



「Python在金融領域上的應用」

股票分析 LineBot - FinGPT

組員：

資財三甲111AB0011吳承諺



FinGPT 發想



對股市不熟

- 股市小白，對分析股票有困難



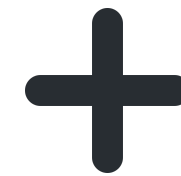
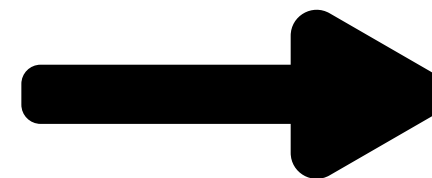
時間不足

- 想存股，但沒時間分析



被心態影響

- 散戶心態，停利不停損



快速分析、簡單易懂
FinGPT幫你解決！

FinGPT 簡介

- 一款自行研發的 Line官方帳號
- 結合股市資訊 & AI分析



▲掃我加入好友！



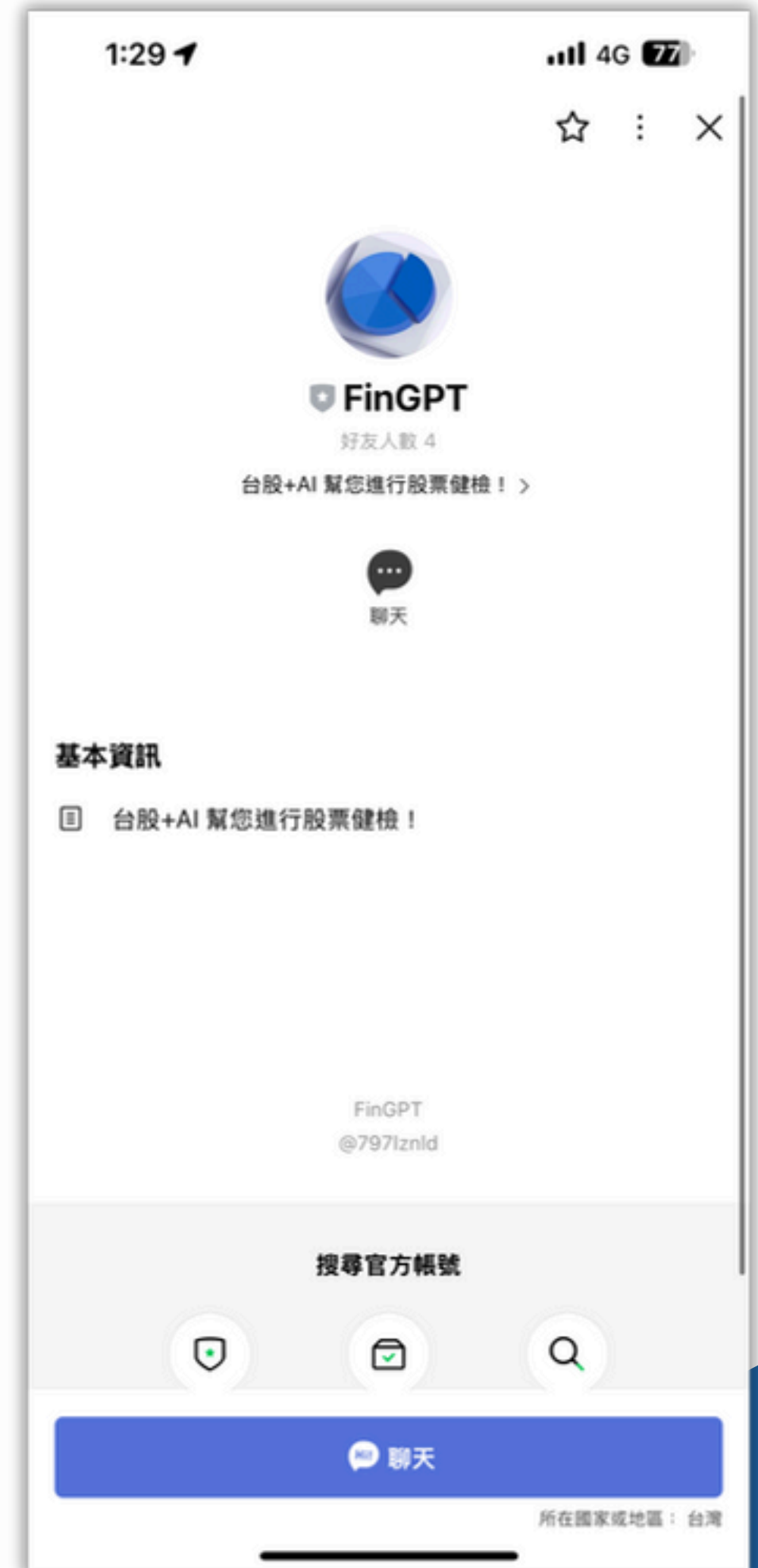
Line官方帳號

