

# Democratizing Algorithmic Trading Powered By Deep Learning

CM3015 Machine Learning and Neural Networks

**Project Idea:** Deep Learning on a public dataset

## 1. Background

The New York Stock Exchange(NYSE) was founded in 1792 but it gained prominence and structure in the 19th century. Since the late 1990s the advent of electronic trading platforms and the proliferation of online brokerages opened up the stock market to individual investors and traders. This era saw a surge in day trading activity, fueled by increased access to real-time market data, lower trading costs, and popularity of internet-based trading.

Despite the newly realized leverage gained by retail investors over the recent years, and the attempts by regulators such as the SEC to maintain fair play within the Markets, the ground is significantly lop sided. Institutional investors such as Investment Banks, Hedge Funds, and Sovereign Wealth Funds still maintain the higher ground.

The advantagious effects of the advance of financial technology is not exclusively felt by the average retail investor. In many cases the early adopters/inventors of these technologies are the capital juggernauts themselves, introducing Algorithmic Trading bots with the aims to:

- Achieve favorable pricing
- Short-term trades that aim to profit from small price movements, for example, due to arbitrage
- Behavioral strategies that aim to anticipate the behavior of other market participants
- Trading strategies based on absolute and relative price and return predictions

With substantial financial resources (Billions of Assets Under Management [AUM]), unique access to material information, and the professional experties to deploy said information, the average retail investor who is on the other side of the trade does not stand a chance.

Now, with the advant of Machine Learning, the power of algorithmic trading has multiplied the gap between institutional investors and retail investors exponatially. This in advertantly leads to disproportionate advantages between the two respectively. We have all heard of the extreme lengths Wall Street firms have gone in order to increase their alpha (aka edge) in the market. Some of these strategies are listed below but not limited to [1,2]:

- Data mining to identify patterns, extract features and generate insights
- Supervised learning to generate risk factors or alphas and create trade ideas
- Aggregation of individual signals into a strategy
- Allocation of assets according to risk profiles learned by an algorithm
- The testing and evaluation of strategies, including through the use of synthetic data

- The interactive, automated refinement of a strategy using reinforcement learning

A study in 2019 showed that around 92% of trading in the Forex market was performed by trading algorithms rather than humans [20].

It is widely used by investment banks, pension funds, mutual funds, and hedge funds that may need to spread out the execution of a larger order or perform trades too fast for human traders to react to.

Majority of retail investors do not have the time, or "know-how" to take advantage of these methodologies [5], hence they rely on a discretionary approach to investing. Even when considering highly motivated retail investors, they may only go as far as utilizing fundamental (market indicators) and regressive approaches to the market.

### **Alternative Data**

The vast majority of retail investors steer clear from incorporating Alternative Data into their investment strategies, due to its complexity and time intensive learning curve.

*"I understand my brain better than I understand some algorithm. If things go wrong, I can blame myself, but I can't blame an AI." - Johan Yao (An interviewed day trader for this project)*

Some examples of the incorporation of alternative data include but not limited to [2]:

- Online price data on a representative set of goods and services can be used to measure inflation
- The number of store visits or purchases permits real-time estimates of company or industry-specific sales or economic activity
- Satellite images can reveal agricultural yields, or activity at mines or on oil rigs before this information is available elsewhere

The incorporation of alternative data offers an informational advantage that has the potential to yield a high alpha through an edge derived from material information. This edge could provide insights into potential liquidity events that could change the directional movement of a stock's future. One may think this approach is a well-decorated encouragement to promote insider trading on a technological level, but it is important to note that the source of the information is generally publicly available, rendering these methods legal. The material understanding derived from these publicly available sources of information is not obvious to the naked eye until a systemized machine applies pattern recognition strategies on the available data. With advantages to the use of alternative data becoming more apparent, institutional investors are expanding beyond the fundamental approach to the market, spending \$3 billion+ (as of 2020) on data, talent, and technological capabilities in order to capitalize on the new approach to investing. Retail investors have sadly been left behind.

## **2. Aims, Objectives and Datasets**

## 2.1 Aims and Objectives

We wish to make efforts in leveling the playing field between institutional & retail investors. This will take the form of developing an algorithmic trading bot that is based on some of the technical approaches they currently exercise, as well as using some discretionary, fundamental analysis. More significantly, it will be incorporating the use of alternative data machine learning analysis that hedge funds & other institutional investors utilize. We will be entering the world of quantitative finance through the use of Algorithmic trading powered by Deep Learning with Alternative Data.

### 2.1.1 Algorithmic Trading Strategy

Meet "**Viklund**" our machine learning trading bot (*named after the musical artist, Simon Viklund: artist of the album 'Steal From The Rich, Give To Myself')***[19]**

#### 2.1.1.1 Trading Strategy

Our first iteration of Viklund will utilize a technical analysis approach to deciding when to buy/sell a stock. This means it will use market indicators such as Exponential Moving Average (EMA) to signal whether or not to buy, short, or liquidate a position. We will also be utilizing a deep learning model to carry out an event driven trading strategy. This event driven strategy will take the form of our Neural Network signals influencing both Viklunds trading position as well as its position sizing during an earnings call for a given company. This model will analyze aleternative data in order to signal to Viklund how much capital to allocate to said position, based on its analysis. These signals will influence Viklund's buying/selling power & direction.

### 2.1.2 Alternative Data

Our source of alternative data needs to be one that is publicaly available to all, in order to properly meet the requirment of the average retailer's access to information. We will be choosing text data as our format, this text data can be derived from news articles, tweets, and transcribed earnings calls from public companies.

We will not be training the Neural Network on tweets, as there have been numerous ocassions where market makers have utilized the public square to manipulate the markets in their favour. Our selection of text data needs to be one with a certail level of crediility an accountability to its name. This is why we will rely on news articles produced by some of the most reputable financial news outlets. While we cannot be certain about the freedom from influence exerted by the market makers mentioned above on the news produced by these media companies, we can still attribute a certain level of certainty to the strong track record of these companies.

Specifically we will be scrapping the web for news articles involving the company around the time of their scheduled earnings call. Our objective is to obtain sentiment on the article concerning the companies quarterly performance. Specifically, if they missed, hit, or surpassed

their estimated Earnings Per Share (EPS). This information will be very valuable to us as price movement is most active surrounding geopolitical, social, and financial reporting events.

### 2.1.3 Considerations

- For our first consideration, we will be using the SMA indicator as our fundamental decision making for two reasons:
  1. **Familiarity:** After sample interviews of retail investors, we have deduced that a large percentage of them use SMA due to both its effectiveness and simplicity of understanding. Adoption of AI has been an issue to those who do not understand it, it is often viewed as an added risk to business operations due to its black box nature of decision making. We need to make sure there are strands of familiarity found in Viklund's first iteration.
  2. **Risk:** Due to the average investors relatively conservative buying power, and appetite for risk, we need to make sure that the signal to buy and sell is being sent from as solid a ground as possible. Neural Network (NN) models are notorious for being difficult in tracing back decision making, at the same time, when it comes to complex decision making, they are the best option. This is why in the second iteration we will attempt to create a NN model for those with a larger appetite for risk & returns. The signals sent from the NN model will be excised, if at all, within certain risk parameters.
- As this is a data analysis intensive task we will not have the time to create a UI for our retail investors as this is outside the scope of this project. Time aside, we do not have the connections & capital to connect a self made program to multiple financial markets let alone one. So we will utilize an existing platform that allows us to upload our own trading algorithms and offer us the ability to connect directly with brokers and live trade our algorithm. I have chosen the platform QuantConnect, an open-source, cloud-based algorithmic trading platform designed for quantitative traders and developers. It allows users to design, test, and deploy trading algorithms that can operate on various financial markets, including stocks, forex, cryptocurrencies, and more. Their mission coincides with ours, to democratize algorithmic trading.
- We will not be live trading this algorithm as this will require us to pay brokerage fees. The extent of Viklund's usage will be constrained to backtesting.

### 2.1.4 Ethics

As we are not financial advisers. We will not be recommending the use of Viklund to any retail investor, this is simply for educational purposes on how the approach to incorporating alternative data & machine learning could increase your current strategies alpha in the market.

## 3. Literature Review

### 3.1 Twitter mood predicts the stock market

[7] Johan Bollena, Huina Maoa, Xiaojun Zeng, Received 15 October 2010, Revised 2 December 2010, Accepted 5 December 2010, Available online 2 February 2011, *Twitter mood predicts the stock market*, Journal of Computational Science, Volume 2, Issue 1, March 2011, Pages 1-8

**Link:** ": <https://www.sciencedirect.com/science/article/pii/S187775031100007X>  
<https://www.sciencedirect.com/science/article/pii/S187775031100007X>).

The study that explores the possibility of predicting the stock market based on the sentiment or mood of Twitter users [3]. Traditionally, stock market prediction was believed to be unpredictable due to the Efficient Market Hypothesis (EMH)[6], which suggests that stock prices follow a random walk pattern influenced by new information rather than past prices. However, recent research challenges this view and proposes that public sentiment, including emotions and moods expressed on social media platforms like Twitter, may play a significant role in stock market fluctuations.

The study analyzes Twitter data to extract indicators of public mood and sentiment. This is done using sentiment tracking techniques that process large-scale Twitter feeds to represent the overall mood of the public. The researchers use two tools, OpinionFinder and GPOMS, to measure variations in public mood over time and then correlate these mood time series with the Dow Jones Industrial Average (DJIA), a stock market index, to assess their predictive ability.

The results indicate that certain mood dimensions, such as Calmness and Happiness, as measured by GPOMS, seem to have a predictive effect on the stock market, while general happiness measured by OpinionFinder does not show the same predictive power.

Overall, the study suggests that the sentiment of Twitter users can potentially be used as a predictor for stock market changes, and including specific mood dimensions may improve the accuracy of stock market prediction models. As we aim to introduce the use of alternative data into the strategies of the average retail investor we will be focusing on the marriage of technical analysis and sentiment analysis of texts from sources like twitter.

**Takeaway:** Although, the paper is published in a scientific journal, its lower reliability stems from the inherent risks associated with using social media sentiment as a basis for financial decisions. Twitter has been known to be easily susceptible to bot attack, with the aim of poisoning the flow of information for market manipulation reasons. This body of work increases our confidence in our mission. We will not be utilizing twitter text data as our alternative data source for the reasons just mentioned. However, we will be creating our own Neural Network sentiment analyzer as opposed to using an out of the box version. It is important to note that their experiment was for the purpose of exploration into what could be possible, however, our body of work needs to be grounded within a sense of familiarity to the average retail investor. At first, we will not outsource the core decision making process on whether or not to trade on to a sentiment analyzer as they have. Rather, we will constrain the use of a sentiment analyzer to handling position sizing, allowing us not only to reduce perceived risk but also the fear in lack o

understanding its core decision making process due to the black box effect. This is inline with

## 3.2 Quantifying Wikipedia Usage Patterns Before Stock Market Moves

[8] Moat, H., Curme, C., Avakian, A. et al. Quantifying Wikipedia Usage Patterns Before Stock Market Moves. *Sci Rep* 3, 1801 (2013).

Link <https://doi.org/10.1038/srep01801> (<https://doi.org/10.1038/srep01801>)

The study explores changes in how often financially related Wikipedia pages are viewed in order to provide early signs of stock market moves, offering valuable insights into the decision-making process before trading. The data spans from December 10, 2007, to April 30, 2012. The study aims to understand if there is any correlation between the information gathering behavior on Wikipedia and stock market activities.

In the study, the researchers calculate two measures of Wikipedia user activity: the average number of page views and page edits for a specific Wikipedia page in a given week. They define weeks as ending on Sundays. To analyze changes in information gathering behavior, they select one of the activity measures (page views or page edits) and calculate the difference between the activity in the current week and the average activity over the previous  $\Delta t$  weeks. This allows them to quantify the changes in user activity over time. The formula used for calculating the difference is  $\Delta n(t, \Delta t) = n(t) - N(t - 1, \Delta t)$ , where  $N(t - 1, \Delta t)$  represents the average activity over the previous  $\Delta t$  weeks.

The first strategy uses data on Wikipedia page views or edits to trade on the Dow Jones Industrial Average (DJIA). If the volume of views or edits increases in a given week, they sell the DJIA and buy it back the following week. If the volume decreases or remains the same, they buy the DJIA and sell it the following week.

The cumulative return of the strategy is calculated by taking the natural log of the ratio of the final portfolio value to the initial value. Transaction fees are neglected for simplicity. They compare the returns of this strategy to the returns from the random strategy, where decisions to buy or sell the DJIA are made randomly with a 50% probability each week.

The study finds that the random strategy results in no significant profit or loss, while the Wikipedia data-based strategies are compared to this random strategy in terms of standard deviations above or below the mean cumulative return. The analysis uses non-parametric tests to account for the non-normal distribution of returns.

The study's conclusion suggests that historical Wikipedia usage data from December 2007 to April 2012 might provide useful insights into future financial market trends. The analysis shows evidence of a link between increased page views on articles related to companies or financial topics and subsequent stock market declines. However, there is no such relationship observed for changes in page views on articles about actors and filmmakers, which have less clear financial implications. Overall, the study indicates that Wikipedia data could be valuable in understanding financial market behavior.

**Takeaway:** This article reinforces our hypothesis on the indicators that lie within alternative data in the field of financial engineering. One of our core fundamental beliefs with this field is in its ability to offer valuable insights into the public's decision-making process before trading. Although a single wikipedia page view on a financial topic may not be indicative of the public's intention to trade a related security, multiple page views within a concise period of time could be the launchpad signal to an unprecedented set of actions a trading algorithm could perform when it comes to risk management, and overall hedging for a potential downturn despite a stronger signal persuading the algorithm to carry out an alternative trade. Even though we will not be utilizing Wikipedia data or the exact approach shown here for our project, this study certainly exposes the possibilities when incorporating alternative data into trading strategies.

### 3.3 Revisiting the use of web search data for stock market movements

[9] Zhong, X., Raghib, M. Revisiting the use of web search data for stock market movements. Sci Rep 9, 13511 (2019).

**Link** <https://doi.org/10.1038/s41598-019-50131-1> (<https://doi.org/10.1038/s41598-019-50131-1>)

The paper talks about using search volume information of financial phrases to forecast stock market movement. It clarifies earlier research that suggested specific financial search phrases had greater predictive potential than those connected to other issues. Some studies choose a predetermined set of terms linked to finance for automated trading using a quantitative methodology. The long-term predictability of these parameters must remain constant for this assumption to hold, which runs counter to the behaviour of the financial markets. This conclusion is supported by the study's experiment, which also demonstrates the poor predictability of predefined search words used in earlier approaches. Web-based search-based techniques operate under the premise that rising search volume signals greater investor uncertainty and prompts a short position, while falling search volume prompts a long position.

