Allegheny College
Department of Computer Science

Senior Thesis Proposal

Using Computer Science To Successfully Predict Stock Price Swings

by

Christian Lussier

ALLEGHENY COLLEGE

COMPUTER SCIENCE

Project Supervisor: **Dr. Bonham-Carter**
Co-Supervisor: **Dr. Mohan**

October 18, 2020

Abstract

Using Computer Science To Successfully Predict Stock Price Swings

While stocks have long been a lucrative money-making option, unfortunately, many regular working-class people with 9-5 jobs lack the time and knowledge to be successful investors. Prior knowledge must be had to read charts to realize trends, ceilings, floors, ideal buying/selling points, and more. Additionally, stock prices are prone to movement due to many political, social, and geographical issues. My senior thesis will investigate the current tools being used to analyze stocks and predict their movements, while also investigating the relationship between stock prices, news, and the company itself.

My senior thesis project will also culminate in the creation of a tool, coded in Python with a Streamlit UI, that analyzes charts for trends and predicts stock price movements using neural networks, which is likely the most common CS resource being used. Stock price data would be scraped from Yahoo Finance, a reputable outlet that has been proven to work well with Python web scraping. The accuracy of the tool's ability to analyze charts could be tested by Pytest test cases containing sample charts, while the tool's prediction accuracy could be checked by archiving predictions and looking at the following stock price. Breaking the project up into phases will make it more feasible as most of the research and background implementation (stock price scraping) will be finished this semester, leaving ample time for the implementation of analysis tools. By adhering to a strict project schedule and researching the different methods in CS being used for stock prediction, I will create a tool that analyzes stock price charts for trends and predicts price movements.

Introduction

The stock market has often been a catalyst for economic booms and busts since its incarnation. In fact, stock market prices are one of the most important indicators of a country's economic growth [9]. Investors look to buy stocks low and sell them high on their quests to be profitable. There are many examples of people who made fortunes trading stocks, such as Warren Buffet who today has a net worth of over $70 billion. However, for every person who becomes a millionaire off of stocks, there are thousands who either see little to no growth in their stocks or even worse - losses. Those who are successful in the market devote a lot of time and effort in analyzing stock trends. Unfortunately, the average person, working a 9-5 job, often doesn't have the time or knowledge necessary to successfully trade stocks. Like William Donaldson, former chairman of the New York Stock Exchange, said, "If an individual investor doesn't have the means to keep up, they're at a huge disadvantage" [8]. My senior thesis will look to address the knowledge barrier necessary to successfully trade stocks by researching successful computer science based stock prediction methods culminating in the creation of a tool that predicts stock price swings.

In addition to the knowledge and time needed to be a successful trader, there are many other issues that make it hard for normal people to be successful in the market. With so many news sources available, it is "humanly impossible" to read all the relevant information in time to draw a conclusion to make a profit-maximizing plan [6]. Adding to this challenge of knowing relevant news is the fact that stock prices and the news surrounding them are constantly changing. This is why stock market prediction is still such a challenging problem. Stock markets are affected by uncertainty and interrelated economic and political factors, both locally and globally [1]. It is hard to predict things when the situation is constantly changing, a problem my tool would help to address.

My tool will help eliminate the need to have extensive trading training and complete extensive research on stocks while helping to reduce uncertainty when making stock trades. My tool will predict stock price swings by using a neural network based approach that takes in inputs regarding stock trends based on data downloaded from "Yahoo! Finance". With this, every time the user accesses the tool, it will predict the overall swing a stock price will have over the next 24 hours. This will give the user a good idea of whether they should buy or sell a stock. For example, if a trader is interested in buying a particular stock, they can use my tool to see what the projected stock price swing will be over the next day, so if they see a stock price is predicted to go down they can wait to buy it at a cheaper price. The tool will help aid users in buying stocks that will be successful.

