# A STOCK FORECASTING APP USING MACHINE LEARNING AND BIG DATA ANALYSIS TO HELP GUIDE DECISION MAKING WHEN INVESTING IN THE STOCK MARKET

Alex Xie[1] and Yu Sun[2]

[1]Chadwick School, 26800 S Academy Dr, Palos Verdes Peninsula, CA 90274
[2]California State Polytechnic University,
Pomona, CA, 91768, Irvine, CA 92620

***ABSTRACT***

*Currently, there is increasing participation in investment, especially in the stock market, as it appeals to more average citizens. One common roadblock these individuals receive is the lack of information about the economy, politics, regulation, etc., which could all affect the stock market. This app addresses this problem by collecting mass information from third-party social media. Information from social media platforms has its advantage mainly due to the citizen involvement and expertise some users may resemble. Pulling large amounts of data from these social media users avoids bias and establishes reliable sourcing. Using this information as a predictor, the app computes data and effectively predicts the performance of the stock for the next three days. By doing this, users of this app could easily get informed through instant quantitative prediction instead of having to read all over social media. This app also indirectly manages people's wealth because it allows users to make smarter decisions, thus increasing their money potential. In certain areas, this app is able to perform the same duty as a licensed wealth manager.*

***KEYWORDS***

*Stock, Asset, Investment, Exchange.*

## 1. INTRODUCTION

The benefits of investment are enormous to both the investor and the company. Business investment often helps small businesses to grow and leads them out of budget hardship. Money that a company receives stimulates the expansion of projects and the efficiency of reaching its goals. Attracting investors also means the company is trusted by others and thus will grow in popularity in the eyes of the citizens [1]. On the other hand, the benefits to investors should not be neglected. Investors hope for long-term returns that add to their income. Investing in well performing companies is the quickest way to multiple assets without having to work physically. Besides, investment also outperforms inflation in the economy, as the return rate is usually higher than the rate of inflation [2].

Since more people are willing to give their money for investing, stock trading platforms are innovating to provide the most convenient experiences to customers in order to attract more users. As the stock market is growing and more people become involved in investing in stocks hoping to enlarge their wealth, many barriers start to unveil due to the lack of expertise and

David C. Wyld et al. (Eds): AIFZ, SIGPRO, NWCOM, CLSB, MLNLP, DTMN, BDIOT, CSITY - 2022
pp. 163-175, 2022. CS & IT - CSCP 2022 DOI: 10.5121/csit.2022.121613

background knowledge of certain companies' performance. There are many unpredicted fluctuations in the stock market. People lose money because they were not able to obtain or be alerted quickly enough to act [3]. Additionally, the reason why more average citizens are transitioning into digital investments is that it appears to be the easiest method to earn money. This illusion lures people into spending their money on investments. In other words, unprofessional traders bet their money and underestimate the treacherous market.

First off, in terms of trading platforms, there are numerous apps and websites where users can legally exchange stocks in the U.S. Inside of those stock trading platforms, the developer usually implements different forums where users and professionals can discuss and share trends and performance of certain stocks. There are also new feed sections where users can be informed of the latest news regarding the current market in the economy. These are ways that help advise users on their trading decisions in order to improve their user experience [4]. However, this method does have its flaw. First of all, time is a huge constraint for users reading news and forums. Spending a long time reading these suggestions and information does not provide users with the quickest method of obtaining knowledge. Plus, there are many forums out there with different descriptors of the companies, choosing the right forum can be super time-consuming. On the contrary, it ruins many users' experience in investing. The second flaw in these forums and news feeds is the lack of comprehension. Keep in mind that these users are often inexperienced with many terminologies that more knowledgeable people use when trading. Even if these average users devote endless hours to reading and researching, they might not be able to understand the general meaning of these texts. Another serious problem with these forums is that it is public to everyone, and the information everyone is sharing could be either fraud or unreliable [5]. Users often only experience a small pool of information from a designated source. The problem with this is that since people are not accessing a wide range of sources, they might often get misled by these biased and subjective voices. There are many cases where institutional investors did a rug-pull, causing retail investors to lose money [6]. All of these are serious problems in the trading system right now, and this stock forecasting app aims to solve these issues.

