Multi-Agent Basket Selection with Option Forecasting and Sentiment Analysis

Adar Schwarzbach, Ben Goldfried, Nikhil Sinha, Yuqi Shan

December 13, 2024

Contents

1 Abstract

Constructing a diversified and optimized investment portfolio remains a significant challenge in quantitative finance, necessitating the integration of multifaceted data sources and advanced analytical techniques. This paper presents a novel multi-agent system that leverages the rise in artificial intelligence to create a diverse basket of stocks that are primed to perform well in the future. Our trading strategy involves combining option forecasting scores along with sentiment analysis scores to assess whether to initiate a stock position and determine the appropriate investment amount. We also implement a robust, three-pronged, risk management strategy that utilizes stop-loss orders, VaR, and volatility-based weighting to successfully mitigate our Drawdown and losses.

2 Background

The rise in prevalence of Artificial Intelligence has touched many different industries, and quantitative finance is no exception. Within quantitative finance, there are many different ways that AI could impact a trading strategy, from security selection to generating trading signals. Our group believed that there were many options to utilize the power of AI, and explored a few of them in this project.

Our motivations for the areas of research in our project come from a few challenges we encountered during the course of the class. As we learned, increasing diversity in a portfolio significantly decreases the risk associated with the portfolio, so we wanted to explore ways in which we could build a portfolio intelligently to diversify the individual component assets while also maintaining good return.

As a group, we were also interested in the different ways Artificial Intelligence could be used in finance. Our group had multiple members with a background in Computer Science, so we wanted to investigate multiple different avenues to generate potential value for a trading strategy, and tried to conglomerate them into a single cohesive strategy.

2.1 Strategy Overview

The strategy first generates a basket of stocks utilizing a Multi-Agent framework described in Section 3. For each asset in the basket, the strategy generates signals based on the current market sentiment (described in Section 4) and the relative pricing of options for the asset (described in Section 5). We generate trading signals weekly, and rebalance our portfolio in a Long/Short fashion. There is a stop loss that runs on an hourly cadence, as well as Value at Risk thresholding and volatility scaling to manage risk, further described in Section 7.

Multi-Agent Background The basket of stocks is generated via a Multi-Agent system, utilizing adversarial Large Language Models in order to iterate towards a final selection. The process utilizes market data, sophisticated prompt engineering techniques and several LLM's in order to make a final selection. Notably, the system can consider previously rejected baskets of securities in order to learn what constitutes a unsatisfactory basket as it iterates towards ultimately selecting the basket.

Sentiment Background Sentiment analysis extracts the general sentiment of language. In the context of our project, we extract the sentiment about financial articles related to stocks to predict their price movements 7.

2.2 Options Forecasting

Options prices reflect information about trader's beliefs in the future movements of the stocks, as well as hedging against their current beliefs on the market. Using this pricing data, we believe it is possible to find signal to execute profitable trades. In Lin (2013) 9, we found a paper that backed up this idea, showing strong correlation between options pricing and stock movement. We explored this idea further to generate valuable trading signals.

3 Multi-Agent Systems for Security Selection

3.1 Introduction

In the realm of quantitative finance and algorithmic trading, constructing a diversified and optimized investment portfolio is a complex task that requires careful analysis and strategic decision-making. Traditional methods can rely

on singular models or static criteria, which may not adequately capture the dynamic and multifaceted nature of financial markets. To address these challenges, we propose a novel approach utilizing a multi-agent system for security selection, inspired by the frameworks presented by Ganesh et al. in JP Morgan's study on multi-agent simulations in dealer markets 1.

Our approach leverages the capabilities of advanced artificial intelligence (AI) models, specifically large language models (LLMs), and incorporates advanced prompt engineering techniques as systematically surveyed by Sahoo et al. 3. By deploying specialized agents with distinct roles and responsibilities, the system iteratively refines a basket of stocks that meet predefined criteria such as intrigue, diversification, and low correlation. This method enhances the robustness and adaptability of the selection process, ensuring that the final portfolio aligns with investment objectives. By utilizing this approach for security selection, we are able to use the most advanced models availible to the general public. This is not something our group would be able to achieve in QuantConnect's live environemnt.

A critical aspect of our methodology is the elimination of look-ahead bias in backtesting. Look-ahead bias occurs when future information is inadvertently used in historical simulations, leading to overestimation of a strategy's performance. By strictly instructing the agents to consider only the data provided, we aim to produce realistic and reliable backtesting results. Furthermore, the system learns from previous unsuccessful portfolio selections (failed baskets), iteratively refining its decision-making process to enhance future outcomes. Both of these prompt engineering techniques are supported by the findings in Sahoo et al. 3.

3.2 System Architecture and Agent Roles

To explore an interactive architecture diagram, visit our **Figma diagram**.

The proposed multi-agent system comprises six specialized agents, each built with specific foundation models and functions to perform designated tasks within the security selection pipeline:

1) **Coordinator Agent (GPT-o1)**

2) **Stock Generation Agent (GPT-4)**

3) **Correlation Analysis Agent (Claude Sonnet 3.5)**

4) **Filtering Agent (GPT-4)**

5) **Evaluation Agent (Claude Sonnet 3.5)**

6) **Final Selection Agent (GPT-4)**

This architecture embodies the principles of modularity and specialization inherent in multi-agent systems, allowing for the decomposition of complex tasks into manageable components 1. Each agent operates autonomously but collaborates with others to achieve the collective goal of constructing an optimized investment portfolio.