While the overall problem my tool aims to address is that of the great knowledge and research barriers required to successfully trade stocks, my tool would cater to stock traders of all experience levels. Not only does my tool appeal to the regular 9-5 worker newly looking to make some side money by trading, but it will also have some appeal to more experienced and professional traders. This more skilled group of traders has the trading knowledge necessary to be successful but are still forced to complete hours of research to determine when or if they should buy or sell a stock. While the experienced

Year	S&P500 Average Annual Return
2015	1.4%
2016	12%
2017	21.8%
2018	-4.4%
2019	31.5%

Table 1: Average annual SP500 stock returns over the last five years [12].

trader will likely still complete background research on a stock, my tool would also give them a quick view of what to do with a stock and another indicator of how a stock is going to perform. By providing stock price swing predictions, both the experienced and the inexperienced trader will be able to make quicker, better decisions when it comes to buying and selling stocks.

My senior thesis project will investigate how computer science concepts can improve outcomes for stock market traders, both experienced and inexperienced. A lot of time and knowledge is necessary to successfully trade stocks but there are a number of research works that look to address this work using methods like sentiment analysis and neural networks. By researching these tools and the scholarly articles accompanying them, I will be able to determine what the best methods are for computer science-based stock market prediction today. I can then take ideas from these methods and create my own tool, likely using an artificial neural network, that predicts stock price swings. My tool will eliminate the issues of knowing extensive background information and having lots of free time to allow both new and old traders to make smarter stock trading decisions.

Related Work

Unsurprisingly, stocks and the field of computer science are closely intertwined. Since the stock market is so closely tied to the economy, determining the exact movements of stock prices is "considerably regarded" [9]. Innovations in recent years have led to the introduction of more and more CS concepts in the area of stock prediction. AI technologies such as neural networks, fuzzy logic, and expert systems are now employed in the area of stock forecasting [14]. As the spread of information quickens, more methods are being used to analyze it, such as viewing the general sentiment felt by Twitter users about a stock. The introduction of Computer Science concepts in the stock market have changed the game, allowing for more accurate stock forecasting predictions.

One of the most common and strongest CS concepts being used for stock prediction comes in the form of neural networks. One such example of a neural network based tool came when Kusuma et. al. proposed a method that used Deep Convolutional Neural Networks to analyze candlestick charts to indicate potential future stock price directions. The method showed a 92.2% and 92.1% accuracy for the Taiwan and Indonesian stock market datasets respectively [13]. Chootong and Sornil present similar

work in this area, as they present a trading strategy that combines a number of price patterns and indicators as they work to increase the returns on investment [5]. This tool uses a neural network ensemble to determine buy and sell signals for the next trading day. The results of Chootong & Sornil's tool showed that the proposed strategy yielded higher returns than traditional trading methods. Also, the work of Kimoto et. al. used modular neural networks to predict the Tokyo Stock Exchange Price Indexes. This modular neural network method proved to be successful, providing accurate predictions and an "excellent profit" in simulation exercises [11]. There are a variety of different neural networks that have been employed to forecast stock prices.

With this, Artificial Neural Networks (ANNs) have also commonly been used for different types of stock prediction. They learn from experience and can be made up of multiple layers. For instance, Di Persio and Honchar used an ANN to predict stock market indices trained on time series data with great success. This multi-layer ANN was made up of a convolutional neural network and recurrent neural network [7]. Similarly, another work using ANNs was found to be a dominant model for stock market forecasting [9]. Another successful work that used ANNs to predict stock market indices was completed by Kar, whose ANN design can be seen in Figure 3. This ANN achieved a best-case accuracy of 96% when tested on the Nifty stock index dataset [10]. Additionally, Al-Shayea used artificial neural networks in his work to successfully predict the future close prices of Dow Jones Index Stocks [2]. It seems that ANNs might be the best neural network method that can be applied to stock prediction because they can have multiple layers of neural networks and tools to help make a stronger prediction.

News articles are a huge factor in driving the prices and trading volumes of stocks. If bad news releases, obviously you will see stock prices go down and vice versa. As mentioned previously, it is virtually impossible to track down all the necessary information (news) needed to maximize stock profits. Some tools have worked to address this issue regarding stock news, with many using sentiment analysis techniques to analyze articles and predict stock price trends. Chowdhury and Routh propose a tool that gathers real-time news headlines and PRs, then analyzes them to predict the sentiment around the respective companies. The results of their research showed a 70% accuracy in identifying the correct sentiment and a 67% co-relation between positive sentiment and stock price trends [6].

