# Report for portfolio optimization and forecasting

[1.Introduction 2](#_Toc183594345)

[2. Data Source 2](#_Toc183594346)

[(1)Ethics and Privacy 2](#_Toc183594347)

[(2)Integrity 2](#_Toc183594348)

[(3)Validity 2](#_Toc183594349)

[(1)Data Dates 2](#_Toc183594350)

[(2)Nature of Data 2](#_Toc183594351)

[(3)Data Structure 2](#_Toc183594352)

[4. Data Deficiencies 3](#_Toc183594353)

[5.Analytics approaches 3](#_Toc183594354)

[**Step 1**: Select the Top 20 Stocks by Market Capitalization 3](#_Toc183594355)

[**Step 2**: Constructing the Portfolio 5](#_Toc183594356)

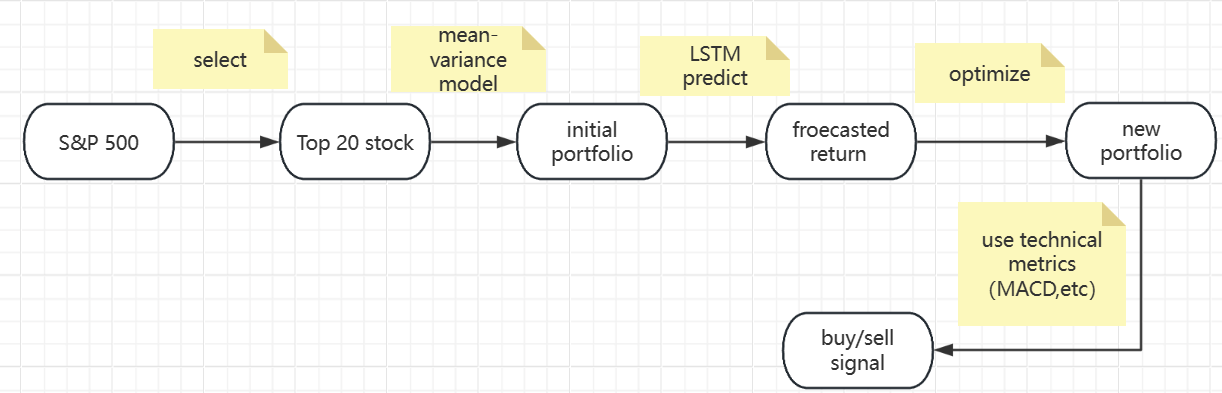
[**Step 3**: Forecasting with LSTM and use predicted prize optimize portfolio 7](#_Toc183594357)

[**Step 4**: Identifying Trading Opportunities 12](#_Toc183594358)

[5. Conclusion 13](#_Toc183594359)

[6. Personal Reflection 14](#_Toc183594360)

[7. Reference 15](#_Toc183594361)

1. ****

# 1.Introduction

Financial markets are complex, with non-linear relationships and significant noise, making analysis challenging. This project builds a portfolio of the top 20 stocks by market capitalization, using machine learning to optimize predictions and identify buy/sell points. Data from Yahoo Finance and Wikipedia was used to filter these stocks, and the mean-variance model (Markowitz, 1952) was employed to construct the portfolio, balancing risk and return. Future stock prices were forecasted using an LSTM model (Hochreiter & Schmidhuber, 1997) , suitable for capturing time dependencies in financial data, followed by portfolio optimization to maximize returns through identified trading opportunities.

