Predicting America’s Stock Market during Pandemic using Machine Learning

Abstract

Stock market plays a crucial role in the growth of the industry and commerce of the country that eventually affects the economy of the country to a great extent [1]. The year 2020 has witnessed a dramatic decline in stock market for most industrial and government sectors due to the potential worsening of COVID-19 pandemic – a threat for lives mentally and economically. Even after years of study by the smartest brains in finance, there was not a great approach to understand the unexpected failure or success in the current stock market. Innovation and technology has always been a tool to forecast unprecedented changes that could result in stock market. Predictions go hand-in-hand with supervised machine learning [2], the algorithms and procedures can define the current state and predict the future for an improved analysis. In this project we pursue an approach to find valuable insights from social media notably from Twitter which has billions of users who put forward their thoughts and sentiments. Combining these human interactions on Twitter with data-sets from NASDAQ especially of those dealing with S&P 500 which is readily available and applying it with different regression algorithms such as Random Forest Regressor, Bagging Regressor and so on has shown results with better analytics and forecasting.

1 INTRODUCTION

A billion-dollar question for stock investors is if the price of a stock will rise or not. The variability of stock market is fierce and there are many convoluted financial indicators. Forecasting of stock trend has long been a fascinating topic of research and it has been broadly studied by researchers from different fields. Machine learning is a well-recognized algorithm in a comprehensive range of applications which has been extensively studied for its potentials in prediction of financial markets. Machine learning is an opportunity for ordinary people to gain steady fortune from stock market and also can help experts to dig out the most informative indicators and make better prediction. Popular algorithms, including support vector machine (SVM) and reinforcement learning, have been reported to be quite effective in tracing the stock market and help maximizing the profit of stock option purchase while keep the risk low [3]. However, in many of these literatures, the features selected for the inputs to the machine learning algorithms are mostly derived from the data within the same market under concern. Such isolation leaves out important information carried by other entities and make the prediction result more vulnerable to local perturbations [4]. Efforts have been done to break the boundaries by incorporating external information through fresh financial news or personal internet posts such as Twitter. These approaches, known as sentiment analysis, replies on the attitudes of several key figures or successful analysts in the markets to interpolate the minds of general investors. Despite its success in some occasions, sentiment analysis may fail when some of the people are biased, or positive opinions follow past good performance instead of suggesting promising future markets. Furthermore, only people with widespread experience and knowledge can understand the meaning of the indicators, use them to make good prediction to get prosperity. Most of other people can only rely on the luck to earn money from stock trading. Machine learning is an opportunity for ordinary people with less expertise on the financial aspects to gain steady prosperity from stock market and also can help experts to dig out the most informative indicators and make better prediction.

In this project, we propose the use of global stock data preferably from NASDAQ initially referred as National Association of Securities Dealers Automated Quotations to associate it's data with of other financial products as the input features to machine learning algorithms such as regression algorithms. There are numerous stock market forecasting tools, analysis which is being done day in and day out, there are mobile and web applications which provide handy information on stocks, some of those applications use Twitter sentiment analytics with machine learning algorithms and some applications use traditional forecasting by using just the fundamental analysis of how the company is being tied up to their products and the total number of shares owned [5]. As stated, the work which is being proposed has a potential to provide a better insight by including ensemble learning and regression algorithms such as Bagging Regressor, Random Forest Regressor and so on. Human interactions on twitter together with graphical visualization tools which benefits not only the industries but to a common set of population in general who would like to have an eye on their investments and understand the forecasting on their stocks. The graphical visualization for S&P 500 stocks, a very well-known stock market index along with the effects of pandemic is being combined. In particular, we are interested in the correlation between the closing prices of the markets that stop trading right before or at the beginning of US markets.

2 RELATED WORK

Stock prediction is a complicated and interesting problem. Most researchers focus on stock selection problem and the prediction of stock return. In our project we refine some of the ideas collectively capitulated by researchers and scientists over the last two decades. Refenes et al. [6] applied neural networks to predict stock performance. They found that even simple neural learning procedures showed better prediction accuracy than classic statistical techniques, e.g., multiple linear regression. They also claimed that with careful network design, model performance can be further improved.

Levin [7] designed a multilayer feedforward neural networks to select stocks. He showed that his model can make good prediction even if data is contaminated by large ratio of noise. Ghosn and Bengio [8] also investigated artificial neural networks to predict future returns of stocks. With a serials of experiments, they concluded that artificial neural networks have the best performance, when the neural networks for different stocks do not share any parameter or only share some parameters. In another word, to get the best prediction, one always needs to train model specifically for each stock and there is no universal model for all the stocks with the best performance.

