Stock Price Prediction

### stock_market_prediction_presentation_slide_design_836101.jpg

### Stock Price as a Time Series Data

Treating stock data as [time-series](https://www.projectpro.io/article/time-series-projects/444), one can use past stock prices (and other parameters) to predict the stock prices for the next day or week. Machine learning models such as Recurrent Neural Networks (RNNs) or LSTMs are popular models applied to [predicting](https://www.projectpro.io/article/predictive-financial-modeling-projects/611) time series data such as weather forecasting, election results, house prices, and, of course, stock prices. The idea is to weigh out the importance of recent and older data and determine which parameters affect the “current” or “next” day prices the most. The [machine learning model](https://www.projectpro.io/article/common-machine-learning-algorithms-for-beginners/202)assigns weights to each market feature and determines how much history the model should look at for stock market prediction using machine learning project to work out.

Evolution of [Machine Learning Applications in Finance](https://www.projectpro.io/article/projects-on-machine-learning-applications-in-finance/510) : From Theory to Practice

**Stock Price Prediction using**[**Moving Average Time Series**](https://www.projectpro.io/article/moving-average-time-series-model/716)

**To begin with, we can use moving averages (or MA) to understand how the amount of history (or the number of past data points) considered affects the model's performance. A simple moving average computes the mean of the past N data points and takes this value as the predicted N+1 value.**

**So,**

Stock Price Prediction using Moving Average Time Series 

Where P1 to Pn are n immediate data points that occur before the present, so to predict the present data point, we take the SMA of the size n (meaning that we see up to n data points in the past). The SMA is our predicted value. The precision of the model will vary significantly with the choice of n. Higher n would mean that we are willing to go deeper into the past to compute the present value. For example, n=2 means that we take the average of the stock price of the past two days, while n=50 would consider 50 days' worth of stock prices. Obviously, 50 days’ worth of data will have more information about the trends of the stock and would lead to better predictions. However, based on context, a large n can also destabilize the model as the more granular fluctuations are smoothened off – looking at prices from the past 300 days would be sub-optimal.

Another moving average is the exponential moving average (EMA), giving more weight to the more recent samples. With this, we can look at more data points in the past and still not diminish the more recent trends in fluctuations.

exponential moving average

Where Pt is the price at time t and k is the weight given to that data point. EMA(t-1) represents the value computed from the past t-1 points. Clearly, this would perform better than a simple MA. The weight k is computed as k = 2/(N+1).

While implementing these methods, we will see how EMA performs better than SMA, proving that assigning higher weights to more recent data points will yield more fruitful results. But for now, let us assume that that is the case with stock prices as time series data.

So considering more past data and giving more importance to newer samples, EMA performs better than SMA. However, given the static nature of its parameters, EMA might not perform well for all cases. In EMA, we have fixed the value of k (the weight/significance of past data), and it is linked with the window size N (how much past we wish to consider).

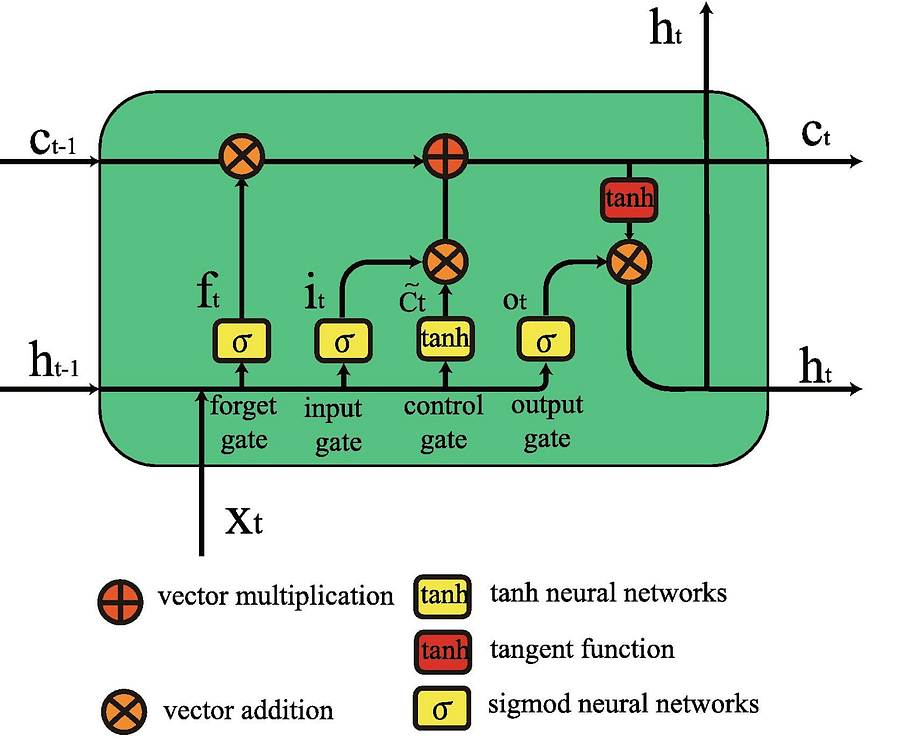
It can be difficult to set these parameters manually and impossible to optimize for this project on stock market prediction using machine learning. Thus, we can use more complex models that can compute the significance of each past data point and optimize our predictions. This can be achieved with weight updation while training a machine learning model. And thinking of using past data to compute the future, the most immediate model that comes to mind is the [LSTM](https://www.projectpro.io/project-use-case/time-series-forecasting-deep-learning) model!

