Stock Catcher

Ali Moazzam
Electrical and Computer Engineering dept.
Michigan Technological University
Houghton, USA
amoazzam@mtu.edu

Parth Mishra
Computer Science dept.
Michigan Technological University
Houghton, USA
parthm@mtu.edu

Aaron Filip
Electrical and Computer Engineering dept.
Michigan Technological University
Houghton, USA
atfilip@mtu.edu

I. Background

Machine learning is all about data; the more data you have, the better your model analyzes and predicts trends. One avenue where people have tried to make their predictions work is the financial markets. Since their advent, people have been attempting to use the best available technology for their time to predict the movement of stocks and financial markets.

In this day and age, with modern technology ingrained in the financial markets, the amount of data they generate each second is tremendous. It is nearly impossible to predict with high credibility the future price of a stock, as the amount of variability and volatility involved is just too large. Moreover, there are human behavioral biases at work in the market continuously. According to the efficient market hypothesis(EMH), the price of stock instantly showcases all the publicly available information. In this project, we try to take on the task of recommending stocks for the user based on the profitability and the risk, and we do it for multiple stocks.

This project will try to achieve its objective by analyzing the closing prices of stocks, and in case the LSTM also the volumes traded, we will predict the profitability of the stocks and reasonable accuracy, the future prices of the stock. For this purpose we are going to use Long short-term memory(LSTM) which is a sub-category of Recurrent Neural Network, as they can mitigate the very important problems which are encountered while training recurrent neural networks; the vanishing gradient problem and the exploding gradient problem and ARIMA which stands for Auto-Regressive Integrated Moving Averages, which is a classic time-series analysis method, once we have the result for the next thirty days from these two models we analyse the risk and profit from the variance analysis from these two models.

II. motivation

We as a group were very intrigued by stock markets and time -series analysis, and we believe that with careful analysis of data, hidden trends can be identified, which can be used for practical purposes. Currently, various financial institutions and big banks like JP Morgan had a technology budget of $11.4 billion. Significant investment houses Bridgewater Associates have been modeling markets since the start, making them a market leader.

In this project, we try to combine our two areas of interest; the stock market and machine learning. There are solutions out there that perform way better, and that too in multiple investment products, but they have teams of engineers behind them, complete funding, computing power, and extremely sophisticated sets of algorithms that provide probabilistic analysis over thousands of factors to provide more risk-free predictions. As mentioned in the introduction, financial markets are highly volatile. But, in this project, we try to do that, with limited resources, but in a reasonable capacity.

III. Existing solutions & Problem review

A. *Stock Analysis by Fundamental Analysis*

As mentioned in the introduction, the stock market has always been a place of intrigue; the most creative human Minds have been applied in predicting securities. The ultimate goal is always to predict the profitability of security with the risk associated with them, for which there is a myriad of available techniques. But the two broad methods are fundamental analysis and technical analysis. Fundamental analysis is based on the financial statements of the company, for example, P&L statement, cash flows, and balance sheets. Here, different ratios are calculated based on profits, debt, revenue, and various other factors; one example of a ratio is the price-earnings ratio which is the price of the shares of the company divided by earnings per share. This ratio can give us an understanding of how undervalued or overvalued the company is. Once these ratios are calculated, they are compared with the proportions of other companies of a similar type, and a peer analysis is done, which finally gives us more confidence to decide whether to buy or sell a stock.

B. *Stock Analysis by Technical Analysis*

The second type of analysis is technical analysis, in which we look at different graphs to analyze trends and patterns and make different indicators that facilitate in predicting the future price of a security. A sub-branch of technical analysis is quantitative trading. With the sheer amount of data being generated in financial markets, the use of quantitative trading is increasing exponentially. Quantitative trading applies traditional technical analysis tools combined with new automation ideas, including machine learning in artificial intelligence approaches. Currently, quantitative trading has become an integral part of decision-making, portfolio management, and risk analysis in various big investment firms. This project defines a quantitative trading strategy based on LMST and

ARIMA models, giving us an idea of which stocks to select and give us confidence in the security.

C. Recent Approaches and Bottlenecks

In recent times with modern technology, there are a lot of tools at the disposal of an institutional investor as well as a retail investor, one of which is algorithmic trading. In which trader can define algorithms based on current market conditions and train them on previous data, and then deploy them in the market to book profits. Still, the problem is that the algorithm becomes obsolete as soon as the underlying market conditions change.