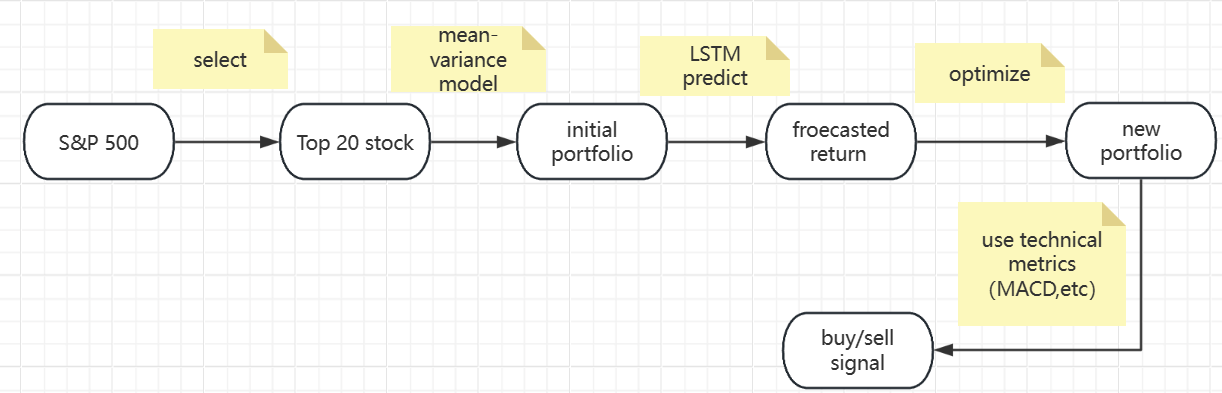


Fig 1. Flow chart of this project

# 2. Data Source

The data is sourced from Yahoo Finance (using the yfinance Python library) and Wikipedia.

(1)Ethics and Privacy: The data is publicly available, related to market information, and does not contain Personally Identifiable Information (PII). It is used responsibly for research purposes only to create optimized portfolios. No privacy concerns or anonymization are required.

Reliability: Yahoo Finance data is from official exchanges and financial providers, generally reliable for major stocks and indices. The analysis uses historical data, so delays are not an issue.

(2)Integrity: Missing values in the historical data are handled by dropping NaN entries during data cleaning.

(3)Validity: Financial data is affected by market volatility, which impacts forecast validity. Machine learning is used for time series forecasting instead of traditional methods.

**3.** Data Characteristics

(1)Data Dates: The data spans from 1 January 2013 to 31 December 2023, covering different market conditions. Five years of data is used for neural network forecasting.

(2)Nature of Data: The data is time series and consists of adjusted closing prices for multiple stocks. Daily returns are calculated as input features for optimization and forecasting.

(3)Data Structure: The data is stored in a pandas DataFrame with stock symbols as columns and dates as rows.

# 4. Data Deficiencies

Missing values are handled using forward and backward filling methods, and removing rows/columns with all null values. Stocks with less than 1000 trading days are filtered out. However, These processes may cause problems due to market instability.

# 5.Analytics approaches

## **Step 1**: Select the Top 20 Stocks by Market Capitalization

The process involved selecting the top 20 companies by market capitalization from the S&P 500 component stocks. Specifically, the selection criteria included:

Market Capitalization: Each company's market capitalization was calculated to determine the largest firms by total value.

Data Collection: Historical stock data for each company was downloaded from Yahoo Finance for the period between January 1, 2013, and December 31, 2023. A retry mechanism was implemented to handle data retrieval issues, such as "Too Many Requests" errors.

Minimum Data Requirement: Companies with insufficient historical data (fewer than 1000 records) were excluded to ensure data quality.

Top 20 Stocks by Market Cap: After calculating the market capitalization, the top 20 companies were selected for visualization.

This step relied on quantitative analysis to identify companies with the highest total value in the market, reflecting investor confidence and the overall size of these firms. The visualization showed these companies' market capitalizations, highlighting the largest players in the S&P 500.

