

# AI Powered Investment Management System

Yash Jain  
UNI: ycj2103

Ayush Raj  
UNI: ar4283

Sunjana Ramana  
UNI: sc4921

**Abstract**— Previously, there have been several platforms where a user can either buy/sell a stock. There also have been platforms where users can just analyze a stock or platforms where a user can just manage his/her portfolio. This leads to a need for several different applications/platforms for a user to manage his/her investments. This might lead to some confusion that a person might not want while managing their investments. It will be more helpful for a user to have a comprehensive platform where they can buy/sell investment, analyze a particular investment (crypto, stocks, mutual funds), manage their portfolio and many more things. This paper demonstrates a complete end-to-end investment management platform/application for users to make a trade, manage their portfolio, analyze an investment using graphs, news headlines and twitter sentiment about stocks. The live data of the S&P500 stocks is fetched everyday using the Yahoo finance API and stored in a MySQL Database. This data has various features like opening price, closing price, day's high, day's low, etc. Using these features, a recommendation system has been built that uses industry standard indicators like golden cross decision, moving averages divergence and convergence, and the relative strength index to recommend investment options to a user. A user can also perform a thorough analysis of investment options using several charts like line chart, candle sticks, forecasting using FB prophet algorithm, twitter sentiment analysis using live data fetched from Twitter API and also read only the relevant headlines for a particular investment option. These disparate options give users a way to perform thorough analysis of an investment and make an informed decision about it. Additionally, this paper also includes a way for the users to track their current portfolio and make a trade or add a particular investment option to a watchlist where they can perform an even deeper analysis. In our project, we use Streamlit to build the frontend and Python to build the entire backend of the application.

## I. INTRODUCTION

More than 50% of the US population invests in the stock market. As the banks in the US offer very small interest rates on savings, most people feel that they cannot cope up with inflation if their money is idle in the banks. Hence, people like to invest their money and since the US stock market has consistently performed well in the past, users have grown comfortable investing their money. Users have realized that investing in the stock market is the right way to save and multiply their money.

Historically, there has been a misconception that trading is some kind of gambling which is purely based on luck. This is a misconception among the people who do not have knowledge about how the stock market works and how money is in the stock market. In the recent years, investing has become a trend and a lot of people have started buying and selling stocks, it is still a question whether they are doing it with the right knowledge or not. So, obviously, it turn out to be a gamble if a person puts his money in a stock based on luck and without any financial literacy. Though it is a known fact that one can make money if one has knowledge of the stock market, the learning process is not easy and the outcome is not guaranteed. Therefore, people who jump into the process without having a solid understanding of the stock market and how trading works, struggle a lot initially.

When people invest in the stock market, they seek high returns. However, it is likely that they might not make the profits they expect. This is because the stock market is driven by the emotions of an individual. To overcome the uncertainty caused by human emotions, most investors look to employ AI to make these decisions for them. Investing just does not stop with stocks, other options like cryptocurrencies, ETFs are viable investment options. Retail investors struggle because of the wide variety of options available.

It's a no-brainer that choosing the right stocks is the pain point for most of the investors. Even the most learned investors like Warren Buffett, Charlie Munger etc. spend a lot of time analyzing stocks from various perspectives and then picking the right stock to add to their portfolio.

Trading platforms like Robinhood allow investors to buy and sell stocks by creating a trading account, however, it does not help the users to pick the stock matching their interests and aspirations. Due to this, an investor who is not experienced enough needs to use some other platforms like 'Tradingview' etc. On 'TradingView' the investor can analyze the stock and come up with a potential list of stocks to invest in. Platforms like 'Tradingview' have a third-party API integration to connect it with the trading account of the users. But the downside of this is that the users feel it is too slow. Timing is key in investing in the stock market. To validate his intuition about whether it is the right time to buy that stock, an investor might want to do some more research on public

sentiment. For this, he would have to visit a community platform like 'Fintwit' where he can understand the overall emotion associated with the stock and get trading signals from Social Media opinions. After doing all this research on separate platforms the user would shortlist the stocks he prefers and again come back to a trading platform like Robinhood to finally make that purchase.