The study suggests a fresh method for forecasting market changes based on information from web searches. This system takes an adaptive approach and automatically selects pertinent search terms for each prediction, unlike earlier strategies that used fixed search terms.

Every week, the prediction model is retrained using the adaptive technique, enabling it to adjust to shifting market conditions and search behavior's. The study uses the Google Correlate service (GCS), which returns up to 100 search phrases rated by their connection with the Dow Jones Industrial Average (DJIA) index, in order to choose the most pertinent search terms. The DJIA index is used as the prediction's target time series.

The study employs a linear regression model that combines the search volumes of the phrases supplied by GCS to guarantee that the search terms are pertinent. The DJIA is predicted by this model for the following two weeks. They use an automated feature selection method to avoid overfitting, which can produce incorrect predictions. The GCS search phrases are curated using this method, and only the most pertinent ones are selected to be used in the prediction model.

The project intends to increase the accuracy of market predictions based on web search data, perhaps providing useful insights into market movements, by adopting an adaptive strategy and utilizing GCS to automatically identify pertinent search keywords.

The study assesses a regression model and online search data-based adaptive technique for stock market movement prediction. In order to cover the period from January 6th, 2008, to March 26th, 2017, they repeat a trading experiment put forth by Preis et al.

Each week, the adaptive method entails trading two options: "long" and "short." The Dow Jones Industrial Average (DJIA) is purchased in the "long" option at the closing price of the first trading day of the following week ( $t+1$ ), and it is sold at the closing price of the first trading day of the following week ( $t+2$ ). In the "short" option, on the other hand, they sell the DJIA at the closing price of  $t+1$  and purchase it again at  $t+2$ .

They employ a regression model to forecast the DJIA's trajectory from  $t+1$  to  $t+2$  in order to choose between the long and short options. They select the long option if the model indicates an upward trend; otherwise, they select the short option.

The performance of their adaptive strategy is compared to a baseline strategy, which entails purchasing and holding the DJIA for the duration of the study. It also compares this strategy to 3 other ones, namely:

1. Preis et al.: using debt, the best-performing search term found among the 98 semantically different keywords studied in1 to trade the DJIA.
2. Kristoufek (2013)20: using tickers of companies (e.g. XOM) as search terms to diversify portfolio among stocks in the DJIA.
3. Heiberger (2015)18: using company names (e.g. ExxonMobil) as search terms to trade individual stocks in the DJIA.

The objective of the comparison is to determine whether the adaptive strategy can outperform the straightforward buy-and-hold strategy in predicting market moves based on web search data. As well as compare the adaptive strategy with other alternatives that aim towards the same goal.

By merely following the market, the fundamental buy and hold strategy generates a total return of 61%. With a cumulative return of 96%, Kristoufek's technique outperforms the competition by constantly diversifying the portfolio based on web search data. However, it is vulnerable to downturns since it lacks a method to deal with bearish markets.

The approach used by Preis et al., which enters and leaves the market based on anticipated risk levels, produced a cumulative return of 327% over a given time frame. The Preis et al. technique generates a 95% total return in our longer-term experiment.

Heiberger's approach produces an 81% return at the end of the experiment during two specified times. Their adaptive method, however, greatly surpasses all benchmarks, generating a cumulative return of over 500%, which is 404% more than the benchmark strategy with the

**Takeaway:** We will be taking some inspiration from their evaluation process. Namely, the comparison of Viklund to a benchmark. Except, our benchmark will not be a buy and hold, it would be a technical analysis strategy without the inclusion of a machine learning analysis on

alternative data. For our 2nd iteration of Viklund (for those with an increased appetite for risk) we will be incorporating the use of google trends data to aid in Viklund's core decision making. This

## 4. Project Design

### 4.1 An Overview

**Project Idea:** Deep Learning on a public dataset to influence an algorithmic trading bot on position sizing

We aim to introduce retail investors into algorithmic trading powered by machine learning. But why would a retail investors require an AI trading bot like Viklund in the first place?

As we stated earlier, the majority of retail investors largely utilize a technical approach to the market. A technical approach is largely a manual process. It involves intense focus on:

- Studying price charts to gain insight into historical price movement
- Identifying support and resistance levels on price charts
- Utilizing indicators and oscillators to gain insight into market momentum, volatility, and potential turning points.

If you may be thinking this methodology is focus & time consuming for a part time/hobbyist investor...you would be right. In actuality, the average retail investors execution of this approach in the market ends up just shy of technical. This is due to emotional bias and market complexity, leading to impulsive decisions made on short-term market movements, rather than sound investment strategies. This is to be expected because after all, there is more to life than trading equities.

This is why an algorithm is needed. It doesn't get tired, it doesn't take a break, it cannot be distracted, and it can understand multiple dimensions of complexity, freeing the average day trader from the chains of their monitors. Allowing them to yield the results from trading equities, while giving them back their time.

As we discussed above, Viklund will utilize a fundamental approach to deciding when to buy/sell a stock. This means it will use a set of market indicators such as Exponential Moving Average (EMA) to signal whether or not to buy, short, or liquidate a position. We will then utilize a deep learning model to influence both its trading & position sizing. This model will analyze alternative data surrounding earning calls in order to signal to Viklund a position to take as well as how much capital to allocate to said position, based on its analysis. These signals will influence both Viklund's buying/selling power & direction.

## 4.2 Algorithm Frame Work

### 4.2.1 Modules

# The Algorithm Framework



**Figure (1) 4.2.1.1 -** The algorithmic framework work explaining the designated tasks each module will be responsible for.

We will modularize the tasks Viklund will be responsible for into 5 separate departments as shown in **Figure(1) 4.2.1.1**:

- **Universe Selection:** Selects the asset classes our strategy will focus on. The assets will be within the natural gas sector, specifically Shell P.L.C & BP P.L.C. These companies will be defined to the trading bot whom will proceed to identify the assets on the stock exchange and connect to the relevant brokers offering to sell & buy said equities.
- **Alpha Creation:** Universe selection passes the relevant securities to the Alpha model. The Alpha model is where our edge comes into play. This is where our Neural Network model will live. It assigns alpha scores to the securities that were passed on from the universe selection model. These alpha scores represent a directional assumption as well as a level of confidence in this prediction. In other words this is where a trade signal is generated. But note that this trade signal is not a trade order. The trade order will be handled by the next model.
- **Portfolio Construction:** This is where the trade signals from the alpha models are turned into a potential portfolio. In other words, this is where it is actually decided how much capital is allocated to each trade based on the trading signal
- **Execution Module:** The model responsible for sending out any trade orders is the execution model. This model decides how a new position will be opened

- **Risk Management:** Manages the risk. Sets risk parameters and is authorized to block any signals that will cause the execution model to send out trade orders that will violate the stated risk parameters. Closes a position as soon as it is down past a certain percentage.

## 4.2.2 Focus

As we have limited time on this project and are not financial advisors, it would not be wise to focus on all the modules as this will require extensive research and time wasted. It will be best to leave the potential implementation of some of the modules to left over development time. We will be focusing on:

- Alpha Creation Module(aka our edge)

More specifically, the marriage between our Neural Network and the execution of the trade. Although we will leave the implementation of the risk management module to left over development time, we will still implement risk management strategies within the alpha module where we can. The modules that will be implemented with left over development time are the following:

- Risk Management
- Universe Selection

### 4.2.2.1 Alpha

#### Viklund

Viklund has 2 main Machine Learning attributes the dwell within the Alpha section:

- **Viklun\_News\_Classifier:** Classification of news articles ('BUSINESS', 'POLITICS', 'SPORTS', 'ENTERTAINMENT')

This model will be important as it will allow us to categorize the kind of articles Viklund wishes to accept. We do not want Viklunds NN model sending sentiment signals that originate from articles referencing our selected companies involvement in aspects of life that do not have a foundational impact on its stocks underlying value.

- **Viklun\_NN\_Sentiment\_Analyzer:** Analyzes Financial articles and derive sentimental impact to stock price (Negative, Neutral, Positive,)

This model will be used to derive sentiment surrounding news related to earnings report on our chosen company. This model will deliver signals which will then be converted into valid trading signals depending on certain risk parameters.

As the application for our neural network only comes into effect once we enter a specific period, we require an additional strategy that will allow us to trade equities for the remaining of our trading term. As stated above, this strategy must be founded on methodologies familiar to our retail traders. Our chosen strategy will be:

## **Exponential Moving Average Cross**

This strategy is a technical trading approach to the market. It utilizes the fundamental building blocks for a desired trend following outcome (highly familiar with retail investors) through the use of two exponential moving average indicators of a provided security. The EMA indicators give more weight to recent price data, making the indicators more responsive to recent price changes, compared to that of the simple moving average (SMA) indicator. This is particularly very useful to us as we can utilize this component to warn us of the imminent down turn of a stock if used right.

### **(i) Building Blocks of the EMA Cross Strategy**

- **Slow EMA Indicator:**

The purpose of this EMA is to track the moving average change over a longer period of time. It reacts more slowly to price changes and helps smoothen/ filter out noise in the price data. We will be tracking this moving average price data over a 30 day rolling window.

- **(i) Fast EMA Indicator:**

The purpose of the second EMA is to track the moving average change over a shorter period of time. We will use this to represent a short term trend tracker as it will react more quickly to price changes and capture short term momentum behaviour in the price data.

### **(ii) Bringing It All Together**

- **Buy Signal:** When the Fast EMA indicator crosses above the slow EMA indicator. We will use this moving average behaviour to represent a bullish signal. This bullish signal will be processed by the Portfolio Construction component of our algorithm as a buy signal, which will then be converted into such via our Risk Management strategy & Execution Component.

- **Sell Signal:** When the Slow EMA indicator crosses above the Fast EMA indicator. We will use this moving average behaviour to represent a bearish signal. This bearish signal will be processed by the Portfolio Construction component of our algorithm as a sell/short/liquidate signal, which will then be converted into such via our Risk Management strategy & Execution Component.

## 4.2.3 Evaluation

### Deep Learning Model

We will evaluate the Deep Learning Model by utilizing cross fold validation, optimizing for accuracy and F1 score. Accuracy will be determined by the proportion of correctly classified sentiments, while the F1 score will be determined by the harmonic mean of precision and recall, providing a balanced measure for imbalanced classes.

### Trading Algorithm

We will evaluate the results of the total trading performance of Viklund. We will do this by not only evaluating its percentage returns but also considering its **Probabilistic Sharpe Ratio (PSR)**. This is a widely used measure of risk-adjusted return in finance. The **Sharpe Ratio (SR)** compares the excess return of an investment (above the risk free rate) to its volatility, indicating how much return an investor receives per unit of risk. We will be watching for an increase to the benchmark PSR & SR (the psr & sr value without the use of our deep learning model) after our Deep Learning Model has been applied to the position sizing task. We will also consider **Drawdown** as this can represent an overview of a strategy's risk. A drawdown tracks one's individual trading performance or analyzes the historical risk of other investments. It is typically expressed as a percentage of the difference between the peak and the following trough. A trading account had a 10% drawdown if it had 10,000 dollars in funds and those funds dropped to 9,000 dollars before rising back to 10,000 dollars. These three main metrics will be evaluated against the technical analysis strategy without the use of our machine learning alternative data analysis.

### Stock Picks

We will apply this algorithm and evaluation metrics to equities that are highly sensitive to earnings per share news. If we select stocks that are not sensitive to their quarterly earnings calls, our NN will be virtually useless. Generally, Technology stocks are not highly sensitive to EPS releases due to the future/appreciating nature attributed to technology products, specifically, the importance placed on scalability, rather than current day realized profits. Market reactions to earnings reports are also stronger at companies with bigger market capitalization. This is largely attributed to the sheer size of investment held by institutional investors as they react to surprises. This reaction can dictate the short term direction of any stock. Research shows that stocks with a market cap of at least USD4 billion dollars have a high correlation of 54% with earnings announcements, while stocks with USD3 billion dollars and below have a correlation of only 23 percent [11]. Our target industry will be Oil & Gas. Further evidence on the effect of quarterly earnings calls on price action in the oil & gas industry will be given later on in this report. Our top selections for Oil & Gas companies will be Shell p.l.c(SHEL) & BP p.l.c(BP)

## **4.2.4 Plan**

- Week 2: Initial Setup, Planning, and Research
- Week 4: Test out variations of initial News Classifier Model
- Week 6-8: Test out variations initial Sentiment Analyzer Model

### **Milestone: Working Prototype of at least 1 Feature Completed (Week 10)**

- Deliver a working prototype of sentiment analyzer.
- Demonstrate the potential for it to generate buy/sell signals for a later implemented EMA crossovers trading strategy.
- Document initial findings and prepare for further development.

### **Weeks 11-22: Enhancements and Final Integration**

#### **Week 11-12: Enhance News Classifier and Research**

- Improve accuracy and robustness of the news classifier.
- Implement additional preprocessing steps based on feedback.
- Conduct research on advanced classification techniques.
- Design and implement slow EMA indicator over a 30-day rolling window.
- Design and implement fast EMA indicator for short-term trend tracking.
- Integrate EMA indicators to generate buy/sell signals.
- Test EMA Cross Strategy with historical stock data.
- Conduct research on EMA strategies and their performance.

#### **Week 13-14: Enhance Sentiment Analyzer and Research**

- Improve accuracy and robustness of the sentiment analyzer.
- Implement additional preprocessing steps based on feedback.
- Conduct research on advanced sentiment analysis techniques.

#### **Week 15-16: Advanced Integration, Testing, and Research**

- Fully integrate enhanced news classifier and sentiment analyzer.
- Perform extensive testing with various datasets.
- Optimize model performance and accuracy.
- Conduct research on system optimization and integration best practices.

#### **Week 17-18: Develop Portfolio Construction Component and Research**

- Design and implement portfolio construction based on buy/sell signals.
- Integrate portfolio construction with the risk management strategy.
- Conduct research on portfolio construction methodologies and risk management.

#### **Week 19-20: Develop Risk Management and Execution Component and Research**

- Design and implement risk management strategies.
- Integrate execution component for automated trading.
- Conduct research on risk management strategies and automated trading systems.

### **Week 21: Final Testing, Validation, and Research**

- Perform end-to-end testing of the entire system.
- Validate system performance with real-world data.
- Make necessary adjustments based on test results.
- Conduct research on final system optimization and validation techniques.

### **Week 22: Final Review, Documentation, and Deployment**

- Conduct a final review and refine the system.
- Prepare comprehensive documentation and user guides.
- Deploy the system for live trading.
- Conduct post-deployment research on system performance.

## **5. Implementation**

### **5.1 Viklund News Classifier**

As this isn't the main neural network, we will try not to focus unnecessarily on its implementation to conserve space.

