RO4

Using Machine Learning and Algorithmic Trading to Beat the U.S. Stock Market Index

by

Chih-yu LEE and Matthew Johann Siy UY

RO4

Advised by

Prof. David Paul ROSSITER

Submitted in partial fulfillment

of the requirements for COMP 4981

in the

Department of Computer Science and Engineering

The Hong Kong University of Science and Technology

2020-2021

Date of submission: April 14, 2021

Abstract

The project aims to develop investment strategies that can achieve higher returns with lower risks than the S&P 500 Index by applying Machine Learning techniques to traditional investment strategies. In total, we implemented and backtested three Machine Learning-based investing strategies: Harry Browne Permanent Portfolio with LSTM (HBLSTM), Stock Selection with Natural Language Processing (SSNLP), and Pairs Trading with PCA and Clustering (PTPC). From December 2004 to March 2021, the S&P 500 Index's annual return was 9.85% with a Maximum Drawdown (MDD) of 55.1%. On the other hand, two of our Machine Learning-based investment strategies, HBLSTM and SSNLP, were able to achieve both a significantly higher annual return and a lower MDD. HBLSTM had an annual return of 20.4% with an MDD of 18%, while SSNLP had an annual return of 14.8% and an MDD of 33.4%. Although PTPC merely had an annual return of 2.97% with an MDD of 31.6%, we noticed that its role is insurance that generates positive return over time as it performed well during the crises. Finally, our Combined Portfolio had an annual return of 9.99% with an MDD of 7.8%, providing further evidence of the benefits of using Machine Learning techniques in investing strategies.

Table of Contents

List of Tables

List of Figures

1 Introduction

1.1 Overview

1.1.1 Algorithmic Trading (Algo Trading)

Algorithmic trading, also known as automated trading, is a method of executing orders that allows traders to establish specific rules for entering and exiting a trade based on variables such as time, price, and volume that, once programmed, can be automatically executed by a computer. The advantage of algo trading compared to manual trading involves improving order entry speed, minimizing irrational decisions, backtesting strategies, and preserving discipline. Because of its advantages, algorithmic trading has seen increasing popularity among different market participants.

Among buy-side institutional investors, like mutual funds and pension funds, algorithmic trading is often used to spread out the execution of large orders to avoid influencing prices or to execute trades as quickly as possible or at the best possible price. Similarly, sell-side participants, such as brokerages and investment banks, often use algorithmic trading to automate trades to provide sufficient liquidity in the market. Finally, systematic-traders, like hedge funds or quant funds who often use very complex trading strategies, find it much more efficient to program their trading rules and let the program trade automatically. According to Electronic Broking Services (EBS) [3], a wholesale electronic trading platform, 70% of orders on its platform now originate from algorithms, a large contrast from 2004, when all trading was manual. Additionally, a study in 2016 showed that 80% of transactions in the foreign exchange (Forex) Market are automated [4]. These examples highlight the rise of algorithmic trading throughout the entire financial industry.

1.1.2 Algorithmic Trading Strategies

Any strategy for algorithmic trading is about identifying unexplored opportunities that is not only profitable but provides a satisfactory risk-to-return ratio. Some common strategies involve:

1. Trend-following Strategies
2. Arbitrage Opportunities
3. Index Fund Rebalancing
4. Statistical Model-based
5. Mean Reversion
6. Volume-weighted Average Price (VWAP)

The common thread between all these different trading strategies is that they involve patterns, trends, and knowledge that humans have learned about trading securities. However, the problem with using identified strategies is that over time the alpha (See Section 7: Appendix B: Glossary) or return generated from these trading strategies declines because the market learns to adjust for them. Because of this, there is a constant demand for new trading ideas and strategies. Therefore, in this project, we developed some new algorithmic trading strategies.

1.1.3 Backtesting and Optimization

Backtesting focuses on testing and validating the automated trading strategy. This process includes checking the code to make sure that it works as expected. Additionally, it also involves analyzing how the strategy performs over different market conditions and time frames, especially during black-swan events, such as the coronavirus outbreak this year.

Backtesting is an essential part of developing a viable algorithmic trading strategy, because it allows the developer to understand whether their trading strategy is scalable to different market conditions. The underlying logic behind this is that if a strategy was able to consistently work over a long period of time, then it is likely to continue to work well in the future, assuming that the strategy doesn't get revealed to the rest of the market.

Aside from this, backtesting provides developers with statistical figures that measure the overall risk-adjusted performance of their trading strategy. These metrics enable them to compare the performance of their strategy against others. Risk-adjusted performance is used, because the more risk is taken, the higher the returns that can be gained. Thus, in order to compare the performance of one strategy against another, it is critical to take into account the amount of risk each trading strategy had.

Key Performance Indicators in Backtesting

In our project, we focused on two key performance indicators for backtesting.

1. Sharpe Ratio: This ratio is the average return earned in excess of the risk-free rate per unit of volatility or total risk. (See Section 7: Appendix B: Glossary for formula)

2. MAR Ratio: This is the ratio of annualized return to the Max Drawdown, the maximum observed loss from the beginning of the backtesting. This ratio, by its reciprocal, reflects the time it takes to recover from the Max Drawdown on average. (See Section 7: Appendix B: Glossary for formula)

Avoid Overfitting

Backtesting enables developers to maximize the risk-adjusted performance of their trading strategy. However, it is crucial to avoid overfitting the trading strategy to the historical data; otherwise, it would become useless for future trading purposes.

1.1.4 Live Execution

After backtesting our trading strategy, trading with real money was the next step. Our ultimate goal for creating an algorithmic trading strategy is to make money by trading in the financial markets. For this stage of the process, it was important to select an appropriate broker (See Section 10.1: Appendix E: Additional Information for details) and to implement mechanisms to manage both market and operational risks.

These issues include selecting an appropriate broker and implementing mechanisms to manage both market risks and operational risks, such as potential hackers and technology downtime.

The importance of managing operational risks cannot be understated. For example, one of the largest algorithmic trading firms, Knight Capital Group, lost hundreds of millions of dollars and was eventually acquired, because one of their software developers forgot to copy a specific program onto one of their main servers [5].

1.1.5 Types of Investments

- **Passive Investment**

  Passive Investment is also known as the buy-and-hold strategy because, it involves investing in assets like stocks over a long period of time. The prime example of passive investing involves buying an index fund that tracks the price movement of an index, which is a group or collection of stocks.

  - **S&P 500 Index**

    An example of the most commonly followed equity indices is the S&P 500. It tracks the stock performance of top 500 large companies listed on stock exchanges in the United States.

  - **Exchange Traded Fund**

    An exchange traded fund (ETF) is a basket of securities (e.g., stocks, bonds) that usually tracks an underlying index. ETFs are very similar to index funds, but unlike index funds, ETFs are listed and traded on exchanges.

    The most well-known example is the SPDR S&P 500 ETF (SPY), which tracks the S&P 500 Index. ETFs can contain many types of assets, including stocks, commodities, bonds, or a mixture of asset types.

  - **Harry Browne Permanent Portfolio**

    The permanent portfolio is an investment portfolio invented by Harry Browne in 1980s [6]. It comprises four components: Stocks, Bonds, Gold, and Cash. It is designed to thrive over four major macroeconomic periods: Prosperity, Deflation, Inflation, and Recession. In this project, we use the following four ETFs for tracking the trends of the components in the portfolio:

    1. SPDR S&P 500 ETF Trust (SPY): This is one of the most liquid ETFs on U.S. exchanges, incepted in 1993. According to the information provided by ETF.com [7], its average dollar volume is around $36.4 Billion. Additionally, its bid-ask spread is about $0.01 (0.00%), which is relatively low compared to other S&P 500 Index ETFs.
    2. iShares 20+ Year Treasury Bond ETF (TLT): It is a fund incepted in 2002 that approximates the return of the long term U.S. Treasury bonds by tracking the Barclays Capital U.S. 20+ Year Treasury Bond Index. In addition, its average dollar volume is around $1.7 Billion, and its bid-ask spread is about $0.01 (0.01%).
    3. SPDR Gold Shares (GLD): It is the largest ETF to invest directly in physical gold and was incepted in 2004. It is intended to offer investors a means of participating in the gold bullion market without the necessity of taking physical delivery of gold, by

simply trading a security on a regulated stock exchange [8]. Its value is determined by the LBMA PM Gold Price, so it has a close relationship with Gold spot prices. Moreover, its average dollar volume is around $2.0 Billion, and its bid-ask spread is about $0.01 (0.01%). It is traded as SPDR Gold Trust (2840.HK) in Hong Kong.

4. iShares 1-3 Year Treasury Bond ETF (SHY): It represents the "Cash" component in the originally proposed portfolio, since it can earn some profits during normal periods while hedging against risk during a recession. It is an ETF incepted in 2002 that tracks the ICE U.S. Treasury 1-3 Year Bond Index, a market-weighted index of debt issued by the U.S. Treasury with 1-3 years remaining to maturity. Furthermore, its average dollar volume is around $0.5 Billion, and its bid-ask spread is $0.01 (0.01%).

- **Active Investment**

Active Investment involves trying to beat the performance of the entire stock market by buying and selling stocks regularly based a the chosen trading strategy. The rationale behind active investment is that by taking advantage of market inefficiencies, an investor can earn a higher risk-adjusted return. The problem, however, is that too many people and firms aren't able to see the flaws in their trading strategies and don't have insider information, which is why it is so difficult for normal people to outperform the market in the long-run. Nevertheless, we believe that by applying Machine Learning to algorithmic trading, it is possible to create a trading system that can outperform the market consistently in the long-run. At least we are trying to do so on a short-term trial basis.

1.1.6 Machine Learning Applications in Trading

In the past decade, researchers have made large breakthroughs in the fields of Machine Learning and Artificial Intelligence. Artificial Intelligence (AI), at its core, is about enabling machines to carry out human tasks such as being able to recognize languages and texts.

Machine Learning (ML), on the other hand, is a sub-branch of AI that focuses on applying algorithms that teach machines how to learn by themselves and improve from experience without needing to be programmed directly to do so. Broadly speaking, Machine Learning has three main types of learning problems:

1. **Supervised Learning**

   In Supervised Learning, machines are provided with a complete, clean, and labeled data set from which it can learn from. Supervised learning normally deals with two main types of problems: classification and regression [9].

2. **Unsupervised Learning**

   Unsupervised Learning is the complete opposite of supervised learning. Instead of being given a complete labeled data set to learn from, machines try to look for patterns in unlabelled datasets that they are given to predict the output. As such, unsupervised learning is normally used for clustering data and looking for hidden features.

3. **Reinforcement Learning**

   Finally, Reinforcement Learning involves using a learning agent that learns the best way of doing something through experience gained from interacting with its environment.

One of the areas where Machine Learning is becoming increasingly popular is algorithmic trading. Traditionally, algorithmic trading is simply automating a strategy that has been developed by a human. However, the rise of Machine Learning has opened a whole new set of possibilities for the field. This is because Machine Learning can identify potential new trading patterns or strategies that humans have never considered before.

1.1.7 Deep Learning

Another area in Machine Learning that shows a lot of promise in algorithmic trading is Deep Learning. Deep Learning uses neural networks, which are algorithms modeled after the structure of the human brain.

One example of a neural network is a Recurrent Neural Networks (RNN). As its name suggests, it applies the same function for all input data [1]. Additionally, an RNN has an internal memory, which allows it to take the previous output as input when making a decision. Hence, unlike other neural networks where all the inputs are independent of each other, RNNs stand out because their inputs are related. This enables RNNs to model time-series data, where each data point is dependent on previous one, like daily stock prices.

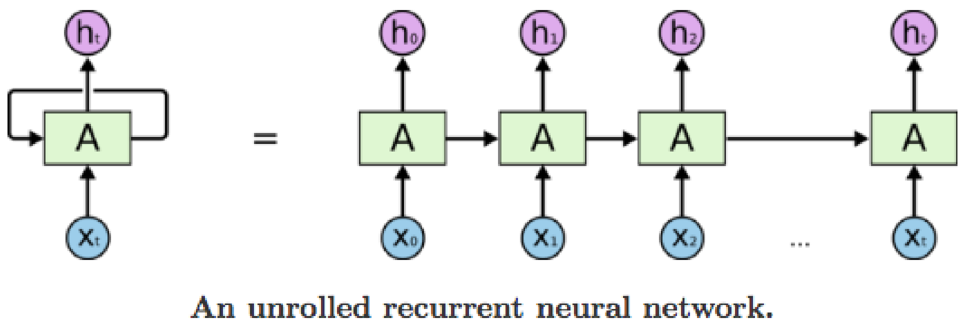


Figure 1: Illustration of an RNN [1]

Disadvantages

Unfortunately, RNNs have a major drawback, namely, the vanishing gradient problem, which simply means that the neural network is incapable of learning from the earlier layers in the network. This problem becomes worse as the number of layers in the architecture increases. As such, traditional RNNs cannot process very long series of input data.

Solution

One possible solution for the vanishing gradient problem presented in RNNs involves using Long Short-Term Memory (LSTM) networks. Long Short-Term Memory (LSTM) networks are a modified version of RNNs which resolve the vanishing gradient problem, and thereby make it easier to remember past data in the memory. For these reasons, LSTM is suitable to predict time series data given a specific time lag, like the price of a stock three days from now.

1.1.8 Natural Language Processing

One area in which AI and ML can be applied in trading involves analyzing the financial reports filed by publicly-traded companies. Traditionally, analysts from financial institutions would have to peruse hundreds of pages to forecast a company's performance in the future so they can assign a buy, hold, or sell rating to a company. Not only is this process very long and tedious, but humans are often prone to mistakes. Studies have often shown that "expert" analysts are often wrong with their forecasts.

According to a study of 45 research papers, financial analysts' estimates were off by over 25% on average from 2003 to 2014 [10]. Moreover, the study also found that the analysts' predictions were on average consistently above the actual value, which suggests that analysts are prone to optimistic bias [10]. However, with Natural Language Processing (NLP), it is possible to do sentiment analysis on the financial statements with a clearer and unbiased company outlook.

1.1.9 Clustering

Clustering is a type of Machine Learning technique used in Unsupervised Learning problems. Most commonly, it involves grouping unlabeled examples or data points together based on their different characteristics. Its main advantage lies in its ability to identify patterns that are usually hidden or difficult to identify. It is this characteristic that makes clustering especially useful in Pairs Trading.

Pairs Trading

Pairs Trading is based on the idea of finding two stocks whose price movements are correlated, and then longing (buying) one stock and then shorting (selling) the other. The rationale behind this is that if the price movements of two stocks are highly correlated then that means they usually move in the same direction. (See Section 7: Appendix B: Glossary for formula) Therefore, if one stock's price goes up, the other's price is likely to go up as well. This characteristic means that the trend is not being followed if the price of the stocks diverge over a certain period of time. As such, there is a high likelihood for the stock prices to reconverge towards one another. By longing the stock that has underperformed and shorting the stock that has overperformed, we can take advantage of the temporary break in the trend.

How Clustering is Applicable in Pairs Trading

Clustering can be used to identify pairs of stocks that are highly correlated (e.g. same industry or similar businesses). This solves the largest problem in Pairs Trading namely finding pairs of stocks that are suitable for Pairs Trading. This is because, not all pairs of stocks that are correlated are automatically suitable for pairs trading. Each correlated pair of stock needs to be tested statistically to see if they are suitable. However, this is an extremely time-consuming and resource intensive task. Given a universe of trading assets with n stocks, it would take $O(n^2)$ times to identify all suitable pairs.

1.2 Objectives

The goal of this project was to create a trading system that outperforms the U.S. stock market, which is proxied by the S&P 500 Index, and maximizes the Sharpe Ratio or risk-adjusted return. Additionally, we also did live trading using the completed trading system.

Overall, our project focused on three different aspects of Machine Learning: Natural Language Processing, Clustering, and Deep Learning. There were four stages of development:

1. **Machine Learning Trading Strategy Stage**

   This stage mainly focused on developing all the individual trading strategies with each one incorporating a different Machine Learning area. To accomplish it, we set four goals:

   (a) Implement Harry Browne's Permanent Portfolio (or other passive investment strategies) by using Deep Learning models and portfolio theories to determine the weights of the components for monthly rebalancing.

   (b) Develop trading strategies using Natural Language Processing (NLP) to analyze textual contents of financial statements for picking which stocks to buy or sell.

   (c) Conduct pairs trading using Principal Component Analysis (PCA) and Clustering (e.g. K-Means, DBSCAN) to select stock pairs.

2. **Strategy Combination Stage**

   This stage mainly focused on combining all of the individual trading strategies together into one portfolio. In order to do this, we set out one goal:

   (a) Determine the optimal weights of each strategy in the overall portfolio to maximize sharpe ratio by a grid search with step size of 0.05.

3. **Backtesting Stage**

   This stage mainly focused on backtesting the different trading strategies that were created. In order to accomplish this, we set out two goals:

   (a) Develop a backtest system to evaluate the performance of all the strategies above with visualization that includes several useful measures, such as Compounded Annual Growth Rate (CAGR), Sharpe Ratio, Max Drawdown, Equity Curve, Rolling Volatility, and so on. (See Section 7: Appendix B: Glossary for further information on important measures)

   (b) Conduct a backtest on all four trading strategies developed in Stage 1 to make sure that they perform well and work as intended.

4. **Live Trading Stage**

   This stage mainly focused on using our completed and backtested trading system to do live trading. In order to accomplish this, we set out two goals:

   (a) Execute the automated trading using the Interactive Brokers API to eliminate latency issues or fat-finger errors caused by manual trading.

   (b) Analyze and track the performance of live trading to see if functions work as intended.

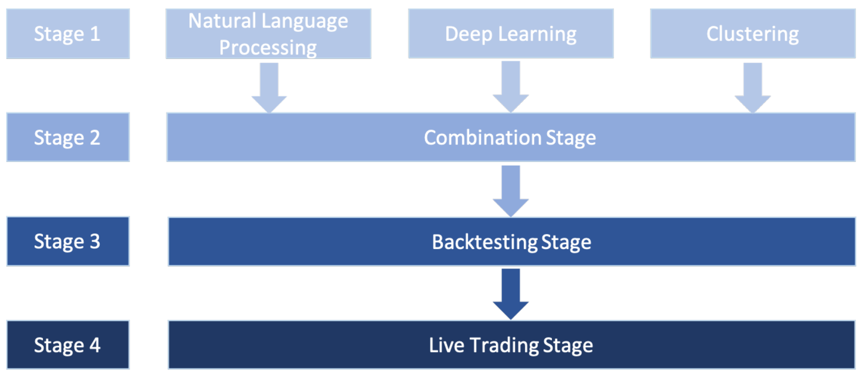


Figure 2: Illustration of 4-Stage Developments

1.3 Literature Survey

1.3.1 Harry Browne Permanent Portfolio

According to James Chen from Investopedia [6], the Harry Browne Permanent Portfolio could generate an 8.65% annualized return with a 7.20% standard deviation (or called volatility) over the forty years span from 1976 to 2016. In the meantime, the S&P 500, the major United States (U.S.) market index which tracks the performance of the top 500 companies in the U.S., had an annualized return of 8.15% with a volatility of 15.49% [11]. The statistics showcases two main characteristics of the Permanent Portfolio: Profitability and Stability.

For profitability, the Permanent Portfolio has an even higher annual growth rate than the S&P 500 Index. Although 0.50% annual difference seems to be small, it could be a gap of 465.31% after the compounding effect over 40 years. It will result in an extra 4.65 million dollar profits than the one investing in S&P 500 after the 40-year investment in Permanent Portfolio if the initial capital was set to be 1 million dollars. For stability, the volatility in the Permanent Portfolio is merely 46% of the one in S&P 500. With the log-normal model in stock price movement applicable, it implies that there will be around 10 times higher chance for S&P 500 to drop more than 7% in a single year where the probability of deviate 1 standard deviation from mean for Permanent Portfolio is 1.5% (Z-Score = 2.15) and the one for S&P 500 is 16% (Z-Score = 1). We are much more confident in investing in the Permanent Portfolio with satisfactory gain while bearing acceptable risk.

According to the Lazy Portfolio ETF [12], the Maximum Drawdown of the permanent portfolio during the coronavirus pandemic is 1.68%; on the other hand, the S&P 500 has suffered a Max Drawdown of 33.67% during the same period. Additionally, Chih-yu LEE, one of the authors, has conducted research on how to enhance the performance of the Permanent Portfolio. More detailed information of the Permanent Portfolio can be found in the report [13].

1.3.2 Long Short-Term Memory (LSTM) or Recurrent Neural Network (RNN)

In their research paper, Chariton Chalvatzisa and Dimitrios Hristu-Varsakelis [14] present a LSTM-based neural network for predicting asset prices, together with a successful trading strategy. From 2010 to 2018. their LSTM model and trading strategy was able to outperform a buy-and-hold strategy (BnH) on the S&P 500, Dow Jones Industrial Average (DJIA), NASDAQ and Russel 2000 stock indices, with cumulative returns of 340%, 185%, 371% and 360% respectively vs BnH returns of 166%, 164%, 338%, and 182% respectively.

1.3.3 Natural Language Processing

According to "Lazy Prices" by Lauren Cohen, Christopher J. Malloy, Quoc Nguyes [15], "changes to the language and construction of financial reports" provides a very strong signal for the firm's future performance. Their study found that from 1995 to 2014, a portfolio that shorts "changers" and buys "non-changers" earns up to 188 basis points in monthly alphas (or 22% per annum).

1.3.4 Clustering in Pairs Trading

In recent decades, there are several topics regarding the application of clustering on trading. Hongxin He, Jie Chen, Huidong Jin, and Shu-Heng Chen [16] have proposed a data mining approach with K-Means Clustering and Regression Model for trading, while Roshan Adusumilli [17] developed a pairs trading strategy with DBSCAN. In our research, we analyzed the performance from different clustering methods.

2 Methodology

2.1 Design

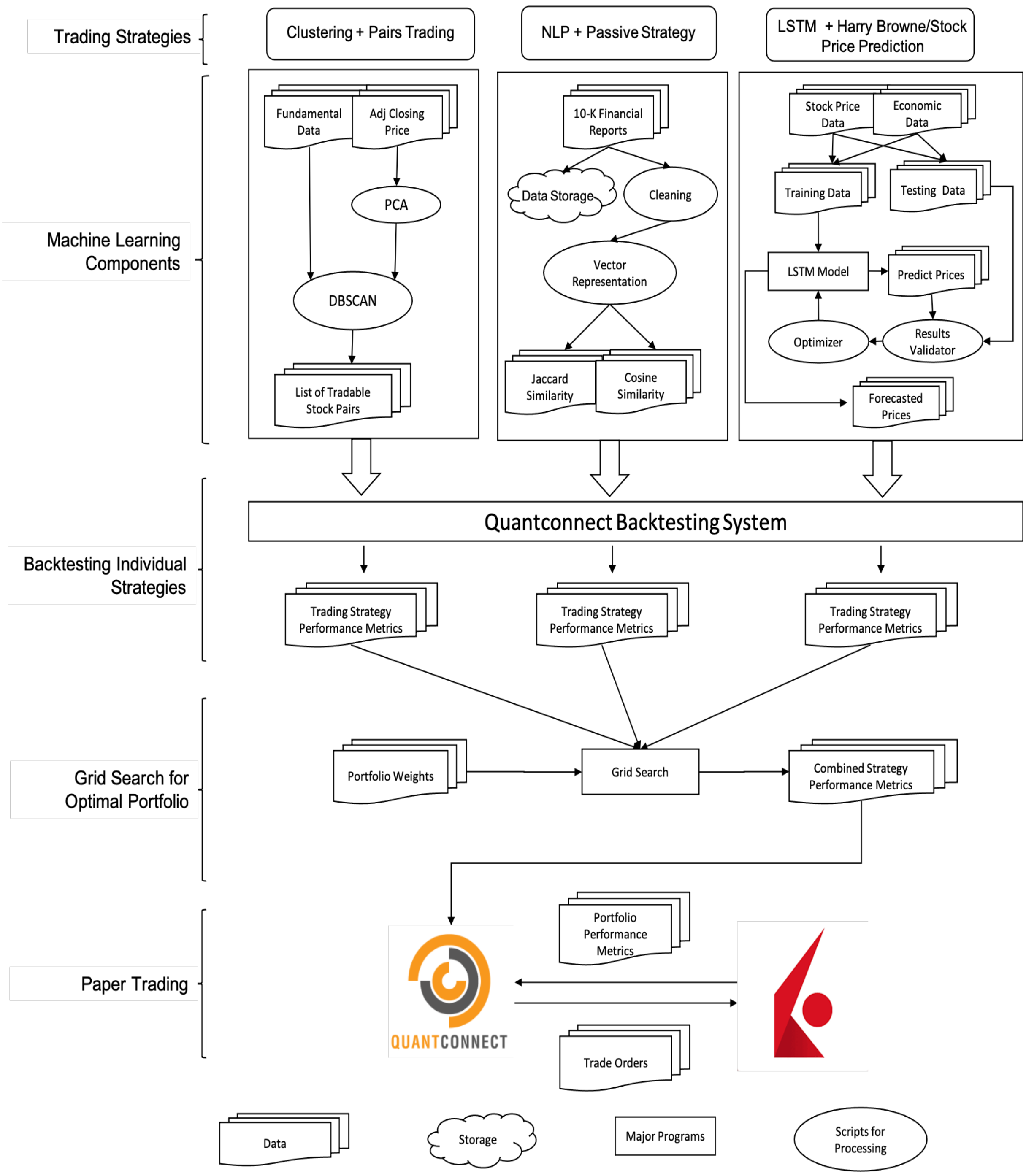


Figure 3: System Design

2.1.1 Rebalancing of Harry Browne Permanent Portfolio

As all of the Machine Learning-based strategies are categorized as active investments, we would like to invest our idle cash into the well-known passive investment strategy: Harry Browne Permanent Portfolio. It comprises Equity, Bonds, and Commodity, which diversify its exposure to different asset classes. It invests in the stock index, treasury bond, gold, and cash equally with annual rebalancing. According to some research online [18; 19; 20], we discovered that both annual rebalance frequency and equal weight allocation are not optimal in the terms of risk-adjusted return such as Sharpe Ratio and MAR Ratio. Therefore, we adopted several adjustments on the rebalance frequency and weight allocation for both the Machine Learning model-free and the Deep Learning methods.

Harry Browne Permanent Portfolio without Machine Learning

For the Machine Learning model-free approach, we changed the frequency to monthly rebalancing and allocated the weight by the volatility targeting. With the emergence of electronic trading, the pace of the current financial market is much faster than the one when the Harry Browne Permanent Portfolio was introduced in the 1980s. Hence, we increase the rebalancing frequency to monthly to strike a balance between better capturing the change in market dynamics and reducing the transaction costs.

On the other hand, volatility targeting adjusts the weight of each component according to its annualized standard deviation of daily returns (volatility or σ), and it is defined as follows:

$$w_i \propto \frac{1}{\sigma_i}, \sum_{i \in universe} w_i = 1$$

The universe is defined to be the set of four components, i.e. {SPY, TLT, GLD, SHY}, and the computation of w_i will be repeated on the first business day of each month.

However, according to Clive Granger and Robert Engle, two Nobel Memorial Prize laureates in Economics Science in 2003, a big change in price tends to be followed by another big change and vice versa [21; 22]. Therefore, we believe that the recent volatility has more representing power than the remote one. We consider the adjustment of an exponentially weighted moving average (EWMA) for the lookback period N days to replace the simple moving average one on volatility calculation. The EWMA volatility is defined as:

$$\sigma_{EWMA} = \sqrt{252}\sqrt{\frac{\sum_{i=0}^{N-1} \lambda^{N-1-i}(r_i - \bar{r})^2}{\sum_{i=0}^{N-1} \lambda^{N-1-i}}}$$

λ is a weighted coefficient between 0 and 1, r_i is the return on the day i given that the current day is day N, and $\bar{r}$ is the average return over day 0 to day N-1. The reason why we multiply $\sqrt{252}$ is to annualize the volatility. After obtaining the new σ_{EWMA}, we could apply the same formula and derive the weight of each component correspondingly.

Harry Browne Permanent Portfolio with Machine Learning

There are two possible applications of Machine Learning on passive investment: weight determination and price prediction. For the first one, we could simply obtain the weights of each component from the Deep Learning model and invest accordingly. On the other hand, for the price prediction, we would incorporate the historical price data and the macroeconomic data to optimize the accuracy of price prediction. After predicting the price by the model, we obtain its expected return and the covariance-variance matrix for each component. Then, we implement different portfolio theorems, such as Markowitz Mean-Variance Portfolio (MVP), Inverse Volatility Portfolio (IVP), Risk Parity Portfolio (RPP), and Maximum Decorrelation Portfolio (MDCP), to finalize the weight allocation for each component. According to these portfolio theories, the optimal weight will be obtained if the expected return and covariance matrix are given. Overall, we expect both Machine Learning-based strategies could beat the one that we just proposed in the previous section, as Machine Learning models have more predictive power than volatility targeting.

2.1.2 Stock Price Prediction by Neural Networks such as Recurrent Neural Networks (RNN) and Long Short-Term Model (LSTM)

The LSTM model will be trained with price and economic data with the goal of predicting the price of a stock 1 month into the future. After the price of stocks have been predicted, the following strategy will be implemented. Buy the stocks that have the highest predicted return and short the stocks that have the lowest predicted return.

Implementation of LSTM Cell

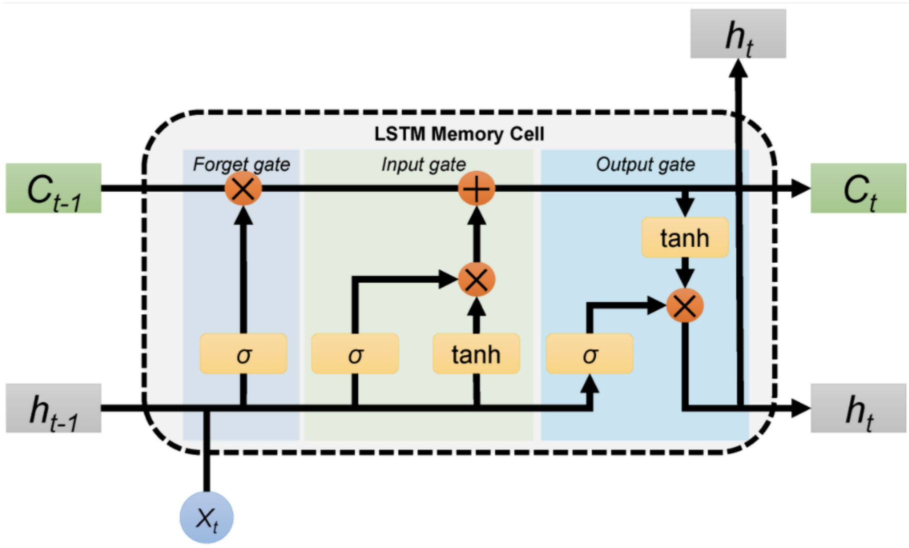


Figure 4: Illustration of LSTM Model Cells

The LSTM cell contains 3 gates: the forget gate, the input gate, and the output gate. The gates mimic the neurons in human brains by controlling the flow of information through cell. Generally, LSTM is used to make predictions that involve time series data since there can be large time gaps between important events in a time series. As mentioned in the introduction, the LSTM model resolves the vanishing gradient problem that normal recurrent neural networks (RNNs) experience. The main difference between the two is that in an LSTM block, whenever information is given as an output, they can also remain in the block's memory. This information can then be continuously fed back in each of the gates until the end of the training. This is the main reason we chose LSTM model over other neural networks. In order to predict stock price data, it is essential that the neural network is capable of remembering important information that happened a long time ago.