3.2.1 Coordinator Agent (GPT-o1)

The **Coordinator Agent** serves as the central orchestrator of the system, defining the overarching objectives, constraints, and criteria for security selection. Utilizing the GPT-o1 model, it sets parameters such as desired levels of diversification, acceptable correlation thresholds, and factors contributing to the interest of stocks 5. Additionally, the Coordinator Agent ensures compliance with data handling policies to prevent look-ahead bias, explicitly instructing all subsequent agents to consider only data available up to the current time point.
The Coordinator Agent also has the final say in wether or not a finalized basket of securities generated by the five other agents is accepted. If the basket is accepted, it is used to seed the strategy. If the basket is rejected, it is sent back to the stock generation agent, and the entire system repeats while considering and learning from all previous failed baskets.

Key responsibilities include:

- **Objective Definition**: Establishing the investment goals and criteria for stock selection.
- **Constraint Specification**: Assessing risk levels, correlation thresholds, and diversification requirements.
- **Policy Enforcement**: Ensuring adherence to data handling protocols to eliminate look-ahead bias.
- **Finalizing Security Basket**: Accepting a basket to seed the strategy, or rejecting the basket with a justification of why the basket fails to meet the criteria.

3.2.2 Stock Generation Agent (GPT-4 Functions)

Leveraging GPT-4's advanced language processing capabilities, the **Stock Generation Agent** generates an initial pool of 50 potential stocks. The agent processes a comprehensive dataset of ticker information, focusing on companies listed on U.S. exchanges such as the NYSE 11. Through prompt engineering techniques 3, it filters and selects stocks that meet the initial criteria provided by the Coordinator Agent, while only considering securities presented in the data. In particular, the model is instructed to ignore what it may "think" a ticker represents, and consider tickers as abstractions of securities that it has no knowledge of, other then the presented dataset. The initial data contains features such as market capitalization, exchange, share price, and more.

In order to deal with limited context window's, a divide-and-conquer algorithm is used to split up the security data into batches, generate a set of interesting securities from each batch, and then consolidate until 50 finalists are selected. This would allow us to hypothetically run the process on the over 50,000 securities traded globally without exceeding an LLM's context window on any given generation request.

Key functions include:

- **Data Loading and Parsing**: Utilizing functions to load ticker data from specified sources, ensuring data integrity and relevance.
- **Filtering for Relevance**: Applying filters based on exchange listings, market capitalization, and industry sectors.
- **Batch Processing**: Dividing the dataset into manageable batches to efficiently process large volumes of data.
- **Stock Selection**: Generating lists of up to 10 stocks per batch, providing ticker symbols and brief factual descriptions for each company.

3.2.3 Correlation Analysis Agent (Claude Sonnet 3.5)

The **Correlation Analysis Agent** employs the Claude Sonnet 3.5 model to assess potential correlations among the stocks generated by the previous agent 6. It identifies pairs or groups of stocks likely to exhibit high price correlations due to operating in the same industry, sector, or market.

Key responsibilities:

- **Correlation Estimate**: Estimates statistical correlation between stock pairs and groups.
- **Sector and Industry Analysis**: Categorizing stocks based on industry affiliations to identify inherent correlations.
- **Identification of High Correlations**: Flagging stocks that it predicts will exhibit significant correlation .

3.2.4 Filtering Agent (GPT-4 Functions)

Based on the insights from the Correlation Analysis Agent, the **Filtering Agent** refines the list of stocks. Utilizing GPT-4, it removes stocks that contribute to high overall portfolio correlation, enhancing diversification by selecting stocks that offer unique exposure to different sectors or markets.

Key actions:

- **Correlation Threshold Application**: Excluding stocks that are highly correlated with others in the list.
- **Diversification Enhancement**: Ensuring that the refined list provides exposure to a variety of sectors and industries.
- **Constraint Compliance**: Verifying adherence to the criteria set by the Coordinator Agent.
- **Maintaining intrigue**: Preserving stocks that align with the intrigue factors defined in the initial criteria.

3.2.5 Evaluation Agent (Claude Sonnet 3.5)

The **Evaluation Agent** performs a comprehensive assessment of the filtered stock list. Utilizing the analytical capabilities of Claude Sonnet 3.5, it evaluates diversification across sectors, market capitalization, and geographical regions 6. The agent identifies any over concentration and recommends substitutions to optimize the portfolio's diversification while maintaining its overall quality.

Key evaluation criteria:

- **Sector Balance**: Checking for balanced representation across different industry sectors.

- **Market Capitalization Diversity**: Ensuring inclusion of companies across various market cap segments (large-cap, mid-cap, small-cap).
- **Geographical Diversification**: Evaluating the global presence of companies to enhance international exposure.
- **Overconcentration Identification**: Detecting any biases or overweights in particular sectors or regions.
- **Recommendation Provision**: Suggesting substitutions or adjustments to improve diversification.

3.2.6 Final Selection Agent (GPT-4)

The **Final Selection Agent** consolidates all feedback and finalizes the basket of stocks. Leveraging GPT-4's advanced reasoning and language generation capabilities, it provides justifications for each selection, ensuring that the final portfolio meets the criteria of interestingness, diversification, and low correlation 5. It also ensures adherence to the data usage policies to prevent look-ahead bias.