The purpose of this study is to develop an AI-based intelligent investment and wealth management system. This will help retail and institutional investors to get involved in the market, but without wasting much time to choose the right stock or choose the right time to enter. The main objective of this study is to develop a model which will help investors to make investment decisions based on users' investment amount, investment period, risk appetite, expected returns, etc. Investing doesn't stop with just buying shares/ bonds for the first time. Managing the portfolio in an efficient way and tracking the investments are two vital challenges that investors face in the real world. If buying the shares is by itself a huge deal and difficult to do manually, then managing the portfolio is nowhere near easy. Therefore, this project doesn't stop with AI-based investing. It will also provide an automated way to manage the wealth intelligently and track investments in a portfolio.

The outcome of the above-mentioned challenges faced by investors, we develop a platform where the investor can analyze the performance of a stock, read the latest news about the stock, understand the overall sentiment score, and as soon as they make their decision, they wouldn't have to visit any other website, they can directly trade from the same platform they are currently in. So, the objective of the project is to develop an end-to-end smart application for the user to become an informed investor without wasting their time on multiple platforms. Instead of wasting time hopping from one platform to another, the user can their entire time analyzing and finalizing the stock they want to invest in on our platform.

## II. RELATED WORK

Stock markets, cryptocurrencies, Mutual funds and ETF prices are dynamic, unpredictable, volatile and non-linear in nature. Predicting their future prices is a herculean task as the prices depend on several qualitative and quantitative factors. Quantitative prediction takes in account factors like closing prices, opening prices, etc. to predict a future price. The second type of prediction called the Qualitative prediction takes in account several other factors like government policies, public sentiment, company profile, text information in news articles, etc while predicting the future prices. Nowadays, several techniques are available that can help convert the qualitative factors to quantitative factors and perform a neat and close to accurate analysis of the future prices [2]. Data available for making future predictions for prices is immense. Along with large amounts of available data, the

computational power has also risen. On top of more data and increasing computational power, there has been some great advancement in several machine learning algorithms. New machine learning algorithms have improved efficiencies by 60-86 percent when compared to previous methods [4]. This has led to the development of intelligent techniques to predict patterns in historical data in order to predict the future prices with great accuracy. Therefore, to minimize losses and maximize profit such techniques could prove to be highly useful. [3][4]

Mehar Vijn et. al [5] implemented a stock price prediction model using a random forest regressor and artificial neural networks. The models were trained using the stock price data from the past 10 years. Along with the already existing features in the dataset namely Open, High, Low and Close prices, new features were also created using these existing features in order to get a better training model. The evaluation metrics that were used were RMSE and MAPE. The introduction of new features using the already existing features could be something that can help improve the model to a great extent as the model can now have more features to get trained on and analyze the trends in the data in a more detailed fashion. Also, RMSE can be a great metric to measure the performance of our model as well. We have incorporated these two techniques in our approach.

Mohankumari C et. al [6] used the Autoregressive integrated moving average (ARIMA) model to make predictions for future stock prices. ARIMA models are robust and efficient, especially for making short-term predictions, when compared to the popular ANN techniques. Using the Arima model for our implementation can prove to be useful. It will help us give better forecasting of the future prices.

Alvin Ho et. al [7] used a hybrid model including Linear regression + Long short-term memory (LSTM) cells. This model proves to be an effective model as it uses LSTM cells which help in increasing the short-term memory of the model and in turn make better predictions. In our implementation, we use a hybrid model that encompasses the ARIMA model with LSTM cells. This will ensure we get a better forecasting of the prices. Additionally, Alvin Ho et. al [7] has also employed a GUI to provide investment recommendations to user by analyzing their previous purchases as well. However, the user's investment strategy may change with time. Hence, this is not a good way of providing users with buying recommendations. Instead, we first make a user profile by assessing what investment strategy the user wants to employ and depending on that recommend investment options

## III. DATA

In our project, we worked with historical and real-time stock price data for the S&P 500 stocks obtained from Yahoo

Finance. This data is obtained using the Yahoo finance API. This data has the following features:

- Opening price: Opening price of a stock
- Closing price: Closing price of a stock
- High: Highest price of a stock on a particular day
- Low: Lowest price of a stock on a particular day
- Volume: Total volume of a stock traded

We collect the past 5 years of stock price data for the S&P500 stocks. We store this data in a MySQL server Database and we have scheduled the data to refresh every day, 9 am EST to make room for the new day's stock price data and to incorporate this data in our future recommendations and predictions.

