Machine Learning Engineer Nanodegree

Prepared by: Chun Wang Chan
21 October 2018

Proposal

Investment and Trading Capstone Project
Build a Stock Price Indicator using Machine Learning techniques

Domain Background

A company completed an initial public offering then its shares will become public. As a result, the shares can be traded on a stock market. The stock market is a venue for buyers and sellers of shares to trade. Of course, it is where the stock prices are decided. The most common way to set stock prices is through the auction process involve buyers and sellers place bids and offer to buy or sell stock. A bid is the interested buying price and an offer is selling price. A trade is complete if both the bid and ask matched.
Source: [1]

The price of a stock can represent the value of the company. The share price is affected by changes in the stock supply and demand. So if there are more buyers interested to buy at a given moment than seller selling it then the price moves up. On the other hand, if more shareholders intended to sell than people to buy then the price will fall. But this is the direct and simple reason for stock price movement. There are much more complex or indirect hidden factors affecting shareholders make decision resulting price change. Thus, the problem of predicting future share price is a challenge.
Source: [2]

Problem Statement

Problem definition: To predict the adjusted close price of the day immediately after the end of the given training end date, of a given stock as accurate as possible using historical stock prices.

The problem of predicting stock price itself is not replicable. The problem is dynamic and different every time. This is because there is no definite pattern or clear indication of stock price movement. Using the same strategy or model to predict different stock price every time will not be effective. Thus the problem is not replicable and needs machines to learn the problem as it is too complex for humans.

It is measurable by metrics for calculating the differences between predicted values with actual values. It can be also observed by comparing the result plotting the predicted and actual values on graphs. The predicted stock prices can be used to provide guidance or indication of the potential momentum of the stock price direction. The predicted prices on average should be aimed to achieve the range within +/- 5 percent of actual prices.

Datasets and Inputs

The dataset from Yahoo Finance API contains historical stock prices. This can be used for training a machine learning model or a deep neural network to predict future stock prices. The target value to predict is the adjusted close price of given stock. The data will be obtained from the Yahoo Finance API. The required features or values needed are Open, High, Low, Close, Volume and Adjusted Close.

A backtest is a test simulation of trading strategy to check its performance if it were traded historically. In-sample backtesting is when we test the trading strategy or model using the same time period data as we trained it. This problem can be related to in machine learning where we must not use the training data for testing the model. The result can never be trusted. So we cannot use the same date range for training and testing the model.

Another issue was having something called a look-ahead bias. A look-ahead bias happens when using data in model training that would not have been known or exist during the period being trained. As this will affect the accuracy of the result. So we cannot train the model with a date range (e.g. future dates) and test the model with a historical date range. For example, the model was trained using 2014 data then we cannot test with 2008-2013 data. This will cause a look-ahead bias.

Therefore, there will be restrictions in passing the start and end date parameters to the training interface because of the issues mentioned above. For example, I can set the model to train with data from a date range from 1st Jan 2008 to 1st Jan 2013. For testing the model it can only allow to use data from date range after the end of training date.
Source: [15], [16], [17]

Characteristics of the dataset

Data Name	Data Type	Example	Description
Adjusted Close	Float	35.83	A stock's closing price of given trading date after amended any distributions and corporate actions occurred before next trading day opens.
Close	Float	35.83	The final and most up-to-date stock price traded on a given trading day. It does not reflect on after-hours and adjusted close price.
Date	Date	2014-04-29	The trading date or date for querying the stock price.
High	Float	35.89	The highest traded price of the stock for a given trading day.
Low	Float	34.12	The lowest traded price of the stock for a given trading day.
Open	Float	34.37	The first traded price of stock upon the opening of an exchange on a trading day.
Price	Float	36.84	The price of the stock at the time of making the query.
Symbol	String	APPL	The ticker symbol which abbreviates and represent a stock.
Volume	Float	28720000.0	The total quantity of shares traded for given stock.

Source: [3], [4], [5], [6]

Solution Statement

Potential solutions are using supervised learning method or deep learning approach. These can be used to train and build a model of stock behaviour. The supervised learning method of regression can be used to predict quantity or continuous values. This matches the problem of predicting stock prices. Similarly, a deep learning approach is capable of predicting continuous values.

A training interface and query interface are required in the project. The training interface will accept a date range and list of ticker symbols. The date range includes the start date and end date. This will set the dates for the model to train within the range. The ticker symbols represent the which stocks' historical prices to train with. The query interface also accepts a list of dates and ticker symbols. The outputs are the predicted stock prices for each stock on given date range.

