Stock market forecasting using statistical tools.

Sarthak Joshi

Abstract: Stock market forecasting is a timeseries data fitting and forecasting problem that can be solved using various ways. Here I have shown three different methods to forecast the data; polynomial regression, ARIMA and LSTM RNN in increasing level of complexity and accuracy. I have used data from National stock exchange of India.

Introduction

Stock market forecasting is a well-known problem and has been worked on a lot. There have been various methods to predict it. Here I have shown how to fit the data using three different methods. Polynomial regression (linear model) with time as a feature and Closing price as the target variable; Auto-Regressive Integrated Moving Average model (ARIMA) with automatic hyperparameter prediction using auto_arima from pmdarima package for python; and long short-term memory recursive neural network using TensorFlow library from google.

Polynomial regression

Polynomial regression is a form of regression analysis in which the relationship between the feature x and the target variable y is modelled as an n^{th} degree polynomial in x.

$$y = \beta_0 + \beta_1 x + \beta_2 x^2 + \cdots \beta_n x^n$$

And

$$\widehat{\vec{\beta}} = \left(\mathbf{X}^{\mathsf{T}}\mathbf{X}\right)^{-1} \mathbf{X}^{\mathsf{T}}\vec{y},$$

ARIMA

Auto-Regressive Integrated Moving Average model (ARIMA) is a class of time series prediction models. The backbone of ARIMA is a mathematical model that represents the time series values using its past values. ARIMA is defined by three parameters p, d, and q that describe the three main components of the model. The model can be represented by the following equation.

$$\left(1 - \sum_{i=1}^{p'} \alpha_i L^i\right) X_t = \left(1 + \sum_{i=1}^{q} \theta_i L^i\right) \varepsilon_t$$

Where, L is the lag operator, α_i are the parameters of the autoregressive part of the model, the Θ_i are the parameters of the moving average part and the $\mathcal{E}_t$ are error terms.

LSTM RNN

Long short-term memory recursive neural networks are a type of recursive neural networks where the individual cells of a RNN have a complex interaction of four layers. From these layers, there is a derived cell state that runs down then entire recursion and allows flow of information relatively unchanged as compared to a normal RNN. The cell state can be changed but is carefully regulated using three gates. This architecture allows the cells to have a memory.

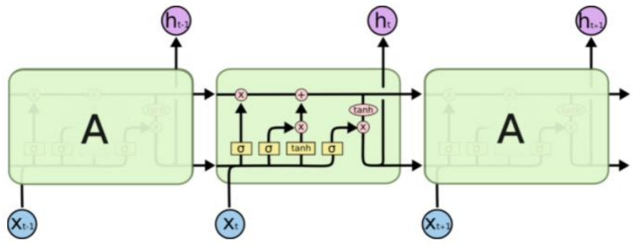


Fig: Repeating module in LSTM showing the 4 layers of the cell. (Source: http://colah.github.io/posts/2015-08-Understanding-LSTMs/)

Data and Methods

The historical data of the NIFTY50 was obtained from www.nse.com using package nsepy from www.nsepy.xyz. Polynomial regression was done using sklearn package. LSTM networks were build using TensorFlow keras package.

Results and Conclusions

Polynomial regression

To test the model, I tried to fit all the stocks from nifty50. (n=50) using polynomial regression (order=5). Average RMSD of the model for the 50 stocks was 1250.

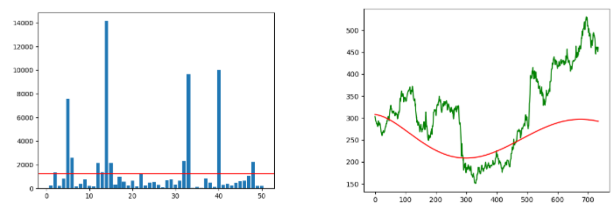
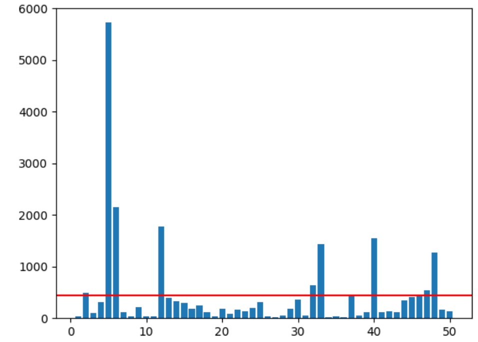


Fig: RMSD of predicted vs known values of stocks by fitting the data to a polynomial of the order 5. (Stocks: NIFTY50). curve obtained by fitting the data to a polynomial of the order 5. Stock: SBIN (NSE)

ARIMA

To test the model, I tried to fit all the stocks from nifty50. (n=50) using ARIMA. Average RMSD of the model for the 50 stocks was 447.

