RO1

FYP Final Report

Evaluating the Performance of Machine Learning Based Portfolio Management in the Cryptocurrency and U.S Stock Market

by

CHO Hangsun, YANG Wenting, TAM Ching Lung, Ferdy

RO1

Advised by

Prof. David Rossiter

Submitted in partial fulfillment

of the requirements for COMP 4981

in the

Department of Computer Science

The Hong Kong University of Science and Technology

2022-2023

Date of submission: 2023-04-20

Abstract

Nowadays more and more investors and traders are utilizing machine learning when generating trading ideas. This project aims to create a portfolio management system on the stocks listed in the United State by incorporating different machine learning models via ensemble learning. In total, we implemented four different strategies using machine learning which are: Multi-factor stock selection model using LSTM and GRU, stock selection using Natural Language Processing models. Then, risk mitigation was done by calculating similarity scores and matrix using machine learning for the selected stocks. The best model of Multi-factor stock selection is GRU model trained with both technical and fundamental data, which can achieve annualized return 24.2% and max drawdown 17.6%. The stock selection using Natural Language Process model able to filter out stocks with negative sentiment on its 10-K report, creating a more effective trade by holding the only necessary stocks and overall enhancing the strategy's return per trade and minimize the Max Drawdown when compared to a passive strategy that involves buying and holding the S&P 500 index. The stock selection using social media sentiment was able to extract sentiment and predict stocks and cryptocurrency prices using XGBoost, providing more diverse data sources and assets. The risk mitigation model removed similar stocks and created a new portfolio, which had 3 percentage points less in maximum drawdown and 8 percentage points less in return compared to the original portfolio. In the end the combined model had higher yearly return than that of the SPY, and outperformed SPY’s maximum drawdown by 7.3 percentage points.

Table of Contents

1. Introduction

1.1. Overview

Ever since the emergence of the first financial institutions, financial markets have continued to evolve. Nowadays millions of stakeholders participate in financial markets to conduct transactions. Widely ranging from basic products such as stocks and commodities, to exotic derivatives, multitudes of financial products are being traded in financial markets. People have always tried to profit from trading in the financial markets, and their strategies have continued to evolve. Before the 4th industrial revolution, traders mostly focused on fundamentals. Nowadays, traders depend highly on computers and algorithmic trading, which allows them to use pre-programmed rules with parameters, such as time, price, and volume, to automate the trading process. Traders can exploit the reliability, robustness, and speed of computer programs when executing orders. Although computer algorithms can reduce the tendency to make irrational decisions and human mistakes, they can also lead to losses during unexpected market crashes.

Different algorithmic trading strategies rely on different types of data and various technologies. Arbitrage, one of the widely known strategies, takes advantage of price differences of similar assets in different markets. Different products following the same basket of securities, or even the same products, can have different values in different markets due to market inefficiencies. In this case, the profit is the market price difference. Another notable strategy is named the momentum strategy. This strategy assumes that the price of a financial product tends to follow past trends.

Machine learning can be used for the prediction of unknown data, after a model is trained with training data. It has a great variety of fields of applications, like speech recognition and food classification. Some models can even perform predictions better than humans, given their ability to process vast amounts of data. With such ability, they are also used for performing financial predictions in order to trade according to expected asset values so as to make money.

However, there is still no foolproof model for trading under volatile market conditions and macroeconomic factors, such as the COVID-19 outbreak, the Russia-Ukraine War, and high inflation rates. This is because trading strategies perform totally differently under different market conditions.

Therefore, we are developing a stock portfolio management system that uses four different strategies mainly on stocks and integrates them together to seek a consistent return with risk as low as possible even under volatile market conditions. The four different strategies that we are focusing on are the multi-factor equity model, NLP-based strategy which analyzes the SEC reports, pairs trading strategy on baskets of stocks using clustering, and rank-based sentiment analysis on the internet. These four strategies are being integrated together using Harry Markowitz's modern portfolio theory in order to minimize the risk. We are also building a platform that traders can use with the combined strategies.

1.2. Objectives

The goal of this project is to create a portfolio management system that hedges against market risks and potentially gains profit through prediction using machine learning models and portfolio management using portfolio optimization methods. To improve upon previous projects, we are working on some newer trading strategies, using portfolio optimization and ensemble learning with more comprehensive testing.

To achieve this goal, we are working on the following objectives:

1. Data Collection: We develop software to automatically retrieve relevant price and textual data.

2. Data Preprocessing: We develop several models to clean the data and extract relevant indicators from the data storage.

3. Financial Prediction: We develop several models to predict the asset expected returns and merge them through ensemble learning.

4. Risk Quantification: We develop programs to estimate the asset expected risks through risk models.

5. Portfolio Optimization: We develop software to generate an optimal portfolio for multiple assets given their expected returns and risks.

6. Backtesting: We test and analyze the portfolio management system using historical stock price data and compare the results with benchmarks.

7. User Interface: We develop a web-based platform for users to select stocks, strategies, and time periods and visualize the backtesting results

We believe combining and determining optimal weighting for each strategy will be a massive challenge. Due to their unique data collection method, designing an optimal parameter grid-searching algorithm will require heavy computing resources, which slows down the development process. Consequently, we will allocate a suitable amount of time for the grid-searching process to anticipate whether a manual trial and error parameter searching is required.

1.3. Literature Survey

1.3.1. Multi-factor Stock Selection Model with Machine Learning

Multi-factor models are one of the most important active investments in quantitative asset management, which explain returns on individual stocks with different factors. There are two types of Multi-factor models: Arbitrage Pricing Theory (APT) models, which focus on the linear relationship between the expected returns on individual stocks and macroeconomic variables [1], and Fama-French type models, which use companies' fundamental factors to do the return prediction [2]. The report focuses on the latter one. The multi-factor model in this report is defined in formula (1).

$$R_{it} = \sum_{n=1}^{k} X_{int} f_{nt} + \epsilon_{it} \text{ — (1)}$$

in which R_{it} means company i's equity return in t period, X_{int} is risk exposure of factor n for company i in t period, f_{nt} is factor n return in t period, and ϵ_{it} is the special return.

Compared to using traditional statistical methods to explain Multi-factors' influence on stocks' expected return, machine learning methods, which usually have advantages in analyzing high-dimensional non-linear data, have been proven to be more effective, especially under the current big data era [3]. Previous machine learning methods used in the multi-factor model include Support Vector Machine (SVM), Neural Networks (NN), and Gradient Boosting Decision Tree (GBDT) [4].

However, recent research demonstrates that in terms of performance in stock prediction tasks, deep learning methods like LSTM are better than SVM, and random forest[5]. At the same time, the prediction capability of a Gated Recurrent Unit (GRU) is better than that of an LSTM [6]. [7] proposes a multi-factor stock selection using GRU. This method performed quite well, achieving a 13.13% excessive return on the China Securities index CSI 300 with an MDD of 17.38%. However, the paper mainly focuses on the Chinese stock market. Besides, according to our research, seldom research has been done in multi-factor models using LSTM or GRU in the U.S. stock market.

1.3.2. Price Prediction by Rank-based Sentiment Analysis on the Internet

The efficient market hypothesis states that asset prices reflect all available information. So, the more information we have, the better we can predict market prices.

Social media is highly popular nowadays for the discussion of cryptocurrencies like Bitcoin and Ethereum. Many researchers and professional traders have started using social media post sentiment for predicting cryptocurrency prices. [8] combines Reddit post sentiment and its importance to model cryptocurrency prices. One limitation of this project is that it was only tested on cryptocurrencies. A previous HKUST student project aimed to find the most trending cryptocurrency symbols on Reddit based on popularity and then analyze the sentiment of these cryptocurrencies and trade them [9]. However, it is limited to only the top few symbols. News and Google Trends have also been found to have a great correlation with cryptocurrency prices [10]. However, one limitation of this project was that it was only tested on cryptocurrencies.

Therefore, our system is also using internet sentiment analysis as one component of our ensemble learning.

1.3.3. Natural Language Processing (NLP)

NLP has been proven to be able to parse the complexities of text hundreds of times faster than humans [11]. It is not only limited to some general topics but can be extended to help humans analyze business and financial news, as well as complicated industry jargon, numbers, currencies, and product names that require prior domain knowledge. Based on a study of forty-five research papers [12], in general, financial analysts' predictions usually include an optimistic bias, such as insisting on a higher target price, resulting in higher predicted prices. Using NLP can help to eliminate this optimistic bias because we can extract and ignore the optimistic bias from the report.

1.3.4. Risk Mitigation Using Different Similarity Indicators

There are several conventional ways to find out stocks that would perform similarly in the future. However no single approach is foolproof, and each has its strengths and weaknesses. [13] proposes a new similarity measure called DMPSM (Dynamic Multi-Perspective Similarity Measurement) to describe the similarity between a pair of time series. It weights and scales the given stock series, and then Canberra distance is embedded into the DTW (Dynamic Time Warping) to measure the similarity. This way, the DMPSM can reflect the personalization of stock time sries and apply to one-to-many matching by eliminating the impact of singularities. The study has validated the efficiency of DMPSM through experiments using 285 stocks from the Shanghai Stock Exchange, and concluded that it outperformed other similarity measures including Euclidean distance, Canberra distance, and DTW. However, one limitation from the noted study is that all three similarity measures used the technical data of stocks from the Shanghai Stock Exchange. In order to incorporate different measures and methods, it would be reasonable to use the fundamental data of the stocks too when calculating the similarity.

Therefore, our study will aim to use both technical data and fundamental data to calculate the similarity between stocks. Plus, rather than using one single similarity, the different similarity scores will be integrated to return a single similarity score.

1.3.5. Portfolio Management using Machine Learning

A previous FYP proposal [14] combined several trading strategies to beat the S&P 500 Index. Compared with the market's annual return of 9.85% and Maximum Drawdown (MDD) of 55.1%, their machine learning algorithm beat the market index benchmark by 0.14% with only 7.8% of MDD. Our system will use a similar methodology but with different machine learning methods and try to achieve higher annualized return and lower MDD while using a single asset class: stock.

2. Methodology

2.1. Design

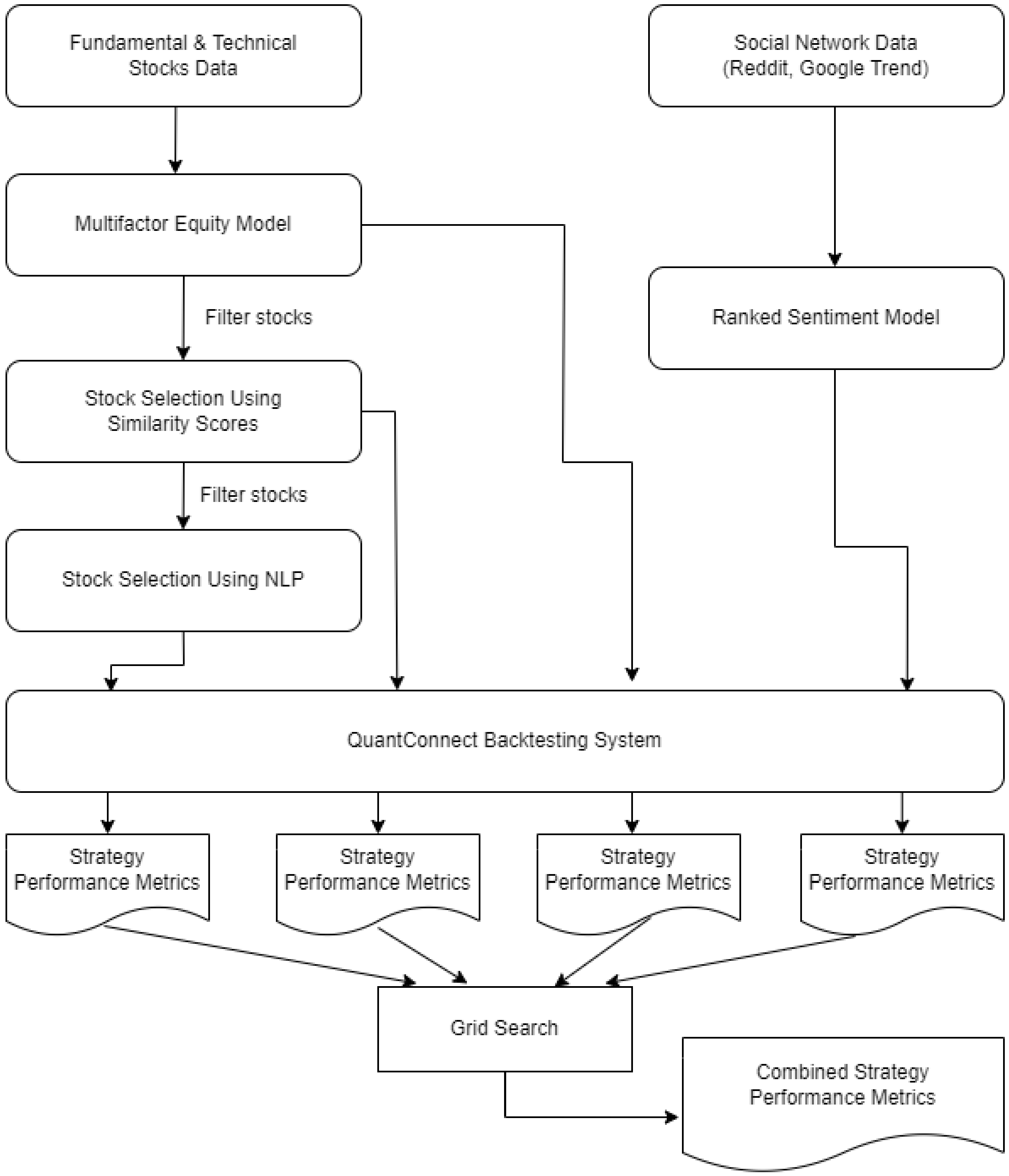

Fundamental & Technical Stocks Data
Social Network Data (Reddit, Google Trend)
Multifactor Equity Model
Ranked Sentiment Model
Filter stocks
Stock Selection Using Similarity Scores
Filter stocks
Stock Selection Using NLP
QuantConnect Backtesting System
Strategy Performance Metrics
Strategy Performance Metrics
Strategy Performance Metrics
Strategy Performance Metrics
Grid Search
Combined Strategy Performance Metrics

Figure 1: Overall design of the project

2.1.1. Multi-factor Equity Model

For data collection, there are two kinds of data: price and company fundamental data. We collect the price and fundamental data from third-party platforms by using their APIs.

During the data preprocessing, we derive the technical data using historical stock prices and we build datasets with both technical and fundamental data. We split the dataset into a training set and a test set. For the training set, we pick the window of 1993-2020. For the test set, the window slide is 2020-2022. We tackle the NA data by replacing them with 0. We give the labels to the stocks in S&P 500 based on their next month's excessive return.

Then, we feed our preprocessed data to the machine learning models which are Long short-term memory (LSTM) and Gated Recurrent Unit (GRU) to do the predictions. The input should be stocks' several months' features including fundamental data and technical data. The output should be the next months' rising possibility of 500 stocks. Based on the prediction, we sort the possibility and select the top 30 stocks. We use different techniques to generate better predictions. We construct our trading strategies and use the quantconnect to evaluate the models. We pick the best one and give this model to the following risk management processes.

