Stock Price Prediction using Machine Learning Algorithms

S.D. Gokhale[1], Pranav S.M.[2], Pritam S.A.[3], Prathamesh M.C.[4], Hritik S.A.[5]

[1] *Associate Professor , Department of Electronics Engineering ,DKTE'S Textile and Engineering Institute Ichalkaranji ,Maharashtra, India*
[2] *Electronics Engineer , Department of Electronics Engineering ,DKTE'S Textile and Engineering Institute Ichalkaranji , Maharashtra, India*
[3] *Electronics Engineer, Department of Electronics Engineering ,DKTE'S Textile and Engineering Institute Ichalkaranji , Maharashtra, India*
[4] *Electronics Engineer , Department of Electronics Engineering ,DKTE'S Textile and Engineering Institute Ichalkaranji , Maharashtra, India*
[5] *Electronics Engineer , Department of Electronics Engineering ,DKTE'S Textile and Engineering Institute Ichalkaranji , Maharashtra, India*

ABSTRACT

Investment firms, hedge funds and even individuals have been using financial models to better understand market behavior and make profitable investments and trades. A wealth of information is available in the form of historical stock prices and company performance data, suitable for machine learning algorithms to process. Can we actually predict stock prices with machine learning? Investors make educated guesses by analyzing data. They'll read the news, study the company history, industry trends and other lots of data points that go into making a prediction.

For data with timeframes recurrent neural networks (RNNs) come in handy but recent researches have shown that LSTM, networks are the most popular and useful variants of RNNs.

We have used the Keras to build a LSTM to predict stock prices using historical closing price and trading volume and visualize both the predicted price values over time and the optimal parameters for the model.

INTRODUCTION

Predicting how the stock market will perform is one of the most difficult things to do. There are so many factors involved in the prediction – physical factors vs. psychological, rational and irrational behaviour, etc. All these aspects combine to make share prices volatile and very difficult to predict with a high degree of accuracy.

Can we use machine learning as a game-changer in this domain? Using features like the latest announcements about an organization, their quarterly revenue results, etc., machine learning techniques have the potential to unearth patterns and insights we didn't see before, and these can be used to make unerringly accurate predictions.The core idea behind this article is to showcase how these algorithms are implemented. I will briefly describe the technique and provide relevant links to brush up on the concepts as and when necessary.

1. SOFTWARE REQUIREMENT SPECIFICATION

1.1 FUNCTIONAL REQUIREMENTS:

Functional requirements deal with the functionality of the software in the engineering view. The component flow and the structural flow of the same is enhanced and described by it.

The functional statement deals with the raw datasets that are categorized and learning from the same dataset. Later the datasets are categorized into clusters and the impairment of the same is checked for the efficiency purpose. After the dataset cleaning the data are cleansed and the machine learns and finds the pattern set for the same it undergoes various iteration and produce output.

1.2 NON-FUNCTIONAL REQUIREMENTS:

Non functional requirement deals with the external factors which are non functional in nature It is used for analysis purpose. Under the same the judgment of the operations is carried out for its performance. Stock is feasible and is ever changing so these extra effects and the requirements helps it to get the latest updates and integrate in a one goes where the technicians can work on and solve a bug or a draft if any.

The non-functional requirements followed are its efficiency and hit gain ratio. The usability of the code for the further effectiveness and to implement and look for the security console. The System is reliable and the performance is maintained with the support of integration and portability of the same.

2. DATA PREPROCESSING

Data pre-processing is a process of preparing the raw data and making it suitable for a machine learning model. It is the first and crucial step while creating a machine learning model.

When creating a machine learning project, it is not always a case that we come across the clean and formatted data. And while doing any operation with data, it is mandatory to clean it and put in a formatted way. So, for this, we use data pre-processing task.

2.1 Why do we need Data Pre-processing?

A real-world data generally contains noises, missing values, and maybe in an unusable format which cannot be directly used for machine learning models. Data pre-processing is required tasks for cleaning the data and making it suitable for a machine learning model which also increases the accuracy and efficiency of a machine learning model.