- 不必安裝其他APP
- 一鍵加入好友



ChatGPT

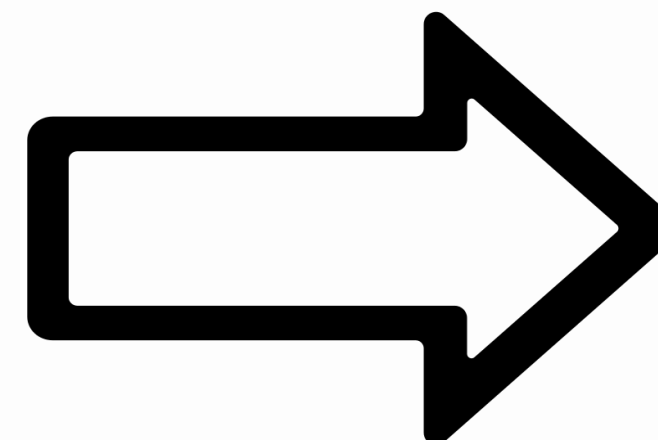
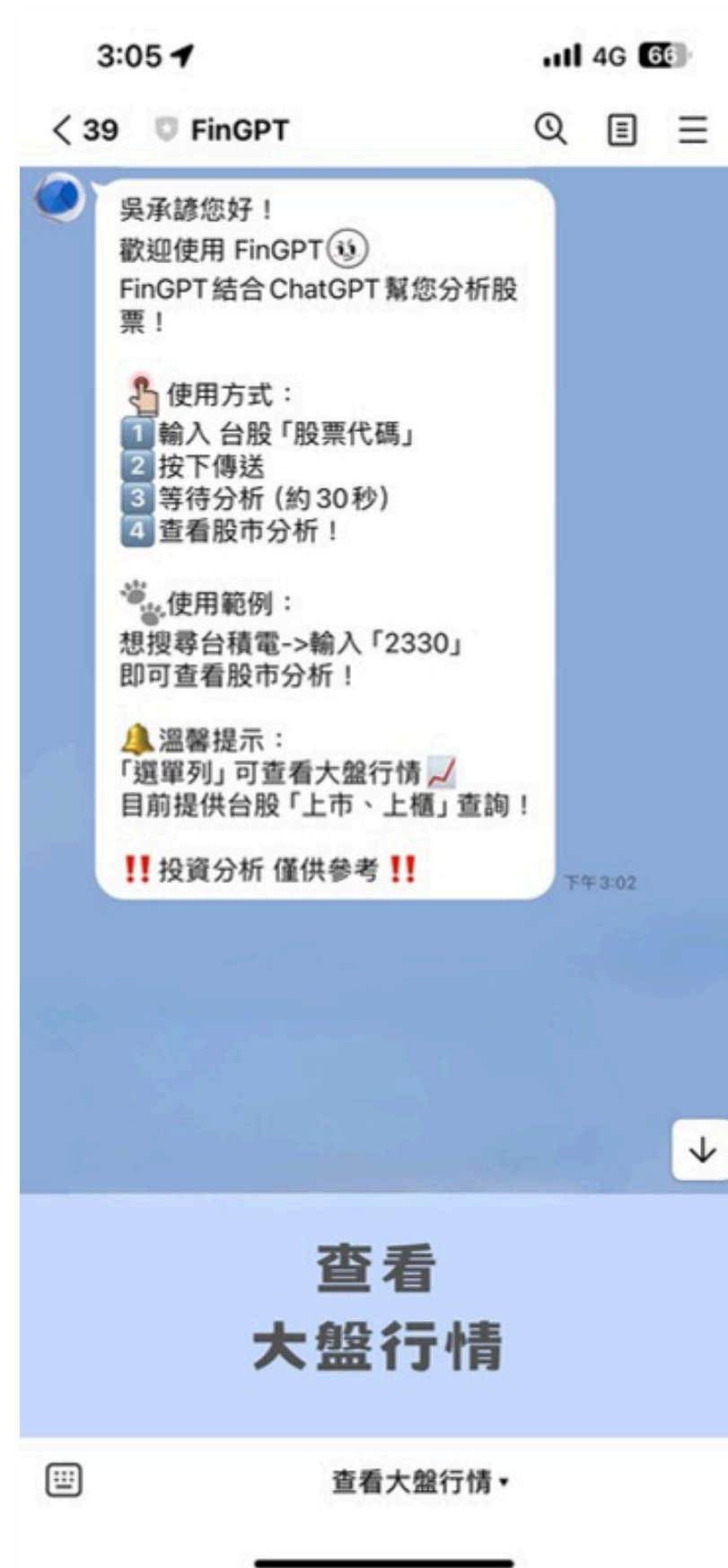
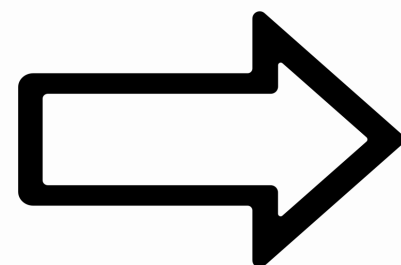
- 串接AI工具 - ChatGPT
- 快速分析、易讀易懂