2.1.2. Price Prediction by Rank-based Sentiment Analysis on the Internet

As discussed in the literature survey, Reddit, Twitter, and Google Trend data are quite useful for sentiment and price prediction. So to analyze them for price prediction, here are the steps:

1. Collect data from social media Reddit and Twitter, Google Trend, and YFinance

2. Analyze text rank using its author rank, sentiment, and date

3. Train model to predict price using the sentiment, prices, and indicators

2.1.3. NLP Sentiment Analysis - Integrating Sentiment Analysis and Textual Similarity

Imagine we have 2 movie reviewers reviewing the same movie. Although they could use different wordings, if both of them liked the movie, then the sentiment of the meaning will be similar. Using a text similarity approach will result in low similarity for both reviews, therefore, it would be better not to use similarity measures only but leverage a sentiment analysis model that could confirm the sentiment of words contained in the reviews. Here, we are going to fine-tune and use a deep learning model called Finbert, a Bidirectional Encoder Representation of Transformers (BERT) that has been trained on SEC-10K filings that can better analyze financial terms and jargon compared to the original BERT. Therefore, we hope that this model can better classify if the word belongs to a particular sentiment class (i.e., negative, positive, uncertainty).

An example to visualize these approaches would be a company having financial distress. As a result, their report will be dominantly filled with negative words. Sharp changes in the similarity of the report compared to the previous report could be also used as an indication that major changes are going to affect the stock performance, and the sentiment analysis from the FinBERT model will confirm the direction of the changes.

2.1.4. Risk Mitigation Using Different Similarity Indicators

The original design of this section was intended for pairs trading using clustering. After changing the model into risk mitigation model, the data collection stage remained the same. Stock data, including but not limited to technical indicators, chart data, and fundamental data, were collected. Data cleansing had to be performed on the collected data because the data were not standardized. For instance, technical indicators like the moving average had values similar to the closing price, while some indicators like the moving average convergence divergence (MACD) were roughly averaged on zero. After that, different similarity indices will be covered. Some similarity indices include DTW (dynamic time warping) distance, Euclidean distance, correlation, and cointegration. Each similarity index will be based on different sets of data. After that, the similarity indices will be integrated together for a single similarity score. Then the model will be tested on a dummy portfolio to see if it can properly filter out the stocks that have high similarity scores. Lastly, the filtered portfolio will be backtested to check if it has a lower risk.

2.1.5 Portfolio Management System

According to Figure 1, We will design a flow or a system to combine different strategies by introduced a weighting. Firstly, Multi Factor Equity Model will generate 30 high performing stocks that will be used as a baseline for the portfolio strategy. Then, we will improve the portfolio by introducing Risk Mitigation Algorithm using Similarity Scores to filter highly correlated assets. Moreover, we will further refine the stock selection using NLP to drop some negatively sentiment stocks based on their 10-K report. Finally, we will backtest the Portfolio system and evaluate the metrics to the performance of S&P 500 index and each individual algorithm performance.

2.1.6. Future Roadmap

Going further from the project, we are planning to add some additional features to create a seamless experience for the user to evaluate our algorithm.

1. Add User interface

   We are going to visualize the data in our portfolio management system by creating the Frontend, using React.js. and Backend, using Fast API.

2. Database for persisting user portfolio performance

   We are going to persist user portfolio performance throughout the years using MySQL as RDBMS.

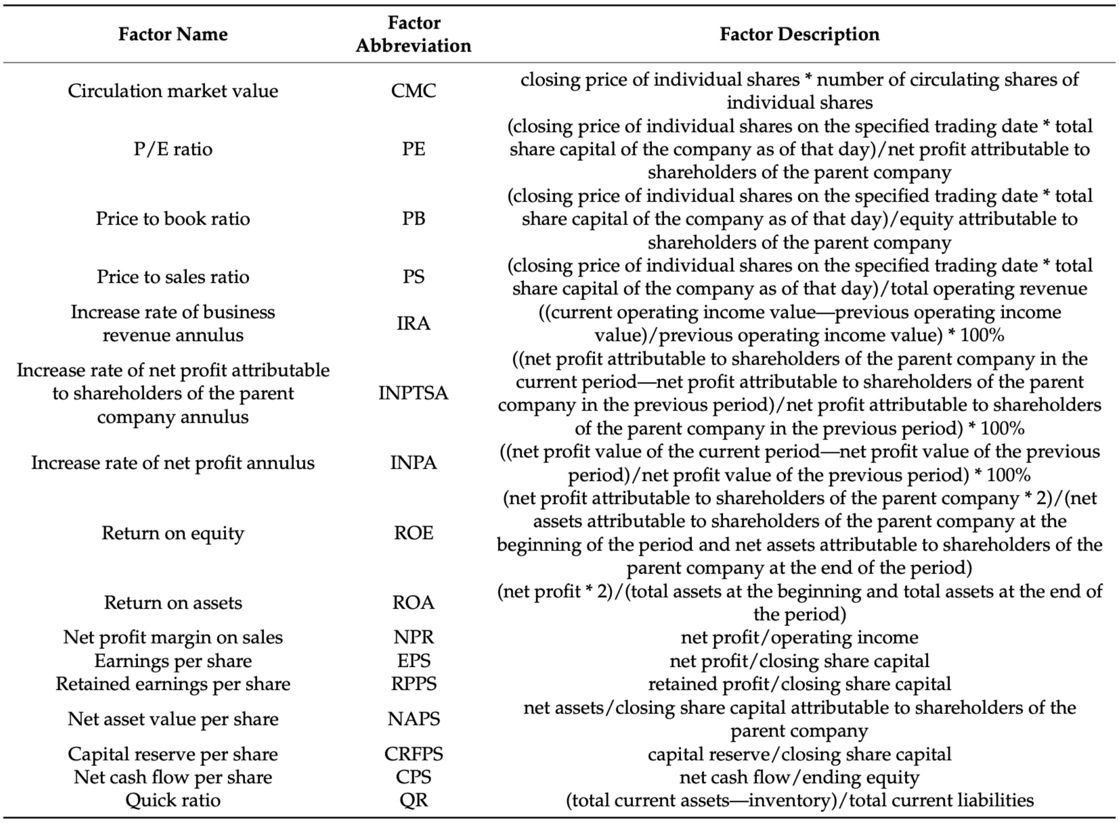
3. Build a real-time trading platform

   We plan to build a platform that can update the data from online sources automatically so that the algorithm can send signals in real-time.

2.2. Implementation

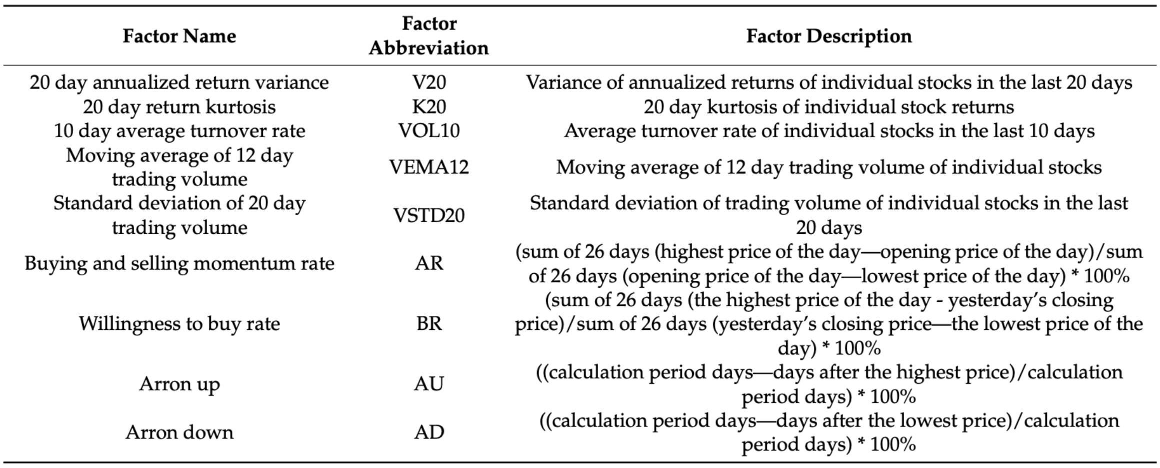
2.2.1. Multi-factors Equity Model

We have found all the stocks in the S&P 500 and built a factor database starting from 1993.02.01 to 2022.12.31 including these stocks' 16 financial factors and 9 technical factors shown in figure 2, and figure 3 respectively.



Factor Name	Factor Abbreviation	Factor Description
Circulation market value	CMC	closing price of individual shares * number of circulating shares of individual shares
P/E ratio	PE	(closing price of individual shares on the specified trading date * total share capital of the company as of that day)/net profit attributable to shareholders of the parent company
Price to book ratio	PB	(closing price of individual shares on the specified trading date * total share capital of the company as of that day)/equity attributable to shareholders of the parent company
Price to sales ratio	PS	(closing price of individual shares on the specified trading date * total share capital of the company as of that day)/total operating revenue
Increase rate of business revenue annulus	IRA	((current operating income value—previous operating income value)/previous operating income value) * 100%
Increase rate of net profit attributable to shareholders of the parent company annulus	INPTSA	((net profit attributable to shareholders of the parent company in the current period—net profit attributable to shareholders of the parent company in the previous period)/net profit attributable to shareholders of the parent company in the previous period) * 100%
Increase rate of net profit annulus	INPA	((net profit value of the current period—net profit value of the previous period)/net profit value of the previous period) * 100%
Return on equity	ROE	(net profit attributable to shareholders of the parent company * 2)/(net assets attributable to shareholders of the parent company at the beginning of the period and net assets attributable to shareholders of the parent company at the end of the period)
Return on assets	ROA	(net profit * 2)/(total assets at the beginning and total assets at the end of the period)
Net profit margin on sales	NPR	net profit/operating income
Earnings per share	EPS	net profit/closing share capital
Retained earnings per share	RPPS	retained profit/closing share capital
Net asset value per share	NAPS	net assets/closing share capital attributable to shareholders of the parent company
Capital reserve per share	CRFPS	capital reserve/closing share capital
Net cash flow per share	CPS	net cash flow/ending equity
Quick ratio	QR	(total current assets—inventory)/total current liabilities

Figure 2. Chosen Financial Factors



Factor Name	Factor Abbreviation	Factor Description
20 day annualized return variance	V20	Variance of annualized returns of individual stocks in the last 20 days
20 day return kurtosis	K20	20 day kurtosis of individual stock returns
10 day average turnover rate	VOL10	Average turnover rate of individual stocks in the last 10 days
Moving average of 12 day trading volume	VEMA12	Moving average of 12 day trading volume of individual stocks
Standard deviation of 20 day trading volume	VSTD20	Standard deviation of trading volume of individual stocks in the last 20 days
Buying and selling momentum rate	AR	(sum of 26 days (highest price of the day—opening price of the day)/sum of 26 days (opening price of the day—lowest price of the day) * 100%
Willingness to buy rate	BR	(sum of 26 days (the highest price of the day - yesterday's closing price)/sum of 26 days (yesterday's closing price—the lowest price of the day) * 100%
Arron up	AU	((calculation period days—days after the highest price)/calculation period days) * 100%
Arron down	AD	((calculation period days—days after the lowest price)/calculation period days) * 100%

Figure 3. Chosen Technical Factors

Here are data sources:

Data	Source	Method
Prices	Yahoo Finance API	yfinance · PyPI [26]
Fundamental Data	fundamentalanalysis API	Financial Modeling Prep [34]

Table 1: Data sources for each data

Both the technical data and monthly excessive returns are calculated from prices. The excess return for an individual stock was defined in formula (2).

$$r_{i,t} = \frac{P^{l}_{i,t+1} - P^{f}_{i,t+1}}{P^{f}_{i,t+1}} - \frac{P^{l}_{500,t+1} - P^{f}_{500,t+1}}{P^{f}_{500,t+1}} \text{ — (2)}$$

where $r_{i,t}$ is the excess return of next month of month t for individual stock i, $P^{l}_{i,t+1}$ is the the last trading day price of month t+1 for individual stock i, $P^{f}_{i,t+1}$ is the the first trading day price of month t+1 for individual stock, $P^{l}_{500,t+1}$ is the the last trading day price of month t+1 for S&P 500, $P^{f}_{500,t+1}$ is the first trading day price of month t+1 for S&P 500.

We transformed our stock selection model into a classification model which contained two categories: rise and fall or 1 and 0. The top 25% (around 126 stocks out of 500) of individual stocks' next month's excess returns were classified as "rise" / "1", while others were classified as "fall" / "0". The output of the model was the individual stock's next month's rising possibility.

We feeded our datasets which are with and without fundamental data into the LSTM model as baselines and did the training and testing. We are using the same methods to construct our GRU stock selection model[9,10]. We have 359 monthly data points in total.

We tried different techniques to generate better predictions:

1. Change Training Period. Considering the some stocks currently in the S&P500 may not exist 30 years ago, we tried both training period of 1993-2020 and 2000-2020.
2. Change HyperParameters. We changed HyperParameters like whether to shuffle the datasets during training.
3. Utilize different Preprocessing Methods. When we generated the labels, we allocated "1" to stocks with high next month return and "0" to others. Considering we only assigned 126 "1" labels among 500 stocks, it may cause data unbalance and affect final prediction performance. We also proposed assigning "3" to top25% stock, "2" to top 25%-50% stocks, and "1" and "0" to the rest.
4. Adjust Time frame. When we passed the data into the model, we passed several years' data together so that the models can learn past years' sequence patterns. We tried to feed 5 years or 9 years data to the models simultaneously.
5. Implement Rolling Strategy. Since the output lists of our models are fixed, we decided to include the latest month data to fine-tune the model when generating the next month stock list.

2.2.2. Price Prediction by Rank-based Sentiment Analysis on the Internet

1. The data is collected as follows:

Reddit's historical data are called archived data, and Reddit does not have any official APIs to allow users to retrieve them. The only way to get this historical data is through third-party websites that archive Reddit data. The most popular and well-archived website is Pushshift [30]. They have most of the Reddit submissions and comments stored as a dump file and also provide an API for users to query it more easily. However, the Pushshift API is very unstable, with many reports of its downtime, and recently underwent a migration, causing the service to be down. Therefore, the dump files are now used as the data source. The dump needs to be downloaded, uncompressed, extracted from relevant subreddits, and stored as JSON. A library called ps_reddit_tool [37] is able to do so.

Historical Google Trends data is all available through the PyTrends API [27]. All historical price data is available in the yfinance API [26].

2. The data are preprocessed as follows:

The goal of reddit data is to obtain the symbol’s submission sentiment, rank the sentiment based on the submission score, and count the number of comments. So first, the relevant subreddits of the symbol are queried by searching the keywords of the symbol in Reddit. Then they are sorted based on the number of subscribers to find the most influential subreddit with more than millions of subscribers.

Symbols	Subreddits
BTC-USD	dogecoin, ethtrader, ethereum, btc, binance
ETH-USD	ethtrader, ethereum
AAPL	technology, wallstreetbets, stocks, investing, business, options, dividends
GOOG	technology, wallstreetbets, stocks, investing, options
AMZN	wallstreetbets, stocks, investing, pennsystocks, options

MSFT	technology, wallstreetbets, stocks, investing, pennsystocks, options, dividends, microsoft
TSLA	technology, wallstreetbets, stocks, dogecoin, investing, pennystocks, options

Table 2: Symbols and subreddits

The keywords are selected based on their symbol name (with a "$" in front for better searching in Reddit) and their company name (first two words to make it more searchable). For example, Apple, Apple Inc. Then, based on the keywords, match the title and body of the submissions to filter out the relevant submissions. The sentiment is analyzed by fitting the title and body into the FinBert model. Aggregate the sentiment, scores, and number of comments into scores*sentiment and comments*sentiment.

