EPiC Series in Computing

Volume 91, 2023, Pages 164–171

Proceedings of 38th International Conference on Computers and Their Applications

A Stock Recommendation System

Jashanjit Singh[1] and John Jenq[2]

[1, 2] Montclair State University, Montclair, New Jersey, U.S.A.

singhj14@montclair.edu, jenqj@montclair.edu

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 fundamental difficulty is that a variety of factors will influence the price fluctuations. Stock recommendations are vital for investment firms and individuals. However, no unique stock selection approach 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 the value of recalling earlier activities and examines how this recollection may be used.

1 Introduction

Often times, we ask for recommendations from others. For example, we may ask a realtor to recommend a house to purchase within our budget in a good location. We recommend friends to visit a nice restaurant, to watch a good movie or an interesting online video, etc. Online system such as Yelp is a user centric recommendation system based users' review and rating. Companies such as Google, Yahoo, etc., provides advertisement clips to recommend users to buy things. These advertisement clips will pop up automatically based on their user profile modules. Some recommendation system insert these advertisements on the article that users are reading or a webpage a user is visiting. As an investor, you may need recommendation from financial advisors, or your friends to maximize your investment return. 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, 2].

Stock market is a volatile system. There are so many variables, which affect the market expression and performance. 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 dynamic system applied to temporal arrangement data to better understand the data and make future predictions in the setup. These models give solid and precise anticipation. There are research articles related to this method see for example [3, 4]. ARIMA models had been utilized for determining oil, gas, and characteristics and in addition stack estimating in control framework with great outcomes [5, 6].

There are different machine learning models. Machine learning is an artificial intelligence area that allows you to examine stock returns data statistically. 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). See [7] as an example, which gives introduction about machine learning among many others. Chen and Hossain gave a very through and nice survey about machine learning on software quality assurance and testing [8].

The majority of previous investment recommendation methods are using formulas, the stock(s) considered, as well as the degree of effectiveness and user data, are all factors within the calculation. By optimizing the estimation outcome using a formula based on a specific performance challenge, stock(s) would then recommended as sell, hold, or buy. As we know, stock market is a very volatile system. The indicators used for stock prediction may change from time to time. It implies the formula utilized in a formula-based system will also change frequently, often times depending on a recommender's experience. As a result, stock recommendations made using a formula-based method are very likely to favor the recommender. In [9], three statistical models, regression, multi-layer perceptron, and Hidden Markov were used with seventy-five companies' datasets to predict stock market values.

In this report, we prototype a stock recommendation system using Python programming language. Section 2 describes our proposed system. In section 3, we details the component implementations. Experimental results and screenshots are in section 4. Section 5 concludes with possible future improvement and remarks.

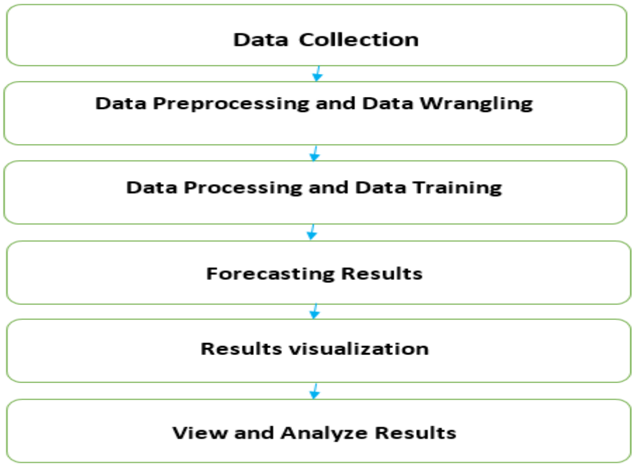
2 Proposed System and Design

We propose a prototype system with graphical user interface. It provides a simple interface so even a beginner investor can use it. The controller responds to the user’s input will perform data analysis. Corresponding data related to user’s input would be retrieved from the internet. It will be preprocessed, test on various model and generate the results. **Figure 1** shows the workflow of the controller.

There is a simple view for this prototype system, so even for a naïve user knows how to use. It showed as in **Figure 2**. This figure shows when user pressed Tab1. When the user input data and press the Weekly Forecast button, it will trigger the controller to perform its designed tasks to generate the desired results and give recommendations. User can press Tab2 as well. The Tab1 and Tab2 provide different functionalities to be explained further in the later section.

The data can be stored as csv file format in the user’s local computer. Graph files would be produced after the running of the controller and will be stored on the user’s computer for later usage.