Similarly, insights from things such as social media and search engines can be quite revealing toward the public's general sentiment about a topic. For instance, search engine traffic has been used to track and help predict social dynamics such as unemployment levels, home sales, and epidemic spread [4]. Though few works have investigated this, the same idea could be applied to market trends and stock prices. Bordino's work investigated this and found that trading volumes of stocks on the NASDAQ-100 are correlated to the daily volume of queries related to that stock(s) [4]. The same concept can be applied to social media. Bollen, Mao, and Zeng created a tool that analyzes the text of tweets submitted on a given day to provide daily time series of the general public mood. With this, the tool achieved an 86.7% accuracy in predicting the daily changes in closing values of the Dow Jones Industrial Average [3]. By looking at arti-

Method	Success Rate
News Headline Sentiment Analysis [Chowhurdy]	70%
Twitter Sentiment [Bollen]	86.7%
Deep Convulational Neural Networks [Kusuma]	92.1%
Deep Neural Network [Huynh]	65%
Artificial Neural Networks for Stock Indices [Kar]	96%

Table 2: Stock price or index prediction success rates by other scholarly works/tools.

cles, search engine queries, and social media, you can get a good feel for how a stock will perform based on the general public sentiment.

Computer Science concepts have been applied to predicting stock market trends for decades now. Recent years have brought about more innovations in this area as predictions become more accurate despite being harder to make than ever because of the uncertainty surrounding stocks and the constant spread of news. With this, techniques such sentiment analysis have become prevalent as news headlines and social media posts are analyzed in an attempt to determine the general sentiment around a stock and thus how it will perform. As AI techniques emerge, neural networks have likely become the most popular technique being used to predict stock market prices and price swings today. With this, there are many types of neural networks though Artificial Neural Networks seem to be one of the best options because you can combine different neural networks and tools in different layers of the ANN. A number of works have investigated using techniques like neural networks and sentiment analysis to predict stock trends with varying degrees of success.

Method of Approach

There many research and technical tasks that will need to be completed for my project to successfully predict stock price swings and address the field's knowledge barrier. For instance, before I begin working on the technical aspects of the project I will need to complete more research regarding the methods I will incorporate into my stock price swing prediction tool. Additionally, I will need to complete several technical tasks in the Python language, such as implementing a stock price web scraper and the neural networks that will be used for the price swing prediction itself. During my literature research and technical implementation, I will also have to constantly test and evaluate the tool to improve it while also ensuring it is error-free. In addition to my literature research, I will also complete testing and implementation tasks to produce a stock price swing prediction tool.

There are a few research tasks I will need to complete by finding and investigating different scholarly works. So far, I have looked into the different computer science methods that are used to predict trends in the stock market. As discussed in the "Related Works" section, these methods include sentiment analysis based prediction and chart based prediction using neural networks. Generally, it seems that neural network based prediction methods using number-based data are more successful than sentiment

methods. With this, my project will use neural networks that take in stock price trend information and indicators as inputs, to provide a stock price swing prediction. While my future research will generally continue to investigate all CS methods being used for stock prediction, the bulk of it will be used to investigate the use of neural networks for stock price prediction. There are many different types of neural networks and different inputs being used in them to predict stock price changes. By researching the different neural networks and inputs being used, I will be able to determine the best aspects of these research projects and incorporate them into my own stock price swing prediction tool.

There are many ways in which stock indicators and stock price data can be scraped from the web. As my tool is to be coded in Python, my web scraping method should also be implemented in this language. Luckily, it seems Python is commonly used to scrape stock data as when you Google "how to web scrape stock prices", all of the top results are associated with using Python for this task. Most of these articles and tutorials rave about the ease of using Python to scrape stock information while using packages such as "BeautifulSoup" to help complete this work. For instance, Figure 1, contains sample code using this package in Python to scrape real-time stock data, in this case the current price, from Yahoo! Finance. With this, this also begs the question of what outlets are the best at providing the most accurate and up to date stock prices. While this is something my final research may briefly address, in this day in age most websites seem to be relatively the same in terms of timeliness of updating prices and accuracy. Many Python tools used to web scrape stock prices use Yahoo! Finance because you can get free real-time stock market prices for stocks on the NASDAQ and NYSE, while these quotes come straight from the source, NASDAQ. My tool's stock price web scraper will be coded in Python and scrape stock data from Yahoo! Finance.