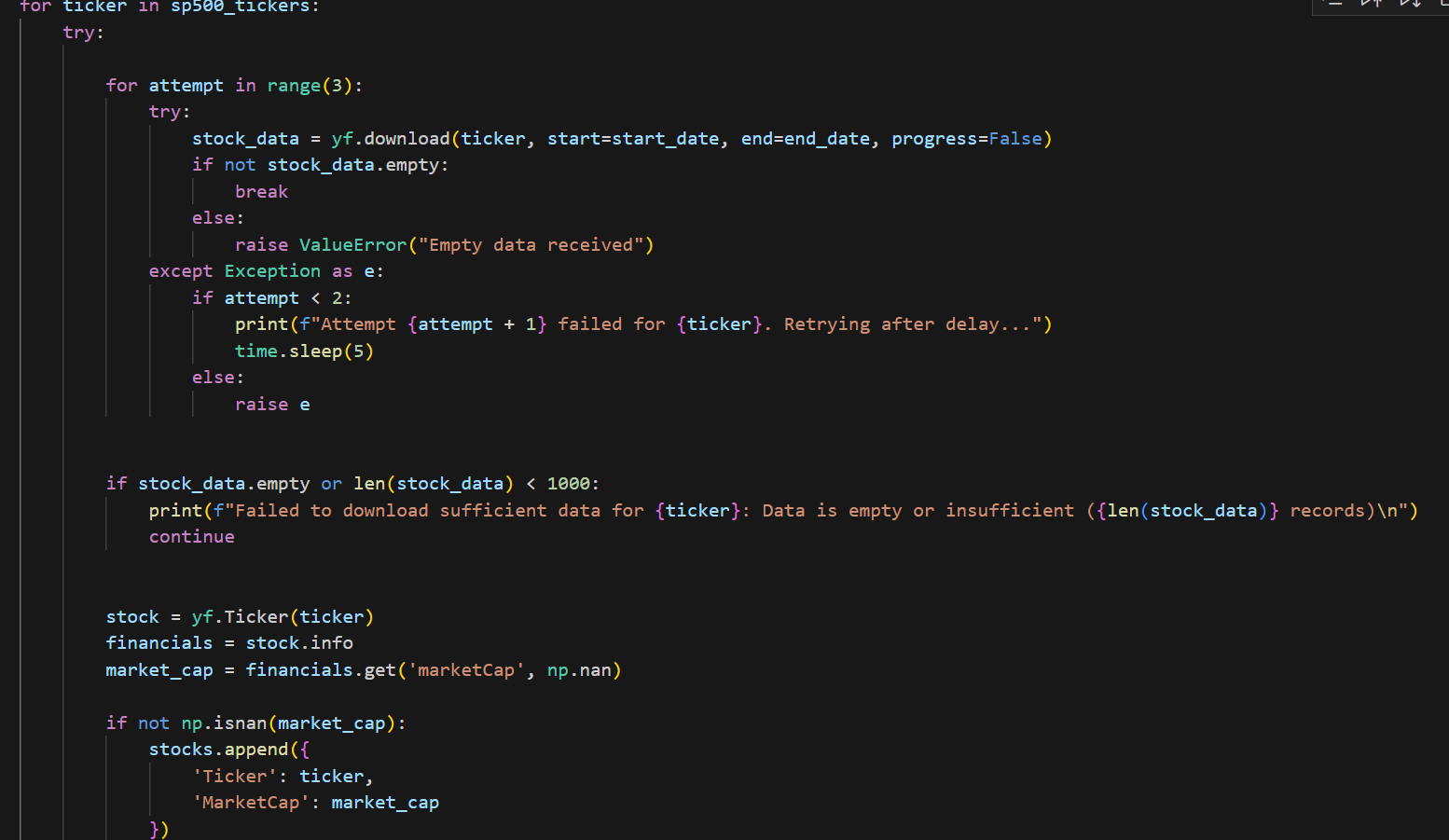


Fig 1. The main part of select Top 20 Stock

Libraries /tools used

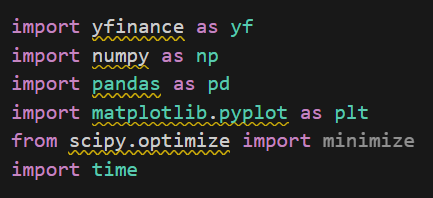


Fig 2. Python library used in this step

Visualization

The under visualization showed the top 20 Stock. It also shows the stock market capitalization in descending order to provide a pool of stocks for later portfolio building using Markowitz's model(mean-variance).

图表, 直方图

描述已自动生成

Fig 3. Top 20 Stock by market capitalization

## **Step 2**: Constructing the Portfolio

A portfolio of selected stocks was constructed using a simple mean-variance optimization technique to allocate weights to different stocks. The use of optimization (such as minimizing variance while maximizing return) ensures a balanced risk-reward ratio. By selecting weights that maximize Sharpe Ratio or minimize risk, the constructed portfolio aims to achieve optimal performance.