Another one of the more recent and game-changing ideas is to use reinforcement learning to deploy agents in the market based on the iRDPG(imitative recurrent policy gradient) model. This approach defines the stock market as a partially observable MDP and uses the iRDPG model to solve this partially observable MDP.

The main objective of the agent is to arrive at an optimal policy, and as the name suggests, during the training, there is a recurrent exploration and exploitation phase following these two processes, the agent gradually converges towards the policy. The model has shown to produce significantly positive results; in other words, a return of 34%, but it still has its shortcomings as the agent can still make mistakes, and when these mistakes are made with massive amounts of capital, it can have a negative impact on the portfolio.

D. Stock Data Volatility and Proposed Outline

To put it in perspective, predicting the stock market is like solving an NP-Complete problem, in other words, there is no foolproof solution to this problem of predicting the market, the very nature of the market, and the efficient market hypothesis makes sure of it. The most we can do is to make an approximate guess. This is where time-series forecasting comes into play.

Considering the scale of the project and the resources available, it would not have been feasible to construct and deploy an iRDPG model, but there are other, although not as creative but simpler and reasonably efficient models to get information from this time series data. Recurrent neural networks, or RNNs, but there is a problem with RNNs, the exploding gradient problem, and the vanishing gradient problem. A solution to these problems is addressed by an extension of RNN called Long-Short Term Memory or the LSTM model. ARIMA has been a time tested approach to analyze time-series data which has also been used in the stock market for a long time. Both of these models and their workings will be discussed in detail in further sections. Finally, risk and profitability analysis would be performed on the data produced by these models, and the output would be the resulting stocks based on the risk and profit preferences of the user.

IV. DESCRIPTION OF MACHINE LEARNING APPROACHES

Stock data sets include data that is recorded over sequential time frames, which is classified as time series data. Due to our investigation over machine learning methods which can be used for time series data sets, we wind up with two methods: *LSTM* and *ARIMA*. The goal of these methods is to anticipate the next step of the sequence.

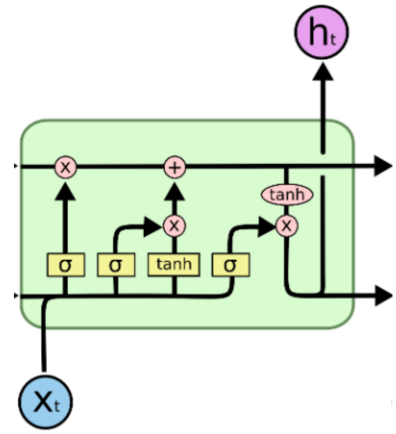

Fig. 1. Example LSTM module [1]

A. LSTM

We know RNNs are great for time-series prediction but, there is a problem with RNNs, the exploding gradient problem, and the vanishing gradient problem. Considering that we have a deep RNN, the inputs supplied to the different nodes of RNN produce some gradient; if the gradient is less than one, then in the future, it can vanish completely, whereas if the gradient is greater than one, it can explode. In both cases, the outcome accuracy of the model decreases, and the training time increases. A solution to these problems is addressed by an extension of RNN called Long-Short Term Memory or the LSTM model.

LSTM addresses the vanishing and exploding gradient problem by introducing four neural network layers in each module, As seen in the fig 2.

Each node introduces three different gates:

- **Forget gate:** This gate decides which information to throw and how much to throw it takes into account h_{t-1} and x_t and finally returns a value between 0 and 1 where 0 is 'remove all' and 1 is 'keep all'.

$$f_t = \sigma(W_f.[h_{t-1}, x_t] + b_f) \tag{1}$$

- **Input gate:** This works in two parts, and the first sigmoid layer decides which values should be updated; this is done by the sigmoid layer; second, the $tanh$ function creates a new vector and then combines it with the output values from the sigmoid layer and adds to the cell state.

$$i_t = \sigma(W_i.[h_{t-1}, x_t] + b_i) \tag{2}$$

$$\widetilde{C} = tanh(W_C.[h_{t-1}, x_t] + b_C) \tag{3}$$

- **Output Gate:** After passing through the forget and input gate, the output of these decides the cell state. The