#### **5.1.1 Libraries**

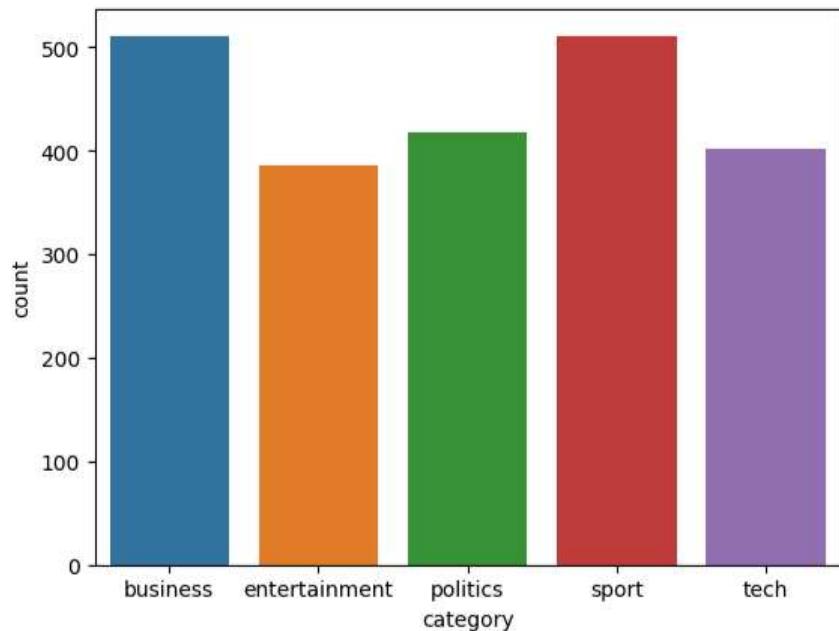
Viklund news classifier was built using the panadas and sklearn library for the purpose of preprocessing the text data, as well as constructing the chosen Machine Learning Model.

#### **5.1.1 Dataset & Preprocessing**

##### **5.1.1.1 Dataset**

We utilized the BBC news corpus used in the study "**Practical Solutions to the Problem of Diagonal Dominance in Kernel Document Clustering**"[10] as our dataset which consists of 2225 documents from the BBC news website corresponding to stories in five topical areas from 2004-2005. Class Labels: 5 (business, entertainment, politics, sport, tech).

```
<Axes: xlabel='category', ylabel='count'>
```



**Figure (1) 5.1.1.1** - A histogram showing the distribution of data samples across their respective classes (business, entertainment, politics category, sport, tech).

As shown above in *Figure (1) 5.1.1.1*, entertainment has the least amount of samples representing its classification category.

### 5.1.1.2 Pre-processing

Text data is incredibly content-rich but also quite unstructured, necessitating considerable preparation before an ML system can extract useful information from it. Converting text into a numerical format without losing its meaning is a difficult task. We utilize the following for our pre-processing pipeline:

**Lower() -> Mentions -> Links -> Punctuation & numbers -> Single Character -> Lemmatize -> Stopwords**

We begin by bringing all texts into a lowercase format. We then proceed by removing all mentions '@BBC' as well as any links that may be found within the articles. We address all punctuations and numbers by removing them as they hold no meaning to our model for our objective. All single characters are removed as well as any multiple spaces within sentences. These processes make the final text normalization process easier to undergo. Our objective here is to convert text into standard or canonical form, making it easier to analyze, compare, and contrast.

For our final normalization process we will select lemmatization which is the process of reducing words to their base or dictionary form. It considers the context and parts of speech before the reduction to its lemma format, ensuring that the resulting word is actually valid. This is

something that stemming does not consider, often rendering the resulting word an invalid dictionary term. We then remove the stopwords ('a', 'the', 'man') from the resulting corpus as they also have no meaning when carrying out document categorization.

### 5.1.1.3 Vectorization

Quantifying the text data into numerical format is vital for Viklunds understanding of the category to which a news article belongs. To transform text to its numerical format we will need to vectorize it.

**Count Vectorizer** allows the text input to easily be transformed into a numerical format that machine learning algorithms can understand using vectorizer. It is helpful for activities like text analysis, document classification, and clustering. It is less effective in capturing the semantic meaning of text because it doesn't take into account the significance of words or their context.

When compared to straightforward word count representations (like CountVectorizer), the **TFIDF** method takes into account the relevance of terms in the documents based on their TF-IDF values, which frequently results in enhanced performance. Practically speaking, this approach incorporates both the local and global relevance of words in texts, making it suitable for our particular text analysis task...document categorization. The TF-IDF scores of words that are common in a given text but uncommon elsewhere in the corpus tend to be higher, and these words are frequently regarded as key terms or features for that document.

### 5.1.2 Machine Learning Algorithm

We will be utilizing the Naive Bayes algorithm based on the Bayes theorem, which is a cornerstone of probability theory. We are specifically interested in its extension, Multinomial Naive Bayes model. The Bayes theorem enables us to update our beliefs (probability) about an event whenever new information becomes available by connecting conditional probabilities. In the context of document categorization, the presence or absence of words or terms in a document acts as the evidence, whereas the class label serves as the event. It is well-suited for problems involving discrete data, such as text data where the features are counts of occurrences.

$$P(C_i|D) = \frac{P(C_i) \cdot P(D|C_i)}{P(D)}$$

Where:

- ( $C_i$ ) is the class label.
- ( $D$ ) is the document (feature vector).
- ( $P(C_i|D)$ ) is the posterior probability.
- ( $P(C_i)$ ) is the prior probability.
- ( $P(D|C_i)$ ) is the likelihood.
- ( $P(D)$ ) is the evidence.

**P(D|C\_i)**: is the likelihood of observing the feature vector D given a class. This likelihood is derived from the multinomial distribution of word occurrences.

$$P(D|C_i) = \prod_{j=1}^n P(w_j|C_i)^{x_j}$$

Where:

- ( $P(D|C_i)$ ) is the likelihood of observing the feature vector (D) given class ( $C_i$ ).
- ( $w_j$ ) is the (j)-th word (term) in the vocabulary.
- ( $P(w_j|C_i)$ ) is the probability of observing word ( $w_j$ ) in documents of class ( $C_i$ ).
- ( $x_j$ ) is the count of word ( $w_j$ ) in the feature vector (D). \$\$

The goal is to find ( $C_i$ ) that maximizes the posterior probability

$$P(C_i|D)$$

To make a classification driven decision, the model will need to compute **P(C\_i|D)** for each class  $C_i$  at which it will select the class with the highest posterior probability as the predicted class label for the document D

### 5.1.3 Training & Testing

I carried out training (split=0.2) on both the count vectorized features as well as the TFIDF vectorized features and yielded slightly higher accuracy with our TFIDF method.

#### Count Vectorizer (Test data: 98.43%)

```
Index(['business', 'entertainment', 'politics', 'sport', 'tech'], dtype='object')
```

	0	1	2	3	4
0	98	0	2	0	2
1	0	76	0	0	1
2	0	0	83	0	1
3	0	0	1	101	0
4	0	0	0	0	80

**Figure (1) 5.1.3** - Confusion matrix representation on the results from a 80/20 split on training/test data for the count vectorizer.

#### TFIDF Vectorizer (Test data: 98.65%) +0.22%

```
Index(['business', 'entertainment', 'politics', 'sport', 'tech'], dtype='object')
```

	0	1	2	3	4
0	100	0	1	0	1
1	1	75	1	0	0
2	0	0	83	0	1
3	0	0	0	102	0
4	1	0	0	0	79

A TFIDF vectorized MNB model was ran on the following news articles related to our chosen public company and its industry Shel Plc (SHEL: NSDAQ). Shown by the results, we have correctly built a document categorizer to help filter out non-business related news articles, getting us closer to only providing our Neural Network Finance/ business related news articles. There are some limitations with this model, namely it can struggle to separate politics from business as they often go hand in hand. However, this model is strong when separating other categories from each other.

**Test sample:** <https://www.cbc.ca/news/canada/calgary/capping-emissions-alberta-oil-gas-ottawa-1.6237432> (<https://www.cbc.ca/news/canada/calgary/capping-emissions-alberta-oil-gas-ottawa-1.6237432>)

**Result: 100% Accuracy: Class 2 = Politics**

**Test sample:** <https://www.bloomberg.com/news/articles/2023-05-04/shell-maintains-pace-of-share-buybacks-as-profit-beats-estimates> (<https://www.bloomberg.com/news/articles/2023-05-04/shell-maintains-pace-of-share-buybacks-as-profit-beats-estimates>)

**Result: 100% Accuracy: Class 0 = Business**

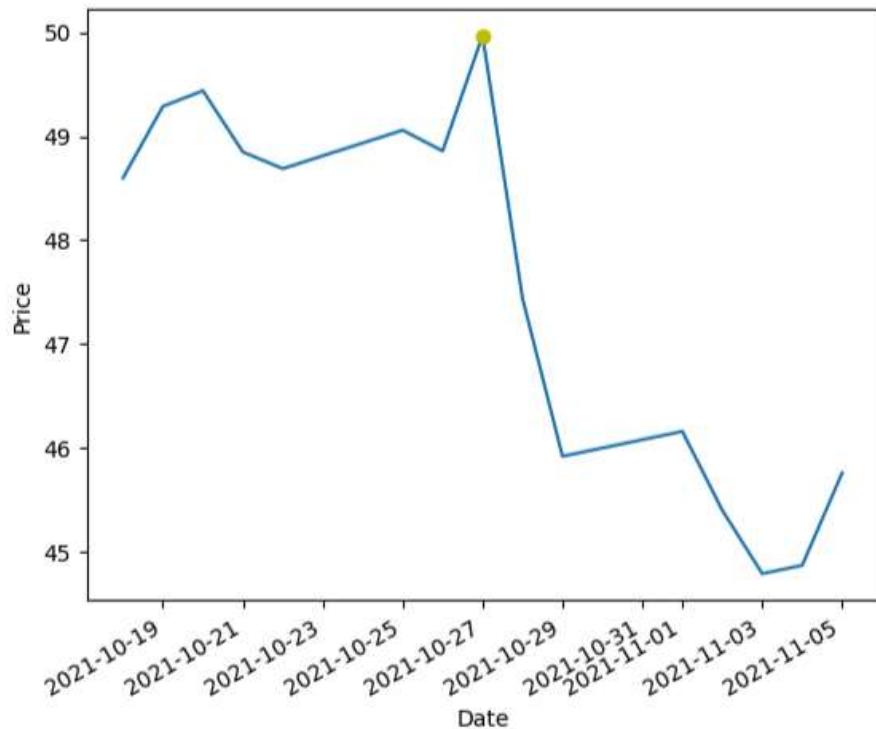
## 5.2 Viklund Neural Network Sentiment Analyzer

We will be using Viklunds Neural Network to predict the sentiment of news articles reporting on companies. As discussed, we will focus on articles covering said companies financial reports, specifically, earnings per share (EPS). In absence of market expectations, it has been proven via Henry Ong's [11] analysis that there is little correlation between stock prices and a securities earnings growth results. Based on the analysis of the resulting statistical data between the historical three day returns of the 30 stocks in the PSE index and their resulting first quarter earnings announcement, there was only a 13% correlation. With the addition of market expectations the correlation improved significantly to 41%.

As an example, say analysts on Wall Street are said to be expecting a companies earnings per share to be valued at 2.00 dollars. However, the company announces later that morning on their earnings call that their earnings per share value is actually 1.00 dollar per share. This is likely to trigger price discovery which re-evaluates the underlying value of the companies security. In the

absence of any favouring geopolitical events, price discovery can give birth to sell offs, driving

Out[187]: [`<matplotlib.lines.Line2D at 0x261c5335400>`]



**Figure (1) 5.2** - Oct 27, 2021 Royal Dutch Shell Plc Stock(SHEL) - Share price drop after earnings call (yellow dot) reviewed they fell short of the markets estimated earnings per share by 24% (estimated earnings per share 1.40, vs actual earnings per share 1.06)

For this reason, a neural network capable of detecting a negative, neutral, or positive sentiment on a document describing our companies quarterly EPS is critical to the success of our algorithmic trading bot.

Our neural network construction flow will as shown below:

**Raw Text(dataset) -> Pre-processing -> Train/Test Split -> Prep Embedding Layer & Tokenization -> Word Embedding -> Model Construction -> Train Model -> Test Model**

### 5.2.1 Libraries

We utilized over 40 different modules which will be sited depending on their relevance to the project

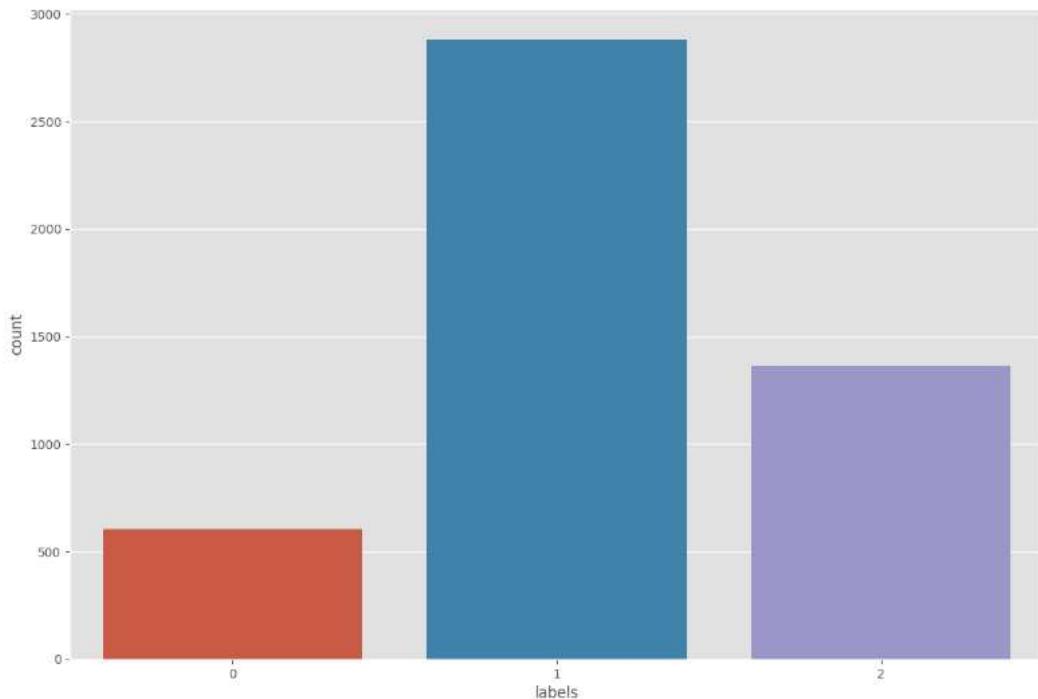
### 5.2.2 Dataset

Finding a sentiment labeled dataset of full articles related to financial subjects proved to be very difficult. I was unable to obtain this, so I began seeking a dataset of classified sentences related to financial subject matters alone. I discovered Financial Phrase Bank [23], a dataset of

sentences related to financial subjects that have been annotated into 3 classes, Negative, Neutral, and Positive.