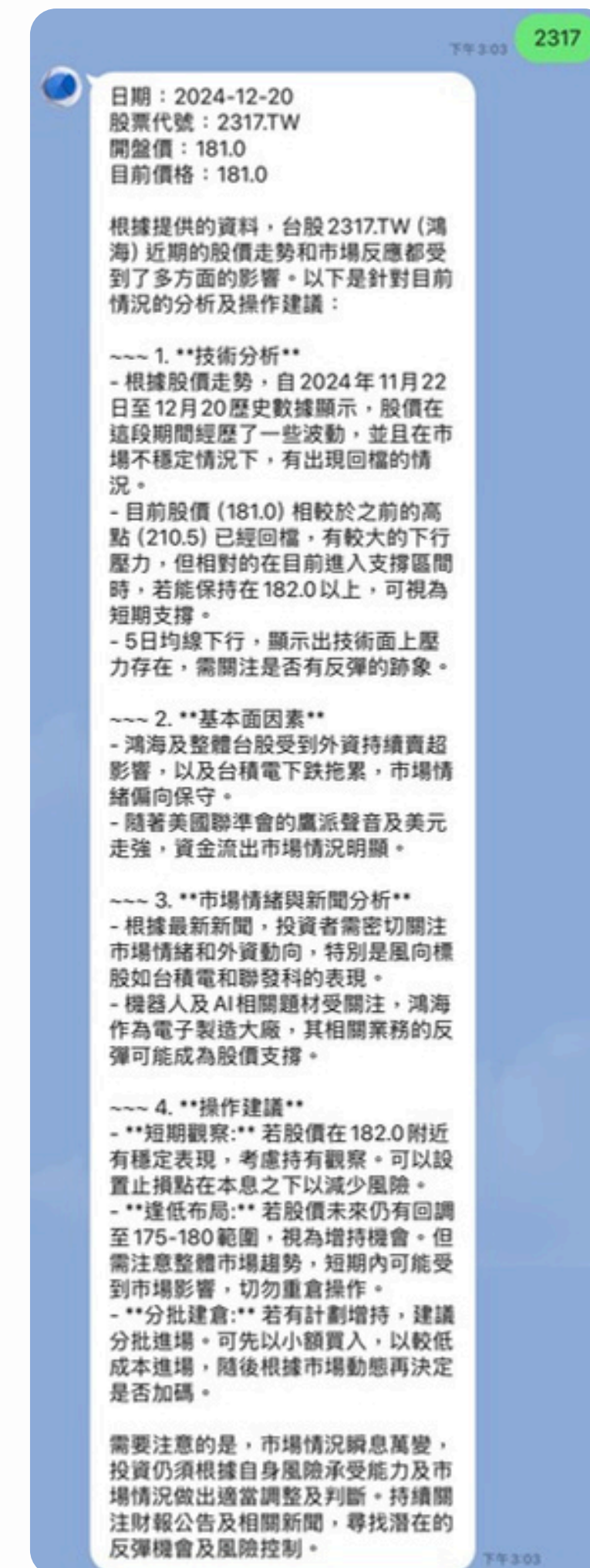
FinGPT 操作介面



▲掃我加入好友！



傳送股票代號
等待約1分鐘



FinGPT 架構

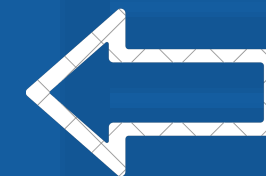


Python



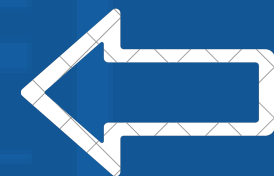
Google Cloud

後台伺服器



Office Account

帳號管理



Line

前端接收訊息

FinGPT 架構 - Python

OpenAI 套件

- 結合「股價」、「新聞」
- ChatGPT 4o-mini 分析



yfinance 套件

- 取得「股價」資訊
- 進行「技術面」分析



Anue 鉅亨網 (爬蟲)

- 取得「新聞」資訊
- 進行「消息面」分析

FinGPT 後台

-Google Cloud

Google Cloud

Linebot Project

搜尋資源、文件、產品和其他項目 (請按 / 鍵)

搜尋

+

📁

📄

🔍

🔧

⋮

👤

(...) Cloud Run 函式

← 函式詳細資料

編輯

刪除

複製

瞭解詳情

↺

linebotFin Cloud Run function (1st gen)

版本 16 已在 2024年12月20日 凌晨2:54:26 部...