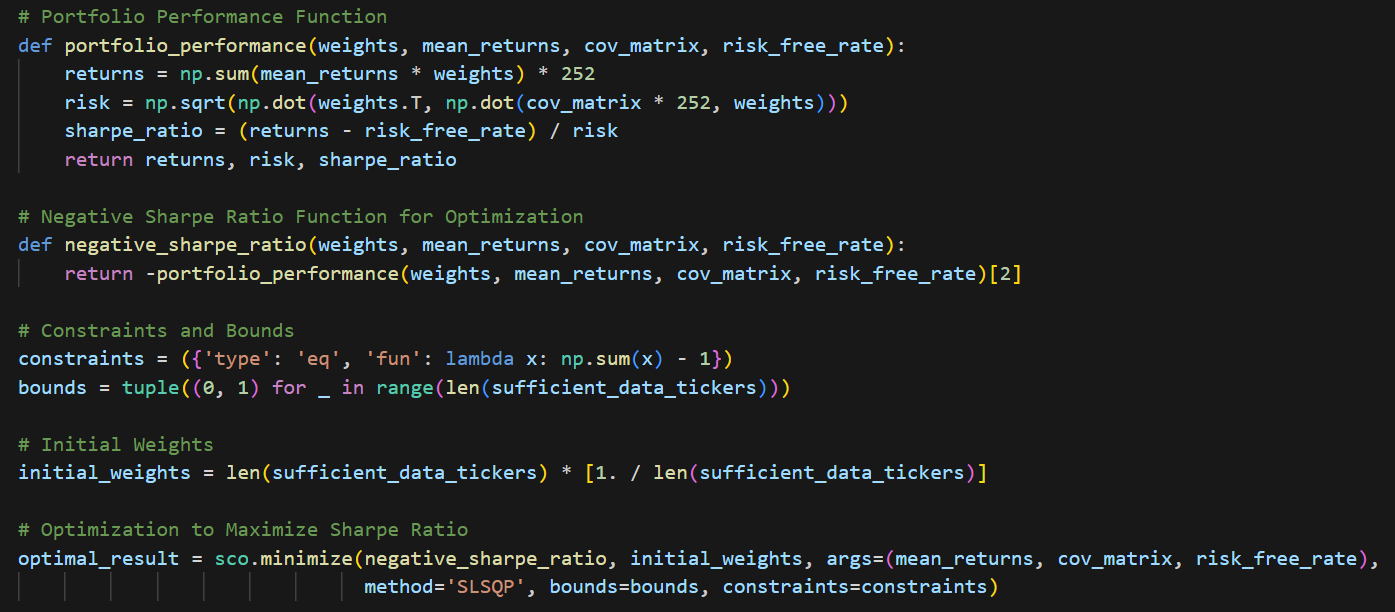
Stock Selection and Data Collection: A list of top 20 stock tickers was provide. The historical price data for these stocks was downloaded using the Yahoo Finance API (yfinance). The data was collected for the period from January 2013 to December 2023, focusing on the adjusted closing prices.

Data Cleaning and Filtering: Stocks with insufficient trading data (less than 1,000 trading days) were filtered out to ensure the reliability of the dataset. This filtering step ensured that only those stocks with significant trading history were included in the portfolio.

Mean and Covariance Calculation: This step calculates the mean return and covariance matrix of a stock after screening its daily returns. It is then input into the mean-variance model for use.

Mean-variance optimisation: The mean-variance method is used to optimise the portfolio to maximise the Sharpe ratio (Sharpe , 1994). The Sharpe ratio is a measure of risk-adjusted returns and is calculated by dividing the difference between the portfolio return and the risk-free rate (set at 2% in this project) by the volatility of the portfolio. Later, to find the optimal portfolio weights that maximise the Sharpe ratio, the Sequential Least Squares Programming (SLSQP) method is used.

Portfolio Allocation: Using the optimized weights, stocks with a small percentage are filtered out. The final portfolio obtained is represented as a pie chart with the weights allocated to each stock. The goal of this method is to achieve a balanced risk-return ratio by investing in a portfolio of stocks with the best trade-off between expected return and risk.