Our goal in developing this app is to ensure all prospective traders, no matter what prior experience they resemble, are able to be better informed on their investing decision-making. We want to open out the gate and let all information into the hands of the users efficiently and effectively. This is not a stock trading app, it is simply an app where users can track stocks and see predictions of the stock's performance. When we built this app, we considered many features worth adding to improve the functionality of the app. The first and most important feature is pulling social trends from third-party social media and compiling them for analysis. This is a very crucial step of the development because we need to make sure we have access to all information out there in order to make our predictions more inclusive and reliable. The second feature we included comes after compiling the information, which is to use numbers to make predictions of the stock's performance in the next three trading days. Displaying the percentage increase or decrease of the stock to inform people effectively. Last but not least, we have features that focus on user experiences, such as a search to all 6000 stocks, a watch list, and a daily trend that helps pertain to the user's needs.

Determining whether the prediction is accurate was what we mainly focused on while building this app. Since we are using a machine learning model, there needs to be large amounts of data and various trials to formulate a representative machine. Choosing a display model of the graph is also important to our design. Whether to use linear or polynomial expression affected the outcome. We tested these graph algorithms and found out that each model delivers a different result, even though the data and information inputted are identical. We also ran many algorithms when testing our data and prediction. We started off by only implementing a small amount of

stock to run that and check if the system was able to output the data we wanted. I recalled we started with only six stocks in our system. After compiling social trends on these stocks, we put them into the algorithms we used to form the graph and compute the necessary computations. Now, the graph would display the correct numbers regarding the price and social trends of the stock. Training machine learning also took large amounts of data. We would have to determine if the output data matches the current trend and is likely going to display an accurate prediction. After analyzing the graphs and the data, we were able to conclude that the prediction modeled highly accurate predictions. With multiple trials implemented, we were able to begin implementing more stocks into the system for computation and data extraction. Once we had one algorithm working successfully, the successive runs would almost always predict accurately.

The rest of the paper will follow a similar format to most research papers. Section 2 explains the challenges we faced when building our app and testing for the experiment; Section 3 dives deep into the details of the program, including codes and solutions we came up with to resolve the challenges identified in part 2. We will be showing the structure, and the diagram of meaningful tests methods we have used; Section 4 contains the specific experiment we have done, including an evaluation and analysis of the results; Section 5 lists many relevant works that have already been done by someone else regarding stock investing and stock forecasting. This section compares our work to others' work; finally, Section 6 will conclude our study and state any future plans and improvements we may continue to target.

## 2. CHALLENGES

In order to build the project, a few challenges have been identified as follows.

### 2.1. Implementing Large Numbers of Stocks

In order for our app to be as professional as possible, we need to store almost all the stock in the stock market in our database. Unlike those large trading apps, we do not have a huge cloud to store all these stocks compatibly. We had to write the tickers for these stocks manually. We tried some of them with our current system ability, but it turned out that on the user's end, it took forever to load, and it caused a huge lag for the user. Also, if we would successfully implement these 6000 stocks into our backend, it would be extremely hard to display to the users in the frontend. This part of the process took us the longest time in our build, but we were able to minimize the lag by only displaying a small number of stocks in the search bar and displaying the rest when users intentionally search them up using the search bar. We noticed a huge decrease in lag and more fluency on the screen.

### 2.2. Creating a Convenient Frontend

We were mainly concerned with the user's experience when developing this app. The method we used when creating the user frontend was to use Android Studio to program the screen using Dart language. We tried to implement many buttons and clickable features, but some of them turned out to be unresponsive. We found some people to test out the app, and many of them reported that some of the buttons do not work and functions such as watchlist and daily trend do not work. There was also a problem with the back button at some point, which in turn caused users to have to reopen the app to be reset to the home screen. It was mainly due to the problem with our code, so we had to go back to parts of the code and locate where the error is occurring. On top of fixing it, we also had to continuously brainstorm ways to make the app more inclusive. In fact, we are still in the process of improving our UE by adding other useful features.