Key functions:

- **Portfolio Finalization**: Assembling the final list of stocks based on inputs from all previous agents.
- **Justification of Selections**: Providing detailed explanations for the inclusion of each stock.
- **Compliance Verification**: Confirming adherence to all criteria and constraints set by the Coordinator Agent.
- **Documentation**: Preparing comprehensive reports on the selection process and final portfolio composition.

3.3 Eliminating Look-Ahead Bias

Eliminating look-ahead bias is crucial for ensuring the validity and reliability of backtesting results. In our system, we implement stringent measures to prevent any inadvertent use of future information:

The dataset utilized consisted of open use US market data from FinancialModelingPrep 11. The data consisted of features such as industry, sector, market cap group, company description and other data points intended to offer a view into the company without considering recent success directly.

- **Temporal Data Restrictions**: All agents are restricted to use the provided data.
- **Prompt Engineering Constraints**: Prompts provided to LLMs explicitly instruct agents to ignore any external knowledge or data, aligning with best practices in prompt engineering (3, 10).

By enforcing these protocols, we ensure that the system's decisions are based solely on information that would have been available at the time, enhancing the credibility of backtesting and performance evaluations.

This is further highlighted in our back test results. Several securities in our final basket perform very poorly at different times during the back test period. You will also notice companies like NVIDIA and Tesla are not present in our basket. If you ask an LLM to give you top performing and interesting stock tickers, NVIDIA and Tesla are empirically two of the most selected securities due to their prolific performance in the LLM's base training data.

Prior to implementing the look-ahead prevention strategies listed above, NVIDIA **or** Tesla appeared in 90 percent of baskets we generated. NVIDIA **and** Tesla appeared in 60 percent. Given NVIDIA and Tesla are not present in our basket, we are also more confident in our implementation.

While prompt engineering is an inexact science in its infancy, the techniques utilized have proven to have significant impact in research studies 3.

3.4 Learning from Failed Baskets

An innovative aspect of our multi-agent system is its ability to learn from failed baskets - previous portfolio selections that did not meet performance expectations. This iterative learning process allows the system to refine its strategies and improve future outcomes. The system is non-deterministic and

without compute restrictions would run indefinitely until the Coordinator approves a security basket.

Learning mechanisms include:

- **Feedback Integration**: Incorporating insights from performance analysis into the decision-making criteria of various agents.
- **Reinforcement Learning Principles**: Aligning with reinforcement learning concepts where agents adjust their actions based on feedback from the environment [**?**].

This adaptive approach enhances the system's ability to navigate the complexities of financial markets and optimize portfolio performance over time.

3.5 Final Basket Selection

After iterative refinement and learning from past experiences, the system finalized a basket of 16 stocks:

1. **PFE**: Pfizer Inc. is a leading pharmaceutical corporation known for its innovation and diverse healthcare products.
2. **KO**: The Coca-Cola Company is a global beverage leader with a strong market presence.
3. **DGX**: Quest Diagnostics Incorporated provides essential health diagnostics, contributing crucial healthcare services.
4. **PKG**: Packaging Corporation of America is critical in the packaging industry, offering essential products.
5. **TYL**: Tyler Technologies, Inc. provides software solutions, further diversifying the technology investment.
6. **WLL**: Whiting Petroleum Corporation is an independent oil and gas producer, important for energy diversification.
7. **MSFT**: Microsoft Corporation offers large-cap tech exposure with a strong market presence and diverse product offerings.

8. **HSBC**: HSBC Holdings plc provides international financial exposure, enhancing global diversification.

9. **UNH**: UnitedHealth Group Incorporated is a large-cap healthcare company with a diverse portfolio of health services.

10. **ASML**: ASML Holding N.V. is a leading European semiconductor equipment manufacturer, crucial for tech hardware.

11. **TSM**: Taiwan Semiconductor Manufacturing Company Limited offers vital exposure to the Asian semiconductor market.

12. **NEE**: NextEra Energy, Inc. is a major player in the utilities sector, focusing on renewable energy.

13. **BHP**: BHP Group Limited is a global leader in basic materials, providing international exposure to mining and resources.

14. **DIS**: The Walt Disney Company is a strong player in the entertainment sector with diverse revenue streams.

15. **WMT**: Walmart Inc. is a leading retail giant with a robust market presence, enhancing retail exposure.

3.5.1 Justification of Selections

The final basket reflects a well-diversified portfolio across various sectors, industries, and geographical regions. The selections were justified based on:

- **Sector Diversification**: Inclusion of companies from healthcare (PFE, DGX, UNH), consumer staples (KO, WMT), technology (MSFT, TYL), energy (WLL, NEE), financials (HSBC), materials (PKG, BHP), entertainment (DIS), and semiconductors (ASML, TSM).

- **Geographical Exposure**: Incorporation of companies with significant international operations (HSBC, ASML, TSM, BHP) enhances global diversification.

- **Market Capitalization Balance**: A mix of large-cap, mid-cap, and small-cap companies ensures a balance between stability and growth potential.

- **Interestingness Factors**: Companies selected exhibit innovation, market leadership, and unique value propositions, aligning with the interestingness criteria.
- **Low Correlation**: The portfolio composition minimizes correlation among stocks, reducing systemic risk.

3.6 Benefits and Industry Use Cases

The multi-agent system approach offers several advantages over traditional methods:

- **Enhanced Accuracy**: Specialized agents with advanced AI models improve the precision of stock selection.
- **Adaptability**: The system learns from past experiences, adjusting strategies to adapt to changing market conditions.
- **Scalability**: Modular design allows for the addition or modification of agents as needed.
- **Compliance and Risk Management**: Strict adherence to data handling protocols and risk assessment enhances compliance and reduces exposure.

