Udacity machine Learning Nanodegree
Capstone Project Proposal

Investment and Trading

Build a Stock Price Indicator

Krutika Bhosale
(krutika.bhosale@fractalanalytics.com)

Feb 3rd, 2019

Abstract

Many of the investment instruments like equity based insurance, futures and options, mutual funds, hedge funds depends on the share price of the underlying scrip, or index. Even the investors and speculators frame their policies of transactions in stock market based on the prices of shares. Therefore it is prudent that financial models be made, that describe the prices of shares as they can prove important in framing the instruments. The success of fund managers who manage these instruments for individuals and institutions, as well as the individual investors and speculators, depends on maximizing the profits. Financial models to describe the share prices can impart lot of information about price declines and price soarings and thus identify the events where the stock or index holding will be profitable. Here we will develop such a financia model with deep learning methods.

Domain background

To build the financial model, we have sufficient information available today, than it was a decade ago. Therefore the available information along with available enhanced computing power must be leveraged to build better financial models. [1]

The information is available in form of stock price related data, most common of them being opening price, closing price, high and low for all trading days for almost two

decades. This information is available in form of time series data and deep learning algorithms can learn a lot from such data to find out patterns and make relevant predictions. A specific type of deep learning algorithm, called LSTM (long short term memory) which is a variant of a class of deep learning algorithms called RNNs (recurrent neural networks). [2]

RNNs were designed to incorporate the time effect of data. LSTMs were designed to solve a key problem with RNNs, that RNNs were unable to store any information older than around past 100 events. The memory of RNNs was short and LSTM has longer memory from past. LSTMs can capture the most essential features of time series data and model its dependencies. [3]

Problem statement

A large financial institution Southern Horizons Bank (fictional) based in Mumbai, India has recently launched a new insurance product whose premium depends on sector average. For example, the premium for health insurance depends on how well the healthcare stocks are performing in stock market. A more general insurance plan that covers health, house and vehicle has a premium that depends on the general market's aggregate index. The insurance product has a part that fetches return from market performance. The more is the jump in index from last period, higher is the premium and the more is the fall in index, lower is the premium. This gives psychological and financial satisfaction to the customer and thus fetches more customers because if the index has a better rise, there will be more profits from the market and thus higher premiums shall work. If market falls, there will be less profits from market and thus premium shall be less. Since this premium collection is dynamic, there is no definite calculation of premium inflow to the bank. To keep the bank finances deterministic, the bank needs an idea of when will index rise and when will it fall, so that they have approximate upper and lower ranges for premium inflow available. Thus the bank wants to build a predictor for NIFTY 50 which is the flagship index of the National Stock Exchange of India (NSE) in Mumbai.

Dataset and Inputs

The data is a discrete time spaced closing prices of NIFTY 50, as obtained from the website of NSE [4]. The data consists of historical closing prices for all days market was open in last 9 months. An example snapshot of data is shown below for all information available. The data is all numeric and continuous.

Historical Data for NIFTY 50						
For the period 03-05-2018 to 03-02-2019						
Date	Open	High	Low	Close	Shares Traded	Turnover (₹ Cr)
03-May-2018	10720.15	10720.60	10647.45	10679.65	190869804	11018.21
04-May-2018	10700.45	10700.45	10601.60	10618.25	192296041	9721.56
07-May-2018	10653.15	10725.65	10635.65	10715.50	173620240	9239.44
08-May-2018	10757.90	10758.55	10689.40	10717.80	278118616	12783.49
09-May-2018	10693.35	10766.25	10689.85	10741.70	222115640	10390.92
10-May-2018	10779.65	10785.55	10705.00	10716.55	197988475	10526.38
11-May-2018	10741.95	10812.05	10724.45	10806.50	209392114	10432.43
.						
.						
23-Jan-2019	10931.05	10944.80	10811.95	10831.50	298876314	14736.91
24-Jan-2019	10844.05	10866.60	10798.65	10849.80	361082096	15298.48
25-Jan-2019	10859.75	10931.70	10756.45	10780.55	463444758	20542.36
28-Jan-2019	10792.45	10804.45	10630.95	10661.55	419682627	21144.33
29-Jan-2019	10653.70	10690.35	10583.65	10652.20	356908994	18832.06

The Nifty50, in trade since april 1996, is a very efficient benchmark index for equity and derivatives, that tracks behavior of the largest and most liquid securities in Indian stock market. It consists of 50 out of approximately 1600 companies listed on the NSE and the companies are included such that they captures approximately 65% of its float-adjusted market capitalization. This ensures that Nifty50 explains most of the Indian stock market very well. All major sectors are covered in Nifty50 which the investment managers can utilize for their general major decisions. [5]

The data is explained below.