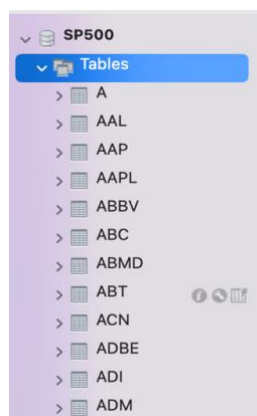


Figure 1. Schema structure for stock price data

The velocity of the data available is very high as the stock prices keep changing every microsecond so every microsecond, we have new data available. The volume of data available is also very high and we are limited only by the computational resources. For this project, we use the several features available to us like opening price, closing price, day's high, day's low and hence there is a great variety of data as well.

We use this data to generate features for various technical indicators that we later use to recommend stocks to a particular user. We create features for the following technical indicators and assign them a Boolean value based on the conditions satisfied. These technical indicators are industry standard and are used by leading platforms to recommend stocks to users.

- Golden cross decision: This indicator is marked 'True' when the 50-day moving average crosses the 200-day moving average and is increasing. In other scenarios this indicator returns a 'False' value.



Figure 2. Golden cross decision

- Moving Average Convergence and Divergence: This indicator is marked 'True' when the MACD line crosses the signal line and is increasing. This depicts an upward trend in market.



Figure 3. Moving Average Convergence and Divergence

- Relative strength index (RSI): This indicator is 'True' when the value of the ratio of RSI to SMA (RSI/SMA) is less than 30%. When the value is less than 30%, it shows that a particular stock is oversold and hence its price is low enough to buy it.



Figure 4. Relative strength index decision

Now, if for a particular stock, all of these indicators return 'True', then a new feature that indicates to buy a stock is set to 'True' and the stock is recommended to the user on the platform. If any one of these indicators returns a false value, then the feature will indicate to not buy the stock and the stock will not be recommended.

This project also uses data from twitter to analyze the sentiment of a particular stock on twitter. The fetch live twitter data about a stock using the Twitter developer's API. The latest top 5000 tweets of a particular stock are taken into consideration and then the overall twitter sentiment of a particular stock is determined.

Once the tweets for a particular stock are retrieved, we apply some preprocessing before judging the sentiment of the tweet. The preprocessing includes removing the common stop words like 'the', 'this', etc. Once these stop words have been removed the polarity and subjectivity of a tweet is determined. Polarity tells the true sentiment of a particular tweet. If polarity is close to '1' then it is a positive tweet about the stock and if it is close to '-1' then the sentiment for that tweet will be negative.

```
def preprocess_tweets(tweet, custom_stopwords):
    processed_tweet = tweet
    processed_tweet.replace('^\\W+', '')
    processed_tweet = " ".join(word for word in processed_tweet.split() if word not in stop_words)
    processed_tweet = " ".join(word for word in processed_tweet.split() if word not in custom_stopwords)
    processed_tweet = " ".join(Word(word).lemmatize() for word in processed_tweet.split())
    return(processed_tweet)
```

Figure 4. Preprocessing tweets

The twitter data that we access is not stored by our platform. We do not store the tweets in a database as the tweets are accessed using the twitter API every time the user wants to see the sentiment of a particular stock. All the processing and sentiment analysis happens in real time so as to provide the user with the most up-to-date information.

## IV. METHODS

The entire project can be divided into three important segments:

- Stock Recommendation
- Analysis methods for user
- Trading and managing portfolio

### A. Stock Recommendation

The stock recommendation segment is built using the stock price data that we fetch and store in our MySQL database. To recommend a particular stock, the stock must return a true value for all the technical indicators mentioned below namely

- Golden cross decision
- Moving average convergence and divergence decision
- Relative strength index

Even if one of the indicators returns a false value then that particular stock won't be a good buy for the user and hence, won't be recommended to the user. We tried recommending a stock to the user based on user preferences like risk, amount to invest, duration of investment, etc. For this we had also experimented with and developed multiple stock price forecasting models and many machine-learning models that incorporate inputs taken from the user to provide a personalized stock recommendation. However, this does not guarantee a good stock recommendation and a personalized stock recommendation might not mean recommending the stocks that provide the best monetary returns in the long run. Such recommendations would also not take into account factors like the current market trends, market recession etc.