2.1.3 Stock Selection by Sentiment Analysis on 10-K Reports

Passive Investing Strategy Framework

Between 1995 and 2020, passive mutual funds and ETFs have attracted over $5.2 trillion dollars in cumulative net flows. In contrast, active funds only gained $1.8 trillion during the same period [23].

The rise of passive investing is driven by the belief that no one can beat the returns of the market consistently over a sufficiently long horizon. Because of this, plenty of investors prefer to simply track the performance of the S&P 500 (e.g. SPY ETF) or Russell 2000 index (e.g. IWM ETF). SPY has an asset under management (AUM) of around $316.46 billion [24], while IWM has an AUM of around $61.98 billion [25]. However, the problem with simply tracking the index is that one's portfolio has exposure to all the stocks in a particular index. This is not ideal because it means that one would be investing in some companies that are not doing well or may be going bankrupt soon. With the assistance of NLP, we can identify some bad companies and short them instead of longing them. By longing good companies and shorting bad companies, we expect to outperform the market index.

Improving Strategy Using NLP

In order to improve the traditional passive trading strategy of buying the index, we choose to analyze the 10-K reports of companies through NLP in order to determine if they should be bought or sold. To analyze the 10-K reports, we use a combination of sentiment and similarity analysis to generate trading signals.

Before we start analyzing the 10-K reports, it is essential to get a vector representation of the text files. To do this, we transform the 10-K reports using Bag-of-Words (BoW) Model and Term Frequency-Inverse Document Frequency (TF-IDF) Model.

Bag-of-Words (BoW) vs Term Frequency-Inverse Document Frequency (TF-IDF)

Suppose, you were comparing 2 movie reviews:

1. This movie was absolutely terrible, terrible, terrible.
2. This was the worst movie I've ever watched.

In both the BoW and TF-IDF models, the first step is to create a vocabulary, which contains all the words that appear in the texts that you are comparing.

After that, we can create a vector representation for each text.

vocab = ["This", "movie", "was", "absolutely", "terrible", "the", "worst", "ever", "watched", I've"]

In the BoW Model, for every word in the vocab, check if it is present in the text. If it is present, place a 1 for that word otherwise place a 0.

Table 1: Table for BoW Model

	This	movie	was	absolutely	terrible	the	worst	ever	I've	watched
Text1	1	1	1	1	1	0	0	0	0	0
Text2	1	1	1	0	0	1	1	1	1	1

In the TF-IDF Model, the steps can be broken down into 2 parts: Term Frequency and Inverse Document Frequency.

Term Frequency Calculation

Table 2: Term Frequency Table for TF-IDF Model

	This	movie	was	absolutely	terrible	the	worst	ever	I've	watched
Text1	$\frac{1}{7}$	$\frac{1}{7}$	$\frac{1}{7}$	$\frac{1}{7}$	$\frac{3}{7}$	0	0	0	0	0
Text2	$\frac{1}{8}$	$\frac{1}{8}$	$\frac{1}{8}$	0	0	$\frac{1}{8}$	$\frac{1}{8}$	$\frac{1}{8}$	$\frac{1}{8}$	$\frac{1}{8}$

Inverse Document Frequency Calculation

$$idf_t = log\left(\frac{number\ of\ documents}{number\ of\ documents\ with\ term\ 't'}\right)$$

Table 3: Inverse Document Frequency Table for TF-IDF Model

	This	movie	was	absolutely	terrible	the	worst	ever	I've	watched
tf-idf	0	0	0	0.3	0.3	0.3	0.3	0.3	0.3	0.3

TF-IDF Vector Representation

$$(tf\_idf)_{t,d} = tf_{t,d} * idf_t$$

Table 4: Table for TF-IDF Model

	This	movie	was	absolutely	terrible	the	worst	ever	I've	watched
Text1	0	0	0	$\frac{3}{70}$	$\frac{9}{70}$	0	0	0	0	0
Text2	0	0	0	0	0	$\frac{3}{80}$	$\frac{3}{80}$	$\frac{3}{80}$	$\frac{3}{80}$	$\frac{3}{80}$

Calculating Similarity Score

After getting the vector representations of each document, the next is step is find a similarity score between both documents. Some of the most popular ways for getting text similarity scores include:

1. Jaccard Similarity
2. Cosine Similarity

The Jaccard Similarity metric compares two sets to see what proportion of members are shared between both sets relative to the total number of members in both sets. The Jaccard Similarity score should range from 0 to 1. The higher the Jaccard similarity score, the more similar the two sets are.

The formula for Jaccard Similarity is:

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|}$$

Notice that BoW vector representation of text documents is equivalent to the set of all words present in the document. Hence, we can use the BoW vector representation of text documents to get the Jaccard similarity coefficient between both documents.

In contrast, the Cosine Similarity metric measures the cosine of the angle between two n-dimensional vectors projected in a multi-dimensional space. The Cosine Similarity score should range from 0 to 1. The higher the Jaccard similarity score the more similar the orientation of the two vectors are.

The Jaccard Similarity of the example above is equal to $\frac{3}{9} = 0.333$.

On the other hand, to get the Cosine Similarity scores between two text documents, the TF-IDF vector representation should be used.

The formula for the Cosine Similarity is:

$$similarity = cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\|\|\mathbf{B}\|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\sqrt{\sum_{i=1}^{n} B_i^2}}$$

The Cosine Similarity of the example above is equal to 0.

Although the results of the Jaccard Similarity and Cosine Similarity seem really bad, this is mainly because the size of the texts were extremely small. In our project, 10-K reports on average have around 42,000 words in length [26].

Integrating Sentiment Analysis and Textual Similarity

While Cosine Similarity and Jaccard Similarity provides a relatively good starting point for analyzing the similarity between text documents, they are insufficient by themselves alone. This is because they don't provide any sentiment analysis. Consider the previous example of the 2 movie reviews. Notice that while they use different words, the meaning is almost the same.

To resolve this issue, instead of simply using all the words in a text as part of the vocabulary, it would be better to use an existing financial dictionary with each word containing a specified sentiment. Luckily, there is already such a dictionary that has been created and used in plenty of different research papers and studies. This is the Loughran and McDonald's Master Dictionary. It has over 80,000 words, with flags to indicate if the word belongs to a particular sentiment class (i.e., negative, positive, uncertainty, litigious, constraining). By using this dictionary, it is possible to see the similarity of the documents based on a particular sentiment. An example of this is if a company was suddenly facing legal troubles. Then their latest 10-K report would contain significantly more litigious words than before.

Aside from this, notice that there are words in the examples like “this”, “the”, “was” and “I’ve” that don’t provide much meaning to the sentences. These words are known as “stop words”, and they should be removed before conducting the similarity tests since their presence in the text doesn’t contribute to the overall similarity.

2.1.4 Pairs Trading by Principal Component Analysis (PCA) and Clustering

Pairs Trading Framework

As pairs trading is a common practice for investors to have zero net exposure while gaining profits from the convergence of the spread of two stocks, we believe that the stock selection process could be optimized by applying the Machine Learning technology. We process the price data by PCA and incorporate it with the macroeconomic data and fundamentals for each firm. Finally, we construct Co-integration tests to determine the most optimal stock pairs for pairs trading.

Specifically, the strategy involves identifying a pair of stocks whose price movements are correlated and monitoring the "spread" between the two stocks in order to determine when to buy and sell. The spread can be defined as:

$$Spread\ S_t = Y_t - \beta X_t,$$

where Y_t the price of stock Y at time t and X_t the price of stock X at time t, and β is the hedge ratio between stock X and Y.

The hedge ratio β is normally computed by regressing the historical price of Y against the historical price of X over a predetermined period of time as follows:

$$Y_t = intercept + \beta X_t + \epsilon_t$$

Using the spread, it is possible to generate trading signals. For example, if the spread between stocks Y and X today is significantly greater than the mean of the previous daily spreads over a period of time, then it is likely that the spread will decrease in the near future, so buy stock X and sell stock Y. The advantage with pairs-trading is that it is a market-neutral strategy, which aims to avoid market-specific risk.

However, the pairs trading strategy will only work if the spread between stocks Y and X remains relatively constant over a long period of time. For this reason, correlation is not a good indicator to use for identifying pairs of stocks. Correlation only measures whether the returns of the stocks move in the same direction; however, it doesn't guarantee that the spread remains constant. An example of this is shown in Figure 5:

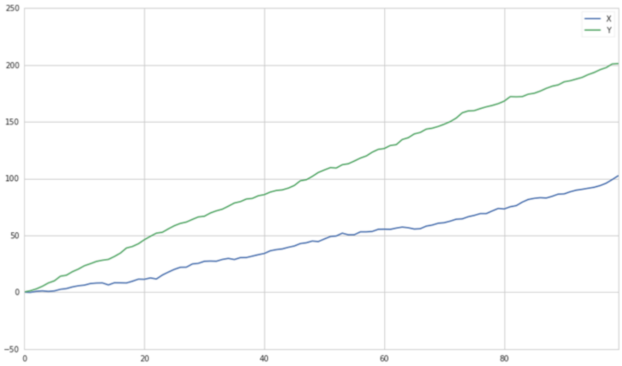


Figure 5: Example of Correlated But Not Co-Integrated Stock Pairs

Indeed, while stocks X and Y are correlated, their spread has dramatically increased over time, which means that it is not suitable for pairs trading. To guarantee that the spread between stocks X and Y remains constant, it is necessary to check whether the price history of the two stocks are co-integrated (See Section 7: Appendix B: Glossary for formula). The frequency of re-evaluating the Co-integration relationship is later determined as every 6 months since it yielded the best backtesting result.

Main Problems with Pairs Trading

The problem with pairs trading is finding suitable pairs of stocks. If every pair of stocks in the stock market is tested for co-integration, thousands of pairs can be founded to be co-integrated. However, just because a pair of stocks is co-integrated, it doesn't mean that they are suitable for pairs trading. This is because the companies that the stocks represent may be completely unrelated and different from one another. Moreover, as there are around 1500 U.S. stocks in the trading universe, the total number of pairs is O(n^2), which is over 1 million possible pairs. As speed is an important factor for earning profits in the financial market, evaluating millions of pairs will not be feasible in practice.

The only way to determine suitability is to backtest trading several pair of stocks to see if they are profitable. Not only is this method very time-consuming but it is also impractical because this process must be repeated in the future to make sure that the stock pairs continue to remain profitable over time.

The reason that it is so difficult to identify profitable pairs is due to the lack of an effective filtering method to determine which cointegrated pairs are suitable.

Using DBSCAN and PCA for Pairs Selection

One way to solve the aforementioned problem is to use clustering. By grouping stocks together into clusters and then testing which pairs of stocks in each cluster are co-integrated, it is possible to reduce the number of tradable pairs that need to be tested and increase the likelihood that the co-integrated pair is suitable.

There are plenty of popular clustering algorithms such as K-means and Hierarchical Clustering. However, for the purposes of clustering stocks, they are unsuitable. This is because, both fail in identifying clusters of arbitrary shapes and varying densities. One way to solve this issue is to use density-based spatial clustering of applications with noise (DBSCAN) [27]. As its name suggests, it identifies clusters by looking at the local density of data points. The main advantage of DBSCAN over K-Means is that it does not require the number of clusters to be told beforehand, which is crucial for clustering stocks since the number of clusters that exists is unknown at the start. It only requires two arguments: the maximum distance between points to be considered in a cluster and the minimum number of points in a cluster.

While DBSCAN can serve as a way of filtering out the stock pairs, it is also important to select which data points to include when trying to form clusters. Our goal for clustering is to identify similar companies that have similar historical price movements, which reduces the number of false positives generated by the co-integration test. With this goal in mind, the following features were selected:

1. Market Cap
2. Debt-to-Equity Ratio (Financial Health in Quantopian)
3. Historical Price

Market Cap was selected because it is an important differentiator between companies. Consider a large and established company like Coca-Cola and a small soft drink company like Jones Soda. It would be unreasonable to do a pairs trade between both stocks. Even though both companies are in the soft-drinks industry, they are very different from each other. Coca-Cola has a very stable business model while Jones Soda is still experiencing a lot of volatility. As a result, there is a lack of stability in the relationship between both stocks, which makes them a poor pairing for pairs trading.

The Debt-to-Equity (D/E) Ratio is used to evaluate a company's financial leverage and is important for determining the financial health of a company. Put simply, it measures the degree to which a company is financing its operations through debt against equity. The D/E ratio can enable us to determine whether two companies are financing their operations in similar ways. It would be inappropriate to compare two companies that have greatly different levels of D/E ratio. The company that has higher levels of debt relative to equity is at a higher risk of going bankrupt.

The Historical Price of the company is necessary to determine if the price movement of the two companies are similar. However, it is not appropriate to use the adjusted closing prices of the companies because there is too much noise. A better way is to conduct principal component analysis (PCA) on the price history of the stocks, and then use the 'principal components' as features in the DBSCAN.

PCA is a machine learning technique used to identify the principal components that explain the variation in the dataset through singular value decomposition (SVD). By conducting PCA on the price history of the stocks, it is possible to identify the main factors that explain the variation in the price movements of different stocks.

PCA Demonstration

Consider a dataset that contains the height and weight of a person. This dataset can be plotted as points in a plane as a combination of both factors, called "principal components". It is illustrated as the Figure 6 shows:

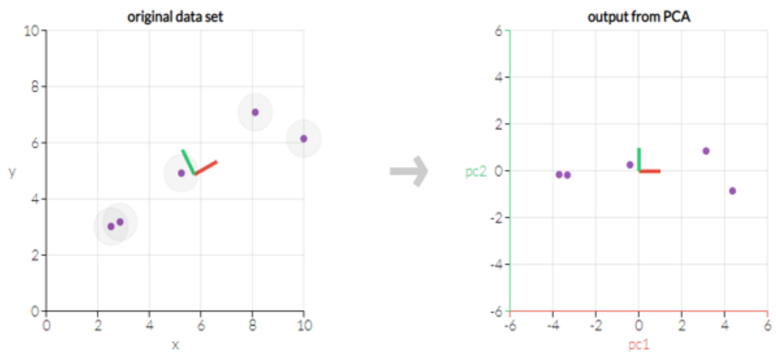


Figure 6: Illustration of 2-Dimension Data Before and After PCA

PCA finds a new coordinate system in which every point in the original dataset has a new (x,y) value on the pc1 and pc2 axes. The axes do not mean anything physical; they are combinations of the height and weight called "principal components". The illustration is shown in Figure 7:

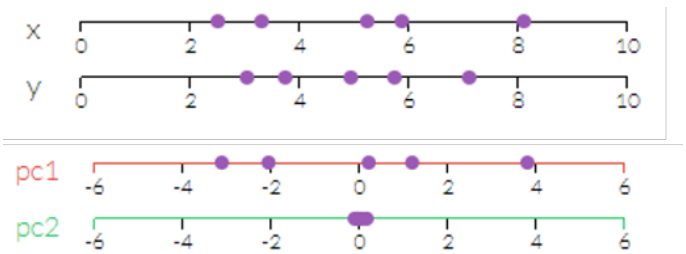


Figure 7: Illustration of The Axes Transformation After PCA

Application of PCA to Historical Price of Stocks

Going back to applying PCA to our price history dataset for the trading universe, which is composed of 1500 stocks. We consider the dataset as representative of the stock market, where each stock represents one feature of the stock market. By using PCA, we can identify the key 'principal components' that explain most of the variation in the stock market. For example, the first principal component which explains most of the variation in the price history is the stock market index, and the numerical value on the coordinate system of the principal component is the stock's beta.

Hence, by using PCA we can reduce the noise when comparing the distance between the price history of different stocks. Originally, the noise in the price data is difficult to observe. However, after conducting PCA, the effect of the noise is mostly be eliminated during dimension reduction as the noise is orthogonal to the principal components.

2.1.5 Development of the Backtesting System and Automated Trading Execution

Backtesting System

After developing the strategies on Quantconnect, we built a custom backtesting system to obtain more performance measures, analyze a more comprehensive equity curve, and set a benchmark other than S&P 500. Moreover, the custom backtesting system could expand the security trading universe that is not available on QuantConnect or other backtesting platforms. For building a backtest system, there are two ways to accomplish the goal:

1. Import the backtesting package(s) such as Zipline, PyAlgoTrade, and bt.
   (a) Advantages:
      i. Zipline is developed by Quantopian, so it might be easier for us to move our works in Quantopian to the Zipline IPython Notebook.
      ii. PyAlgoTrade is a fully-documented and mature backtesting package that could support both live- and paper-trading.
      iii. bt is specifically designed for asset weight allocation and portfolio rebalancing.
   (b) Disadvantages:
      i. The credibility of the data source might be relatively low: It is really important to conduct backtesting on the accurate data; however, as the data is provided by the package directly, it will be difficult for us to assure the correctness of the data.
      ii. It might take more time or even impossible for us to customize the measures we need for evaluating the strategy performance.
2. Collect the data from Interactive Brokers API, Bloomberg, or Thomson Reuters Eikon first, and then build a backtest system accordingly.
   (a) Advantages:
      i. The credibility of the data source is relatively high: As Bloomberg and Thomson Reuters are the financial data providers, their reliability on the data is higher as the data should be examined by many financial experts in the company.
      ii. We can customize the measures we need for evaluating the strategy performance as we build everything from scratch.
   (b) Disadvantages:
      i. It is inconvenient for conducting backtests: We need to define every measure by ourselves and we might need to conduct data cleaning for most of the time.

ii. The storage issue of the data: We have to store data collected locally and it might take up a lot of spaces.

Besides a backtesting system to generate the result, there will also be a visualization system for us to understand the advantages and disadvantages of the strategy in details.

Automated Trading Execution via the Interactive Brokers (IB) API

After implementing and backtesting our Machine Learning-based trading strategies, we will begin paper trading the Combined Portfolio Strategy, which is a combination of our three individual machine learning strategies with the optimal weight. The purpose of paper trading is to evaluate the performance of our Combined Portfolio Strategy in the real market data. For the paper trading, we used the Live Execution feature on QuantConnect, which allows us to connect to many brokers including the Interactive Brokers (IB) Paper Account with 1 million U.S. Dollars (paper money) as the initial capital.

2.2 Implementation

Before the shutdown of Quantopian, as it had a comprehensive dataset in price, fundamentals, and macroeconomics indicators for the U.S. Equity and Futures market, we decided to develop our strategies on Quantopian first. However, after the shutdown of Quantopian in November 2020, we implemented our code and executed our trades on QuantConnect. In the following sections, we introduced our implementations on QuantConnect.

2.2.1 Implementation of Trading Strategies using QuantConnect

Weight Allocation for Machine Learning Model-free Harry Browne Permanent Portfolio

In the passive investment, selecting the profitable securities with proper allocation had the most significant effect to the performance. In our project, we decided to adopt Harry Browne Permanent Portfolio as the passive investment target for the following 2 reasons:

1. Compared to holding 100% S&P 500 Index ETF or 60/40 Portfolio (60% in Stocks, and 40% in Bonds), the Permanent Portfolio invested in more asset classes such as Gold to hedge the risk from stocks and bonds.
2. Compared to Ray Dalio All Weather Portfolio (30% Stocks, 55% Bonds, 7.5% Gold and 7.5% Commodity), the Permanent Portfolio did not invest in the Commodity Sector, which lost 9.37% annually in the past 10 years [28].

With the advancement of technology and the development of algorithmic trading, the current market condition is totally different from the one in 1980s when Harry Browne proposed the Permanent Portfolio. Therefore, we decided to adopt some changes in the original Harry Browne by implementing the following steps for getting the final results:

1. Set the starting date on December 1, 2004 as all four components have incepted since then.
2. Compute the EWMA volatility for the components and calculate the weights accordingly.
3. Compute the difference (Δw_i) between the existing weights and the new weights that will be allocated for each of the component
4. The amount of shares to buy (or sell if negative) is calculated as $\frac{\Delta w_i \cdot Total\ Portfolio\ Value\ (NAV)}{Current\ Price\ of\ ETF\ i}$.
5. Repeat the process for every month until March 31, 2021.

Weight Allocation for Machine Learning-based Harry Browne Permanent Portfolio by Long Short-Term Memory (LSTM) Model

In the project, we applied LSTM model to actively predict the stock price instead of allocating the weight simply based on the volatility that we did in the Machine Learning model-free one. Since December 2004, we predicted the future 1-month stock prices by the existing data for every month. Then, we calculated the corresponding expected returns and the variance-covariance matrix for the components in the Harry Browne Permanent Portfolio. For finding the optimal weights for the passive investment, we implemented it in the following steps:

1. Split the existing stock price data for one of the ETFs (e.g. SPY) as 80/20 training and testing ratio.
2. Predict the stock price on the training data by the LSTM model.
3. Validate the model result on the testing data.
4. Apply the trained model for the prediction of the future one-month stock movement.
5. Repeat Step A for all other three components.
6. Calculate the expected monthly return by considering how much it grows (or declines) within the one month forecast.
7. Compute the variance of each ETF and the covariance between them and derive a covariance-variance matrix based on the price movements forecast.
8. Input the expected monthly return and the covariance-variance matrix into Markowitz mean-variance portfolio theory
9. Repeat Steps 1 to 8 for each month starting from Dec 2006 (2-year warm-up period).

Note that in the first two years, we applied the volatility targeting for determining the weights as the existing data was not sufficient to make predictions.

Stock Selection Using NLP

Currently, there are over 10,000 stocks with equivalent 10-K reports in the database of the SEC's Electronic Data Gathering, Analysis, and Retrieval (EDGAR) website. To download all the necessary 10-K reports, these are the following steps:

1. Gather a list of tickers and central index keys for all the stocks you are interested in.
2. Use Beautiful Soup package to scrape the url address of all 10-K reports.
3. Use the Requests package to download all the documents in each 10-K report. Note that there are multiple documents filed for every 10-K report, but only one of the documents contains the text for the 10-K annual report.
4. Check each document for its sequence and type to find the correct document.

After downloading the 10-K report, they needed to be cleared and processed before they can be analyzed. It is important to note that all complete 10-K filings are in HTML text format. Therefore, the first step is to remove all the HTML tags. The next thing to do is to lowercase the entire text. After that, it is necessary to normalize the text data because there are unicode symbols from encoding the data during the download. Finally, we can start getting the similarity scores between consecutive 10-K reports for a particular company.

To get the sentiment score, convert each 10-K report to a Term Frequency-Inverse Document Frequency (TF-IDF) vector representation and a BoW (Bag-of-Word) vector representation with the vocabulary being the Loughran-McDonald Master Dictionary.

Quantconnect doesn't have the 10-K report text data within their system. Because of this, the computation for the similarity scores between 10-K reports has to be done locally on your own computer. From there, the similarity score dataset needs to be uploaded to dropbox before it can be imported into the quantconnect.

Trading Strategy

1. Everyday, check if there is a 10-K report for a stock that is in your universe.
2. For each 10-K report, check what the similarity score for the current 10-K report with the previous year 10-K report.
3. If the similarity score > 0.9, then long the stock, else short the stock.
4. At the start of every year, compute the weights for each stock in the portfolio using a market-value-weighted index.

Pairs Trading with PCA and Clustering

Pairs Selection

First of all, we needed to find the pairs for doing pairs trading. For finding the stock pairs that are suitable for the pairs trading, we implemented it in the following steps:

1. Select the stock tickers for the top 1500 U.S. stocks by Market Cap.
2. Extract the stock prices for the 1500 stocks in the past 2 years.
3. Apply PCA on the stocks price and obtain the top 50 principal components.
4. Use the top 50 principal components, market cap, and financial health (D/E Ratio) as the features.
5. Implement DBSCAN with some minimum epsilon and number of sample in the group.
6. For each cluster, evaluate all possible pairs within the same cluster.
7. Conduct the co-integration test on all possible pairs and select 2 pairs with the lowest p-values.
8. Repeat Steps 1 to 7 for every 6 months.

Trading Strategy

In this section, we mainly focused on the trading strategy after the pairs are selected. After the pairs have been selected, the next step is to check everyday if the daily spread is statistically significant relative to the 20-day spread moving average. We define a Z-score to assess the deviation:

$$Z - score = \frac{S_t - \bar{S}}{\sigma_{S_{20}}},$$

where S_t is the spread on day t, $\bar{S}$ is the average spread for the past 20 days, and $\sigma_{S_{20}}$ is the standard deviation of the spread over the past 20 days.

There are three important things to note:

1. Trading Signals:
   (a) Buy spread if Z-score is below -0.5 (Buy stock Y and sell stock X)
   (b) Sell spread if Z-score is above 0.5 (Sell stock Y and buy stock X)
2. Exit Position:
   (a) Unwind position when Z-score hits +0.1 or -0.1.
   (b) Generally a trade-off between profits per trade and number of trades for the trading threshold level
3. Tradeoffs to consider:
   (a) High threshold $\rightarrow$ higher profits
   (b) Low threshold $\rightarrow$ higher number of trades

2.2.2 Visualization and Automated Trading System

Visualization System

For the visualization system on QuantConnect, it is illustrated in Figure 8, 9, and 10. The report can be generated after the backtest has finished.

Figure 8: Illustration of QuantConnect Visualization Report (Page 1)

Figure 9: Illustration of QuantConnect Visualization Report (Page 2)

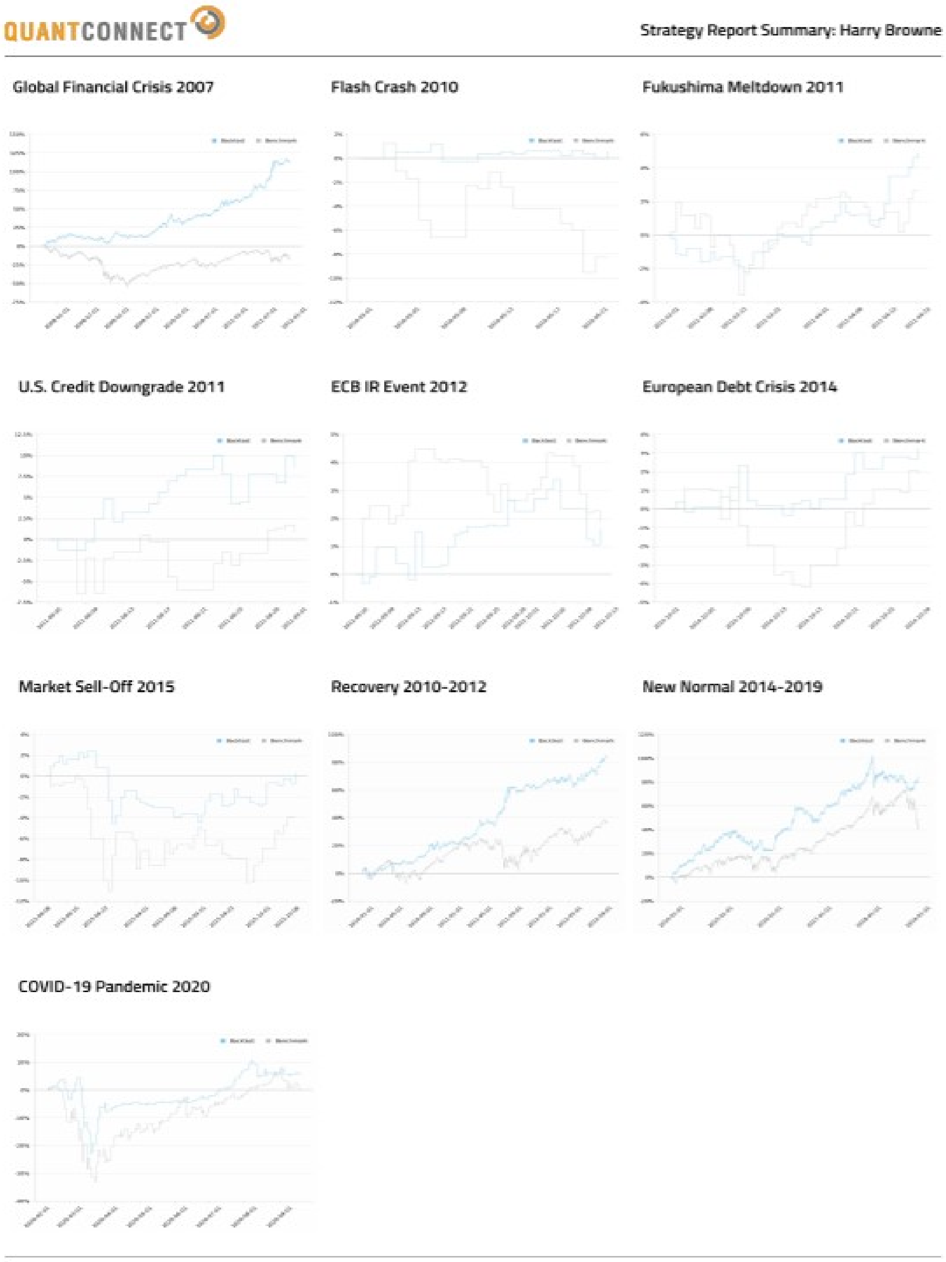


Figure 10: Illustration of QuantConnect Visualization Report (Page 3)

After developing the strategies on the Quantopian and QuantConnect, we built a visualization system that could demonstrate the key performance in a clearer manner than the one provided by the platforms. For our custom visualization system, we developed it with interactive plots designed by a Python library named Plotly embedded on the HTML webpage. It was accomplished by a Python library named Dash.

