股票管理分析系統

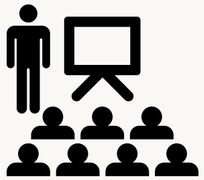
指導教授：羅峻旗 副教授
組員：楊依姍、林倩茹、蔡謙雩

Overview

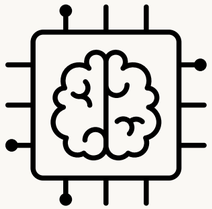
- 研究動機
- 研究目標
- 系統架構
- 研究步驟
- 系統展示
- 目前問題
- 未來展望
- 結論

研究動機

股票市場資訊繁雜且快速變動。



新手與進階投資者缺乏整合化工具。



AI 預測模型缺乏模擬驗證

研究目標

- 提升投資決策效率與精準度:

整合多種資料來源並結合 **AI** 股價預測，可幫助投資者快速判斷買賣時機，提高決策效率與精準度。

- 支援個人化投資管理與學習:

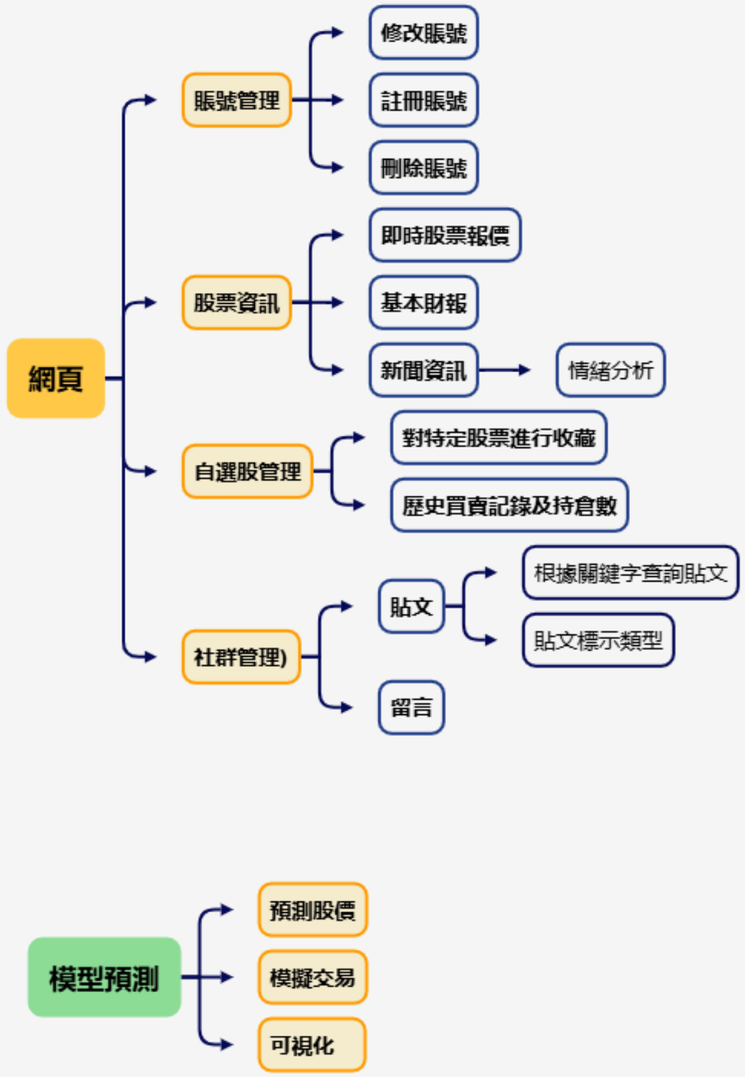
提供個人化投資記錄與視覺化圖表，幫助用戶管理與學習投資策略。

- 驗證 **AI** 預測與交易模擬的應用價值

結合 **AI** 預測與模擬交易策略，驗證 **AI** 在實際應用效益，並提供未來改進的參考依據。

系統架構

系統介紹


網頁
賬號管理
修改賬號
註冊賬號
刪除賬號
股票資訊
即時股票報價
基本財報
新聞資訊
情緒分析
自選股管理
對特定股票進行收藏
歷史買賣記錄及持倉數
社群管理)
貼文
根據關鍵字查詢貼文
貼文標示類型
留言
模型預測
預測股價
模擬交易
可視化

