The University of Hong Kong

COMP4801 Final Year Project

Empowering Financial Decisions with AI-Driven Forecasting and Personalized Investing

Detailed Project Plan

Lam Ngok Fung Herman (3035928328)
Wang Yujing (3035835595)
Sun Haoyu (3035856044)

Supervisor
Dr. Difan Zou
Assistant Professor of the Department of Computer Science

1 October 2024

Table of Contents

List of Figures

1 Project Background

In recent years, the prevalence of the Internet has resulted in the generation of vast amounts of data, particularly in the stock market. Investors are increasingly exposed to a multitude of information from various sources. Leading financial data websites such as Bloomberg (bloomberg.com), HKEX (hkex.com.hk), and Yahoo! Finance (finance.yahoo.com) provide real-time market data and extensive financial news, empowering users to make informed decisions regarding their portfolio. Similarly, popular stock trading platforms in Hong Kong, including WeBull, Futu, and Longbridge, feature multiple panels that display stock market statistics, trading strategies, portfolio rankings, news, forums, and bulletins. While these functionalities aim to provide users with more comprehensive information, they can overwhelm investors, especially those who are new to the market, making it difficult to navigate the platforms and extract valuable insights.

In this context, it is crucial for financial investors to be equipped to efficiently gather and analyse information from diverse sources. Unfortunately, for individuals with limited professional experience but want to predict the stock market and outperform it, the challenge of switching among different websites to identify relevant news and figures can be exceedingly frustrating and time-consuming. This issue is further exacerbated by the high costs associated with premium financial services, for instance, the annual subscription fee to Bloomberg Terminal is 27,660 USD [1].

Even if newcomers become familiar with the financial websites and invest time and money in gathering financial news and figures, summarizing and building their own opinion to make forecasts remain difficult. A foundational concept in predicting the stock market is the Random

Walk Theory [2], which states that the stock market is efficient and reflects all available news making it impossible to forecast the stock price changes in short-term. Although alternative frameworks, such as the Dow Theory, suggests that the market is predictable through technical analysis based on history data, the Random Walk Theory remains widely accepted and extensively referenced in finance literature including university textbooks. This broad acceptance reveals the complexity of accurately simulating the stock market and predicting precise price movements depending solely on historical patterns. However, if only the general trend rather than the exact share price needs to be predicted, financial news and analysis from authoritative sources can be invaluable, as they often shape public confidence in stocks and broader market trends.

Therefore, our objective is to implement an easy-to-use platform that assists novice investors in forecasting stock market trends while enhancing the user experience. While the beginners may be exhausted by the sheer volume of information available needed to discern relevant insights, the transformative power of artificial intelligence (AI) presents a promising solution. Recently, Natural Language Processing (NLP), a subfield of AI, has demonstrated significant effectiveness at drawing insightful conclusions from textual input by performing sentiment analysis and event extraction. With this technique, large amounts of text can be processed and classified into different attitudes towards the stock market movement. AI will then synthesize these assessments into comprehensive evaluations of stock performance. Furthermore, in order to make the best use of the model, a website will be developed to incorporate AI, facilitating personalized investment recommendations tailored to individual risk preferences by analysing user behaviour with paper trading, thereby empowering investors to navigate the complexities of the financial markets more effectively.

2 Project Objective

The goal of this project is to develop an AI-driven one-stop financial management application incorporated with a paper trading system and a robust NLP model for financial market forecasting, which will provide help to beginners with personalized recommendations.

The intermediate goal for this project is to collect data for the model training, extract features, and train the model. After training, the model will be tested, and its performance will be evaluated for possible future improvements. For the application, website design and style will be decided to develop a prototype. In addition, the paper trading system and risk control mechanism will be developed.

The final goal is to launch a fully functional user-friendly financial website embedding a paper trading system and a trained NLP model performing sentiment analysis to forecast future market trends and stock price movements, which will continuously analyse market data to notify user real-time market trends and investment opportunities. The personalized recommendation will be accomplished by utilizing the paper trading system to collecte user data, so that users can set their risk profiles and trading rule to receive tailored forecasts and investment strategies. Both the frontend and backend of the website will be fully developed to support all functions. User trails with feedback collected will be carried out to improve the user experience.

3 Methodology

3.1 Data Collection and Preprocessing

Our objective is to gather text data from diverse financial market sources, including financial news (CNBC), financial reports (Apple Inc.), social media information (X.com), and user comments (Yahoo et al.). This data will cover real-time stock market data, corporate financial data, news articles, financial reports, a small amount of macroeconomic data, and social media data, providing a comprehensive view of the financial landscape.

Our data collection strategy is versatile, employing different tools based on the nature of the data. For static pages, we'll utilize Scrapy and Beautiful Soup for web scraping, while dynamic data will be captured using Selenium. Additionally, we'll leverage APIs from financial platforms such as Yahoo Finance, Alpha Vantage, and CNBC to ensure we gather the most relevant and up-to-date data.