After implementing the portion of the tool that scrapes stock information, I will need to begin my analysis of the stock chart. With this, I will need to implement code that analyzes charts for specific patterns and trends that indicate something good or bad is about to happen to a stock, such as a golden cross which indicates bullish (good/up-trending) tendencies. If good or bad patterns are discovered in the analyzed charts, this information is fed into the neural network being used for the stock price swing prediction. With this, my preliminary research will also look into the different inputs that will be fed into the neural network itself. While the chart patterns discovered by the tool will be a heavily weighted input in the neural network, other information such as price-earnings ratios for a particular stock and overall market performance indicators will be key inputs. My preliminary research in this area will allow me to determine which indicators will be the best and help provide the most accurate stock price swing predictions.

As I mentioned previously, the bulk of my preliminary research will be devoted to researching the different neural networks currently being used for stock prediction and other professional tasks. This will allow me to determine which type of neural network is most applicable to my project area. For instance, there are a number of different neural networks such as Artificial Neural Networks, Deep Convolutional Neural Networks, and more. Once this research is complete, I can then begin implementing a neural

```python
"""Simple stock price scraper. Gets real-time stock prices."""

import bs4
import requests
from bs4 import BeautifulSoup


def get_price(abbreviation):
    """Gathers real time stock prices."""
    link = 'https://finance.yahoo.com/quote/' + abbreviation + '?p=' + abbreviation
    url = requests.get(link)
    soup = bs4.BeautifulSoup(url.text, features="html.parser")
    price = soup.find_all("div", {'class': 'My(6px) Pos(r) smartphone_Mt(6px)'})[0].find('span').text

    return price

def main():
    abbreviation = input("Enter stock abbreviation: ")
    print('Current stock price is : ' + str(get_price(abbreviation)))


main()
```

Figure 1: Python code to gather the real time price for a stock when the user inputs it's abbreviation by web scraping this information from Yahoo! Finance.

```
christianlussier@Christians-MacBook-Pro src % python3 scraper.py
Enter stock abbreviation: DKNG
Current stock price is : 43.81
```

Figure 2: Output of running the Python code in Figure 1. The user enters their chosen stock abbreviation and the real-time price is returned. The stock abbreviation, DKNG, for DraftKings, a sports betting service is used in this example. This sample program scrapes the real-time stock price, but can easily be modified to scrape all key stock prices for a day.

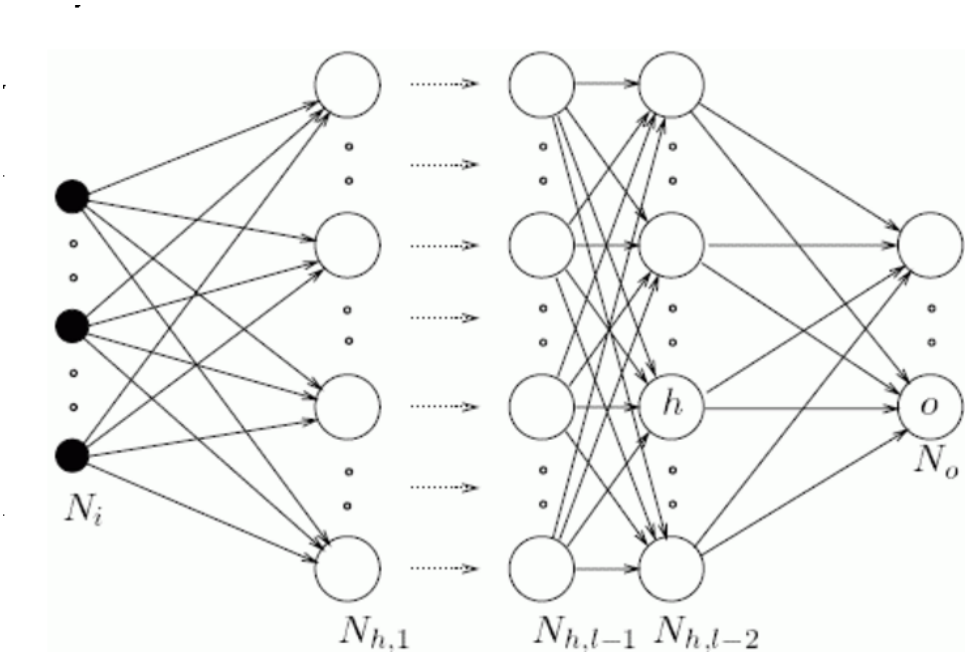

Figure 3: Artificial neural network used for stock market indice prediction [10].