指標

詳細資訊

來源

變數

觸發條件

權限

記錄

測試

記錄

顯示性 預設

篩選條件 搜尋所有欄位和值

🔍

↺

📄

顯示性	時間戳記	函數
>	2024-12-20 14:39:27.899 香港時間	linebotFin Install using:
>	2024-12-20 14:39:27.899 香港時間	linebotFin pip install requests_html
>	2024-12-20 14:39:27.899 香港時間	linebotFin
>	2024-12-20 14:39:27.899 香港時間	linebotFin After installation, you may have to restart your Python session.
>	2024-12-20 15:02:46.687 香港時間	linebotFin Warning - Certain functionality
>	2024-12-20 15:02:46.687 香港時間	linebotFin requires requests_html, which is not installed.
>	2024-12-20 15:02:46.687 香港時間	linebotFin
>	2024-12-20 15:02:46.687 香港時間	linebotFin Install using:
>	2024-12-20 15:02:46.687 香港時間	linebotFin pip install requests_html
>	2024-12-20 15:02:46.687 香港時間	linebotFin
>	2024-12-20 15:02:46.687 香港時間	linebotFin After installation, you may have to restart your Python session.
>	2024-12-20 15:02:46.887 香港時間	linebotFin l3wujqqlv18 Function execution started
>	2024-12-20 15:02:46.953 香港時間	linebotFin l3wujqqlv18 ImmutableMultiDict([[]])
>	2024-12-20 15:02:46.955 香港時間	linebotFin l3wujqqlv18 Function execution took 68 ms, finished with status code: 200
>	2024-12-20 15:02:46.961 香港時間	linebotFin l3wud2eey188 Function execution started
>	2024-12-20 15:02:46.967 香港時間	linebotFin l3wud2eey188 ImmutableMultiDict([[]])
>	2024-12-20 15:02:46.967 香港時間	linebotFin l3wud2eey188 Function execution took 6 ms, finished with status code: 200
>	2024-12-20 15:03:22.578 香港時間	linebotFin l3wuyj4ri559 Function execution started
>	2024-12-20 15:03:57.413 香港時間	linebotFin l3wuyj4ri559 Token Usage: 12529
>	2024-12-20 15:03:57.632 香港時間	linebotFin l3wuyj4ri559 2317 89f8f46e33374993a9f7f76d27ef1d47
>	2024-12-20 15:03:57.634 香港時間	linebotFin l3wuyj4ri559 Function execution took 39855 ms, finished with status code: 200
>	2024-12-21 17:13:45.965 香港時間	linebotFin Warning - Certain functionality
>	2024-12-21 17:13:45.965 香港時間	linebotFin requires requests_html, which is not installed.
>	2024-12-21 17:13:45.965 香港時間	linebotFin
>	2024-12-21 17:13:45.965 香港時間	linebotFin Install using:
>	2024-12-21 17:13:45.965 香港時間	linebotFin pip install requests_html
>	2024-12-21 17:13:45.965 香港時間	linebotFin
>	2024-12-21 17:13:45.965 香港時間	linebotFin After installation, you may have to restart your Python session.
>	2024-12-21 17:13:45.763 香港時間	linebotFin vvxz13bq08v Function execution started
>	2024-12-21 17:14:01.892 香港時間	linebotFin vvxz13bq08v Token Usage: 8815
>	2024-12-21 17:14:01.326 香港時間	linebotFin vvxz13bq08v 2317 af9d4401a321436c016730ce11ffc9f6
>	2024-12-21 17:14:01.327 香港時間	linebotFin vvxz13bq08v Function execution took 15564 ms, finished with status code: 200

FinGPT 程式碼介紹



yahoo_fin套件
獲取股價

```
26 # 取得股價資料
27 ✓ def catch_Stock(stock):
28     for suffix in [".TW", ".TWO", ""]:
29         try:
30             id = f'{stock}{suffix}'
31             data = get_data(id)
32             if suffix!="":
33                 data["volume"]=data["volume"]*0.001
34             break
35         except:
36             continue
37     data = data.round(2).iloc[-25:,:6]
38     del data["adjclose"]
39     # data.index.name = "Date"
40     data.index = data.index.strftime("%Y-%m-%d")
41     data["5MA"]=data["close"].rolling(5).mean()
42     data = data.dropna()
43     if id[-3:]=="TW":
44         data = data.drop("2024-11-20", errors='ignore') # 「上市」資料異常
45     return data, id
```