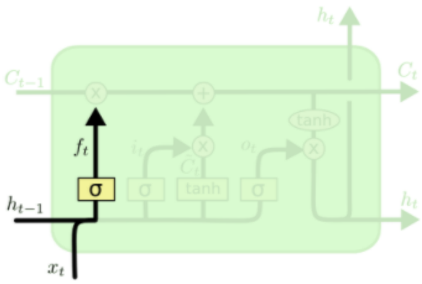

Fig. 2. Forget gate of the LSTM module [1]

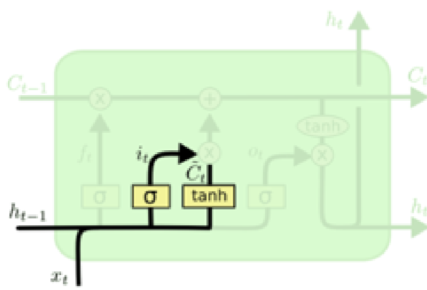

Fig. 3. Input gate of the LSTM module [1]

previous history is passed through another sigmoid layer which is then multiplied by the output from the previous two gates after passing it through another function to produce the next history.

$$o_t = \sigma(W_o.[h_{t-1}, x_t] + b_o) \tag{4}$$

$$h_t = o_t * tanh(C_t) \tag{5}$$

B. ARIMA

AIRMA has been proposed to forecast next step of the chain. *Auto Regressive Integrated Moving Average* (ARIMA) can foresee the next stride from previous sequences. This model has two major part:

- **Auto Regressive (AR):** It stands for a linear regression model, which is used as the fundamental of the mode. Also, it is shown as *"p"*. It shows the number of lags Y as shown in equation (6) [2] which is used to anticipate the basic of the model.

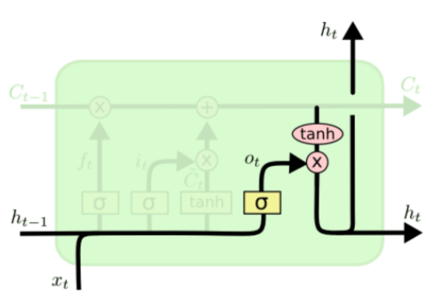

Fig. 4. Output gate of the LSTM module [1]

$$Y_t = \alpha + \beta_1 Y_{t-1} + \beta_2 Y_{t-2} + ... + \beta_p Y_{t-p} + \epsilon_1 \tag{6}$$

- **Integrated:** Takes the difference of lags in a static process. Also, *"d"* has been defined for the integrated part, which stands for the difference value between the current step and the previous step. By finding this step, it tries to find the basic of the hole trend.
- **Moving Average (MA):** MA is the difference between the current step and the next step, and also it is known as *"q"*. MA can be gained by equation (7) [2] as you can see below:

$$Y_t = \alpha + \epsilon_t + \phi_1 \epsilon_{t-1} + \phi_2 \epsilon_{t-2} + ... + \phi_q \epsilon_{t-q} \tag{7}$$

Equation (7) is difference between Equation (8) [2] and Equation (9) [2].

$$Y_t = \beta_1 Y_{t-1} + \beta_2 Y_{t-2} + ... + \beta_o Y_{t-o} + \epsilon_1 \tag{8}$$

$$Y_{t-1} = \beta_1 Y_{t-2} + \beta_2 Y_{t-3} + ... + \beta_o Y_{t-o-1} + \epsilon_{t-1} \tag{9}$$

All in all, ARIMA is a mixture of Autoregressive and moving average model which means ARIMA is the combination of "p" linear lags of Y with "q" linear lags of predicted errors as it shows in equation(10) [2]:

$$\begin{aligned} Y_t = \alpha + \beta_1 Y_{t-1} + \beta_2 Y_{t-2} + ... + \beta_p Y_{t-p} \epsilon_t \\ + \phi_1 \epsilon_{t-1} + \phi_2 \epsilon_{t-2} + ... + \phi_q \epsilon_{t-q} \end{aligned} \tag{10}$$

In ARIMA, the three parameters mentioned above need to be defined. P for autoregressive, d for the integrated part, and q for moving average, and ARIMA can be noted as ARIMA(P,d,q). Beside, ARIMA, AR(p,0,0) , I(0,d,0), MA(0,0,q), ARI(p,d,0) and, ARMA(p,0,q) are exist and provide more flexibility in terms of configuration.

For the purpose of our model, 30 different combination parameters have been investigated, and we wound up with ARIMA(3,2,1) as the best configuration for our model.