In industry, similar multi-agent systems can be applied to algorithmic trading, automated financial advising, and risk management, where AI models can process vast amounts of data to inform decision-making.

3.7 Limitations

More ambitious implementations of our current system are limited by what is available on the QuantConnect platform. We are unable to make API calls to Anthropic's Claude or and models in OpenAI's GPT family. This means that we are only able to generate our basket with code written outside of the QuantConnect environment, and thus we only generate a single basket prior to the start of our algorithm.
If we were able to utilize these models within QuantConnect, we could regenerate a partial or full basket with each portfolio re-balance.

If the QuantConnect external API's were broader, we would also be able to allow the agents to generate their own back-tests, offering the grounds for intriguing future work.

3.8 Conclusion

Our multi-agent system for security selection demonstrates the effectiveness of integrating advanced AI models and prompt engineering techniques in constructing diversified investment portfolios. By deploying specialized agents and enforcing strict data handling protocols to eliminate look-ahead bias, the system enhances the reliability and robustness of the selection process. The ability to learn from failed baskets further refines its decision-making capabilities, aligning with the evolving landscape of AI applications in finance.

3.9 Future Work

Potential areas for future research and development include:

- **Equipping Agents with Additional Tools**: Allowing the agents to trigger their own backtests to calculate statistics, such as correlation, for a given basket.
- **Expanding Datasets for Backtesting**: Utilizing larger and more diverse datasets to improve the system's generalization and performance across different market conditions.
- **Incorporating Advanced Risk Management**: Developing agents focused on volatility forecasting, options-based hedging, and other risk mitigation strategies.
- **Exploring Alternative Learning Models**: Investigating reinforcement learning and other AI paradigms to further enhance the system's adaptive capabilities.

These enhancements could significantly improve the system's effectiveness and applicability in real-world trading scenarios.

4 Sentiment Analysis

4.1 Introduction

Sentiment Analysis involves computationally identifying and categorizing opinions in a piece of text to determine the attitude towards a specific topic. Specifically, for the purpose of our project, we categorize financial news about a stock into categories of [**positive, negative, neutral**]. Our motivation for this comes from research that indicates that the use of sentiment analysis can help predict price movements in stocks 7.

4.2 Gathering Data

To obtain financial articles about our stocks, we utilized **TiingoNews**. Within QuantConnect, TiingoNews operates on an event-driven basis for each stock that you subscribe, so articles were being sent to the on_data method every second. Also, we observed that not all of the articles that were sent by TiingoNews were related to finance. This is because TiingoNews reports any article that has been tagged with the company's name. For example, *'Presumed Innocent' Is a tale of Two Trials - In Court and At Home, star Jake Gyllenhaal Says - Tribeca Festival, Two trials play out in the new* ***Apple+*** *TV adaptation of lawyer-novelist Scott Turow's 1988 bestselling legal thriller, Presumed Innocent* was a news article received for AAPL.

Consequently, we filtered all the articles we received to determine if they were relevant. Ideally we would have utilized NLP like ChatGPT to accurately determine if a given article was related to a stock. However, due to the sheer volume and frequency at which the articles arrived, this was computationally unfeasible. Instead, we created a dictionary of keywords that we believed articles related to a stock should contain. If an article didn't contain any of the keywords, we discarded it because it did not have any valuable information.

4.3 Processing Data

To determine the sentiments of the articles we received, we used FinBERT within QuantConnect. FinBERT is a fine-tuned version of BERT, trained specifically on financial news and data. BERT, which stands for Bidirectional Encoder Representations from Transformers, is a large language model de-

veloped by Google and has been highly successful in understanding and analyzing the sentiment of language 8. FinBERT outputs the sentiment scores for each article in the format [**positive, negative, neutral**]. The sum of these individual scores is always 1, with each score ranging from 0 to 1.

At the end of each rebalance period, the sentiment scores for a specific stock were aggregated by summing all the scores for articles related to that stock and dividing by the total number of articles during the rebalance period. It is important to note that these operations were applied to each stock individually, rather than to the portfolio as a whole. This approach was chosen to generate predictive signals for the performance of each stock independently.

4.4 Limitations

Due to the computational restraints of QuantConnect and the computational power required to use FinBERT, we set a 10 article per week maximum for any given stock. If a stock received more than 10 articles a week, only 10 of those were randomly selected for sentiment analysis. Unfortunately, this was necessary in order to not exceed QuantConnect's time-limit and memory-limit on any given backtest. We recognize this as a potential limitation of the accuracy of our sentiment analysis.

Similarly, utilizing a custom dictionary of words to determine the relevance of an article may result in false-negatives or false-positives.

4.5 Future Work

Future work should entail increasing computational power to increase the number of articles that can be processed to more accurately reflect the total market's sentiment. Also, NLP processing should be used to more accurately determine if a given article is relevant to the underlying company's stock and financials.

5 Stock Movement Prediction Using Options

5.1 Motivation

We noticed during discussions about options that there was extra information included in the options' price that was driven by factors other than intrinsic factors. Specifically, we wondered if there was information to glean from the prices differences of options with the same relative strike to the underlying price and the same maturity, as in the Black-Scholes model these values should be the same. While doing research on this difference, we stumbled upon a paper by Lin, Lu and Driessen (2013) 9 indicating prior research into using option pricing as a signal to predict movements in the underlying stock.