Tsai et al. [9] examined the performance of classifier ensembles on the prediction of stock return and made comparison with single classifiers, i.e., neural networks, decision trees, and logistic regression. They studied the impact of different types of classifier ensembles and majority voting and bagging. They concluded that in general, classifier ensembles perform better than single classifier.

Leung et al. [10] compared the forecasting performance of classification models to predict the direction of index return and level estimation models to predict the value of the return. They concluded that classification models always perform better than level estimation models. They also showed that the forecasting from classification model can be used to develop trading strategies for more trading profits.

Recent data further suggests to try and to improve the stock prediction which has created new features from the base features which provided better insights of the data like 50 day moving average, previous day difference, etc. To prune out less useful features, in Feature Selection, they selected features according to the k highest scores, with the help of an linear model for testing the effect of a single regressor, sequentially for many regressors.

3 METHODOLOGY

The proposed stock forecasting model basically involves two main phases:

a. Data Pre-Processing

A usual scenario in terms of data when it comes to forecasting would be some large datasets with huge number of rows and columns. While that is a likely scenario, it is not always the case — data could be in so many different forms: Structured Tables, Images, Audio files, Videos etc. In any Machine Learning process, Data Preprocessing is that step in which the data gets transformed, or *Encoded*, to bring it to such a state that now the machine can easily parse it. In other words, the *features* of the data can now be easily interpreted by the algorithm.

The pre-processing with respect to our project involves the following sub phases:

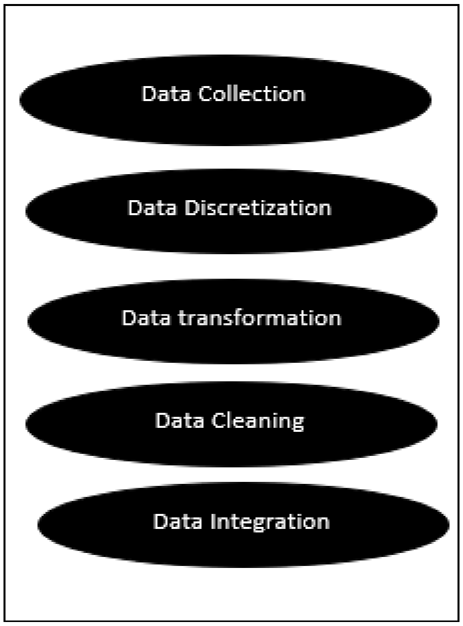

Fig 1. Data Pre-Processing

Data Collection: The data set used in this project is collected from NASDAQ. It contains S& P 500 data and listed in a table format and covers daily price from the company's inception. Since the markets are closed on holidays which vary from country to country.

Data discretization: A process where we convert continuous data attribute values into a finite set of intervals and associate them with each interval of some specific data value. This activity deals with a part of data reduction but with particular importance, especially for numerical data.

Data transformation: This activity consist of normalization of data. Data transformation is the process in which we take data from its raw, isolated and normalized source state and transform it into data that's joined together, dimensionally modeled, de-normalized, and ready for analysis.

Data cleaning: The process of fixing or removing incorrect, corrupted, incorrectly formatted, duplicate, or incomplete data within a dataset. We use NASDAQ as a basis for data alignment and all missing data in other data sources is replaced by linear interpolation.

Data integration: This process involves integration of data files. Once the data-set is transformed into clean data-set, the data-set is divided into training and testing sets so as to evaluate. Here, the training values are taken as the more recent values. Testing data is kept as 5-10 percent of the total dataset.

b. Feature Selection and Feature Generation

In this project, we focus on the prediction of the trend of stock market either increase or decrease. Therefore, the change of a feature over time is more important than the absolute value of each feature. We define xi(t), where i ∈ {1,2,..., 16}, to be feature i at time t. The feature matrix is given by As discussed above, the performance of a stock market predictor heavily depends on the correlation between the data used for training and the current input for prediction. Intuitively, if the trend of stock price is always an extension to yesterday, the accuracy of prediction should be fairly high. To select input features with high temporal correlation, we calculated the autocorrelation and cross-correlation of different market trends increase or decrease.