### 2.3. Stock Performance Prediction

The biggest selling point of our app would probably be the ability to predict the performance of a state and output a percent increase or decrease. This inevitably becomes the biggest challenge in our product. Our program consists of AI machine learning using a modeled representation as a polynomial function graph. We are performing our prediction based on the number of posts on social media about a specific stock. As you might know, participation on social media is very unpredictable every day. Even if we use machine learning to study the day the previous day, it will still be hard to accurately present the data the next day without a huge margin of error. This is something that we would have to dive deeper into in terms of long-term improvement. We need to research and choose the closest related modeling technique for the machine and hope the get the closest result to the actual trend.

## 3. SOLUTION

This app is designed to provide users with a percentage increase or decrease of a specific stock price in the next three days based on the data retrieved from social media. There are three major components that formulate the entire app: the user frontend, the backend database, and the machine learning algorithm. First off, the entire front end was created using Dart in Flutter. Using this software allows us to develop an app accessible to both IOS and Android users [7]. Features include displaying the compiled data to the user in the simplest manner and being able to adapt to changes made in the backend quickly. Another software we used during the build was Google Firebase. Google firebase does a magnificent job at storing large amounts of data and providing analysis of these data [8]. We were able to store data inside the firebase that could later be used to connect to our backend to be computed. Along with Google Firebase, we also used Amazon AWS, which is a cloud platform that provides effective cloud computing solutions [9]. Since we have more than 6000 stocks in our database, we needed an efficient cloud computing software that could deliver the analysis of all the 6000+ stocks we need. Last but not least, the entire backend and machine learning functionality are created using Python. The Python server manages all the data that will be inputted into the app to make predictions about the growth. We have implemented both the linear regression model and polynomial model to represent the prediction that will be displayed correctly.

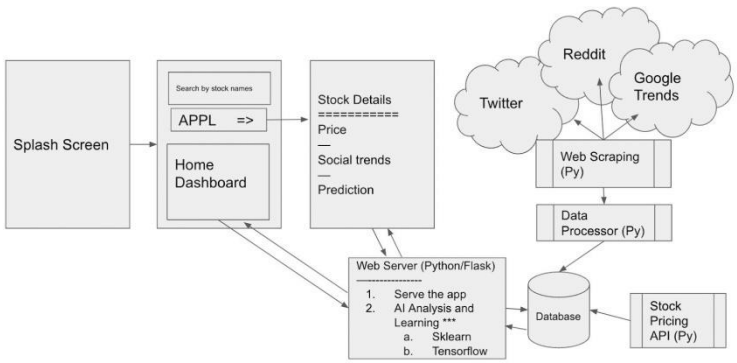


Figure 1. Overview of the solution

1) Web Scraping

    a. Twitter API code + text description
    b. Yahoo finance API code + text description

2) Server/Database

    a. Database
        i. Screenshot
        ii. text to talk about how it works;
        iii. the data structures;

    b. Server
        i. Python Flask code + text description
        ii. AWS and deployment (screenshot) + text description

3) AI/Machine Learning

    a. scikit-learn library
    b. linear regression etc.

4) App

    a. screenshots
    b. find 1-2 important code excerpts
    c. text to talk about how the app works and what the code do

The back end of the app contains the web scraping algorithm, servers/databases, and machine learning models. In order to make an accurate prediction, the machine learning server and database have to be constantly communicating with each other. The AI learns from the data provided by the cloud to output results that predict the future trend. These data all came from web scraping Twitter and Yahoo Finance.

```
    requests.get('https://api.twitter.com/2/tweets/counts/recent?
    query=%23' + stock, headers=headers)
27
28 ▼  if r.status_code == 200:
29      res = r.json()
30
31      global res_stat
32      res_stat = { }
33
34 ▼    for count_record in res['data']:
35        day = count_record['start'].split('T')[0]
36        # print(day, count_record['tweet_count'])
37 ▼      if day in res_stat:
38          res_stat[day] = res_stat[day] +
count_record['tweet_count']
39 ▼      else:
40          res_stat[day] = count_record['tweet_count']
41    return res_stat
42 ▼  else:
43      print("Failed to get the count from Twitter. ")
44      print(r.status_code)
45      return {}
```