研究步驟

分析系統需求並規劃網頁架構

設計並實作網頁前後端功能

建立資料取得流程

進行資料前處理與特徵工程

訓練並測試 **LSTM** 預測模型

規劃模型結果的可視化呈現方式

技術分析

使用技術：

-開發語言：**Python**、**JavaScript**

-前端技術：**HTML**、**Bootstrap**、**Chart.js**、**Plotly.js**

-後端框架：**Django**

-資料串接：**GNews API**、**Yahoo Finance API**

-資料庫：**MySQL**

-模型：**LSTM**

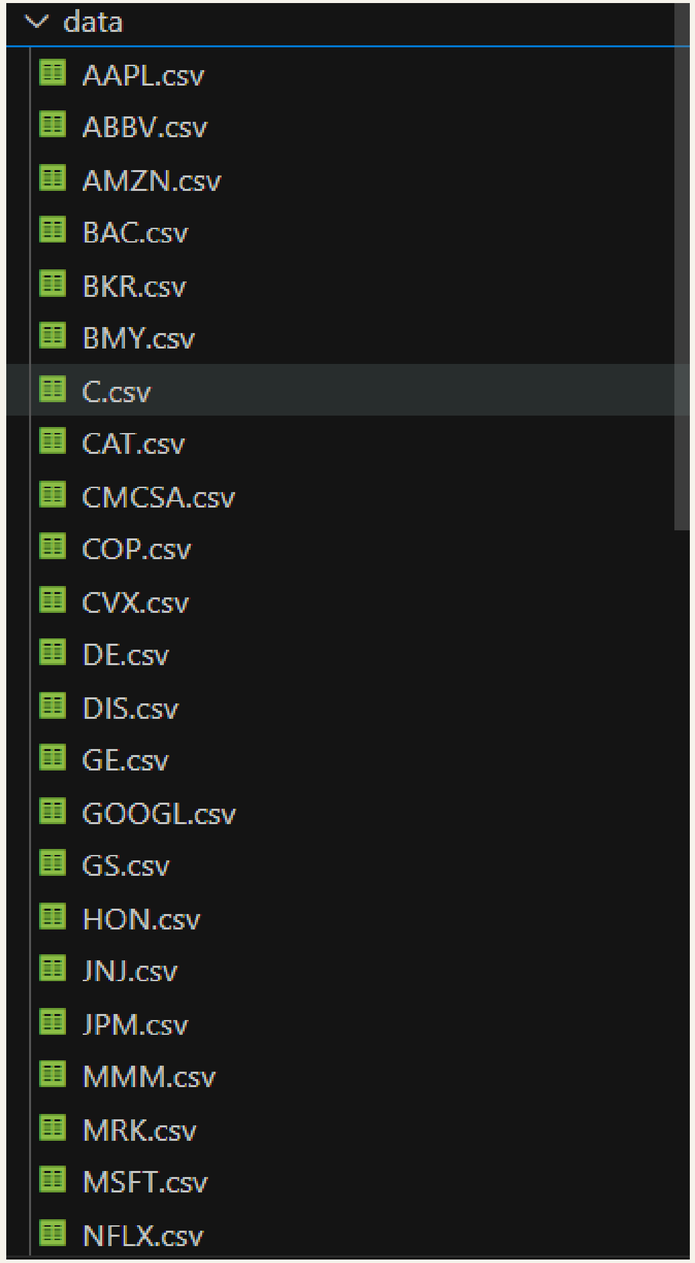
開發工具：**Visual Studio Code**、**Git**

系統介面展示

模型訓練步驟

step1.