After obtaining the portfolio value (or Net Asset Value, NAV) from the trading strategies, we processed it to generate the following graphs by Plotly. Then, we embedded all the interactive plots into a webpage by Dash:

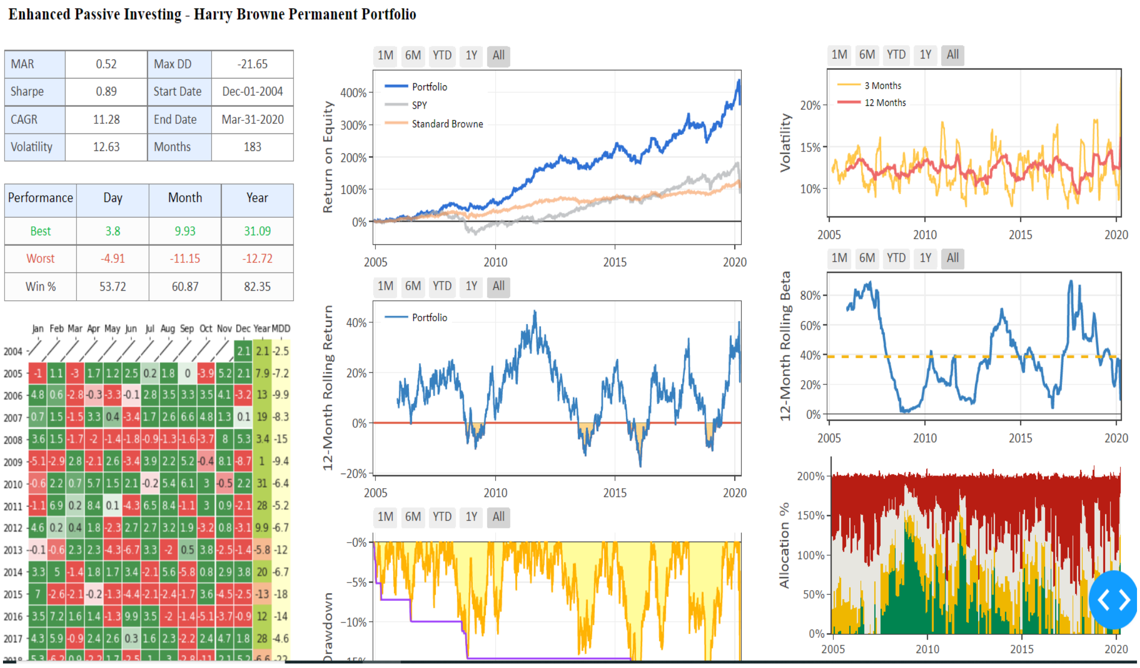


Figure 11: Upper Part of The Visualization Webpage

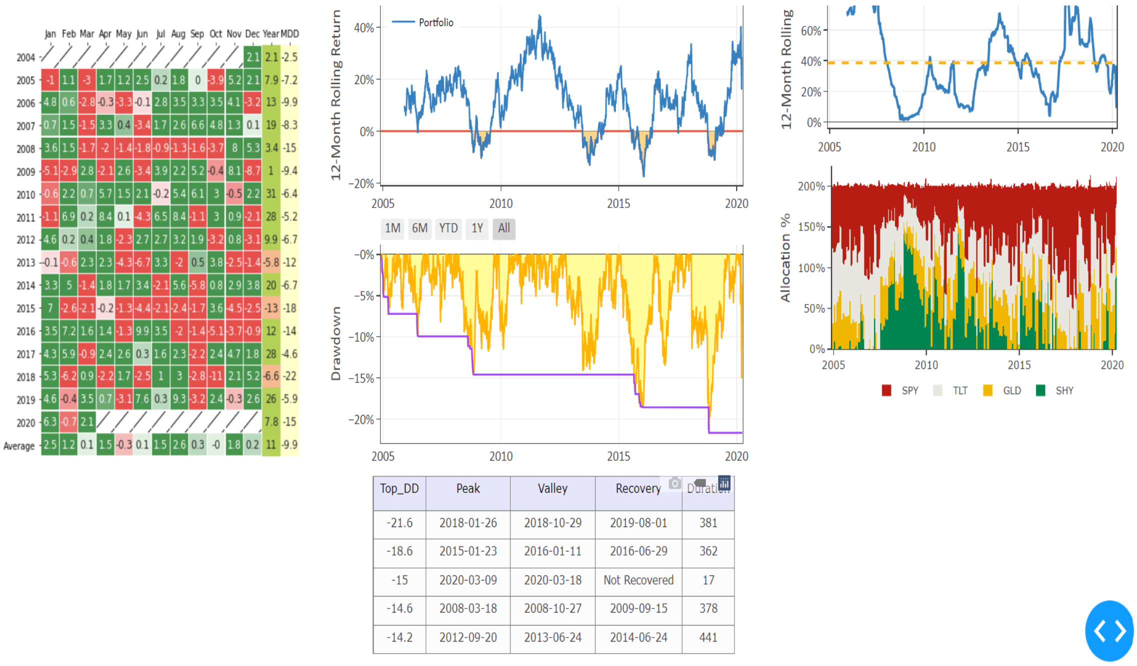


Figure 12: Lower Part of The Visualization Webpage

For the visualization page, we could split it into the following 8 parts:

1. **Basic Measures and Statistics**

   In this part, we included some metrics like CAGR, Sharpe Ratio and Volatility. We also added MAR Ratio, which is defined as CAGR divides by the Max Drawdown. It is an indicator to reflect, on average, how long it takes to recover from the biggest drawdown. If the MAR ratio is 0.5, it will averagely take us $\frac{1}{0.5} = 2$ years to recover from the max drawdown. Moreover, we summarized the total percentage of days, months, and years that are earning money (i.e. having positive returns) to see whether the strategy could earn money most of the time.

MAR	0.52	Max DD	-21.65
Sharpe	0.89	Start Date	Dec-01-2004
CAGR	11.28	End Date	Mar-31-2020
Volatility	12.63	Months	183

Performance	Day	Month	Year
Best	3.8	9.93	31.09
Worst	-4.91	-11.15	-12.72
Win %	53.72	60.87	82.35

Figure 13: Basic Metrics of The Strategy

2. **Heatmap**

We also constructed a heatmap to identify whether there is any specific month or a series of months that our portfolio performed differently throughout the backtesting period. For example, this portfolio performed poorly from February 2015 to December 2015.

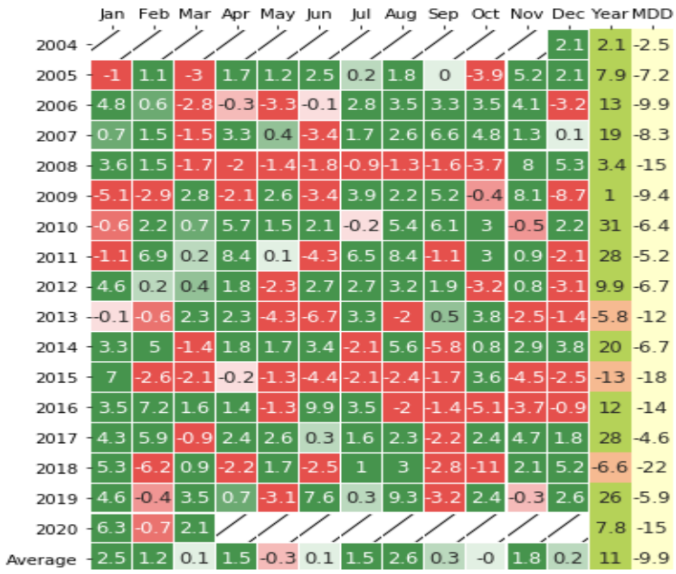


Figure 14: Heatmap of The Monthly Returns

3. **Equity Curve**

For the equity curve, we add the standard Harry Browne Permanent Portfolio as another benchmark. As Figure 16 showed, the image is designed in an interactive mode.

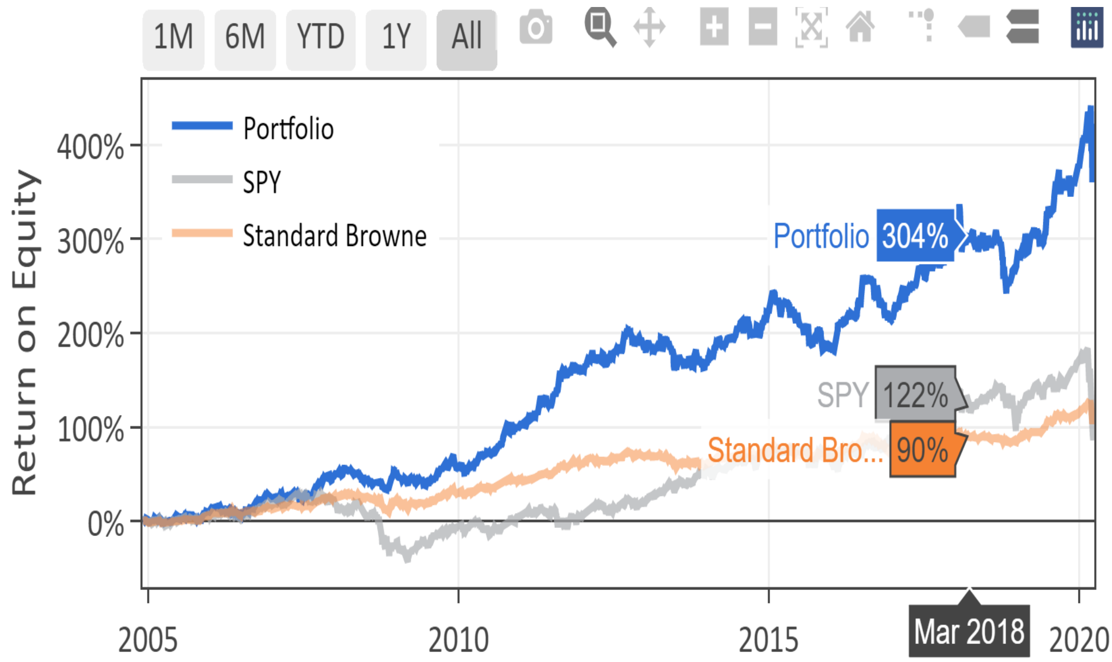


Figure 15: Interactive Equity Curve

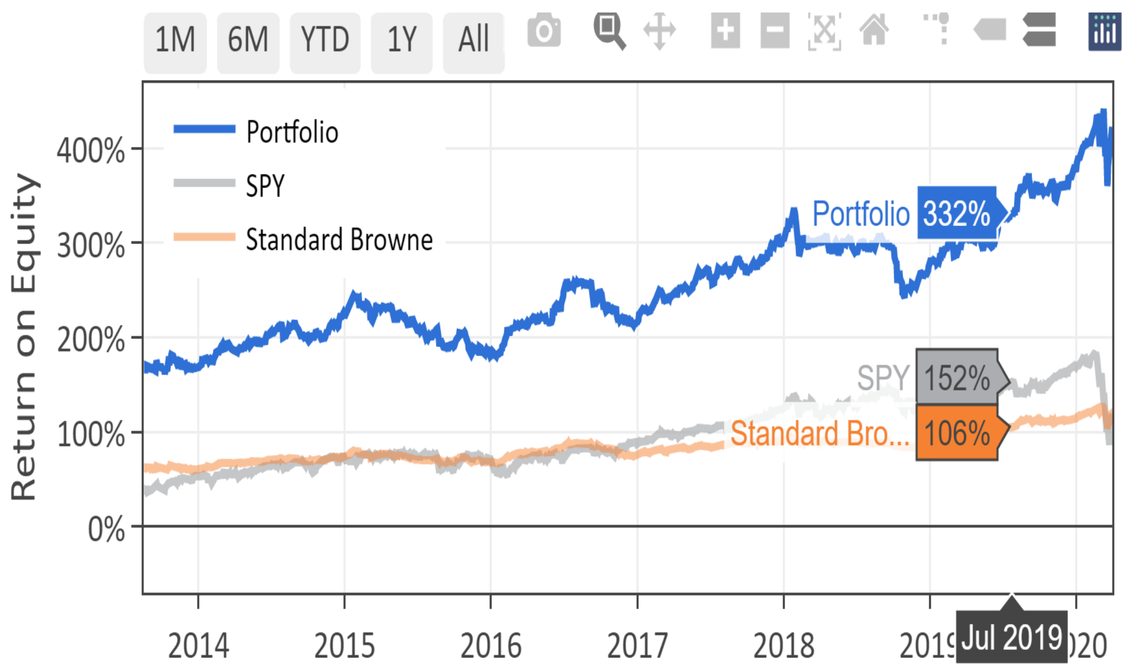


Figure 16: Interactive Equity Curve for The Recent 6 Years After Zooming in

4. **12-Month Rolling Return**

Although it is common to lose money in a month, losing money in a year is something that we want to prevent. Therefore, it is important for us to understand the performance of the portfolio in recent 12 months as the annual return for each year showed in Figure 14 might not be an exact and sufficient indicator for knowing how the portfolio performed recently. Therefore, it would be beneficial to visualize the performance in the recent year. In Figure 17, this portfolio had an 8% loss from April 2008 to April 2009.

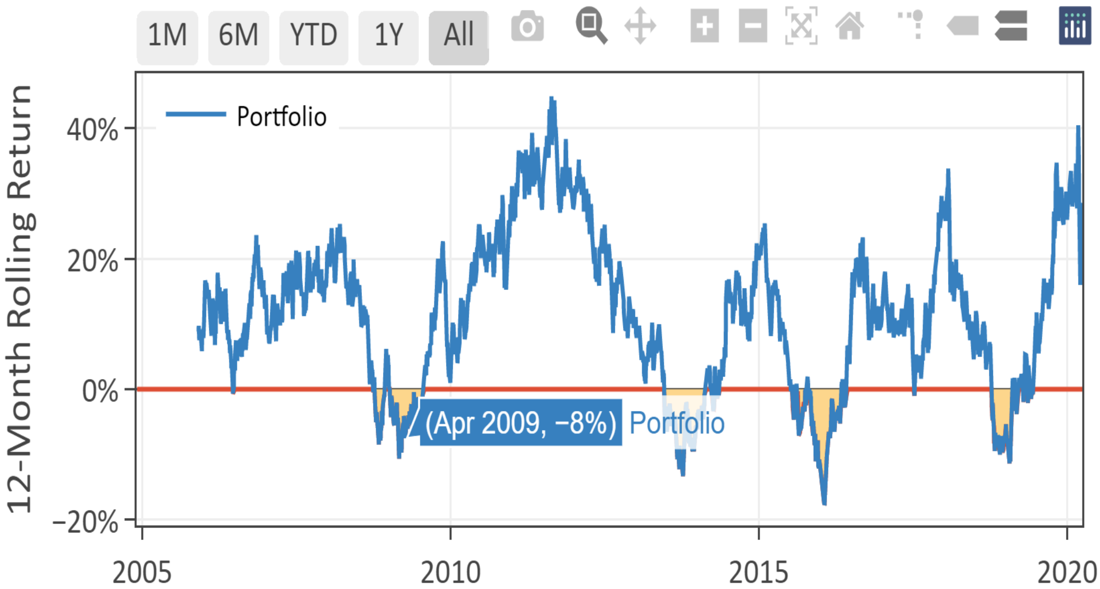


Figure 17: 12-Month Rolling Return

5. **Drawdown**

As suffering losses makes people panic and not willing to invest their capital in the financial market, it could significantly improve the confidence of investing if the previous record of the maximum capital loss has been revealed. Therefore, in Figure 18, we observed all the huge drawdowns ever happened in the strategy and could analyze the root after identifying the time of the drawdown.

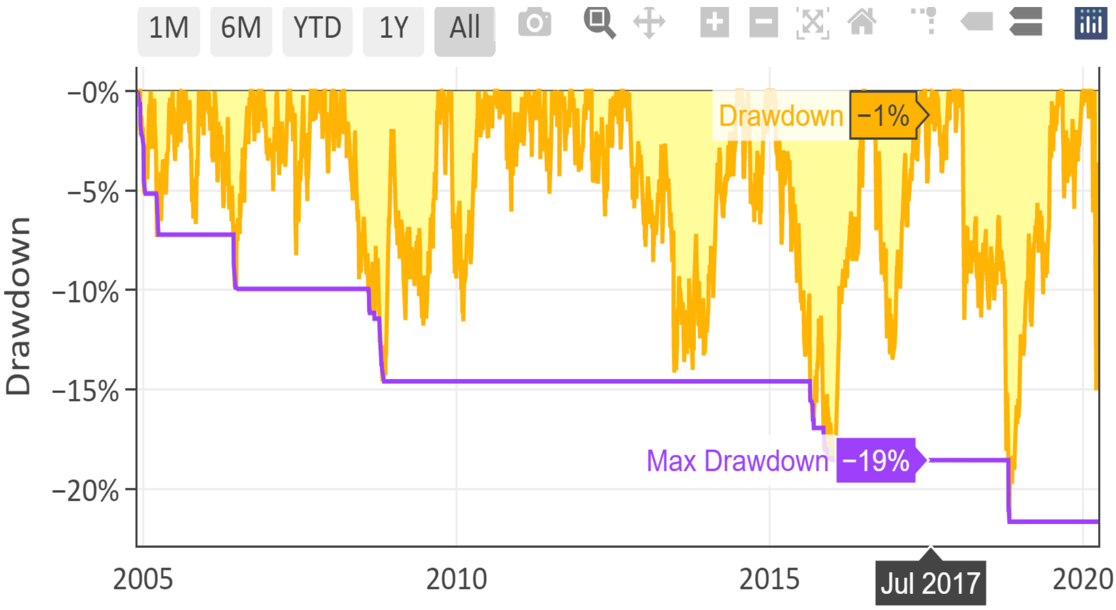


Figure 18: Drawdown Diagram

Moreover, we have created a table for listing the top 5 worst event throughout the backtesting period. For instance, this portfolio had a 18.6% drawdown from January 23, 2015 to January 11, 2016, which matched with our observation in the heat map.

Top_DD	Peak	Valley	Recovery	Duration
-21.6	2018-01-26	2018-10-29	2019-08-01	381
-18.6	2015-01-23	2016-01-11	2016-06-29	362
-15	2020-03-09	2020-03-18	Not Recovered	17
-14.6	2008-03-18	2008-10-27	2009-09-15	378
-14.2	2012-09-20	2013-06-24	2014-06-24	441

Figure 19: Drawdown Summary Table

6. **Volatility**

As we previously mentioned, it is important to know the volatility of the portfolio as it indicates the higher possibility of losing certain amount of money in the coming period if the portfolio is more volatile. In the project, we adopted the 3- and 12-month as the period of measuring the volatility. As the volatility has the clustering effect, examining the short-term (3-month) and the long-term (12-month) volatility could give us a better picture of the market condition.

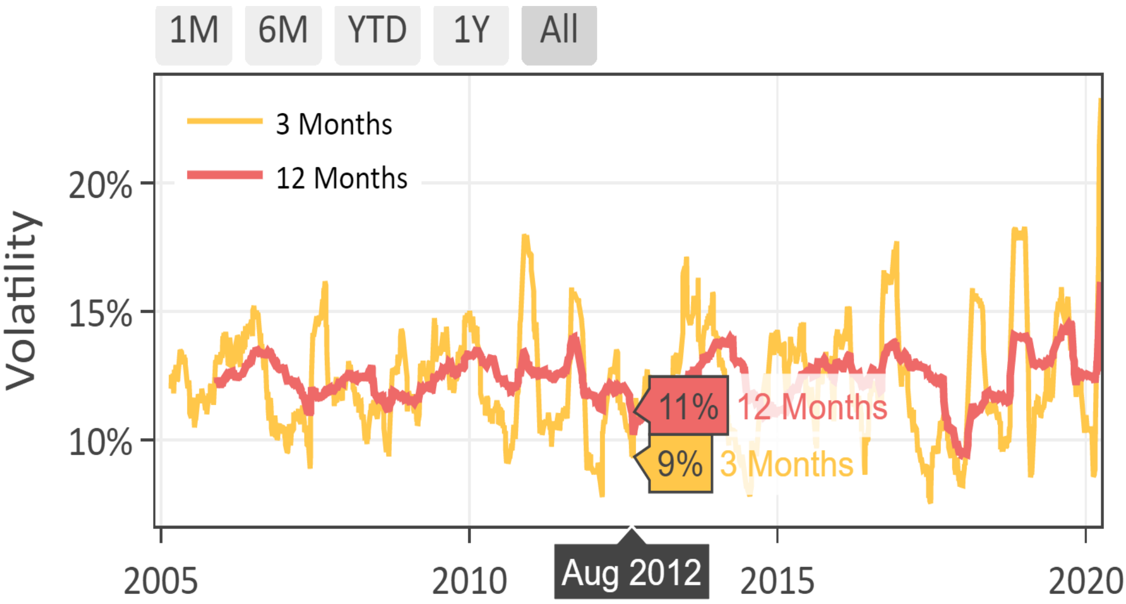


Figure 20: Volatility Diagram

7. **12-Month Rolling Beta**

As the portfolio is susceptible to the market movement, we measure the beta to examine the risk explained by the market. In the project, we decided to adopt 12-Month Rolling Beta as the indicator.

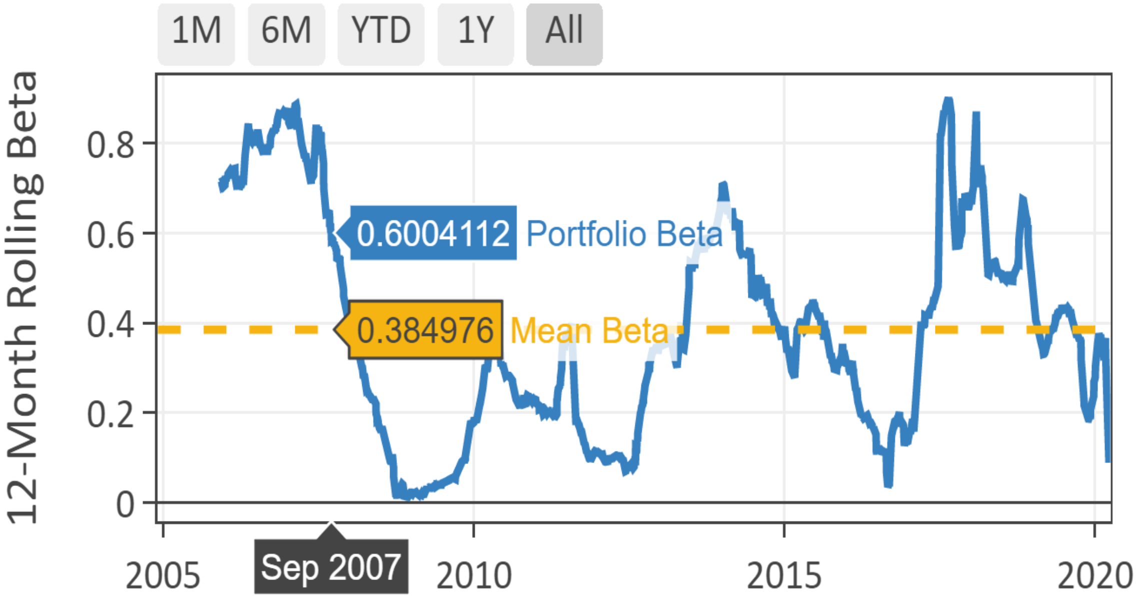


Figure 21: Rolling Beta

8. **Weight Allocation**

As allocating the capital properly is the key factor to the passive investment, it is a major concern for us to see how the allocation is during different period of time. In this portfolio, we invested in SPY, TLT, GLD, and SHY.

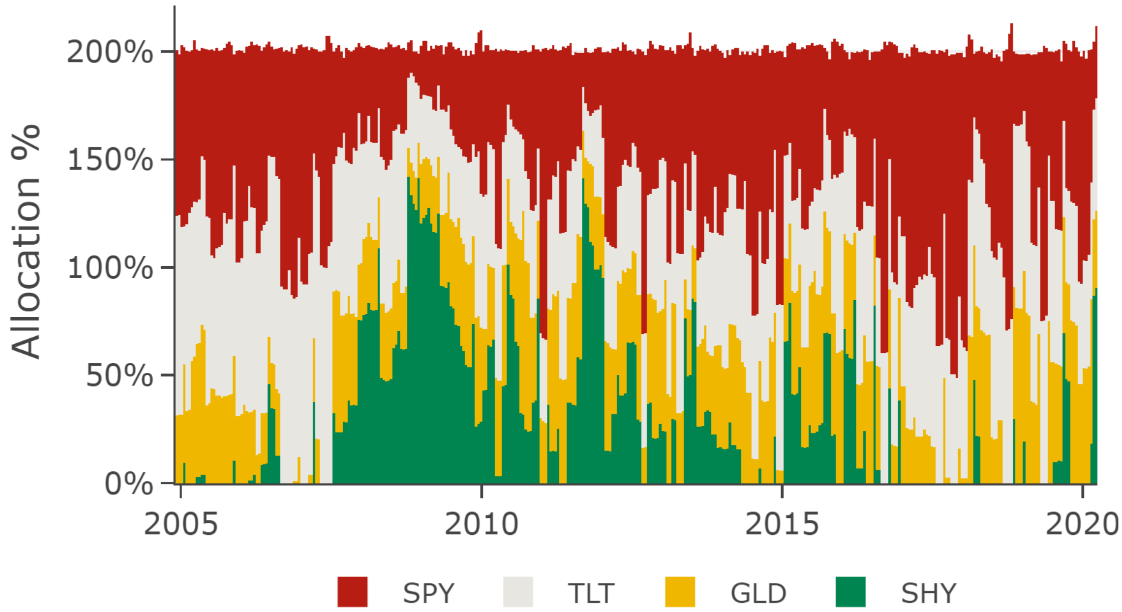


Figure 22: Weight Allocation

Automated Trading

After all the strategy-wise parts were done, we executed the trades on the Interactive Brokers paper trading account with the our proprietary auto-trading system. As we just started on April 14^{th}, it will last for one month to understand the performance of our strategies in the real-time market.

Specifically, we executed our Combined Portfolio Strategy developed on QuantConnect in the Interactive Brokers through the live execution functions in QuantConnect. We just started our 1-month paper trading period to examine the performance properly on April 14^{th}, and we will be presenting the result during the oral presentation.

2.3 Testing

2.3.1 QuantConnect Platform Testing

For all of our trading strategies, we created them using the QuantConnect platform. More specifically, it means that we relied on the platform's backtesting system and financial data to implement our Machine Learning-based trading strategies and verify their performance. The advantage of using this platform is convenience and ease of use. We didn't have to develop our backtesting system. Aside from this, accurate financial data is costly and requires the proper infrastructure for proper storage and quick access. On the other hand, the downside of relying on QuantConnect is that we have to trust that QuantConnect's backtesting system is properly implemented and that their data is accurate. Overall, we believe the benefits of using QuantConnect outweigh the possible cost as examining strategies on the accurate financial data is the top priority in the project.

For our trading strategies, we required the daily open, high, low, close price for the assets we were trading to generate our trading signals. Hence, to ensure that the data we were using was correct, we developed our data-cleaning tool, which analyzed the adjusted closing price history of a stock by using multiple data sources including Bloomberg and Thomson Reuters. We concluded that while there were some minor discrepancies in the data, for the most part (about 99% of the data), the differences were negligible.

To verify that QuantConnect's backtesting system works as described, we conducted testing according to the established practices of software testing including grey-box testing and unit testing. This is because we only have partial information on the internal structure of QuantConnect's backtesting system based on the documentation they provided for developers. For example, we developed basic trading algorithms that included buying and holding stocks at specific schedules. From there we verified the list of orders made in the backtest to see if it matched with what we expected to see for that specific trading algorithm. Aside from order executions, another aspect that we tested involves the performance metric calculations. First, we calculated what the expected performance metrics were for simple trading algorithms, then we compared the results with QuantConnect's backtesting results. Our tests showed that QuantConnect's backtesting system worked as described in their documentation.

2.3.2 Individual Machine Learning Strategies Testing

Machine Learning-based Harry Browne Permanent Portfolio

Testing for the Machine Learning-based Harry Browne Permanent Portfolio strategy was done through a combination of unit testing and integration testing. First off, the functions used to process and manipulate the data have been debugged and verified to ensure that they work properly. Aside from this, we assessed and trained our LSTM models using measures of prediction accuracy like root mean square error (RMSE).

Beyond developing the LSTM model, we also manually checked that the function used to compute for the variance and covariance matrix necessary for Markowitz mean-variance portfolio theory. Next, we conducted unit testing on the program that allocates the portfolio weights for each ETF depending on optimal portfolio theory.

After we tested that all of the individual programs worked separately, we then conducted integration testing within QuantConnect's backtesting system. We used the platform's log function to check that all programs still worked as intended. Finally, we also ensured that the orders executed only on the appropriate schedule. This trading strategy is supposed to be rebalanced at the beginning of each month. However, we discovered that if the first weekday of the month was a holiday then the backtesting system would fail to execute a trade for that specific month. We solved this problem by using the holiday tracker within the pandas' package to manually track which holidays fall on the first trading weekday of the month.

Stock Selection Using NLP

Testing for the NLP Stock Selection strategy was done through a combination of unit testing and integration testing. In the beginning, we gathered the 10-K raw text data from the SEC Edgar Database using their API. Afterward, we processed the data into a vector representation of the text document as described in section 2.2.1. Unit testing was conducted for all the functions used to complete this transformation to ensure that the final vector representation of the text is accurate.

After that, we stored the vector representation of each 10-K document in the school's web server for FYP projects. In the next stage, we computed the similarity score between succeeding 10-K reports for a particular company. To ensure that the signals were accurate, we tested the functions that compute for Jaccard and Cosine Similarity.

Lastly, we implemented the trading strategy described in section 2.2.1. For testing, we conducted integration testing between QuantConnect's backtest system and the helper functions used to implement our trading strategy. We used the platform's log function to check that all programs function as intended. In the end, we went over the order history to ensure that the orders made in the backtest followed our algorithm: Only execute a buy or sell order if, on the previous trading day, the company released their annual report. By implementing the strategy properly, we prevented look-ahead bias.

Pairs Trading with PCA and Clustering

We conducted unit testing for the functions that manipulated and processed the data to ensure that the signals they generated were correct. Specifically, we verified that, for our universe of stocks, the PCA and clustering algorithm identified groups of stocks that were similar in market cap, financial health, and industry. Aside from this, we tested our program to make sure that the trading logic described in Section 2.2.1 has been properly implemented.

Beyond this, we performed integration testing between the functions used with the QuantConnect's backtesting system to ensure that the final output is correct. QuantConnect has a feature that allows us to log information during a backtest, which can be downloaded afterwards as a txt file. We used this to ensure that the outputs of each function was correct. Finally, we also checked to make sure that the orders executed only occurred when the appropriate signal was generated (i.e. when spread was statistically significant).