Figure 2. Twitter API application code

One of the major components of the design is web scarping. This figure shows the utilization of Twitter API to scrape all the data and tweets and Twitter on stocks. These codes are written so that we are able to examine the tweet count from different days of the week. If the system failed

to scrape the data from Twitter, then a failure message will be displayed so we can fix it accordingly. The reason why this is an essential part of the program is that we need this data to formulate our prediction. Without this data, there is no way we could move forward with our app. The app is centered around social media trends, so scraping data from social media platforms is necessary. Later, this information will connect with our machine learning model to be analyzed even further.

```
def get_stock_info(stock, day):
  msft = yf.Ticker(stock)
  res = {
    'Open' : -99,
    'Close' : -99,
    'High' : -99,
    'Low' : -99
  }
  # get historical market data
  hist = msft.history(start=day, end=get_next_day(day))
  # print(hist) # table
  if len(hist) > 0:
    res['Open'] = hist.iloc[0]['Open']
    res['Close'] = hist.iloc[0]['Close']
    res['High'] = hist.iloc[0]['High']
    res['Low'] = hist.iloc[0]['Low']
  else:
    print("No history is found.")
  return res
```

Figure 3. Yahoo Finance API application code

This figure is the other part of web scraping included in our program. This part is relatively straightforward. We scarped stock information, including the opening and closing price of the stock, as well as the highest and lowest price of the stock during the day. Again, we used Yahoo Finance API to get the information of the stock and scrape relevant information granted with the API code. If no history is found, an error message will be displayed. This part of the algorithm is also crucial in our app because the basic information about the stocks will be presented to the users when they choose one to look at. This information will be stored in Google Firebase and connected straight to the app's front end.

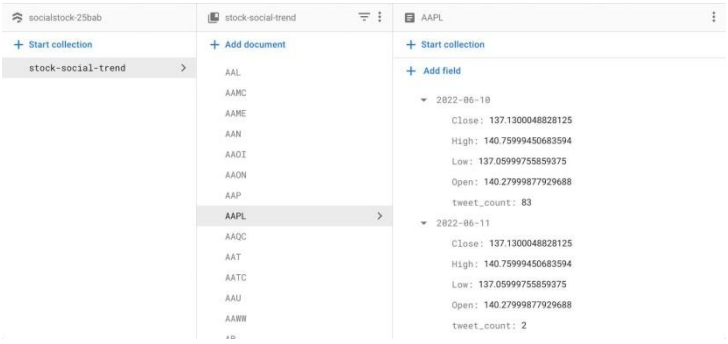


Figure 4. Google Firebase data storage

We used Google Firebase to create bigger storage to store all the trends scaped from social media and financing websites. The Firebase contains a list of all the stocks we will have in our app. Inside those stocks, we have a list of information saved regarding these stocks. We pulled all this information from Yahoo Finance which is a reliable source that displays the basic information about the stocks. Figure 4 is just a screenshot taken from the database in Google Firebase. Information includes opening and closing prices, the highest and lowest prices, and the date when this information was retrieved.

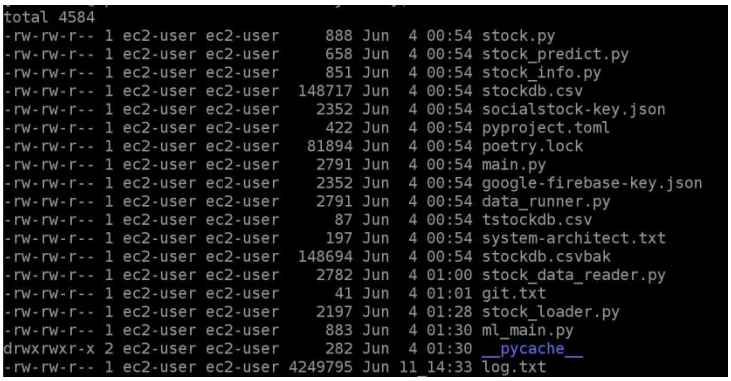


Figure 5. Amazon AWS cloud computing process

On top of the Google Firebase we used for storage, we also used Amazon AWS for cloud computing and analysis. In order to smoothly implement the 6000+ stocks into our app, the information needs to be computed constantly at the backend. We chose not to put all 6000+ stocks directly available to the users in the front to avoid lags, but they do exist and are consistently being computed with AWS. With AWS cloud computing, more stages and analyses of the data can be delivered and be available to the users upon request. Figure 5 is a screenshot of the server taken at a random time, but it shows the data being computed continuously.

```
38 # Linear Model
39 print("\nModel 1")
40 model = linear_model.LinearRegression()
41 model.fit(input_data, output_data)
42 print(model.predict([ [120, 5, 1], [300, 5, 2]]))
43
44 # Polynormial
45 print("\nModel 2")
46 model2 = make_pipeline(PolynomialFeatures(2),
   linear_model.LinearRegression())
47 model2.fit(input_data, output_data)
48 print(model2.predict([ [120, 5, 1], [300, 5, 2]]))
```