It involves below steps:

- Getting the dataset
- Importing libraries
- Importing datasets
- Finding Missing Data
- Encoding Categorical Data
- Splitting dataset into training and test set
- Feature scaling

3. ALGORITHMS

RNN is a Supervised Deep Learning used for Time Series Analysis. Recurrent Neural Networks represent one of the most advanced algorithms that exist in the world of supervised deep learning. The idea is that weights have Long Term Memory abbreviated as LTM. For example, in a classical ANN known the weights, and so whatever input, we put into the ANN, it will process it in the same way as it would yesterday. The weights can be located in the Temporal Lobe of a human brain, because the Temporal Lobe is responsible for Long Term Memory LTM. RNN is like Short Term Memory, because it can remember things that are just happened in the previous couple of observations. The figure below is a representation of RNN.

3.1 Long Short-Term Memory (LSTM):

Sequence prediction problems have been around for a long time. They are considered as one of the hardest problems to solve in the data science industry. These include a wide range of problems; from predicting sales to finding patterns in stock markets' data, from understanding movie plots to recognizing your way of speech, from language translations to predicting your next word on your iPhone's keyboard. With the recent breakthroughs that have been happening in data science, it is found that for almost all of these sequence prediction problems, long short-Term Memory networks, LSTMs have been observed as the most effective solution. LSTMs have an edge over conventional feed-forward neural networks and RNN in many ways. This is because of their property of selectively remembering patterns for long durations of time.

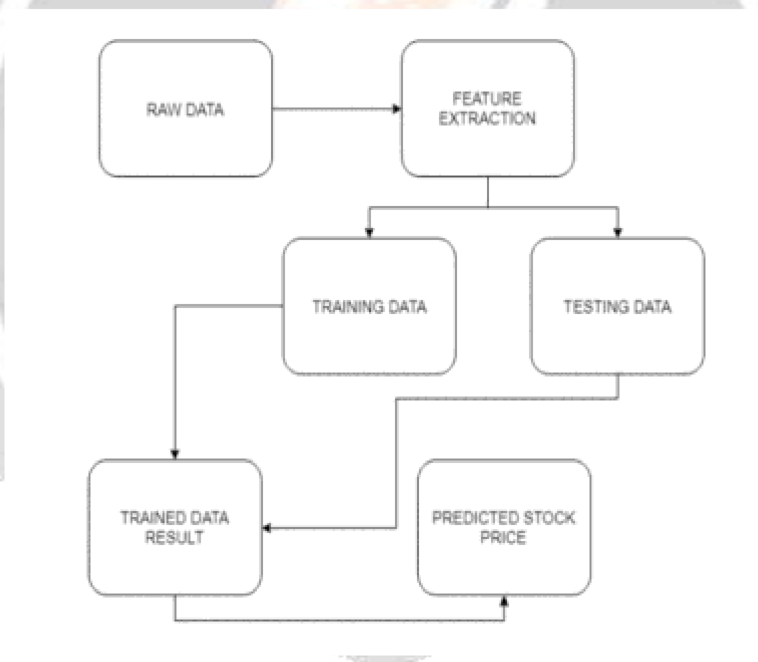

Fig -1: Block Diagram Of LSTM

The purpose of this article is to explain LSTM and enable us to use it in real life problems.

LSTMs on the other hand, make small modifications to the information by multiplications and additions. With LSTMs, the information flows through a mechanism known as cell states. This way, LSTMs can selectively remember or forget things. The information at a particular cell state has three different dependencies. Industries use them to move products around for different processes. LSTMs use this mechanism to move information around.

4. MODEL EVALUATION AND VALIDATION

Once the model is built, the next step is to evaluate and validate the model. Model Evaluation is an essential part of the model development process. It is used to test the final performance of the algorithm and is done on the test set. Also, it helps to find the best model that represents your data and how well the chosen model will work in the future.

Model validation is the set of processes and activities intended to confirm that models are performing as expected. Effective helps you to ensure that models are sound validation .

There are multiple measures that can be used to find out how good a regression model is predicting or how good a classifier is classifying the data.

Model Evaluation is an integral part of the model development process. It helps to find the best model that represents our data and how well the chosen model will work in the future. Evaluating model performance with the data used for training is not acceptable in data science because it can easily generate overoptimistic and overfitted models. There are two methods of evaluating models in data science, Hold-Out and Cross-Validation. To avoid overfitting, both methods use a test set (not seen by the model) to evaluate model performance.