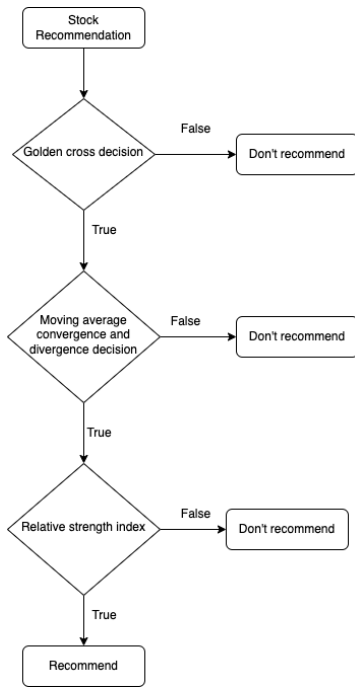


Figure 5. Stock recommendation logic

the long run. Such recommendations would also not take into account factors like the current market trends, market recession etc. We can have situations where a stock might follow everything that a user prefers but still not give the monetary returns intended. This may lead to a bad recommendation to the user and in turn make the user lose money. Hence, this may not be of much use for a person using our platform. Therefore, using industry standard technical indicators like Golden cross decision, Moving average convergence and divergence decision, Relative strength index to recommend stocks to a user is a more robust approach.

## B. Analysis methods for user

Now, once the good stocks to buy have been recommended by our algorithm, we provide the users with multiple tools using which they can perform an in-depth analysis of the stock and then only proceed to make a trading decision about the stock. The various analysis methods that we provide to the user are shown below

- Line chart
- Candle sticks
- Twitter sentiment
- Recent news/headlines
- Forecasting

Using these analytics mechanisms, a user can understand a particular stock behavior, perform an in-depth analysis and then decide whether to buy or sell a stock

### a. Line chart

A line chart is a graph that is used to connect data points using a line. It is used to represent the performance of a particular stock over a span of time. The x axis represents the day/time and the y axis represent the price of a particular stock on that particular day/time. We have used this line chart so that a user can understand the trend of a particular stock over a long period of time.

Usually, in data visualization, line charts are used to indicate the trend of a particular object. Similarly, line charts illustrate the trend of that specific chart. In the stock market, a line chart is one of the basic visualizing components to follow the pattern of a stock price. A stock is determined by four different prices - open price, close price, high price and low price. Traders still consider line charts as one of the best charts to analyze the stock market because it helps the trader to not get distracted by the noises in the chart. A line chart is also the best way to illustrate the support and resistance level of a particular stock in a period of time.



Figure 6. Line chart on the UI

### b. Candle stick

A candle stick is a highly useful graph used for stock market analysis. It is used by investors/traders to gain key insights of a particular stock by analyzing just one graph. Just by one of these graphs, a user can understand the opening price, closing price, day's high, day's low for that particular stock. This provides a very good visualization to the user and helps them perform an in-depth analysis of a stock. The graphs that we have incorporated in our application are interactive i.e., they the user can perform multiple

operations like zoom in, zoom out, change the time scale of the graph etc.



Figure 7. Candle Stick chart on the UI

#### c. Recent news/headlines

While analyzing a particular stock, a user might also want to check the top headlines for a particular stock. While searching on google, the user might come across very different reliable and unreliable news sources. This method does not provide a consolidated platform to the user to review the latest headlines/news. Therefore, to mitigate this we have built a system that scrapes the web for the top financial news channels/blogs and provides the relevant news about a particular stock.

We perform web scraping on the blog/headline page of popular blogs like ‘Stockwits’ and look for the headlines that mention a stock that the user is interested in. Once this is done, we display the most recent news for that stock on the UI to help the user perform their analysis.