For the graph-ranked part, the submission author id and submission comment author id form edges, then the edges form a graph, and then use the pagerank algorithm to find the top author ids. Then, for the features, aggregate the sentiment and rank into rank*sentiment. Page rank was originally implemented by multiplying a rank vector with a stochastic matrix from a directed graph. In this project, we can simply use the PageRank function from NetworkX [33] to perform the ranking. The left image of Figure 4 below is an example of the top-ranked nodes (in red) in a graph, and we can give higher weightings to their posts sentiment. The right image of Figure 4 is an example of the ranked graph of authors of a subreddit.

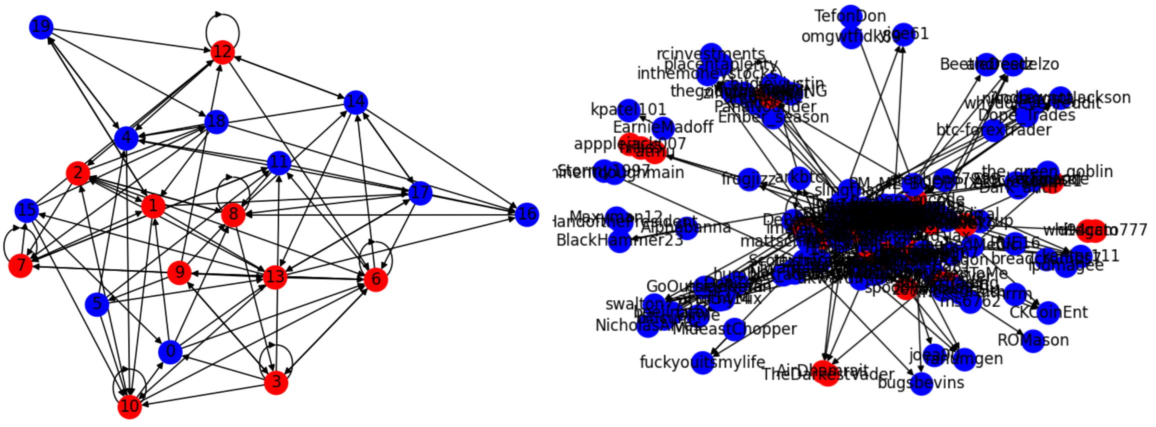


Figure 4. PageRank demonstration and example using authors of a subreddit

Google Trends data is converted to its monthly popularity. This is done by calling the historical API. The data is then normalized, and the percentage change is calculated.

Price data is used to calculate its change percentage and other technical indicators. This is done by calling up monthly prices.

Subreddit each symbol uses; these subreddits are based on the returned subreddit by Reddit based on keywords, and they are sorted to have the top subscriber subreddit to be extracted.

Name	Data type	Method
close	float	monthly close
target	float	monthly close change of next month-this month
reddit volume	int	sum of submissions
reddit total sentiment	float	sum of sentiment
reddit weighted sentiment	float	sum of score*sentiment
reddit ranked sentiment	float	sum of authorrank*sentiment
google trend popularity	float	google trend data
* normalized	float	normalized all features
* change	float	all features row diff

Table 3: The ending features

3. The modeling is done as follows:

The preprocessed data is used to make up a final feature data frame before fitting in the models. For the models, the Random Forest and XGBoost models are used for prediction because previous papers have shown their positive effects, especially XGBoost. This month's and next month's price changes will be the target.

Given all the expected returns given by the model, these data are passed to the portfolio optimization model using minimum volatility. This model will generate weightings for the assets, including shorting positions if their expected returns are negative.

2.2.3. Stock Selection using NLP

Here are the details for implementing the strategy:

1. Indicate the tickers and central key index of the stocks that we are going to analyze.
2. Collect 10-K data to S3 bucket.
3. Implement regex matching to extract Management Discussion and Analysis (MDNA).
4. Chunk the paragraph to list of sentences.
5. Passes each sentences to FINBERT model, find the sentiment of each sentence.
6. Accumulate the sentiment score for each ticker's MDNA.
7. Find the average score and drop tickers with negative sentiment.

Trading Strategy

1. Every time there is a new report for a particular ticker, rebalance our portfolio holdings.
2. Long stock if the sentiment if positive, liquidate stock if sentiment is negative

2.2.4. Risk Mitigation Using Different Similarity Indicators

Initially in the proposal, pairs trading using clustering was to be used. However after some research and development, it has been found out that it is difficult to integrate pairs trading with other trading idea models. This is because the other models select stocks with good performances, while pairs trading does not consider the stock's performance, but rather similarity between stocks. In order to adapt the previous idea and fully utilize the data that had been collected, risk mitigation through similarity between stocks was selected to be used. The basic idea behind the risk mitigation model is that removing stocks that perform similarly can reduce the risk and drawdown.

1. Data Collection

Both fundamental data and technical data were collected for preprocessing. For technical data, moving average for 5 days was used. Investors and traders commonly use moving average to analyze trends and price movements of a security or market over a period of time. Moving average smooths out the fluctuations in the price and can be used to identify the overal direction of the trend. By using moving average, the noise and abnormal fluctuations in the price could be reduced. Technical data was collected using the yfinance API.

To collect fundamental data, the fundamentalanalysis API was utilized, which provided access to five important fundamental ratios: dividend payout ratio, return on equity, net income margin, gross profit margin, and debt ratio. These ratios were chosen due to their ability to provide insights into a company's financial health and performance. Unlike data such as market capitalization, these ratios offer a standardized comparison regardless of the company's size or nature. By collecting these ratios, a more comprehensive understanding of a company's fundamentals can be obtained, allowing for more informed investment decisions.

2. Data Preprocessing

Technical data were collected daily, while fundamental data were collected only 4 times a year. This is because most firms release financial reports quarterly. Thus, the same set of fundamental data was duplicated and used for around 90 days. Then, typical data preprocessing steps have been performed for both data. Data preprocessing includes but is not limited to: filling in missing values with the median value for a certain period and standardizing and scaling the data to a common range.

Unlike fundamental data which were all ratios, technical data (moving average) had different ranges for different stocks. Thus, the data were standardized using StandardScaler from python library sklearn.

3. Create Distance Matrices

In order to figure out the similarity between stocks using the technical data and the fundamental data, distance matrices were created. This is similar to the process in clustering.

In clustering techniques in machine learning, data points are assigned distances to each other, and then data points that are close together are first grouped into a single cluster. Instead of clustering the stocks and then assigning similarity scores based on the clusters, distance between the stocks were used for the similarity scores.

Since the technical data being analyzed involved time-series data in the form of moving averages for each stock, the DTW (dynamic time warping) distance was utilized. This technique measures the similarity between two time-series data sets that may vary in speed or timing by warping the time axis of one series in a non-linear fashion to optimally align the two series, enabling matching of corresponding time points. By using DTW distance, it was able to compare and analyze the trend hidden behind the moving average time series data in a more robust and accurate manner.

For fundamental data, typical euclidean distance was used to create the distance matrix. This is because unlike technical data, fundamental data only changes four times a year. Thus, for a single stock, the fundamental data were same for a single month. This allowed us to use the typical euclidean distance. To calculate the pairwise distance between the stocks, python library sklearn was used.

Finally, the method of correlation was utilized to establish similarity between two stocks, which is a conventional approach. A correlation matrix was generated, and its values were scaled up by a factor of 10 to so that the range and scale is consistent with that of the two previously created distance matrices. In contrast, the proposed method of cointegration for pairs trading was not employed because it does not measure similarity between two time series data, but instead determines whether the difference between the two time series data exhibits a mean reverting characteristic.

4. Filtering simlar stocks out

The model receives a list of stocks and the selected timeframe (year and month) as the input and filters out the stocks that have high similarity score. Depending on the target size of the final list of stocks, the model will remove the stocks that have high similarity score from the original list of stocks.

2.2.5 Portfolio Management System

We introduce three weights to demonstrate the effect of each individual algorithm to the combined portfolio system. A grid search will be done to discover the ideal weights namely (w_{NLP}, $w_{SIMILARITY}$, w_{GRU}) such that the combined portfolio could produce the greatest risk-adjusted return and smallest drawdown. w_{NLP}, $w_{SIMILARITY}$ represents the proportion of the stocks that need to be dropped because of negative sentiment or high similarity. In the grid search, we set the step size to 0.05 for both w_{NLP} and $w_{SIMILARITY}$.

Because our aim is to discover the ideal combination portfolio with high risk-adjusted returns, we rated the grid search results by Sharpe Ratio and will take the top 10 results out of 50 trials.

2.3. Testing

During the development process, we will unit test each of our prediction strategies and backtest our portfolio management system with historical data.

2.3.1. Multi-factor Equity Model

Our trading strategy for the multi-factor model was that we constructed the portfolio by buying stocks with the top 30 rising possibilities predicted by the stock selection model. At the end of each month, we changed our position to newly selected 30 stocks. If the stocks selected didn't change, we would keep holding the stocks. We did our backtesting using this trading strategy on QuantConnect. Our test period was 2 years which was from 2021.1.1 to 2022.12.30.

2.3.2. Price Prediction by Rank-based Sentiment Analysis on the Internet

Before we pass the predicted prices into the portfolio optimization layer, some preliminary testing can be performed on the model to assess its accuracy and performance. For accuracy, we will compare the actual and predicted price changes and measure their error by comparing whether the changes are in the same direction. For performance, we will have a preliminary trading strategy using sentiment and the predicted price as the factors. When the expected return is positive, buy, or else sell. Then, we measure the overall return by subtracting the buy and sell execution prices. The data will be in a monthly time frame. The training period will be from 2018–2020, with testing in 2021–2022. The stocks collected at the end are limited, namely AAPL, AMZN, BTC-USD, ETH-USD, GOOG, MSFT, and TSLA. The assets are picked because, due to limited resources, only a limited number of assets can be selected. These assets are picked because they are well known, well discussed, and easy to distinguish from each other, which makes them more likely to appear in social media.

Accuracy = predicted return * actual returns > 0

There are a few baselines for reference.

- Price features including volume, high, low, and close
- sentiment and prices features
- all reddit features, google trends, and price features

All the models, including Random Forest and XGBoost, are tested and compared for their effectiveness. Another experiment is to select the good symbols based on their previous performance using individual rolling 12-month trading strategies. The results will be shown in the evaluation part.

The portfolio weighting is optimized based on minimal volatility, with max weights of 0.5 and -0.5, preventing dominant asset weighting and supporting short selling as well. The portfolio weightings will be converted into an integer allocation using the discrete allocation function.

Finally, the performance of the portfolio model will be tested on its cumulative returns, Sharpe ratio, and max drawdown to see if it can yield positive returns.

2.3.3. Stock Selection Using NLP

We ran unit testing to evaluate every preprocessing step that we used to produce the vector representation of every 10-K document. Then, we stored the vector representation in our database, so we could compare it with future reports. For every new report, we created a scoring function with FinBERT model as we mentioned in section 2.2.3. There were two units testing done on the scoring function. First, to ensure that the score are ranging between -1 (very negative) and 1 (very positive). Second, to summarize longer sentence in order not to break the scoring model that can only receive a maximum 512 tokens.

Then, we did integration testing with the QuantConnect backtesting system. We leveraged the QuantConnect log system to do the monitoring. Lastly, we tested that our algorithm would only execute trades if and only if the company released its yearly or quarterly report.

2.3.4. Risk Mitigation Using Different Similarity Indicators

The reliability of data sources is crucial for any data-driven analysis. In this study, two APIs were utilized - yfinance and fundamentalanalysis - both of which are considered reliable sources of financial data. Nevertheless, it was necessary to ensure that the extracted data were accurate. To achieve this, unit testing was conducted during the data collection stage. Moreover, for some stocks, the fundamental data extracted from the APIs were cross-checked with the actual financial reports to verify their accuracy. Similarly, for technical data, each function for every technical indicator was subjected to unit testing to ensure the accuracy and reliability of the extracted data. Overall, these measures were taken to ensure that the data used in this study were dependable and could be relied upon to draw meaningful conclusions.

To test whether the risk mitigation model works well, 30 stocks selected from the multi-factor model was used. Below is a table of 5 pairs of stocks with the highest similarity scores. Lower similarity score means that the pair is more similar and expected to perform similarly.

Stock 1	BKR	EOG	BKR	DVA	MOS
Stock 2	MRO	APA	EOG	DRI	WDC
Similarity Score	-8.353	-8.070	-8.008	-7.917	-7.889

Table 4: Top 5 similar stock pairs and similarity scores

To test the backbone idea behind the risk mitigation model, which is an idea that removing stocks that perform similarly can reduce the risk and drawdown, a simple buy-and-hold strategy was used. Two portfolios, one with 30 original stocks and the other with 20 filtered stocks, were backtested with the same strategy for the same period of time. The results of the testing will be discussed in the evaluation section.

2.3.5 Portfolio Management System

As the correctness for each individual algorithm has been tested before combining the strategy, we only ran two unit tests. Both unit tests were run to ensure that we drop the correct amount of stocks according to the W_{NLU} and $W_{SIMILARITY}$ proportion respectively.

2.3.6 Backtesting System Test

For each strategy, we ran a unit test to ensure there has not been any margin call, cancelled or unfilled trades on the system. We did it by logging the trades each time the order has been executed and assert the list of executed order with our intended stock list.

2.4. Evaluation

2.4.1. Multi-factor Equity Model

In this section, we evaluated the models based on their sharpe ratio because our objective is to find a model that can achieve a better return and minimize the risk. Our testing period is from 2021.1.1 to 2022.12.30.

Baseline	PSR	MDD	Sharpe Ratio	Net Profit	Annual Standard Deviation	CAGR
S&P 500 Index	9.10%	23.5%	0.22	6.24%	0.155	3.07%

Table 5: Performance of the benchmark index

As we mentioned above, we use some techniques to improve the model performance.

We first compared the model trained with shuffled datasets and the model with not shuffled datasets.

Model	PSR	MDD	Sharpe Ratio	Net Profit	Annual Standard Deviation	CAGR
LSTM - Technical only	22.94%	12.3%	0.58	16.99%	0.105	8.15%
GRU - Technical only	19.27%	13.0%	0.50	13.81%	0.101	6.67%
LSTM - Fundamental + Technical	21.92%	10.3%	0.56	15.23%	0.096	7.33%
GRU Fundamental + Technical	**48.64%**	**17.6%**	**1.06**	**54.23%**	**0.163**	**24.15%**

Table 6: Results of models trained in period 1993-2000, with n_past=9, Shuffle=True

Model	PSR	MDD	Sharpe Ratio	Net Profit	Annual Standard Deviation	CAGR
LSTM - Technical only	26.16%	12.2%	0.64	18.14%	0.098	8.68%
GRU - Technical only	21.76%	14.8%	0.56	18.60%	0.119	8.89%
LSTM - Fundamental + Technical	32.37%	16.6%	0.77	35.30%	0.157	16.29%
GRU Fundamental + Technical	34.76%	9.8%	0.80	25.04%	0.105	11.81%

Table 7: Results of models trained with period 1993-2020, with n_past=9, Shuffle=False

Overall, in terms of the sharpe ratio, models trained with unshuffled data perform better than shuffled ones. This might be because the market has the momentum implying that time order can affect the model performance and the recent factors are more useful than the data far from the current time slot. Although the best model trained from shuffled datasets, the corresponding unshuffled model “GRU Fundamental + Technical” also has the highest sharpe ratio among other models.