Figure 6. Machine learning model code

Here comes the machine learning part of the program, an essential area of focus that outputs the final prediction. Figure 6 shows the code we have written while programming machine learning. We used the scikit-learn library to construct two models that we will be using when making predictions. Both of these models require construction for a line of best fit. For the linear regression model, some values are inputted for the machine to construct a line of best fit in a linear manner. For the polynomial regression model, other values are inputted to construct a line of best fit in a curvilinear manner. Once the structure of the model is formed, the machine will learn from its previous data to output a prediction for data not part of the model.

Lastly, the front end of our app was constructed with Flutter to display all the functionalities to the users in a friendly and convenient manner.

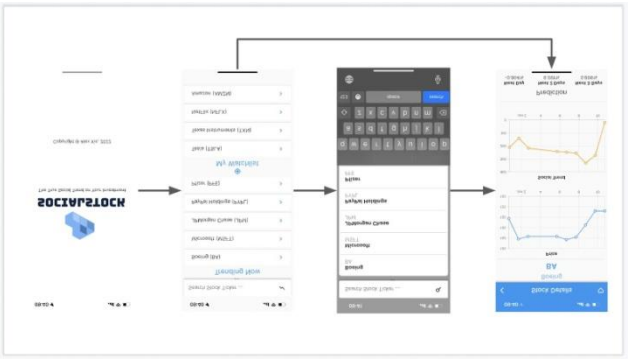

Figure 7. App storyboard

Figure 7 is a diagram of the front end of the app. When prospective users download the app, their interfaces will look like the one shown here. There will be a splash screen with the logo and name of the app, along with the copyright. The splash screen disappears after two seconds, and the user will be prompted to enter the home page of the app. On the home page, there are three components. The first one is a search bar where users can search all the 6000+ stocks we have in our backend. They can simply type the ticker or the name of the company and select their stock. The second component is a daily trend of the stocks that are most discussed on social media recently. The third component is the ability to save a stock to the user's watchlist for easier access. Finally, once the user decides on a stock that they want information on, a detailed information page will be displayed that contains the stock name, price, trend, and, most importantly, its predictions.

```
onTap: () async {
  print(
      "onTap! " + LocalDB.stocks_watchlist[index]['name'].toString());
  // go to third_page
  await Navigator.push(
    context,
    MaterialPageRoute(
        builder: (context) => MyThirdPage(
            title: 'Home Page',
            stock: LocalDB.stocks_watchlist[index])),
  );
  setState(() {
```

Figure 8. Watchlist function code with flutter

Figure 8 is just a snapshot of one of the functions we included in our app, which is a watchlist [15]. Many online investing apps have this function, so we included it to improve the user experience. How this works is that each user has their own watchlist that they can manage. Whenever they search for stock or click on a stock, the user has the ability to add that stock to their watchlist by clicking a button on the top right corner. There is nothing special about the function, just to make the app more inclusive.

## 4. EXPERIMENT

### 4.1.Experiment 1: The Accuracy siung Different Machine Learning Algorithms

For all the three experiments we have conducted, we used a total of 50 input data that contains information on ten stocks we picked. Each stock contains five lines of information with four columns: trend count, month, day, stock No. This input data remains the same for the first two experiments. For the last experiment, we have to cut down the number of columns in order to meet the criteria of the experiment.