As for the backend process, we hypothesize that the Logistic Regression, Multinomial Naive Bayes, and Passive Aggressive Classifier algorithms were chosen for the study because they are often used in classification tasks and have proven to be effective in text classification tasks. These algorithms are especially useful for studying market patterns and evaluating the tone of news stories about equities.



Data Collection

Data Preprocessing and Data Wrangling

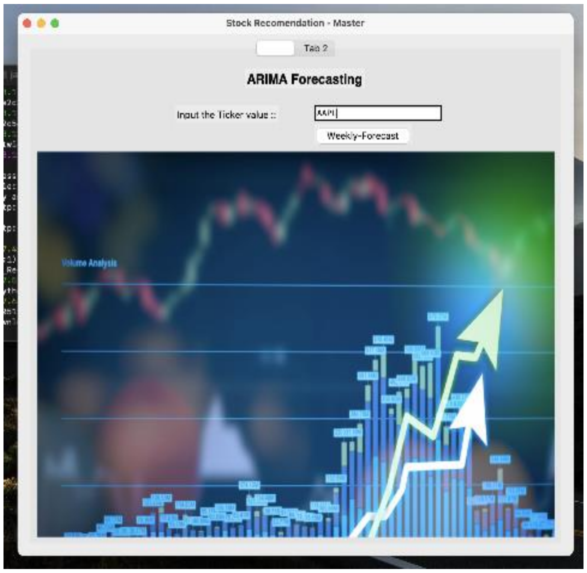
Data Processing and Data Training

Forecasting Results

Results visualization

View and Analyze Results

Figure 1. Controller Design



Stock Recomendation - Master

Tab 2

ARIMA Forecasting

Input the Ticker value :: AAPL

Weekly-Forecast

Figure 2. The tab1 input View

A popular linear model for binary classification issues is logistic regression. It gains knowledge of a decision boundary between the two groups. A probabilistic model that is frequently used for text categorization is multinomial naive Bayes. The probability of each feature given each class is estimated under the assumption that features are conditionally independent given the class. A linear model that is commonly adopted for online learning tasks is called the passive aggressive classifier. Based on samples that were incorrectly categorized, it changes the model's weights.

To discover which algorithm performs the best for forecasting market movements and making stock recommendations based on Google search patterns, the authors analyzed the performance of these three classifiers. The findings demonstrated that the Passive Aggressive Classifier consistently outperformed all other experimental designs in their investigation. This implies that it could be the method that is best suited for this task.

3 System Implementation

We will use yahoo financial data to build the recommendation in Python. The download data from Yahoo contains stock information. The system is a Tkinter app, Tkinter is a Python library that provides graphical user interface tools. The app will produce a future forecast for the following days using model ARIMA, where you choose the stock and the estimation will be displayed. A user can search stock in Yahoo website based on ticker value (stock price and share price). The 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).

On the main input window, there are two textboxes, which allow user to enter the desired stock ticker name. The system will provide the most searched stock at the time of using the system. It should output a csv file, which contains top trading stocks. The proposed system will recommend a stock to purchase from all the stocks in the Yahoo Finance dataset based on MACD and RSI. Our system will retrieve stock data from Yfinance [10]. The data set is then be used to forecast future stock values. For web scraping, we will utilize the pandas function web reader from the Python designed package.

After loading the dataset, it will enter the phase of pre-processing, which will include producing timestamps, changing the date/time column's data type, converting the column binary, and filling in missing data. For the model-training phase, we will employ a variety of statistical tools, investing algorithms, and time series forecast at this stage.

After testing the dataset on different models, the system will create output in this stage. The results are then presented in graphic formats. We will also calculate error and refine our model depending on the results. In addition to the displaying of graphic diagrams, the results can also be saved on the local file system. The system will ask 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.

4 Experimental Results

As we have seen in **Figure 2** When Tab1 pressed, it will produce forecasting information about the stock under consideration. It asks user to input the ticker on the "Ticker" textbox. The controller to retrieve the stock data from Yahoo finance uses this ticker. **Figure 3** shows the diagram for the historical closing price of NYC REIT Company. When the use press the "Weekly-Forecast" button, the forecast results will be generated as explained before. It also saves these results as data files into the user’s local file system

Figure 4 shows the weekly forecast value for NYC REIT stock. **Figure 5** depicts a diagram to forecast the daily value for the same stock using training and testing data. The forecast values on **Figure 4** shows a table, which represents the changes from the previous period.