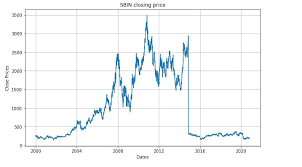
**>Get Closer To Your Dream of Becoming a Data Scientist with 150+ Solved**[**End-to-End ML Projects**](https://www.projectpro.io/projects?utm_source=TXTCTA1&utm_medium=ConvertLink&utm_campaign=Blog571)

**Understanding Long Short Term Memory Network for Stock Price Prediction**

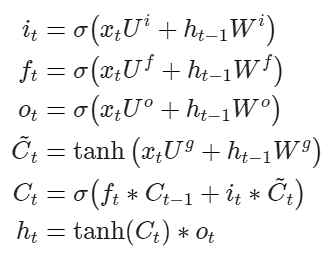
LSTM is a Recurrent [Neural Network](https://www.projectpro.io/article/neural-network-projects/440) that works on data sequences, learning to retain only relevant information from a time window. New information the network learns is added to a “memory” that gets updated with each timestep based on how significant the new sample seems to the model. Over the years, LSTM has revolutionized speech and handwriting recognition, language understanding, forecasting, and several other applications that have become the new normal today.

A standard LSTM cell comprises of three gates: the input, output, and forget gate. These gates learn their weights and determine how much of the current data sampleshould be remembered and how much of the past learned content should be forgotten. This simple structure is an improvement over the previous and similar RNN model.





As seen in the equations below, i, f, and o represent the three gates: input, forget, and output. C is the cell state that preserves the learned data, which is given as output h. All of this is computed for each timestamp t, considering the learned data from timestamp (t-1).



The forget gate decides what information and how much of it can be erased from the current cell state, while the input gate decides what will be added to the current cell state. The output gate, used in the final equation, controls the magnitude of output computed by the first two gates.

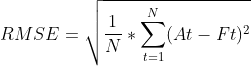
So, as opposed to standard feed-forward neural nets, LSTMs have the potential to remember or erase portions of the past data windows actively. Its feature of reading and training on windows (or timesteps) of data makes its training unique. Let’s build the model in Python.

**Upskill yourself for your dream job with industry-level**[**big data projects with source code**](https://www.projectpro.io/projects/big-data-projects?utm_source=TXTCTA1&utm_medium=ConvertLink&utm_campaign=Blg297)

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) 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/).

Downloading the Stock Prices Dataset for Project on Machine Learning Price Prediction of Stocks

Go to [finance.yahoo.com/](https://finance.yahoo.com/) and search the company whose data you want to seek for stock price prediction. For our example, we will look at the Netflix (NFLX) stock over 3 years.

Going to [finance.yahoo.com/quote/NFLX/history?p=NFLX](https://finance.yahoo.com/quote/NFLX/history?p=NFLX) in the “Historical Data” section, we see the stock data listed each day. We can filter out the time for which we wish to analyze and download the CSV file using the download button on the right.