2.3.3 Combined Portfolio Testing

After properly testing the individual strategy in the previous steps, we examined whether the combined portfolio with equal weight on each strategy has the same result we were expecting for. However, the return of a combined portfolio is not simply the weighted sum of the return of each individual strategy due to the compounding effect of time value and the holdings of different strategy. Therefore, we created three strategies where we just buy-and-hold a different stock. Then, we combined the strategy and ensured it worked properly. Next, we record the holding of each stock according to the strategy by the self.Plot function (See Figures 23 and 24 for illustration) on QuantConnect. After viewing the corresponding holding, we conducted unit testing and verified the correctness of the combined portfolio.

```
self.Plot('Holdings', 'leverage', round(self.leverage, 2))
self.Plot('Securities_Positions', 'Long_count', round(self.long_count, 2))
self.Plot('Securities_Positions', 'Short_count', round(self.short_count, 2))
self.Plot('Securities_Positions', 'Total_Securities', round(self.total_securitites , 2))
self.Plot("Margin", 'Margin_remaining', round(self.margin_remaining_percent, 2))
```

Figure 23: Code for Plotting Necessary Variables

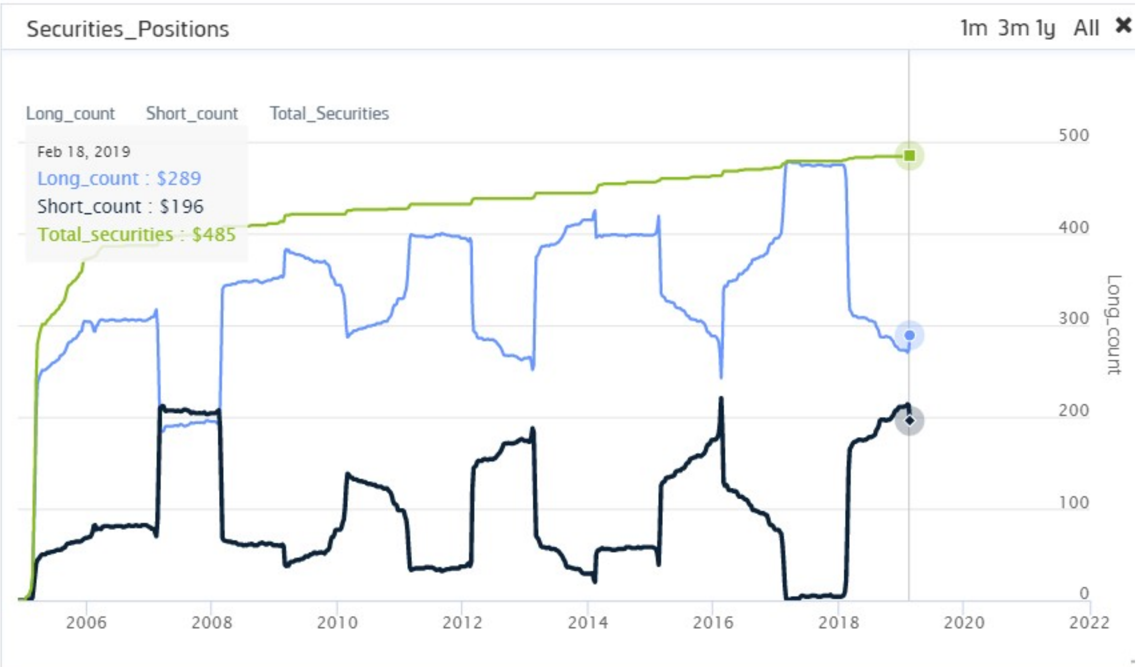


Figure 24: The Custom Variable Tracker from The Self.Plot Code

2.3.4 Visualization System Testing

Unit testing has been conducted when we created each graph that is on the page. Before embedding them in the page, we created a graph with the simulated data and tested whether the graph showed the same thing that we expected. For the summary statistics, we ensured that the results such as CAGR, Sharpe Ratio, and MAR Ratio are same as the QuantConnect result as well as our calculations based on the Bloomberg data for a buy-and-hold S&P 500 Index ETF strategy. For the heatmap, we input the simulated data with a clear pattern of growing and declining to see whether it showed properly. For other graphs, we compared them with the QuantConnect results first, and then we also computed the corresponding numbers for each time step. Overall, all the codes have been properly debugged and implemented.

2.3.5 Automated Trading System Testing

For the automated trading system on QuantConnect, we did testing for two parts:

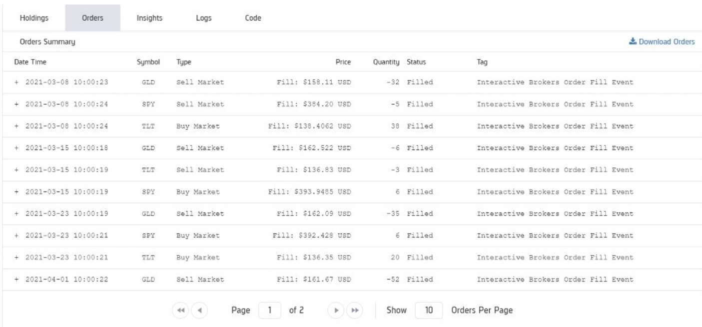
1. Dashboard on QuantConnect:

   An important aspect to consider when trading is to ensure that you are receive accurate up-to-date trade statistics like unrealized profit. Otherwise, you won't be able to keep track of the overall performance of your strategy. To solve this, we also did end-of-day portfolio calculations using the trade data from Interactive Brokers and price data from Bloomberg.

Figure 25: QuantConnect Automated Trading Dashboard

Besides, there are 3 statuses in during live trading, which are "launch" (when it first connected to IB account), "runtime error" (when there is an error in trading), and "stopped" (when the user paused the connection) (See Figures 26, 27, and 28 for details)

Figure 26: QuantConnect Automated Trading Dashboard (Status = Launch)



Holdings | Orders | Insights | Logs | Code

Orders Summary — Download Orders

Date Time	Symbol	Type	Price	Quantity	Status	Tag
+ 2021-03-08 10:00:23	GLD	Sell Market	Fill: $158.11 USD	-32	Filled	Interactive Brokers Order Fill Event
+ 2021-03-08 10:00:24	SPY	Sell Market	Fill: $384.20 USD	-5	Filled	Interactive Brokers Order Fill Event
+ 2021-03-08 10:00:24	TLT	Buy Market	Fill: $138.4062 USD	38	Filled	Interactive Brokers Order Fill Event
+ 2021-03-15 10:00:18	GLD	Sell Market	Fill: $162.522 USD	-6	Filled	Interactive Brokers Order Fill Event
+ 2021-03-15 10:00:19	TLT	Sell Market	Fill: $136.83 USD	-3	Filled	Interactive Brokers Order Fill Event
+ 2021-03-15 10:00:19	SPY	Buy Market	Fill: $393.9485 USD	6	Filled	Interactive Brokers Order Fill Event
+ 2021-03-23 10:00:19	GLD	Sell Market	Fill: $162.09 USD	-35	Filled	Interactive Brokers Order Fill Event
+ 2021-03-23 10:00:21	SPY	Buy Market	Fill: $392.428 USD	6	Filled	Interactive Brokers Order Fill Event
+ 2021-03-23 10:00:21	TLT	Buy Market	Fill: $136.35 USD	20	Filled	Interactive Brokers Order Fill Event
+ 2021-04-01 10:00:22	GLD	Sell Market	Fill: $161.67 USD	-52	Filled	Interactive Brokers Order Fill Event

Page 1 of 2 Show 10 Orders Per Page

Figure 27: QuantConnect Automated Trading Dashboard (Status = Runtime Error)

Figure 28: QuantConnect Automated Trading Dashboard (Status = Stopped)

2. Trade executions through the code on QuantConnect:

   Although we can automate execution using the Live Trading feature in QuantConnect, it is essential to still manually go over each trade that is made by the algorithm. This is to ensure that the algorithm is working as intended. If we fail to do this, then we could potentially lose a lot of money since the backtest results are no longer relevant. Our solution to resolve this issue is to keep record of all trades made in an excel file and then after 3 days do a backtest on QuantConnect to see whether or not a trade should have executed. Figure 29 showed how the orders look like and Figure 30 illustrated the equity value for the strategy on a certain day.

Figure 29: Orders on QuantConnect Automated Trading Dashboard

Figure 30: Equity Value on QuantConnect Automated Trading Dashboard

To conclude, the testing on each aspect of the project have been completed and all of the implementations worked properly as we proposed.

2.4 Evaluation

In this section, we evaluated our strategy performance by answering the following six questions:

Note: In Questions 1 to 3, we discuss the profitability of three Machine Learning-based strategies. As the return and Max Drawdown can be amplified with respect to the leverage, we emphasized more on the Sharpe Ratio and the MAR Ratio which are independent to the financial leverage.

1. **How is the performance of the Harry Browne Permanent Portfolio?**

   Machine Learning Model-free

   On QuantConnect, we obtained an 18.07% Compounded Annual Growth Rate (CAGR), a Sharpe Ratio of 1.08, and the Max Drawdown of 26.10% over 188 months of the backtesting from December 2004 to August 2020 with the initial capital of 10 million U.S. Dollars under a 2x leverage.

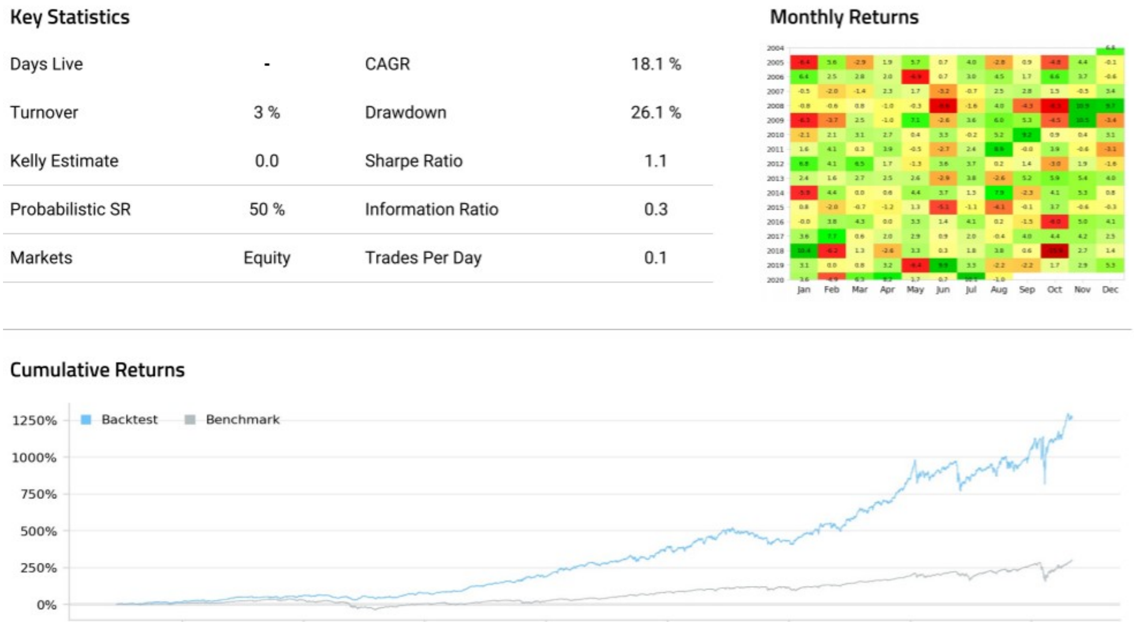


Figure 31: Backtest Result of Harry Browne Permanent Portfolio from December 2004 to August 2020

Machine Learning Model-based (LSTM)

For the Machine Learning model-based strategy, we obtained an annualized return of 20.4%, a 1.25 Sharpe Ratio, and the Max Drawdown of 18.00% from December 2004 to March 2021.

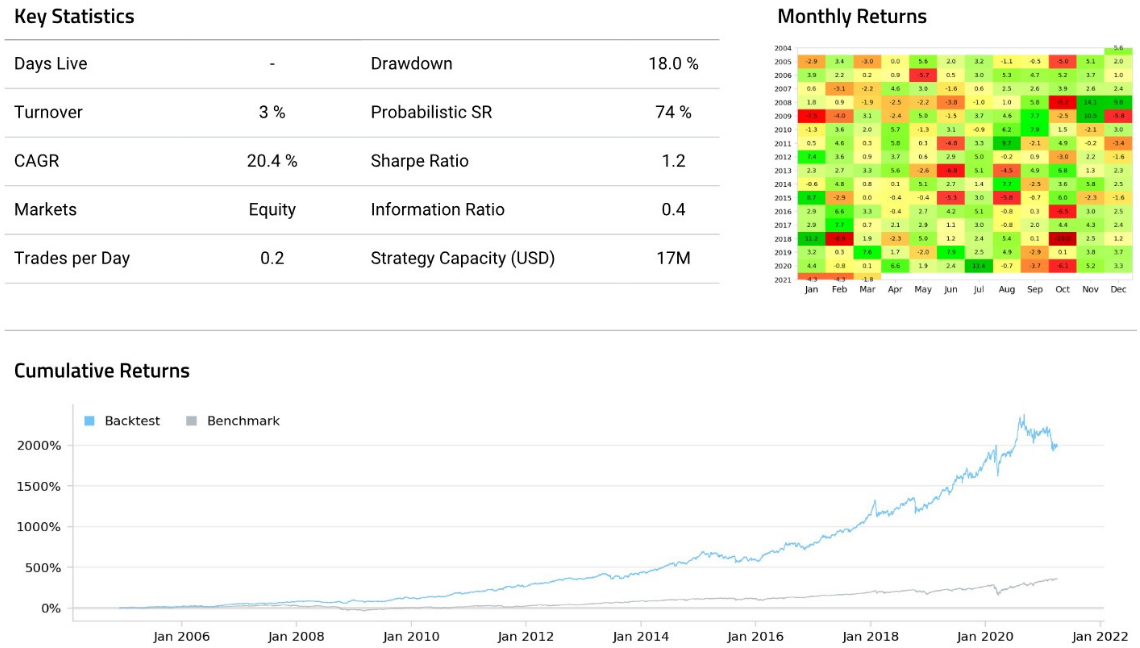


Figure 32: Backtest Result of Machine Learning Model-based Harry Browne Permanent Portfolio from December 2004 to March 2021

To compare the results, we noticed that the one with the stock price prediction from LSTM has a higher annualized return (around 2% higher) while a smaller Max Drawdown throughout the past 16 years. We believe that the predicting power of LSTM on the stock price helped us to identify some downturns as we did not invest in any asset with a negative expected return when we constructed the Markowitz Mean-Variance Portfolio.

For example, the one based on the Machine Learning models was almost not affected by the coronavirus crash in February and March 2020. On the other hand, the one simply based on volatility targeting had a monthly loss of 4.9% in February 2020. This indicates how the prediction of the asset performance could improve our performance during the financial crisis.

2. **How is the performance of "Stock Selection Using NLP" strategy?**

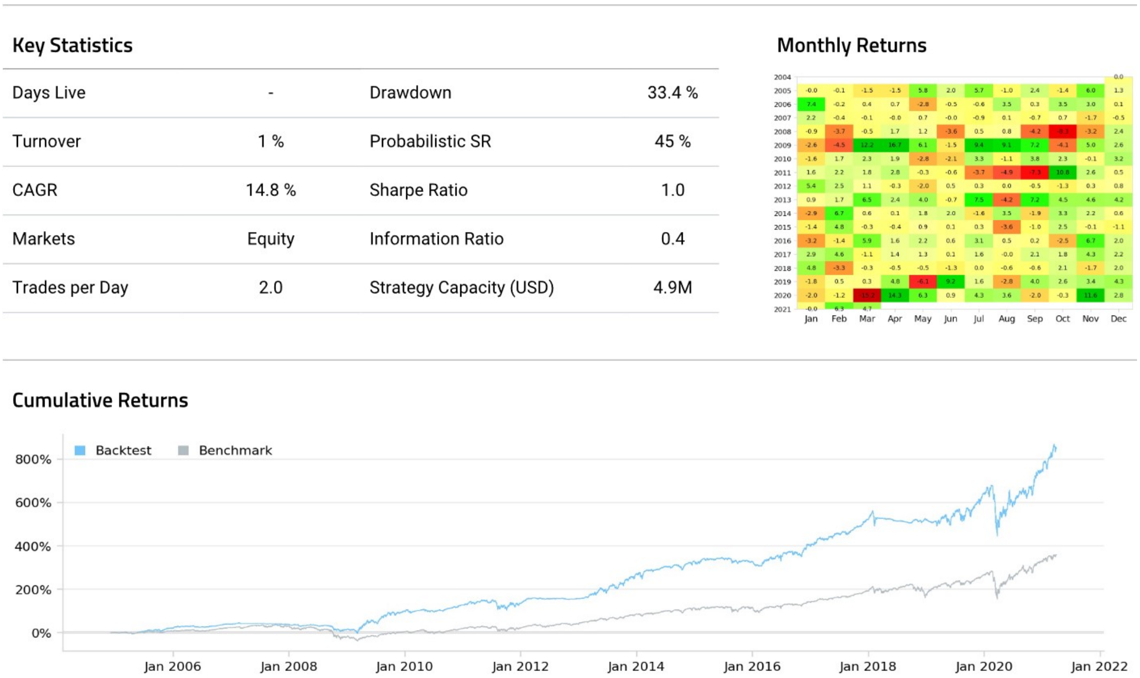


Figure 33: Backtest Result of Stock Selection Using NLP from December 2004 to March 2021

As shown in Figure 33, we reaped an annualized profit of 14.8%, a Sharpe Ratio of 1.06, and the Max Drawdown of 33.4% for the past 16 years since December 2004. During the same period, SPY generated a CAGR of 9.85%, a Sharpe Ratio of 0.62, and a 55.10% Max Drawdown. The Sharpe Ratio and the CAGR of the NLP stock selection strategy is 72% and 50% higher than those of SPY respectively. Overall, it illustrates that the NLP-based stock selection technique consistently beats the U.S. stock index over a long period of time.

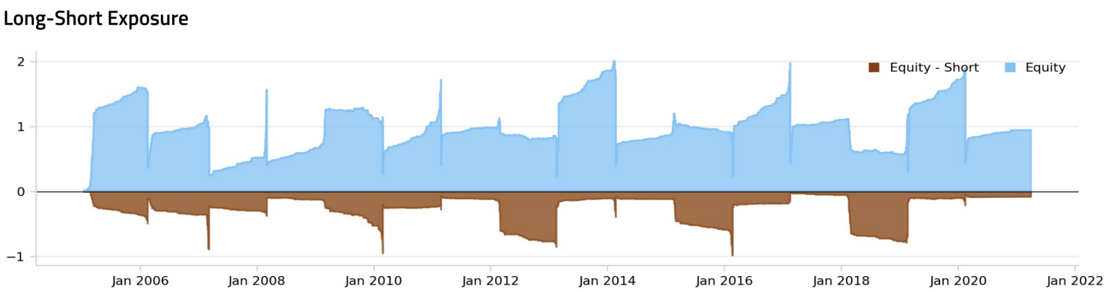


Figure 34: Long-Short Exposure of Stock Selection Using NLP from December 2004 to March 2021

Looking at the long-short exposure of the NLP stock selection trading strategy, it is clear that the NLP trading strategy seems to have relatively good predictive power. This is because, throughout most of the 16 years of backtesting, the strategy was shorting a certain portion of stocks in the S&P 500. As a result, the strategy achieved the original goal of enhancing the buy and hold market index strategy by identifying which stocks are likely to underperform in the coming year.

3. **How is the performance of "Pairs Trading With PCA and Clustering" strategy?**

   Although we have finished the backtesting on Quantopian, after the shutdown, we replicated the results on QuantConnect. Originally, we planned to do a complete backtesting from December 2004 to March 2021; however, due to the insufficient computational power and the limitation of total number of trades in a backtest, we split it into 4 parts:

   (a) December 1^{st}, 2004 to December 31^{st}, 2008 (See Figure 35)

   (b) January 1^{st}, 2009 to December 31^{st}, 2012 (See Figure 36)

   (c) January 1^{st}, 2013 to December 31^{st}, 2016 (See Figure 37)

   (d) January 1^{st}, 2017 to March 31^{st}, 2021 (See Figure 38)

   Note that, although we had to split the whole duration into 4 periods due to the technical issues, we still need to start every backtest on the first trading day in January or July to select pairs for the trading.

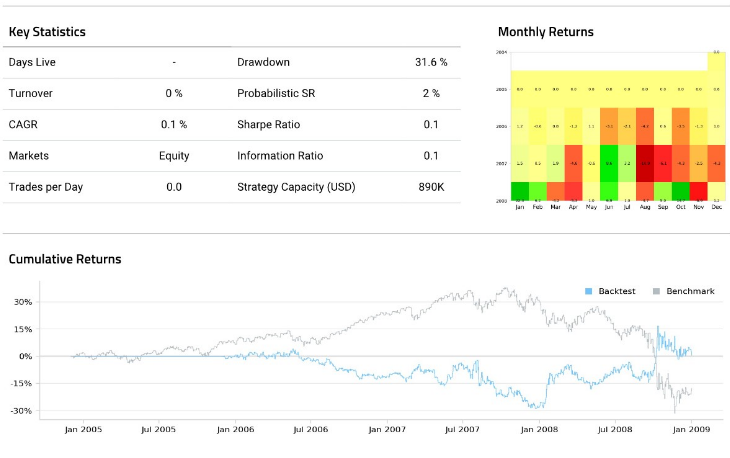


Figure 35: Backtest Result of Pairs Trading Strategy with PCA and DBSCAN from December 2004 to December 2008

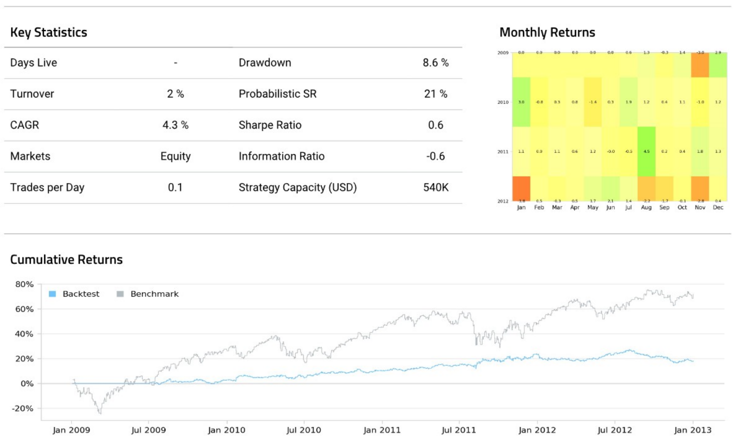


Figure 36: Backtest Result of Pairs Trading Strategy with PCA and DBSCAN from January 2009 to December 2012

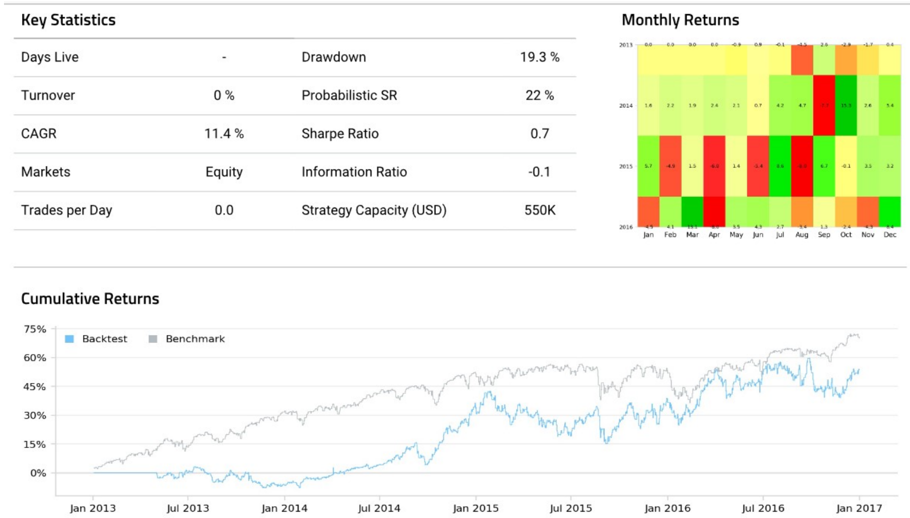


Figure 37: Backtest Result of Pairs Trading Strategy with PCA and DBSCAN from January 2013 to December 2016

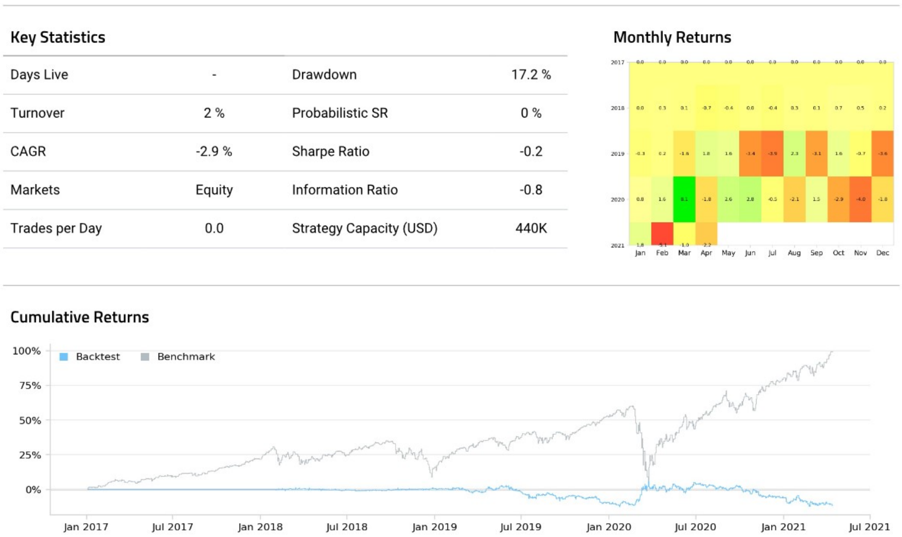


Figure 38: Backtest Result of Pairs Trading Strategy with PCA and DBSCAN from January 2017 to March 2021

After doing the calculations on its equity values over time, the summary statistics of pairs trading is provided in Table 5.

Table 5: Summary of Pairs Trading With PCA and Clustering

CAGR	Sharpe Ratio	Max Drawdown
2.97%	0.22	31.6%

Although the overall summary statistics for Pairs Trading is poor, we noticed that the strategy has a significant crisis alpha as shown in Figures 39, 39, and 39. As it also had a positive return over time, holding a portion of this strategy could help offset the big drawdown from other components during the crises.

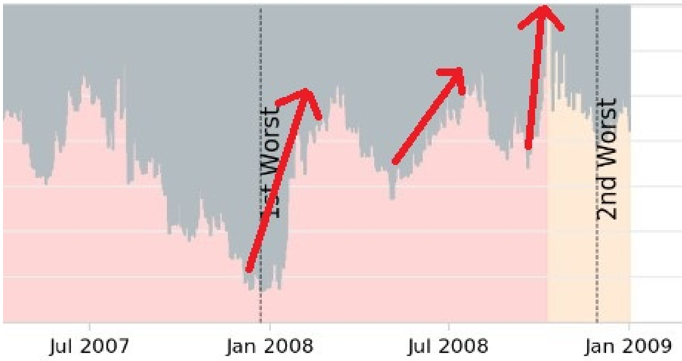


Figure 39: Pairs Trading Result From 2007 to 2008 (1^{st} Part of The Global Financial Crisis)

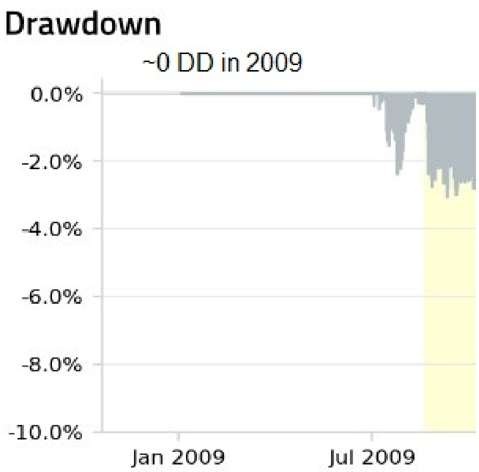


Figure 40: Pairs Trading Result From 2007 to 2008 (2^{nd} Part of The Global Financial Crisis)

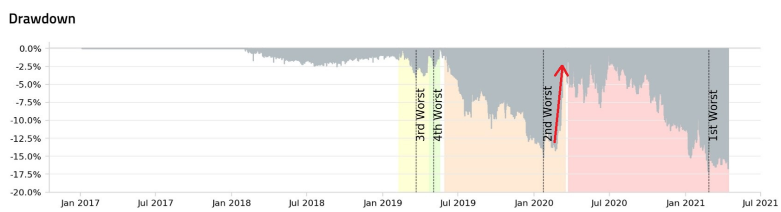


Figure 41: Pairs Trading Result In 2020 (The Coronavirus Crisis)

To conclude, Pairs Trading Strategies is a bad strategy in the strategy-wise scope. However, having this strategy is beneficial to be a component inside a portfolio as it could offset the risk from other "profitable" strategies during the downturns.

4. **Is there any discrepancy between the backtesting results from QuantConnect and Quantopian? If yes, why and how do we solve it?**

   As we already finished the Machine Learning model-free Harry Browne Permanent Portfolio strategy before the shutdown of Quantopian, we also obtained some results from Quatopian. On Quantopian, it, with 2x leverage, had a 19.22% Compounded Annual Growth Rate (CAGR), a Sharpe Ratio of 1.26, and the Max Drawdown of 21.83% over 188 months of the backtesting with the initial capital of 10 million U.S. Dollars from December 2004 to August 2020.

   It is clear that there are some discrepancies between two results from the platforms, and we believe in the following two reasons:

   (a) The transaction costs on QuantConnect may be closer to the one in reality as we assumed a $1 fixed cost per trade on Quantopian, so the returns from QuantConnect is lower.

   (b) The two platforms are using two different data source. For Quantopian, they are providing their own data as they are a hedge fund; on the other hand, QuantConnect's data is from a data provider named QuantQuote.

   As the information provided by QuantConnect is more transparent (e.g., the data source and the source code of the backtesting system), we believe in the results from QuantConnect more. Nonetheless, the results in both platforms showcase a significantly better strategy compared to all-in the capital into SPY.

5. **How did we determine the hyper-parameters in the model?**

Stock Selection Using NLP

In the strategy, for every new 10-K report we measured its sentiment similarity score with the previous year's 10-K report. The idea behind this is that if the sentiment similarity is low then the company is undergoing a lot of uncertainty at the moment, which normally negatively impacts the price of the stock in the coming year. So if a company's recent 10-K report has a similarity score greater than the threshold then we long the stock until next year, otherwise we short the stock. However, the question becomes what threshold should the score be so that we can maximize the Sharpe ratio.

On QuantConnect, we conducted a grid search on the threshold from 0.50 to 1.00 with the step size of 0.01. As Figure 42 showed, the stock performance in terms of the equity curve could differ a lot with respect to the threshold. The reason we started with 0.5 is because if the threshold is too low then it becomes meaningless since the strategy will just long all the stocks.

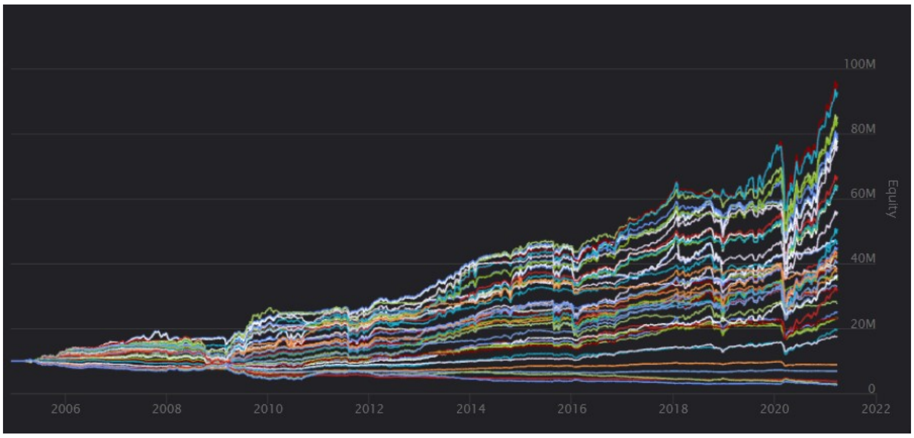


Figure 42: Equity Curve for Stock Selection Using NLP for Different Thresholds from Decemeber 2004 to March 2021

From Figures 43, 44, and 45, we finally determined to set the threshold at 0.83 as it had the highest return, one of the highest Sharpe Ratio, and one of the lowest Max Drawdown. Besides, in terms of sensitivity, it is stable as the points next to it had roughly the same return, Sharpe Ratio, and the Max Drawdown.