We tried scraping the web for all the news about a particular stock on Google but it wasn’t that helpful for the user as it displayed a lot of redundant and repetitive news/headlines about a particular stock. This might confuse the user even further and not assist them in their analysis. Therefore, web scraping a reliable and trusted blog/news channel can prove to be more helpful for the users as the news/headlines are more cohesive, concise and relevant.

#### d. Twitter sentiment

Twitter is a social media application where most of the companies announce their important decision so that the common people can know about it. It can also form a good place to know the feedback received for a particular decision made by an organization. Since most people invest in stocks using their own judgement which is heavily based on human emotions, stocks are affected by how people feel about a company. So, twitter can act as an important means to understand how the stock price of a particular stock might get affected.

To leverage this aspect of Twitter, we have employed a twitter sentiment analysis system that streams the top 2000 tweets of a particular stock using the Twitter’s developer API. Stores these streamed tweets and then performs a sentimental analysis on these tweets. This sentimental analysis can help a user perform their in-depth analysis. The more negative public sentiment of a stock the more user can expect the price of a stock to fall and vice versa.

Twitter Sentiment

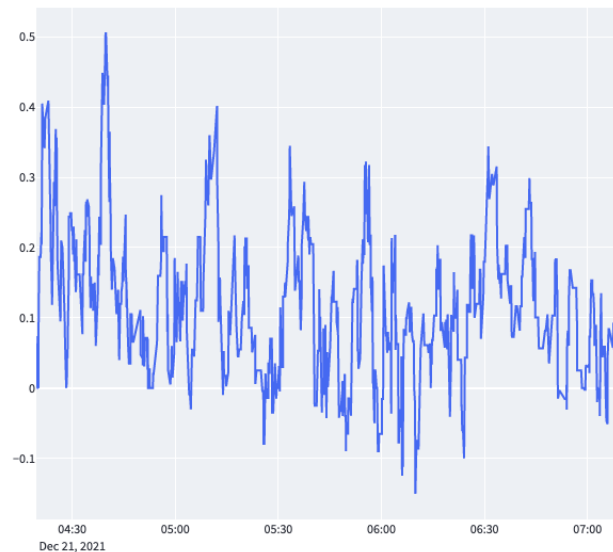


Figure 8. Twitter Sentiment chart on the UI

As you can see from the above chart, the twitter sentiment is mostly  $> 0$ . So, the inference that can be drawn is that the stock price may increase in near future.



### e. Forecasting

Forecasting is an important aspect for an investor. A user might want to see how much returns does he/she get after investing for a particular period of time. This forms a very critical part of user's analysis as the user is investing in a stock for some returns. If a particular stock meets the user's investment returns then the user will be highly interested in that particular stock. If a stock's returns do not match the user's expectation, then the user might not be interested in the stock.

To employ the forecasting functionality for the user, we forecast a particular stock price using FB prophet. We provide the user with the flexibility to choose the number of years he/she want the prediction for. This forecasting gives a robust forecasting price for a particular stock and an upper and lower bound on these forecasting prices. These forecasting prices can help a user better understand the stock behavior in future and then make a wise decision about the stock.

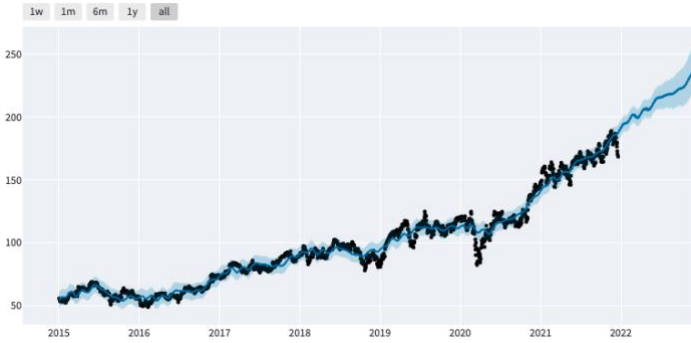


Figure 9. Forecasting chart on the UI

Initially, we tried to implement forecasting using sklearn's linear regression. The forecasting results were not as good as using FB prophet. Hence, we decided to go ahead with FB prophet as along with the future prices FB prophet also provides an upper and lower bound to the future stock prices which can help a user understand the amount of risk, he/she is taking.