In other words, creation of new features from the base features which provided better insights of the data like 50 day moving average, previous day difference, etc. To prune out less useful features, in Feature Selection, we select features according to the k highest scores, with the help of an linear model for testing the effect of a single regressor, sequentially for many regressors. We used the Select K Best Algorithm, with f regression as the scorer for evaluation. Furthermore, we added Twitters Daily Sentiment Score, as an feature for each company based upon the users tweets about that particular company and also the tweets on that company’s page.

c. Usage of Ensemble Models and Algorithms

In this project we have incorporated Ensemble methods, these are the models which are composed of multiple weaker models that are independently trained and whose predictions are combined in some way to make the overall prediction. There has been much effort which is being laid on what types of weak learners to combine and the ways in which to combine them. Some the ensemble learning methods which has been incorporated in our project are depicted in Fig 2. Adaboost Regressor, Fig 3. K Neighbours Regressor and Fig 4. Random Forest Regressor.

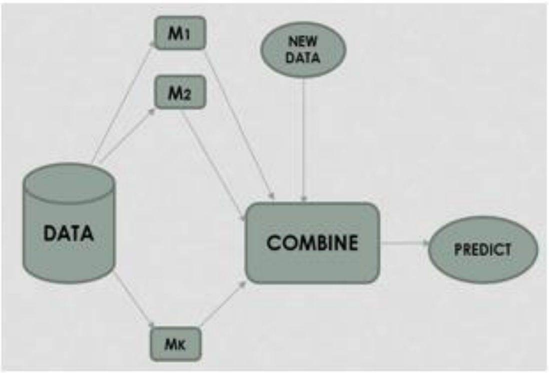

Fig 2. Adaboost Regressor

To briefly state each of these methods, Adaboost uses stumps to classify, weak learners are being used however certain drawbacks are associated with Adaboost which is they are sensitive to outliers, can't scale up and does over fit.

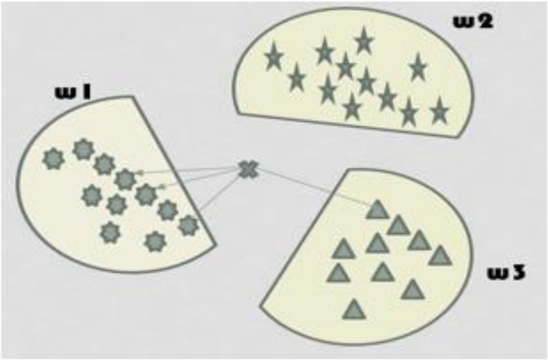

Fig 3. K Neighbours Regressor

In terms of K Neighbours they work well for small datasets, simple and easily interpretable, no training is required but it comes with a setback as the dimensions are prone to error, these are basically slow algorithms, sensitive to outliers and cannot deal with missing values hence data cleaning would be difficult to handle.

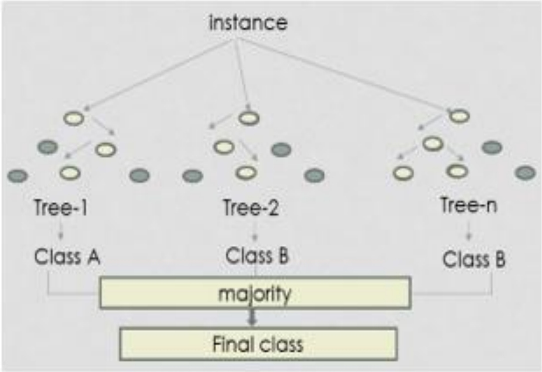

Fig 4. Random Forest Regressor

We choose Random Forest Regressor has our main model in contention to predict the forecasting since Random Forest model is good for both classification and regression hence solving most of our prediction requirement, it also does not over fit and quite good for higher dimensionality. Random Forest regressors maintain accuracy which is most important attribute to maintain efficiency in our proposed model.

4 ANALYSIS AND RESULTS

To analyze the efficiency of the project we used Root Mean Square Error(RMSE) and R2 Score Value.

1. Root Mean Square Error(RMSE) – It can be defined as The square root of the mean or average of the square of all of the error. The usage of RMSE is very widespread and it makes an outstanding general purpose error metric for numerical predictions. In comparison to the similar Mean Absolute Error, RMSE amplifies and severely punishes large errors.
2. R-Squared Value(r^2 value) – The value of R2 ranges between 0 and 1, and the higher the value the more accurate the regression model is since the more variability is explained by the linear regression model. R2 value indicates the proportionate amount of variation in the response variable explained by the independent variables. R-squared is a statistical measure of how close the data are to the fitted regression line. It is also known as the coefficient of determination, or the coefficient of multiple determination for multiple regression.