取得「上市or上櫃」資料

選擇近一個月資料

計算 5MA

FinGPT 程式碼介紹



Anue 鉅亨網
透過爬蟲 取得新聞

```
54 # 取得「新聞資訊」
55 def get_news(stock_id):
56     data = []
57     col = ["Date", "Title", "Content"]
58     if stock_id=="^TWII":
59         url = f"https://api.cnyes.com/media/api/v1/newslist/category/tw_stock_news?page=1&limit=10&isCategoryHeadline=1"
60         json_news = requests.get(url).json()['items']['data']
61         for i in range(10):
62             content = json_news[i]["content"].replace("<p>","").replace("</p>","").replace("&nbsp;","").replace
63             summary = json_news[i]["summary"]
64             t = json_news[i]["publishAt"]+28800
65             news_time = time.strftime("%Y/%m/%d", time.gmtime(t))
66             data.append([news_time,summary,content])
67     else:
68         stock_name = catch_stock_name(stock_id)
69         url = f"https://ess.api.cnyes.com/ess/api/v1/news/keyword?q={stock_name}&limit=10&page=1"
70         json_news = requests.get(url).json()['data']['items']
71         for item in json_news:
72             id = item['newsId']
73             title = item['title']
74             t = item['publishAt']+28800
75             news_time = time.strftime("%Y/%m/%d", time.gmtime(t))
76             news_url = f"https://news.cnyes.com/news/id/{id}"
77             news = requests.get(news_url).text
78             news_bs = bs(news, 'html.parser')
79             news_find = news_bs.find_all("p")[2:-9]
80             news_data = "\n".join(x.text.strip() for x in news_find)
81             data.append([news_time,title,news_data])
82
83     df = pd.DataFrame(data,columns=col)
84     df_str = df.to_string().replace(" ", "")
85     return df_str
```

← 大盤新聞

← 個股新聞

詳細程式碼：<https://github.com/Ynn622/FinGPT/blob/main/function-source>

FinGPT 程式碼介紹



OpenAI分析

```
17     # OpenAI聊天
18     def talk(chatbot,record):
19         response = chatbot.chat.completions.create(
20             model="gpt-4o-mini",
21             messages=record,
22             max_tokens = 16000
23         )
24         return response

87     # 生成
88     def generate(input):
89         analysis = ""
90         try:
91             stock_data, stock_id = catch_Stock(input)
92             news_data = get_news(stock_id)
93             analysis += f"日期: {stock_data.index[-1]} \n股票代號: {stock_id} \n開盤價: {stock_data.iloc[-1,0]} \n目前價格: {stock_data.iloc[-1,3]}\n\n"
94             # 初始化ai設定
95             key = os.environ["OpenAI_key"]
96             ai = OpenAI(api_key=key)
97             # 對話開始
98             talked = [{"role": "user", "content": f"這是台股{stock_id}的資料\n{stock_data.to_string()} 以下是相關近期新聞可參考: {news_data} 請分析 並給出操作建議 Reply in 繁體中文"}]
99             del stock_data
100             response = talk(ai,talked)
101             print("Token Usage:",response.usage.total_tokens)
102             analysis += response.choices[0].message.content
103             analysis = analysis.replace("#", "~")
104         except Exception as e:
105             analysis = "錯誤! 請檢查代碼 或稍後再試~"
106             print(e)
107         return analysis
```

詳細程式碼：<https://github.com/Ynn622/FinGPT/blob/main/function-source>

FinGPT 程式碼介紹



Line 收發訊息

```
109 # LineBot 接收&傳送
110 ✓ def linebot(request):
111     try:
112         access_token = os.environ["Line_token"]
113         secret = os.environ["Line_secret"]
114         body = request.get_data(as_text=True)
115         json_data = json.loads(body)
116         line_bot_api = LineBotApi(access_token)
117         handler = WebhookHandler(secret)
118         signature = request.headers['X-Line-Signature']
119         handler.handle(body, signature)
120         msg = json_data['events'][0]['message']['text']
121         tk = json_data['events'][0]['replyToken']
122         ans = generate(msg)
123         line_bot_api.reply_message(tk, TextSendMessage(ans))
124         print(msg, tk)
125     except:
126         print(request.args)
127     return 'OK'
```

← Line 驗證 & 接收訊息

← 生成回傳文字（上頁）

← Line 回傳訊息

資料來源

- <https://tw.stock.yahoo.com>
- <https://developers.line.biz/en>
- <https://www.cnyes.com>
- <https://openai.com>
- <https://chatgpt.com>
- <https://console.cloud.google.com>
- <https://steam.oxxostudio.tw/category/python/example/line-webhook.html>
- <https://www.youtube.com/watch?v=Mw3cODdkaFM&t=20s>





THANK YOU!



資財三甲111AB0011吳承諺