The paper discussed using 2 main signals to determine relationships to the movement of the underlying stocks:

- **Implied Volatility Skew**: The difference in implied volatility of an out-of-the-money (OTM) put and an at-the-money (ATM) call. The paper describes this as a negative predictor of stock return.
- **Implied Volatility Spread**: The difference in implied volatility of at-the-money puts and calls. The paper describes this as a positive predictor of stock return.

This paper utilized linear regression techniques and these two indicators, as well as some additional data, to build a model that determined that the options market was a statistically significant predictor of the stock market.

We wanted to utilize the learnings from this paper and see if we could expand upon them. To do this, we decided the things that we could try were to use different additional data, as well as a more powerful modeling technique, to see if we could build a model that could predict movement.

5.2 Modeling Decisions

We had a few interesting decisions to make with regards to the modeling of the problem. First was the time scale of the predictions. QuantConnect does not have great support for options history in the research environment, generally only allowing history for options with 10-20 days expiry. We decided that we would use 14 days, as it is the only round multiple of weeks within that time period.

Another modeling decision was the desired ouputs of the model. Because we are working in a 2-model system, we wanted to make integrating the signals as painless as possible. Since we are using FinBERT, we determined the best course of action would be to output the same outputs as FinBERT. FinBERT's output is a 3-tuple of probability scores, the output of a softmax over 3 inputs. The inputs are the positive score, the negative score, and the neutral score. To replicate this output, we used the difference between the underlying price at the current date and the underlying price 14 days in the future. We then classified this into a positive move, defined as the future price being $> 3\%$ higher than the current price, a negative move, defined as the future price being $> 3\%$ lower than the current price, and a neutral move, defined as the future price being within 3% of the current price.

Once we split the outputs into their target classes, we realized there was a significant imbalance between the target classes, so we randomly down-sampled from each class to match the minimum class's number of samples. This ended up being around 8200 samples per class.

The next modeling decision we needed to make was the inputs. The inputs we decided on were:

- **Put Call Parity**: The difference between the prices of an ATM put and an ATM call. This should give us similar information to the IV spread.
- **Difference between OTM put price and ATM call price**: This should give us similar information to the IV skew.
- **Difference between OTM call price and ATM put price**: This should give us a different view on the IV skew, specifically it may be a good indicator of positive sentiment for the stock. Because pricing of puts and calls is asymmetric, this will give different information than the previous input.
- **Difference between OTM call price and ATM call price**
- **Difference between OTM call strike and ATM call strike**: In conjunction with the prior input, this should give information about how quickly call price rises varying strike.
- **Difference between OTM put price and ATM put price**

- **Difference between OTM put strike and ATM put strike**: In conjunction with the prior input, this should give information about how quickly put price rises varying strike.
- **ATM Call price**: Normalized by dividing by the underlying price.
- **ATM Put price**: Normalized by dividing by the underlying price.
- **Historical volatility**: Important factor in determining how price relates to the Black-Scholes price.
- **Difference between ATM call price and Black-Scholes price**: Theoretically should give information about demand for ATM calls.
- **Difference between ATM put price and Black-Scholes price**: Theoretically should give information about demand for ATM puts.
- **Difference between ATM call/put strike and underlying price**: Needed for the model to normalize.
- **Difference between ATM call/put expiry and desired expiry**: Needed for the model to normalize.

The last modeling decision we had to make was the actual modeling technique to use. We wanted to use something more powerful than linear regression, but not so powerful that it would be guaranteed to overfit. We used a 3-layer feedforward neural network with significant regularization in the form of dropout layers. At each layer (excluding the output layer) we applied a layer norm, which normalizes the outputs of the layer going into the next layer to prevent large gradients. The loss function we used was the cross-entropy loss as this was a classification problem.

5.3 Training and Evaluation

We split the data into a training, validation and testing set, with breakdown 80%, 10%, 10% respectively. We trained for 250 epochs on our training set, and saw similar performance in training and validation sets, which gave us belief that our model was not overfitting.

1 displays a graph of our training loss and validation loss over the training cycle. Training took around 30 minutes with no GPU available due to

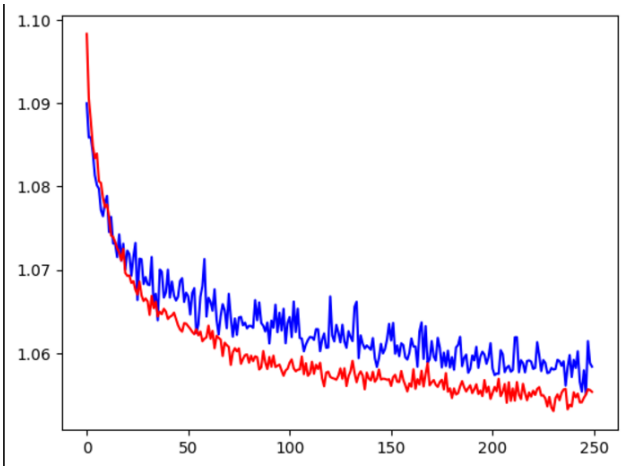

Figure 1: Training and validation loss for options model

the QuantConnect research environment. However, loading the data took around 30 minutes due to the earlier mentioned issues with options data in QuantConnect's research environment. We used the given basket from the multi-agent framework as discussed in Section 2, as well as a random selection of 30 Nasdaq 100 stocks to set up the training set.

