Venkatesh Venkatraman

Udacity Machine Learning Engineering Nanodegree

Proposal of predicting stock prices using LSTM

June 11,2019

Domain Background:

Investments are very great tool for making money in this growing financial world.Traditional investment which every common people know is like Fixed deposit, recurring deposit, national pension scheme.These are the investment being followed by most of the people.We do have share market which is very big sector where returns would be unlimited.We have all the history of data available in web.

Is it possible to predict the stock market price?
Of course, it is possible.We have two kinds of analysis which are fundamental analysis and technical analysis. Fundamental analysis are based on inside news of the organization.Technical analysis are based on the history of data available.In this project we can prefer technical analysis for predicting future stock prices.Current stock market industry is migrating towards algorithmic trading.

In a research of ***Support Vector Machines for Prediction of Futures Prices in Indian Stock Market***, Shom Prasad Das and Sudarsan Padhy discuss Back Propagation Technique and Support Vector Machine Technique to predict futures prices traded in Indian stock market. The reported NSME for all the futures stock index taken into consideration fall in the range of 0.9299 to 1.1521, which is a decent

result. In a study of ***A deep learning framework for financial time series using stacked auto encoders and long-short term memory***, Wei Boa, Jun Yue and Yulei Rao discuss combining wavelet transforms (WT), stacked auto encoders (SAEs) and long-short term memory (LSTM) to forecast stock index prices. The performance is reported across a year worth of data across various types of market and report Mean absolute percentage error (MAPE), correlation coefficient (R) and Theil's inequality coefficient (Theil U). In developed market they predict CSI 300 index, with average value of MAPE and Theil U of WSAEs-LSTM as 0.019 and 0.013. It seems like machine learning and even deep learning methods can yield good result in stock price prediction.

ACADEMIC LINK:
https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0180944#sec002

INPUT DATASET LINK:
https://www.nseindia.com/products/content/equities/indices/historical_index_data.htm

Problem Statement:

We need to predict the future price of the Indian index stock Bank Nifty.It represents index of some of the top private and government banks in india.
This project will predict the stock price of the bank nifty for given future period.Based this prediction result, user can plan for his investments and for return on investment for the proposed period.User can also use this prediction for option trading where the investment would be constant but profit is unlimited.For this problem i will use Long short term memory networks to predict the closing price of BANK NIFTY by using its history of data.

Goals:

1.Explore the stock price

2.Implement linear regression model

3.Implement LSTM using Keras

4.Compare the results and report submission

Datasets and Inputs:

I will be using the daily prices of BANK NIFTY for past 10 years.This is a series of data indexed in time order.Aim of this project is to predict the closing of the stock for the given time after training the model.Sample of the dataset is below

Historical Data for NIFTY BANK						
For the period 01-07-2018 to 12-06-2019						
Date	Open	High	Low	Close	Shares Traded	Turnover (₹ Cr)
02-Jul-2018	26364.45	26371.60	26060.60	26230.30	174872280	4472.48
03-Jul-2018	26210.10	26316.10	26142.00	26204.10	104588044	2547.80
04-Jul-2018	26249.25	26480.50	26132.90	26433.95	82888322	2201.98
05-Jul-2018	26481.55	26598.45	26415.20	26503.30	91431872	2632.52
06-Jul-2018	26427.55	26609.85	26410.00	26493.85	120291139	3037.92
09-Jul-2018	26642.35	26781.20	26611.55	26753.30	101833058	2986.56
10-Jul-2018	26844.55	26939.55	26778.50	26894.55	103574099	3700.96
11-Jul-2018	26900.65	26938.85	26774.60	26816.20	91728787	2611.23
12-Jul-2018	26937.50	27164.80	26936.05	27026.55	108520112	2970.49

Dataset will be split based on chronological order since we deal with time series data.

Solution Statememt:

For this project according to my analysis the best possible solution is to utilize a LSTM Neural Net model capable of learning from time series data. This project will be programmed in a Jupyter Notebook (iPython) for ease of reproducibility. Using a Keras implementation of the Tensor Flow library, the solution will utilize a LSTM Neural Net model and will be supported by Pandas DataFrame library for convenient

time series data schema.The measures of performance will be based on the predicted stock ticker price in comparison to both the actual price and the benchmark model's predicted price.

Benchmark Model:

We can use a Linear Regression model as its primary benchmark. As one of my goals is to understand the relative performance and implementation differences of machine learning versus deep learning models.

Evaluation Metrics:

For this project i will measure performance using the mean squared difference between predicted and actual values of the target stock at adjusted close price and the delta between the 4 performance of the benchmark model (Linear Regression) and our primary model (Deep Learning).

Project Design:

This project will be implemented through the Keras/Tensor Flow library using LSTM Neural Networks. Development workflow will follow the below sequence:

0. Set Up Infrastructure

- iPython Notebook
- Incorporate required Libraries (Keras, Tensor flow, Pandas, Matplotlib,Sklearn,Numpy)
- Git project organization

1. Prepare Dataset

- Incorporate BANKK NIFTY data
- Process the requested data into Pandas Dataframe
- Develop function for normalizing data
- Dataset will be used with a 80/20 split on training and

test data across all models

2. Develop Benchmark Model

- Set up basic Linear Regression model with Scikit-Learn
- Calibrate parameters

3. Develop Basic LSTM Model

- Set up basic LSTM model with Keras utilizing parameters from Benchmark Model

4. Improve LSTM Model

- Develop, document, and compare results using additional labels for the LSMT model

5. Document and Visualize Results

- Plot Actual, Benchmark Predicted Values, and LSTM Predicted Values per time series
- Analyze and describe results for report.

Let us call this target stock's Close price as target price. First we will use pandas-datareader module to query all the Open, High, Low, Close and Volume(Shares traded) from BANK NIFTY index, from 2009 to 2019. I will use Pearson correlation to calculate the correlation of target price and select features that has the highest correlation with the target price.We may think today's target price will only depend on the features till one-month ago, then the sequence length would be one month. This is a hyperparameter we need to tune, and I just assume it is one-month in this proposal. The sequence generation will be as follows: set a rolling window of length 20 days(approximately the number of trading days in a month), move this window along the time series, and record the data in each step as the feature sequence for the next day. This will gives us features in our feature set. After we have generated our feature sets, we can input these features into the benchmark linear regression and LSTM networks. I will use scikit-learn's LinearRegression class to do the linear regression, and use Keras's LSTM function to build the neural networks. The networks will have two or three layers, each with hidden states

dimension of 100-50 or 150-100-50. Each layer will use Relu as activation function, and each layer will have a dropout layer with keep probability of 80% in order to regularize. The output layer will have dimension of 1, and linear activation function. I will split the data into 2009 to 2016 as training, and 2017 as validation data. I will use R-square and RMSE to check the model performance, on both benchmark model and LSTM model. Once a successful model is built, the user can choose which 20 days of data they want to use as input, and the model will predict the next date's target price

NOTE : The actual implementation may vary a little from proposed design as i have not yet implemented the project. But the flow of working process will be same as mentioned above.