Figure 43: CAGR for Stock Selection Using NLP for Different Thresholds from Decemeber 2004 to March 2021

Figure 44: Sharpe Ratio for Stock Selection Using NLP for Different Thresholds from December 2004 to March 2021

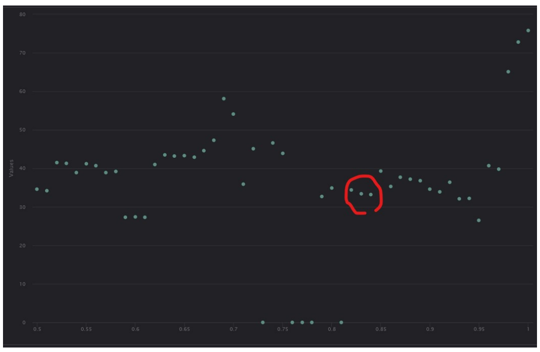


Figure 45: Max Drawdown for Stock Selection Using NLP for Different Thresholds from December 2004 to March 2021

Pairs Trading With PCA and Clustering

In the pairs trading strategy, there are three hyper-parameters such us entry Z-score, exit Z-score, and how long to re-evaluate the co-integration relationship. In the following paragraphs, we introduced it one by one:

(a) Exit Z-Score:

In terms of the Z-Scores, we first fixed the entry Z-Score to be 1.0 to fine-tune the exit Z-Score as we believed that we could lose a lot of money by not exiting the trades at the right time. For the entry Z-Score, we tried out different values and obtained the following results shown in Figures 46 and 47:

Back Testing Result			
Performance	Total Returns		48.32%
	Specific Returns		60.48%
	Common Returns		-7.68%
	Sharpe		0.28
	Max Drawdowns		-8.62%
	Volatility		0.07
Risks	Sector Exposures		Managed
	Style Exposures		Not Managed
	Turnover	8.83%	Not Managed
	Beta to SPY	-0.03	Not Managed
	Postion Concentration	27.09%	Not Managed
	Net Dollar Exposure	-8.48%	Not Managed

Z score exit signal= 0.2, -0.2
Z score entry signal = 1.0, -1.0
Idle cash not utilized
Pairs-selecting interval: 6 months

CAGR: 3.7351%
MAR Ratio: 0.4333

Figure 46: Summary Table for Pairs Trading With PCA and Clustering for Exit Z-Score = +0.2 and -0.2

Z score entry signal = +1.0, -1.0	
Z score exit signal = +0.1, -0.1	
Total Returns	52.27%
Specific Returns	62.29%
Common Returns	-6.28%
Sharpe	0.34
Max Drawdowns	-8.43%
Volatility	0.07
Z score exit signal = 0.0	
Total Returns	50.97%
Specific Returns	56.86%
Common Returns	-3.89%
Sharpe	0.34
Max Drawdowns	-9.36%
Volatility	0.06

Figure 47: Summary Table for Pairs Trading With PCA and Clustering for Different Exit Z-Scores

By trying out different values, we finally fixed our exit Z-score to be **+0.1 and -0.1** for unwinding the position as it generated the best result.

(b) Entry Z-Score:

After fixing the exit Z-Score at +0.1 and -0.1, we evaluate the performance for the entry Z-Score for (+1.0, -1.0), (+0.9, -0.9), (+0.7, -0.7), and (+0.5, -0.5). It turned out that setting the entry Z-Score at **+0.5 and -0.5** could yield the best result as Figure 48 showed.

Z score entry signal = +1.0, -1.0	
Z score exit signal = +0.1, -0.1	
Total Returns	52.27%
Specific Returns	62.29%
Common Returns	-6.28%
Sharpe	0.34
Max Drawdowns	-8.43%
Volatility	0.07
Z score exit signal = 0.0	
Total Returns	50.97%
Specific Returns	56.86%
Common Returns	-3.89%
Sharpe	0.34
Max Drawdowns	-9.36%
Volatility	0.06

Z score exit signal = +0.1, -0.1	
Z score entry signal= +0.9, -0.9	
Total Returns	51.21%
Specific Returns	62.55%
Common Returns	-7.08%
Sharpe	0.34
Max Drawdowns	-9.32%
Volatility	0.07
Z score entry signal = +0.7, -0.7	
Total Returns	62.68%
Specific Returns	66.33%
Common Returns	-2.28%
Sharpe	0.49
Max Drawdowns	-8.37
Volatility	0.07

Setting z score = 0.1 and –0.1 as exit signals and z score = 0.5 and -0.5 as entry signals boosts the returns from 48.32% to 69.36%.

Back Testing Result			
Performance	Total Returns		69.36%
	Specific Returns		67.92%
	Common Returns		0.75%
	Sharpe		0.58
	Max Drawdowns		-7.57%
	Volatility		0.07
Risks	Sector Exposures		Managed
	Style Exposures		Managed
	Turnover	10.37%	Not Managed
	Beta to SPY	-0.03	Not Managed
	Postion Concentration	28.76%	Not Managed
	Net Dollar Exposure	-6.96%	Not Managed

Z score exit signals = 0.1, -0.1
Z score entry signals = 0.5, -0.5
Idle cash not utilized
Pairs-selecting interval: 6 months

CAGR: 5.0230%
MAR Ratio: 0.6635

Figure 48: Summary Table for Pairs Trading With PCA and Clustering for Different Entry Z-Scores

(c) Frequency for Co-integration Testing:

Lastly, as the statistical relationships between stocks might change due to different market condition or unexpected event, we needed to re-evaluate the pairs for a certain amount of time. Therefore, we consider 3-month, 6-month, 9-month and 12-month re-evaluation, and it finally turned out that **6-month** re-evaluation has the best result as shown in Figure 49

Change pairs-selecting interval.

Select Pairs Every 9 Months			
Performance	Total Returns		115.85%
	Specific Returns		153.41%
	Common Returns		-17.35%
	Sharpe		0.62
	Max Drawdowns		-19.81%
	Volatility		0.11
Risks	Sector Exposures		Not Managed
	Style Exposures		Not Managed
	Turnover	10.28%	Managed
	Beta to SPY	0.09	Not Managed
	Postion Concentration	34.48%	Not Managed
	Net Dollar Exposure	82.44%	Not Managed

Select Pairs Every 12 Months			
Performance	Total Returns		198.86%
	Specific Returns		178.88%
	Common Returns		2.10%
	Sharpe		1.03
	Max Drawdowns		-14.57%
	Volatility		0.10
Risks	Sector Exposures		Not Managed
	Style Exposures		Not Managed
	Turnover	10.63%	Managed
	Beta to SPY	0.12	Not Managed
	Postion Concentration	40.56%	Not Managed
	Net Dollar Exposure	100.78%	Not Managed

Z score exit signals = 0.1, -0.1; Z score entry signals = 1.0, -1.0; Idle cash evenly allocated to SHY, TLT, SPY and GLD

After testing multiple pairs-selecting intervals (3 months, 9 months, 6 months and 12 months), the results indicate that we should keep our original strategy, which yields significantly higher returns.

Figure 49: Summary Table for Pairs Trading With PCA and Clustering for Different Frequency for Co-integration Testing

6. **How is the performance of the final combined portfolio? Is it better or worse than the benchmark?**

   We conducted the grid search on finding the optimal trio $(w_1, w_2, 1 - w_1 - w_2)$ such that the combined portfolio could yield the best risk-adjusted return such as Sharpe Ratio and MAR Ratio. In the grid search, we set the step size to be 0.05 for both w_1 and w_2, and the equity curves, Sharpe Ratio, Max Drawdown, and CAGR are shown in Figures 50, 51, 52, and 53.

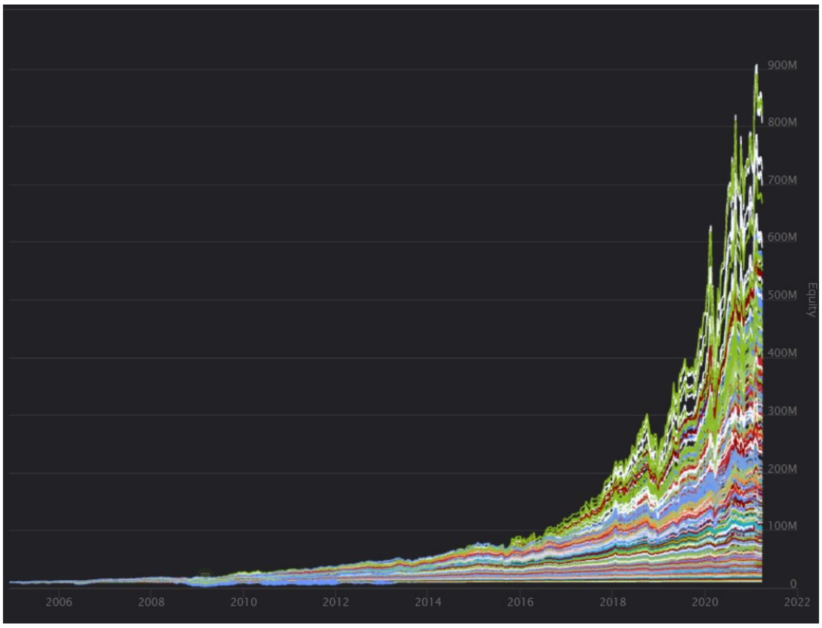


Figure 50: Equity Curve for The Combined Portfolio with Different Weights from Decemeber 2004 to March 2021

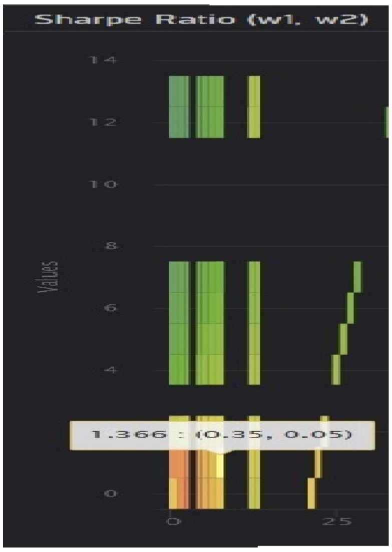


Figure 51: Sharpe Ratio for The Combined Portfolio with Different Weights from Decemeber 2004 to March 2021

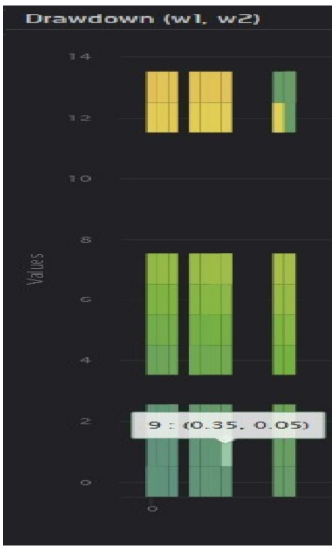


Figure 52: Max Drawdown for The Combined Portfolio with Different Weights from Decemeber 2004 to March 2021

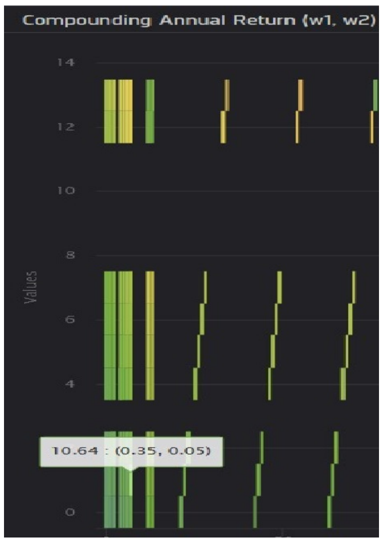


Figure 53: Annualized Return for The Combined Portfolio with Different Weights from Decemeber 2004 to March 2021

As our goal is to find the optimal combined portfolio with high risked-adjust returns, we ranked the grid search result according to Sharpe Ratio and obtained the following top 20 results out of 200 possibilities:

Table 6: Summary of Combined Portfolio

Rank	w_1	w_2	w_3	**CAGR**	**Sharpe Ratio**	**Drawdown**	**MAR Ratio**
1	0.10	0.00	0.90	5.37%	1.62	4.1%	1.31
2	0.05	0.05	0.90	6.17%	1.62	3.8%	1.62
3	0.15	0.00	0.85	6.14%	1.59	5.3%	1.16
4	0.10	0.05	0.85	6.95%	1.59	4.3%	1.62
5	0.00	0.05	0.95	5.38%	1.58	3.4%	1.58
6	0.05	0.00	0.95	4.59%	1.57	3.8%	1.21
7	0.15	0.05	0.80	7.68%	1.54	4.9%	1.57
8	0.20	0.00	0.80	6.78%	1.52	6.0%	1.13
9	0.20	0.05	0.75	8.46%	1.49	5.9%	1.43
10	0.25	0.00	0.75	7.44%	1.45	7.4%	1.01
11	0.25	0.05	0.70	9.16%	1.44	6.8%	1.35
12	0.30	0.00	0.70	8.37%	1.43	6.8%	1.23
13	**0.30**	**0.05**	**0.65**	**9.99%**	**1.41**	**7.8%**	**1.28**
14	0.35	0.00	0.65	9.07%	1.37	7.9%	1.15
15	0.40	0.05	0.55	11.65%	1.37	10.0%	1.17
16	0.35	0.05	0.60	10.64%	1.37	9.0%	1.18
17	0.40	0.00	0.60	9.93%	1.36	8.4%	1.18
18	0.45	0.05	0.50	12.44%	1.34	11.2%	1.11
19	0.45	0.00	0.55	10.61%	1.32	9.5%	1.12
20	0.50	0.00	0.50	11.58%	1.31	10.4%	1.11

For our Combined Portfolio Strategy, we chose the one that has the 13^{th} highest Sharpe Ratio. The reason for our choice is because we wanted to find the right balance between CAGR and Maximum Drawdown. Our goal is to beat the U.S. stock index while minimizing the risk such as the MDD. However, we cannot achieve our minimum return requirement if we select a model in the top 10 for Sharpe Ratio. These models are already using 2x leverage, which makes it difficult to achieve a higher return with them given that the maximum leverage accepted for most of the brokers in the market is 2x.

Note that although the result of Machine Learning-based Harry Browne Permanent Portfolio is already impressive on its own, having a diversification with other strategies could indeed improve its Sharpe Ratio and MAR Ratio by 18.0% and 17.5% respectively.

3 Discussion

3.1 Performance Comparison with S&P 500 Index

Table 7: Summary of All Models Performance

Model	CAGR	Sharpe Ratio	MAR Ratio	Max Drawdown
Baseline Model (S&P 500 Index)	9.85%	0.62	0.18	55.1%
Model 1 (HBLSTM)	**20.40%**	1.25	1.13	18.0%
Model 2 (SSNLP)	14.80%	1.06	0.44	33.4%
Model 3 (PTPC)	2.97%	0.22	0.09	31.6%
Model 4 (Combined Portfolio)	9.99%	**1.41**	**1.28**	**7.8%**

The goal of this project was to create a portfolio that utilizes machine learning to get higher returns with lower risks in comparison to the S&P 500 Index. We have successfully achieved this goal as Model 1 and 2 both have a higher CAGR, Sharpe Ratio, MAR Ratio, and a lower MDD than the Baseline Model. Our results suggest that different machine learning techniques can be effective in enabling a trading strategy to outperform the U.S. stock market index. However, this wasn't true for all of our trading strategies. Our Model 3 failed to outperform the Baseline Model, with only a 2.97% annual return.

Despite this, we were still able to combine all three strategies into Model 4, which has the highest Sharpe Ratio and MAR Ratio among all our models. The weights of Model 4 include 30% in Model 1, 5% in Model 2, and 65% in Model 3. Although the annual return of Model 4 is just barely above the annual return of the S&P 500 Index, its Maximum Drawdown is almost seven times smaller than the Baseline Model. This means that, on average, it only takes about 9 months for the trading strategy to recover from its trough. From an investor's perspective, this is very ideal because the waiting time for the portfolio rebounded from a downswing isn't too long.

For all these reasons, we believe that we did achieve our goal by combining Machine Learning models and traditional investment strategies to outperform the U.S. stock index.

3.2 Performance during Black Swan Events

In this section, we defined Black Swan Events as the ones caused a drawdown more than 30% in the S&P 500 Index, and we identified two events: (1) 2007-2009 Global Financial Crisis, and (2) 2020 Coronavirus Pandemic Crisis.

In the following part, we will show the performance of each strategy during the 2 periods.

3.2.1 Machine Learning-based Harry Browne Permanent Portfolio

1. 2007-2009 Global Financial Crisis

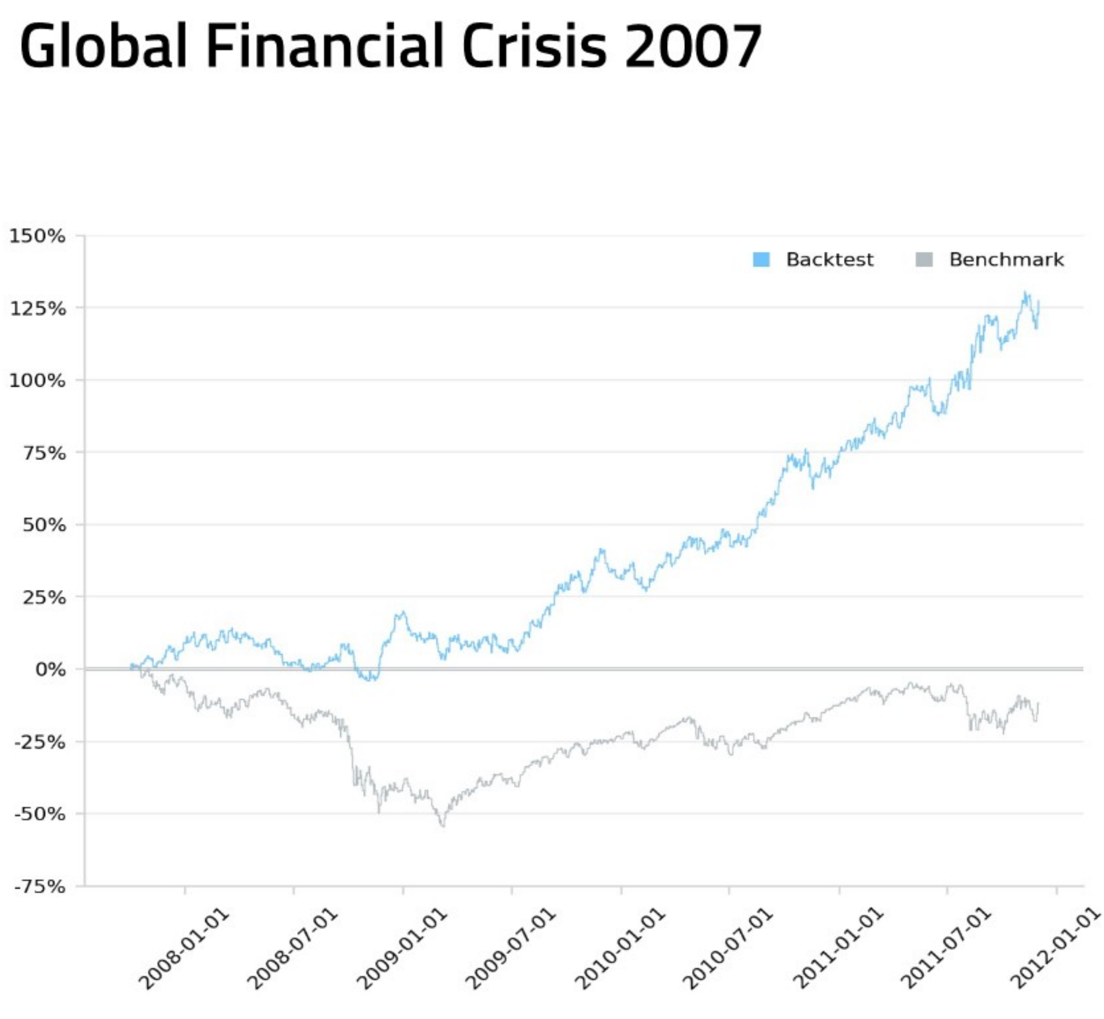


Figure 54: Performance of Machine Learning-based Harry Browne Permanent Portfolio During Global Financial Crisis

2. 2020 Coronavirus Pandemic Crisis

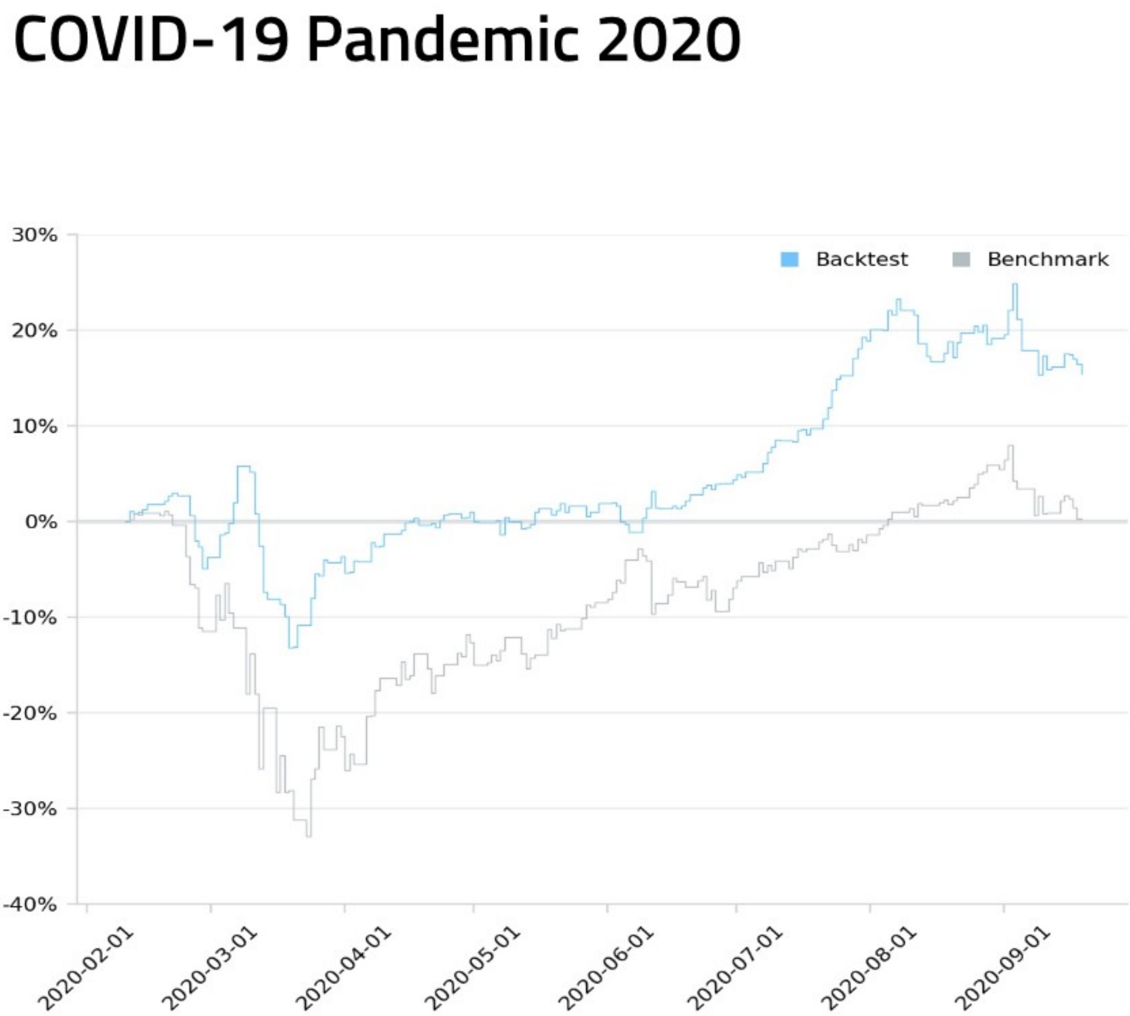


Figure 55: Performance of Machine Learning-based Harry Browne Permanent Portfolio During COVID-19 Pandemic

As shown in Figures 54 and 55, the Machine Learning based Harry Browne Permanent Portfolio was able to significantly outperform the benchmark index during both the Global Financial Crisis and the COVID-19 pandemic. This can be explained by diversification as the Harry Browne Permanent Portfolio invested in both risky and safe assets. Additionally, the LSTM prediction model enables for a dynamic weighting strategy that captures the market conditions.

3.2.2 Stock Selection Using NLP

1. 2007-2009 Global Financial Crisis

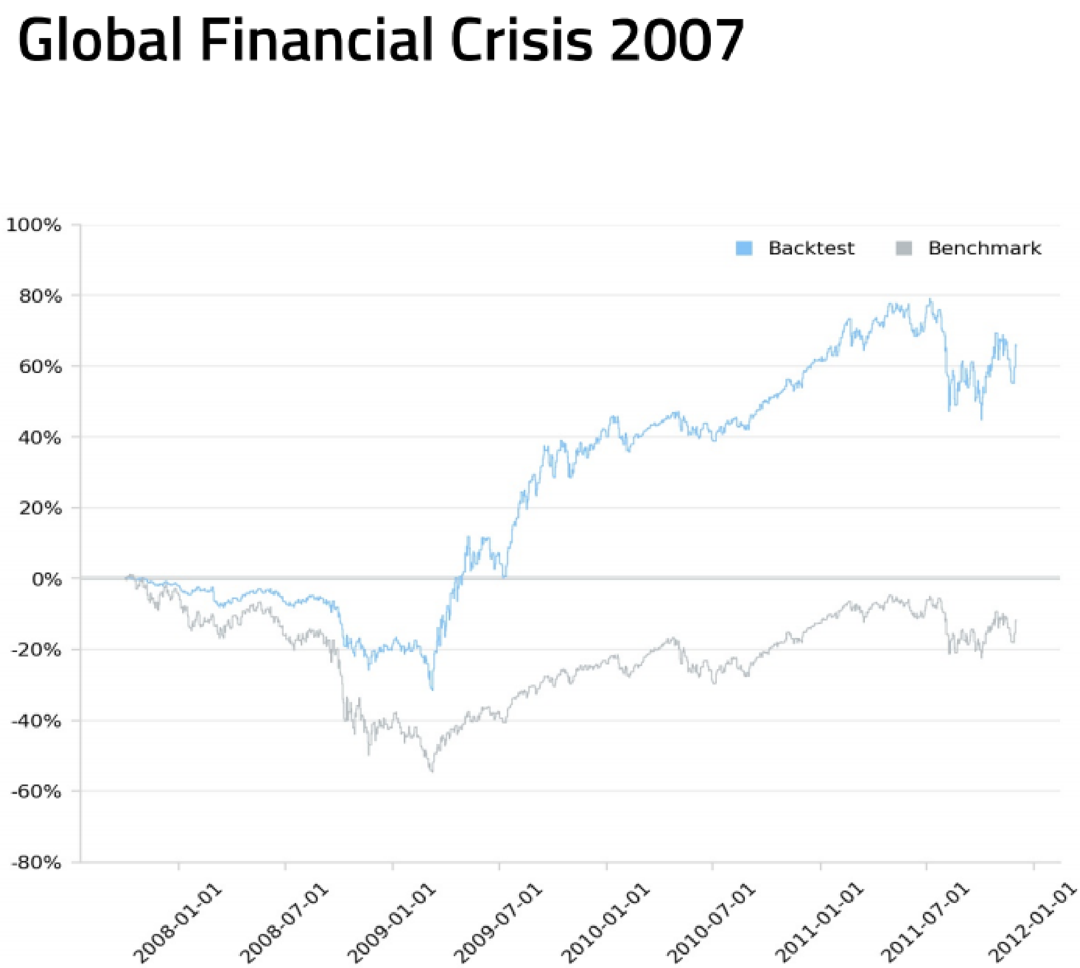


Figure 56: Performance of NLP Stock Selection Strategy During Global Financial Crisis

2. 2020 Coronavirus Pandemic Crisis

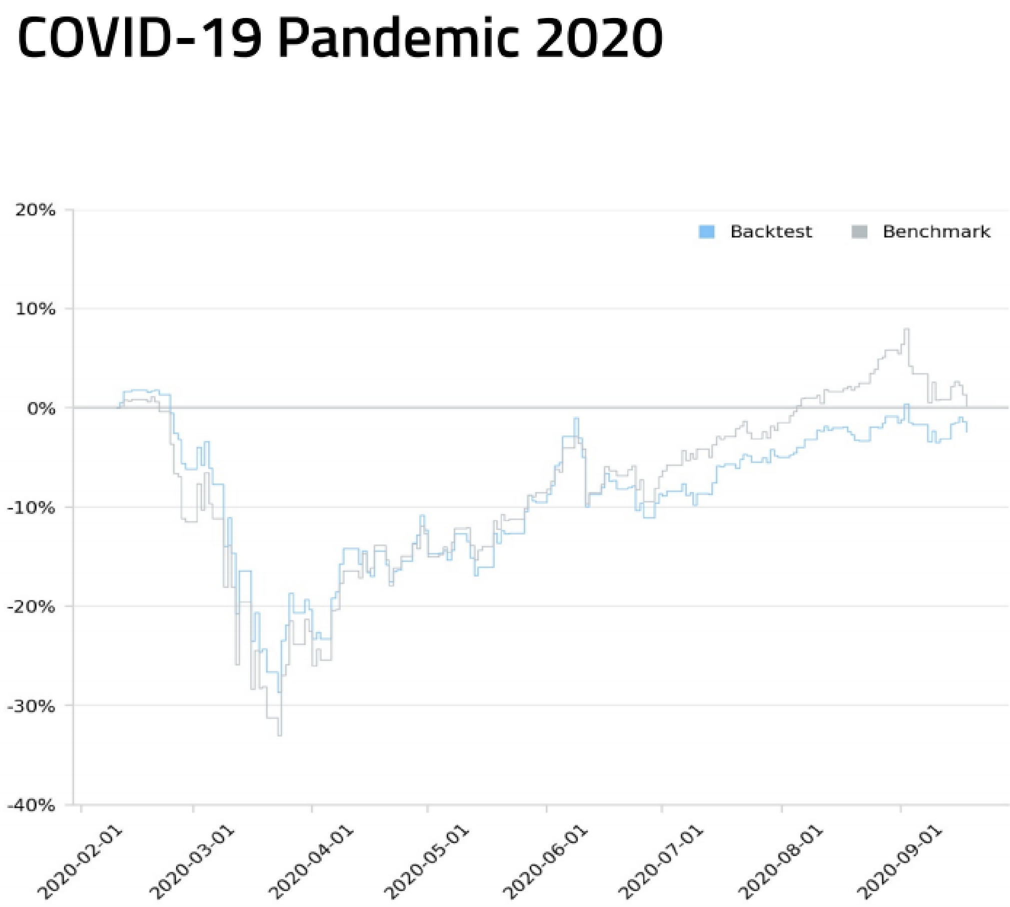


Figure 57: Performance of NLP Stock Selection Strategy During COVID-19 Pandemic

As shown in Figure 56, during the Global Financial Crisis, the NLP Stock Selection Strategy was able to significantly outperform the benchmark index. Meanwhile, during the 2020 COVID-19 pandemic as illustrated in Figure 57, the strategy was able to perform a little bit better than the US stock market index during the market crash. However, during the recovery period, the trading strategy performed worse. We believe the reason behind the strategy's poor performance during a pandemic is due to poor timing. The NLP Stock Selection Strategy relies on annual 10-K reports to make future judgments regarding the stock's performance in the coming year. However, given the sudden nature of the pandemic, companies failed to adequately assess the dangers posed by the COVID-19 pandemic in their annual reports.