The data used includes English language articles about all companies listed on the OMX Helsinki stock exchange. It was curated by Frankie Robertson and hosted on Hugging face. An automated web scraper was used to pull news from the LexisNexis database. To ensure complete coverage of small and large businesses, organizations across a variety of industries, and news sources, a subset of 10,000 articles were randomly selected from this news database. Following the method of Maks and Vossen (2010), they eliminated all sentences that did not contain lexical elements. As a result, the total sample was reduced to 53,400 sentences, each containing at least one or more lexical entities. The sentences were then classified based on the different types of entity strings found. Subsequently a random sample of 5000 sentences were chosen and annotated by 16 financial experts with the intent of identifying the sentences that would either have an effect on the stock market (Positive, Negative), as well as the sentences that had no effect, Neutral. We have selected the dataset in which 50% of all annotators were required to agree on a given class, as this dataset had the largest class representation.

```
Out[292]: <Axes: xlabel='labels', ylabel='count'>
```

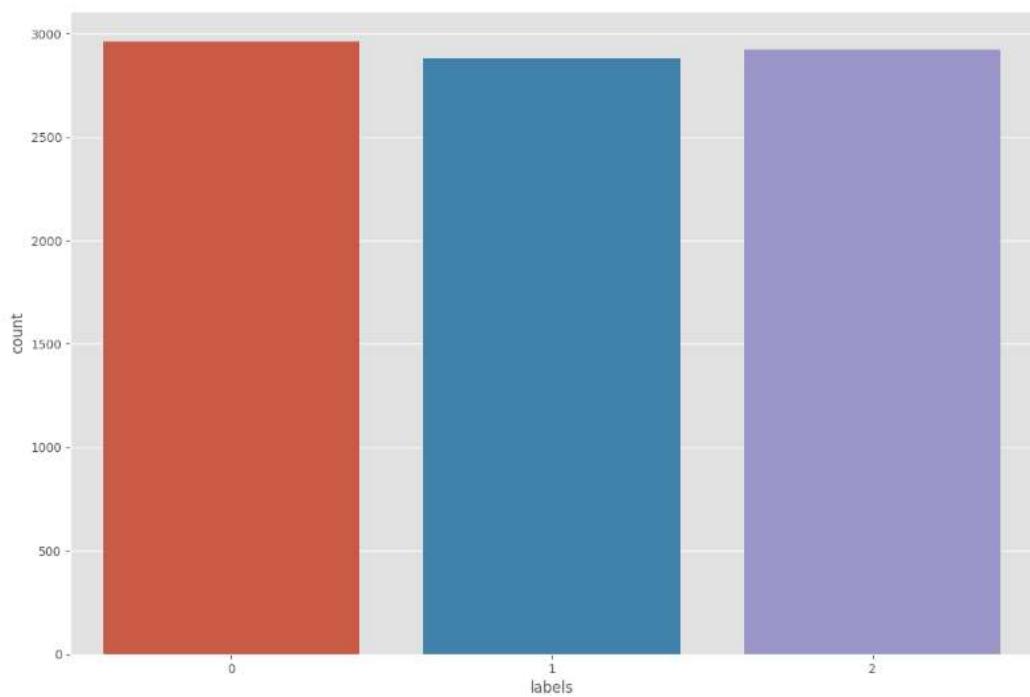


**Figure (1) 5.2.2 - Financial Phrase Bank, an imbalanced dataset of sentences related to financial subjects that have been annotated into 3 classes, Negative, Neutral, and Positive (0,1,2 respectively). In this subsection of the dataset, 50% of all annotators were required to agree on a given class.**

Training our model on the now foundational components of our final dataset did not yield the optimal results, despite utilizing class weights in an attempt to balance the lack of representation within the minority classifications (negative:0, neutral:1, positive:2). This is why we had to combine other datasets in order to obtain a more balanced representation of the three classifications.

```
In [676]: 1 sns.countplot(x='labels', data=fin_news_optimized_6)
2
```

```
Out[676]: <Axes: xlabel='labels', ylabel='count'>
```



**Figure (2) 5.2.2** - Custom dataset based on the Financial Phrase Bank, merged with 6 other datasets created to achieve an optimal categorical representation (negative:0, neutral:1, positive:2) of 8,761 samples

We were able to achieve the categorical representation shown above by adding data samples from 6 other datasets that focused on classification of news headlines and by extension, financial sentences [12,13,14,15,21,22]. We also employed some positive reinforcement learning techniques by curating & adding data samples that represent difficult sentence structures containing terms the previous models failed to capture contextual meanings around due to the lack of representation in the original training dataset [24].

### 5.2.3 Pre-processing

The text preprocessing function is a crucial process when it comes to sanitizing the training data we are going to use. All the text data is converted to lowercase to ensure consistency. References and URLs are removed from the text, along with any punctuation and numerals. The result would be text data composed of solely alphabetic characters. Isolated letters are cleaned up by removing single characters as they usually hold no meaningful information to a machine, and multiple spaces are converted to single spaces. After the text has been tokenized into individual words, POS (part-of-speech) tags are applied to each word to represent its grammatical function. Lemmatization is applied to these tags, which reduce words to their most basic forms (e.g., "running" to "run"). After completing these procedures, the text is cleaned and normalized and is prepared for additional NLP analysis.

We attempted to utilize lemmatization with dependency parsing (Sapcy) and concept value pairing within our preprocessing step. However, after training the models on the processed data, the results were significantly lower than other trials that did not include the use of dependency parsing and concept value pairing. The objective was to predefine a set of syntactic relationships, left & right dependent relationships to be specific. To start, one would identify the nsubj in order to extract the subject and the governing terms from the sentence. By establishing these two terms, we are able to recursively reconstruct the sentences to include sentiment rich verbage.

For example, if the following dependency sets were defined as:

- *right\_dependents\_set = {"prep", "advmod", "dobj", "advcl", "compound"}*
- *left\_dependents\_set = {"amod", "compound", "advmod"}*
- *predefined\_start\_pos = {"NOUN", "PROPN", "ADJ"}*
- *Test sentence = "Shell increased its quarterly dividend by 15% to 0.33 dollars per share, as previously communicated in mid-June"*

The concept value pairing algorithm identifies the subject as "Shell" and the governor term as "increased". It does this by identifying the nsubj relationship within the sentence, seeking a relationship between a Noun/Proper Noun and an Adjective. Once those two variables are set, the algo will then begin identifying relationships found in either the right dependency set or the left dependency set between the governor term and other terms in the sentence. In the event any of the terms that held a left or right dependency with the governor term are Nouns, Proper Nouns, or Adjectives, we will then re-run the search for right/left dependencies on that term, using our pre-defined left/right dependency sets. This takes a recursive nature until there are no more relationships to explore. This results in the reconstruction of noun or subject terms and their adjectives, or descriptive terminologies. Our test sentence is converted into the below.

- **[['Shell', 'increased', [['quarterly'], 'dividend'], 'by', 'to', 'communicated']]**

This can be flattened out into "**Shell increased quarterly dividend by to communicated**". However, this pre-processing step did not yield ideal results when used as training data on all the models we tested (will be expanded upon moving forward). We also attempted to remove stop words and this too did not yield optimal results. Therefore, we removed the un-productive pre-processing steps and proceeded with bringing all text to lowercase format, removing all mentions, links, punctuations, single characters, and multiple spaces. This way we achieved an optimal dataset for which training can begin

## 5.2.4 Word Embedding

When speaking on sentiment analysis used on financial text, the VP of AI research Oana Frunza at Morgan Stanley stated:

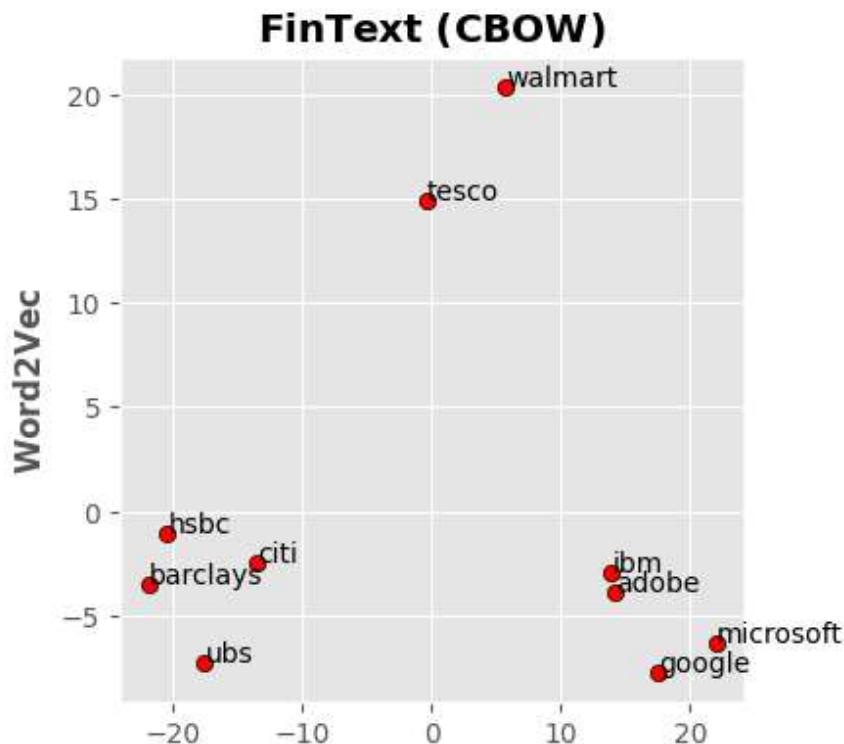
*"While I have looked at the shelf tools...I think it's well known that all these tools, they don't scale up so well for financial text, they come out as **neutral** and on top of that its very hard to fine meaningful sentiment on very short text"*

What she is describing is the ineffectiveness that current sentiment analysis shelf tools have on domain specific usage. These are more often than not the tools currently available to retail investors. They simply are not effective.

High Jargon within our chosen problem space begs the need for our model to make decisions from a domain specific-specific knowledge base. We can think of our knowledge base as our Word Embedding, a dictionary/vocabulary for a neural network. The standard word2vec models utilize default vectors that lower the models predictive power as they are unable to assign vectors to words not used very often. As new goods or technologies are developed, industry-specific language found in documents or every day usage evolves over time. Furthermore, Glove vectors trained on Wikipedia articles fall short in capturing the nuanced idioms seen in the terminology used in company financial reports.

As we do not have enough time to train our own domain specific word embedding, we opted to use a domain specific word embedding, called FinText [16]. It uses the Dow Jones Newswires Text News Feed database from 1 January 2000 to 14 September 2015. While specialising in corporate news, this big textual dataset is among the best textual databases covering finance, business, and political news services worldwide. The diagram below shows its power in word representation as well as artificial context. We will attempt to plot the names of companies operating in different industries

- `token_list1 = ['microsoft', 'barclays', 'ibm', 'google', 'adobe', 'citi', 'ubs', 'hsbc', 'tesco', 'walmart']`



**Figure (1) 5.2.4** - 2 dimensional spatial reference to represent the embeddings understanding of the relationship between domain specific financial terms. This figure showcases its understanding between company names and their lines of business by clustering them together

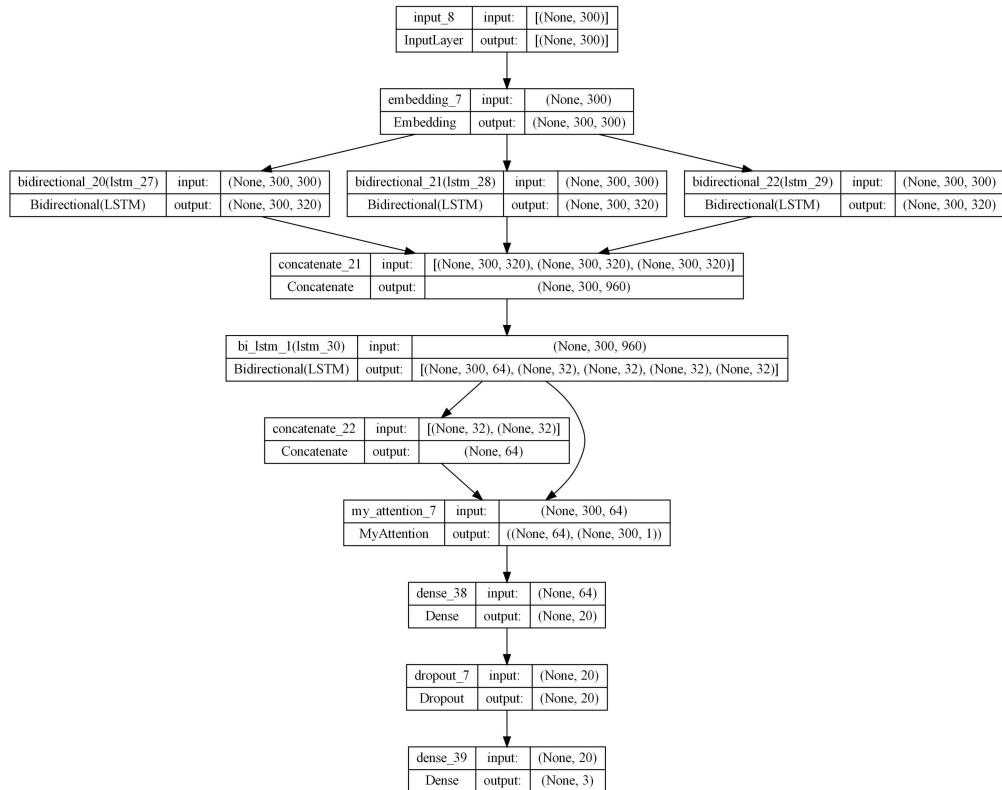
*respectively.*

We do not just identify the vector representation of these names as one giant cluster due to their unified attribute of being business establishment. As figure (1) [**Figure (1) 5.2.4**] shows, FinText accurately factors in the industry in which these companies operate within. Financial establishments such as HSBC, Citi, Barclays etc. are grouped together, Technology establishments such as IBM, Microsoft, Google etc. are grouped together, and Food related establishment Walmart and Tesco are grouped together. This visually displays our chosen word embeddings intelligence on what would be out of dictionary nouns and domain specific terminology.