V. Risk and Profitability

Risk is very much defined by the individual, group, or company with stakes in a matter at hand. Or, in other words, they have something to lose. How much they are willing to lose becomes the risk factor they are comfortable with in conjunction with the probabilistic odds of the situation. Therefore, while there are many general strategies and general guidelines, risk is very much a freely defined principle that is unique.

Even the general forms of risk take into consideration many factors that we do not focus on in this trend based prediction. Due to the stock market being highly volatile, even the best of trends and prediction techniques can't accurately predict the market at all times. This is where risk comes into concern. In this paper, we predict a direction of a given stock and a general approach to risk. It is up to the user to take this generalization and form their own definition of risk before delving into the stock market.

A. Calculation of Risk

Two models, ARIMA and LSTM, run independently, generating an array of predicted stock prices in relation to the original closing price the day before the predictions began.

Risk is calculated by first subtracting the initial price from the final predicted price to get a predicted profit margin from each stock from each algorithm. A negative or positive trend denoted by a single binary point saves the decreasing or increasing trend of the stock.

All data points are normalized in relation to one another by converting it to a percentage increase or decrease.

$$PercentChange = (X_{final} - X_{initial})/X_{initial} \tag{11}$$

In this process, the trend produced from each model is compared, and the variance is calculated from which. This study attempts to predict stock prices primarily as an increasing or decreasing trend, with price specific volatility given in the form of a percentage in comparison to two different models. Therefore, the definition of risk will be more straightforward, more conservative, and more tuned to our application to allow the end user to develop their own definition further.

Risk is separated into three different categories:

1) HIGH RISK
2) MEDIUM RISK
3) LOW RISK

Risk in this model is highly subjective to the models correlating with one another. Typically calculating a higher variance between two models in stocks or high variance in stock change of price, volume is an indicator of high volatility. This high volatility directly translates into a more risky decision. The profit can be immense, but so can the loss.

Low risk is thus defined as anything in general with a positive trend predicted by both models. This is an attempt to minimize loss as the user has indicated as low risk as possible.

Medium risk, however, starts introducing some of the volatility back in. There are many strategies to investing in a declining market. Short selling and "buying the dip" are but a few. Each comes with an increased risk of losing money due to volatility and more potential for an incorrect prediction. However, if the variance is too high, then the stock will be classified as a high risk stock.

High risk is defined by our stock catcher as anywhere that both model's produce is diverging higher variance results. Thus, due to the assumed increase in volatility predicted by this, the stock is classified as a high risk, leaving the ultimate decision up to the end user.

Each type of risk is a level built upon the last. Thus, High Risk includes Medium and Low Risk, and Medium includes Low Risk. The idea behind your level of risk specification is not to see everything just of that level, but instead to see all your options with an upper bound as your defined risk threshold of acceptance.

B. Addressing Diverging Models

In the case of two models diverging, it is explained above that it is automatically calculated as a high risk stock. This is a particular case in which we label our correlation as 0.5, halfway between increasing and decreasing. To simulate the volatility of the market, we add a randomly generated float value from -0.5 to 0.5. In theory, this could raise the correlation to 1 or lower to 0, but the odds of such are highly unlikely.

This aims to achieve some prediction value with the inclusion of a bias. With diverging results, it is already considered volatile, so we attempt to predict the volatility.

VI. DATASET DESCRIPTION

S&P 500 stock data has been used for the purpose of this project. Since the goal of our project was to find the best mode for stock analysis, the first option was available stock data set on *Kaggle* website. The website includes many different popular stock data sets, but between all of them, *S&P 500 stock data* has been selected. This data set includes five parameters of 500 companies for five years. *Open Price*, *Close price* , *day highest price*, *day lowest price* and, *volume* of each stock have been considered in this data set. By considering companies name and dates, this data set contains all the information needed for stock evaluation problems.

By the investigation has been done over the data set, it was shown, the data points for all companies are not identical. Furthermore, since stocks are independent, We had to run the machine learning models over each stock individually, so as a result, we wind up with high computational time. So, for these reasons, we decided to choose a subset from the data set and implement that subset in our model. The subset was included first, 25 stocks which also have the same data points. By selecting this subset, all machine learning processes have been done over stocks with the same conditions and data points, and also we resolved the computational problem.