3.2.3 Pairs Trading with PCA and Clustering

1. 2007-2009 Global Financial Crisis

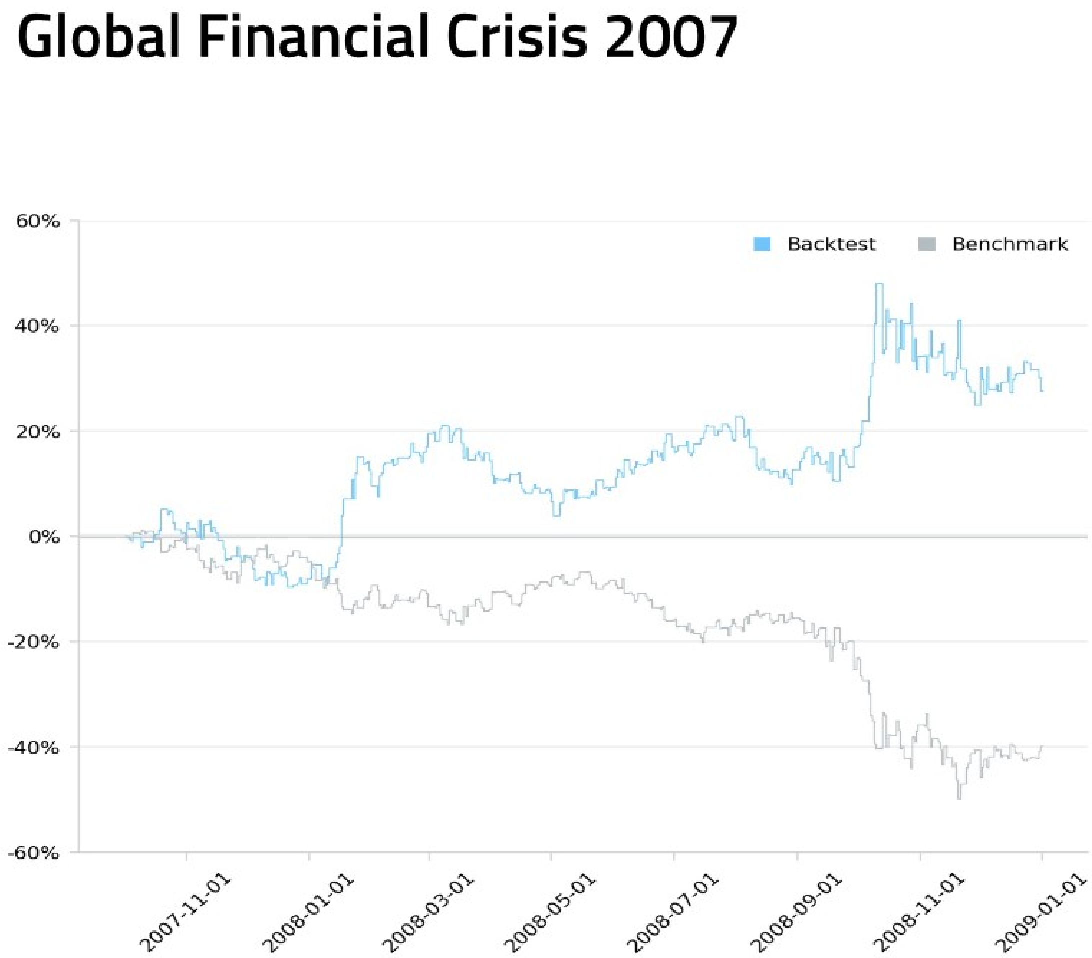


Figure 58: Performance of Machine Learning-based Harry Browne Permanent Portfolio During Global Financial Crisis (2007-2008)

Global Financial Crisis 2007

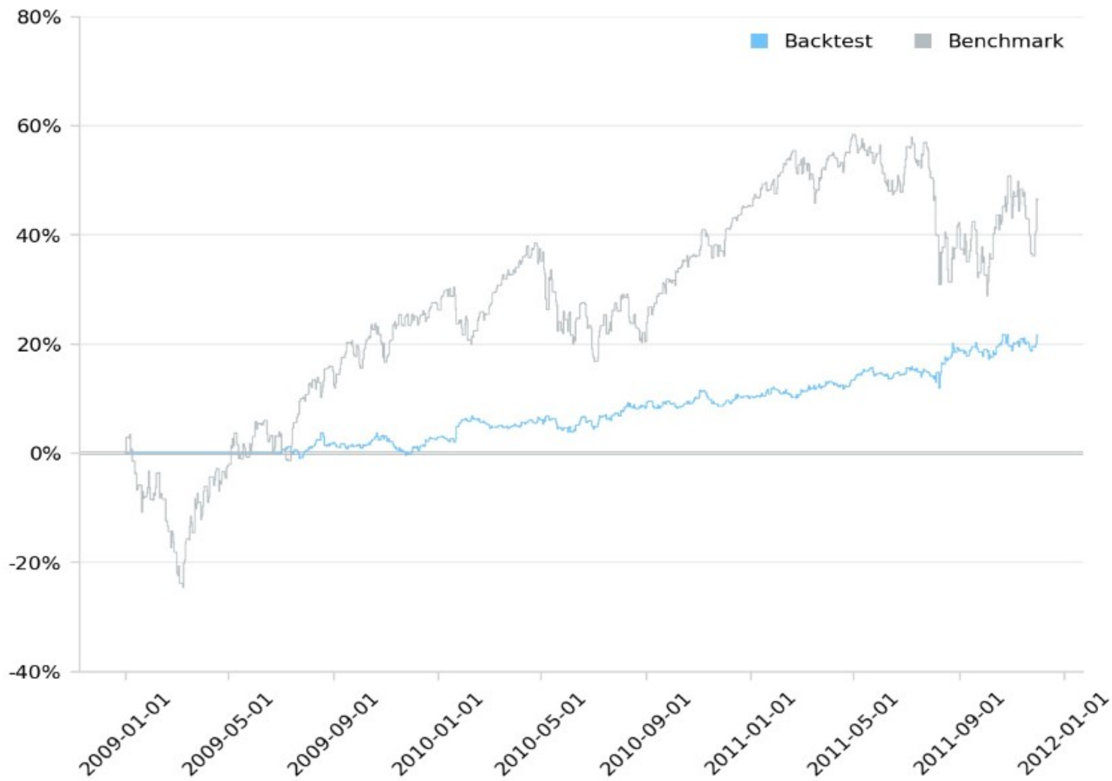


Figure 59: Performance of Machine Learning-based Harry Browne Permanent Portfolio During Global Financial Crisis (2009 Onwards)

2. 2020 Coronavirus Pandemic Crisis

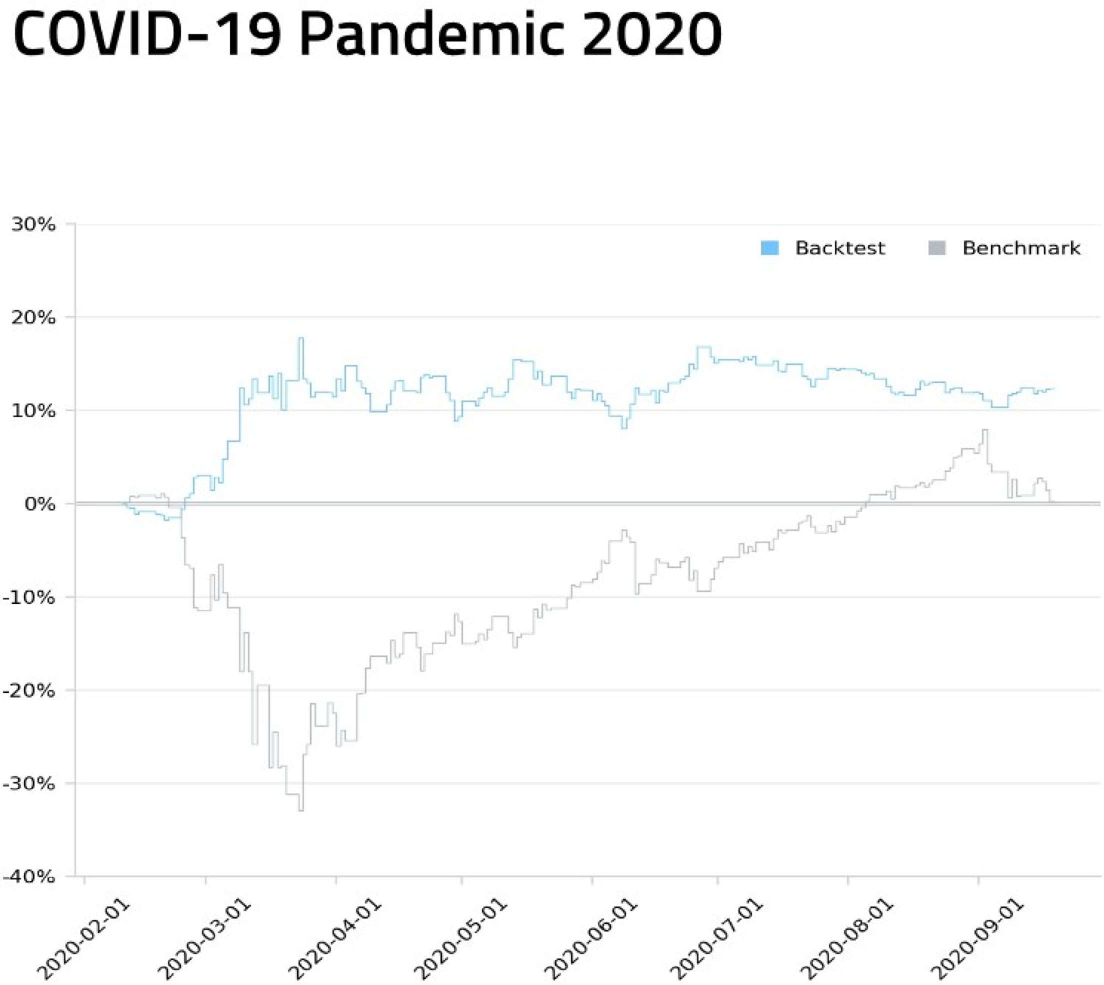


Figure 60: Performance of Machine Learning-based Harry Browne Permanent Portfolio During COVID-19 Pandemic

As shown in Figure 58, during the first two years of Global Financial Crisis, the Pairs Trading with PCA and Clustering Strategy was able to outperform the benchmark, S&P 500 Index, by a lot during the market crash. However, during the recovery stage, the strategy didn't do as well as the benchmark as the strategy is a market-neutral strategy that is supposed to generate positive alpha regardless of the market conditions. However, its overall return is not that high.

Meanwhile, during the 2020 COVID-19 pandemic as illustrated in Figure 60, the strategy also outperformed the S&P 500 Index. Moreover, the strategy didn't face any major downswings during this period. Therefore, we concluded that this strategy is a good crisis alpha generator for hedging.

3.2.4 Combined Portfolio

1. 2007-2009 Global Financial Crisis

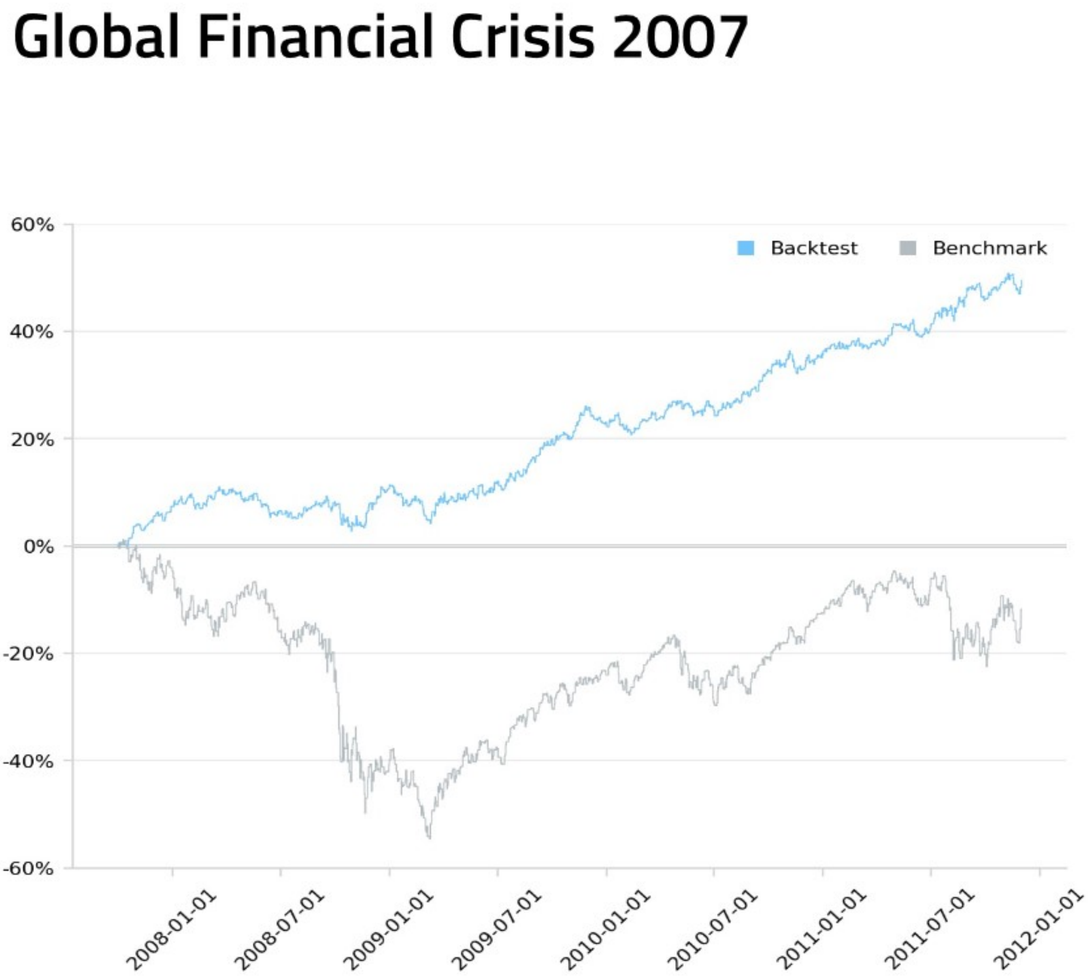


Figure 61: Performance of Combined Portfolio During Global Financial Crisis

2. 2020 Coronavirus Pandemic Crisis

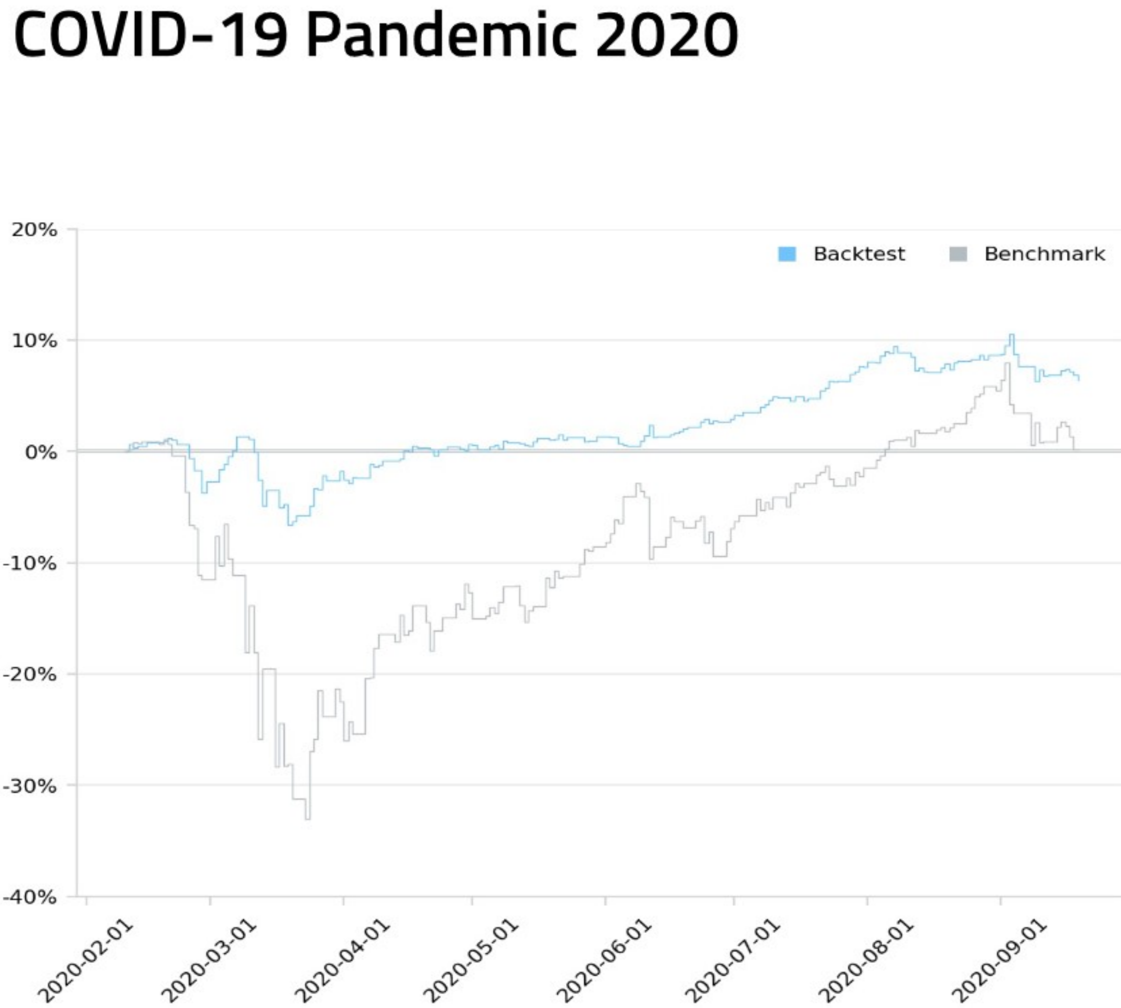


Figure 62: Performance of Combined Portfolio During COVID-19 Pandemic

As shown in Figures 61 and 62, the Combination Portfolio was able to outperform the S&P 500 Index during both the Global Financial Crisis and COVID-19 Pandemic. Aside from this, the strategy experienced no significant downside during these two periods. Overall, we coould conclude that the Combined Portfolio is earning profits regardless of the crises.

3.3 Limitations and Challenges

Although the strategies perform well compared to the S&P 500 Index, there are still some limitations for each strategy that are worth discussing.

3.3.1 Machine Learning-based Harry Browne Permanent Portfolio

One possible limitation for the LSTM model to predict the stock is the inability to capture the black swan events when the stocks plummeted by 30% of skyrocketed over 20%. For example, as Figure 63 showed, the LSTM model did not capture both the rise in January 2020 and the plummet in March 2020 in the S&P 500 Index. This may under- or over-estimate the expected returns, underestimate the variance and lose money accordingly.

Figure 63: Illustration of The Inability for LSTM to Predict Black Swan Events

3.3.2 Stock Selection Using NLP

For their backtesting system, QuantConnect has a two times maximum leverage limit. For our trading strategy, whenever the leverage ratio gets close to two, a margin call would occur, and we were forced to liquidate some of our stock positions to get more cash and reduce the leverage. However, this negatively impacted the performance of the strategy by reducing our overall portfolio holdings leading to fewer returns. We encountered this problem because this investment strategy lacks a proper exit condition. Although we can avoid margin calls and forced liquidation from happening by not using leverage, this would negatively impact the overall CAGR of the strategy. Aside from this, notice that even though our strategy uses two times leverage, it still has a lower Max Drawdown, at 33%, compared to the one for S&P 500 Index, at 55.10%. This suggests to us that the strategy is effective in mitigating systemic market risk by shorting certain stocks.

The reason we don't have an exit condition is that the research paper we based our strategy on did not have a proper exit condition. According to the authors, the market was unable to process the negative implications of a 10-K report with a low sentiment similarity score in the short term, so it is crucial to hold the stock position long enough so that stock prices reflect this "hidden" information. We can resolve this issue by creating a liquidate order for all stocks after a certain period has passed. By doing this, we can potentially reduce our risk exposure to market crashes, and avoid the margin calls, which can lead to non-optimal sell orders.

3.3.3 Pairs Trading with PCA and Clustering

For the Pairs Trading with PCA and Clustering strategy, we have two possible limitations:

1. Clustering:

   The following reasons are why we adopted DBSCAN as the clustering algorithm:

   (a) No need to specify the number of clusters in advance: Compared to the partitional clustering such as K-means Algorithm, we were not required to specify the number of clusters in the model. Even we knew how many clusters the top 1500 stocks belong to, the number of clusters would change over time due to the ever-changing market condition and the different stocks that we picked according to the market cap, not to mention that it is impossible to know how many clusters given any time step. Although we could use the Elbow Method to determine the proper K, it is not efficient to do so in the fast-paced financial market and the determination of K is no longer machine-learning based but a subjective opinion of a human beings.

   (b) Robustness to outliers: Compared to many clustering algorithms such as K-means and Hierarchical Clustering with Single-Link Proximity, as DBSCAN has a parameter requiring minimum number of points in a cluster, it could effectively eliminate the outliers effect. As the financial data is noisy compared to image data or textual data, it is important to keep the outliers outside the scope of analysis.

   However, there are still some disadvantages for using DBSCAN:

   (a) The performance to cluster data with different density is poor: As the density and minimum number of points in the cluster and the distance across each cluster are fixed and not cluster-specific, DBSCAN worked poorly for the data with different density in different clusters. In terms of financial data, the situation in the financial data could be the tech-stock cluster where the market capitalization for top technology stocks could differ by a lot most of the time. By setting the fixed distance, we had to strike the balance between seizing the opportunity in tech-stocks and spending more time on evaluating the pairs in the cluster if we considered increasing the distance.

   (b) It is not deterministic for the boundary points: As the only requirement is to find the point(s) which satisfies the distance requirement, for the points lie on the boundary of two clusters, they might be grouped into different clusters different time. Hence, the backtest result might be different over time.

2. Dimensionality Reduction:

   As the price data over the past two years could be in 500 dimensions and it might cause some problems to conduct the clustering on the high dimensional spaces such as the curse of dimensionality [29]. However, as the implicit assumption for PCA is that the variance is highly correlated to the information, there is a limitation of PCA where the dimension dropped by PCA might have some important features. Therefore, before applying PCA, we should be confident that the remaining principal components are useful and meaningful for conducting the further analysis.

3.3.4 Combined Portfolio

In the section, we considered an optimal fixed weight over the past 16 years by grid search. By doing so, we implicitly assume that all possible market conditions have been represented in the past 16 years. However, the implicit assumption does not hold as three major black swan events in the history: (1) the Great Depression in 1920s to 1930s, (2) the Black Monday in 1987, and (3) the dot-com bubble in the early 2000s are not captured in the data in the past 16 years. Besides, allocating capital into different strategy with a given fixed weights is impossible to be the most optimal choice as the financial market is dynamic, ever-changing and hard-to-predict. Therefore, we believe one limitation for our combined portfolio is lacking flexibility to allocate different weights to different strategies at different time. The possible improvement could be applying the portfolio theory based on the returns and variance-covariance matrix or implementing a reinforcement learning model and training an agent to allocate the capital.

4 Conclusion

4.1 Summary of Achievements

In our project, we developed three profitable strategies based on Machine Learning models including:

1. Long-Short Term Memory (LSTM) Model

   In our project, we implemented LSTM to predicts the future 1-month stock price. Then, we computed the expected return and the variance-covariance matrix for each asset in the Harry Browne Permanent Portfolio. The final step is to determine the weight od each asset based on the Markowitz Mean-Variance Portfolio Theory.

2. Natural Language Processing (NLP)

   NLP was used for analyzing the 10-K reports of companies to determine which stocks are likely to underperform relative to the general U.S. stock market index in the coming year. We also used the sentiment similarity scores to long and short the S&P500 stocks to increase the strategy's return and lower the Max Drawdown relative to the passive strategy of buy-and-hold the S&P 500 index.

3. Principal Component Analysis (PCA) and Density-Based Spatial Clustering of Applications with Noise (DBSCAN)

   After extract 2-year data, we implemented PCA to reduce the dimensionality of stock prices into 50 principal components in order to make DBSCAN, the clustering algorithm, easier to identify suitable pairs of stocks to conduct pairs trading. Overall, DBSCAN will cluster stocks into different group according to the principal components, market cap, and the financial health of each firm. Then, we conducted the Co-intergation Test to examine proper pairs to do trades.

After developing 3 Machine Learning-based trading strategies, we combined them into a Combined Portfolio with the optimal weight found in grid search in order to maximize the risk-adjusted returns, such as Sharpe Ratio and MAR Ratio, while maintaining an annual return greater than the performance of the S&P 500 Index.

Aside from developing trading strategies, we also developed a visualization system with different types of graphs. It is useful for analyzing the overall performance that is not captured by the visualization system on QuantConnect. Lastly, we also execute our Combined Portfolio by the automated system on QuantConnect. We started assessing the backtesting result in the real-time market via an IB paper account with the initial capital of 1 million U.S. dollars starting from April 14^{th}.

4.2 Future Work

Although the project seems to be complete with a better result compared to the U.S. stock index after applying grid search to combine all 3 strategies we developed, we might consider extending our works in the following aspects:

1. **Reinforcement Learning-based Strategy Combination** (See Section 11.1 Reinforcement Learning for further details): As mentioned in Discussion, allocating capital to strategies dynamically according to the macroeconomics condition and the correlation between the strategy could be a better way to combine strategies. Besides, building a Reinforcement Learning agent that could learn from interacting with the environment is also another way to optimize our final results. By building a proper environment for the agent to interact with and setting the returns made as the feedback to the agent, we believe that the overall portfolio will perform better.

2. **Stock Price Prediction in Active Investment by LSTM** (See Section 11.2 Stock Price Prediction in Active Investment by LSTM for further details): In our project, we simply used LSTM models to predict the expected returns and variance-covariance matrix by the 1-month predicted values on the 4 components in Harry Browne Permenent Portfolio. Moreover, the technique could be extended to all stocks. After predicting the prices in the coming month, we could select those stocks with significant, in terms of absolute value, t-stat, $\frac{expected\ return}{standard\ deviation\ of\ returns}$, to long if t-test is positive or short if t-stat is negative.

3. **Construction of Website for Investors to Use** (See Section 11.3 Construction of Website for further details): After developing profitable strategies, it will be convenient for investors to access to our strategies via a website. Therefore, by building a website, we could influence more people, encourage more people to invest and bring those capital to the firms bringing social benefits through investing activities.

4. **A Custom Backtesting System to Change the Trading Universe Outside U.S.** (See Section 11.4 Custom Backtesting System for further details): In this project, we implemented our strategies on QuantConnect. Therefore, our trading universe has been restricted to the U.S. equity. Although the U.S. market is mature for conducting the algorithmic trading strategies, we believe that building strategies in other developing country markets such as Vietnam and Indonesia might be challenging yet more profitable due to the market inefficiency. Besides, having a custom backtesting system can best utilize the visualization system we built as one of the features.

5 References

[1] A. Mittal. Understanding RNN and LSTM. (2020, September 17). [Online]. Available: https://towardsdatascience.com/understanding-rnn-and-lstm-f7cdf6dfc14e

[2] H. Xu, J. Li, L. Liu, Y. Wang, H. Yuan, and X. Wang, "Generalizable control for quantum parameter estimation through reinforcement learning," *npj Quantum Information*, vol. 5, no. 1, pp. 1–8, 2019.

[3] S. Chatterjee. How to train your machine: JPMorgan FX algos learn to trade better. (2020, September 17). [Online]. Available: https://www.reuters.com/article/us-jpm-trading-machines/how-to-train-your-machine-jpmorgan-fx-algos-learn-to-trade-better-idUSKCN1S61JG

[4] A. Bigiotti and A. Navarra, *Optimizing Automated Trading Systems*, 01 2019, pp. 254–261.

[5] B. Tabbaa. The Rise and Fall of Knight Capital — Buy High, Sell Low. Rinse and Repeat. (2020, September 17). [Online]. Available: https://medium.com/dataseries/the-rise-and-fall-of-knight-capital-buy-high-sell-low-rinse-and-repeat-ae17fae780f6

[6] J. Chen. Permanent Portfolio. (2020, September 17). [Online]. Available: https://www.investopedia.com/terms/p/permanent-portfolio.asp

[7] ETF.com. Find the Right ETF - Tools, Ratings, News. (2020, September 17). [Online]. Available: http://www.etf.com/

[8] SPDR. Bringing the gold market to investors. (2020, September 17). [Online]. Available: https://www.spdrgoldshares.com/

[9] S. Arora. Supervised vs Unsupervised vs Reinforcement. (2020, September 17). [Online]. Available: https://www.aitude.com/supervised-vs-unsupervised-vs-reinforcement/

[10] A. Stotz. Are financial analysts' earnings forecasts accurate? (2020, September 17). [Online]. Available: https://becomeabetterinvestor.net/are-financial-analysts-earnings-forecasts-accurate/

[11] macrotrends. S&P 500 Historical Annual Returns. (2020, September 17). [Online]. Available: https://www.macrotrends.net/2526/sp-500-historical-annual-returns

[12] Lazy-Portfolio-ETF. Harry Browne Permanent Portfolio: ETF allocation and returns. (2020, September 17). [Online]. Available: http://www.lazyportfolioetf.com/allocation/harry-browne-permanent/

[13] C. yu LEE, "Enhanced passive investing-harry browne permanent portfolio with forex strategies," 2020.

[14] C. Chalvatzis and D. Hristu-Varsakelis, "High-performance stock index trading: making effective use of a deep lstm neural network," *arXiv preprint arXiv:1902.03125*, 2019.

[15] L. Cohen, C. Malloy, and Q. Nguyen, "Lazy prices," *The Journal of Finance*, vol. 75, no. 3, pp. 1371–1415, 2020.

[16] H. He, J. Chen, H. Jin, S.-h. Chen, P. Wang, and T. Kuo, *Trading Strategies Based on K-means Clustering and Regression Models*, 01 2007.

[17] R. Adusumilli. DBSCAN Clustering for Trading. (2020, September 17). [Online]. Available: https://towardsdatascience.com/dbscan-clustering-for-trading-4c48e5ebffc8

[18] Seeking-Alpha. Enhancing Harry Browne's Permanent Portfolio. (2020, September 17). [Online]. Available: https://seekingalpha.com/article/4247711-enhancing-harry-brownes-permanent-portfolio

[19] Logical-Invest. Enhanced Permanent Portfolio Strategy. (2020, September 17). [Online]. Available: https://logical-invest.com/app/strategy/pp/enhanced-permanent-portfolio-strategy

[20] Optimized-Portfolio. Harry Browne Permanent Portfolio Review, ETFs, & Leverage. (2020, September 17). [Online]. Available: https://www.optimizedportfolio.com/permanent-portfolio/

[21] Z. Ding, C. W. Granger, and R. F. Engle, "A long memory property of stock market returns and a new model," *Journal of Empirical Finance*, vol. 1, no. 1, pp. 83–106, 1993. [Online]. Available: https://www.sciencedirect.com/science/article/pii/092753989390006D

[22] Z. Ding and C. W. Granger, "Modeling volatility persistence of speculative returns: A new approach," *Journal of Econometrics*, vol. 73, no. 1, pp. 185–215, 1996. [Online]. Available: https://www.sciencedirect.com/science/article/pii/0304407695017372

[23] K. Anadu, M. Kruttli, P. McCabe, and E. Osambela, "The shift from active to passive investing: Risks to financial stability?" *Financial Analysts Journal*, vol. 76, no. 4, pp. 23–39, 2020.