### **5.2.5 Model Architecture**

Our neural network is engineered to analyze financial articles covering a company's earnings per share performance, specifically whether they surpassed it or fell short. We will use this as trading signals to our algorithmic trading bot which will go long on a stock when the company surpasses its estimated earnings per share, and shorts the stock when the company falls short of its estimated earnings per share. The whole concept of synthesizing public information is to act on it within the market, this model should be able to automate that process, speed is needed within the model in order to realize the effectiveness of a model's signals in a live trading market where price movement occurs by the second. Due to the high jargon used in this industry meaning can be conveyed in shorter sentences. This is why our neural network model architecture needs to be able to capture sentiment surrounding both short and long sentences.

We tested over 50 types of model architecture, the most optimal ones are shown in **table..**, however, the model architecture that showed the best results was the following:



**Figure (1) 5.2.5** - This diagram illustrates the architecture of the sentiment analysis model, featuring an embedding layer followed by three Bidirectional LSTM layers, whose outputs are concatenated. An attention mechanism is applied before passing through dense and dropout layers, with the final output layer classifying sentiment into three categories.

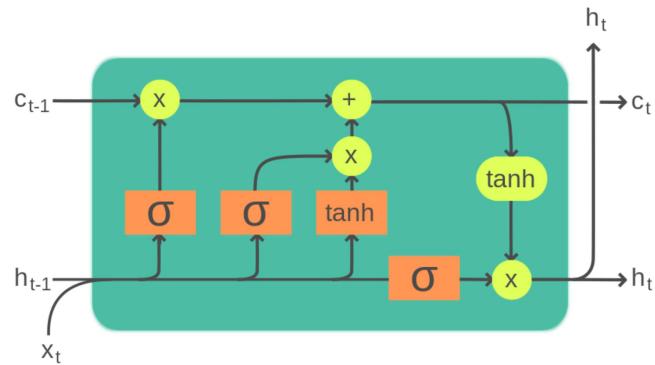
1

### 5.2.5.1 Embedding Layer

The model begins with an Embedding Layer that creates dense word embeddings from the incoming text data. The word embeddings for financial terminology and jargon are initialized using domain-specific word embeddings (FinText). The embeddings are fixed during model training since they are already pre-trained (trainable=False).

### 5.2.5.2 Long Short Term Memory (LSTM) Layer

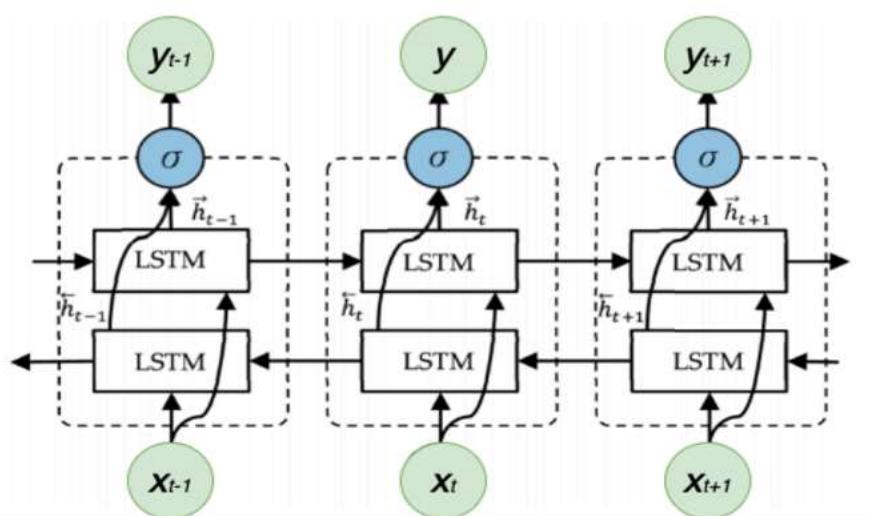
- **Intuition:** ability to recognize and recall patterns through time series data. (Very promising)
- **Basic Usage Structure:** Input -> Embedding layer -> LSTM -> Dense layer -> Output
- **Speciality:** Memory and self-learning.
- **Data Type:** Trained with sequence data
- **Applications:** Time series analysis, Machine Translation, Language Modeling, and Multilingual Language Processing



Legend:

Layer	Componentwise	Copy	Concatenate

**Figure (1) 5.2.5.2 - A Long Short-Term Memory is a type of recurrent neural network structured to address the vanishing gradient problem. Depicted is the inner workings of an LSTM cell capable of processing data sequentially.**



**Figure (2) 5.2.5.2 - Bidirectional LSTM Layers showing the 3 forward LSTM units responsible for processing the sequential data (sentence) from left to right, while 3 additional backward LSTM units below it are responsible for processing the sequential data from right to left.**

The word embeddings are processed by the model using a number of Bidirectional Long Short-Term Memory (Bi-LSTM) layers [Figure (2) 5.2.5.2]. In order to fully comprehend the sentiment in financial articles, bidirectional LSTMs are preferred over generic LSTM [Figure (1) 5.2.5.2] algorithms as they are more capable of capturing contextual information from both forward and backward directions in the text. Long-range relationships in the text are easier for the model to train thanks to the greater gradient flow provided by bidirectional LSTMs, this by extension aids in mitigating the vanishing gradient problem, especially when working with recurrent neural

networks. In the context of sentiment analysis, this means the model takes into account both the words that come before as well as the words that come after a specific word. Our models ability

---

### ***Bidirectional LSTM Layer construction:***

Refer to **Figure (2) 5.2.5.2** for a visualization of the forward and backward pass functioning together

*Forward Pass:*

- The forward pass takes the input sequence and processes it from left to right (forward direction).
- For each time step in the sequence, the LSTM units in the forward pass determine the hidden states and cell states.

*Backward Pass:*

- The input sequence is processed from right to left (backward direction) by the backward pass.
- In the backward pass, the LSTM units generate hidden states and cell states for every time step in reverse order.

*Combining Forward and Backward Passes:*

- Concatenating the results from the forward and backward passes results in a full representation of the sequence that includes context from both the forward and backward passes.

The hidden state and cell state calculations for a single Bidirectional LSTM layer can be expressed as the following:

#### ***Forward LSTM:***

- $(h_{t^f})$  and  $(c_{t^f})$  are the forward hidden state and cell state at time step  $(t)$ , respectively.
- $(x_t)$  is the input at time step  $(t)$ .
- $(W^f)$  and  $(U^f)$  are weight matrices for the forward LSTM.
- $(b^f)$  is the bias vector for the forward LSTM.

$$\begin{aligned} i_t^f &= \sigma(W_i^f \cdot x_t + U_i^f \cdot h_{t-1}^f + b_i^f) \\ f_t^f &= \sigma(W_f^f \cdot x_t + U_f^f \cdot h_{t-1}^f + b_f^f) \\ o_t^f &= \sigma(W_o^f \cdot x_t + U_o^f \cdot h_{t-1}^f + b_o^f) \\ \tilde{c}_t^f &= \tanh(W_c^f \cdot x_t + U_c^f \cdot h_{t-1}^f + b_c^f) \\ c_t^f &= f_t^f \odot c_{t-1}^f + i_t^f \odot \tilde{c}_t^f \\ h_t^f &= o_t^f \odot \tanh(c_t^f) \end{aligned}$$

#### ***Backward LSTM:***

- $(h_{t^b})$  and  $(c_{t^b})$  are the backward hidden state and cell state at time step  $(t)$ , respectively.
- $(x_t)$  is the input at time step  $(t)$  (in reverse order).
- $(W^b)$  and  $(U^b)$  are weight matrices for the backward LSTM.
- $(b^b)$  is the bias vector for the backward LSTM.

$$\begin{aligned}
 i_t^b &= \sigma(W_i^b \cdot x_t + U_i^b \cdot h_{t+1}^b + b_i^b) \\
 f_t^b &= \sigma(W_f^b \cdot x_t + U_f^b \cdot h_{t+1}^b + b_f^b) \\
 o_t^b &= \sigma(W_o^b \cdot x_t + U_o^b \cdot h_{t+1}^b + b_o^b) \\
 \tilde{c}_t^b &= \tanh(W_c^b \cdot x_t + U_c^b \cdot h_{t+1}^b + b_c^b) \\
 c_t^b &= f_t^b \odot c_{t+1}^b + i_t^b \odot \tilde{c}_t^b \\
 h_t^b &= o_t^b \odot \tanh(c_t^b)
 \end{aligned}$$

*Combining Forward and Backward Passes:*

- The outputs of the forward and backward passes at each time step are concatenated to create the final output sequence.

$$h_t = [h_t^f; h_t^b]$$

Viklund's architecture is equipped with multiple Bidirectional LSTM layers stacked on top of each other, constructing a deeper network powerful enough to capture increasingly complex patterns and dependencies found within the input sequence. After Hyperparameter tuning we identified that 3 bidirectional layers was the optimal number of layer to include within Viklunds final model architecture.

Viklunds bidirectional nature enables it to effectively represent word dependencies amongst each other. It may comprehend, for example, that the words appearing later on in a sentence or even earlier in the sentence may have material impact on the sentiment expressed in a certain area of the sentence.

### 5.2.5.3 Concatenation Operation

After the processing of the sequential representation of text data by the Bidirection LSTM layers, the hidden states as well as the cell states are concatenated in order to construct a comprehensive sequence representation. As a result of doing this we combine the information captured by the BiDi LSTM from both directions, enhancing the quality of the represented text data.

### 5.2.5.4 Custom Attention Mechanism Layer

---

*Attention Layer construction:*

---

The relevance of each time step in the sequence is calculated using an attention mechanism applied to the concatenated Bidirectional LSTM outputs. When identifying sentiment, attention enables the model to concentrate on pertinent sections of the text. The attention weights show the relative importance of each word in the sequence.

The attention layer is initialized with a number of units that determine the dimensionality of its weights. Trainable Dense layers are defined in the layer and given the responsibility of learning the appropriate weights during training.

The attention technique is used in the call method of the class. `tensorflow.expand_dims` is used to expand the hidden state along a new axis (the time axis). To execute element-wise operations between features and hidden, this step is required.

An activation function called hyperbolic tangent ( $\tanh$ ) is used to determine the attention scores. The ratings indicate how important or relevant each attribute is in relation to the concealed state.

$$\text{score} = \tanh(W_1 \cdot \text{features} + W_2 \cdot (\text{hidden} \otimes \mathbf{1}_{\text{sequence\_length}}))$$

To acquire attention weights along axis 1, the scores are routed via a Dense layer with a softmax activation function applied to the scaled attention scores in the previous step. These weights define how much emphasis should be placed on each feature's component.

$$\text{attention\_weights} = \text{softmax}(V \cdot \text{score}, \text{axis} = 1)$$

The `attention_weights` and features are combined to create the context vector, which is calculated as their element wise product. This vector displays the features' weighted total, where each component is scaled according to how much attention it deserves. By adding along axis 1, the final context vector is produced.

$$\text{context\_vector} = \sum_i (\text{attention\_weights}_i \cdot \text{features}_i)$$

- Above we calculate the context vector by deriving the element-wise product of attention weights and features, and then we take the sum along the sequence dimension.

#### 5.2.5.5 Dense Layers

A Dense layer with a ReLU activation function receives the context vector produced by the attention mechanism. To reduce overfitting, a dropout layer with a dropout rate of 0.3 is introduced. The value of the dropout rate was fine tuned. The final Dense layer produces a probability distribution across the three sentiment classes—negative, neutral, and positive—with a softmax activation function. The reason for the use of a softmax activation function on the final dense layer is due to our desire to introduce artificial certainty to the model across all three of the classifications it is being trained on. With the introduction of artificial certainty Viklunds risk management module within its algorithmic trading component can filter out low certainty sentiment predictions and rely on strong certainties to execute trades upon.

### 5.3.6 Model Architecture Summary

This model was optimized using ADAM, specifically focusing on categorical cross entropy as a loss function due to the multi-classification nature of the problem we wish to solve. The model holds strong contextual understanding through the use of our domain specific word embedding as well as the use of our Bidirectional LSTM fused with our attention mechanism, enabling Viklund to obtain an **83% in accuracy, 95% in AUC**, as well as an **83% and 84% in f1 & recall** respectively, on the unseen test dataset. These modules allow the model to capture the contextual information provided within sentences related to financial subject matters rather effectively (Further discussed in evaluations). By processing the textual representations sequentially, the model factors in the order of words and sentences. The attention layer then provides further insights into which parts of the sentence contribute a larger meaning to the overall sentence sentiment. This also helps with the black box problem that unfamiliar parties have with Neural network, because the inclusion of the attention layer makes the model's decisions more interpretable.

## 5.4 Viklund Algorithmic Trading Bot

### 5.4.1 Exponential Moving Average Cross

The Exponential Moving Average Cross (EMA) is a technical analysis strategy employed in the financial markets during trading. Our implementation of this involves two EMAs with various time periods. Buy and sell signals are generated depending on the crossover of these EMAs.

1. Calculate the Fast EMA (EMA\_Fast):

$$EMA\_Fast_t = \alpha \cdot Close_t + (1 - \alpha) \cdot EMA\_Fast_{t-1}$$

where:

- ( $N$ ) is the number of periods for the short-term EMA which is **5 days** for Viklund.
- ( $EMA\_Fast_t$ ) represents the short-term EMA at time (t).
- ( $Close_t$ ) is the closing price of the asset at time (t).
- ( $\alpha = \frac{2}{N+1}$ ) is the smoothing factor for the short-term EMA, which is **0.3** for Viklund .

2. Calculate the Slow EMA (EMA\_Slow):

$$EMA\_Slow_t = \beta \cdot Close_t + (1 - \beta) \cdot EMA\_Slow_{t-1}$$

where:

- ( $M$ ) is the number of periods for the long-term EMA which is **30 days** for Viklund.
- ( $EMA\_Slow_t$ ) represents the long-term EMA at time (t).
- ( $Close_t$ ) is the closing price of the asset at time (t).

$\alpha = \frac{2}{R+1}$  is the smoothing factor for the long-term EMA, which is 0.8 for Viklund

### 3. Generate Trading Signals:

- "Buy" signal: EMA-Fast is greater than (**crosses above**) the EMA-Slow.
- "Sell" signal: EMA-Fast is less than (**crosses below**) the EMA-Slow.



**Figure (1) 5.4.1** - This figure illustrates the conditions that need to occur under viklunds core trading strategy for a **BUY** signal to be sent out to the executinary module to convert it into a valid buy trading signal

### Sell signal executed



**Figure (2) 5.4.1** - This figure illustrates the conditions that need to occur under viklunds core trading strategy for a **SELL** signal to be sent out to the executinary module to convert it into a valid buy trading signal

As shown in the **Figure(1) 5.4.1** above, Viklund places a **Buy** order when EMA-Fast (Light-blue) surpasses EMA-Slow (Dark-grey) past our set **Threshold**, aka difference between EMA-Fast and EMA-Slow current price points. The same is shown for the execution on a **Sell** signal when EMA-Slow surpasses EMA\_Fast in **Figure(2) 5.4.1**. The buy/sell orders are not filled immediately, it is common for the brokerages to take some time before filling an order, however,

this latency should not harm the results of the bot in any meaningful way. It is clear that Viklund searches for the lowest point of the asset as an indicator on when to enter the market and then.