network that predicts stock price swings using the previously discussed inputs. To stay consistent with my stock information scraper, chart analyzer, and testing methods, this neural network will be coded in Python. Luckily Python already comes with a number of different neural network libraries, many of which contain different types of neural networks consistent with those used in stock prediction research articles. These libraries include "TensorFlow", "NeuroLab", and "sci-kit-learn". Part of my research will involve investigating which libraries are the best once I narrow down the type of neural network to be used. This will make my implementation work easier, as I will not have to worry about coding the bulk of the neural network internals. Implementing the neural network and tweaking its inputs to reach the most accurate stock price predictions will be the most intensive part of my project.

To use the tool and display its results, I will create an app interface using Streamlit, another easy-to-use Python library. This will be done toward the end of the project, as I will mainly work on the project and test it in the command line. Then once things are working in the command line, I will translate it over to the web interface coded in Streamlit. Streamlit allows you to display text, graphs, images, and more using Python. I also used it to build the web interface for the tool in Software Engineering last year. This makes it ideal for use with the project as it can be used to quickly produce intuitive interfaces that work with already existing Python code.

While this is covered more extensively in the "Evaluation Strategy" section, testing is going to be another important part of the project. As my language of choice for this project is Python, it makes sense to use a Python-based test suite, Pytest. Pytest allows you to write test suites and test cases to ensure your programs are bug-free. I already have experience using Pytest to write dozens of test suites from previous courses making it perfect for use with this project. Additionally, I will also be using

linters, such as “pylint” and “black” to ensure my code is of good quality and the proper syntax. Pylint alerts programmers of syntax errors in Python code while black automatically re-formats code to meet most industry standards. I will have to create test suites using Pytest and use linters to ensure the code is bug free and accurate, as described further in the “Evaluation Strategy” section.

My project seems to be feasible for a number of reasons. At this point, I feel I am a fairly experienced researcher with skills in identifying useful and informative scholarly articles. While there is a lot of research I will need to do for this project, especially in the areas of neural networks and stock indicators, I feel that my research skills and the plethora of articles that look into using computer science concepts with stock prediction will help. I plan on completing my tool’s implementation in Python, which I have been using for four years and feel is my strongest programming language. Python is one of the world’s most popular languages meaning there is a lot of content on it and many useful libraries, like those that help implement neural networks, that will make a lot of the project’s coding easier. With this, there are also many publicly available tutorials on using Python to scrape stock information from the web. Additionally, testing the project will also be quite easy as I am experienced in Python’s main testing library, Pytest, and also have experience ensuring the quality of Python code syntax using tools like pylint and black. In addition to my experience in the areas of research and technical implementation using Python, I also have devised a strict research schedule that will keep me on track throughout the project. By following a strict research schedule and using Python libraries to make implementation easier, I will be able to create a high-quality tool on time.

There are a variety of tasks that will need to be completed to produce a body of quality writing and a tool that successfully predicts stock price swings. The beginning of the project will involve a lot of background research in the areas of what computer science concepts are being used for stock prediction, neural networks, and key stock indicators. While I will be writing throughout the duration of the project, once my preliminary research is complete I will be able to finalize the neural networks and indicators to be used in the project. With this, I can then begin to implement a stock information web scraper and stock chart pattern analyzer. Then I can implement my neural network that will be used for the stock price swing prediction itself using this data. Additionally, to ensure my project is of high quality I will be working on testing, using a Pytest test suite and Python linters, throughout the project. By completing background research and using my skills with Python, I will be able to successfully implement a stock price swing prediction tool.