The use of mean-variance optimization techniques, such as maximizing the Sharpe ratio or minimizing risk, ensures a balanced risk-return profile. With the objective of maximizing returns at a given level of risk, investments are allocated among the top 20 stocks by market capitalization, resulting in an optimal portfolio.Fig 5.The main part portfolio allocation coding

Libraries /tools used

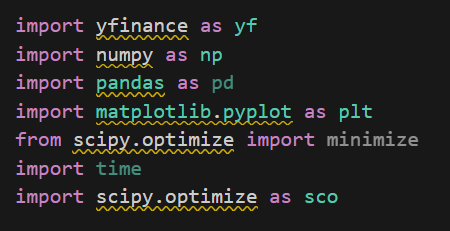


Fig 6. Python library used in this step

Visualization

This visualization showed the allocation of the portfolio of top 20 stock that is selected by market capitalization .

图表, 饼图

描述已自动生成

Fig 7.Optimal Portfolio Weight Allocation

## **Step 3**: Forecasting with LSTM and use predicted prize optimize portfolio

The LSTM model is chosen because it is suitable for time series data, which is crucial in financial forecasting’s has the advantage of processing on long-term dependencies data (Yong,2019) and therefore can effectively capture market trends and volatility’s is better than traditional machine learning models in handling continuous data and better than other time series models (ARIMA) can handle non-linear models better, which is why LSTM is widely used in financial forecasting.

Analyses conducted

The analysis involves collecting historical stock data, preprocessing it, training the LSTM model and using it for forecasting. The portfolio is then optimised based on the predicted returns. The following are the key components of the analysis:

Data Acquisition and Preprocessing: Historical stock data from and for the six stocks in the portfolio in the previous step, from 1 January 2013 to 20 November 2024, was collected using yfinance. Missing values were processed using forward and backward padding. The data was then scaled using MinMaxScaler to normalise the values and thus stabilise the LSTM learning process.

Data Conversion: The scaled data is converted into a sequence suitable for LSTM input with a 50-day lookback window. This window size was tuned to capture enough temporal information to make accurate predictions while maintaining computational efficiency (to calculate the predictions for the six stocks individually, a window of 100 would be too long and would affect the efficiency of the actual use).

Model Architecture and Training: The LSTM model used has two layers, each with 128 units. To prevent overfitting, a culling layer with a ratio of 0.2 was added to the model. The model was compiled using the Adam optimizer and trained with a 20% validation ratio. To avoid overfitting and ensure efficient learning, callbacks such as ReduceLROnPlateau and EarlyStopping were used.

Evaluation and prediction: The model was evaluated using Mean Squared Error (MSE), Mean Absolute Error (MAE) and R-squared (R²). If the evaluation metrics are satisfactory (MSE < 10,000, MAE < 50, R² > 0.7), the model is used to forecast stock prices for 2024. Each forecast is generated iteratively using the last 50 days of training data as input.

Portfolio optimisation: Expected returns are calculated based on the predicted prices and portfolio optimisation is performed using scipy.optimize.minimize to maximise the Sharpe ratio. In order to even out the risk of the portfolio to prevent the possibility of having only a single stock, the boundaries of each stock weight are set between 0.05 and 0.4.

Libraries/ Used

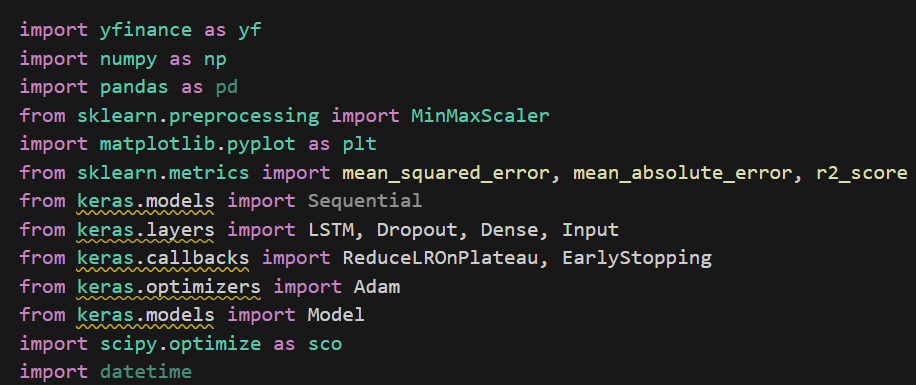


Fig 8 . Python library used in this step

Answering Research Questions Using the Models

The LSTM model was used to predict future stock prices, and the predicted prices were then used to estimate expected returns. This allowed us to answer questions related to the expected performance of each stock. The optimization step provided insights into how the portfolio could be adjusted to maximize returns while minimizing risk, thereby addressing questions related to optimal investment strategies.

