที่มีการสรุปด้วย T5-base และนำมาจัดกลุ่ม sentiment
 - ไมเดล: ใช้ FinBERT ที่ fine-tune เพิ่มเติมบนข่าวหุ้นอินเดีย

Cumulative Return

Cumulative Return คือผลตอบแทนรวม “เป็นเปอร์เซ็นต์” เช่น ถ้าคุณลงทุน เงินจำนวนใด ก็ตาม และให้กลยุทธ์ทำงานตามนั้น เงินคุณจะโตขึ้นกี่เปอร์เซ็นต์ หรือหดลงกี่เปอร์เซ็นต์

ถ้า Cumulative Return = 0.25 หรือ +25%
หมายความว่า ถ้าคุณลงทุน:

- 100,000 บาท → คุณจะได้กำไร 25,000 บาท
- 50,000 บาท → กำไร 12,500 บาท
- 200,000 บาท → กำไร 50,000 บาท

เพราะ cumulative return คิดเป็น “สัดส่วน” ของเงินเริ่มต้น (initial capital)

1. คำนวณ Strategy Return:

python

Copy Edit

```
df['Strategy_Return'] = df['Sentiment'].shift(1) * df['Return']
```

- ใช้ `.shift(1)` เพราะ ข่าวเกิดก่อนการเทรด 1 วัน

วันที่	Sentiment	Return	Strategy_Return	⋮
2024-01-01	1	0.01	NaN	
2024-01-02	0	-0.02	$1 * -0.02 = -0.02$	
2024-01-03	-1	0.01	$0 * 0.01 = 0$	
2024-01-04	1	0.03	$-1 * 0.03 = -0.03$	

2. คำนวณ Cumulative Return:

python

 Copy  Edit

```
df['Cumulative_Return'] = (1 + df['Strategy_Return'].fillna(0)).cumprod()
```

วันที่	Strategy_Return	Cumulative_Return
2024-01-01	NaN	1.0000
2024-01-02	-0.02	$1 * (1 - 0.02) = 0.98$
2024-01-03	0	$0.98 * 1 = 0.98$
2024-01-04	-0.03	$0.98 * (1 - 0.03) = 0.9506$

```
# Configuration
MODEL_NAME = "ProsusAI/finbert"
MAX_LENGTH = 512
BATCH_SIZE = 16
EPOCHS = 3
LEARNING_RATE = 2e-5
OUTPUT_DIR = "./stock_sentiment_results"
```

MACD (Moving Average Convergence Divergence)

- คือตัวชี้วัดแนวโน้มที่ดูความสัมพันธ์ระหว่างเส้นค่าเฉลี่ยเคลื่อนที่สองเส้น (EMA 12 และ EMA 26)
- สูตรหลัก:
 - $\text{MACD} = \text{EMA}(12) - \text{EMA}(26)$
 - พร้อมเส้น Signal Line (EMA 9 ของ MACD)
- การใช้งาน:
 - ถ้า $\text{MACD} > \text{Signal} \rightarrow$ สัญญาณ ซื้อ
 - ถ้า $\text{MACD} < \text{Signal} \rightarrow$ สัญญาณ ขาย

RSI (Relative Strength Index)

- คือตัวชี้วัดไม่เมนต์ วัดความแรงของราคากุ้นในช่วง 14 วันที่ผ่านมา
- ช่วงค่า: 0 – 100
- การใช้งาน:
 - $RSI > 70 \rightarrow Overbought$ (อาจปรับตัวลง)
 - $RSI < 30 \rightarrow Oversold$ (อาจเด้งกลับขึ้น)

SMA (Simple Moving Average)

- คือค่าเฉลี่ยของราคาก้อนย้อนหลังในช่วงเวลาที่กำหนด (เช่น 10, 20, 50 วัน)
- สูตร: $SMA = (P_1 + P_2 + \dots + P_n) / n$
- การใช้งาน:
 - ใช้ดูแนวโน้มระยะสั้น/ยาว
 - ถ้าราคาขึ้นเหนือ SMA \rightarrow แนวโน้มขาขึ้น
 - ราคาลงต่ำกว่า SMA \rightarrow แนวโน้มขาลง

BB (Bollinger Bands)

- คือตัวชี้วัดความผันผวนของราคาหุ้นโดยใช้ SMA และส่วนเบี่ยงเบนมาตรฐาน (standard deviation)
- สูตรหลัก:
 - Middle Band = $SMA(20)$
 - Upper Band = $SMA + 2\sigma$
 - Lower Band = $SMA - 2\sigma$
- การใช้งาน:
 - ถ้าราคาแตะ Upper Band → อาจ Overbought
 - ถ้าแตะ Lower Band → อาจ Oversold

Maximum Drawdown (MDD)

"ขาดทุนหนักที่สุดจากจุดสูงสุด (peak) ไปยังจุดต่ำสุด (trough)"

- วัดว่าในช่วงเวลาใดเวลาหนึ่ง กลยุทธ์มีการ "ร่วงหนักที่สุด" เก่าได้
- แสดง ความเสี่ยงด้าน downside ของกลยุทธ์
- ค่าเป็น ลบ เสมอ → ยิ่งลบมาก ยิ่งแย่
- ตัวอย่าง: MDD = -0.70 หมายถึง เคยขาดทุน 70% จากจุดสูงสุด

ใช้เปรียบเทียบว่าไมเดลaid "ร่วงแรงสุด" และไมเดลaid "เสถียรกว่า"

Sharpe Ratio

"ผลตอบแทนต่อหน่วยความเสี่ยง"

- ยิ่งค่า Sharpe สูง → กลยุทธ์ยิ่ง "คุ้มความเสี่ยง"
- ค่า Sharpe:
 - < 0 → ผลตอบแทนไม่คุ้มเสี่ยง
 - ~0.5 → ปานกลาง
 - 1 → ดีมาก

หมายสำคัญ: ใช้เปรียบเทียบโดยเดลที่ผลตอบแทนใกล้เคียงกัน แต่ความผันผวนต่างกัน

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

- R_p : ผลตอบแทนกลยุทธ์
- R_f : อัตราผลตอบแทนไร้ความเสี่ยง (Risk-free rate)
- σ_p : ความผันผวนของผลตอบแทน