Open Price, *day highest price*, *day lowest price* and, *volume* parameters provide valuable information about each stock, although, the final price of the stock (*close price*) is the most important information and it shows the actual value of the stock. In our inspection, Only *Close price* has been considered.

The data set has the information of each stock for five years, from 2013 till 2018. Since we need a testing set as well, We assumed for years and 11 months as the training set and one month as the test set. The training set has been used to train the model, and the testing set evaluates the price predicting accuracy. For the reason that each stock is independent, individual train and test sets have been made for the learning process.

VII. OVERVIEW ARCHITECTURE

A. Preprocessing procedure

Data preprocessing is an integral of any machine learning project; since we have two different models, we will be doing preprocessing for each of them separately, the resulting output for each of these models will be a CSV file which can be fed for risk analysis, the data frames will have start and end predictions for 25 stocks and will have a 2X25 shape. For each stock a common train/test dataset will be used; the train set, in this case, will include all the close prices(volumes will also be included in case of the LSTM) except for the last month(Last 21 values), the test set will consist of the last 21 values. Below we will describe the preprocessing methods used for the different models:

- **LSTM:** We will first normalize the data in a range of 0 to 1, then since LSTM is recurrent in nature, we will have to create batches of data, and for each batch, we will take into account the previous 12 months, so each element in the X_train is a list of 12 values which represents the last 12 months, and the 13th-month value is stored in the y_train.
- **ARIMA:** Compare to LSTM, ARIMA follows a different approach for the prepossessing step. Since ARIMA has less computational time than LSTM, we will take into account the last four years and 11 months and predict the following month. This model has been tasted over Log values of the training set, and as a result, the output was pretty much the same as a standard training set, so for our purpose, a normal four years and 11 months data points have been processed.
- **Risk** Calculation of risk and the recommendation of stocks is initialized from ingesting the output of the LSTM and ARIMA models in the form of a csv. Both model generation processes are independent of the final calculation as there is no user entered data directly influencing the stock predicting engine. The process of evaluating risk and generating final user content is further defined in the section **Risk and Profitability**.

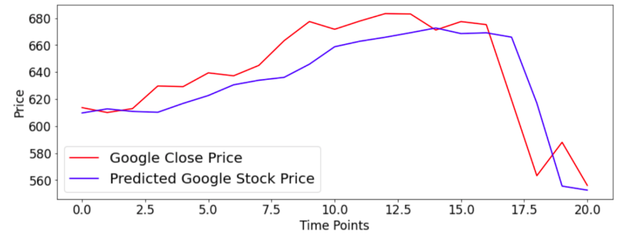

Fig. 5. LSTM predictions for Google. The X-axis shows 21 data points which represent the last month. Since LSTM takes volume, it changed the Y-axis scale, but apparently, it anticipates the sequence very well..

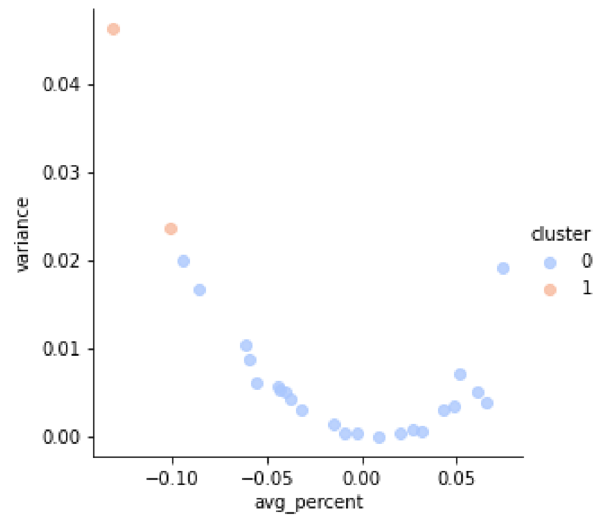

Fig. 6. Example K-Means result on 25 stocks. A Cluster Label of 1 represents a higher risk stock, particularly due to higher variance as shown.

VIII. Experimental Results

We used Google Colab to implement our model trained on the first 25 stocks. This could easily be scaled up to include processing of all stocks as described in the *DATASET Description* section.