We tested the model on the test set. We only considered instances where the output score was > 0.5. This means that we would have a ¿ 50% confidence that the stock would move a certain direction. Considering that the expected accuracy would be 0.33 since our dataset is balanced, our model had a test accuracy of 0.49. We also tested the 2-class accuracy and got 56% accuracy with 2 classes (positive move and negative move). While these models are clearly not perfect, they show clear evidence of learning. We confirmed these changes are statistically significant by running a null hypothesis significance test at 99% confidence interval.

5.4 Limitations

This part of the algorithm worked relatively well, but there were definitely things we would want to improve given further time to work on it. First, because we are modeling solely directional movements and not magnitude, we do not know when the algorithm makes a decision whether it believes the asset will increase by 3% in price or by 30%, just that it is confident the

asset will increase. We use signal strength as a rough indicator for magnitude of movement, however there is nothing in our modeling approach that necessarily indicates this is true.

We also would like a more interpretable model going forwards. When making trading decisions, we have no insight into why the model believes the price will go up or down. All we receive are probabilities and we make decisions based off of those - in a real fund, we would want to be able to vet our incorrect decisions and understand why the model is making the choices that it is.

6 Combining Signals To Make Trading Decisions

Our signal generators effectively function as an ensemble model. Ensemble models are a class of model where multiple weak models get combined into a single strong model. An example of this is a random forest, where multiple weak decision trees get combined into a single powerful model. We compiled a linear combination of the outputs of the two models to get an overall positive and negative score for each asset. Specifically, the weights in the linear combination were .10 for sentiment scores and .90 for option scores. This is because we discovered that option scores were far more effective at predicting price movements than sentiment scores. This is likely due to the limitations of our sentiment scores as discussed in Subsection 4.4.

We take the max of the positive and negative score, tracking whether we chose a positive or negative score for each share. Next, we threshold the scores by checking if they are greater than 0.5. Because both of our models are probabilistic, this would mean that we have a ¿ 50% confidence that the asset will move in the specified direction. If the value is below the confidence threshold, we do not invest in the asset for the upcoming time period.

We then divide the signal by its annualized historical volatility and pass it through another threshold to make sure we are not taking on any exceedingly risky assets 7.3. This is somewhat akin to computing a signal to noise ratio. In addition, we add extra risk management by calculating the daily VaR for each asset 7.2. If the VaR is greater than .05, we will not invest. This allows us to ensure with 95% confidence that we will not lose more than 5% of our investment value with this specific security on any given day.

Once we have a final set of scores, we weight our portfolio based on the relative strength of these scores (i.e. divide each score by the sum of all the scores). We long assets with a positive score at the portfolio weight we calculated, and we short the assets with a negative weight at 80% of its portfolio weight. We artificially decrease the weight of our short positions because they are inherently more risky as their downside is infinite. Also, we wanted to ensure that we can maintain enough margin in case these trades go awry.

Note that it is entirely possible for our algorithm to not invest at all during any given rebalance period. This is because our algorithm only executes a trade for a given stock on the premise that it meets all three criteria:

- The stock generates a trading signal that is strong enough to act on (as determined by passing a threshold value).
- The stock's daily VaR is less than 5%.
- The signal-to-annualized volatility ratio is less than another threshold.

If a stock does not meet all three of these criteria, we do not place an order for it. Thus, if no stocks in our basket meet these criteria, no orders are placed.

7 Risk Management

There are 3 main parts to our risk management strategy. We employed stop-loss, VaR thresholding and volatility based weighting to manage the risk that the algorithm was taking on.

7.1 Stop Loss

We implemented a maximal stop-loss of 10% to help prevent our drawdown from exceeding our desired amount. Instead of using the average price of the asset for the stop-loss criteria, we used the maximal value of the asset (the minimum if the position was short). This way, if we were up 10% then down back to the original value, this would be considered a loss rather than a neutral move, as it would have an impact on our drawdown metric.

7.2 VaR Thresholding

We implemented a simple thresholding around Value-At-Risk. When deciding whether to enter a position, we checked the 50-day VaR metric (at a 95% confidence threshold) and only entered the position if the VaR was ¡ 5%. This prevented us from taking on exceedingly risky assets, and also helped ensure we meet the criteria of Daily PnL Volatility ¡ 5%. If the position was short, we reversed the VaR check to be the 95th percentile gain rather than the 5th percentile loss.

7.3 Volatility weighting

As discussed in the Combining Signals section, we divided our signal strength metric by the 50-day implied volatility (annualized). This is effectively an implementation of a signal-to-noise ratio (SNR), that we then weighted our portfolio off of. We also thresholded again here such that if we saw a SNR that was too low, we did not invest in the security as we cannot be confident that our signal strength was actually an indication of movement or just of high volatility. Once we finished thresholding and balancing the weights, we had a final portfolio weight to apply to our portfolio.

7.4 Covid-19 Stress Test: 2/1/2020 - 5/1/2020

Figure 2: Covid-19 Backtest

For the 2020 Covid stress period, we experience a **loss of 0.5%** and a **Drawdown of 7%**. Our backtest is linked here. By implementing a stop-loss strategy, we were able to liquidate our positions before we incurred losses that were too great. Also, our strategy did not invest between the end of February and the beginning of March, as well as from mid-April to April 20th. This is represented as the flat bars in Figure 2. This is because our algorithm disregarded all trade signals from our stocks due to their VaR being greater than the threshold of .05.