Benchmark Model

The benchmark models to be used are a simple linear regression model and a persistence model. These benchmark models will be trained using the dataset and record the results for later comparison with the solution models. The persistence model will use a common baseline method which is the Zero Rule algorithm suggested in Source 18 article.
Source: [18]

Evaluation Model

The formula of RMSE:

$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{j=1}^{n}(y_j - \hat{y}_j)^2}$$

The evaluation metric to be used to quantify the performance is the Root Mean Square Error or RMSE. This metric is very popular for the regression problem and continuous values.
It can provide prediction accuracy of the model and it is said to be computationally simple.
The characteristic of RMSE is that it penalises higher difference more than metric like Mean Absolute Error or MAE. This factor is important and useful for the problem. Very inaccurate prediction should be avoided and heavily penalised. So the model can minimise chances to lose huge money by following very inaccurate prediction.

The evaluation model will be implementing a train-test cycle to measure the performance of the model. Using it to test prediction accuracy for query dates at different intervals after the training end date. For example intervals of training end date 7 days later, 14 days and 28 days.
Source: [7]

Project Design and Workflow

The project design will consist of building the training interface, query interface and then a simple user-friendly interface. The training interface will be called to initialize the data gathering from Yahoo! API and train the solution model. Once the model is ready for prediction, the query interface will be used to make prediction and output results. The simple user-friendly interface can be a prompt-based interface at first attempt when it is script based application. But if time is sufficient then a web app interface can be built for a better graphical user interface. It was told to be optional so it may not be included in the completion of the project.

Theoretical workflow:

1. Data gathering
2. Data exploration
3. Data preprocessing (feature observation and selection)
4. Implementation and model building
5. Test, evaluate and visualization

Data analysis and preprocessing:

The dataset will be observed and analysed by plotting into graphs and tables. Feature selection can be used to filter unnecessary features in the dataset. This allows to speed up training speed by selecting only important features. The dataset might require data cleaning involve filling missing data, remove outliers and fix the inconsistency. Data transformation and reduction. Also, one-hot encoding for some data fields. But after all, the data preprocessing needed may vary in actual implementation.
Source: [8], [9]

The algorithms to be considered for my implementation includes:

Supervised learning ensemble method, XGBoost
Deep learning architectures, Multi-layer Perceptron (MLP) or Recurrent Neural Net (RNN).

XGBoost is an implementation of gradient boosting machine. It is being quite popular in many machine learning competition. It is faster and performs better than naive gradient boosting in general. It is able to solve regression and prediction problems. For using a supervised learning method, the XGBoost is a good option to explore.
Source: [10], [11]

Deep learning architectures are capable of regression and prediction task. Two architectures to be considered are MLP and LSTM RNN. RNN has been used for prediction tasks such as stock market prediction. It uses sequential of inputs and takes consideration of historical inputs. But other neural network architectures only treat input independently. This characteristic may be beneficial for detecting correlation and related patterns in the price movement. A particular extension of RNN to consider using is the Long-Short Term Memory or LSTM. RNN has internal memory for remembering the inputs it received. The LSTM extension is to extend the internal memory capacity. As a result, it helps to learn important experiences or patterns over a longer period of time.
Source: [12], [13]

References:

1. How stocks trade - https://www.investopedia.com/university/stocks/stocks3.asp
2. Valuing stocks - https://www.investopedia.com/university/stocks/stocks7.asp
3. Adjusted Closing Price - https://www.investopedia.com/terms/a/adjusted_closing_price.asp#ixzz5UI5qgT7B
4. Closing Price - https://www.investopedia.com/terms/c/closingprice.asp#ixzz5UI6D6c90
5. Opening Price - https://www.investopedia.com/terms/o/openingprice.asp#ixzz5UI7vO07S
6. Volume of Trade - https://www.investopedia.com/terms/v/volumeoftrade.asp#ixzz5UI8vvnlW
7. ML metrics - https://medium.com/usf-msds/choosing-the-right-metric-for-machine-learning-models-part-1-a99d7d7414e4
8. Data processing - https://medium.com/datadriveninvestor/machine-learning-ml-data-preprocessing-5b346766fc48
9. Feature selction - https://towardsdatascience.com/why-how-and-when-to-apply-feature-selection-e9c69adfabf2
10. Xgboost - https://machinelearningmastery.com/gentle-introduction-xgboost-applied-machine-learning/
11. Xgboost - https://www.kdnuggets.com/2017/10/xgboost-top-machine-learning-method-kaggle-explained.html
12. LSTM - https://towardsdatascience.com/recurrent-neural-networks-and-lstm-4b601dd822a5
13. RNN - https://medium.com/mindorks/understanding-the-recurrent-neural-network-44d593f112a2
14. https://docs.google.com/document/d/1ycGeb1QYKATG6jvz74SAMqxrlek9Ed4RYrzWNhWS-0Q/pub
15. Bias - https://www.investopedia.com/terms/l/lookaheadbias.asp#ixzz5Ugc6xNb0
16. https://www.linkedin.com/pulse/avoiding-forward-bias-time-series-machine-learning-rohit-walimbe-1/
17. https://blog.quantopian.com/9-mistakes-quants-make-that-cause-backtests-to-lie-by-tucker-balch-ph-d/
18. https://machinelearningmastery.com/persistence-time-series-forecasting-with-python/