Evaluation Strategy

In order to evaluate the accuracy and strength of my proposed method, there are a variety of different steps that will need to be taken. For instance, the quality of the code should be tested using things such as pylint while the usability of the tool will need to be manually tested. Additionally, the accuracy of the code will have to be tested using a test suite created for the project. My senior thesis project will be evaluated by using

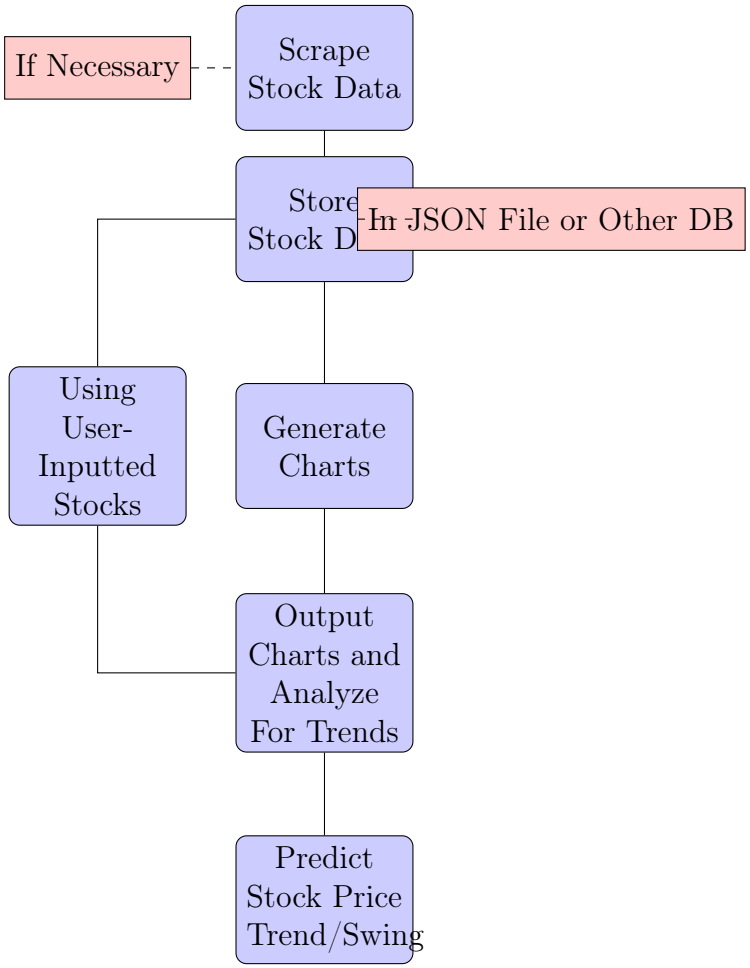

Figure 4: Simple flow diagram for the Stock Price Swing prediction tool.

automated testers, code linters, manual interface testing, and result accuracy testing.

One such step comes in the form of testing. Since my tool is being created in Python, I will use Pytest to create a test suite for my tool. This tool will evaluate the correctness of my code by ensuring each function provides the expected results. For instance, in order to ensure my tool is correctly analyzing specific chart trends, I will provide sample input stock chart data and ensure my tool analyzes the trends within them correctly.

The quality of the code and it's syntax should also be tested. This would involve using linters, such as pylint, to ensure the code is free of formatting errors and meets industry standards. With this, black, a python reformatter will also be used on all project code to ensure it has quality syntax. Additionally, to ensure the code is being properly tested and is accurate, I should strive for a high code coverage, of at least 90%, which I could test using cover.

Outside of ensuring my the internals of my tool are sound, I also want to ensure it's results, price swing predictions, are of adequate accuracy. My tool will allow it's users to save the stock price swing predictions locally to their machines. This means this data can be stored for future analysis to see if my tool's stock price swing predictions are accurate to the stock's actual price movement. This could be done manually with a certain group of stocks daily or, if time allots, a historical analysis feature could be created by the tool to automatically compare the predictions to the real stock prices. After ensuring my tool is bug-free and functions, a large portion of my time will go toward testing the accuracy of it's stock price swing predictions versus the actual prices of those stocks. In doing this evaluation, I will be able to improve my predictions if they are not that accurate by experimenting with using different inputs and weights in the neural network being used for stock price prediction.

Evaluating the usability of my tool will involve more manual methods than testing the correctness of my code. To do this, I will have to find a number of testers (fellow students) and have them use my tool. While they use the tool, they will fill out a survey about things such as bugs found, interface usability, and more. This will allow me to ensure my tool is accessible and usable for all potential users.