For the 2008 stress period, we were unable to run our model successfully. Both the options history in QuantConnect and the Tiingo history do not go back to 2008, so our model does not receive any signals to execute trades.

8 Results

8.1 In Sample Trading Period: 1/1/2017 - 1/1/2021

Figure 3: In Sample Backtest

Figure 3 shows our backtest from the In Sample period. Our **Net Profit** was **23.02%**, the **Annual Return** was **5.31%**, the **Sharpe Ratio** was **0.256**, and the **Drawdown** was **14.8%**. However, we attribute these subpar statistics due to the financial downturn during Covid-19. In the first 3/4 of 3, the Drawdown was hovering between 0% and 5% and returns were steadily increasing. However, once the algorithm reaches February 2020 and the onset of Covid begins, we incur significant losses and a significant dip in the Drawdown. Furthermore, once July 2020 comes and the market begins to rebound, so does our profitability.

8.2 Out Of Sample A Trading Period: 1/1/2016 - 1/1/2017

Figure 4: Out Of Sample A Backtest

From 1/1/2016 - 1/1/2017, our algorithm had a **Net Profit** of **18.66%**, **Annual Return** of **18.66%**, **Sharpe Ratio** of **1.584**, and **Drawdown** of **7.3%**. We successfully met the metrics of excessive Annual Returns, a Sharpe Ratio greater than 1, and Drawdown less than 10%. Without any unexpected market downturn, our model was able to successfully profit by predicting price changes based on our trading signals.

8.3 Out Of Sample B Trading Period: 1/1/2022 - 11/1/2022

Figure 5: Out Of Sample B Backtest

Our second Out Of Sample backtest also performed above the desired metrics, although it did slightly worse than the OOS A backtest. The **Net Profit** was **14.00%**, the **Annual Return** was **16.98%**, the **Sharpe Ratio** was **1.189**, and the **Drawdown** was **8.6%**

8.4 Out Of Sample C Trading Period: 3/1/2024 - 9/1/2024

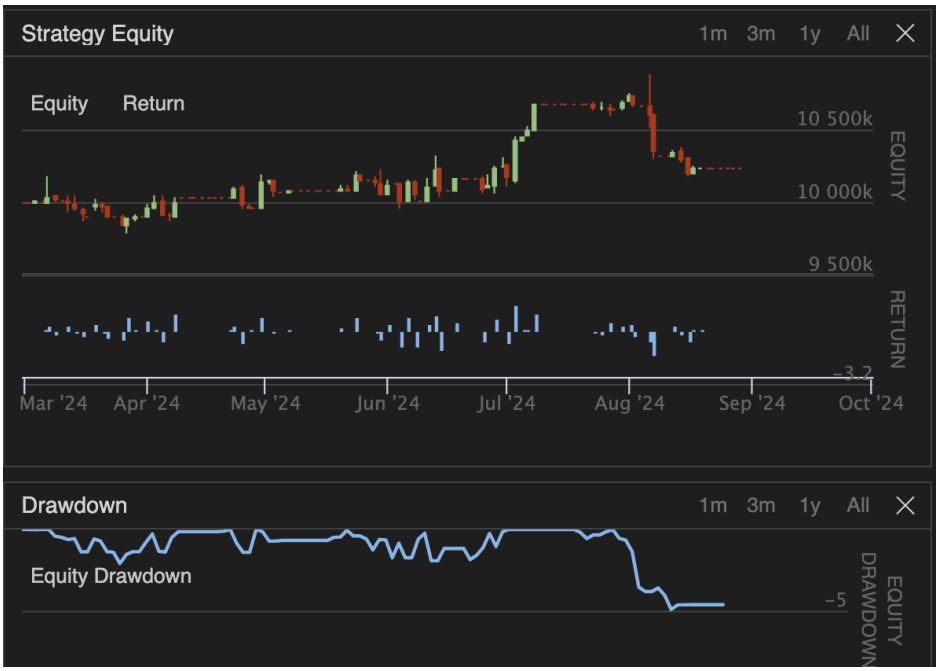

Figure 6: Out Of Sample C Backtest

The last Out of Sample backtest performed the worst. It got a poor **Sharpe Ratio** of **-.239**, **Net Profit** of **2.39%**, **Annual Return** of **4.81%**, and a **Drawdown** of **6.4%**. Interestingly enough, during this backtest, our algorithm had a **Win Rate** of **58%**. Even though our Sharpe Ratio was negative, we still met the Drawdown requirement and generated positive returns. Upon further inspection, we lose a large portion of our capital from August 2024 to September 2024. After checking the order logs, our strategy longs ASML at the end of July, and then shortly thereafter ASML's stock price plummets. Subsequently, our algorithm takes on a short position in ASML, but then their stock price begins to rebound to its original value by September. This suggests that our algorithm fails to perform well in short time frames when the market also predicts incorrectly. When ASML's stock started to plummet, the market sentiment was extremely negative, so our algorithm began to short it. However, ASML's price actually started to rebound shortly after, thus making us incur more losses. We believed this volatility would inhibit

us from reentering a position, so in the future we may have to further lower the volatility weighting threshold. Also, our losses were amplified because no other stock generated a trade signal strong enough to act on, so ASML was weighted very heavily in our portfolio.