Model	PSR	MDD	Sharpe Ratio	Net Profit	Annual Standard Deviation	CAGR
LSTM - Fundamental + Technical (n_past = 9)	27.22%	12.5%	0.666	21.21%	0.111	10.08%
LSTM - Fundamental + Technical (n_past = 5)	16.97%	11.3%	0.445	11.30%	0.093	5.49%
GRU Fundamental + Technical (n_past = 9)	23.75%	12.5%	0.594	14.81%	0.09	7.14%
GRU Fundamental + Technical (n_past = 5)	36.02%	9.8%	0.82	24.60%	0.101	11.61%

Table 8: Results of Models trained with period 2000-2020, Shuffle=True

Model	PSR	MDD	Sharpe Ratio	Net Profit	Annual Standard Deviation	CAGR
LSTM - Fundamental + Technical (n_past = 9)	29.71%	12.9%	0.72	27.76%	0.132	13.01%
LSTM - Fundamental + Technical (n_past = 5)	31.14%	10.8%	0.73	20.62%	0.097	9.81%
GRU Fundamental + Technical (n_past = 9)	20.62%	16.5%	0.53	19.93%	0.135	9.50%
GRU Fundamental + Technical (n_past = 5)	26.50%	11.5%	0.65	18.27%	0.096	8.74%
LSTM Rolling Strategies	8.10%	27.5%	0.16	2.02%	0.179	1.04%
GRU Rolling Strategies	10.73%	22.1%	0.26	6.14%	0.259	3.15%

Table 9: Results of Models trained with period 2000-2020, Shuffle=False

According to the tables above, the performance of the models trained with period 2000-2020 is quite similar to the models trained with period 1993 -2020.

Besides, GRU performed better when the training sets were with the past window of 5 years, while LSTM performed better when the training sets were with the past window of 9 years. This can be because GRU has only two gates in its architecture while LSTM has three gates. If the dataset is small then GRU is preferred otherwise LSTM for the larger dataset.

The rolling strategies which are the model finetuned with the latest datasets performed not good because the model may have bad marketing time when adjusting the portfolio and frequent trading will need high agency or commission cost.

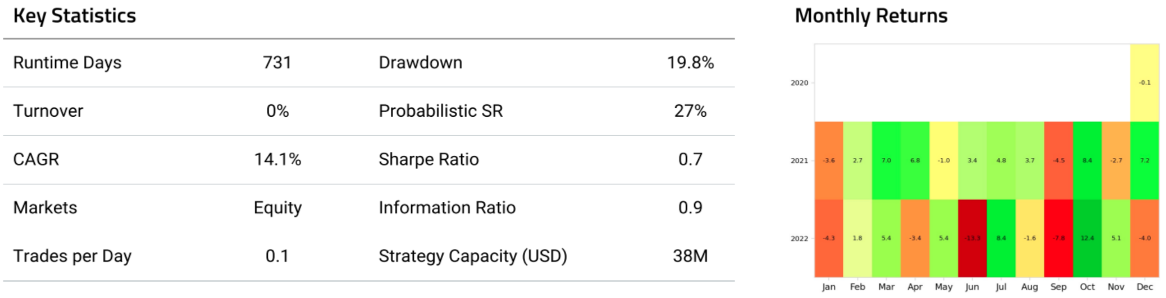
Model	PSR	MDD	Sharpe Ratio	Net Profit	Annual Standard Deviation	Compounding Annual Return
GRU - Techical	36.13%	10.6%	0.813	17.19%	0.084	8.61%
LSTM - Techical	23.55%	11.7%	0.583	12.84%	0.094	6.49%
GRU - Techical + Fundamental	35.62%	13.6%	0.812	23.08%	0.110	11.42%
LSTM - Technical + Fundamental	24.82%	12.6%	0.608	14.34%	0.098	7.23%

Table 10: Results of the models with preprocessing label "0123" with period 2000-2020, with Shuffle = False

The models with preprocessing label "0123" performs better than the models with preprocessing label "01" in general because the new proposed preprocessing method solved the data unbalanced issue and the models can learn more features from the datasets.

In conclusion, our models' performance is much better than S&P 500 in terms of all aspects. Our best model is GRU trained with both technical and fundamental data, period 1993-2020, n_past = 9 and Shuffle = True which can achieve sharp ratio 1.06, CAGR 24.15% and Max Drawdown 17.6%. Compared with S&P500 in the same testing period, the model can have an excessive CAGR of 21.08%, which is aligned with our objectives.

2.4.2. Stock Selection Using NLP Strategy



Key Statistics

Runtime Days	731	Drawdown	19.8%
Turnover	0%	Probabilistic SR	27%
CAGR	14.1%	Sharpe Ratio	0.7
Markets	Equity	Information Ratio	0.9
Trades per Day	0.1	Strategy Capacity (USD)	38M

Monthly Returns

	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
2020												-0.1
2021	-3.6	2.7	7.0	6.8	-1.0	3.4	4.8	3.7	-4.5	8.4	-2.7	7.2
2022	-4.3	1.8	5.4	-3.4	5.4	-13.3	8.4	-1.6	-7.8	12.4	5.1	-4.0

Figure 5: Backtest Result of Stock Selection Using NLP from January 2021 to December 2022

The NLP strategy had dropped twenty-two negative sentiment 10-K filings from a total of sixty 10-K filings spanning from January 2021 to December 2022. As shown in Figure 5, we reaped an annualized profit of 14.1%, a Sharpe Ratio of 0.7, and the Max Drawdown of 19.8% for the previous 2 years since January 2021. During the same period, SPY generated a CAGR of 2.436%, a Sharpe Ratio of 0.186, and a 24.5% Max Drawdown. It is shown that the Sharpe Ratio of the NLP stock selection was higher by 0.614 points, CAGR and the max drawdown were higher by 11.664 points and 4.7 than those of SPY, respectively. Overall, it illustrates that the NLP-based stock selection technique gave higher returns and reduced the volatility over the U.S. stock index.

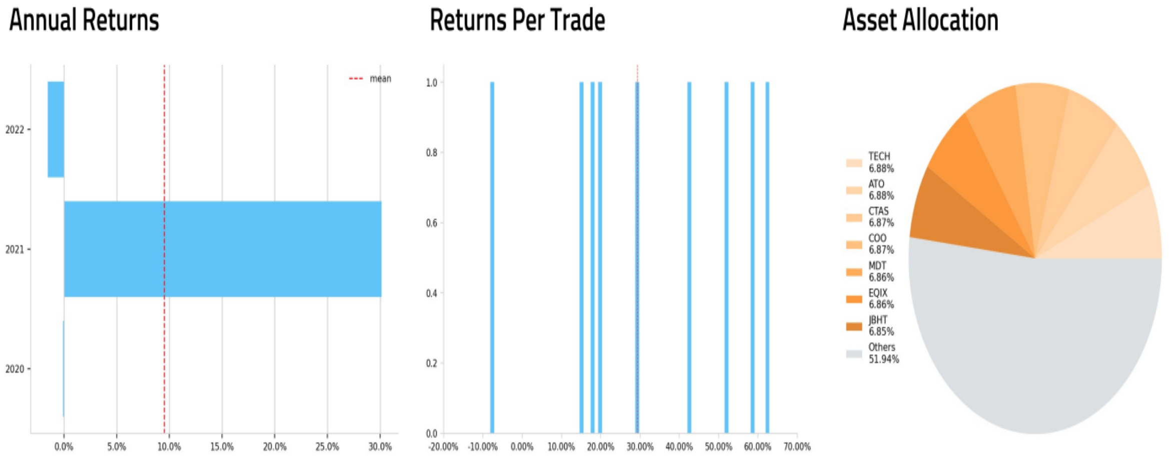


Figure 6: Annual Return (left) , Returns per trade (middle), and Asset Allocation (right) of Stock Selection Using NLP from January 2021 to December 2022

Looking at the long exposure of the NLP stock selection trading strategy, it is clear that the NLP trading strategy seemed to have relatively good predictive power in measuring the volatility of the portfolio. In 2022, the CAGR for SPY was -18.71%, while we were able to maintain our CAGR at -5.24%. Our algorithm was efficient in terms of generating a positive return per trade as shown in the middle figure. The Asset allocation chart described our objective to hedge the risk by diversifying our portfolio into multiple assets, thus minimizing our downturn risk.

Overall, the algorithm performed well compared to SPY benchmark, but inferior to other algorithm such as Multi Factor. It is mainly becaused NLP only use textual information to create a trading decision while Multi Factor have various modalities such as technical time series data and fundamental data that captures the price of the stocks holistically.

2.4.3. Price Prediction by Rank-based Sentiment Analysis on the Internet

Selection of features

As mentioned, to prove the features work, the baseline of only price features (close, volume, high, low) and limited features (close, sentiment) are first tested. However, they had low accuracy and negative returns. Instead, when all the features were considered (close, sentiment, ranked sentiment, and Google Trends interest), the accuracy was high and yielded positive returns. Therefore, all the features will be added for further testing.

Individual asset model preliminary performance

2020-01 to 2020-11 in rolling basis, using all previous month's features (from 2018-01 to 2019-12), accuracy is as follows:

Symbol\Model	Random Forest	XGBoost
BTC-USD	54.5%	**72.7%**
ETH-USD	**63.6%**	54.5%
AAPL	45.5%	**81.8%**
GOOG	27.7%	45.5%
AMZN	45.5%	54.5%
MSFT	45.5%	**63.6%**
TSLA	36.4%	45.5%

Table 11: Comparison of each individual asset and machine learning model

When comparing the decision tree and XGBoost model, XGBoost outperforms. The result is consistent with the reference paper; their best models are also getting around 70% accuracy, and their best model was also XGBoost overall. However, results could be slightly different because we are not testing in the same period with the same assets. As we can see, cryptocurrency and some stocks like AAPL and MSFT have higher accuracy for prediction, which shows that they are more correlated with the subreddit sentiment than other assets. Other asset prices might be based on information elsewhere.

Therefore, XGBoost is used to further generate the expected returns for the portfolio model. Also, stocks that most closely correlate with social media sentiment are selected for the portfolio.

Feature Importance

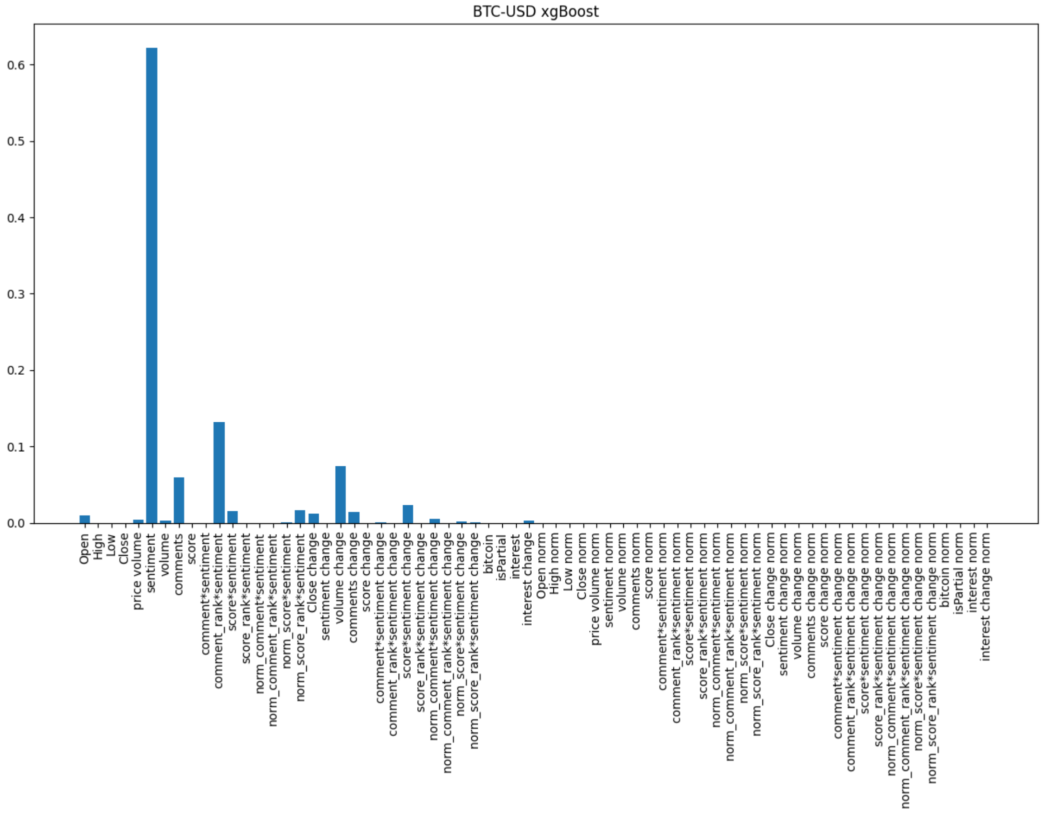


Figure 7: Feature importance of xgBoost

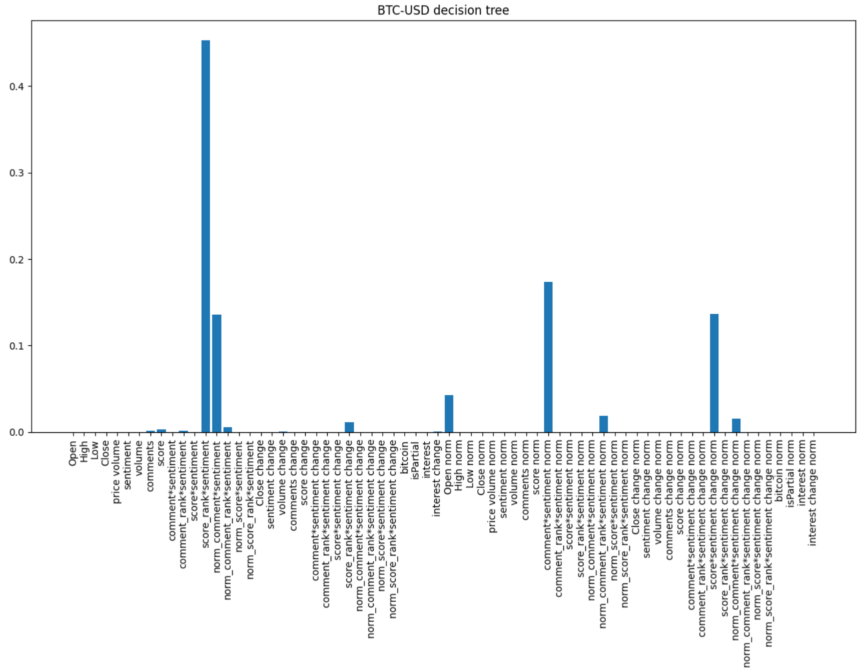


Figure 8: Feature importance of decision tree

The way the decision tree worked is that, based on the previous month's data, it created some conditions to divide the features and the target. As we can see, features like sentiment, score*sentiment, and comment*sentiment are important for the decision tree for making predictions, which show that the normal or ranked sentiment is somewhat correlated with the next month's price change. However, for some assets and subreddit sentiment, they may not have a correlation, and there are not any outstanding important features. The decision can be interpreted in such a way that the previous sentiment has a range of greater and lesser values. So if the next month's sentiment is suddenly very high and potentially unprecedented, then it's likely indicating a signal.

Another thing is that, surprisingly, Google Trends interest was not a key driver of feature importance. Also, the score or comments by themselves do not contribute to the prediction, but only when multiplied by the sentiment.