While the bulk of the work on my project will go towards research and technical implementation of the tool itself, I will also have to take the time to evaluate the correctness and accuracy of it. I will have to ensure the correctness of my code by testing it's functions and components by writing a Pytest test suite that covers at least 90% of the project code. I will also make sure my code meets industry standards by using linters like Pylint and Black to ensure my code's syntax is readable and efficient while also performing manual testing on the interface of my project. After this, I will also work to ensure the accuracy of my tool's stock price swing predictions by comparing these predictions with the actual stock price changes. By evaluating my tool using a variety of different methods like code testing, code linting, and accuracy testing, I will ensure the final version of my tool is solid.

Time Frame	Task
11/20/2020	Finalize Research on Methods
11/23/2020	Complete Draft of Chapter 1 on Methods
11/27/2020	Finalize Research on Key Stock Trends/Indicators
11/30/2020	Complete Draft of Chapter 2 on Inputs
12/7/2020	Finalize Research and Writing For First Two Chapters
12/30/2020	Complete Code Implementation for Stock Scraping
1/18/2021	Complete Code Implementation for Chart Analysis
1/25/2021	Complete Half of Remaining Chapters
2/10/2021	Complete Code Implementation for Stock Price Swing Prediction
2/17/2021	Complete Code Implementation for App Interface using Streamlit
2/24/2021	Finish Code Testing/Quality Checks
3/1/2021	Complete Bulk of Chapter Writing
3/1/2021	Finish Accuracy/Manual Testing
3/10/2021	Final Touches

Table 3: Tentative research schedule for Senior Thesis Project.

Research Schedule

In order to complete my project ahead of schedule, I will have to adhere to a strict timeline. I have a variety of tasks that I will need to complete to successfully finish the project. I have to complete preliminary research before I begin coding the tool so I can finalize my chosen method. In addition to research and writing, I must implement a web scraping tool and chart analyzer in order to produce the final product which is a tool that successfully predicts stock price swings. By following the schedule displayed in Figure 3, I will be able to create a project that exceeds expectations in a timely fashion.

First, I will need to complete my research on different computer science methods currently being used for stock price swing predictions. More specifically, I will look into neural networks and evaluate them to decide which is to be used in my tool. Additionally, I will also need to look into the key stock trends and indicators that will be used as inputs in my stock price prediction tool. This will allow me to complete key writing for the project.

After this preliminary research is complete, I can move more into the technical side of the project. With this, I will need to implement code in Python that scrapes stock data from Yahoo! Finance. I will then need to store this data and analyze it. With this, I will have to implement code that analyzes charts of stocks to determine their recent trends and key indicators. Once this task is complete, I will then be able to work on the stock price swing prediction portion of the tool and later, the app interface created in Streamlit.

While I plan to work on the test suite of the program as I implement code, I will also allot myself time to finalize my testing toward the end of the project. Additionally,

during the "Finish Accuracy/Manual Testing" portion of the project, I will test the accuracy of my program's stock price swing predictions against the real stock prices. I will also perform manual testing of the program to ensure the interface is functional and simple for users.

By creating and following a realistic research schedule that breaks my project up into key components, I will be able to successfully research and implement a stock price swing prediction tool. I will begin by completing more research on the methods and indicators commonly being used for stock predictions right now. While I will complete more research and writing throughout the duration of the project, this preliminary research will give me the background knowledge necessary to create a successful swing prediction tool. Afterward, I will be able to begin implementing key components of my tool like the stock information scraper, chart/trend analyzer, and the neural network that will be used for stock price swing prediction itself. With this, I will also have to test the project code and accuracy of the tool's predictions to ensure it is of high quality. A set research schedule will make it possible for me to complete the work for this project.

Conclusion

The stock market can be financially lucrative for savvy traders. Successful traders usually have a lot of time to trade and extensive background knowledge with regard to reading charts, stock news, and trends. Unfortunately, the common person doesn't have the time or skills necessary to net large returns on the stock market. Additionally, the stock market has long been plagued by uncertainty stemming from political factors, social issues, and more. My work will look to address the problem of stock market uncertainty and the knowledge barriers that make it hard for most people to be successful traders by creating a stock price swing prediction tool.