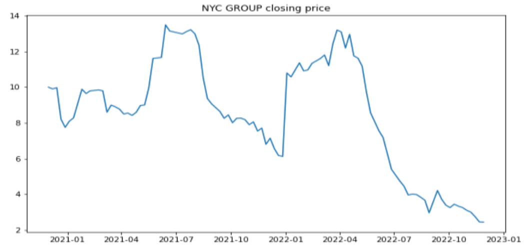
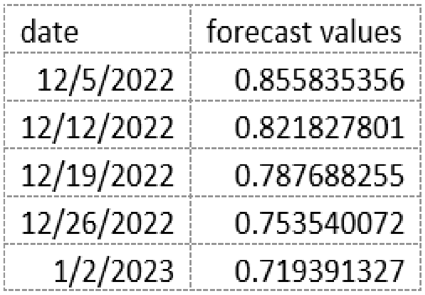


Figure 3. Closing Price



date	forecast values
12/5/2022	0.855835356
12/12/2022	0.821827801
12/19/2022	0.787688255
12/26/2022	0.753540072
1/2/2023	0.719391327

Figure 4. Weekly Forecast

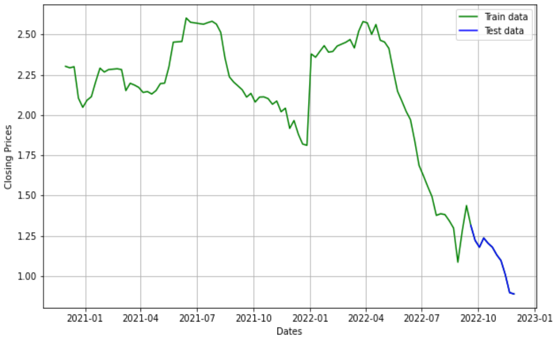


Figure 5. Use training and testing data to predict stock value

A time series is broken down into its trend, seasonal, and residual components using the forecasting technique known as seasonal decompose. This approach can be employed to forecast stock trends using past data in stock recommendation systems.

The seasonal component indicates recurring patterns or cycles that occur within the time series, whereas the trend component represents the time series long-term behavior. The random variations that cannot be explained by the trend or seasonal components are represented by the residual component.

We may obtain an understanding of the fundamental patterns and swings that govern the performance of a stock by dissecting historical time series data into its trend, seasonal, and residual components. With this data, it is possible to develop a more precise stock recommendation system that accounts for the company's performance's cyclical patterns as well as random variations in addition to its general trend. In general, the seasonal breakdown prediction can offer useful insights into the variables that affect a stock's performance, enabling more precise and knowledgeable investing choices. **Figure 6** shows the seasonal Forecast for NYC REIT stock values.

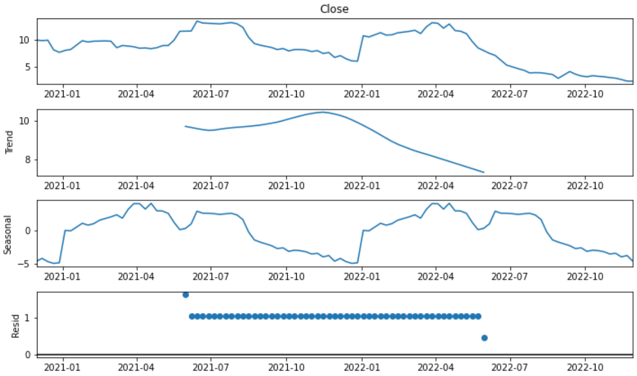


Figure 6. Seasonal Decompose Forecast

As can be seen in **Figure 7**, there are two options on the screen, Tab1 and Tab2. In **Figure 7**, user pressed the Tab 2. There are two buttons, which provide two functionalities.

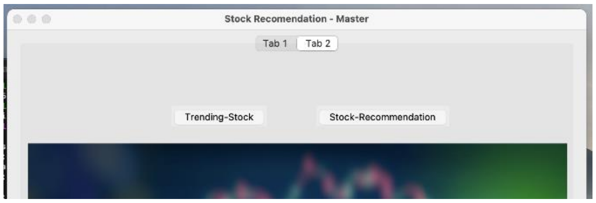
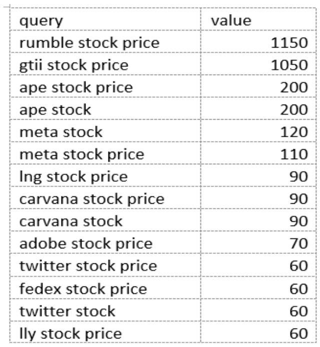


Figure 7. Tab2 GUI

(1) Trending-Stock: It gives the latest information for the stocks, which are in trend. **Figure 8** shows the most search stocks on Google searching engine. Since it only count searches using Google and does not have the counts of other searching engines, this result is only a reference to the user. On **Figure 8**, the most search stock in this case is Rumble and the second searched is gtii, the Global Tech Industries Group Inc. It just gives us some hint about the stocks internet users are interested.