Regarding data volume, we plan to collect approximately 1.2 million valid data points to ensure that complex models can capture meaningful patterns. The data span should cover various market conditions, such as bull, bear, and volatile markets, and will span approximately 10 years to take different economic cycles into account.

The collected data will be formatted into JSON or CSV files for preprocessing. Further, the data will be cleaned by removing noise, stopping words, and normalizing the text (e.g., converting it to lowercase). TensorFlow will be used for tokenization, with particular attention paid to terms related to financial domains. Stemming and lemmatization will reduce word diversity, and low-frequency and high-frequency words will be filtered out to improve model efficiency. The text will be transformed into vector representations using

Bag-of-Words, TF-IDF, or word embedding techniques. Finally, the dataset will split into training, validation, and test sets, and, if necessary, data augmentation techniques will be applied to generate additional training samples.

3.2 NLP Architecture

To help users make informed investment decisions, we will employ NLP technology to analyse news articles and social media trends. This project will integrate multiple deep-learning models to process large amounts of textual data with time series as an integrated model. The integrated model (or cascade model) provides more accurate predictive outcomes in financial market sentiment analysis by connecting multiple deep learning models sequentially, with the output of one model serving as the input to the next, to take full advantage of the strengths of each model in different tasks [3].

In the integrated model, long short-term memory network (LSTM) will first be used to capture text-dependent dependencies. LSTM model can effectively deal with the dynamic characteristics of time series data, such as the temporal fluidity of financial news and the changing trend of social media emotions [4]. LSTM processes long-term and short-term information through its built-in "memory unit" and is particularly suited to capturing the short-term impact of financial events, such as breaking market news or policy announcements. The output of the LSTM is passed as input to the Transformer model, which captures global context information in the text through its self-attention mechanism and can identify long-term dependencies and complex semantic associations [5], such as the potential link between long-term stock fluctuations and market sentiment.

In practice, we will implement this cascade architecture using PyTorch, train the LSTM and Transformer models in series, and optimize the weights of the entire model through backpropagation.

3.3 Personalized Investment Recommendation System

After successfully training the NLP model, the following steps will be followed to integrate the model into the application to achieve personalized investment recommendations:

3.3.1 Data Input and User Preference Collection

In order to output user-specific recommendations, we will make the collection of user's investment style, risk appetite, financial goals, assets, and other relevant data. In the application's user interface design, a particular module will be set up for the user to fill in the aforementioned information, and it could also be analysed through the user's historical behaviour and transaction data.

3.3.2 Integration of the NLP Model

This stage embeds the trained NLP model into the back-end system. The model's primary function is to analyse market sentiment and related news to extract meaningful information that may affect market movements. This information will be used to generate investment recommendations.

The NLP model can communicate with the front end through an application programming interface (API) in the system architecture. When a user requests an investment advice, the front end passes user and real-time market data to the NLP model. After the calculation of overall sentiment score with the data and sentiment quantification by the NLP model, rules for investment suggestions are set according to the range of sentiment scores and the user's style data, for instance:

$sentiment > 0.6$: Buy

$0.3 < mood\ score \leq 0.6$: Hold

$sentiment\ score \leq 0.3$: Sell

3.3.3 Personalized Recommendation Algorithm

Use the model to generate portfolio recommendations appropriate for the user, combining the user's preferences and risk tolerance level. For example, if a user is a conservative investor, the model may prefer stocks or funds that are less volatile.

Potential market opportunities can also be added to recommendations based on market sentiment and news hotspots analysed by the NLP model. For example, when the model detects that sentiment in a particular sector or stock is more optimistic, it can recommend it to the user first.

3.3.4 Recommendation Result Display and User Interaction

In the application, we will design an intuitive user interface to display the recommended portfolio. As a result, the application can display the underlying reasons for the recommendation to users through charts and risk distributions to help them better understand the logic.

Moreover, users can choose whether to adopt the recommendations or fine-tune them, such as altering the proportion of certain assets to fit their risk tolerance level.

3.4 Paper Trading System

We will develop a paper trading system to enhance user experience and integrate a personalized investment recommendation system, allowing users to test investment advice in a virtual environment. This system provides users with virtual funds to execute simulated buy/sell operations based on recommendations generated by the NLP model and real-time stock market data. All transaction data will be stored in a MySQL database, with real-time data ensuring market-based decisions. A trading engine will update users' virtual holdings and account balances, and users can assess the performance of recommendations through trade analysis and adjust strategies based on risk and return rates. Additionally, users can experiment with basic automated trading rules to further validate recommendations.