8.5 Live Paper Trading: 12/9/2024 - 12/13/2024

Our group does not have any results for the Live Paper Trading period. This is because our algorithm enters the first trade after the end of the first rebalance period to get sentiment information. Since we rebalance weekly and the Live Paper Trading period is only one week, our strategy did not enter any positions.

8.6 Backtest Summary

Backtest	Time Period	Net Profit (%)	Annual Return (%)	Sharpe Ratio	Drawdown (%)
IS	1/1/2017 - 1/1/2021	23.02	5.31	.256	14.8
OOS A	1/1/2016 - 1/1/2017	18.66	18.66	1.584	7.3
OOS B	1/1/2022 - 11/1/2022	14.00	16.98	1.189	8.6
OOS C	3/1/2024 - 9/1/2024	2.39	4.81	-.239	6.4

Table 1: Backtest Results Overview

The above table summarizies our backtesting results. We were able to meet all of the required metrics in OOS A and OOS B. We were performing well in the IS period; however, our algorithm struggled to succeed during Covid-19 which is why our Annualized Returns, Drawdown, and Sharpe Ratio are not up to standard. In OOS C, our algorithm meets the Drawdown criteria and generates positive returns; however the Sharpe Ratio indicates that our strategy does not perform well on a risk-adjusted basis during this time.

9 Conclusion

In conclusion, our strategy of combining sentiment analysis and option forecasting was effective at generating steady annual profit. We generated positive returns in each of our backtests with the exception of the stress tests. Also, our three-pronged risk management strategy also proved to be an effective way of managing risk. During our stress test, we only had a loss of 5%,

and we only dropped below the 10% Drawdown threshold during Covid-19 in the IS period.

We also strongly believe that our algorithm will do even better in the future and during the live-trading competition. We have already seen that our algorithm successfully mitigates risk and generates actionable trading signals. However, we believe that we will have even better metrics in the future because our multi-agent framework selected our universe of stocks by optimizing our returns for the future. Thus, we believe that our algorithm will be even more successful.

10 Acknowledgments

Our group would like to sincerely thank Dr. Ye for his guidance and input throughout our project. We would also like to thank Dr. Ye for making class engaging throughout the whole semester, and sharing his infectious enthusiasm about Algorithmic Trading with us all. We would also like to thank our TA, Rick Presman, for his support, availability, and assistance throughout the semester and the duration of our project.

11 Supporting materials

To explore an interactive architecture diagram of our Multi-Agent Security Selection system, please visit our **Figma diagram**.

References

1. Ganesh, Sumitra, Nelson Vadori, Mengda Xu, Hua Zheng, Prashant Reddy, and Manuela Veloso. "Multi-Agent Simulation for Pricing and Hedging in a Dealer Market." *36th International Conference on Machine Learning, Workshop on AI in Finance: Applications and Infrastructure for Multi-Agent Learning*, 2019. Available at: `https://www.jpmorgan.com/content/dam/jpm/cib/complex/content/technology/ai-research-publications/pdf-10.pdf`.

2. Financial Modeling Prep. "Financial Modeling Prep - Financial Data and Stock APIs." 2024. Available at: `https://site.financialmodelingprep.`

com/.

3. Sahoo, Pranab, Ayush Kumar Singh, Sriparna Saha, Vinija Jain, Samrat Mondal, and Aman Chadha. "A Systematic Survey of Prompt Engineering in Large Language Models: Techniques and Applications." *arXiv preprint arXiv:2402.07927v1 [cs.AI]*, 2024. Available at: https://doi.org/10.48550/arXiv.2402.07927.

4. Wu, S., Koo, M., Blum, L., Black, A., Kao, L., Scalzo, F., and Kurtz, I. (2023). *A Comparative Study of Open-Source Large Language Models, GPT-4 and Claude 2: Multiple-Choice Test Taking in Nephrology.* Available at: https://arxiv.org/pdf/2308.04709.

5. OpenAI. "Introducing OpenAI o1: Advanced Reasoning AI Models." 2024. Available at: https://openai.com/o1/.

6. Anthropic. "Claude 3.5 Sonnet: Next-Generation AI Models." 2024. Available at: https://www.anthropic.com/news/claude-3-5-sonnet.

7. Shobayo, Olamilekan, et al. "Innovative Sentiment Analysis and Prediction of Stock Price Using FinBERT, GPT-4 and Logistic Regression: A Data-Driven Approach." Big Data and Cognitive Computing, 2024 Available at: https://www.mdpi.com/2504-2289/8/11/143?

8. Devlin, Jacob , et al. "BERT: Pre-training of Deep Bidirectional Transformers of Language Understanding", 2019 Available at :https://research.google/pubs/bert-pre-training-of-deep-bidirectional-transformers-fo

9. Lin, Tse-Chun and Lu, Xiaolong and Driessen, Joost, Why Do Options Prices Predict Stock Returns? (July 1, 2013). Netspar Discussion Paper No. 07/2013-079, Available at: http://dx.doi.org/10.2139/ssrn.2400955

10. Chen, Banghao and Zhang, Zhaofeng and Langrene, Nicolas and Zhu, Shengxin, Unleashing the Potential of Prompt Engineering in Large Language Models: A Comprehensive Review (October 2023). Available at: https://arxiv.org/abs/2310.14735v5

11. Financial Modeling Prep (FMP). Comprehensive Stock Market API and Financial Data API. Available at: https://financialmodelingprep.com. Accessed on [Insert Access Date].