### C. Trading and managing portfolio

Now after a user has performed their in-depth analysis and has been convinced that the particular stock is suitable for them, we provide the user with options to buy and sell that stock. With these options the user can add that particular stock to their portfolio or remove it from their existing portfolio. Additionally, if the user is not yet convinced that a particular stock is suitable for them or no, we provide an option to the user to add that particular stock to the user's watchlist so that the user can perform an even more in-depth analysis of the stock.

Along with this, we also provide a functionality where a user can track their current holdings in order to give a bigger picture of their financial health. Therefore, this acts as a very comprehensive all-in-one platform for a person to become a good investor by becoming more financially aware and in-turn make good financial decisions.

## V. EXPERIMENTS

### A. Forecasting

Initially, for forecasting we employed the LSTM cell to make our forecasting model robust. The LSTM cells employ the concept of long short-term memory that incorporates historical data while predicting the output for a new instance. Due to this our new output is dependent on data from several past data points. This method gave good results in terms of forecasting the future prices of a particular stock. However, the LSTM model lacked ease-of-use, low training time and in providing an upper and lower bound for the forecasted price. Therefore, we used FB prophet for the forecasting of our stock prices. FB prophet is an open-source tool used for forecasting time series data and has been developed by Facebook. It is a very easy to use forecasting model that can be used to forecast any time series data easily and quickly. In addition to this, it also provides a method to assess the upper and lower bounds for a particular forecasted time series model. In terms of stock price prediction, this can be very helpful as it can help understand the maximum and minimum gains a person might get after investing the amount in a stock for a particular amount of time

### B. Recommendations

Initially for recommendations, we were taking preference from the user and then suggesting a particular investment option based on the preference. However, this method is not a robust method in order to help the user benefit financially. A case might arise where a particular investment might follow all of the user's preference and still not benefit the user financially. To counter this, we introduced industry standard indicators like golden cross decision, moving average convergence and divergence, relative strength index indicators to recommend an investment to the user with more industry level credibility. These indicators being accepted industry-wide have a greater chance to make the user benefit financially.

Additionally, we tried to give a buying decision to the user based on a positive output of just one of these indicators. However, this didn't turn out to be quite useful for the end user as such a condition returned a huge list of the recommend stocks which could have been difficult for the user to keep a track off. Therefore, we decided to recommend only those stock that return a true value for all these indicators thereby, proving that these are the highly

recommend stocks fundamentally and also returning a very concise list of stocks to the user.

### C. User Interface

Initially, we tried building the User Interface using Django and Python. However, running Machine Learning models with Django and Python seemed to be quite bit tedious. Therefore, we then built our ‘User Interface’ using ‘Streamlit’ and Python as it is easy to use and provides greater compatibility with the Machine Learning models built in Python.

## VI. SYSTEM OVERVIEW

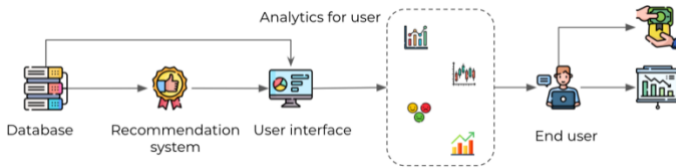


Figure 10. Block diagram of the built system

The above diagram explains the entire system architecture that we have used for our project.

### A. Database system

We have used the MySQL Database installed in our local system to store the stock price data for the last 5 years of all the S&P 500 stocks. We have used MySQL database as it provides the following benefits

- It is open-source and compatible
- It is fast and reliable
- It is available 24/7
- As data grows, it can be easily scaled to handle large data
- MySQL is highly secure as it offers encryption using Secure Sockets Layer
- It has a very friend user interface

### B. Recommendation system

The recommendation system is built using rule base that uses industry standard technical indicators named below. The logic behind these technical indicators is written in Python programming language.