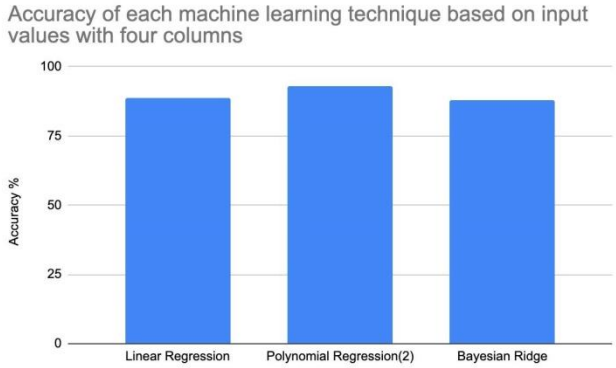


Figure 9. Graph displaying the accuracy of different machine learning algorithms

Clients may constantly be suspicious of the actual precision of the app. To address that, we have to test out the reliability of our machine learning program. The first step in creating a machine learning program is to select which machine learning method is most appropriate to use to compute our data. There are many learning techniques available, but only some are designed to solve our circumstances. After researching, we have picked three machine learning techniques that would yield the highest accuracy. This first experiment we performed was to simply test out which of the selected machine learning techniques would be most accurate when it comes to predicting great numbers of inputs. All three models showed slight variation, with polynomial regression being the most accurate. The concept of polynomial expression is relatively broad. Different degrees yield different results. In this experiment, we only used polynomials to the second degree for the simplest. We will be testing polynomials to different degrees in our second experiment. One reason for this outcome could be because the data we used neither shows a linear pattern (linear regression) nor shows a huge fluctuation (bayesian ridge). The result of this experiment informs us that the parabolic shape generated by the polynomial regression to the second degree most appropriately represents the trend that the stock information contains. This is not to say that the other two methods are not accurate; they are still pretty accurate based on the result but might only work in less likely conditions which stock data we are analyzing.

## 4.2. Experiment 2: The Accuracy of Using Different Parameters for Polynomial Regressions

To know if our user experience is good and if the user is satisfied with the UI design, we publish our app in the market and advertise the production in communities and schools. A total of more than 1,000 students used our app to help their parents, and we received a total of 20 feedback questionnaires. We collect the data and make a diagram to show the feedback result.

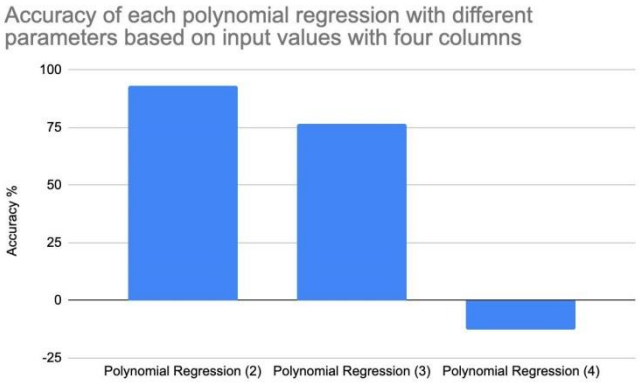


Figure 10. Graph displaying the accuracy of each polynomial regression with different parameters

Since we tested that polynomial regression yields the best result out of all the three methods, we went even further to test out polynomial regression with different parameters. The word polynomial means an equation with several terms and different powers. In order to make even more accurate results to improve the app, testing out which degrees predict the best result is necessary. In this experiment, we tested polynomials with degrees of 2, 3, and 4. As a brief overview of what a polynomial graph looks like in math, any polynomial with an even degree resembles the shape of a parabola, and any polynomial with an odd degree resembles the shape of a snake. Given the two completely different shapes a polynomial graph represents, the outcome of the result would be a lot different. As shown in the chart above, we tested out that a polynomial regression with a degree of 2 predicts our given input the best. Polynomial regression with a degree of 3 did show some consistency when predicting the outcome, so it is still a reliable algorithm but might not be as inclusive as the former choice. What is interesting about this experiment is the fact that a polynomial regression with a degree of 4 showed negative results when the score came out. Theoretically, a polynomial with a degree of 4 looks very similar to a polynomial with a degree of 2. The only difference between them is that a degree of 4 makes the parabola sharper. In other words, it makes the parabola closer to a U shape with noticeable vertices. A reason behind this might be that the sharp turn created by the high degree limits the condition of the data. Since the data almost never follows the domain of the graph, stricter conditions underperform.