A. *LSTM*

We have earlier discussed how we preprocess data for the LSTM. Now, we will discuss the model specifics. For implementing the LSTM, we will be using TensorFlow. For each stock, we follow a similar pipeline; first, we preprocess the train and test set as mentioned in the preprocess section, then, we initialize our model; the model will have five layers: 4 LSTM layers containing 60, 60, 80, 120 nodes respectively, and a Dense layer which would be giving us a final single value output. We will use ReLu as the activation function, mean squared error as the loss, and Adam as our optimizer; we now compile this model and predict the values. Fig x shows the example of the plot for Google generated by this pipeline, and all other 25 stocks are generated in a similar fashion.

B. *ARIMA*

The ARIMA model has been discussed earlier in this report. For the ARIMA implementation approach, First, the stock has been selected, and it has been run by the ARIMA model. To achieve the best configuration of the ARIMA machine, 30 different configurations of ARIMA have been considered, and the best one has been elected. For a visual illustration, Google stock, testing set, and predicted values were compared, as shown in figure 7.

C. *K-Means*

The K-Means algorithm uses a clustering value of 2 to generate the centroid's and organize the data. Since we are looking for correlations in the data and potentially higher risk results, the algorithm is left to a more automatic or default configuration runtime.

It's sole purpose is to help with the classification of high risk stock which is usually due to high variance, without manually specifying a value to run off. Therefore, the determination of variance is more relative to the rest of the stocks than a hard-coded value.

After the successful prediction, the ARIMA model, has been run over the 25 selected stocks in the same path, and the output data has been passed through the recommendation system.

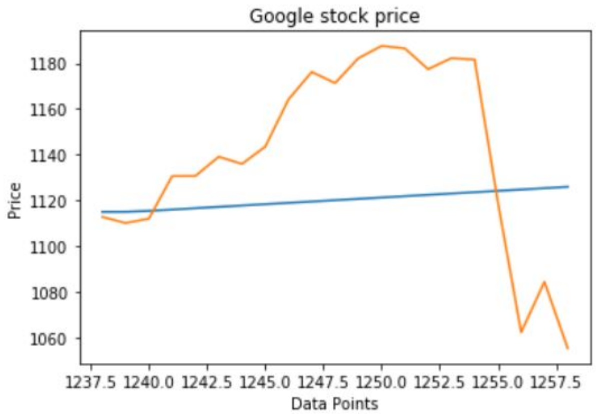

Fig. 7. The orange line stands for the testing set of data for last month, and the blue line stands for The predicted trend from the ARIMA model. The X-axis shows 21 data points which represent the previous month, and the Y-axis is the google stock price for the specified period.

D. Recommendation system

There are three options available in effort to help customize the list of recommended stock to a user.

1) Max Price of Stock
2) Number of Stocks to Recommend
3) Level of Risk

By implementing these, the user has the ability to set a filter on the max price of stock and specify the level of risk they are comfortable with. The return recommended stocks are filtered to therefore be within the user's expected price range and which are appropriate to how much risk they are willing to take.

All of this is sorted by average profit and returned in a number specified by the user. Both medium and high risk options will show two data sets; one with stocks predicted to decrease, and one with stocks predicted to increase.

Low risk filtered to 3 results is shown in Figure 8. Due to our definition of low risk, only those values in which both LSTM and ARIMA models have predicted an increase will be shown. Any stocks that the K-Means algorithm has detected and grouped in a higher variance will be excluded.

Medium risk as represented in Figure 9 and Figure 10 builds upon the low risk but allows for negative predictions from the models. It once again will show the user defined number of stocks but will show two sets. One for negative and one for positive sorted on max increasing or decreasing profit.

High risk as represented in Figure 11 and Figure 12 builds upon the low and medium risk. Unlike the first two, high risk has no restrictions and no bounds on the variance, allowing for highly volatile predictions to be represented. Any stock in which the LSTM and ARIMA models have predicted different trends will automatically be classified as a high risk stock.

The returning data from the model is broken into six sections. The **avg_adjprofit** is directly used in the calculation of the **avg_percent** which shows the percent change for a single stock as averaged by the two models. A negative value in both of these fields represents the stock as predicted to decrease in value over the next month.