- Golden cross decision: This gives a buying signal to the user when the 50 day moving average crosses the 200 day moving average

```

def Goldencrossdecision(self, df):
    df['SMA50'] = ta.trend.sma_indicator(df.Close, window = 50)
    df['SMA200'] = ta.trend.sma_indicator(df.Close, window = 200)
    df['Signal'] = np.where(df['SMA50'] > df['SMA200'], True, False)
    df['Decision GC'] = df.Signal.diff()
  
```

Figure 11. Logic for Golden Cross decision

- Moving average convergence and divergence: This gives a buying signal when the MACD line crosses the signal line and is increasing

```

def MACDdecision(self, df):
    df['MACD_diff'] = ta.trend.macd_diff(df.Close)
    df['Decision MACD'] = np.where((df.MACD_diff > 0) & (df.MACD_diff.shift(1) < 0), True, False)
  
```

Figure 12. Logic for MACD Decision

- Relative strength index: This will give a buying signal when the RSI/SMA < 30 % and RSI/SMA > 200 days moving average

```

def RSI_SMAdecision(self, df):
    df['RSI'] = ta.momentum.rsi(df.Close, window = 6)
    df['SMA200'] = ta.trend.sma_indicator(df.Close, window = 200)
    df['Decision RSI/SMA'] = np.where((df.Close > df.SMA200) & (df.RSI < 30), True, False)
  
```

Figure 13. Logic for RSI decision

### C. User Interface

The user interface of our system has been built using ‘Streamlit’. Streamlit is a very convenient tool to build highly interactive web applications. Streamlit is a very fast, iterative and interactive tool. This user interface acts as a mediator between the user and the ML models, databases, logic and therefore it is very important for the user interface to be intuitive for the user.



Figure 14. User Interface



#### D. Analytics for user

On the UI, the user is offered several analytical methods to perform an in-depth analysis for a stock/investment as explained above. Once the user has performed his/her in-depth analysis of a stock, he/she has an option to make a trade for that particular stock as well.

The forecasting analysis makes use of FB prophet to forecast prices of a particular investment. The Twitter sentiment analysis method makes use of the Twitter Developer's API and tweepy Python library to retrieve tweets from Twitter and then uses the python library NLTK for processing the tweet and evaluating the sentiment out of it.

#### E. How to use the application?

Step 1: Once a user profile for a user has been created, the user can login to our application

Step 2: After logging in, the user can check the status of their current portfolio in the dialog box given present on the UI

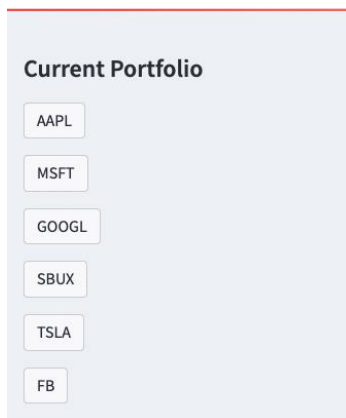


Figure 15. Stock Recommendation view on the UI

Step 3: After seeing the current portfolio, the user has an option to glance through the stock that have been recommended by our algorithm that uses standard state-of-the-art technical indicators to recommend a stock

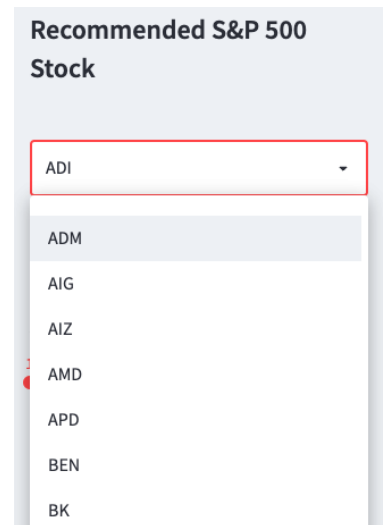


Figure 16. Stock Recommendation view on the UI

Step 4: After checking the recommended stocks, a user might want to perform an in-depth analysis on the stock/investment option

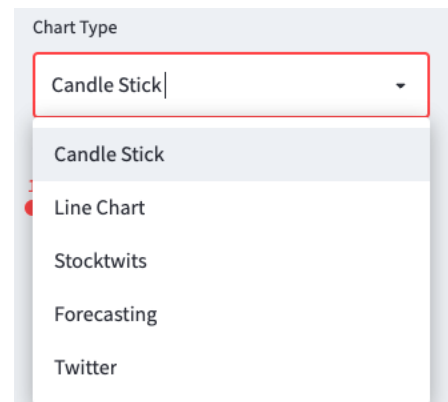


Figure 17. Chart options view on the UI

Step 5: After the user has performed the complete analysis of a particular stock, the user has options to buy/sell/watch a stock.