批次下載各檔股票的歷史資料，利用 yfinance API取得股價與成交量等資訊，依日期計算 MA、RSI、MACD、布林通道、ATR 等技術指標，整理成以日期為索引的結構化表格資料，最後將每檔股票輸出為對應的 CSV 檔供後續模型訓練使用。




```python
# ============================================================
# 数据下载/获取函数
# ============================================================
def get_stock_data(ticker, start_date, end_date):

    data = yf.download(ticker, start=start_date, end=end_date)
    if data.empty:
        print(f"⚠️ {ticker} 下载失败或返回空数据。")
        return pd.DataFrame()

    data = calculate_technical_indicators(data, start_date, end_date)
    return data
```

模型訓練步驟

step2.

先將歷史股票特徵資料標準化並切成 LSTM 所需的時間序列，接著進行模型訓練以學習漲跌幅規律，最後把預測的「下一日漲跌百分比」轉換回實際股價，生成對應日期的預測結果，用於後續績效分析與視覺化。

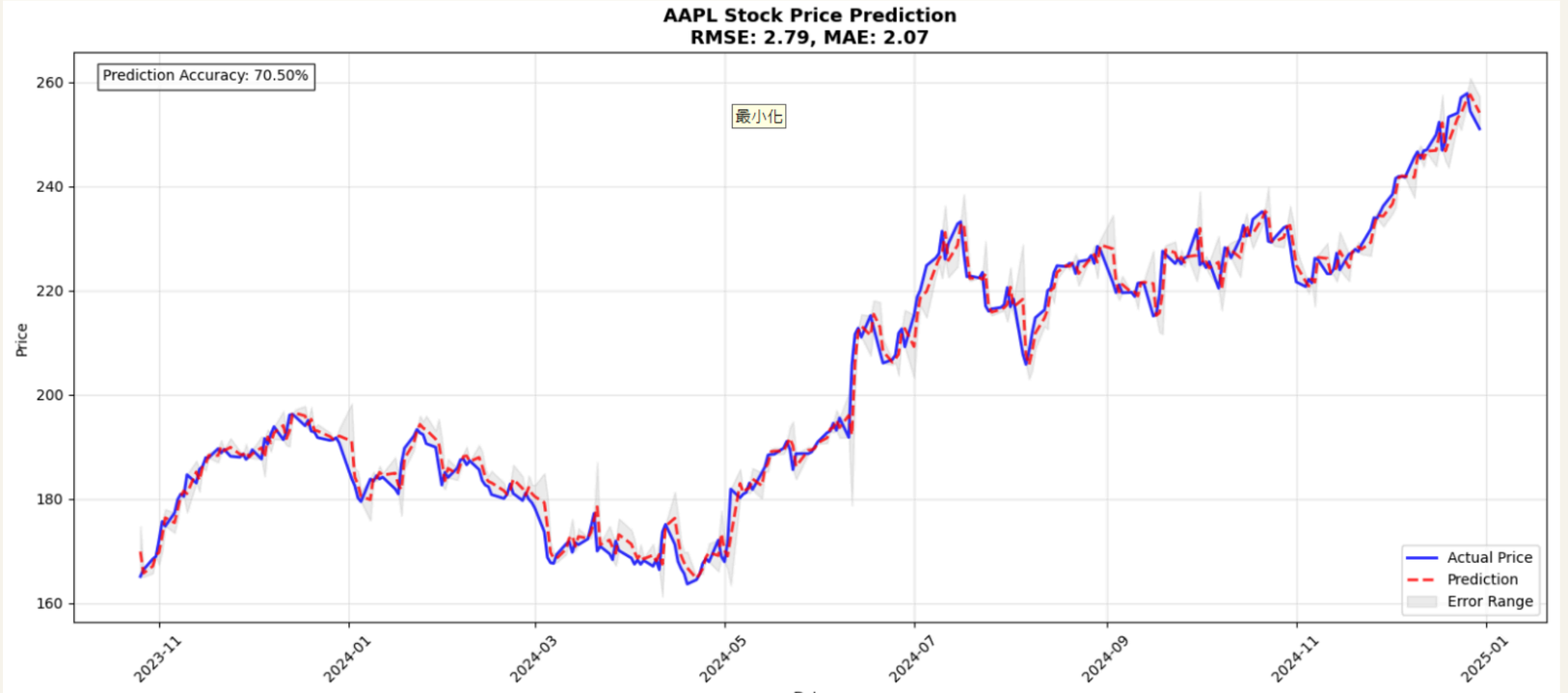
```python
# ==================== 预测 ====================
model.eval()
predictions, test_indices = [], []
valid_dates = y.index

with torch.no_grad():
    for i in range(split_index, len(X_scaled)):
        if i < n_steps: continue

        window = torch.tensor(X_scaled[i - n_steps:i], dtype=torch.float32).unsqueeze(0).to(device)
        pred_pct = scaler_y.inverse_transform(model(window).cpu().numpy())[0][0]

        date_t = valid_dates[i - 1]
        price_t = data.loc[date_t, 'Close']
        pred_price = price_t * (1 + pred_pct)

        current_loc = data.index.get_loc(date_t)
        if current_loc + 1 < len(data):
            date_t_plus_1 = data.index[current_loc + 1]
            test_indices.append(date_t_plus_1)
            predictions.append(pred_price)
```