Assumptions and Dataset Modifications

Assumptions: It was assumed that the historical price data could adequately represent future market conditions, which is common in financial modeling. Additionally, a look-back period of 50 days was assumed to be sufficient to capture relevant trends without overfitting.

Dataset Modifications: The data was normalized using MinMaxScaler to ensure that all features were on a comparable scale, aiding in the stability of LSTM training. Missing values were handled by forward and backward filling to ensure continuity in the time series, which is crucial for LSTM models.

文本

描述已自动生成

Fig 9.The main part of LSTM forecasting

Visualization

This visualization, Shows a comparison between the predicted (Orange Line) and actual performance (Blue Line) of each stock in the portfolio that I have selected and optimized for mean-variance over the course of 24 years, with indices such as R-squared MAE and the images showing that the predictions are better allowing for the optimization of portfolio and the use of predictive models for post-November '24 trading opportunities.

图形用户界面, 图表

描述已自动生成

Fig 10. Price forecasts for individual stocks in the original portfolio

After the forecasting of LSTM Based on the projected future returns, we optimized the portfolio again, using the same mean-variance model with the following weightings for each stock in the portfolio. This visualization, showing the optimized portfolio weight shares

图表, 饼图

描述已自动生成

Fig 11. Optimal portfolio weight allocation

In order to prove that the optimized portfolio (Orange Bar) has significant advantages over the first portfolio (Blue Bar) based on mean-variance, I measured the Sharpe ratio return and volatility of this portfolio, and the data shows that the optimized portfolio based on LSTM prediction has higher return and Sharpe ratio, and less volatility, and the visualization shows the comparison between the two.

图表, 条形图

描述已自动生成

Fig 12. Comparison of portfolio in step2 and step3

## **Step 4**: Identifying Trading Opportunities

The analysis used Moving Average Convergence Divergence (MACD) and Simple Moving Averages (SMAs) to identify price trends and generate buy/sell signals. MACD effectively highlights momentum shifts, while SMAs smooth out price data to reveal trends. These techniques are well-supported in financial literature.

A smoothed price was calculated using an exponential moving average (EMA) to reduce noise. SMAs were then computed over different periods (15, 30, 60, 100) to identify crossover points like the Golden Cross. MACD was calculated as the difference between 20-period and 50-period EMAs, with a 9-period Signal Line. Buy/sell signals were based on MACD/Signal Line crossovers.

Libraries/Tools Used

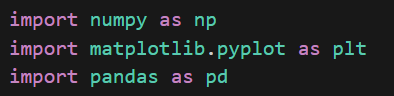


Fig 13 . Python library used in this step

The model was primarily used to answer the following questions:

When to Buy?: By identifying golden cross points using SMAs, buy signals were generated whenever shorter-term averages crossed above longer-term averages.

When to Sell?: RSI values were used to determine overbought conditions, generating sell signals when RSI exceeded 70, indicating high price momentum and potential reversal.

The predictive model was applied to the 2024 price data to generate both SMA crossovers and RSI indicators. These signals were then used to recommend potential buy and sell opportunities within the year.

Assumptions and Modifications

It was assumed that historical price trends and the relationships captured through SMAs would continue to be indicative of future price movements.

Market efficiency was partially relaxed, assuming that technical indicators could identify short-term opportunities not immediately reflected in the price.

The dataset was filtered to focus on companies with high capitalization, assuming their future performance would yield substantial returns. Outliers and missing data points were removed to ensure model robustness and avoid undue influence on results.

Visualization

The “Buy Signal” (green triangle) and “Sell Signal” (red triangle) are marked on the chart to help investors recognize when it might be a good time to buy or sell.

The blue line, which represents the smoothed projected price action, can help to see the overall trend of the portfolio's stock price in 2024.

The MACD line (purple dashed line) and the signal line (orange dashed line) are used to assist in analyzing changes in price action in order to identify trend reversals.

