

STOCK RECOMMENDATION SYSTEM

Using Python

Jashanjit Singh

singhj14@montclair.edu

Department of Computer Science

Montclair State University, Montclair, New Jersey, 07043

Abstract:

One of the most difficult analyses of all time is stock market predictions. Expert analysts and software engineers are collaborating to create a stable and reliable platform for predicting future stock value. The nature of weekly price fluctuation, which is influenced by a variety of factors, is the fundamental difficulty. Stock recommendations are vital for investing firms and individuals. No stock selection approach, however, can capture the dynamics of all stocks without adequate analysts. Nonetheless, the majority of extant recommendation techniques are built on prediction algorithms ANN (Artificial Neural Network) to buy and keep high-yielding companies.. We offer a unique strategy in this paper that uses reinforcement learning to recommend a stock portfolio based on the Yfinance data sets. We will present an ARIMA framework for recommendation systems, as well as a foundation for determining the system's value. Within this paradigm, we do probabilistic studies of algorithmic approaches. These studies illustrate how much usefulness can be acquired from recalling earlier activities and how this recollection may be used.

Keywords—Artificial Neural Network, ARIMA, stock forecasting, Yfinance.

CHAPTER I: INTRODUCTION

Recommendation systems also referred to as Collaborative filtering is an interaction by which data on the inclinations and activities of a group of users is followed by a framework that then, in view of the patterns it notices, attempts to make valuable recommendations to individual clients [1,8].

Stocks are a high-risk, uncertain investment, however with the exceptional yield to financial backers. The person has numerous ways of anticipating the cost of stock and among others is utilizing time-series examination where quite possibly the most often utilized procedure is Autoregressive Integrated Moving Average (ARIMA). The Autoregressive Integrated Moving Average (ARMA) explanation is a dynamical system that is applied into temporal arrangement data to gain a better

understanding of the data and to make future predictions in the setup. These models give solid and precise anticipation. ARIMA models have just been utilized for determining oil, gas, and characteristics and in addition stack estimating in control framework with great outcomes[2][3].

Machine Learning Model [4]: Machine learning is an artificial intelligence area that allows you to statistically examine stock returns data. In this strategy, the system learns from the evaluated data set and recommends new stocks to investors using techniques such as support vector machine (SVM), neural network (NN), genetic algorithms (GA), linear regression (LR), and association rule mining (ARM).

The majority of previous investment recommendation methods are designed on formulas. The stock to be advised, as well as the degree of effectiveness and user data, are all factors within calculation. The stock is suggested by optimizing the estimation outcome using a formula based on a specific performance challenge. The formula utilized in a formula-based system is frequently changed depending on a recommender's experience. As a result, stock recommendations made using a formula-based method are very likely to favor the recommender.

CHAPTER II: PROPOSED SYSTEM

Using yahoo financial data to build the recommendation in Python. This will contain all stock information. It will be a tkinter app, a Python library that provides graphical user interface tools, where the app will produce a future forecast for the following day using model ARIMA, where you choose the stock and the estimate will be displayed. The value of the most searched stock in "US" or any other country the user wants, based on ticker value (stock price and share price). Stock recommendation will utilize data from the Yahoo Finance API to propose a stock based on technical indicators such as MACD (moving average convergence divergence) and RSI (Relative Strength Index) signals. The basic features of ARIMA, it is made up of three parts: AR (autoregressive term), I (differencing term), and MA (multivariate analysis) (moving average term).

1. Stock value forecast for the coming week
 - a. The user need to provide the stock name.
2. Using the Pytrends module in Python, which stock is now the most searched
 - a. It will produce a csv file with the top trending stocks without any user input.
3. The system will recommend a stock to purchase from all the stocks in the Yahoo Finance dataset based on MACD and RSI.
 - a. In the Tkinter window, the values and the stock name will be displayed.
4. Additional improvements and adjustments will be communicated if they are implemented.

CHAPTER III: SYSTEM IMPLEMENTATION

This section we will discuss the model of our system. Our system consists of several stages which are as follows:-

Existing stock data is obtained from Yfinance <https://in.finance.yahoo.com> at this step, and this data set is utilized to forecast future stock values. For web scraping, we'll utilize the pandas function web reader from the Python designed package.

- Data must be loaded: The first step for model building is of course to load the dataset.
- Data Pre-processing: The phases of pre-processing will be determined by the dataset. This will cover things like producing timestamps, changing the date/time column's data type, converting the column binary, and filling in missing data, among other things.
- Training Model: We will employ a variety of statistical tools, investing algorithms, and time series forecasts at this point.
- Generating Results: The abovementioned procedure may be used to create output in this stage. We normally present the results in a graphic format. We'll also calculate error and refine our model depending on the results.