Open: The price on which the index opened on that day.
Close: The price on which the index closed on that day.
Low: The lowest price on the index on that day.
High: The highest price of the index that day.
Shares traded: Total number of shares of the index that got transactions completed on that day.
Turnover: Total amount worth buy and sells in that day.

All the prices are in Indian Rupee.

The chart below shows the trend of Nifty50 in last 3 months.

Solution Statement

We are most interested in predicting the “Close” price. With the kind of data available, we have other information such as “Open”, “High”, “Low”, “Shares traded” and “Turnover”. We will first create a benchmark model through linear regression where Open, High, Low, Shares traded and Turnover as independent variables and Close as dependent variable. We will use data from Nov 2017 to Oct 2018 to train the model and will test it on the training data span of 12 months as well as on the last 3 months (Nov 2018 - Jan 2019) on which the linear regression model was not trained. The benchmark model will predict the value of Close and the predicted values will be mapped to one of the classes “down” or “up” depending upon whether the current Close price was lower or higher than the last Close price. We will find out precision and recall from that mapping. We will find p-score in the linear regression analysis to find out variables which impact the most and variables which are not impacting the closing price. We will use PCA to find the components in the data that describe most of the data. After this model, we will create another model over the LSTM algorithm and compare it against the benchmark model. The LSTM model would predict the Close price for Nov 2018 - Jan 2019. Like the benchmark model, these predictions will be mapped to “down” or “up” class to calculate precision and recall.

Benchmark model

The LSTM model will be fine tuned so that its accuracy reaches above the benchmark model without any overfitting. Because the insurance premium the bank earns is sensitive to our predictions, we need our model to correctly predict the values rather than predicting all values. Thus we shall be looking forward to fine tune the model so that it has a higher precision compared to the benchmark model. In this application, a recall lower than precision is completely acceptable.

Evaluation metrics

1. MAE =

$$\sum_{i=1}^{n} \frac{\left| y_{pred} - y_{actual} \right|}{n}$$

A lower MAE is desired.

2. R squared value =

1 - (Explained Variation / Total Variation)

A higher R-squared value is desired.

3. AIC =

−2ln (L) + 2k

Here L is maximized value of likelihood function and k is the number of model parameters. A lower AIC is desired.

4. BIC =

−2ln (L) + k(ln(n))

Here L is maximized value of likelihood function, n is total number of examples and k is the number of model parameters. A lower BIC is desired.

5. Precision =

true positives / (true positives + false positives)

A higher precision is desired.

6. Recall =

true positives / (true positives + false negatives)

7. Binary accuracy of deep learning model output =

mean (total counts where predicted y = actual y)

A higher binary accuracy is desired.

8. Loss value of deep learning model output =

*mean (max(1-y_actual*y_predicted,0))*

A lower loss value is desired.

Project design

The project will be built on Python 3.7 and carried out on a Jupyter notebook. The data is free and available to download from the NSE website. The 9 months data file, as shown above will be split in two parts, i.e., train.csv (Nov 2017 - Oct 2018) and test.csv (Nov 2018 - Jan 2019). The train data will be used to train a linear regressor for prediction and corresponding p-value analysis. LSTM deep learning predictions will be done on Keras with TensorFlow backend.

References

[1] Deep Recurrent Factor Model: Interpretable Non-Linear and Time-Varying Multi-Factor Model, Nakagawa et. al., AAAI-19 Workshop on Network Interpretability for Deep Learning

[2] LSTM for time series forecasting

[3] RNN and LSTM

[4] Nifty50 historical data

[5] Nifty50 on Moneycontrol