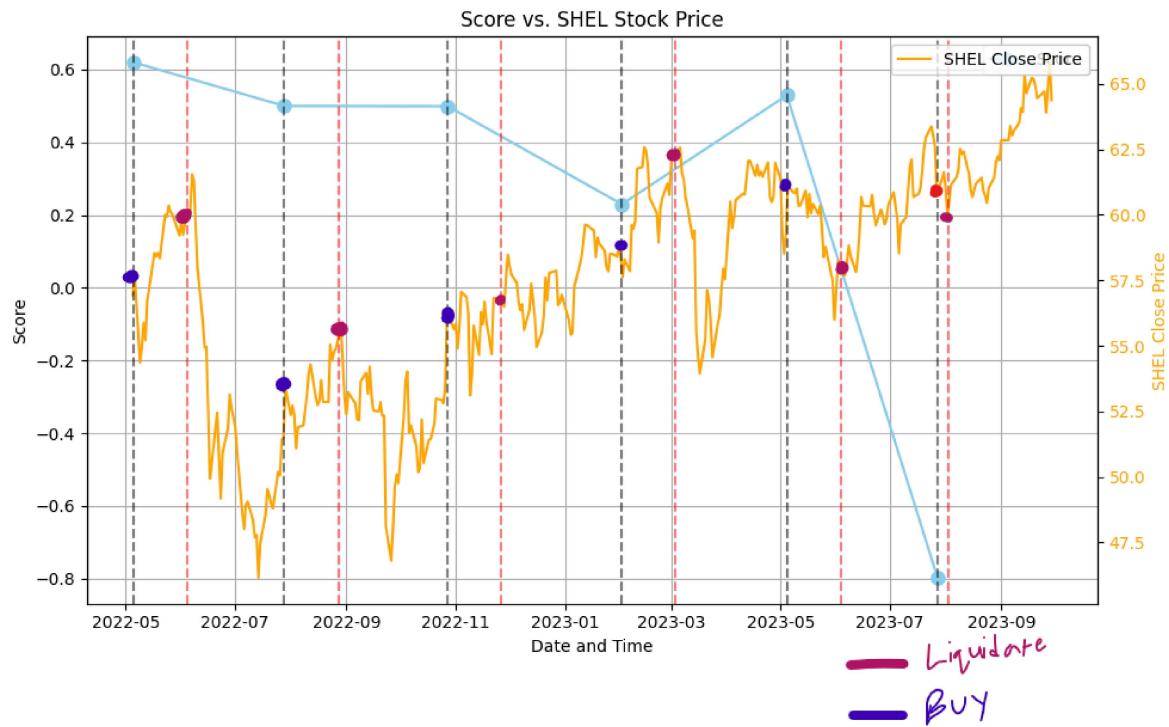
### 5.4.2 Viklund Neural Network Integration

Viklunds core trading strategy will seamlessly abort once articles reporting on SHELs quarterly earnings report are available before market open. Viklund will allow for multiple articles to be released before deriving a trading signal. This step is needed to aggregate the information from multiple sources, naturally filtering out potentially biased voices concerning the security. It will begin by pre-processing the i'th article pursuant to the steps stated in **5.2 Viklund Neural Network Sentiment Analyzer** within the pre-processing category **5.2.3**. Notably, it will break the article into individual sentences and then feed its tokenized format into our Neural Network. The source code can be found within the "*Viklund Model Live Usage-Multi Stock*" Jupyter Notebook. The network will output a probabilistic score for each of the 3 classes concerning the given article. We do the same for the rest of the articles pulled within the pre-determined earnings report date which has been publicly announced months ahead of time. We will then store all the probabilities for all the given articles within a list and compute their median value. By doing this, the final median probabilities for the 3 classes will represent the level of certainty Viklund has on its predicted sentiment for that quarter. If the value is above 0.20 or below -0.20, and the rest of the values class probabilities are within the model, Viklund will execute a trading order. If all the values are the same, or the probability is highest within the neutral class, Viklund will not execute a trading order and will simply revert back to its core trading strategy.

#### SHEL

	Date_time	Sentiment	Score	Sentiment_ID	Quarter
0	2022-05-05T08:00:00-04:00	POSITIVE	0.620060	2	1
1	2022-07-28T08:00:00-04:00	POSITIVE	0.500228	2	2
2	2022-10-27T08:00:00-04:00	POSITIVE	0.499517	2	3
3	2023-02-01T08:00:00-04:00	POSITIVE	0.229923	2	4
4	2023-05-04T07:30:00-04:00	POSITIVE	0.529570	2	1
5	2023-07-27T07:32:15-04:00	NEGATIVE	-0.797909	0	2

**Figure (1) 5.4.2** - Resulting data table from Viklunds sentiment analysis evaluation process with SHEL stock, sending long trading signals to the executionary module for every quarter except q2 2023, where a strong short signal is sent at an **80%** certainty. This is largely due to the fact that SHEL did miss its eps expectations in this quarter



**Figure (2) 5.4.2 - Viklund Executing Trading Signals When Score is above 0.20 or below -0.20 for SHEL**

When comparing our predicted results to the earnings per share performance within these quarters, we see that Viklund accurately conveys the right sentiment 100% of the time. Our model works very well.

However, the share price movement does not agree with us in Q4 of 2022 (2023). We see here that despite the company surpassing its estimated EPS leading to a correct positive sentiment prediction, the stock still dropped. This may likely be due to other geopolitical or financial problems of which could not be overshadowed by a positive quarterly EPS.

On a positive note, we see the 80% short signal generated (shown in **Figure(1) 5.4.2**) in the 2nd quarter of 2023 was a profitable trade, as the share price dropped from \$60.80 to 59.92, yielding a 1.4% return on the short.

## BP

	Date_time	Sentiment	Score	Sentiment_ID	Quarter
0	2022-05-03T11:31:03-04:00	POSITIVE	0.524051		2
1	2022-08-02T15:57:00-04:00	POSITIVE	0.470639		2
2	2022-11-01T13:12:00-05:00	POSITIVE	0.259945		3
3	2023-02-07T12:53:00-05:00	POSITIVE	0.304934		4
4	2023-05-02T12:05:00-04:00	POSITIVE	0.475735		1

**Figure (3) 5.4.2** - Resulting data table from Viklunds sentiment analysis evaluation process with BP stock, sending long trading signals to the executionary module for every quarter. The key focus here is that BP missed its eps expectations in 2022 Q4 (Feb 07, 2023). However, Viklund



**Figure (4) 5.4.2** - Viklund Executing Trading Signals When Score is above 0.20 or below -0.20 for BP. The black lines signify the moment a buy signal is triggered, the dark pink lines signify when a liquidate/sell signal is trigger

As mentioned in **Figure(3) 5.4.2** Viklund went against the fundamental hypothesis we initially had regarding the effect of an earnings call on a shareprice. Viklund sent a buy trading signal after 2022 Q4 earnings call, which reaveled an eps of 0.20 however, the street eps expectation was 0.21 [25] . Granted, this was a 5% difference, further investigation shows that Viklund's article analysis generated a 27% negative sentiment certainty (short position signal) from the quarters news articles, and it generated a 30% positive sentiment certainty (long position signal) from the quarters news articles. The long position signal prevailed by a slim 10% lead. This slim lead is largely attributed to BPs profits hitting a 2022 record high in Q4 despite its eps missing the mark [26]. Nevertheless, this was a profitable trade.

The share price did not go in our direction in 2023 (Q2). However, our BND protection did little to protect us in this area. (Explanation of this would have been provided if further time was available)

## 5.4.3 Risk Management

### 5.4.3.1 Correlation Hedging & Position Sizing

We utilize negatively correlated stocks to hedge our buy/short positions. Once Viklund attempts to make a valid trade off of a trade signal from its Neural Network it makes sure to allocate 80% of available capital in the direction of the provided trading signal, however it places the remaining 20% of available capital towards a negatively correlated stock in the oposite direction, in order to protect its losses in the event the market reacts the opposite way to the neural networks signal. For our negatively correlated stock we have chosen the Vangaurd Total Bond Market Index Fund ETF (BND). It is well known that bonds have proven to have a negative correlation to equities. The diagram below shows just how accurate this is. Alternatively, Airline parent company stocks could have been chosen as they also have a negative correlation between oil stocks, however, time did not permit us to carry out that additional research.



**Figure (1) 5.4.3.1 - Plot of SHEL price movement verse Bonds price movement. As you can note, they tend to react in opposite directions**



**Figure (2) 5.4.3.1 - Plot of BP price movement verse Bonds price movement. As you can note, they also tend to react in opposite directions**

There is a noticeable convergence, and negatively correlated reactions. On June 7th 2022, As BND goes up, SHEL & BP stock drops, and to a degree, vise versa as the year goes on. Investors would often rebalance their portfolio to favour BND when there is high volatility in the equities market as BND tend to be more stable during those periods.

### 5.4.3.2 Threshold & tolerance

The following are the threshold and tolerance formula under the buy & liquidate/sell conditions:

*Buy*

$$\text{EMA}_{\text{fast}} > \text{EMA}_{\text{slow}} \times (1 + \text{tolerance})$$

**Where:**

- ( $\text{EMA}_{\text{fast}}$ ): The value of the fast Exponential Moving Average, which responds more quickly to recent price changes.
- ( $\text{EMA}_{\text{slow}}$ ): The value of the slow Exponential Moving Average, which reacts more gradually to price changes.
- (tolerance): A threshold percentage used to ensure that the fast EMA must be at least  $(1 + \text{tolerance})$  times greater than the slow EMA to trigger the condition.

*Liquidate*

$$\text{if } \text{Portfolio}_{\text{SHEL}} > 0 \text{ and } (\text{EMA}_{\text{fast}} \times (1 + \text{tolerance})) < \text{EMA}_{\text{slow}}$$

**Where:**

- ( $\text{Portfolio}_{\text{SHEL or BP}}$ ): The quantity of **SHEL/BP** held in the portfolio. If greater than 0, this indicates the algo has allocated a certain percentage of the portfolio towards said securities. Validating this condition provides the basis to liquidate said positions in order to safeguard any potential earnings from loss.
- ( $\text{EMA}_{\text{fast}}$ ): The value of the fast Exponential Moving Average, which reacts more quickly to price changes.
- ( $\text{EMA}_{\text{slow}}$ ): The value of the slow Exponential Moving Average, which responds more gradually to price changes.
- (tolerance): A threshold percentage that adjusts the fast EMA to ensure it is significantly lower than the slow EMA, triggering the condition.

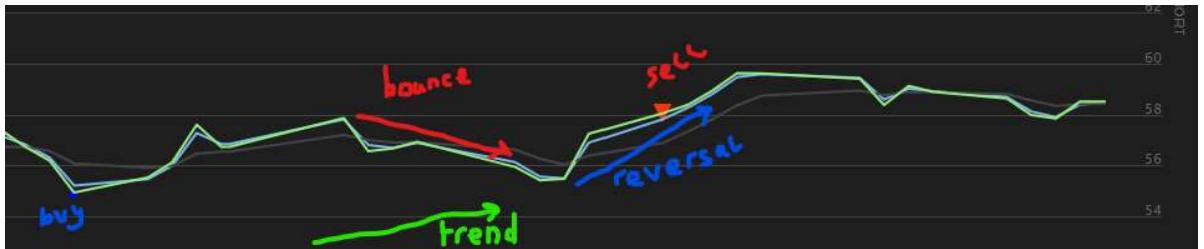
We incorporate threshold & tolerance to the risk management module in EMA to mitigate the false crossovers to address two events.

#### Event 1: Dead cat bounce

As shown in both **Figure (1 & 2) 5.4.1** the threshold created is aims to mitigate risks such as the dead cat bounce [18]. This essentially defines the event in which a short reversal on a direction of a stock occurs where at first the bounce may appear to be a reversal of the prevailing trend, but it is quickly followed by a continuation of the downward price move. This is largely due to traders or investors closing out a short position, or buying on the assumption that the security has reached a bottom. Often times these occurrences scare retail investors who

were initially invested in the trending direction of the security, and may even squeeze them into margin calls from their brokerages. However, it is vital that our bot factors this in and does not loose out on the upside of a trade due to this "sucker's rally".

Based on a series of back tests on securities, we have found the optimal threshold to be 0.015. Any lower and we would be overly exposed to margin calls triggered by dead cat bounces. Any higher and we will limit the upside on a long position, ultimately entering the position later than we needed to.



**Figure (1) 5.4.3.2** - This figure illustrates Viklunds performance specifically with the risk management strategy explained above (threshold). Here after we enter into a long/ **buy** position on SHEL stock, we see a **bounce** in the market (on Dec 27th, 2022) after SHEL share price hits its 25 day high. This is likely due to investors belief that the stock has topped out or the closing of thier short positions. However, Viklund does not fall for this due to its tolerance/ threshold. The **reversal** then takes a turn back into the prevailing **trend** which allows us to **sell/ liquidate** our position at a higher price, yielding a 5.4% profitable return on this trade.

### Event 2: High Market Volatility

The trading bot factors in a tolerance buffer when carrying out an EMA cross strategy. If there is high volatility, the fast and slow lines can become unstable, leading to multiple crosses that trigger trading invalid trading orders. This can drain available capital and take us to a loss. This is why we would need to base the foundational value of this tolerance on market volatility. We have currently assigned the value 0.015 to the tolerance variable, the same value as our threshold as they have are similar in utility with slight outcomes. However, before live trading, this variable should be assigned a value based on market volatility going back some period of time. Due to time limitations I was unable to derive the appropriate formula.

## 6. Evaluation

### 6.1 Viklund Neural Network Sentiment Analysis Evaluation

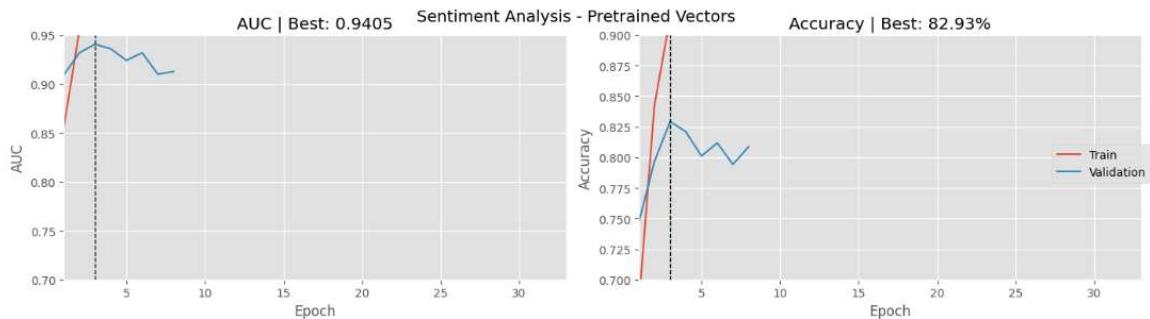
The final model stands strong in its predictive capabilities, particularly over the 50 other models that came before it. It achieved the following results.