[24] MarketWatch. SPDR S&P 500 ETF Trust Overview. (2021, February 12). [Online]. Available: https://www.marketwatch.com/investing/fund/spy

[25] MarketWatch, iShares Russell 2000 ETF Overview. (2021, February 13). [Online]. Available: https://www.marketwatch.com/investing/fund/iwm

[26] V. Monga and E. Chasan, "The 109,894-word annual report," *The Wall Street Journal*, Jun 2015. [Online]. Available: https://www.wsj.com/articles/the-109-894-word-annual-report-1433203762

[27] M. Huang, Q. Bao, Y. Zhang, and W. Feng, "A hybrid algorithm for forecasting financial time series data based on dbscan and svr," *Information*, vol. 10, no. 3, p. 103, 2019.

[28] Lazy-Portfolio-ETF. Ray Dalio All Weather Portfolio: ETF allocation and returns. (2021, April 13). [Online]. Available: http://www.lazyportfolioetf.com/allocation/ray-dalio-all-weather/

[29] M. Köppen, "The curse of dimensionality," in *5th Online World Conference on Soft Computing in Industrial Applications (WSC5)*, vol. 1, 2000, pp. 4–8.

[30] Treelogic. Reinforcement Learning: Applications in Finance. (2020, September 17). [Online]. Available: https://financetalks.ie.edu/reinforcement-learning-applications-in-finance/

[31] A. Filos, "Reinforcement learning for portfolio management," *arXiv preprint arXiv:1909.09571*, 2019.

6 Appendix A: Meeting Minutes

6.1 Minutes of the 1^{st} Project Meeting

1. Arrangement

   - Date: April 29, 2020
   - Time: 3:00pm
   - Place: Room 3554
   - Present: Chih-yu LEE and Prof. David Paul ROSSITER
   - Absent: Matthew Johann Siy UY
   - Recorder: Chih-yu LEE

2. Approval of minutes

   - This was the first group meeting; thus, no minutes were required to be approved.

3. Report on progress

   - As Chih-yu has self-proposed this project, he was required to find other groupmate(s) to form a FYP group.
   - He suggested having a topic regarding Machine Learning in finance.

4. Discussion items

   - Machine Learning is a broad topic, so we should conduct the literature reviews in order to understand the status quo of the area.
   - There are different types of Machine Learning and we would like to develop at least one strategy correspondingly:
     - Supervised Learning
     - Unsupervised Learning
     - Reinforcement Learning
   - The research directions could be:
     - Develop a brand-new strategy by combining several Machine Learning models together.
     - Assess existing works concerning investment and trading using Machine Learning techniques, and boost its profitability by changing trading universe, model, or signal.

5. To-do lists before the next meeting

    - Find at least one suitable teammate and form a group.
    - Submit the outline of the project.
    - Read several research papers regarding the Machine Learning application in finance.

6. Meeting adjournment and arrangement of the next meeting

    - The next meeting will be at 4pm on May 6, 2020 at Room 3554.

6.2 Minutes of the 2^{nd} Project Meeting

1. Arrangement
   - Date: May 6, 2020
   - Time: 4:00pm
   - Place: Room 3554
   - Present: Chih-yu LEE and Prof. David Paul ROSSITER
   - Absent: Matthew Johann Siy UY
   - Recorder: Chih-yu LEE
2. Approval of minutes
   - The minutes of last meeting were approved without amendment and objection.
3. Report on progress
   - Chih-yu has tried to find some teammates from the Groupmate Finder; however, they already formed a group of 4 and rejected him.
   - He requested Prof. ROSSITER to recruit students by sending the email to those who are interested in doing FYP under the supervision of the professor.
   - We have set up several directions of the project:
     (a) Neural Networks for asset allocation
     (b) Sentiment Analyses on financial statements for price prediction
     (c) PCA and Clustering for pairs trading

4. Discussion items

   - As data is an important issue in this project, we decided to use Quantopian as it provides the tick-wise price data as well as macroeconomic and corporate financial data.
   - Chih-yu suggested we should spare some time for real-time trading to evaluate the performance of the project.
   - Chih-yu also mentioned that he noticed many research papers in the area have some common mistakes:
     (a) Confirmation Bias
     (b) Look-ahead Bias
     (c) Overfitting
     (d) Survivorship Bias
   - Chih-yu suggested the following to cope with the issues:
     (a) Verify the conclusion by double checking with the derived statistical results (Confirmation Bias)
     (b) Leverage the information only on the day of decision making. Prevent parameters tuning only for the financial crises. (Look-ahead Bias)
     (c) Ensure that the result is reasonable in the financial perspective. For example, it is nearly impossible to earn profits without taking risks. (Overfitting)
     (d) Remember to add the delisted stocks back to the trading universe. (Survivorship Bias)

5. To-do lists before the next meeting

   - Find the suitable teammate(s).
   - Read more research papers on Machine Learning application in investment and trading specifically.

6. Meeting adjournment and arrangement of the next meeting

   - The next meeting will be a talk with a new partner, Matthew, at 2pm on June 13, 2020 via Zoom.

6.3 Minutes of the 3^{rd} Project Meeting

1. Arrangement
   - Date: June 13, 2020
   - Time: 2:00pm
   - Place: Zoom
   - Present: Chih-yu LEE, Matthew Johann Siy UY, and Prof. David Paul ROSSITER
   - Absent: None
   - Recorder: Chih-yu LEE
2. Approval of minutes
   - All the discussed contents are well-conveyed to Matthew, and he did not object to any previous minutes.
3. Report on progress
   - Matthew proposed that he can take charge of price prediction sections as he has some previous experience.
   - Besides writing some codes on Quantopian, Chih-yu has also started developing his own backtest system through Python.
4. Discussion items
   - We decided to add Reinforcement Learning into our outline as it would be a brand-new strategy to allocate the weight among different Machine Learning-based strategies which have its own weights.
   - Chih-yu mentioned that he has received some supports from the Finance Department. Thus, he has price data from January 1996 to March 2020 in every trading minutes for some popular Exchange Traded Funds like SPY.
5. To-do lists before the next meeting
   - Finish the GANNT graph and discuss the job delegation.
   - Start drafting the project proposal.
6. Meeting adjournment and arrangement of the next meeting
   - The next meeting will be at 2pm on August 24, 2020 via Zoom.

6.4 Minutes of the 4^{th} Project Meeting

1. Arrangement
   - Date: August 24, 2020
   - Time: 2:00pm
   - Place: Zoom
   - Present: Chih-yu LEE, Matthew Johann Siy UY, and Prof. David Paul ROSSITER
   - Absent: None
   - Recorder: Chih-yu LEE
2. Approval of minutes
   - The minutes of last meeting were approved without amendment and objection.
3. Report on progress
   - We have added the software engineering related goals in our project: building our own custom backtest system and visualization system.
   - We also successfully connected to the IB via its API and two third-party API. We could send the order at the pre-determined price with certain quantity.
4. Discussion items
   - Chih-yu mentioned that he built a simple algorithm on the Permanent Portfolio and obtained a 19.22% return with a 21.83% Max Drawdown and a Sharpe Ratio of 1.26 by 2x leverage.
   - We decided to change our benchmark to this portfolio.
5. To-do lists before the next meeting
   - Finish the draft of the project proposal.
   - Build the visualization system.
6. Meeting adjournment and arrangement of the next meeting
   - The next meeting will be at 11am on September 12, 2020 via Zoom.

6.5 Minutes of the 5^{th} Project Meeting

1. Arrangement

    - Date: September 12, 2020
    - Time: 11:00am
    - Place: Zoom
    - Present: Chih-yu LEE, Matthew Johann Siy UY, and Prof. David Paul ROSSITER
    - Absent: None
    - Recorder: Chih-yu LEE

2. Approval of minutes

    - The minutes of last meeting were approved without amendment and objection.

3. Report on progress

    - We have finalized our project proposal with one more goal: build an app or a website for users' deposit and withdrawal of the money.
    - We finished the draft of the proposal report.

4. Discussion items

    - We should consider using Latex instead of Google doc for the final version of the proposal submission
    - We should not exceed 30 pages and might elaborate more on the financial terms in the project.

5. To-do lists before the next meeting

    - Finish the project proposal.

6. Meeting adjournment and arrangement of the next meeting

    - The next meeting will be at 11am on September 17, 2020 via Zoom.

6.6 Minutes of the 6^{th} Project Meeting

1. Arrangement

    - Date: September 17, 2020
    - Time: 11:00am
    - Place: Zoom
    - Present: Chih-yu LEE, Matthew Johann Siy UY, and Prof. David Paul ROSSITER
    - Absent: None
    - Recorder: Chih-yu LEE

2. Approval of minutes

    - The minutes of last meeting were approved without amendment and objection.

3. Report on progress

    - We finished writing the report in Latex.

4. Discussion items

    - Prof. suggested several points to revise the proposal:

        (a) Explain the terminologies such as Quantopian, Harry Browne Permanent Portfolio and Interactive Brokers to the audience without assuming their background knowledge in these terms.

        (b) Try to add more things in Hardware Requirements.

        (c) Find or recreate the bigger and clearer images for illustration.

        (d) Break long paragraphs into several bullet points.

        (e) Create a glossary instead of having everything in Overview.

        (f) Be aware of format issues like sizing.

5. To-do lists before the next meeting

    - Start building the first model.

6. Meeting adjournment and arrangement of the next meeting

    - The next meeting will be at 11am on September 24, 2020 at Room 3554.

6.7 Minutes of the 7^{th} Project Meeting

1. Arrangement

    - Date: September 24, 2020
    - Time: 11:00am
    - Place: Room 3554
    - Present: Chih-yu LEE, Matthew Johann Siy UY, and Prof. David Paul ROSSITER
    - Absent: None
    - Recorder: Chih-yu LEE

2. Approval of minutes

    - The minutes of last meeting were approved without amendment and objection.

3. Report on progress

    - Chih-yu started revising his Enhanced Harry Browne Permanent Portfolio and successfully obtained an almost 20% annualized return with slightly more than 20% Max Drawdown over 15 years.
    - Matthew attempted to train the NLP models on his local computer.

4. Discussion items

    - Chih-yu suggested Prof. to apply some Interactive Paper Accounts for the real-time trading.
    - We decided to change the benchmark back to S&P 500 as it is a common index that all investors can invest in. Therefore, we need to change our topic title to make it more relevant.

5. To-do lists before the next meeting

    - Finalize the Harry Browne Permanent Portfolio.
    - Start evaluating some 10-K reports by the NLP models.

6. Meeting adjournment and arrangement of the next meeting

    - The next meeting will be at 11:30am on October 6, 2020 at Room 3554.

6.8 Minutes of the 8^{th} Project Meeting

1. Arrangement
    - Date: October 6, 2020
    - Time: 11:30am
    - Place: Room 3554
    - Present: Chih-yu LEE, Matthew Johann Siy UY, and Prof. David Paul ROSSITER
    - Absent: None
    - Recorder: Chih-yu LEE
2. Approval of minutes
    - The minutes of last meeting were approved without amendment and objection.
3. Report on progress
    - Chih-yu started developing the pairs trading strategy on Quantopian.
    - Matthew successfully obtained the sentiments and the change for some corporations in S&P500 from the trained NLP model.
4. Discussion items
    - Prof. replied to us that the Interactive Brokers requested for more information before granting us a free paper trading account.
    - We proposed two possible topics which are:
        (a) Savior of earlier retirement - Using Machine learning and algorithmic trading to improve traditional investment strategies
        (b) Using Machine learning and algorithmic trading to beat the U.S. stock market index
    - We finally decide to change our title into "Using Machine learning and algorithmic trading to beat the U.S. stock market index" as it matches our objectives better.
5. To-do lists before the next meeting
    - Backtest the pairs trading strategy.
    - Evaluate all 10-K reports for the companies within S&P 500.
    - Start developing the Machine Learning-based passive investment strategy.
    - Provide the full information of the project for the paper account application.
6. Meeting adjournment and arrangement of the next meeting
    - The next meeting will be at 3:30pm on October 19, 2020 at Room 3554.

6.9 Minutes of the 9^{th} Project Meeting

1. Arrangement

   - Date: October 19, 2020
   - Time: 3:30pm
   - Place: Zoom (As Prof. took a sick leave on the day)
   - Present: Chih-yu LEE, Matthew Johann Siy UY, and Prof. David Paul ROSSITER
   - Absent: None
   - Recorder: Chih-yu LEE

2. Approval of minutes

   - The minutes of last meeting were approved without amendment and objection.

3. Report on progress

   - Chih-yu finalized the pairs trading strategy on Quantopian, and the strategy has an anuualized return of 10.37%, a max drawdown of 14.30%, and a Sharpe Ratio of 1.25.
   - Matthew successfully obtained the sentiments and the change for all corporations in S&P500 from the trained NLP models.
   - Chih-yu started applying the LSTM models on the components of Harry Browne Permanent Portfolio.

4. Discussion items

   - Prof. mentioned that the Interactive Brokers hasn't replied to his email with the full information of the project yet, and he suggested us to find the alternative channel for the real-time trading.
   - Matthew mentioned that he had a Interactive Broker account, so he might be able to create a paper account for our real-time trading if necessary.
   - Prof. inquired about the implementation of the LSTM model on Harry Browne Permanent Portfolio, and he was surprised about the good fitted result. Chih-yu said he would double check if there was any look-ahead bias during the prediction.
   - Matthew said he cannot find a way to integrate the result on his local computer to the Quantopian as Quantopian did not offer 10-K report nor providing him a way to upload his file.
   - To solve Matthew's problems, we might need to move our works to other platforms or start building our own backtesting and trading platform earlier.

5. To-do lists before the next meeting

    - Assess whether it is necessary for us to find an alternative platform to finish our project.

6. Meeting adjournment and arrangement of the next meeting

    - The next meeting will be at 2pm on November 9, 2020 at Room 3554.

6.10 Minutes of the 10^{th} Project Meeting

1. Arrangement

   - Date: November 9, 2020
   - Time: 2pm
   - Place: Room 3554
   - Present: Chih-yu LEE, Matthew Johann Siy UY, and Prof. David Paul ROSSITER
   - Absent: None
   - Recorder: Chih-yu LEE

2. Approval of minutes

   - The minutes of last meeting were approved without amendment and objection.

3. Report on progress

   - There is no progress during the past two weeks as the Quantopian shutdown with only one-week prior notice. We spent time on downloading the codes we have written on Quantopian.

4. Discussion items

   - We had to postpone our progress for one to two months as it took time for finding an alternative platform, learning the platform-specific languages, changing the syntax on Quantopian, and moving our works to the new platform.
   - We might abandon the plan to create a website.
   - We might need to reduce the duration for the real-time trading as we need more time for developing the strategies and combining them together.

5. To-do lists before the next meeting

   - Find the suitable alternative platforms within two weeks and try to relocate the works on it.

6. Meeting adjournment and arrangement of the next meeting

   - The next meeting will be at 2pm on November 23, 2020 at Room 3554.

6.11 Minutes of the 11^{th} Project Meeting

1. Arrangement
   - Date: November 23, 2020
   - Time: 2pm
   - Place: Room 3554
   - Present: Chih-yu LEE, Matthew Johann Siy UY, and Prof. David Paul ROSSITER
   - Absent: None
   - Recorder: Chih-yu LEE
2. Approval of minutes
   - The minutes of last meeting were approved without amendment and objection.
3. Report on progress
   - After testing out different available platforms, we decided to adopt QuantConnect as our alternative as it also can solve Matthew's problems in stock selection using NLP.
   - A good thing to mention is that QuantConnect is also written in Python and it supports all Machine Learning packages that we might need.
4. Discussion items
   - We needed to update the GANNT Chart.
   - We planned to replicate all our works during the winter break.
5. To-do lists before the next meeting
   - Replicate all works that we have done in Quantopian by the starting of 2021 Spring semester.
6. Meeting adjournment and arrangement of the next meeting
   - The next meeting will be at 3pm on February 2, 2021 on Zoom.

6.12 Minutes of the 12^{th} Project Meeting

1. Arrangement

   - Date: February 2, 2021
   - Time: 3pm
   - Place: Zoom (Chih-yu was not in Hong Kong)
   - Present: Chih-yu LEE, Matthew Johann Siy UY, and Prof. David Paul ROSSITER
   - Absent: None
   - Recorder: Chih-yu LEE

2. Approval of minutes

   - The minutes of last meeting were approved without amendment and objection.

3. Report on progress

   - All the works except pairs trading has been successfully replicated.
   - Matthew's stock selection using NLP models can generate 12.30% annualized return.

4. Discussion items

   - As pairs trading strategy involves the pipline structure in Quantopian, we need to figure out how to rewrite it on QuantConnect.
   - We will be working on writing progress report as there are only 10 days left.

5. To-do lists before the next meeting

   - Finish the progress report by the deadline on February 12.
   - Try to finish pairs trading strategy on QuantConnect as much as possible.

6. Meeting adjournment and arrangement of the next meeting

   - The next meeting will be in March 2021 at Room 3554 after Chih-yu return to Hong Kong and finish his quarantine.

6.13 Minutes of the 13^{th} Project Meeting

1. Arrangement

    - Date: March 3, 2021
    - Time: 11:30am
    - Place: Room 3554
    - Present: Chih-yu LEE, Matthew Johann Siy UY, and Prof. David Paul ROSSITER
    - Absent: None
    - Recorder: Chih-yu LEE

2. Approval of minutes

    - The minutes of last meeting were approved without amendment and objection.

3. Report on progress

    - The automatic trading system on QuantConnect works properly.
    - Due to the time constraint, we might not have time to construct the website.
    - The pairs trading strategy on QuantConnect did not do well as QuantConnect did not support trading delisted stocks.

4. Discussion items

    - We decided to remove the website construction from the to-do list as we only have 1 month left for the final report.

5. To-do lists before the next meeting

    - Start working on the Reinforcement Learning by creating some testing environment and agents.
    - Finish the Machine Learning-based Harry Browne Permanent Portfolio.

6. Meeting adjournment and arrangement of the next meeting

    - The next meeting will be in April 2021 at Room 3554 as there are several midterms for Mathhew and Chih-yu in March.

6.14 Minutes of the 14^{th} Project Meeting

1. Arrangement
    - Date: April 13, 2021
    - Time: 10am
    - Place: Zoom
    - Present: Chih-yu LEE, Matthew Johann Siy UY, and Prof. David Paul ROSSITER
    - Absent: None
    - Recorder: Chih-yu LEE
2. Approval of minutes
    - The minutes of last meeting were approved without amendment and objection.
3. Report on progress
    - We will start doing automatic trading on April 14^{th}.
    - Due to the time constraint, we delete the Reinforcement Learning in our project.
4. Discussion items
    - As the computational power is limited, we could not successfully build a suitable environment for the agent to learn within the 30-minute limitation on QuantConnect. Therefore, we decided to abandon the Reinforcement Learning in our project.
    - We have finished our draft and discussed with the Prof. during the meeting.
    - Prof. suggested us to add the hyperlinks for QuantConnect and Quantopian.
    - Prof. thought the abstract should be revised, and we will send it to him by the deadline of the final report.
5. To-do lists before the next meeting
    - Finish the final report by April 14^{th}.
    - Prepare for the video trailer as well as the oral presentation.
6. Meeting adjournment and arrangement of the next meeting
    - The next meeting will be in late April or early May at Room 3554 to discuss the oral presentation.

7 Appendix B: Glossary

As the project involves a lot of financial concepts, the following are descriptions of the terminologies used in the alphabetic order:

1. Alpha: It is a description of the performance of active investments. If the alpha is positive, it implied that the strategy is beating the market (or certain benchmark) As, in finance, the return gained from the market growth is referred to be related to beta, alpha quantifies the excess return obtained by the investors' management.

2. Beta: It is used for measuring the market volatility. As the expected return has a linear relationship with the market risk, the coefficient is named Beta. It is frequently used for estimating the expected return and risk of the security.

3. Black-Swan Event: Black-Swan Event is the situation where the condition is unpredictable and beyond what is normally anticipated. Most of the time, it indicates some events that have a severe consequences in the financial market, such as crash of housing market in 2008 Global Finanical Crisis and 2020 coronavirus pandemic.

4. Bond: It is kind of a loan that is typically issued by the corporation or government. The bond holder will receive pre-specified interest from the issuer periodically and receive the borrowed principal (face value) on the maturity date.

5. Brokerage: It is a firm or an individual that act as the intermediary between investors and exchanges. It will help its clients to buy or sell securities on their behalf.

6. Buy-side: It is a segment of the financial institutions. They make direct investment in the financial markets and earn money from the investment gains. The notable examples will be insurance firms, mutual funds, and hedge funds.

7. Correlation

   It is a statistic to measure the extent at which two securities are moving in the same direction. As the formula below showed, it is capped at -1 to 1 by Cauchy–Schwarz Inequality: (r_1 denotes the return of Asset 1 and r_2 denotes the return of Asset 2)

   $$\text{corr} = \frac{cov(r_1,r_2)}{\sqrt{var(r_1)}\cdot\sqrt{var(r_2)}}$$

8. Derivatives

   It is a contract (or agreement) between two or more parties. Its value is determined by the agreed-upon financial assets (or securities). It is categorized into several groups and we would describe futures and options in the following context:

   - Futures: It is a standardized legal agreement between two parties to trade assets at the certain price in the future. It reduces the risk of price fluctuation of the assets.
   - Options: It grants the holder the right bet not obligation to trade securities at the pre-determined price at or by the expiration date. In short, it is a right for buying/selling assets with limited loss for the holder. The money paid to the writer, the counterparty, for holding this right is called option premium.

9. Equity: It is another word for saying stock.
10. Factor Investing: It is an investment methodology that focuses on those factors that could explain the variability and return of the stocks price. For example, market is a well-known factor for moving the price. Other examples will be size, momentum and value.
11. Foreign Exchange (FOREX): It is an alternative word for describing the market of the foreign currencies.
12. Hedge Fund: It is a fund that would adopt more complicated trading strategies that would hedge their risk by using derivatives. It is usually only available for the institutional investors.
13. Investment Bank: It is a sector of the financial industry where it provides the service to other participants like high net worth individuals, corporations, and government. Usually, it will help corporations to finance their projects for the economic development.
14. Latency: It represents a delay of trades execution. For example, if we would like to buy Asset A at the price of 15, it could turn out that our trade is not filled since it has been bought with other investors having faster execution. In the project, we would like to reduce its impact as much as possible.
15. Long: It is a financial term for buying or holding the positive amount (or called position) of assets.
16. Market Risk: It is the chance of an investor experiencing losses due to factors that affect the overall performance of the financial markets in which he or she is involved.

17. Max Drawdown

    Denote N as number of trading days and V_t as the portflio value at day t where $t \in \{0, 1, \ldots, N\}$. Drawdown (DD) is the measure of the decline from a historical peak to the current level, and the Max Drawdown (MDD) on day t is the maximum observed loss from a peak to a trough of a portfolio from day 0 to day t. They are defined by:

$$DD_t = \frac{V_t}{max\{V_0, \ldots, V_t\}} - 1, \text{ where } t \in \{0, \ldots, N\}$$

18. Mutual Fund: It is a fund that investors put their money in exchange of some proportion of the collections of stocks, bonds, and other financial securities. There will be one or more fund manager to properly invest these money into the financial securities.

19. Operational Risk: It could be referred to electronic trading risk in the context. As we are running the algorithm for trading, it is likely to have some bugs in the algo so that we could lose money accordingly. Every possible risk for not getting the desired results could be considered as the operational risk.

20. Order: It is an inclination of buying or selling an asset at the certain price. If the order is filled, the investor is required to buy or sell the asset. If it could not be filled immediately, it would be placed on the order book and wait until other people have the inclination of doing trades in the opposite direction.

21. Pension Fund: It is a scheme supported by the government to pay for citizens' retirement plan.

22. Quant Fund: It is a fund that applies quantitative methods in their investment or trading. It would involve statistical models and algorithms most of the times.

23. Return (or Compounded Annual Growth Rate, CAGR)

    Denote N as number of trading days and V_t as the portfolio value at day t where $t \in \{0, 1, \ldots, N\}$. It is an annualized rate of return for an investment to grow in T year(s). We assume that there are 252 trading days per year, so we can simply divide the number of trading days by 252 to get T.

$$\text{T} = \tfrac{N}{252}$$
$$\text{CAGR} = \left(\tfrac{V_N}{V_0}\right)^{\frac{1}{T}} - 1$$

24. Risk-to-Return Ratio: It is a ratio of the risk to its return (or called profit). As it is known that we could earn more profits by taking more risks, it would be a key ratio to reflect whether we have taken some unnecessary risk.

25. Sell-side: It is another segment of the financial institutions. Unlike the buy-side, sell-side professionals will not make direct investments. Instead, they provide information and recommendation to their clients for investments.

26. Sharpe Ratio

    Its formula is shown as follows:

    $$\text{Sharpe Ratio} = \tfrac{CAGR - R_f}{\sigma},$$

    where R_f is the annual risk-free rate and σ is the annualized volatility.

27. Short: It reflects that the investors have a negative position in the assets or that the investors are going to sell the assets.

28. Slippage: It refers to the gap between the targeted price of a trade and the price that is being executed. We would like to reduce slippage as much mas possible in order to get the expected good results.

29. Trade: It is an economical event involving buying and selling the goods, especially the financial assets in our context.

30. Volatility

    Denote N as number of trading days and V_t as the portflio value at day t where $t \in \{0, 1, \ldots, N\}$. Volatility is a statistical measure of the dispersion of returns, and it is measured as the standard deviation between returns during a period. Usually, we calculated the volatility based on the daily return; thus, we have to multiply the result by square root of 252 to get the Annualized Volatility. The formula is defined as follows:

    $$\sigma = \frac{\sum_{t=1}^{N}(R_t - \bar{R})}{N-1} \cdot \sqrt{252},$$

    where R_t is the daily realized return on day t and $\bar{R}$ is the average of all realized returns in N days.

31. Volume: It is total number of shares being traded during the period.

8 Appendix C: Project Planning

8.1 Distribution of Work

Note: In the late October, Quantopian shut down suddenly. Therefore, we postponed our progress for one to two months in order to find the alternative, learn how to code on the new platform, and move and transform our codes from Quantopian. Fortunately, we discovered QuantConnect, which is also a open-source trading platform that is written in Python programming language. Therefore, the GANNT Chart shown in Table 9 is the updated version.

Table 8: Distribution of Work

Task	**Chih-yu**	**Matthew**
Conduct the literature survey	●	○
Formulate ideas and write the proposal reports	●	○
Type up the Monthly Reports	○	●
Do the Progress Report	○	●
Compile the Final Report	○	●
Design the Presentation Poster	○	●
Prepare for the presentation	○	●
Produce the project video	○	●
Collect data and create preliminary settings	●	○
Build a Deep Learning model for passive investment	●	○
Conduct research on Pairs Trading	●	○
Perform sentiment analysis on financial reports with NLP models	○	●
Undertake research on price prediction by LSTM	○	●
Evaluate performance among different RL models	●	○
Construct a custom backtest system	●	○
Create a data visualization system	●	○
Connect to IB Paper Account	●	○
Develop a website or mobile application	○	●
Maintain the connection to the broker	●	○
Optimize the results	○	●

8.2 GANTT Chart

Table 9: GANTT Chart

Task	Jul	Aug	Sep	Oct	Nov	Dec	Jan	Feb	Mar	Apr	May
Conduct the literature survey	■	■	■								
Formulate ideas and write reports	■	■	■								
Create preliminary settings	■	■	■								
Build the model for passive investment strategy on Quantopian		■	■	■							
Build the model for passive investment strategy on QuantConnect					■	■	■	■			
Produce monthly and progress reports				■	■	■	■	■	■		
Do pairs trading by clustering on Quantopian			■	■							
Do pairs trading by clustering on QuantConnect							■	■			
Carry out sentiment analysis on 10-K report by NLP		■	■	■	■	■	■				
Predict stock prices by neural networks					■	■	■	■	■		
Develop the RL model							■	■	■	■	
Set up a custom backtesting system and establish connection with IB for auto-trading	■	■	■	■	■	■	■	■	■	■	
Develop the app or website									■	■	
Execute trades in the real market								■	■	■	■
Write the final report								■	■	■	
Produce the video presentation									■	■	■

9 Appendix D: Required Hardware and Software

9.1 Hardware Requirements

Table 10: Hardware Requirements

Item	Specification
Development PC	MacBook Pro (Retina, 13-inch, 2019, Two Thunderbolt 3 ports)
Development PC	MacBook Pro (Retina, 15-inch, 2018, Two Thunderbolt 3 ports)
AWS Server	
GPU	NVIDIA TITAN V GV100
Server RAM	32GB

9.2 Software Requirements

Table 11: Software Requirements

Item	Version	Specification
Development OS	MacOS Catalina 10.15.6	Environment for development
Git	2.23.0 or after	Version control
Miniconda	4.8.4 or after	Package control
IB Gateway	980.4j	Stable and direct Access to the IB trading system
Python	3.6.10 or after	Programming language
Pytorch	1.6.0 or after	Machine Learning library
Pandas	1.1.2 or after	Data Processing library
Numpy	1.19.2 or after	Data Processing library
ibapi	9.76	Python library for connecting to IB Brokerage Account
dash	1.16.1	Python library for visualization output generation
plotly	1.16.1	Python library for creating interactive plots
React	17.0	A JavaScript library for building UI components
Flask	2.0	A micro web framework for web application development
SQL	5.5	Database Processing Programming Language

10 Appendix E: Additional Information

10.1 Brokerages

A brokerage firm plays the role of an intermediary between investors and securities exchanges. Brokerages make money by charging commissions for each transaction that occurs on their platforms.

To help us decide which brokerage firm to select, we identified several key criteria:

1. Universe of trading assets: The number of markets and asset classes that can be to traded on the platform.
2. Commission: The fee charged for each transaction conducted with the brokerage firm.
3. Access to historical data
4. A simple, well-documented API
5. Continuity and dependability: We don't want a broker that's going to go out of business or discontinue its services.

After comparing several different online brokerages, we chose one, Interactive Brokers (IB).

Not only does Interactive Brokers have one of the lowest commission fees, they provide the largest universe of trading assets, giving traders access to different markets across five continents, as well as one of the most comprehensive, fast, and convenient API systems for coding.

10.2 Quantopian

Quantopian was a U.S.-based company that aimed to provide services to investors around the globe to conduct quantitative analyses on U.S. securities. It operated as a open-source trading platform. It had several features that would have been useful for our project:

1. Written in the Python programming language, which is very popular for machine learning
2. Tick-sized price data of all U.S. stocks since 2002
3. Financial health information about the corporations
4. Macroeconomic data
5. A well-designed notebook for research
6. Precise and accurate backtesting results

Unfortunately, Quantopian shut down in November 2020; therefore, we moved all our works from Quantopian to QuantConnect. This cost us two months to migrate all our works to the alternative platform, QuantConnect.