模型訓練步驟

step3.

利用「深度進化策略（Evolution Strategy）」訓練一個簡易神經網路作為交易代理人，根據 LSTM 產生的股價預測序列決定何時買賣股票；代理人在多次迭代中調整權重以最大化最終報酬，最後根據最佳策略模擬完整交易流程、計算收益與報酬率，並輸出交易結果圖與績效指標。

```python
def process_stock(ticker, save_dir, window_size=30, initial_money=10000, iterations=500):
    try:
        df = pd.read_pickle(f'{save_dir}/predictions/{ticker}_predictions.pkl')
        close = df.Prediction.values.tolist()

        model = Model(input_size=window_size, layer_size=500, output_size=3)
        agent = Agent(model, window_size, close, skip=1, initial_money=initial_money, ticker=ticker, save_dir=save_dir)
        agent.fit(iterations=iterations, checkpoint=10)
        states_buy, states_sell, total_gain, invest_pct = agent.buy()
        plot_trading_result(ticker, close, states_buy, states_sell, total_gain, invest_pct, save_dir)

        return {
            'total_gains': total_gain,
            'investment_return': invest_pct,
            'trades_buy': len(states_buy),
            'trades_sell': len(states_sell)
        }
    except Exception as e:
        print(f"Error processing {ticker}: {e}")
        return None
```

目前問題

Problem

- 因網頁目前透過國外網站佈署，無法直接取得**TWSE**之資料。
- 股價預測時長太短，不能反應市場變化。
- 股價預測誤差偏高，趨勢捕捉不夠穩定。

Solution

- 需要預先做特定資料快取儲存，並定時更新，以供佈署時能夠有資料提供顯示。
- 未來會擴展為能夠預測數個交易日，依次捕捉中期市場趨勢。
- 嘗試多任務學習，同時預測價格和方向，並引入混合模型，強化預測準確度。

未來展望

1. 擴增模型資料來源：加入社群輿情監測以強化預測準確度。
2. 手機應用支援：　發展跨平台或原生 **App**。
3. 自動化交易模組：　可接入券商 **API**，實現買賣動作自動化。
4. 社群功能強化：　在討論區增加留言的多層級回覆機制。

結論

實現一個個人化導向的股票記錄與智能分析平台。透過整合交易紀錄、視覺化與 **AI** 模型預測，幫助使用者更有依據地調整投資策略。

THANK YOU