- loss: 0.4624
- acc: 0.8288

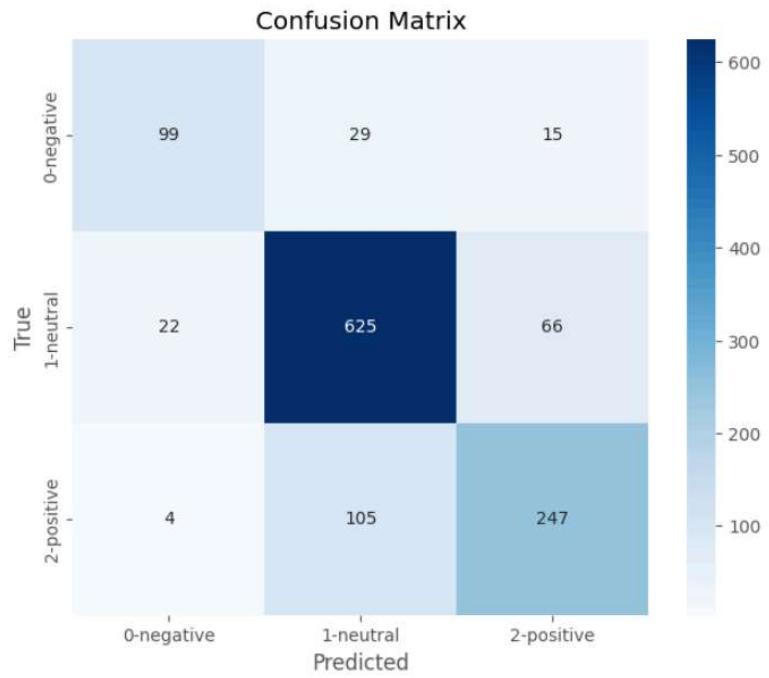
- AUC: 0.9485
- f1\_m: 0.8299
- precision\_m: 0.8368
- recall\_m: 0.8234

The decision was made to optimize the model with ADAM using categorical cross-entropy as its loss function. Our primary metric to represent the model competency was Area Under the Curve (AUC). AUC assess how well classification models work, especially when dealing with binary classification issues. We reworked how the AUC was calculated by calculating the AUC for each class, independently using the "one-versus-rest" (OVR) strategy. Essentially the OVR strategy works like this, for each class, you treat it as a positive class and group all the other classes as the negative class. You calculate the AUC for this formatted binary classification and then repeat this process for each class. You can then report the average AUC across all classes to represent overall model performance. Thus, the AUC evaluates how well a model can differentiate between negative, positive, and neutral classifications.

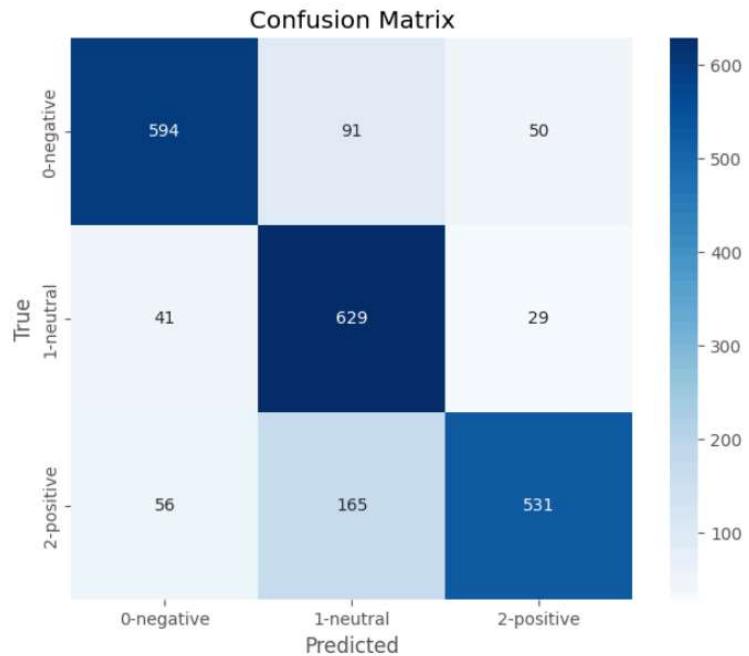
Unfortunately we are unable to display the hyper tuned models performance as this information was lost due to a computer crash, however we do have the graphs of the model without hyper parameter tuning.



**Figure (1) 6.1** - This figure illustrates Viklunds performance on cross validation in the AUC & Accuracy categories. It obtained a 0.9405 in AUC & a 0.8293 in accuracy



**Figure (2) 6.1** - Viklund model performance trained with attention layer on the Financial Phrase Bank Dataset alone, displayed in a Confusion Matrix



**Figure (3) 6.1** - Viklund model performance trained with attention layer on the combined dataset (including Financial Phrase Bank Dataset) displayed in a Confusion Matrix

The inclusion of the 4 other datasets aided in Viklunds robust nature. You can see there is stronger representation across all 3 classes in **Figure (3) 6.1** as opposed to **Figure (2) 6.1**. In regards to the positive and neutral classes, it seems to struggle distinguishing the two classes in **Figure (2) 6.1**. However it has a strong enough representation in **Figure (3) 6.1** that this is not a giant hurdle to overcome as evident in the sharp color distinction between the two classes in **Figure (3) 6.1**.

## Article Classification performance

Below are the results of the models predictive capabilities in regards to some news articles surrounding SHEL & BPs quarterly earnings.

### Q1 2022 May 05

<https://www.nasdaq.com/articles/shell-shel-q1-earnings-beat-on-oil-price-bumps-dividend>  
(<https://www.nasdaq.com/articles/shell-shel-q1-earnings-beat-on-oil-price-bumps-dividend>)

Sentence 1 POSITIVE : Europe's largest oil company Shell plc SHEL reported first-quarter earnings per ADS (on a current cost of supplies basis, excluding items — the market's preferred measure) — of \$2.38.

Sentence 2 POSITIVE : The bottom line came in above the Zacks Consensus Estimate of \$2.12 and surged from the year-earlier quarter's earnings of 82 cents per ADS, backed by stronger commodity prices.

Sentence 3 POSITIVE : Shell's revenues of 83.2 billion were up significantly from first-quarter 2021 sales of 59.1 billion.

Sentence 4 NEUTRAL : Meanwhile, Shell repurchased \$3.5 billion of shares in the first quarter.

Sentence 5 NEUTRAL : The energy group also announced that it has already brought back shares worth \$4 billion of the total 8.5 billion dollars scheduled for the first half of 2022.

Sentence 6 NEUTRAL : The remaining \$4.5 billion will be completed before SHEL comes out with second-quarter earnings.

Sentence 7 POSITIVE : On another positive note, Shell boosted its quarterly dividend by some 4% to 25 cents per share.

- Average: [0.10804257142857142, 0.3349105714285714, 0.5570467142857144]
  - Median: [0.030489, 0.038537, 0.717021] (array([0, 1, 2], dtype=int64), array([0, 3, 4], dtype=int64))
  - Quarter Performance Sentiment: [0.2025385, 0.120443, 0.6200600000000001]
  - **A long/buy position was executed**
- 
- 0th position = negative

- 1st position = neutral
- 2nd position = positive

## **Q2 2023 July 27**

<https://www.nytimes.com/2023/07/27/business/shell-earnings-q2.html>  
[\(https://www.nytimes.com/2023/07/27/business/shell-earnings-q2.html\)](https://www.nytimes.com/2023/07/27/business/shell-earnings-q2.html).

Sentence 1 NEGATIVE : Shell, Europe's largest energy company, said Thursday that its profit fell 56 percent in the second quarter from record-breaking earnings of a year earlier, to \$5.07 billion.

Sentence 2 NEGATIVE : The company blamed several factors for the falloff in adjusted earnings, including lower prices for oil and natural gas.

Sentence 3 NEGATIVE : Shell also said earnings in liquefied natural gas, a crucial business for the company, were sharply lower partly because a less turbulent environment meant there were fewer opportunities to profit from trading.

Sentence 4 NEGATIVE : Shell's performance, which came in below analysts' forecasts, shows that the petroleum industry may have reached a certain equilibrium after an extended period of volatility stemming from the pandemic and the war in Ukraine.

Sentence 5 NEGATIVE : If so, large, petroleum-dominated companies like Shell remain very profitable even in a lower price environment.

...

Sentence 7 POSITIVE : The company raised its dividend 15 percent for the quarter, to about 33 cents a share.

Sentence 8 POSITIVE : Shell also announced \$3 billion in share buybacks, a slight decrease from 3.6 billion dollars in the previous quarter.

Sentence 9 POSITIVE : The energy industry has demonstrated resilience in the last few years despite a very volatile environment. ...

Sentence 11 NEGATIVE : The global oil trade has also so far adjusted fairly smoothly to Western sanctions on Russian crude and refined products. ... Sentence 13 NEGATIVE : Brent crude, the international benchmark, sold for an average of about \$78 a barrel in the second quarter — a 31 percent drop from a year earlier.

Sentence 14 NEGATIVE : Natural gas prices in Europe were 65 percent lower.

- Average: [0.6034255714285715, 0.1165515, 0.2800230714285714]
- Median: [0.725006, 0.057901, 0.1250435]
- (array([0,1, 2], dtype=int64), array([10,0,4], dtype=int64))
- [0.6034255714285715, 0.1165515, 0.2800230714285714]
- **A Short position was executed**

### Showcase of model strengths:

`bp_q1_2022_article['Article'][2]`

Sentence 12 POSITIVE : It sold natural gas at 9.40 per thousand cubic feet compared with 4.11 in the year-ago quarter.

`bp_q1_2022_article['Article'][3]`

Sentence 21 NEGATIVE : Although costly, sources told Reuters in March that Looney had long had reservations about how the stake in Rosneft would fit into BP's plans to shift to renewables.

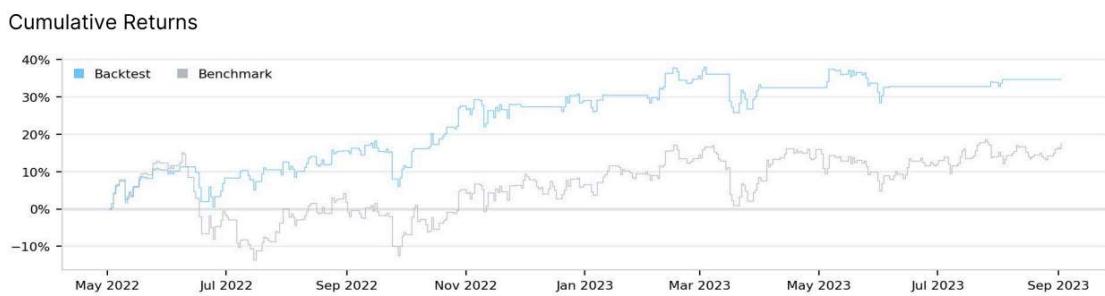
Sentence 3 POSITIVE : The strong trading and refining margins prompted the company to increase the dividend and invest in new oil and gas production

Above you can see examples of the models contextual powers. Further examples can be found in the Viklund Github repo within the "*Viklund Model Live Usage-Multi Stock*" Jupyter Notebook.

## 6.2 Viklund Algorithmic Trading Bot Powered By Deep Learning

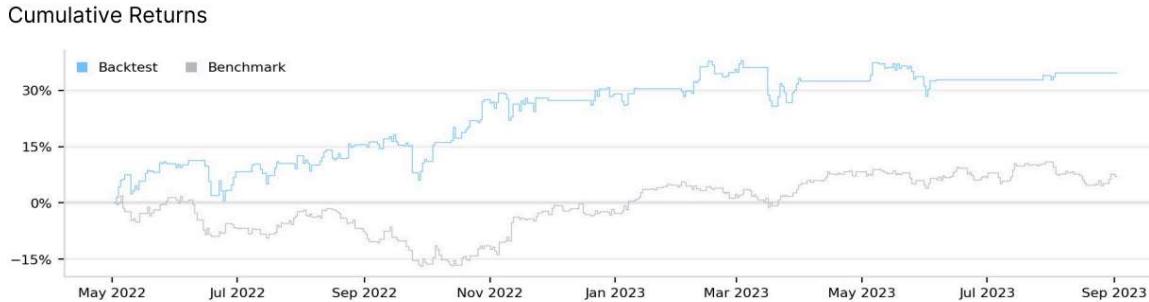
To test the Algorithmic trading bots performance in the stock market, we will look at its Sharp ratio, and percentage of returns against the buy and hold strategy. The buy and hold strategy is simply buying the stock of a company you believe to be strong, and holding the stock for the foreseeable future. The stock market yields around a 10% return on this strategy.

### 6.2.1 Viklund vs Benchmark Performance



**Figure (1) 6.2.1 - Viklunds trading strategy Vs Buy and Hold Bench Mark**

**Figure (1) 6.2.1** above plots the cumulative returns made by Viklunds (blue) vs the Benchmark (grey). The cumulative returns being represented here are the total returns on our portfolio over a specific period of time (May 2022-Oct2023). This visualization offers a way to track the overall performance of an investment strategy over time. The Benchmark represents the buy and hold strategy on the Oil & Gas sector. We see here that Viklund, our algorithmic trading bot **out performs this strategy by 203%**.



**Figure (2) 6.2.1 - Viklunds trading strategy Vs Stock Market**

When comparing Viklunds trading strategy to the stock market in **Figure (2) 6.2.1**, we see that Viklund, **our algorithmic trading bot outperforms the market by 277%** almost 300%. Never slipping into the negative, despite the strong effects of covid.