Algorithm	RMSE Value	R Squared Value
Random Regressor	1.4325434e-07	0.956669
Adaboost Regressor	2.9882972e-07	0.909611
K Neighbours regressor	0.00039015 e-07	-11.01176

Table 1. Classifier Evaluation

Based on the results obtained in Table 1, it is found that the performance of the regressors are in the following order, Random Forest Regressor, Adaboost Regressor and K Neighbour Regressor. Bagging Bootstrap sampling relies on the fact that combination of many independent base learners will significantly decrease the error. Therefore we want to produce as many independent base learners as possible. Each base learner is generated by sampling the original data set with replacement. From the results, it is safe to say that additional hidden layer(s) improve upon the score of the models. Random Forest is an extension of bagging where the major difference is the incorporation of randomized feature selection.

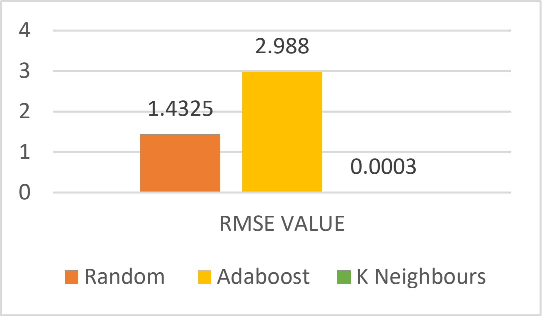

Chart A

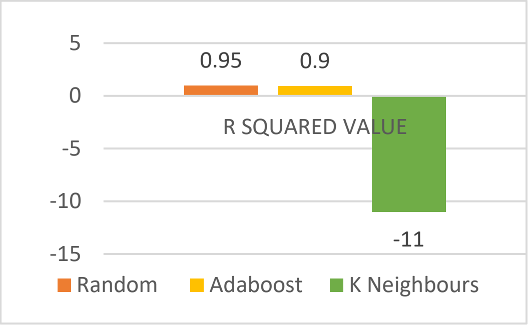

Chart B

The above chart A and Chart B depicts the comparison of different models using RMSE and R-Squared values. Random Forest regressor comes on top in terms of R-Squared value. In reference to this, we also look at some of the below graphs generated through our Stock Prediction application.

By using the above stated models and methodology, this application generates graphs with actual and predicted values which shows the prediction from our model is slightly better than the previous model. The previous model used the data from different stock indices such as NASDAQ, DJIA, DAX and so on whereas the our model uses data from NASDAQ and only with regards to S&P 500.

Fig 5, Fig 6 and Fig 7 depicts three different models – Adaboost, KNeighbors and Random Forest Regressor respectively. An actual value and predicted value shows a clear distinction among these algorithms in terms of accuracy in forecasting stocks where random forest model is closest in being accurate and efficient model.