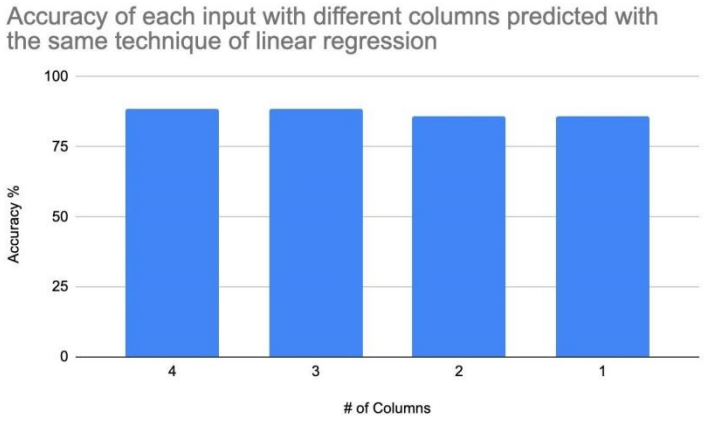


Figure 11. Graph displaying the accuracy of datasets with different columns

Last but not least, testing the suitable number of columns in the data finalizes our machine learning technique. When we web scraped the information on a certain stock, we stored many data such as trend count, price, date, etc. in our storage. One set of data contains many indexes. In terms of testing our algorithm, we picked at most four values to include that are the most relevant to the stocks. Those four values we picked are trend count, month, day, and the serial number we gave for each stock. When each of these values is included in a stored list, the machine will examine each index and study them with the pre-given performance to come up with a result. This means that when one index is removed, the result of the prediction will change. The order given before is listed from most important to least important (trend count, month, day, serial number). With every trial, we removed the index from the right side until we reached only one index of the stock's trend count. Our initial hypothesis was that with fewer indexes, the algorithm would perform more accurately since it does not have to focus on a huge dataset. However, from the final result, it is clear that this hypothesis is turned down. Instead, there is a clear pattern that when the algorithm is given more information, it ends up performing better, even only a slight bit.

## 5. RELATED WORK

Ding, Xiao, et al. proposed a different technique when predicting stock performance [10]. This was an event-driven technique in which developers used many past events as an indicator of how the stock market is likely going to perform in the future. This program extracts news events from outside sources that could be summarized into an inclusive prediction. Our work differs from this individual's since we were extracting social trends such as tweets and posts about a certain stock. By focusing on social trends instead of solely on news events, the data can be even more inclusive because social trends are directly linked to news and many other factors. The social trend represents investors' expectations toward a stock which came from influences like news events.

Holthausen, Robert W., and David F. Larcker used an old-fashion way of predicting stocks [11]. They only looked at the prices of the stocks from many years ago as their evidence for prediction. This method is done by deeply analyzing the price changes, including many performance indicators that give the developer the most appropriate outcome. This is a very straightforward method and probably the simplest, but it has its flaws. As most people are aware, the stock market is full of unpredictable factors. Looking at previous performance only serves as a reference; it does not accurately predict its performance in the future. In our program, we analyze data from a wide spectrum and consider many different trends before reaching a conclusion. The social trend has more reliable information since those analyses essentially came from professionals that knew the topic well. We pull those data to help us formulate our program.

Last but not least, Rather, Akhter Mohiuddin, Arun Agarwal, and V. N. Sastry used a hybrid model of representation for machine learning and big data analysis [12]. In this design, the developer used multiple machine learning techniques far beyond linear and polynomial regression. Many non-linear models were used and tested to improve the accuracy of the predicted result. When designing our work, we omitted using multiple techniques because it requires endless hours of testing and reformulating to match a technique to a stock. We simply tested and found the most effective method that covers most of the trends that came from those 6000+ stocks in our database. This not only made the process more efficient but also alleviated the pressure placed on the algorithm.