The final results will be saved on the system asking the user for the saving location in the form of CSV file and png images of price prediction and information in the form of graphs.

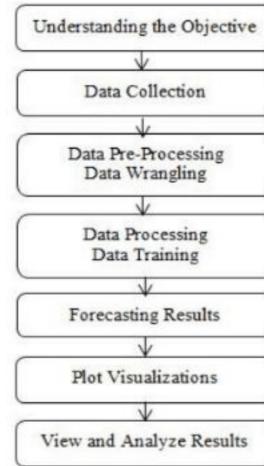


Figure 1: Architectural Design

CHAPTER IV: EXPERIMENTAL RESULTS

We have created a simple but functional graphic user interface using tkinter library in python. Tab1 and Tab2 as seen in *figure 3*, both having different functions.

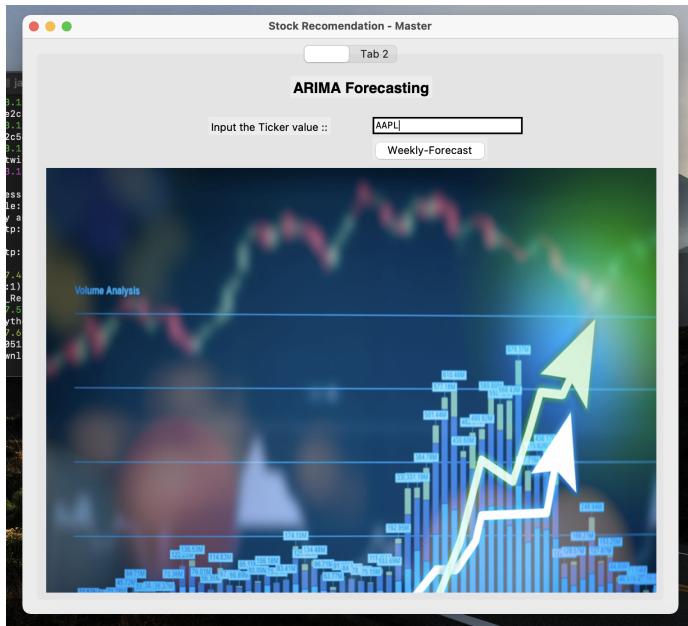


Figure 2: Tab 1 GUI

As seen in *figure 2* Tab1 has the forecasting function where it asks for user input for the “Ticker” value and a “Weekly-Forecast” which as explained before saves file on the system. *Figure 3,4,5,8* are the results for “Weekly-Forecast” button