(2) Stock-recommendation: This button provides the recommended stock accordingly to the historical data of the stock information. It can give us a recommendation stock to buy.



query	value
rumble stock price	1150
gtii stock price	1050
ape stock price	200
ape stock	200
meta stock	120
meta stock price	110
lng stock price	90
carvana stock price	90
carvana stock	90
adobe stock price	70
twitter stock price	60
fedex stock price	60
twitter stock	60
lly stock price	60

Figure 8. Stock search trending

5 Conclusion Remarks and Future Work

In this report, we prototype a simple Python based stock recommendation system. This system can predict a stock's future trend. It can also recommend user to buy a stock among most search stocks on Google searching engine using Google pytrends API [11].

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 of all classifiers used in the predictive system. Overall, the Passive Aggressive Classifier consistently achieves the best performance across all experiments in our study.

By combining the technical and fundamental analytical methodologies, the execution of this article may be improved. Fundamental analysis approaches could well be applied to the assessment of social media analysis, especially on public perceptions, to achieve better outcomes. Therefore, we can give better outcomes for stock market participants, allowing them to make more successful investment selections at a better time. The system can further improved by using other friendlier user interface such as voice activate interface.

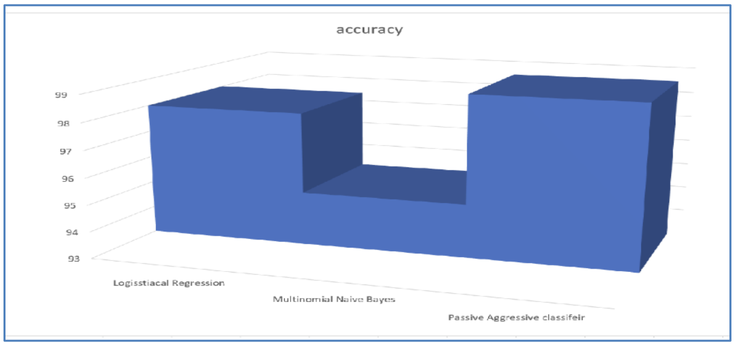


Figure 9. Accuracy for all classifiers in the predictive system

References

[1] A. K. a. A. S. Markus Weimer, "Adaptive collaborative filtering," in *Proceedings 2008 ACM Conference on Recommender Systems*, 2008.

[2] X. He, L. Liao, H. Zhang, L. Nie, X. Hu and T.-S. Chua, "Neural Collaborative Filtering," in *Proceedings of the 26th International Conference on World Wide Web*, 2017.

[3] Chee Ka Chin, Dayang Azra Binti Awang Mat and a. A. Y. Saleh, "Skin Cancer Classification using Covolution Neural Network with Autoregressive Integrated Moving Average," in *Proceedings of the 2021 4th International Conference on Robot Systems and Applications*, 2021.

[4] W. Masood and J. F. Schmidt, "Exploring Autoregressive Integrated Models for Time Synchronization in Wireless Sensor Networks," in *Proceedings of the 2015 Workshop on Wireless of the Students, by the Students, & for the Students*, 2015.

[5] O. B. Fosso, A. Gjelsvik, A. Haugstad, M. Birger and I. Wangensteen, "Generation Scheduling in a Deregulated System. The Norwegian Case," *IEEE Transactions on Power Systems,* vol. 14, no. 1, pp. 75-81, 1999.

[6] J. Contreras, R. Espinola, F. J. Nogales and A. J. Conejo, "ARIMA Models to Predict Next-Day Electricity Prices," *IEEE Transactions on Power Systems,* vol. 18, no. 3, pp. 1014-1020, 2003.

[7] E. Alpaydın, Introduction to Machine Learning, The MIT Press, 2020.

[8] H. Chen and M. Hossain, "Application of Machine Learning on Software Quality Assurance and Testing: A Chronological Survey," in *Proceedings of 37th International Conference on Computers and Their Applications*, 2022.

[9] R. Rosas-Romero and J. P. Medina-Ochoa, "Learning Financial Time Series for Prediction of the Stock Exchange Market," in *CATA*, 2019.

[10] "Yahoo Finance," [Online]. Available: https://finance.yahoo.com.

[11] "pytrends 4.8.0," pypi.org, [Online]. Available: https://pypi.org/project/pytrends/ .