### **Name: R.Leena**

Email id: [rleena0410@gmail.com](mailto:rleena0410@gmail.com) Collegecode : 8201

**Stock price prediction**

Phase 4: Development Part 2

In this part i will continue building my project.

Continue building the stock price prediction model by

Feature engineering

Model training

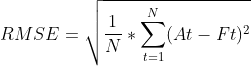
Evaluation.

Dataset Link: https://www.kaggle.com/datasets/prasoonkottarathil/microsoft-lifetime-stocks-dataset

### **Evaluating Prediction Performance for Stock Price Prediction**

Before putting the algorithms into practice, let’s clarify the metric to measure the performance of our models. Stock price prediction being a fundamental regression problem, we can use RMSE (Root Mean Squared Error) or MAPE (Mean Absolute Percentage Error) to measure how close or far off our price predictions are from the real world.

Looking closely at the formula of RMSE, we can see how we will be able to consider the difference (or error) between the actual (At) and predicted (Ft) price values for all N timestamps and get an absolute measure of error.



On the other hand, MAPE looks at the error concerning the true value – it will measure relatively how far off the predicted values are from the truth instead of considering the actual difference. This is a good measure to keep the error ranges in check if we deal with too large or small values. For instance, RMSE for values in the range of 10e6 might blow out of proportion, whereas MAPE will keep error in a fixed range.

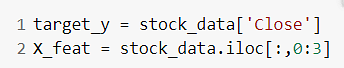
MAPE

## **Stock Market Prediction using Machine Learning Project Code**

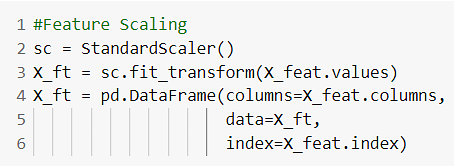
First, we will implement a simple LSTM network using [Keras](https://www.projectpro.io/article/keras-for-deep-learning/830" \o "Keras" \t "_blank) in Python. Let’s take a look at the Stock Prediction using Machine Learning dataset. We can work on actual stock data from major public companies such as Facebook, Microsoft, or Apple by simply downloading the data from [finance.yahoo.com](https://finance.yahoo.com/).

data. This will avoid features with larger numeric values to unjustly interfere and bias the model and help achieve rapid convergence in the machine learning stock prediction project.

First, we define the features and the target as discussed above.



Next, we use a StandardScaler to rescale our values between -1 and 1.



Scikit-learn also provides a popular MinMaxScaler preprocessing module. However, considering the context, stock prices might max out or minimise on different days, and using those values to influence others might not be great. The change in values from using either of these methods would not be much, so we stick to StandardScaler.

RMSE and MAPE values

Thus we observe substantial improvement by adding another LSTM layer to the model. However, further adding even more layers would not be fruitful as the model might overfit or stagnate during training.

Explore More Data Science and Machine Learning Projects for Practice. Fast-Track Your Career Transition with ProjectPro

Now we will try fitting the same model but with increased time steps. We’ll try for n\_steps=10.

We change the value in the block below and rerun the entire process with the same model as before.

How to predict the stock market

The model we used above:

RMSE and MAPE values

Surprisingly, we get similar performance as before!

Get similar performance

Rmse

We note that LSTM was able to achieve decent RMSE and MAPE values despite the data complexity. Further, we note that creating even deeper networks did not help improve the test performance of the stock price prediction model.

Before we conclude, as promised earlier, let’s look at how better or worse LSTMs perform compared with statistical techniques such as SMA and EMA.