## 6. CONCLUSIONS

As technology advances and people invest more, virtual currency and digital exchange will be the trend in almost every nation [13]. However, professional investors oftentimes dominate the market, suppressing amateur investors and rug-pulling on average citizens [14]. In order to create a fair economy, it is important that everybody uses their wealth to the fullest potential. By providing an accurate prediction of the performance in the stock market, we can help amateur investors to quickly and safely adapt to the complicated market and achieve multiplication of wealth. We web scraped data from Twitter to use as our concrete evidence for the prediction. Social media trends directly reflect on the future performance of a stock because, with a high number of discussions on a certain stock, people's expectations rise, thus influencing the performance of the stock. By using the machine learning technique to analyze these social trends, the system is able to predict as close to precision as possible. With this reliable source as a backbone supporting the prediction, users can feel free to use our product without any major concerns. Additionally, our implementation of all the 6000+ stocks creates many more choices for the users. We try to aim for a perfect user experience; providing the basic quantity of stocks is the first step in servicing our users. Our databases safely store these stocks, and all the information on these stocks came from Yahoo Finance. Users can almost find any public stocks. Together with our easy-to-use front end, people from all age groups will feel effortless when using the app.

Even though we have tested our machine learning program thousands of times, accuracy is still a major concern for us. It is reasonable to assume that when it comes to dealing with prediction, there is not a single way to predict the future 100% accurately. Although our model represents the performance of a stock very closely, the margin of error is still present and sometimes bigger on certain stocks. Second, renewing the stock list periodically will be hard to achieve. As more private companies become public and enter the stock market, more stocks will appear on exchange platforms. Our program currently does automatically renew the stock list, it must be done manually by the developers.

Along with updating the app from time to time, it is very important for us to keep improving the accuracy of the prediction. Having a slightly inaccurate prediction can hurt our users. The core concept of this app is to predict. If we cannot effectively accomplish that, the rest of the functionalities are just extras.

## REFERENCES


[1] Stockhammer, Engelbert, and Lucas Grafl. "Financial uncertainty and business investment." Review of Political Economy 22.4 (2010): 551-568.

[2] Ramazani, Jalal, and George Jergeas. "Project managers and the journey from good to great: The benefits of investment in project management training and education." International Journal of Project Management 33.1 (2015): 41-52.

[3] Liu, Jun, and Francis A. Longstaff. "Losing money on arbitrage: Optimal dynamic portfolio choice in markets with arbitrage opportunities." Review of Financial studies (2004): 611-641.

[4] Groth, Sven S. "Does algorithmic trading increase volatility? Empirical evidence from the fully-electronic trading platform Xetra." (2011).

[5] Kieffer, Christine, and Gary Mottola. "Understanding and combating investment fraud." Financial decision making and retirement security in an aging world 185 (2017).

[6] Mazorra, Bruno, Victor Adan, and Vanesa Daza. "Do Not Rug on Me: Leveraging Machine Learning Techniques for Automated Scam Detection." Mathematics 10.6 (2022): 949.

[7] Watanabe, Y., et al. "An experimental study of paper flutter." Journal of fluids and Structures 16.4 (2002): 529-542.

[8] Chatterjee, Nilanjan, et al. "Real-time communication application based on android using Google firebase." Int. J. Adv. Res. Comput. Sci. Manag. Stud 6.4 (2018).

[9] Bermudez, Ignacio, et al. "Exploring the cloud from passive measurements: The Amazon AWS case." 2013 Proceedings IEEE INFOCOM. IEEE, 2013.

[10] Ding, Xiao, et al. "Deep learning for event-driven stock prediction." Twenty-fourth international joint conference on artificial intelligence. 2015.

[11] Holthausen, Robert W., and David F. Larcker. "The prediction of stock returns using financial statement information." Journal of accounting and economics 15.2-3 (1992): 373-411.

[12] Rather, Akhter Mohiuddin, Arun Agarwal, and V. N. Sastry. "Recurrent neural network and a hybrid model for prediction of stock returns." Expert Systems with Applications 42.6 (2015): 3234-3241.

[13] Dibrova, Alina. "Virtual currency: new step in monetary development." Procedia-Social and Behavioral Sciences 229 (2016): 42-49.

[14] Gaskill, N. "Rorty against Rorty: climate change, rug-pulling, and the rhetoric of philosophy." Common Knowledge (2020).

[15] Bannier, Christina E., and Christian W. Hirsch. "The economic function of credit rating agencies–What does the watchlist tell us?." Journal of Banking & Finance 34.12 (2010): 3037-3049.