图表, 直方图

描述已自动生成

Fig 14. buy and sell signals

# 5. Conclusion

In general, the findings demonstrated that combining traditional financial models (mean-variance) with machine learning techniques (LSTM) effectively achieved the aim of maximizing returns while minimizing risks. The constructed portfolio was optimized to handle different market conditions, and the forecasting and trading signals provided actionable insights for decision-making.

However, there are several limitations on my project:

Missing Data: Using forward or backward filling assumes past or future values are accurate estimates, which can be problematic in volatile markets. This approach risks introducing biases, making forecasts less reliable. For example, during significant market events like the 2020 pandemic, using simple imputation may lead to incorrect trend estimations. Exploring other methods, such as using LSTM to predict and fill missing values, could provide better accuracy.

Model Complexity and Overfitting: LSTM's complexity increases the risk of overfitting, particularly with noisy financial data. Although dropout layers were used, high training accuracy does not always translate well to real-world performance, especially under volatile market conditions. For example, when applying the model during a sudden market downturn, overfitting to stable historical trends may fail to adapt adequately.

Mean-Variance Optimization Limitations: Mean-variance assumes normally distributed returns and stable covariance, which rarely hold in financial markets that often exhibit fat tails and volatility clustering. During extreme market conditions, such as 2008's financial crisis, the optimized portfolio might not protect well against significant losses due to these flawed assumptions.

Technical Indicators and Market Efficiency: SMA and MACD are based on historical patterns, conflicting with the efficient market hypothesis, which claims all information is already priced in. For example, using MACD signals may not consistently lead to profits in highly efficient markets where new information is instantly reflected in stock prices.

Bias from Assumptions: Assumptions like a 50-day look-back period and stable market conditions influenced results, limiting their applicability. For instance, a different look-back period might yield vastly different outcomes, affecting the reliability of portfolio optimization decisions during different market phases.

And when managers or retail investors in the use of the above problems may lead to poor prediction ability, affecting the specific use. And because the above needs to be experimented with frequently based on updated data, it is not applicable to the general public.

# 6. Personal Reflection

In this project, I tried for the first time to use LSTM models for predictive portfolio optimization. Also encountered many difficulties in this process:

One major challenge I faced was handling the significant amount of missing data when using historical stock prices. I tried forward and backward filling methods, but I soon realized that these approaches were not suitable during high market volatility, as they led to biased forecasts. To solve this, I experimented with using LSTM-based imputation, which provided better accuracy and more realistic data patterns. This experience taught me the importance of adapting data handling techniques to the specific characteristics of financial data. Additionally, during the LSTM model training process, I faced issues with overfitting, particularly with noisy data. To address this, I added dropout layers and experimented with different learning rates. Although it was a time-consuming process, this iterative tuning allowed me to achieve better generalization.

In this project, I initially tried using LSTM, and as the results were not accurate I added Convolutional Neural Networks (CNN) in the hope of improving the prediction accuracy. However, due to the increased complexity of the model and the risk of overfitting, I decided to simplify the model and focus on optimizing the LSTM by modifying the number of layers and the learning rate. This taught me the value of balancing model complexity and utility, and that interpretability has to be considered paired, out of realistic usage scenarios. In future projects, I plan to explore other portfolio optimization methods such as Support Vector Regression (SVR) and consider incorporating bonds or futures to diversify investment options.

# 7. Reference

1. Markowitz, H. M., & Markowitz, H. M. (1967). Portfolio selection: efficient diversification of investments. J. Wiley.
2. Hochreiter, S. (1997). Long Short-term Memory. Neural Computation MIT-Press.
3. Sharpe, W. F. (1994). The sharpe ratio. Journal of portfolio management, 21(1), 49-58.
4. Yong Yu, Xiaosheng Si, Changhua Hu, Jianxun Zhang; A Review of Recurrent Neural Networks: LSTM Cells and Network Architectures. *Neural Comput* 2019; 31 (7): 1235–1270. doi: <https://doi.org/10.1162/neco_a_01199>

Code in GitHub：

[portfoliooptimization/portfoliooptimization.ipynb at main · YiyunXia926/portfoliooptimization (github.com)](https://github.com/YiyunXia926/portfoliooptimization/blob/main/portfoliooptimization.ipynb)