4.1 Hold-Out: In this method, the mostly large dataset is *randomly* divided to three subsets:

1. **Training set** is a subset of the dataset used to build predictive models.

2. **Validation set** is a subset of the dataset used to assess the performance of model built in the training phase. It provides a test platform for fine tuning model's parameters and selecting the best-performing model. Not all modelling algorithms need a validation set.

3. **Test set** or unseen examples is a subset of the dataset to assess the likely future performance of a model. If a model fit to the training set much better than it fits the test set, overfitting is probably the cause.

	open	high	...	buy_profit	sell_profit
2021-12-08	172.130005	175.960007	...	0.0	-3.116028
2021-12-13	181.119995	182.130005	...	0.0	-3.954956
2021-12-15	175.110001	179.500000	...	0.0	7.290710
2021-12-16	179.279999	181.139999	...	0.0	0.089874
2021-12-22	173.039993	175.860001	...	0.0	3.445602
2021-12-28	180.160004	181.330002	...	0.0	13.043381
2022-01-06	172.699997	175.300003	...	0.0	1.667877
2022-01-12	176.119995	177.179993	...	0.0	2.626648
2022-01-19	170.000000	171.080002	...	0.0	-10.261520
2022-01-27	162.449997	163.839996	...	0.0	-9.862595

Fig -2: Prediction Results

- In the above figure we can see the graph we got from the results show us the actual price and the predicted prices in their corresponding colours of blue and red respectively.
- By taking a particular stock we can also get its opening price and the high price of the stock(the moment at which the price hit its max. throughout the day) and the buying profit and the selling profit.

- We developed LSTM program using Python and TensorFlow for stock prices prediction. We use 80% for training data and 20% for testing data, and the result shown in above figure.

- Detailed result of prediction based on the 80% training set and 20% testing set. It shows the significant accuracy from 5 to 20 epochs.

5. Hardware Requirements

The Stock Price Prediction project is simple, reliable, and a user-friendly project which can result to profitable outcome for an individual who is interested to invest in stocks and for the ones who are expecting for higher gains from the stock market.

The functional/ hardware requirements needed for the user are as follows:
Processor : Intel i5 or above
RAM :Minimum 225MB or more.
Hard Disk : Minimum 2 GB of space
Input Device : Keyboard
Output Device : Screens of Monitor or a Laptop

6. CONCLUSION

However, with the introduction of Machine Learning and its strong algorithms, the most recent market research and Stock Market Prediction advancements have begun to include such approaches in analyzing stock market data. The Opening Value of the stock, the Highest and Lowest values of that stock on the same days, as well as the Closing Value at the end of the day, are all indicated for each date. With this information, it is up to the job of a Machine Learning Data Scientist to look at the data and develop different algorithms that may help in finding appropriate stocks values.

Predicting the stock market was a time-consuming and laborious procedure a few years or even a decade ago. However, with the application of machine learning for stock market forecasts, the procedure has become much simpler. Machine learning not only saves time and resources but also outperforms people in terms of performance. it will always prefer to use a trained computer algorithm since it will advise you based only on facts, numbers, and data and will not factor in emotions or prejudice.

7. REFERENCES

[1]. Podding, T and Rehkugler, H, “A world of integrated financial markets using artificial neural networks”,Nerocomputing, pages 251-273, 1996.
[2]. Nikfarjam, A. , Emadzadeh, E., and Muthaiyah , S. (2010). Text mining approaches for stock market prediction. In Computer and Automation Engineering (ICCAE), 2010
[3]. Fenglong Ma, Radha Chitta, Jing Zhou, Quanzeng You, Tong Sun, and Jing Gao. Dipole: Diagnosis prediction in healthcare via attention-based bidirectional recurrent neural networks. In Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD ’17, pages 1903–1911. ACM, 2017
[4]. Yao Qin, Dongjin Song, Haifeng Cheng, Wei Cheng, Guofei Jiang, and Garrison W. Cottrell. A dual-stage attention-based recurrent neural network for time series prediction. CoRR, abs/1704.02971, 2017.