The system will incorporate two core risk control mechanisms: stop-loss/take-profit and position management. The stop-loss/take-profit mechanism automatically closes trades when preset thresholds are reached, calculated using Pandas and NumPy. Position management will allocate capital for each trade based on total account funds and user risk tolerance, using methods like the fixed percentage method or Kelly criterion. The system will dynamically monitor market changes by integrating Python trading APIs (e.g., Alpaca, Interactive Brokers), helping users practice risk management and optimize investment strategies.

3.5 Front-end and Back-end Construction

3.5.1 Frontend Development

The user interface of this application displays personalized investment recommendations, the user's transaction history, and real-time stock market data. Key stock trends, forecast values, and the latest financial news headlines are presented in chart form, highlighting the most influential positive or negative news from sentiment analysis. The interface will be developed using React, which, with its powerful component-based structure, provides a clean and smooth user experience.

3.5.2 Backend Development

The backend will be built using the Django framework, responsible for handling frontend requests and managing backend tasks. Django utilizes the Django Rest Framework (DRF) to implement REST APIs, facilitating communication with the frontend. The server is primarily responsible for retrieving data from the MySQL database, performing calculations, and returning personalized recommendation results. MySQL, as a relational database, is used to store key data, including user transaction records, market data, and NLP model inputs and outputs. With the tight integration of Django and MySQL, the system shall ensure data processing stability and performance.

4 Project Schedule

<table>
<tr><th colspan="2">Schedule</th><th colspan="2">Milestones
(predicted number of learning hours)</th></tr>
<tr><td colspan="4">Phase 1: Project Inception</td></tr>
<tr><td>2024</td><td>Sep</td><td>• Preparation of Project Plan (10)
• Literature Review (10)
• Setup of Project Website (10)</td><td><u>Deliverables:</u>
1. Detailed Project Plan
2. Project Website Setup</td></tr>
<tr><td colspan="4">Phase 2: Project Elaboration</td></tr>
<tr><td rowspan="3">2024</td><td>Oct</td><td colspan="2">• Training data source selection (10)
• NLP data collection (15)
• NLP data pre-processing and extraction (20)
• NLP model training (20)</td></tr>
<tr><td>Nov</td><td colspan="2">• NLP model training (20)
• Model testing and evaluation and possible further tuning (15)
• NLP model integration (10)</td></tr>
<tr><td>Dec</td><td colspan="2">• NLP model integration (10)
• NLP model performance evaluation and reporting （10）
• Front-end basic web design, style selection （15）
• Set user paper trading system and risk control mechanism (10)
• Interim Report Drafting (20)</td></tr>
<tr><td>2025</td><td>Jan</td><td>• Interim Report Drafting (20)
• NLP model performance evaluation（10）
• User personalized recommendation system (10)</td><td><u>Deliverables:</u>
1. Trained Model Prototype
2. Interim Report
3. First Presentation</td></tr>
<tr><td colspan="4">Phase 3: Project Construction</td></tr>
<tr><td rowspan="3">2025</td><td>Feb</td><td colspan="2">• Frontend and Backend Construction (20)
• Construction of the user-definable trading rule setting page (10)
• User personalized recommendation system (25)
1. User Data collection
2. Recommendations generation
3. Optimization and Updating
• Trading system performance evaluation and optimization （20）
• Final Report Drafting (10)</td></tr>
<tr><td>Mar</td><td colspan="2">• Final Report Drafting (40)
• Frontend and Backend Construction (20)
• Debugging and Testing (15)
• Possible User trial, and collection of feedback (15)</td></tr>
<tr><td>Apr</td><td colspan="2"><u>Deliverables:</u></td></tr>
</table>

		1. Project Exhibition – 3-min Video 2. Final Report 3. Final Application/ Product 4. Final Presentation

Figure 1: Details of the project schedule

Reference

[1] "Bloomberg vs. capital IQ (capiq) vs. Factset vs. Refinitiv," Wall Street Prep, https://www.wallstreetprep.com/knowledge/bloomberg-vs-capital-iq-vs-factset-vs-thomson-reuters-eikon/ (accessed Sep. 26, 2024).

[2] B. G. Malkiel, A Random Walk down Wall Street. New York: Norton, 1973.

[3] K. Cao, T. Zhang, and J. Huang, "Advanced hybrid LSTM-transformer architecture for real-time multi-task prediction in engineering systems," Scientific Reports, vol. 14, no. 1, Feb. 2024, doi: 10.1038/s41598-024-55483-x.

[4] K. Greff, R. K. Srivastava, J. Koutnik, B. R. Steunebrink, and J. Schmidhuber, "LSTM: a Search space Odyssey," IEEE Transactions on Neural Networks and Learning Systems, vol. 28, no. 10, pp. 2222–2232, Jul. 2016, doi: 10.1109/tnnls.2016.2582924.

[5] S. M. Jain, Introduction to Transformers for NLP. 2022. doi: 10.1007/978-1-4842-8844-3.