Special behavior

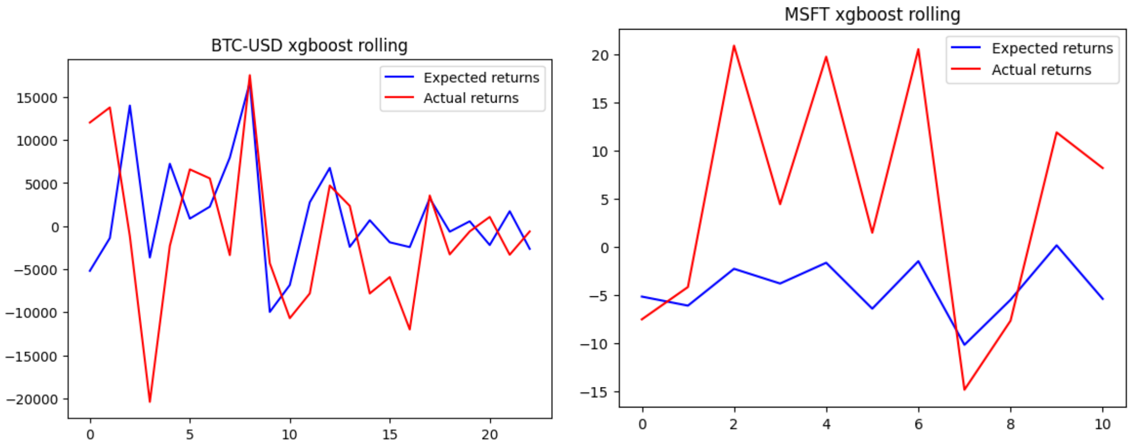
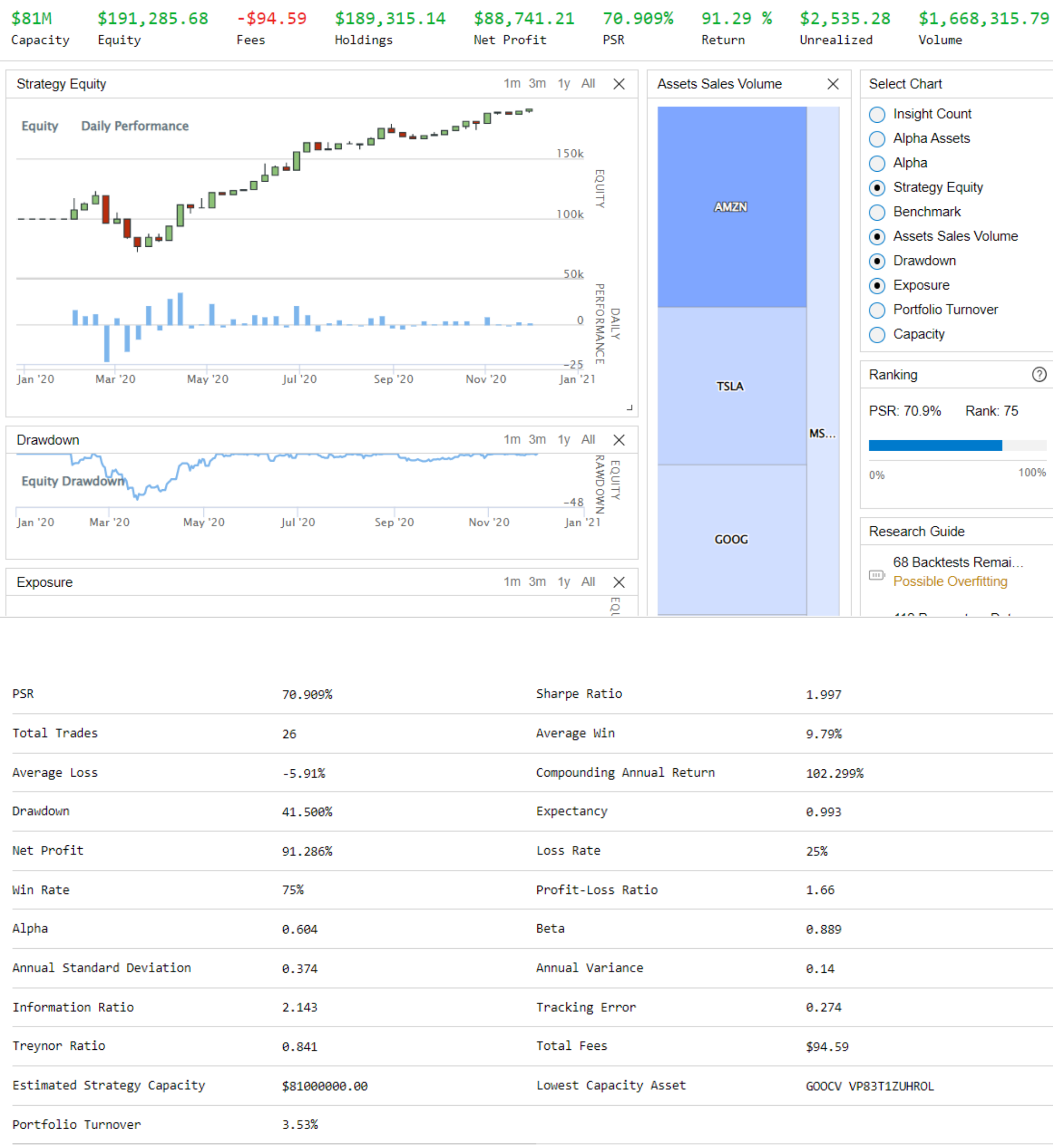


Figure 9: Comparison between AMZN and MSFT on xgBoost rolling

Also, sometimes the predicted and actual return changes look really similar but are a bit laggy. For example, although the shape here looks basically the same, the predicted change is actually faster than the actual change, so an offset of prediction can be added in the future or some value vs. change issues can be resolved. This is consistent with the previous paper, as they mention the laggy situation as well.

Portfolio longing all assets

As we can see, the portfolio was able to achieve financial gain and maintain its positive returns. The resulting portfolio has 90% returns, a 1.997 Sharpe ratio, and a 41.5% drawdown. This shows the high-risk, high-return nature of this strategy.



$81M	$191,285.68	-$94.59	$189,315.14	$88,741.21	70.909%	91.29 %	$2,535.28	$1,668,315.79
Capacity	Equity	Fees	Holdings	Net Profit	PSR	Return	Unrealized	Volume

Select Chart

- ○ Insight Count
- ○ Alpha Assets
- ○ Alpha
- ● Strategy Equity
- ○ Benchmark
- ● Assets Sales Volume
- ● Drawdown
- ● Exposure
- ○ Portfolio Turnover
- ○ Capacity

Ranking

PSR: 70.9% Rank: 75

0% 100%

Research Guide

68 Backtests Remai...
Possible Overfitting

PSR	70.909%	Sharpe Ratio	1.997
Total Trades	26	Average Win	9.79%
Average Loss	-5.91%	Compounding Annual Return	102.299%
Drawdown	41.500%	Expectancy	0.993
Net Profit	91.286%	Loss Rate	25%
Win Rate	75%	Profit-Loss Ratio	1.66
Alpha	0.604	Beta	0.889
Annual Standard Deviation	0.374	Annual Variance	0.14
Information Ratio	2.143	Tracking Error	0.274
Treynor Ratio	0.841	Total Fees	$94.59
Estimated Strategy Capacity	$81000000.00	Lowest Capacity Asset	GOOCV VP83T1ZUHROL
Portfolio Turnover	3.53%		

Figure 10: Backtesting result on QuantConnect

Individual performance in portfolio

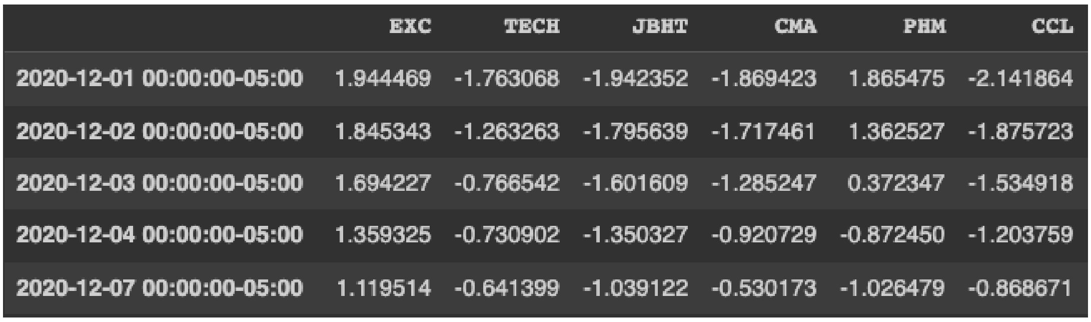
Symbol	Sharpe Ratio & Draw Down
AAPL	2.276 SR & 3.3% DD
AMZN	0.773 SR & 11.5% DD
ETH	1.371 SR & 5.2%
GOOG	1.453 SR & 4% DD
MSFT	0.007 SR & 6.8% DD
TSLA	0.966 SR & 35.9% DD

Table 12: Performance of each asset in the portfolio

As we can see, some assets have a higher correlation with social media sentiment and rely on Reddit for their price changes, as reflected in the features' importance. Compared to other poorly performing assets, the well-performing asset would have some feature importance that is a sentiment, so the price either depends on the general sentiment or the ranked sentiment in Reddit. Poorly performing assets would indicate that their drivers are elsewhere, either from another source or some other subreddit that is not collected. Therefore, this proves that social media sentiment could help predict asset prices if used on the right asset. Bitcoin and Ethereum are always believed to depend on social media sentiment as they do not have traditional financial analysis. Also, cryptocurrency has higher volatility, so it is likely to gain higher profits if timed correctly. Therefore, cryptocurrency and some other stocks outperform stocks in general in this model in terms of the Sharpe ratio.

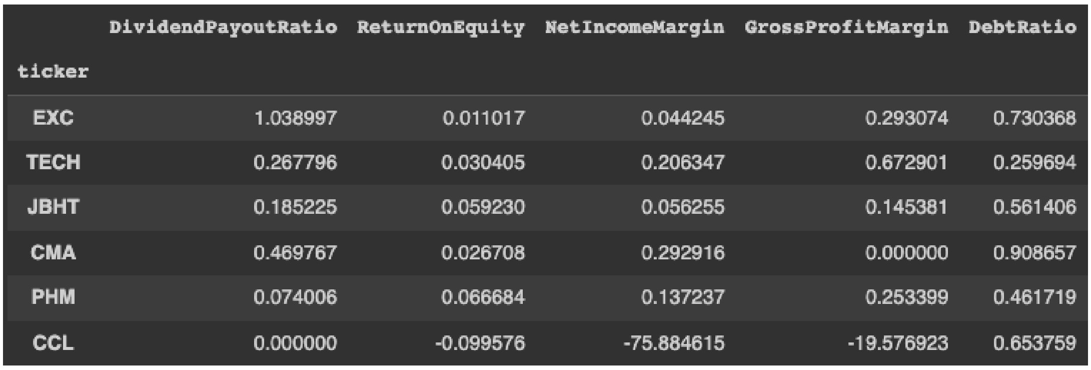
2.4.4. Risk Mitigation Using Different Similarity Indicators

Regarding the objectives in the data side, which are Data Collection, Data Storage, and Data Preprocessing, the risk mitigation model worked well. A set of technical data and a set of fundamental data of the stock listed in S&P500 for the time period of 2010~2022 was collected. The code was able to automate the process so that the user can easily change the timeframe if needed. For the fundamental data, the user can easily select the feature needed by tweaking the code. Since this part of the model focused on the risk quantification side of the other models, evaluation is focused on the objective: Risk Quantification. Below is the result of running the risk mitigation model on the list of 30 stocks selected by multy-factor equity model.



	EXC	TECH	JBHT	CMA	PHM	CCL
2020-12-01 00:00:00-05:00	1.944469	-1.763068	-1.942352	-1.869423	1.865475	-2.141864
2020-12-02 00:00:00-05:00	1.845343	-1.263263	-1.795639	-1.717461	1.362527	-1.875723
2020-12-03 00:00:00-05:00	1.694227	-0.766542	-1.601609	-1.285247	0.372347	-1.534918
2020-12-04 00:00:00-05:00	1.359325	-0.730902	-1.350327	-0.920729	-0.872450	-1.203759
2020-12-07 00:00:00-05:00	1.119514	-0.641399	-1.039122	-0.530173	-1.026479	-0.868671

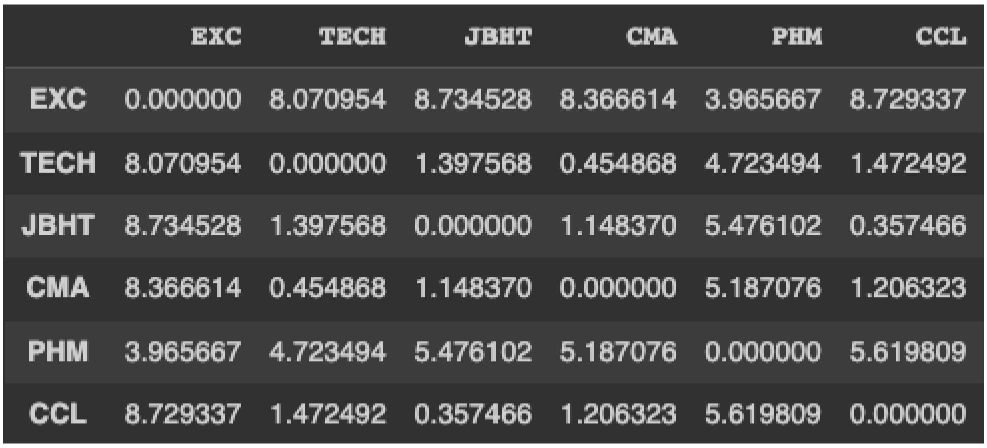
Figure 11: A snapshot of the table with moving average of 30 stocks



	DividendPayoutRatio	ReturnOnEquity	NetIncomeMargin	GrossProfitMargin	DebtRatio
ticker					
EXC	1.038997	0.011017	0.044245	0.293074	0.730368
TECH	0.267796	0.030405	0.206347	0.672901	0.259694
JBHT	0.185225	0.059230	0.056255	0.145381	0.561406
CMA	0.469767	0.026708	0.292916	0.000000	0.908657
PHM	0.074006	0.066684	0.137237	0.253399	0.461719
CCL	0.000000	-0.099576	-75.884615	-19.576923	0.653759

Figure 12: A snapshot of the table with fundamental ratios of 30 stocks

To calculate similarity matrix between stocks for risk mitigation, three similarity scores: DTW distance of moving average, Euclidean distance of fundamental ratios, and correlation, were calculated.