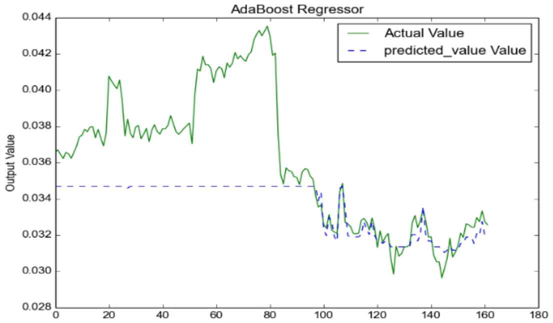

Fig 5. Adaboost Regressor

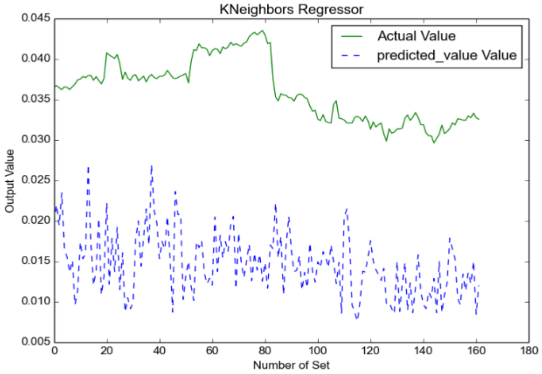

Fig 6. KNeighbors Regressor

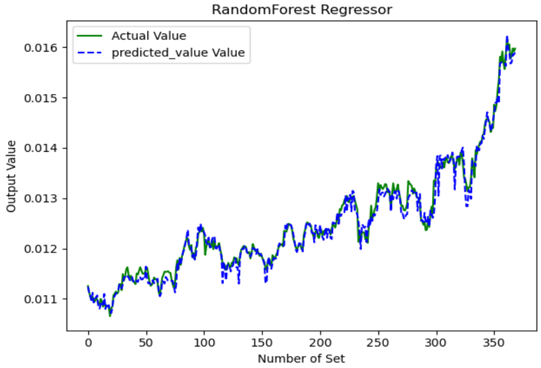

Fig 7. Random Forest Regressor

5 CONCLUSION

In this project, we implemented an application to predict stock market trends of S&P 500 financial markets from the data collected on NASDAQ. We applied various data pre-processing strategies to ensure that there are no missing values in the data to avoid potential glitches during the execution of our application. We extracted several features by using Select K Nearest algorithm. The important aspect of this project is the usage of sentiment analysis on Twitter. With this, through the usage of ensemble learning models such as such as Adaboost regressors, K Neighbor Regressor and Random Forest Regressors to predict the stock trend, we interestingly found Random Forest to be more accurate in terms of forecasting the future value. To analyze the efficiency of the project we used Root Mean Square Error(RMSE) and R2 Score Value.

The *key features or contributions* of this project is to accurately forecast the direction of stock in short term and forecast approximately over long term by utilizing different models, utilize twitter sentiment analysis, stock market data available on NASDAQ, Yahoo and pandemic data available on various sites notably Worldometer and visually represent the output data using a graphical charts by using machine learning models which is in-line with the research and analytics. The S&P Stock trend forecasting application has been maintained on GitHub.
https://github.com/vinaylokesh/MachineLearning2020
Our learnings through the project has been incredible and it can be summarized into four aspects:

1) Application oriented stock analysis rather than theory oriented analysis have an edge in terms of profit since an application provides greater insights in a short amount of time with accuracy and efficiency.

2) Various machine learning based models are proposed for predicting daily trend of US stocks. Numerical results suggests high accuracy.

3) If a practical trading model is built upon the well trained predictor, it could make this model generate higher profit compared to selected benchmarks.

4) Sentiment analysis has a wide range of benefit as it directly impacts the interest in stock market.

6 FUTURE ENHANCEMENT

Though the proposed model has implemented the sentiment analysis from Twitter, usage of paid versions of twitter API's could ease the feature extraction and it could possibly enable superior forecasting and analytics . Deep Learning has been productive in terms of current research, if some of those aspects can be combined with the current model, it could enhance the model's accuracy and efficiency. The current model forecasts data based on the daily basis, if an hourly basis prediction is implemented, it could provide better insights. Overall, though the application generates graph, it is still based on the core of command line interface, if this project can go live as a web-application, in other terms as a complete Graphical user Interface (GUI), it will ease the selection of algorithms and provides a clear cut way to differentiate among different stock market trends and forecasting.

References

[1] International Journal of Economics and Finance; Vol.5, No. 3; 2013 ISSN 1916-971X E-ISSN 1916-9728

[2] W. Huang et al., "Forecasting stock market movement direction with support vector machine," Computers Operations Research, 32, pp. 2513–2522005, 2005

[3] Vatsal H. Shah, "Machine learning techniques for stock prediction," www.vatsals.com.

[4] J. Moody, et al., "Learning to trade via direct reinforcement, "IEEE Transactions on Neural Networks, vol.12, no. 4, Jul. 2001.

[5] S. Zemke, "On developing a financial prediction system: Pitfall and possibilities," Proceedings of DMLL- 2002 Workshop, ICML, Sydney, Australia, 2002.

[6] Refenes, A. N., Zapranis, A., and Francis, G. Stock performance modeling using neural networks: a comparative study with regression models. Neural Networks 7 (1994), 375–388.

[7] Levin, A. U. Stock selection via nonlinear multi-factor models. In NIPS. 1996.

[8] Ghosn, J., and Bengio, Y. Multi-task learning for stock selection. In NIPS. 1997.

[9] Tsai, C.-F., Lin, Y.-C., Yen, D. C., and Chen, Y.-M. Predicting stock returns by classifier ensembles. Applied Soft Computing 11 (2011), 2452–2459.

[10] Leung, M. T., Daouk, H., and Chen, A.-S. Forecasting stock indices: a comparison of classification and level estimation model. International Journal of Forecasting 16 (2000), 173–190.