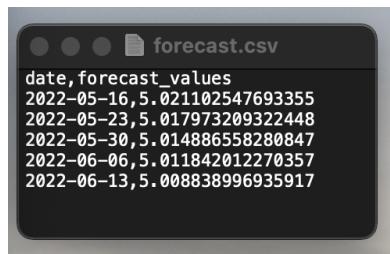


Figure 3: Weekly-forecast for Apple stocks-AAPL

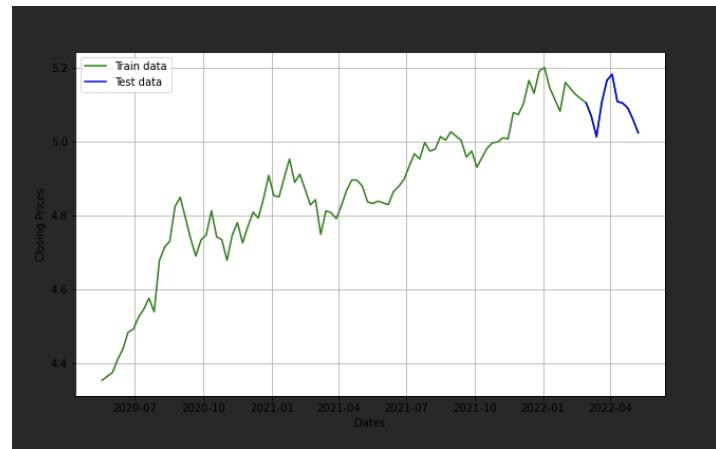


Figure 4: Trained & Test data interpretation of closing prices

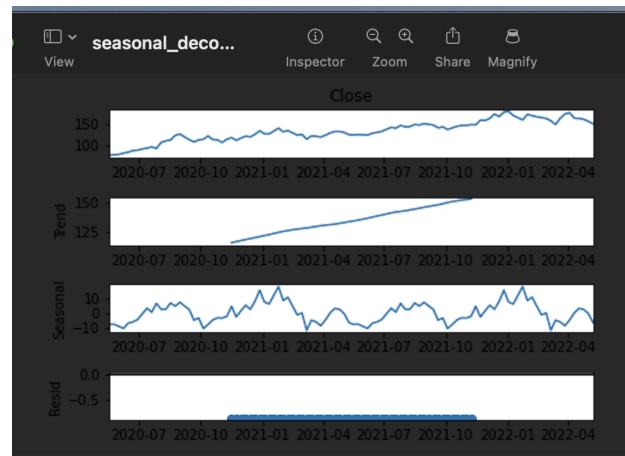
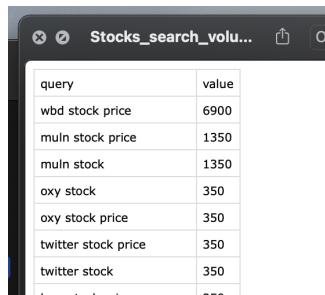


Figure 5: Seasonal Decompose

As can be seen in *Figure 7*, Tab 2 has two options:

1. Trending-Stock: It basically gives the latest information for the stocks which are in trend using the yahoo finance data. Can be seen in in the form of csv *Figure 6*



query	value
wbd stock price	6900
mulin stock price	1350
mulin stock	1350
oxy stock	350
oxy stock price	350
twitter stock price	350
twitter stock	350

Figure 6: Trending- Stock

2. Stock-recommendation: This button provides the recommended stock accordingly to the historical data of the stock information. This data is also saved in the form of csv file

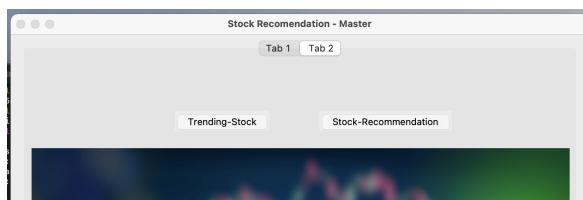


Figure 7: Tab 2 GUI

The "Weekly-Forecast" button in the system is used by the Data Collection module to acquire real-time data. Data is retrieved using the date and the opening and closing values of

the companies which is input by the user(AAPL- Apple Inc, GOOG- Google/Alphabet Corporation, YHOO- Yahoo!, and so on) [16]. It then displays the fetched results in the home screen by plotting a graph of date versus value in the x and y axes, respectively.

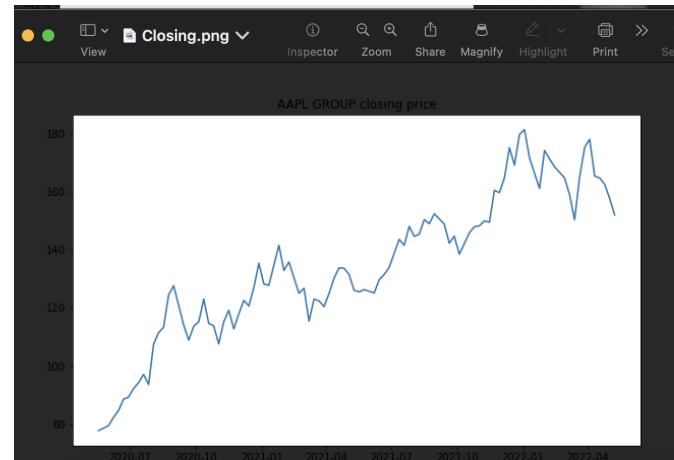


Figure 8: Closing price graph

The pre-processed data is being used in *Figure 6*, as can be seen Train data is the processed data and then based on the trained data the Test data is being predicted in blue with accuracy of 98%.