	EXC	TECH	JBHT	CMA	PHM	CCL
EXC	0.000000	8.070954	8.734528	8.366614	3.965667	8.729337
TECH	8.070954	0.000000	1.397568	0.454868	4.723494	1.472492
JBHT	8.734528	1.397568	0.000000	1.148370	5.476102	0.357466
CMA	8.366614	0.454868	1.148370	0.000000	5.187076	1.206323
PHM	3.965667	4.723494	5.476102	5.187076	0.000000	5.619809
CCL	8.729337	1.472492	0.357466	1.206323	5.619809	0.000000

Figure 13: A snapshot of the DTW distance matrix of 30 stocks

From the graphs below, we can note the difference between stocks that have a low DTW distance and stocks that have a high DTW distance. stocks that have a similar trend and movement had small DTW distance, while stocks that have different trends and movements had large DTW distance. MOS and JBHT had DTW distance of 0.2939, while EXC and JBHT had DTW distance of 8.7345

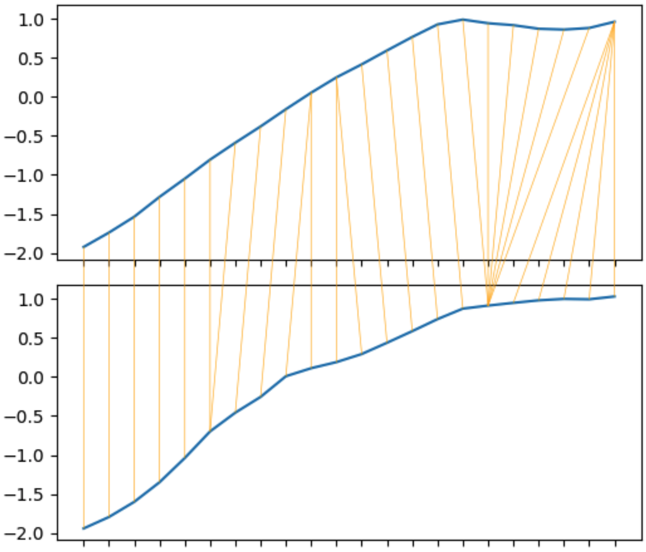


Figure 14: DTW distance between MOS and JBHT

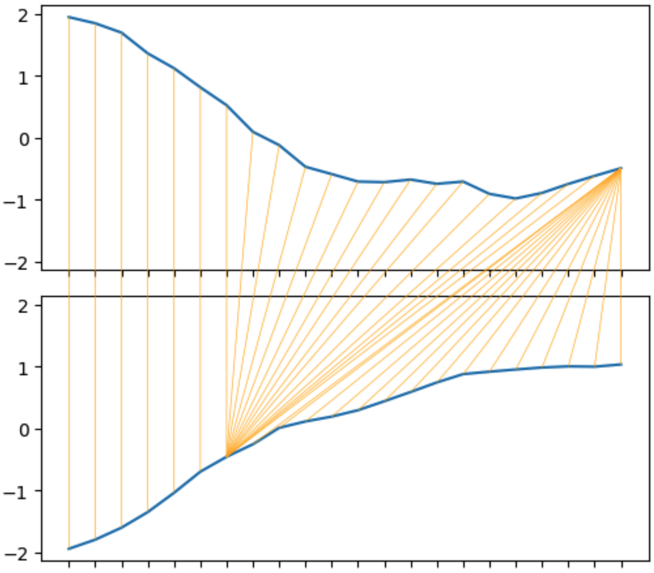
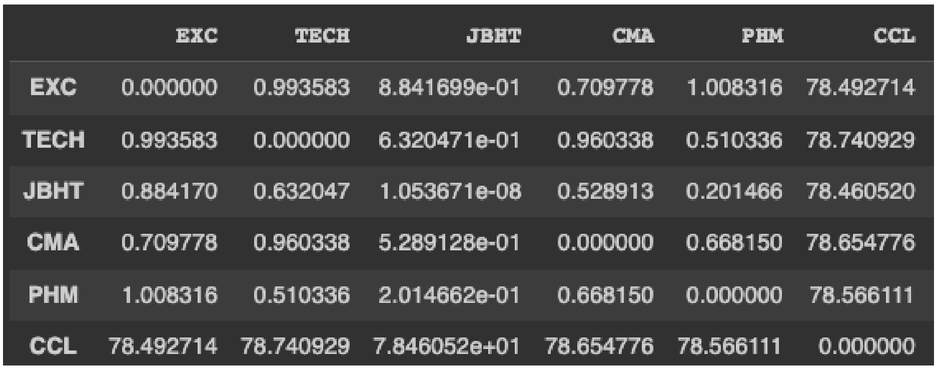


Figure 15: DTW distance between EXC and JBHT

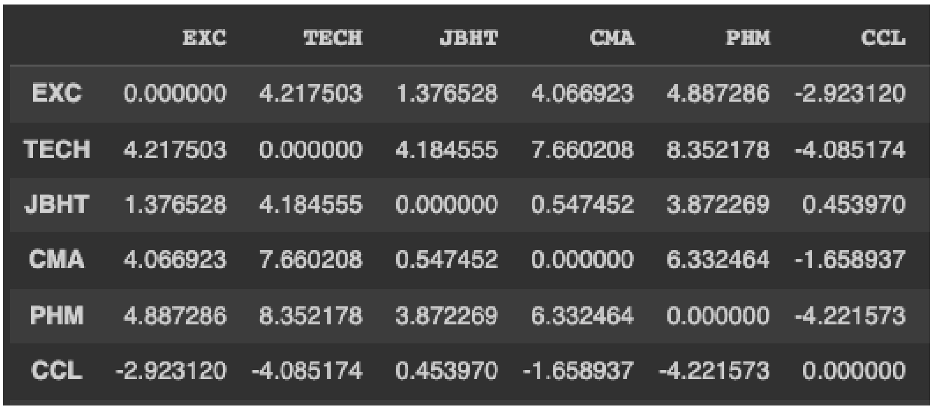
Below shows the Euclidean distance calculated from the fundamental data.



	EXC	TECH	JBHT	CMA	PHM	CCL
EXC	0.000000	0.993583	8.841699e-01	0.709778	1.008316	78.492714
TECH	0.993583	0.000000	6.320471e-01	0.960338	0.510336	78.740929
JBHT	0.884170	0.632047	1.053671e-08	0.528913	0.201466	78.460520
CMA	0.709778	0.960338	5.289128e-01	0.000000	0.668150	78.654776
PHM	1.008316	0.510336	2.014662e-01	0.668150	0.000000	78.566111
CCL	78.492714	78.740929	7.846052e+01	78.654776	78.566111	0.000000

Figure 16: A proportion of the Euclidean distance matrix for 30 stocks

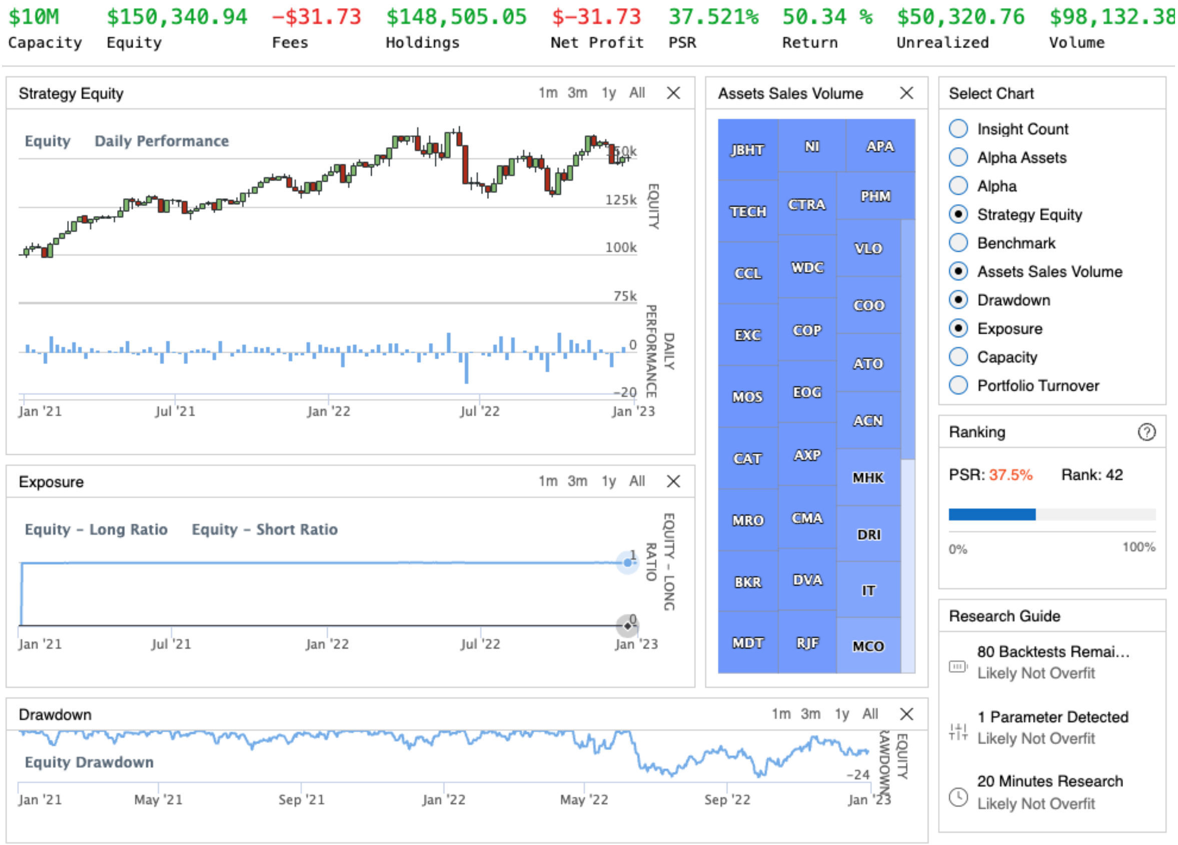
Correlation matrix was easily calculated with an existing function in Pandas. The values were multiplied by 10.



	EXC	TECH	JBHT	CMA	PHM	CCL
EXC	0.000000	4.217503	1.376528	4.066923	4.887286	-2.923120
TECH	4.217503	0.000000	4.184555	7.660208	8.352178	-4.085174
JBHT	1.376528	4.184555	0.000000	0.547452	3.872269	0.453970
CMA	4.066923	7.660208	0.547452	0.000000	6.332464	-1.658937
PHM	4.887286	8.352178	3.872269	6.332464	0.000000	-4.221573
CCL	-2.923120	-4.085174	0.453970	-1.658937	-4.221573	0.000000

Figure 17: A proportion of the correlation matrix for 30 stocks

To evaluate the performance of the risk management model, 30 stocks selected from the multi-factor model was used for the baseline. The risk model first calculated the similarity scores matrix from the list of stocks. The pairs of stocks with high similarity were removed by sorting the scores and respecting the pairs until the list had 20 stocks. A simple buy-and-hold strategy was tested using QuantConnect platform from 2021 January to 2022 December to examine the underlying idea that removing similarly performing stocks can reduce risk and drawdown, even if it may lead to reduced returns or losses. Below is the backtesting results in QuantConnect platform.



$10M	$150,340.94	-$31.73	$148,505.05	$-31.73	37.521%	50.34 %	$50,320.76	$98,132.38
Capacity	Equity	Fees	Holdings	Net Profit	PSR	Return	Unrealized	Volume

Select Chart

○ Insight Count
○ Alpha Assets
○ Alpha
● Strategy Equity
○ Benchmark
● Assets Sales Volume
● Drawdown
● Exposure
○ Capacity
○ Portfolio Turnover

Ranking

PSR: 37.5% Rank: 42

0% 100%

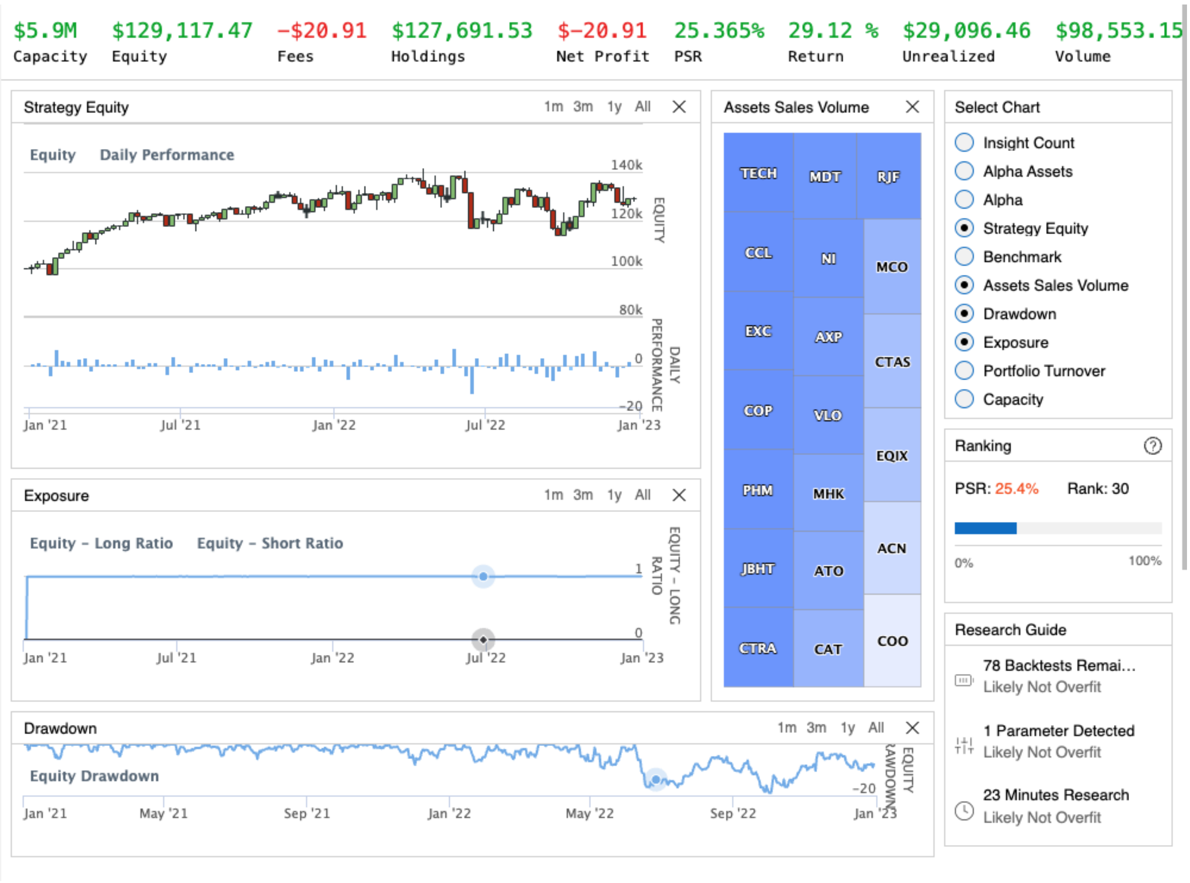
Research Guide

80 Backtests Remai…
Likely Not Overfit

1 Parameter Detected
Likely Not Overfit

20 Minutes Research
Likely Not Overfit

Figure 18: Backtesting result with original 30 stocks



$5.9M	$129,117.47	-$20.91	$127,691.53	$-20.91	25.365%	29.12 %	$29,096.46	$98,553.15
Capacity	Equity	Fees	Holdings	Net Profit	PSR	Return	Unrealized	Volume

Select Chart

○ Insight Count
○ Alpha Assets
○ Alpha
● Strategy Equity
○ Benchmark
● Assets Sales Volume
● Drawdown
● Exposure
○ Portfolio Turnover
○ Capacity

Ranking

PSR: 25.4% Rank: 30

Research Guide

78 Backtests Remai… Likely Not Overfit

1 Parameter Detected Likely Not Overfit

23 Minutes Research Likely Not Overfit

Figure 19: Backtesting result with filtered 25 stocks

The results indicated that the modified list of 20 stocks achieved a CAGR of 13.7% and a maximum drawdown of 19.5%, while the original list of 30 stocks had a CAGR of 22.7% and a maximum drawdown of 22.5%. This clearly supports the notion that having similarly performing stocks in a portfolio can be beneficial if the stocks perform well, but the risk is also high. Below is a summary of the backtesting result.