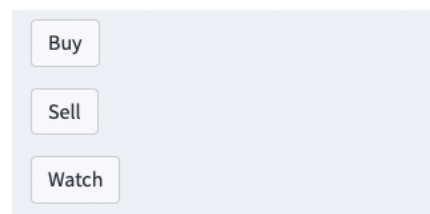


Figure 18. Buy/sell/watch view on the UI

The diagram below depicts the entire user journey of a person that uses our platform



Figure 19. User journey on the platform

## VII. CONCLUSION AND FUTURE SCOPE

In conclusion, we have built a comprehensive investment management platform where a user can watch their portfolio, analyze investments, make trades, and perform every financial activity related to buying/selling a stock. To build such a platform we needed data from multiple disparate resources. This project can act as a great proof of concept for an all-in-one investment management platform. However, to make a production level platform, we will be needing more data points for the investment options, charts, etc, more input from the user to provide a more personalized touch to the product. In sum, it is very beneficial to have such a platform for users where all their financial needs are met and an assistance is provided to them to make a very important decision that might influence their future greatly. A production level platform using this proof-of-concept can prove to be immensely helpful and rewarding to the creators and developers as well.

In terms of future scope for this project, a payment gateway can be integrated to the buy and sell options to help the user make a trade from the platform itself. The platform can be made more personalized by taking the user profile in consideration and then providing personalized recommendations to the user. User preferences like risk, time to invest, investment amount can be taken from the user to provide more personalized options to a

particular user. Also, the machine learning models used can be made more robust to anomalies like a market crash, recession etc. This will help in making the platform react to such situations better and protect the users from any major financial losses

## REFERENCES

- [1] Li, Lei, Yabin Wu, Yihang Ou, Qi Li, Yanquan Zhou, and Daoxin Chen. (2017) "Research on machine learning algorithms and feature extraction for time series." *IEEE 28th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*"
- [2] Hur, Jung & Raj, Manoj & Riyanto, Yohanes. (2006). Finance and Trade: A Cross-country Empirical Analysis on the Impact of Financial Development and Asset Tangibility on International Trade. *World Development*. 34. 1728-1741.10.1016/j.worlddev.2006.02.003.
- [3] Masoud Najeb MH "The impact of stock market performance upon economic growth." *International Journal of Economics and Financial Issues*, 3 (4) (2017), pp. 788-798
- [4] Murkute Amod, Tanuja Sarode "Forecasting market price of stock using artificial neural network." *International Journal of Computer Applications*, 124 (12) (2015), pp. 11-15
- [5] Mehar Vijn, Deeksha Chandola, Vinay Anand Tikkiwal, Arun Kumar, "Stock Closing Price Prediction using Machine Learning Techniques", *Procedia Computer Science*, Volume 167, 2020, Pages 599-606, ISSN 1877-0509,
- [6] Mohankumari C "Analysis of daily stock trend prediction using arima model" *International Journal of Mechanical Engineering and Technology (IJMET)* Volume 10, Issue 01, January 2019, pp. 1772-1792
- [7] Ho A, Vatambeti R, Ravichandran SK (2021) *Bitcoin Price Prediction Using Machine Learning and Artificial Neural Network Model*. *Indian Journal of Science and Technology* 14(27): 2300-2308
- [8] Rout AjitKumar, P.K. Dash, Rajashree Dash, Ranjeeta Bisoi "Forecasting financial time series using a low complexity recurrent neural network and evolutionary learning approach." *Journal of King Saud University-Computer and Information Sciences*, 29 (4) (2017), pp. 536-55

Contributuon: ycj2103: 33.33%, ar4283: 33.33%, sc4921: 33.33%