Fig: RMSD of predicted vs known values of stocks by fitting the data to ARIMA. Stocks: NIFTY50



LSTM RNN

To test the model, I tried to fit 10 stocks from nifty50. (n=10) using the network previously described. Average RMSD of the model for the 10 stocks was 109.7.

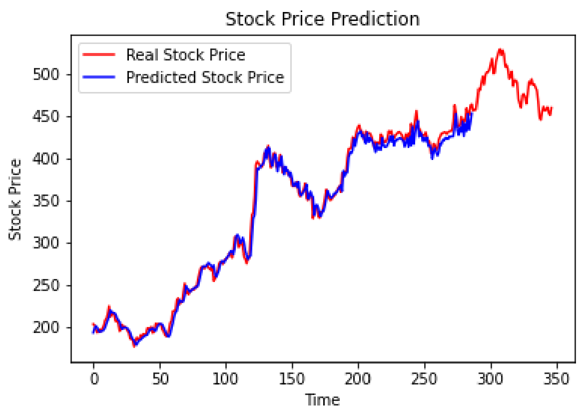


Fig: Observed trend and predicted trend of the stock test dataset. Prediction using LSTM RNNStock: SBIN (NSE)

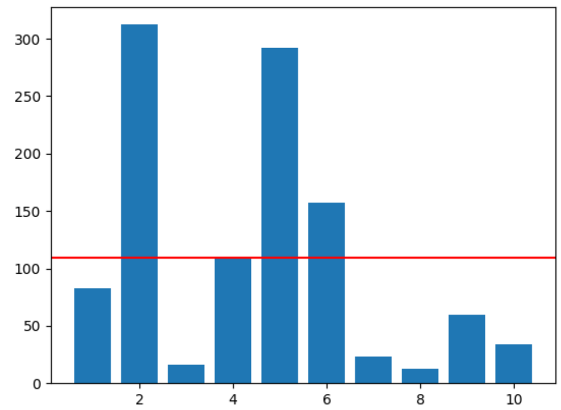


Fig: RMSD of predicted vs known values of stocks by fitting the data to LSTM RNN. Stocks: NIFTY50.

These are very effective methods to predict the stock market, but it needs a lot of further refining and analysis. ARIMA and LSTM RNNs could both be used as tools to do that.

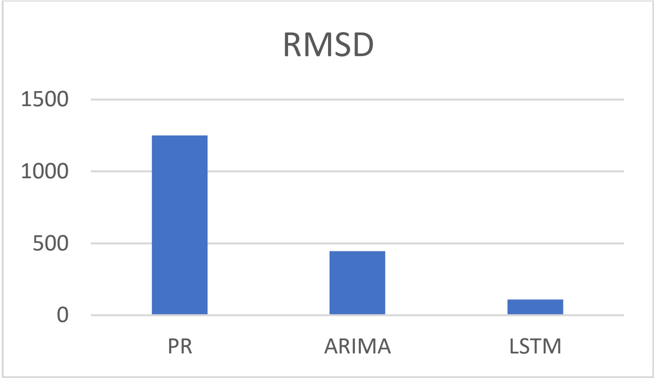


Fig: RMSD of all the three models compared.

References

1. http://colah.github.io/posts/2015-08-Understanding-LSTMs/
2. https://neptune.ai/blog/arima-vs-prophet-vs-lstm
3. https://neptune.ai/blog/arima-sarima-real-world-time-series-forecasting-guide
4. https://otexts.com/fpp2/arima.html
5. My code: https://github.com/SixEyedKnight/stockmarket