LOW RISK

	avg_adjprofit	avg_percent	variance	trend	cluster	Name
9	0.456861	0.042943	0.003073	1.0	0.0	ADM
5	0.767160	0.026477	0.000754	1.0	0.0	ABT
11	0.862313	0.008762	0.000007	1.0	0.0	ADSK

Fig. 8. Example Low Risk Result. Top 3 shown under 100 dollars/stock

MEDIUM RISK (Negative Predicted Trend)

	avg_adjprofit	avg_percent	variance	trend	cluster	Name
21	-3.882141	-0.061500	0.010478	0.0	0.0	AIZ
14	-2.405507	-0.085826	0.016822	0.0	0.0	AEP
10	-2.083747	-0.037622	0.004214	0.0	0.0	ADP

Fig. 9. Example Medium Risk Result. Top 3 Predicted Negative shown under 100 dollars/stock and is indicated by a cluster value of 0

This overall decreasing or increasing trend is shown in the **trend** as a 1 or 0, respectively. The variance, however, is calculated directly from the model's different predicted responses to the same stock. A higher variability will imply a more significant risk and a wider margin in which the models have disagreed with one another. Using the variance, the cluster column is generated by a K-Means 2 Cluster identifying potential points of higher variance and relegating it to a higher risk category. A denotation of 1 symbolizes higher risk/variance, while 0 would indicate a normal stock prediction.

To tie everything together, we have the name of the stock, but we do not include the price. Even though we filter based on the estimated cost of the stock before prediction, this is simply a stock recommendation system. Future work would be needed to provide an accurate, up-to-date value of the current price. Achieving this would require working and gaining access to a trading center API or exchange.

IX. CONTRIBUTIONS OF EACH MEMBER

All members have contributed to the majority stuff of the project. Everyone was involved in the brainstorming session, and we did pair programming and code review to maintain the

MEDIUM RISK (Positive Predicted Trend)

	avg_adjprofit	avg_percent	variance	trend	cluster	Name
3	4.297630	0.061139	0.005144	1.0	0.0	ABBV
22	1.475482	0.048463	0.003488	1.0	0.0	AJG
17	0.970366	0.019736	0.000326	1.0	0.0	AFL

Fig. 10. Example Medium Risk Result. Top 3 Predicted Positive shown under 100 dollars/stock. Not Accepting of High Variance and is indicated by a cluster value of 0

HIGH RISK (Negative Predicted Trend)

	avg_adjprofit	avg_percent	variance	trend	cluster	Name
21	-3.882141	-0.061500	0.010478	0.0	0.0	AIZ
14	-2.405507	-0.085826	0.016822	0.0	0.0	AEP
10	-2.083747	-0.037622	0.004214	0.0	0.0	ADP

Fig. 11. Example High Risk Result. Top 3 Predicted Negative shown under 100 dollars/stock. Accepting of High Variance indicated by a cluster value of 1

HIGH RISK (Positive Predicted Trend)

	avg_adjprofit	avg_percent	variance	trend	cluster	Name
3	4.297630	0.061139	0.005144	1.000000	0.0	ABBV
22	1.475482	0.048463	0.003488	1.000000	0.0	AJG
2	1.391904	0.050982	0.007123	0.312995	0.0	AAP

Fig. 12. Example High Risk Result. Top 3 Predicted Positive shown under 100 dollars/stock. Accepting of High Variance indicated by a cluster value of 1

correctness and the quality of the code. Also, All the members worked on all the documents and the reports. The member major focuses were:

- **Ali:** ARIMA Modelling
- **Parth:** LSTM Modelling
- **Aaron:** Risk and Profitability Analysis

X. Conclusion and Future Work

Our model simulates decently well for what is available in the dataset and considering a more straightforward approach to the stock market. Due to the immense complexity of the stock market, additional research would have to be done on trends, behaviors, concepts on options, and other types and factors in the trading system before beginning to generate a more complex model. Ideal factors and research on different major outside influences would have to be taken into account to have a realistic idea of the prediction of stock behavior.

Increasing options to include recommending whether to buy, sell, or hold stock would be beneficial to the end user. Risk would have to be re-defined as the model gets more complex as there would be more scenarios to cover.

References

[1] "Understanding LSTM Networks," Understanding LSTM Networks – colah's blog. [Online]. Available: https://colah.github.io/posts/2015-08-Understanding-LSTMs/. [Accessed: 29-Apr-2021]

[2] S. Prabhakaran, "ARIMA Model - Complete Guide to Time Series Forecasting in Python," ML+, 22-Mar-2021. [Online]. Available: https://www.machinelearningplus.com/time-series/arima-model-time-series-forecasting-python/. [Accessed: 29-Apr-2021].