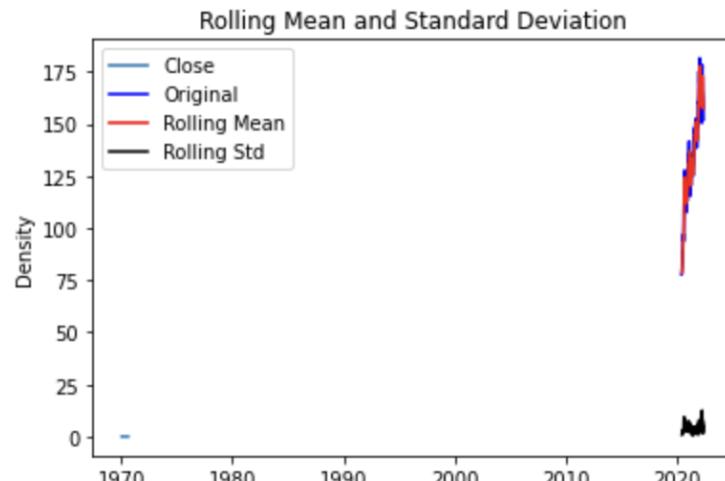


Figure 9

CHAPTER V: CONCLUSION REMARKS AND FUTURE WORK

Stock information was gathered by scanning, then analyzed, and stocks were visually divided by recruiting technique, investment aim, and management approach using the K-Means algorithm. Following the findings, a Prophet-based price prediction software was created. You may verify the future return of the entered stock using this technique. It offers economic feasibility, which saves you time and money by allowing you to understand your investing proclivity without having to spend additional time, and it recommends an appropriate stock. Furthermore, while just stock information query was allowed on the old stock-related website, this document is unique in that it stores the inspection product. The

system was intended with functionality in mind so that people can easily navigate and use it. Because the training result is varied, a very effective suggestion is created through repeated learning and presents distinct outcomes for each individual in the event of financial prediction. In the age of digital revolution, this will facilitate high energy[7].

We compare the performance of all classifiers deployed within the predictive system framework using the accuracy and F1 score metrics. We provide an illustration using one common plot to get a clear at-a-glance comparison, as shown in the following figure. Figure 9 depicts the accuracy all classifiers used in the predictive system. Overall, the Passive Aggressive Classifier consistently achieves the best performance across all experiments in our study.

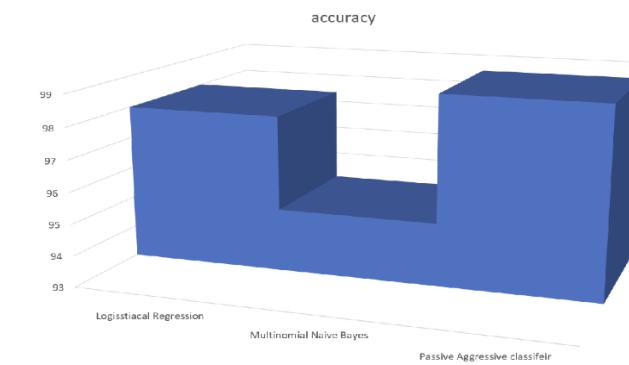


Figure 10: Accuracy for all classifiers in the predictive system

By combining the technical and fundamental analytical methodologies, the execution of this article may be improved. Fundamental analysis approaches could well be applied into

the assessment of social media analysis, especially on public perceptions, to achieve better outcomes. As a consequence, we can give better outcomes for stock market participants, allowing them to make more successful investment selections at a better time.

REFERENCES

- [1] D. Goldberg, D. Nichols, B. M. Oki, and D. Terry. Using collaborative filtering to weave an information tapestry. *Communications of the ACM*, 35:12, pp. 51–60, 1992.
- [2] O. B. Fosso, A. Gjelsvik, A. Haugstad, M. Birger, and I. Wangensteen, “Generation scheduling in a deregulated system. The Norwegian case,” *IEEE Trans. Power Syst.*, vol. 14, no. 1, pp. 75–81, Feb. 1999.
- [3] ARIMA Models to Predict Next-Day Electricity Prices
Javier Contreras, Member, IEEE, Rosario Espínola, Student Member, IEEE, Francisco J. Nogales, and Antonio J. Conejo, Senior Member, IEEE.
- [4] Kirkpatrick dan Dahlquist Technical Analysis: The Complete Resource for Financial Market Technicians (Financial Times Press, 2006), page 3
- [5] D. Ciregan, U. Meier, and J. Schmidhuber, “Multi-column deep neural networks for image classification,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2642–3649, 2012.
- [6] A. Graves, M. Liwicki, S. Fernandez, R. Bertolami, H. Bunke, and J. Schmidhuber, “A novel connectionist system for improved unconstrained handwriting recognition,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31.
- [7] J. W. Kwon and T. S. Song, “Platform-based digital transformation promotion trend for manufacturing innovation,” *Journal of Electronic Engineering*, vol. 46, no. 12, pp. 34–46, 2019.
- [8] W. Hill, L. Stead, M. Rosenstein, and G. Furnas. Recommending and evaluating choices in a virtual community of use. *Proceedings of ACM CHI*, pp. 194–201, 1995.
- [9] ACM SIGGROUP resource page on collaborative filtering. www.acm.org/siggroup/collab.htm
- [10]<https://www.premiumleads.com/en/blog/seo/how-to-get-google-trends-data-with-pytrends-and-python/>