**Figure (3) 6.2.1 - Viklund adapting to earnings call period (This may be hard to see, switch display to high contrast to view SHEL share price details)**

In **Figure (3) 6.2.1** we get the opportunity to see Viklund in action. The figure above starts by showing Viklund **sell/liquidating** a position making the trade profitable just before the price drops. Viklund then identifies the lowest low by utilizing the EMA cross strategy and enters a **buy** position. Once Viklund recognizes we are within the earnings call period by identifying articles that have been provided to it, it rebalances its portfolio by liquidating any positions that do not fit its current agenda. It then proceeds to allocate 80% of its portfolio toward SHEL shares (**VN-Buy**) by scooping up any available shares in the open market, simultaneously allocating 20% of its cash to BND in the opposite direction to hedge, protecting its downside in

the event the stock changes direction and fails to respond positively to the earnings call. After its 30 day buffer period, it sells (**VN-Sell**) with a profit. It continues with this decision making

## 6.2.2 Sharp Ratio and Return

The sharp ratio assesses if the amount of risk taken justifies the potential return on an investment. When evaluating the risk-adjusted performance of various assets or portfolios, the Sharpe Ratio is especially helpful. However, due to the use of our neural network and its use of alternative data to make trading decisions, the traditional Sharp ratio metric will not work for us. If it is used it would look at the NN signal trades made by our trading bot as nonsensical, having no logical reason presented for taking such a spontaneously position. This is why we will use the Probabilistic Sharp Ratio instead. This ranking is used when the returns produced by a trading strategy do not conform to the assumptions of the conventional Sharpe Ratio because they have non-normal or non-standard distributions. Trading bots using neural networks frequently deal with complicated, nonlinear data that may not have a normal distribution. The risk-adjusted performance evaluation should take into account any subjective information, professional opinion, or preexisting opinions regarding the distribution of returns.

\$134,648.24	-\$231.80	\$0.00	\$34,648.24	51.096%	34.65 %	\$0.00	\$2,658,567.33
Equity	Fees	Holdings	Net Profit	PSR	Return	Unrealized	Volume

**Figure (1) 6.2.2 - Viklunds trading statistics**

Currently our PSR sits at 51%. I was unable to find the average PSR for retail traders. In order to compare results I found the average Sharp Ratio for retail traders within the study "*A Note on Trader Sharpe Ratios*"[16]. It stated that the average sharp ratio amongst the 53 participants was **0.70**. For some perspective, Viklunds Sharp Ratio is **1.11**. Even when this metric unfairly judges the use of alternative data/ subjective information, Viklund still comes out on top.

The average day trader earned a market return from 7-12% [17]. However Viklund has exhibited its ability to not only surpass the average day traders returns, but the overall market returns as well for this period of time.

## 7. Considerations

Training an algorithmic trading bot is incredibly difficult and can take many months - years. It would be wise to keep in mind that Viklund has only been tested on Shells & BPs quarterly Period from 2021-2023, and 2022-2023 respectively. He has not been properly tested on other periods as SHEL & BPs historical price data is not available on Quantconnect past certain points due to payment tears as well as SHEL's name change and unification in january of 2022. There was not sufficient time to test Viklund on other companies in other sectors. So this project simply stands as a proof to what Algorithmic trading with the application of deep learning can do for one.

When selecting article during earnings call period, we do not take the entire article, we consider the paragraphs inbetween the first heading and first subheading. This is because certain articles tend to consider other companies within the same article and we do not wish to make decisions

off other companies sentiments. We are able to identify target article being published around our companies earnings call period based on the inclusion of Shell, SHELL, and BP in their titles. My original approach of webscrapping articles was met with beefed up captcha walls for all the credible media outlets. This forced me to have to select each article manually. Within this step, it is possible that certain subconscious biases may have crept in during the selection process. However, I made it a point to select fairly.

The relevance of our models predictive power is entirely dependent on how quick journalists are able to put out their new articles reporting on the companies EPS. This essential holds Viklund's NN functionality hostage until available articles are presented to it to make predictions once the market opens on the following day. Choice of public data is limiting due to this. If one could join the earnings calls and create a transcript, perhaps a neural network could be trained on this, allowing for our signals relevance to be extended, profiting off of price movement before it even happens. There are other conditions as shown by the unexpected signals generated by Viklund in 2022 Q4 & 2023 Q1 from BPs earnings calls that expose the vulnerabilities of relying on the estimated eps being met, or missed. Despite the positive results achieved in the evaluation category of this report, this trading strategy is not full proof[11].

When attempting to upload Viklund Neural Network Model into the quantconnect environment, it was quite difficult as their environment was not compatible with the automatic upload of weights. The model needed to be reconstructed and the weights needed to be added, despite these efforts their environment did not allow the model to use its weights. I have been in touch with their support team for months and they stated they will ticket the issue and repair it in the coming months. We simply carried out Viklunds semantic analysis within jupyter notebook and uploaded a csv file containing the results into a drop box, which the quantconnect environment will do a bulk download on the data and utilize the csv rows sequentially (corresponding to current backtest date time and target earnings call date time) to eliminate lookahead bias.

## 8. Conclusion

We wanted to make efforts in leveling the playing field between institutional & retail investors. This took the form of developing an algorithmic trading bot that is based on some of the technical approaches retail investors currently exercise, as well as using some discretionary, fundamental analysis. More significantly, Viklund, our trading bot incorporates the use of alternative data and the deep learning analysis methods that hedge funds & other institutional investors currently utilize. We have entered the world of quantitative finance through the use of Algorithmic trading powered by Deep Learning with Alternative Data. Viklund has not only succeeded in outperforming retail traders, but it has done so in an understandable and adoptable way, proclaimed by the retail investors I have had the pleasure of showcasing this project to. Further development could incorporate information extraction techniques allowing Viklund to pull out the specific values for debt, earnings per share, and other fundamentals to carry out comparisons based on predetermined factors. Another further development would be to analyze the correlation between oil production and SHEL stock price, perhaps the increase or decrease in their EPS can be predicted months in advanced based on the volume of oil being produced over the past 3 months.

# Source Code

<https://github.com/Ben9991/Viklund> (<https://github.com/Ben9991/Viklund>) (Find report in git repo to view full reference links)

## References

- [1] LAURA COUNTS, *How hedge funds use satellite images to beat Wall Street—and Main Street*, Toggle Search Newsroom, MAY 28, 2019 <https://newsroom.haas.berkeley.edu/how-hedge-funds-use-satellite-images-to-beat-wall-street-and-main-street/> (<https://newsroom.haas.berkeley.edu/how-hedge-funds-use-satellite-images-to-beat-wall-street-and-main-street/>)
- [2] Stefan Jansen, *Machine Learning for Algorithmic Trading, Machine Learning for Trading: From Idea to Execution*, July 31, 2020, [https://www.amazon.com/Machine-Learning-Algorithmic-Trading-alternative/dp/1839217715?pf\\_rd\\_r=GZH2XZ35GB3BET09PCCA&pf\\_rd\\_p=c5b6893a-24f2-4a59-9d4b-aff5065c90ec&pd\\_rd\\_r=91a679c7-f069-4a6e-bdbba2b3f548f0c8&pd\\_rd\\_w=2B0Q0&pd\\_rd\\_wg=GMY5S&ref\\_=pd\\_gw\\_ci\\_mcx\\_mr\\_hp\\_d](https://www.amazon.com/Machine-Learning-Algorithmic-Trading-alternative/dp/1839217715?pf_rd_r=GZH2XZ35GB3BET09PCCA&pf_rd_p=c5b6893a-24f2-4a59-9d4b-aff5065c90ec&pd_rd_r=91a679c7-f069-4a6e-bdbba2b3f548f0c8&pd_rd_w=2B0Q0&pd_rd_wg=GMY5S&ref_=pd_gw_ci_mcx_mr_hp_d) ([https://www.amazon.com/Machine-Learning-Algorithmic-Trading-alternative/dp/1839217715?pf\\_rd\\_r=GZH2XZ35GB3BET09PCCA&pf\\_rd\\_p=c5b6893a-24f2-4a59-9d4b-aff5065c90ec&pd\\_rd\\_r=91a679c7-f069-4a6e-bdbba2b3f548f0c8&pd\\_rd\\_w=2B0Q0&pd\\_rd\\_wg=GMY5S&ref\\_=pd\\_gw\\_ci\\_mcx\\_mr\\_hp\\_d](https://www.amazon.com/Machine-Learning-Algorithmic-Trading-alternative/dp/1839217715?pf_rd_r=GZH2XZ35GB3BET09PCCA&pf_rd_p=c5b6893a-24f2-4a59-9d4b-aff5065c90ec&pd_rd_r=91a679c7-f069-4a6e-bdbba2b3f548f0c8&pd_rd_w=2B0Q0&pd_rd_wg=GMY5S&ref_=pd_gw_ci_mcx_mr_hp_d))
- [3] J., Mao, H. & Zeng, X. Twitter mood predicts the stock market. *J. Comput. Sci.* 2, 1–8 (2011).
- [4] E.F. Fama, *The behavior of stock-market prices*, *The Journal of Business* 38 (1) (1965) 34–105, <http://dx.doi.org/10.2307/2350752> (<http://dx.doi.org/10.2307/2350752>).
- [5] Austin Kilham, *Retail Investors: Definition, Pros, and Cons*, Sofi Learn, December 21, 2022, <https://www.sofi.com/learn/content/retail-investors/> (<https://www.sofi.com/learn/content/retail-investors/>)
- [6] Eugene F. Fama, *Efficient Capital Markets: A Review of Theory and Empirical Work*, *The Journal of Finance*, Vol. 25, No. 2, May, 1970, *Papers and Proceedings of the Twenty-Eighth Annual Meeting of the American Finance Association* New York, N.Y. December, 28-30, 1969 <https://www.jstor.org/stable/2325486> (<https://www.jstor.org/stable/2325486>)
- [7] Johan Bollena, Huina Maoa, Xiaojun Zeng, Received 15 October 2010, Revised 2 December 2010, Accepted 5 December 2010, Available online 2 February 2011, Twitter mood predicts the stock market, *Journal of Computational Science*, Volume 2, Issue 1, March 2011, Pages 1-8":<https://www.sciencedirect.com/science/article/pii/S187775031100007X> (<https://www.sciencedirect.com/science/article/pii/S187775031100007X>).
- [8] \* Moat, H., Curme, C., Avakian, A. et al. Quantifying Wikipedia Usage Patterns Before Stock Market Moves. *Sci Rep* 3, 1801 (2013).\*
- [9] Zhong, X., Raghib, M. Revisiting the use of web search data for stock market movements. *Sci Rep* 9, 13511 (2019).
- [10] Derek Greene, P’adraig Cunningham, Practical Solutions to the Problem of Diagonal Dominance in Kernel Document Clustering, University of Dublin, Trinity College, Dublin 2, Ireland <http://mlg.ucd.ie/files/publications/greene06icml.pdf> (<http://mlg.ucd.ie/files/publications/greene06icml.pdf>)

- [11] Henry Ongs, How Do Quarterly Earnings Reports Affect Stock Prices?, February 26, 2019, <https://realfinancepeople.com/earnings-affect-stock-prices/> (<https://realfinancepeople.com/earnings-affect-stock-prices/>)
- [12] nasdaq.csv, Kaggle, <https://www.kaggle.com/> (<https://www.kaggle.com/>) (Unable to find direct link due to Lack of time)
- [13] djia\_news.csv, Kaggle, <https://www.kaggle.com/> (<https://www.kaggle.com/>) (Available in github repo. Unable to find direct source due to Lack of time)
- [14] gold\_news.csv, Kaggle, <https://www.kaggle.com/> (<https://www.kaggle.com/>) (Available in github repo. Unable to find direct source due to Lack of time)
- [15] fin\_headlines.csv, Kaggle, <https://www.kaggle.com/> (<https://www.kaggle.com/>) (Available in github repo. Unable to find direct source due to Lack of time)
- [16] John M, Lionel Page, A Note on Trader Sharp Ratios, November 25th 2009, <https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0008036#:~:text=We%20then%20calculated%20the%20average%20standard%20deviation%20and%20the%20sharp%20ratio> (<https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0008036#:~:text=We%20then%20calculated%20the%20average%20standard%20deviation%20and%20the%20sharp%20ratio>)
- [17] Trading Thread, *What is the average returns for day traders?*, <https://tradingthread.com/what-is-the-average-returns-for-day-traders/> (<https://tradingthread.com/what-is-the-average-returns-for-day-traders/>)
- [18] James Chen, Dead Cat Bounce: What It Means in Investing, With Examples, August 10, 2022, <https://www.investopedia.com/terms/d/deadcatbounce.asp> (<https://www.investopedia.com/terms/d/deadcatbounce.asp>)
- [19] Guy Hatfield, Simon Hamlin, Simon Viklund, Steal from the Rich Give to Myself, Dec 1, 2019 <https://www.youtube.com/watch?v=q6a2ELvld4U> (<https://www.youtube.com/watch?v=q6a2ELvld4U>)
- [20] Robert Kissell, *Algorithmic Trading Methods: Applications Using Advanced Statistics, Optimization, and Machine Learning Techniques*, Sept. 18 2020, <https://www.amazon.ca/Algorithmic-Trading-Methods-Applications-Optimization/dp/0128156309> (<https://www.amazon.ca/Algorithmic-Trading-Methods-Applications-Optimization/dp/0128156309>)
- [21] combined\_fin\_news.csv, Kaggle, <https://www.kaggle.com/> (<https://www.kaggle.com/>) (Available in github repo. Unable to find direct source due to Lack of time)
- [22] news.csv, Kaggle, <https://www.kaggle.com/> (<https://www.kaggle.com/>) (Available in github repo. Unable to find direct source due to Lack of time)
- [23] Ankur Sinha, LexisNexis databse, Huggingface, Financial Phrase Bank, 2014, [https://huggingface.co/datasets/takala/financial\\_phrasebank](https://huggingface.co/datasets/takala/financial_phrasebank) ([https://huggingface.co/datasets/takala/financial\\_phrasebank](https://huggingface.co/datasets/takala/financial_phrasebank))
- [24] Ben Oguntimehin, Ankur Sinha, Kaggle, Hugginface, fin\_news\_optimized.csv, 2024, (Available in github repo folder dataset).
- [25] Tipranks bp earnings vs expected earnings <https://www.tipranks.com/stocks/gb:bp/earnings> (<https://www.tipranks.com/stocks/gb:bp/earnings>)
- [26] Joshua Warner, BP share price rises as profits hit record high in 2022, February 7, 2023, <https://www.forex.com/en-ca/news-and-analysis/bp-share-price-rises-as-profits-hit-record-high-in-2022/> (<https://www.forex.com/en-ca/news-and-analysis/bp-share-price-rises-as-profits-hit-record-high-in-2022/>)

# Terminologies

- **Technical Analysis:** Technical analysis is based on the idea that historical price and volume data can provide insights into future price movements. It focuses on studying charts, patterns, and technical indicators (e.g., moving averages, RSI, MACD) to make trading decisions.

- **Discretionary Analysis:**

Discretionary analysis involves a more subjective and qualitative approach to trading or investing. Traders or investors who use discretionary analysis rely on their own judgment, experience, and expertise to make decisions. This approach may involve considering a wide range of factors, including technical analysis, fundamental analysis, market sentiment, news, and personal intuition.

- **Fundamental Analysis:**

Fundamental analysis is based on the examination of a security's underlying fundamentals, such as financial statements, economic indicators, and industry trends. It seeks to determine the intrinsic value of an asset by assessing factors like earnings, revenue, profit margins, debt levels, competitive positioning, and management quality.