There are a quite a few works and tools that have been created that look to make stock market predictions using computer science methods, like neural networks and sentiment analysis. My preliminary research will involve looking at these other scholarly works and identifying the most successful methods from them. For instance, there are many different neural networks being used for stock prediction today, such as the convolutional neural network and artificial neural network. Additionally, I will research the key indicators and chart trends that give a picture of how a stock will perform, picking out which are the most important. I will then be able to create my own tool that predicts stock price swings using some type of neural network deemed to be most effective through my research.

This tool will be based in Python, which I am quite experienced in, as it is one of the most popular programming languages worldwide and already features libraries that make it easy to implement neural networks. Additionally, there are also a number of tutorials and ways that make it easy to web scrape stock data via Python. With this, I have also used Pytest and linting tools previously and will use them again in this project to ensure my code is of the highest possible standard. Once my python-based neural network tool that predicts stock price swings is complete I will also take the

time to evaluate its accuracy and usability by comparing the predictions of my tool to the real-world results and by performing manual interface testing.

My research will investigate the most successful Computer Science methods being used for stock prediction today. In doing this I will gain the background knowledge necessary to be able to create my own neural network that predicts stock price swings in Python that is extensively tested by Pytest. To complete the project, I will have to follow a strict research schedule that splits things up into a preliminary research phase, a implementation phase, and a finalization of testing phase. By performing extensive background research I will be able to create a stock price swing prediction tool that exceeds expectations ahead of schedule.

Bibliography

[1] Agrawal, J., Chourasia, V., and Mittra, A. State-of-the-art in stock prediction techniques. *International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering 2*, 4 (2013), 1360–1366.

[2] Al-Shayea, Q. Neural networks to predict stock market price. In *World Congress on Engineering and Computer Science, San Fransisco* (2017), pp. 1–7.

[3] Bollen, J., Mao, H., and Zeng, X. Twitter mood predicts the stock market. *Journal of computational science 2*, 1 (2011), 1–8.

[4] Bordino, I., Battiston, S., Caldarelli, G., Cristelli, M., Ukkonen, A., and Weber, I. Web search queries can predict stock market volumes. *PloS one 7*, 7 (2012), e40014.

[5] Chootong, C., and Sornil, O. Trading signal generation using a combination of chart patterns and indicators. *International Journal of Computer Science Issues (IJCSI) 9*, 6 (2012), 202.

[6] Chowdhury, S. G., Routh, S., and Chakrabarti, S. News analytics and sentiment analysis to predict stock price trends. *International Journal of Computer Science and Information Technologies 5*, 3 (2014), 3595–3604.

[7] Di Persio, L., and Honchar, O. Artificial neural networks architectures for stock price prediction: Comparisons and applications. *International journal of circuits, systems and signal processing 10*, 2016 (2016), 403–413.

[8] Duhigg, C. Stock traders find speed pays, in milliseconds. *Sat 1*, 05 (2009).

[9] Göçken, M., Özçalici, M., Boru, A., and Dosdoğru, A. T. Integrating metaheuristics and artificial neural networks for improved stock price prediction. *Expert Systems with Applications 44* (2016), 320–331.

[10] Kar, A. Stock prediction using artificial neural networks. *Dept. of Computer Science and Engineering, IIT Kanpur* (1990).

[11] Kimoto, T., Asakawa, K., Yoda, M., and Takeoka, M. Stock market prediction system with modular neural networks. In *1990 IJCNN international joint conference on neural networks* (1990), IEEE, pp. 1–6.

[12] Knueven, L. The average stock market return over the past 10 years, Aug 2020.

[13] Kusuma, R. M. I., Ho, T.-T., Kao, W.-C., Ou, Y.-Y., and Hua, K.-L. Using deep learning neural networks and candlestick chart representation to predict stock market. *arXiv preprint arXiv:1903.12258* (2019).

[14] Liu @articlechootong2012trading, title=Trading Signal Generation Using A Combination of Chart Patterns and Indicators, author=Chootong, Chalothon and Sornil, Ohm, journal=International Journal of Computer Science Issues (IJCSI), volume=9, number=6, pages=202, year=2012, publisher=International Journal of Computer Science Issues (IJCSI) , J. N., and Kwong, R. W. Automatic extraction and identification of chart patterns towards financial forecast. *Applied Soft Computing 7*, 4 (2007), 1197–1208.