	Original 30 stocks	Filtered 20 stocks
CAGR	22.66%	13.66%
Drawdown	22.5%	19.5%
Sharpe Ratio	0.879	0.641

Table 13: Comparison of metrics between original stocks and filtered stocks

It is important to note that the risk mitigation model described here does not generate investment ideas on its own, as it does not take into account the performance of individual stocks. Rather, its purpose is to filter out highly correlated and similar stocks from a given portfolio, which are considered to carry additional risk. In practical applications, investors would need to use this model in conjunction with other strategies to construct a well-diversified portfolio. In our study, the risk mitigation model was used in combination with other models to effectively manage portfolio risk.

2.4.5. Combined portfolio

Rank	W_{NLP}	$W_{Similarity}$	W_{GRU}	**CAGR**	**Sharpe Ratio**	**Drawdown**	**list of stocks to be dropped**
1	0.1	0	0.9	24.79%	0.945	21.9%	wdc, ctra, cat
2	0.05	0	0.95	24.55%	0.941	22.1%	wdc, ctra
3	0.15	0	0.85	23.43%	0.908	21.3%	wdc, ctra, cat, vlo
4	0.05	0.05	0.9	20.58%	0.842	18.9%	wdc, ctra, mro
5	**0.1**	**0.05**	**0.85**	**17.57%**	**0.769**	**17.2%**	**wdc, ctra, cat, mro, apa**
6	0.2	0	0.8	17.88%	0.763	21.5%	wdc, ctra, cat, vlo, eog, apa
7	0	0.05	0.95	16.59%	0.725	20.3%	mro, apa
8	0.2	0.05	0.75	16.74%	0.719	22.5%	wdc, ctra, cat, vlo, eog, mro, dri
9	0.15	0.05	0.8	15.71%	0.703	20.8%	wdc, ctra, cat, vlo, mro, apa
10	0.25	0	0.75	12.75%	0.596	21.9%	wdc, ctra, cat, vlo, eog, apa, mro

Table 14: Metrics Comparison of different portfolios ranked by Sharpe Ratio

Since our aim is to discover the ideal combination portfolio with high risk-adjusted returns and minimum drawdown, we rated the grid search results by Sharpe Ratio and achieved the top 10 results out of 50 observations.

We selected the one with the fifth best Sharpe Ratio for our Combined Portfolio Strategy. We made this decision because we wanted to strike the correct mix between CAGR and Maximum Drawdown. The Combined portfolio drop 3 stocks that has the most negative 10-K sentiment computed by the NLP algorithm, and 2 stocks that are highly correlated from the Risk Mitigation Algorithm.

Our objective is to outperform the SPY while avoiding risk factors such as the MDD. It should be noted that, while the outcome of the Machine Learning-based GRU Model is already excellent diversifying by its own, diversifying with different strategies does slightly decrease the MDD by 0.4%.

3. Discussion

3.1. Multi-Factor Equity Model

Although the stock selection ability of the models are quite good and the performance of the stocks selected can beat the S&P 500, the models still have some limitations.

One limitation is that the models are tempted to predict the same stock list every month implying that the market timing of the models is not good.

Another one may be the data preprocessing. Currently, we just stack the multi features together and the models may not learn the features corresponding to that individual stock well because the tensor per month is too long with 10,000 features.

Besides, we only use 365 data points on a monthly basis. We can increase the data points through decreasing the time windows which can be weekly or even daily.

In an attempt to increase the return, we tried to predict the short stock lists which contain the stocks with the most negative next month's return and combined the long position with the short one. However, the performance was not good. It might be because (1) timing is important for short but our models only generate the same tickers all the time and (2) the long & short stock lists were generated separately and could not work well if we combined them.

3.2. Stock Selection Using NLP

The initial cash amount allocated to a portfolio can pose challenges to the execution of our algorithm. To ensure proper execution, we assume a starting cash balance of $1 million, which may be impractical for retail investors. Our algorithm seeks to optimize returns and minimize risk by ensuring every order is filled. However, even with a 2:1 leverage, which is the maximum provided by QuantConnect, low cash flow can lead to a margin call, resulting in unfilled orders or liquidation of existing positions. This can negatively impact the portfolio's performance, by reducing our holdings leading to fewer return. We can resolve this issue by developing an entry strategy that takes into account the price of each stock unit in the initial stock weighting calculation, based on the available cash balance in the portfolio. This will allow us to allocate the available cash in a way that maximizes returns and minimizes risk, by trade off some high priced volatile stocks to more low priced stable stocks. By doing so, we can effectively manage the portfolio's exposure to market risk and aim to achieve our desired risk-adjusted return.

3.3. Price Prediction by Rank-based Sentiment Analysis on the Internet

The data collection was limited but reasonable. Due to the large size of the original compressed dump file and the large volume of submissions and comments, To test for a reasonably long period of time would use up a couple TB of storage and huge processing power. Especially for comments, which are way more than submissions. It is practically impossible to analyze all of them without stronger computers. The graph ranking part was not included as the submission sentiments part is already time-consuming.

For the training and testing period. When compared to the referencing paper [8] that uses page rank and Reddit post sentiment for stock prediction, They only tested for around 4 years of data from 2016–2018, and they only tested on bitcoin and ethereum. Therefore, the experiment in this part was successful, as we have tested more updated prices and data from 2018 to 2022. Also, more assets like stocks and cryptocurrency prices were tested, which gives us new insight into whether the strategy works on stocks too. The data were constantly collected during the three months of project work.Besides, the comment file size is much larger than the submission file size. Therefore, it is quite impossible to scrape all of the 2018–2022 comments. Therefore, only 2018 data is being downloaded and tested. Therefore, the scale for the final year project was reasonable. Therefore, unfortunately, the graph-based ranking part was not tested due to this limitation.

The prediction was also accurate given the sentiment features of Reddit, which show the correlation between the site and price changes.

3.4. Risk Mitigation Using Different Similarity Indicators

In this study, three different indicators were utilized to calculate the similarity index. However, a significant challenge was determining how to weigh the different indicators in the final calculation. The three indicators were first evenly scaled and then given even weights in this study, but using different weights for each indicator would result in a different final similarity index. Another critical challenge was to test the accuracy and reliability of the calculated similarity index. Given the absence of a baseline index for comparison, a simple buy-and-hold trading strategy was used to backtest our risk mitigation model. While the results were reasonable, they only provided experimental support for the idea because there is no genuine truth value for the similarity index.

It would be difficult for future studies to resolve this issue because no single index can accurately measure the similarity between stocks in the future. However, for the similarity scores calculated using alternative methods such as using fundamental data, one could validate the scores by comparing those with the traditional methods such as correlation. Plus, extensive amount of backtesting and changing the weights of each similarity score can help resolving this issue.

3.5. Performance Comparison with S&P 500 Index

Model	**CAGR**	**Sharpe Ratio**	**Max Drawdown**
Baseline (SPY)	2.436%	0.186	24.5%
Model 1 (Multi-Factor Equity Model)	**24.15%**	**1.06**	17.6%
Model 2 (NLP)	14.1%	0.7	19.8%
Model 3 (Removing 10 similar stocks)	13.66%	0.641	19.5%
Model 4 (Combined Portfolio)	17.57%	0.769	**17.2%**

Table 15: Comparison of different models with S&P500 Index

The purpose of this research was to develop a portfolio that uses ensemble machine learning to provide greater returns with lower risk. This portfolio also argues for the advantages of active investing over passive investing. In this research, we select SPY (S&P 500 index) as our benchmark. We have met this aim because Models 1, 2 and 3 have greater CAGRs, Sharpe Ratios, and lower MDDs than the Baseline Model. Our findings imply that various machine learning approaches can help a trading strategy outperform SPY.

We were able to merge all three methods into Model 4, which has the lowest Max Drawdown. Model 4 weights comprise 85% in Model 1, 10% in Model 2, and 5% in Model 3. Although Model 4's yearly return is lower than Model 1, but overall it has higher yearly return than that of the SPY, and outperform its Maximum Drawdown by 7.3 points.

For all of these reasons, we feel we met our aim of outperforming the SPY by integrating Machine Learning models and traditional investment techniques.

4. Conclusion

4.1. Summary of Achievements

4.1.1. Multi-Factor Equity Model

We have employed both LSTM and GRU models trained with multi-feature datasets including both fundamental and technical data to do the U.S. stock selection tasks. Our best model is GRU trained with both technical and fundamental data, period 1993-2020, n_past = 9 and Shuffle = True which can achieve sharp ratio 1.06, CAGR 24.15% and Max Drawdown 17.6%. Compared with S&P500 in the same testing period, the model can have an excessive CAGR with 21.08%, which is aligned with our objectives.

4.1.2. Natural Language Processing

We have utilized Natural Language Processing (NLP) to analyze the 10-K reports of various companies with the aim of identifying stocks that are likely to perform below the U.S. stock market index in the upcoming year. Additionally, the team used sentiment similarity scores to long S&P500 stocks. This was done to enhance the strategy's return and minimize the Max Drawdown when compared to a passive strategy that involves buying and holding the S&P 500 index.

4.1.3 Price Prediction by Rank-based Sentiment Analysis on the Internet

The system was able to extract sentiment from social media posts and rank the sentiment based on the posts metadata. Then the sentiment features are found to be likely to predict the future price change. This shows the correlation between the features in the subreddit and the prices. The expected returns are used to create a rolling portfolio strategy that is able to yield positive returns.

4.1.4 Risk Mitigation Using Different Similarity Indicators

Stocks' similarity calculation is an essential tool for portfolio management and risk mitigation. However, there is no single indicator that works perfectly when calculating stocks' similarities. This is because the calculation is based on past data, but the similarity will be used in the future. However since it is inevitable to use the past data, it is important to consider diverse indicators to obtain a better similarity index. To achieve this goal, three different indicators were integrated: DTW distance of moving averages, Euclidean distance of five fundamental ratios, and correlation. By combining these indicators, a final similarity score was derived that seemed to work well. After removing the similar stocks based on the score, the absolute value of return and drawdown both decreased, indicating the effectiveness of the new similarity score.

4.1.5 Portfolio Management System

We combined three Machine Learning-based trading strategies into a Combined Portfolio with the optimal weight found in grid search to maximize risk-adjusted returns, such as Sharpe Ratio, while maintaining an annual return greater than the performance of the SPY Market Index. Aside from trading tactics, we also visualize our trading metrics using QuantConnect's visualization system. Finally, we use QuantConnect's to backtest our Combined Portfolio.

4.2. Future Work

1. Implementing Web Application for Users: Once we have created successful investment strategies, it would be helpful to have a website that allows investors to easily access and use them. By building such a site, we can reach a wider audience and encourage more people to invest in firms that generate positive social impact through their activities.
2. Covering Short/Selling Options: We can train the models that can generate both buying and selling stock lists to achieve better return.

3. Increasing training sets: Currently, we preprocess our data on a monthly basis. Since 359 monthly data points may be too small to train, we can also preprocess the data on a weekly or even daily basis.
4. Improving the market timing ability of the multi-factor model: Currently, the market timing ability of the model is quite bad. In the future, we can come up with some technique to solve that problem.
5. Utilizing New Machine Learning Techniques: While our study introduced various machine learning techniques using different sets of data, the field of machine learning is constantly evolving, and there are many alternative data sources available today. For example, stock chart images, weather data, and satellite images could be integrated into machine learning models to improve stock performance estimation. In addition, recent machine learning techniques such as generative AIs could be explored for generating trading ideas. Future research should investigate how these new techniques and data sources could be used to enhance portfolio allocation decisions.
6. Incorporating Multiple Asset Classes: Our study only used stocks listed in the United States for portfolio allocation. However, there are many other asset classes available, including foreign exchange, credits, commodities, and swaps. By incorporating multiple asset classes into a portfolio, we can achieve a more diversified and potentially higher performing portfolio. Future research should explore how to best incorporate these asset classes into portfolio allocation models to maximize performance while considering risk management.

5. References

[1] S. A. Ross, “The arbitrage theory of capital asset pricing,” *J. Econ. Theory*, vol. 13, no. 3, pp. 341–360, 1976. Available: https://doi.org/10.1016/0022-0531(76)90046-6. [Accessed: 16-Sep-2022].

[2] E. F. Fama and K. R. French, “The cross-section of expected stock returns,” *J. Finance*, vol. 47, no. 2, pp. 427–465, 1992. Available: https://doi.org/10.1111/j.1540-6261.1992.tb04398.x. [Accessed: 16-Sep-2022].

[3] Q. Feng, X. Sun, J. Hao, and J. Li, “Predictability dynamics of multifactor-influenced installed capacity: A perspective of country clustering,” *Energy (Oxf.)*, vol. 214, no. 118831, p. 118831, 2021. [Accessed: 16-Sep-2022].

[4] S. Sugitomo and M. Shotaro, “Fundamental factor models using machine learning,” *SSRN Electron. J.*, 2018. Available at SSRN: https://ssrn.com/abstract=3322187 or http://dx.doi.org/10.2139/ssrn.3322187. [Accessed: 16-Sep-2022].

[5] K. Sai and J. Lakshminarayanan, “A comparative study of SVM and LSTM deep learning algorithms for stock market prediction,” *Ceur-ws.org*. [Online]. Available: http://ceur-ws.org/Vol-2563/aics_41.pdf. [Accessed: 16-Sep-2022].

[6] Lawi A. Implementation of Long Short-Term Memory and Gated Recurrent Units on Grouped Time-Series Data to Predict Stock Prices Accurately, 2021. Available: https://doi.org/10.21203/rs.3.rs-1057875/v1. [Accessed: 16-Sep-2022].

[7] A. Lawi, H. Mesra, and S. Amir, “Implementation of Long Short-Term Memory and gated recurrent units on grouped time-series data to predict stock prices accurately,” *Research Square*, 2021. Available:https://doi.org/10.3390/math10040566. [Accessed: 16-Sep-2022].

[8] S. Wooley, A. Edmonds, A. Bagavathi, and S. Krishnan, “Extracting cryptocurrency price movements from the reddit network sentiment,” *18th IEEE International Conference On Machine Learning And Applications (ICMLA)*, 2019. [Accessed: 16-Sep-2022].

[9] K. Ngan and D. Rossiter, “Popularity-based trading strategy from reddit posts,” *Edu.hk*, 2022. [Online]. Available: https://cse.hkust.edu.hk/~rossiter/independent_studies_projects/reddit_trading/reddit_trading.pdf. [Accessed: 11-Sep-2022].

[10] J. Abraham, D. Higdon, J. Nelson, J. Ibarra, and J. Nelson, “Cryptocurrency price prediction using tweet volumes and cryptocurrency price prediction using tweet volumes and sentiment analysis sentiment analysis,” *Smu.edu*, 2018. [Online]. Available: https://scholar.smu.edu/cgi/viewcontent.cgi?article=1039&context=datasciencereview . [Accessed: 11-Sep-2022].

[11] T. Mayor, “Why finance is deploying natural language processing,” *MIT Sloan*, 2020. [Online]. Available: https://mitsloan.mit.edu/ideas-made-to-matter/why-finance-deploying-natural-language-processing. [Accessed: 16-Sep-2022].

[12] A. Stotz, “Are financial analysts’ earnings forecasts accurate?,” *Become a Better Investor*, 2016. [Online]. Available: https://becomeabetterinvestor.net/are-financial-analysts-earnings-forecasts-accurate/. [Accessed: 16-Sep-2022].

[13] F. Zhao, Y. Gao, X. Li, Z. An, X. Ge, C. Zhang, “A similarity measurement for time series and its application to the stock market,” *Expert Systems with Applications*, 182, 2021. [Online]. Available: https://doi.org/10.1016/j.eswa.2021.115217. [Accessed: 03-Apr-2022].