10.3 QuantConnect

QuantConnect is a U.S.-based company that aims to provide services to investors around the globe to conduct quantitative analyses of equities, forex, cryptocurrency, contract for difference (CFD), options, and futures. It not only serves as a platform provider but also sells the licenses of some profitable strategies to the investors. Also, it operates as an open-source trading platform. It has several features that were useful for our project:

1. Written in the Python programming language, which is very popular for machine learning

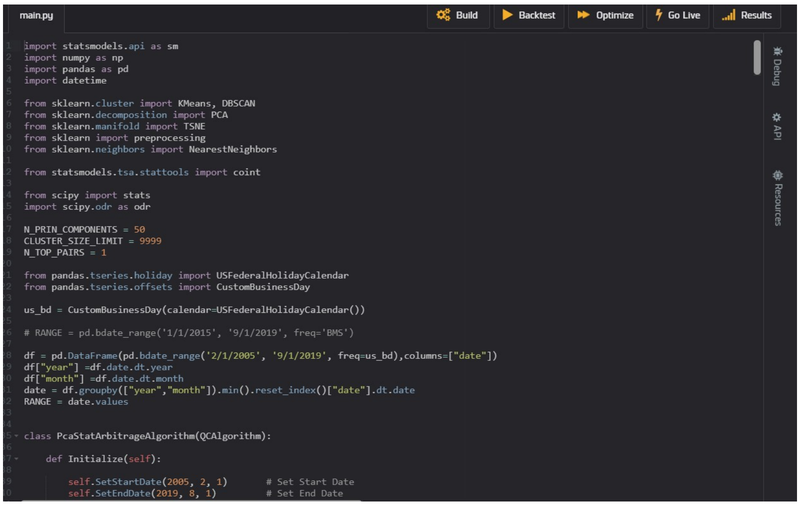


Figure 64: Python Programming Interface on QuantConnect

2. Tick-sized price data of all U.S. stocks since 1998

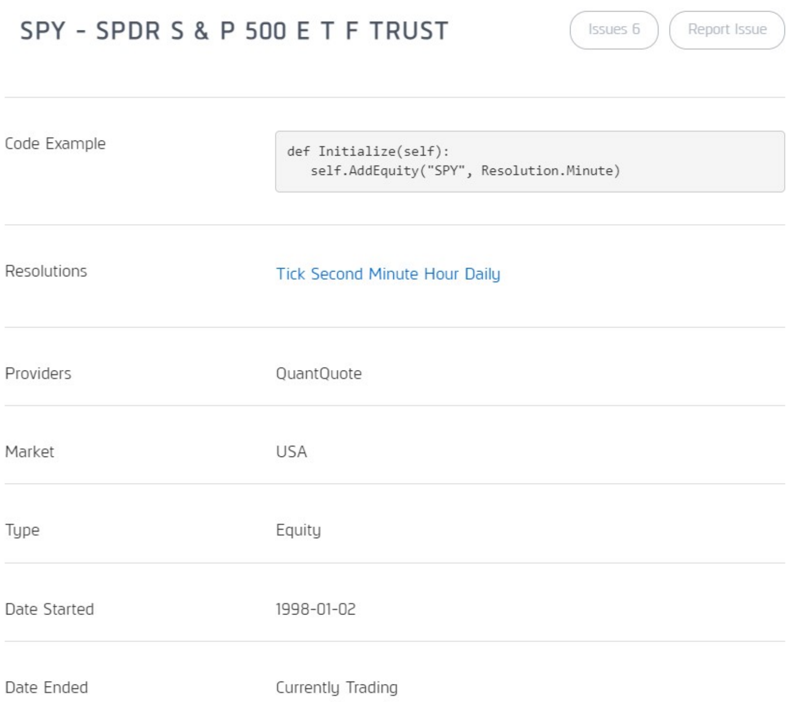


Figure 65: Information of SPY

3. Corporate fundamental data from MorningStar

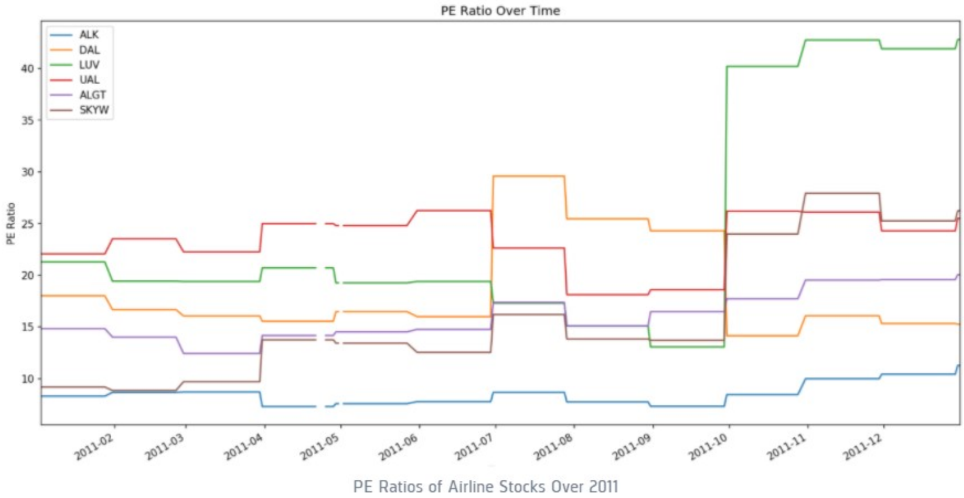


Figure 66: Image of The Price-to-Equity Ratio Over Year 2011 for Some Stocks

4. Access to financial reports such as 10-K reports

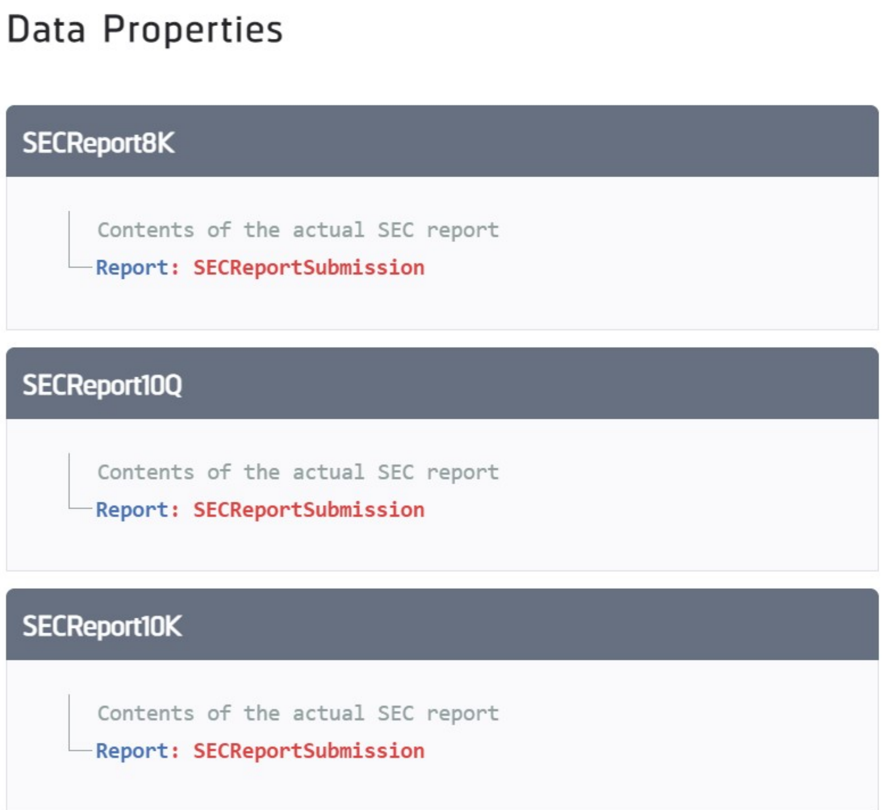


Figure 67: Information about Extracting Financial Reports on QuantConnect

5. A well-designed notebook for research

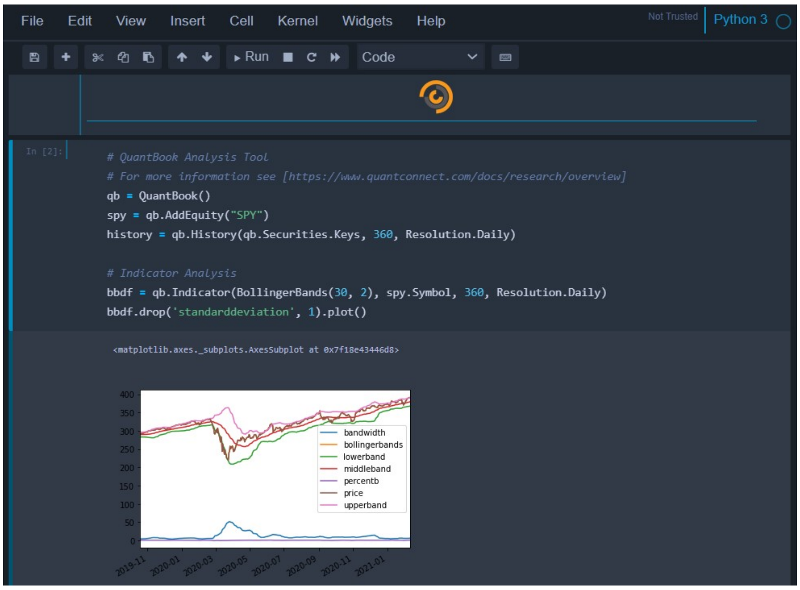


Figure 68: Illustration of The Research Notebook

6. Comprehensive backtest reports

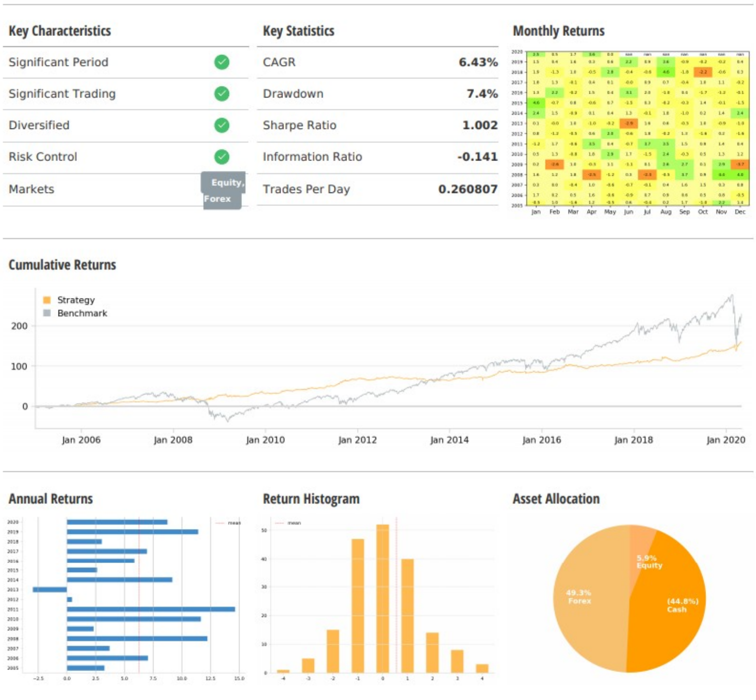


Figure 69: Illustration of The Backtest Report

7. Real-time trading execution

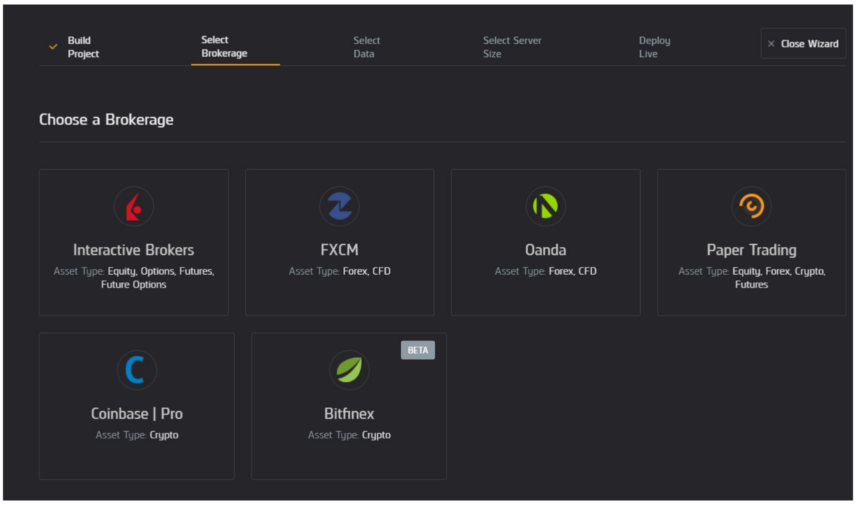


Figure 70: List of Brokers Available on QuantConnect Real-Time Trading

8. A comprehensive list of Machine Learning packages

Supported Libraries

QuantConnect has 11 supported machine learning libraries installed and available. You can import these packages and use them as demonstrated below.

Name	Version	Language	Import Statement	Example
Tensor Flow	1.13.1	Python	import tensorflow	↗
SciKit Learn	0.21.3	Python	import sklearn	↗
Py Torch	1.1.0	Python	import torch	↗
Keras	2.2.4	Python	import keras	↗
Theano	1.0.4	Python	import theano	
hmmlearn	0.2.2	Python	import hmmlearn	
tsfresh	0.12.0	Python	import tsfresh	
fastai	1.0.54	Python	import fastai	
Deap	1.0.54	Python	import deap	
mlfinlab	0.9.3	Python	import mlfinlab	
Accord	3.6.0	CSharp	using Accord.MachineLearning;	↗
AForge.Neuro	2.2.5	CSharp	using AForge.Neuro;	

Figure 71: The Full List of Supported Libraries on QuantConnect

9. Free Alternative data such as News and Economic Data

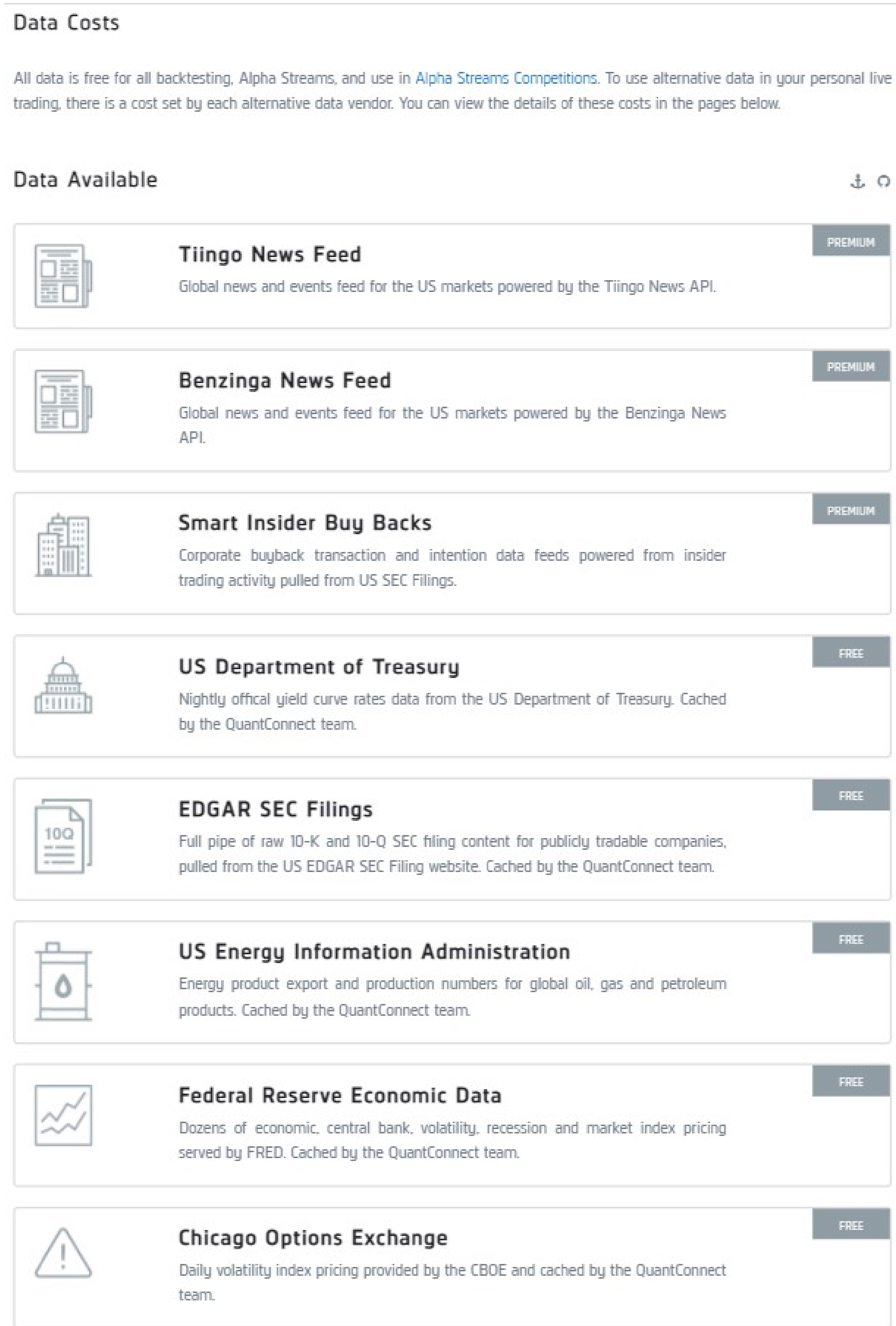


Figure 72: List of Alternative Data Available

11 Appendix F: Additional Information For Future Work

11.1 Reinforcement Learning

11.1.1 Introduction to Reinforcement Learning

Reinforcement learning (RL) is an area of Machine Learning concerned with how learning agents take actions in an environment in order to maximize the reward they receive. The reward function is used to allow the agent to learn by itself the best way to do a specific task.

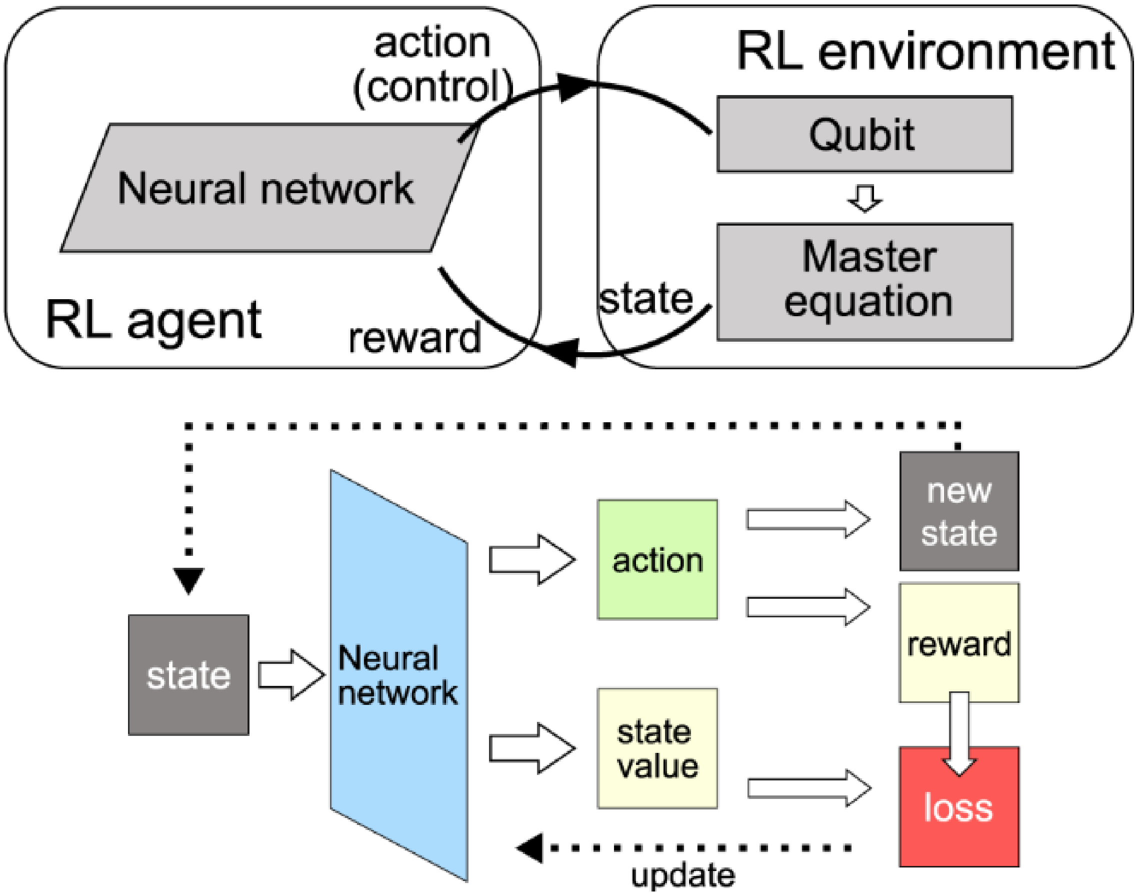


Figure 73: Illustration of RL [2]

Advantage

One of the greatest challenges in portfolio management is to decide how to allocate money towards different trading strategies. This is where Reinforcement Learning comes in. It is possible to use Reinforcement Learning to learn what is the best way to allocate weight among different trading strategies depending on the target that we want to maximize. Some possible examples is Sharpe Ratio and Cumulative Return.

11.1.2 Literacture Survey

According to Finance Talks [30], reinforcement learning has been applied to several topics in finance: portfolio optimization, trades execution optimization, and market making. As we mainly focus on the first area, we noticed that there are some publications online.

Angelos Filos [31] suggested that a family of model-free agents have outperformed in both European stock market (Euro STOXX 50) and U.S. stock market (S&P 500). As we are focusing on strategy-wise diversification, we expect that the structure and underlying relationship among strategies will differ from his research in individual stocks. Moreover, we believe we could achieve a better result, as our non-Machine Learning threshold has already generated more than a 9.2% annualized return.

11.1.3 Weight Allocation of Different Strategies by Reinforcement Learning (RL)

As diversification has been known as the only free lunch in Finance, we believe it will be a crucial part for us to mitigate our risks by allocating the capital properly to the good strategies we have developed. By reinforcement learning, we would set the original weight as equal-weighted or the best weight allocation given by the portfolio theories and let the agent explore any more profitable opportunities.

Since Reinforcement Learning models could construct the portfolio dynamically, it would be a better approach to deal with the rapid-changing financial market than the static, traditional portfolio theory. Moreover, Reinforcement Learning is capable of dealing with sequential decision making, where investment and trading require more time to examine its profitability. In our project, we change different value functions as well as several goals to be maximized.

For the details of the reinforcement learning model, we consider the following settings under the Markov Decision Process (MDP):

- States: The state of the strategy i at time t ($s_{i,t}$) is defined as the sign of the return of the strategy i from $t - \Delta t$ to t ($r_{i,t}$). That is:

$$s_{i,t} = \begin{cases} 1 & \text{if } r_{i,t} > 0 \\ -1 & \text{if } r_{i,t} < 0 \end{cases} \tag{1}$$

- Action: As the agent will do the action, i.e., allocate the weight to the strategy i at time t ($a + i, t$), we could simply discretize the real number between 0 and 1 into 100 segments. That is,

$$a_{i,t} = \{0.00, 0.01, \ldots, 0.99, 1.00\}$$

- Reward: The reward of the strategy i at time t ($R_{i,t}$) could be defined as the return of the strategy i from t to $t + \Delta t$, i.e., $R_{i,t} = a_{i,t}r_{i,t} + (1 - a_{i,t}) \sum_{j \neq i} r_{j,t}$

As we have four strategies, we could simply allocate the equal weight of $\frac{1-a_{i,t}}{3}$ to the remaining three strategies, and we could repeat it for i = 1, 2, 3, and 4. Finally, we could obtain the best result among all i.

11.2 Stock Price Prediction in Active Investment by LSTM

11.2.1 Introduction to Active Investment

In active investment, it is basically about buying "winners" and selling "losers". To do this, we have identified 3 main methods of Active Investment Strategies: Securities Selection via NLP sentiment analysis on 10-K Securities, Securities Selection via LSTM and RNNs Models, and Pairs Trading Strategy via Clustering.

Regarding the implementation of our strategies, we will first develop them on the research notebooks at QuantConnect since they have access to many libraries and data sources, which would greatly ease the research process. After formulating the strategies, we will then backtest them on QuantConnect to see their performance.

11.2.2 Implementation of Stock Price Prediction

1. Split the existing stock price data for the stock X in the universe as 80/20 training and testing ratio.
2. Predict the stock price on the training data by the LSTM model.
3. Validate the model result on the testing data.
4. Apply the trained model for the prediction of the future one-month stock movement.
5. Compute the t-stat of each stock price movement, where the t-stat is defined to be $\frac{\bar{r_X}-0}{\sigma_X}$, where $\bar{r_X}$ is the mean of the predicted daily returns and σ_X is the standard deviation of the predicted daily returns.
6. Track the stocks with t-stat greater than 1.96 or smaller than -1.96, which implies a 95% confidence level that they will be significantly different from 0 return.
7. Repeat Step 1 to 6 for all stocks X in the trading universe.
8. Buy those stocks with t-stat greater than 1.96 and sell those with t-stat smaller than -1.96.
9. Repeat Step 1 to 8 on the first business day of every month.

11.3 Construction of Website

11.3.1 Goal

For the front end, the goal is to create a simple and easy-to-use website that will allow an investor to do the following:

1. Register an account
2. Log-in and Log-out
3. Input Risk Profile
4. Withdraw and Deposit Money
5. Manage Portfolio Weights
6. Analyze Portfolio Growth
7. Present Charts

11.3.2 Add Website Stage After Stage 4 Live Trading Stage

This stage mainly focused on integrating our trading system into a website in order to create a simple and convenient investing experience for clients. In order to accomplish this, we have set out two goals:

1. Develop the website and user interface.
2. Integrate the trading system into the website.

11.3.3 Implementation of Website

For creating the website, the process can be broken down into 3 parts: Front End, Back End, and Database Management.

1. **Front End**

   To create the Front End, the current plan is to use React.js (or simply React), which is an open-source Javascript library for building user interfaces. We chose React because it is very simple to use, scalable, and fast. For example, React allows developers to create large dynamic web applications capable of changing data, without having to reload the page.

2. **Back End**

   To create the Back End, the current plan is to use Flask. The reason it was chosen is because it has access to a wide range of libraries that are very useful and convenient. Aside from this, Flask is one of the most widely used web frameworks, which makes it easier to learn from scratch due to the vast amount of online tutorials.

3. **Database**

   To create the database, the current plan is to adopt MySQL. MySQL is a Relational Database Management System (RDBMS) that uses Structured Query Language (SQL). Aside from being free, MySQL is very convenient to use, as well as quick and efficient in retrieving large amounts of data.

11.4 Custom Backtesting System

11.4.1 Implementation of a Backtesting System

In the project, we developed our strategy on QuantConnect, and the system is fully implemented and maintained by QuantConnect. On the platform, we can utilize the data available and do the backtest within the specified start date and end date. We could also set our initial capital and what we would like to do repeatedly (e.g. daily or monthly).

```python
class HarryBrowne(QCAlgorithm):

    def Initialize(self):
        self.SetStartDate(2020, 1, 1)  # Set Start Date
        self.SetEndDate(2021, 1, 29)  # Set Start Date
        self.SetCash(1000000)  # Set Strategy Cash
        #self.SetRiskManagement(TrailingStopRiskManagementModel(0.15))

        self.lookback = 21 # used for vol targeting
        self.stocks = ['SPY','TLT', 'GLD', 'SHY']
        self.vol_per_symbol = VOL_TGT / len(self.stocks)

        self.AddEquity("SPY", Resolution.Daily)
        self.AddEquity("TLT", Resolution.Daily)
        self.AddEquity("GLD", Resolution.Daily)
        self.AddEquity("SHY", Resolution.Daily)
        self.NAV = {}

        #self.Securities["SPY"].FeeModel = ConstantFeeModel(1)
        #self.Securities["TLT"].FeeModel = ConstantFeeModel(1)
        #self.Securities["GLD"].FeeModel = ConstantFeeModel(1)
        #self.Securities["SHY"].FeeModel = ConstantFeeModel(1)

        #self.Schedule.On(self.DateRules.MonthStart(), self.TimeRules.AfterMarketOpen("SPY", 30), self.rebalance)
        self.Schedule.On(self.DateRules.EveryDay(), self.TimeRules.AfterMarketOpen("SPY", 30), self.rebalance)
```

Figure 74: Illustration of QuantConnect Settings

For our custom one, we could run the code on the data that we obtained from the Thomson Reuters or Bloomberg. Then, we could generate the two csv files from the algorithm with the quantity and weight for each security in the universe. With the two csv files, we could feed them into the built visualization.

Time	SPY	TLT	GLD	SHY	UVXY
12/1/2004 14:30	62227	106655	74046	0	0
12/2/2004 14:30	62227	106655	74046	0	0
12/3/2004 14:30	62227	106655	74046	0	0
12/6/2004 14:30	62227	106655	74046	0	0
12/7/2004 14:30	62227	106655	74046	0	0
12/8/2004 14:30	62227	106655	74046	0	0
12/9/2004 14:30	62227	106655	74046	0	0
12/10/2004 14:30	62227	106655	74046	0	0
12/13/2004 14:30	62227	106655	74046	0	0
12/14/2004 14:30	62227	106655	74046	0	0
12/15/2004 14:30	62227	106655	74046	0	0
12/16/2004 14:30	62227	106655	74046	0	0
12/17/2004 14:30	62227	106655	74046	0	0
12/20/2004 14:30	62227	106655	74046	0	0
12/21/2004 14:30	62227	106655	74046	0	0
12/22/2004 14:30	62227	106655	74046	0	0
12/23/2004 14:30	62227	106655	74046	0	0

Figure 75: Quantity Table from The Custom Backtest System

Time	SPY	TLT	GLD	SHY	UVXY
12/1/2004 14:30	0.741939	0.92204	0.336021	0	0
12/2/2004 14:30	0.747258	0.924117	0.334945	0	0
12/3/2004 14:30	0.733032	0.921249	0.333543	0	0
12/6/2004 14:30	0.732955	0.924622	0.330474	0	0
12/7/2004 14:30	0.73059	0.931468	0.332062	0	0
12/8/2004 14:30	0.727081	0.935982	0.320536	0	0
12/9/2004 14:30	0.733535	0.93474	0.320558	0	0
12/10/2004 14:30	0.733855	0.936528	0.317887	0	0
12/13/2004 14:30	0.731591	0.92755	0.317568	0	0
12/14/2004 14:30	0.731513	0.92786	0.313836	0	0
12/15/2004 14:30	0.72329	0.925053	0.313209	0	0
12/16/2004 14:30	0.734823	0.926483	0.316144	0	0
12/17/2004 14:30	0.730797	0.930717	0.321731	0	0
12/20/2004 14:30	0.728828	0.929649	0.321873	0	0
12/21/2004 14:30	0.728926	0.925471	0.318592	0	0
12/22/2004 14:30	0.732732	0.924946	0.318041	0	0
12/23/2004 14:30	0.733767	0.922542	0.32006	0	0

Figure 76: Weight Table from The Custom Backtest System