[14] C. Lee, M. J. UY, D. Rossiter, “Using machine learning and algorithmic trading to beat the U.s. stock market index,” *hkust.edu.hk*, 2021. [Online]. Available:

https://cse.hkust.edu.hk/~rossiter/fyp/RO4_FYP_final_report_202021.pdf. [Accessed: 16-Sep-2022].

[15] Git. (2023). Git. Accessed: Feb. 14, 2023. [Online]. Available: https://git-scm.com/

[16] Miniconda. (2023). Miniconda. Accessed: Feb. 14, 2023. [Online]. Available: https://docs.conda.io/en/latest/miniconda.html

[17] FinBERT. (2023). FinBERT. Accessed: Feb. 14, 2023. [Online]. Available: https://finbert.ai/

[18] QuantConnect. (2023). Design and trade algorithmic trading strategies in a web browser, with free financial data, cloud backtesting and capital - QuantConnect.com. Accessed: Feb. 14, 2023. [Online]. Available: https://www.quantconnect.com/

[19] Pandas. (2023). pandas. Accessed: Feb. 14, 2023. [Online]. Available: https://pandas.pydata.org/

[20] Numpy. (2023). NumPy. Accessed: Feb. 14, 2023. [Online]. Available: https://numpy.org/

[21] Python. (2023). Python. Accessed: Feb. 14, 2023. [Online]. Available: https://www.python.org/

[22] Tensorflow. (2023). Tensorflow. Accessed: Feb. 14, 2023. [Online]. Available: https://www.tensorflow.org/

[23] Keras. (2023). Simple. Flexible. Powerful. Accessed: Feb. 14, 2023. [Online]. Available: https://keras.io/

[24] VSCode. (2023). Visual Studio Code - Code Editing. Redefined. Accessed: Feb. 14, 2023. [Online]. Available: https://code.visualstudio.com/

[25] TA-Lib. (2023). Lib : Technical Analysis Library. Accessed: Feb. 14, 2023. [Online]. Available: https://www.ta-lib.org/

[26] yfinance. (2023). yfinance. Accessed: Feb. 14, 2023. [Online]. Available: https://pypi.org/project/yfinance/

[27] pytrends. (2023). pytrends. Accessed: Feb. 14, 2023. [Online]. Available: https://pypi.org/project/pytrends/

[28] praw. (2023). The Python Reddit API Wrapper. Accessed: Feb. 14, 2023. [Online]. Available: https://praw.readthedocs.io/en/stable/

[29] tweepy. (2023). Tweepy. Accessed: Feb. 14, 2023. [Online]. Available: https://www.tweepy.org/

[30] Pushshift. (2023). Pushshift. Accessed: Feb. 14, 2023. [Online]. Available: https://pushshift.io/

[31] NetworkX. (2023). NetworkX. Accessed: Feb. 14, 2023. [Online]. Available: https://networkx.org/

[32] Financial Modeling Prep. (2022). Financial Modeling Prep API. [Online]. Available: https://site.financialmodelingprep.com/developer

[33] fundamentalanalysis. (2023). fundamentalanalysis. [Online]. Available: https://pypi.org/project/fundamentalanalysis/

[34] ps_reddit_tool (2022). ps_reddit_tool. [Online]. Available: https://github.com/magnusnissel/ps_reddit_tool

6. Appendix A: Meeting Minutes

6.1. Minutes of the 1st Project Meeting

1. Arrangement
    - Date: 2022-04-23
    - Time: 17:00-18:00
    - Place: Zoom
    - Present: All team members
    - Absent: None
    - Recorder: Hangsun
2. Discussion items
    - Finalized the priority project title
    - Discussed some direction on which market and strategy to use
    - Discussed on whether we should focus more on research (strategy and backtesting) or building a tangible platform that the users can use.
    - Suggested to send an email to professor to narrow down the topic.
3. To do list
    - Send an email to supervisor (Dr.Rossiter) as we will be proposing our own topic related to "Trading Systems for Financial Gain"
    - Research more on which strategy and asset we will use
4. Next meeting
    - The next meeting will be after we contact the supervisor

6.2. Minutes of the 2nd Project Meeting

5. Arrangement
    - Date: 2022-07-11
    - Time: 21:00-22:30
    - Place: Zoom
    - Present: All team members
    - Absent: None
    - Recorder: Ferdy
6. Discussion items
    - Data sources
    - Useful indicators and quant
    - Baseline benchmark
    - Backtesting tools
    - Our strategies
        i. Creditable financial report text mining
        ii. Social media pagerank sentiment
        iii. Multi factor model
        iv. Reinforcement learning
    - Combine strategies by portfolio optimization
7. To do list
    - Continue to read research papers on our strategies
8. Next meeting
    - The next meeting will be 8pm on August 22, 2022 via Zoom

6.3. Minutes of the 3rd Project Meeting

1. Arrangement
    - Date: 2022-08-22
    - Time: 20:00-21:00
    - Place: Zoom
    - Present: all and professor
    - Absent: none
    - Recorder: TAM Ching Lung
2. Discussion items
    - Introduction of our strategies to professor
    - Data
        - Should mention the data source
        - Can try looking at different sources like HKUST Bloomberg terminal, TradingView, business school, interactive broker
    - Design
        - Suggest system strategy to fit different assets
    - Testing
        - Compare with benchmarks like risk-free rate and all cash
        - Better presentation of backtesting metrics and comparison
        - Test with custom random data
    - Suggested ug, msc, fyp projects paper by professor
    - Budget 3000 shared between 3 fyp groups at 1k per group
    - Learning from previous groups
        - Too ambitious objectives
        - Time management
        - Meeting and making progress every week
        - Start early
        - Produce original content
        - Well defined task distribution
        - Reference negative knowledge

3. To do list
    - Read professor suggested paper readings
    - Start planning the work division of writing the proposal report
4. Next meeting
    - The next meeting will be 10pm on September 3, 2022 via Zoom

6.4. Minutes of the 4th Project Meeting

- Arrangement
    - Date: 2022-09-03
    - Time: 22:00-23:00
    - Place: Zoom
    - Present: all team members
    - Absent: none
    - Recorder: Yang Wenting
- Discussion items
    - Distribute work write proposal
    - Confirm our directions and discuss our findings
        - First focused on one asset class and then we explore others
        - For Multi-factor model, we can have a discussion with other groups with similar topics
- To do list
    - Hangsun - Overview, 1st minute, pairs-trading & portfolio theory
    - Wendy - Literature review, 4th minute, Multi-factor model
    - Ferdy - Hardware and Software, 2nd minute, NLP and System & application design
    - Max - objective, 3rd minute, graph popularity-based sentiment analysis
    - Shared - Project planning, Methodology
    - Personal DDL for proposal: Sep 11
- Next meeting
    - The next meeting will be 8pm on September 12, 2022 via Zoom

6.5. Minutes of the 5th Project Meeting

1. Arrangement
    - Date: 2022-09-03
    - Time: 22:00-23:00
    - Place: Zoom
    - Present: all team members
    - Absent: none
    - Recorder: Yang Wenting
2. Discussion items
    - Report layout and formatting
    - More details on Implementation section
    - Possible alternative algorithm for current existing one
3. To do list
    - Use agreed referencing style
    - Research on fine grained details for each algorithm to enhance the implementation section
    - Reduce the number of page
4. Next meeting
    - ○ The next meeting will be 8pm on 10 January, 2023 via Zoom

6.6. Minutes of the 6th Project Meeting

1. Arrangement
    - Date: 2023-01-10
    - Time: 22:00-23:00
    - Place: Zoom
    - Present: all team members, Communication Tutor
    - Absent: none
    - Recorder: Yang Wenting
2. Discussion items
    - Tenses on report
    - Possible alternative market compared to stock
    - Dataset citation
    - GANTT chart
    - Project Title
3. To do list
    - Update works progress on gantt chart
    - Use appropriate tenses
    - Cite tools and dataset used
4. Next meeting
    - ○ The next meeting will be 8pm on 15 February, 2023 via Zoom

6.7. Minutes of the 7th Project Meeting

1. Arrangement
    - Date: 2023-03-01
    - Time: 15:00-16:00
    - Place: Zoom
    - Present: All team members
    - Absent: None
    - Recorder: Ferdy
2. Discussion items
    - Discussed different trading strategies to be used in the project.
    - Hangsun proposed using fundamental analysis instead of time series analysis, and using time series clustering to group ticker symbols with similar price patterns. He suggested trading the fundamental data separately on a quarterly or yearly basis, and using risk quantification and covariance matrix to manage risk.
    - Wenting raised concerns about stock selection and suggested incorporating Hangsun's risk quantification methodology into the stock selection process.
    - Max proposed using subreddit ranking and sentiment analysis to influence stock selection, and suggested working with Wenting to combine their strategies.
3. To do list
    - Further research on the proposed strategies and refine them for implementation in the project.
    - Explore ways to combine the different strategies to create a comprehensive trading system.
    - Consider additional factors to incorporate into stock selection, such as news sentiment, market trends, and sector performance.
4. Next meeting
    - The next meeting will be scheduled after additional research and strategy refinement has been completed.

6.8. Minutes of the 8th Project Meeting

1. Arrangement
    - Date: 2023-04-01
    - Time: 14:00-15:00
    - Place: Zoom
    - Present: All team members
    - Absent: None
    - Recorder: Ferdy
2. Discussion items
    - Each team member gave an update on their progress with the project.
    - Wendy updated the team on changes to the stock selection model.
    - Max updated the team on the output of his model, which is a mixture of the number of comments and sentiment score. He also discussed plans to change the rank part of the model.
    - Ferdy discussed using the FinBERT pre-trained model to output a score ranging from -1 (negative) to 1 (positive).
    - Hangsun presented his work on using cointegration test and clustering to identify similarities between correlated stocks, with the goal of minimizing similarities to diversify the portfolio. He suggested running the selected stocks through the team's metrics to further refine the portfolio.
    - The team discussed the possibility of adding columnar features to incorporate trend data into the selection process. The team also discussed their objectives, which include creating a baseline, adding improvement for trend data, and implementing weighting optimization theory to give optimal weights for each stock on a monthly basis.
    - The team discussed final screening methods, including selecting a top k number of stocks or adding columnar data for Max's model. They also discussed using the Markowitz model for portfolio optimization to determine the optimal weights for each stock.

3. To-do list
    - Further refine the stock selection model and incorporate trend data.
    - Implement Hangsun's clustering approach and run the selected stocks through the team's metrics.
    - Use the Markowitz model for portfolio optimization to determine the optimal weights for each stock.
    - Conduct backtesting to visualize and evaluate the performance of the selected portfolio.
4. Next meeting
    - The next meeting will be scheduled after additional progress has been made on the project, and the team will discuss the results of the backtesting and any necessary adjustments to the model.7. Appendix B: List of Figures

6.8. Minutes of the 9th Project Meeting

1. Arrangement
    - Date: 2023-04-19
    - Time: 20:00 - 21:00
    - Place: Zoom
    - Present: All team members
    - Absent: None
    - Recorder: Hangsun
2. Discussion items
    - Each team member gave final update on the project and the report.
    - We discussed on how to finalize the final report, especially on the newly added parts like conclusion and discussion.
    -
3. To-do list
    - Complete the final report separately on each member’s models
    - Add future works in the conclusion section
    - Finish writing the abstract
    - Change the formats of the reports, including figures and cations, in-text citations, gantt chart, and the order of the figures.
    -
4. Next meeting
    - This is the last meeting before submitting the final report.

7. Appendix B: Project Planning

7.1. Distribution of Work

Task	Hangsun	Max	Wendy	Ferdy
Do the Literature Survey	●	●	●	●
Find Data Sources	●	●	●	●
Data Preprocessing	●	●	●	●
Design Risk Mitigation with Similarity	●	○	○	○
Collect Technical Data and Fundamental Data	●	○	●	○
Calculate Similarity Scores of Stocks	●	○	○	○
Run Risk Mitigation Model	●	○	○	○
Remove Similar Stocks from Portfolio	●	○	○	○
Design the Rank-based Sentiment Strategy	○	●	○	○
Collect Social Media Data	○	●	○	○
Develop Baselines	●	●	●	●
Develop PageRank Mechanism	○	●	○	○
Train Sentiment and Price Models	○	●	○	○
Test the Rank-based Sentiment Strategy	○	●	○	○
Design the Multi-factor Stock Selection Model	○	○	●	○
Train Multi-factor Stock Selection Model	○	○	●	○
Test Trading Strategies on Stock Data	○	○	●	○
Collect SEC 10-K Reports	○	○	○	●
Develop Similarity Measure between 10K	○	○	○	●
Fine Tune FinBERT model	○	○	○	●
Test the Platform	○	○	○	●
Write the Reports	●	●	●	●
Work on Project Video	●	●	●	●
Prepare the Final Presentation	●	●	●	●

Design the Project Poster	○	○	●	●

7.2. GANTT Chart

Task	Jul	Aug	Sep	Oct	Nov	Dec	Jan	Feb	Mar	Apr
Do the Literature Survey	■	■	■							
Find Data Source		■	■	■	■					
Data Preprocessing				■	■	■	■			
Design Pairs Trading using Clustering					■	■	■			
Collect Technical Data and Fundamental Data						■	■	■		
Calculate Similarity Scores of Stocks								■	■	
Run Risk Mitigation Model								■	■	■
Remove Similar Stocks from Portfolio								■	■	■
Design the Rank-based Sentiment Strategy			■	■						
Collect Social Media Data			■	■	■	■	■	■	■	■
Develop PageRank Mechanism						■	■	■		
Train Sentiment and Price Models							■	■		
Test the Rank-based Sentiment Strategy							■	■	■	■
Design the Stock Algorithm					■	■	■	■		
Train Multi-factor Stock Selection Model					■	■	■	■	■	■
Test Trading Strategies on Stock Data								■	■	■
Collect SEC 10-K Reports			■	■	■					
Develop Similarity Measure between 10K					■	■	■			
Fine Tune FinBERT model							■	■	■	
Test the Platform										■
Write the Reports										■
Work on Project Video										■
Prepare the Final Presentation										■
Design the Project Poster										■

8. Appendix C: Required Hardware & Software

8.1. Hardware

Item	Specification (Minimum)
RAM	8GB
HDD	128 GB
Processor	4 x 1.6 GHz CPU
GPU	2x Nvidia RTX 2080Ti
Web Server	2 x 1.6 GHz CPU, 3.5 GB RAM, 40GB HDD

8.2. Software

Item	Version	Specification
Development OS	MacOS Catalina 10.15.6	Environment for development
Git [15]	2.23.0 or after	Version control
Miniconda [16]	4.8.4 or after	Package control
Finbert [17]	ProsusAI	base model for NLP fine-tuning
QuantConnect [18]	4.0.1 or after	Backtesting platform
Pandas [19]	1.5 or after	Data analysis library

Numpy [20]	1.24 or after	Math library
Python [21]	3.9 or after	Programming language
Tensorflow [22]	2.0 or after	Machine learning library
Keras [23]	2.10 or after	Machine learning library
VSCode [24]	latest	IDE
ta-lib [25]	0.4.25	Technical Analysis Library
yfinance [26]	0.2.11	Yahoo Finance library
pytrends [27]	4.9.0	Google Trends library
praw [28]	7.6.1	Reddit posts library
tweepy [29]	4.12.1	Twitter tweets library
Pushshift [30]	2023	Reddit archive posts platform
networkx [31]	3 or after	Graph library
Financial Modeling Prep API [32]	2022	Financial Statements API
fundamentalanalysis [33]	0.2.14	Stock fundamentals
ps_reddit_tool [34]	latest	Reddit Dump Extraction Tool