Investment and Trading Capstone Project

Build a Stock Price Indicator

Investment firms, hedge funds and even individuals have been using financial models to better understand market behaviour and make profitable investments and trades. Stock market is the act of trying to determine the future value of a company stock or other financial instrument traded on an exchange. The successful prediction of a stock's future price could yield significant profit [1]. The efficient-market hypothesis suggests that stock prices reflect all currently available information and any price changes that are not based on newly revealed information thus are inherently unpredictable. A wealth of information is available in the form of historical stock prices and company performance data, suitable for machine learning algorithms to process.

Problem Statement

For this project, we will use deep recurrent Network to build a stock price predictor that takes daily trading data over a date range from 01/01/2000 – 28/07/2017 as input, and outputs projected estimates for given query dates .The inputs contain multiple metrics, such as opening price (Open), highest price the stock traded at (High), how many stocks were traded (Volume) and closing price adjusted for stock splits and dividends (Adjusted Close); my system only needs to predict the Adjusted Close price.

Datasets and Inputs

For this project, stock price indicator, the dataset to be used will be that of publicly traded companies from the S&P 500 stock market, Microsoft Corporation (MSFT), Amazon.com, Inc. (AMZN) and The Walt Disney Company (DIS) obtained for free from Yahoo! Finance. The following links are the sources where I got the historical data. I downloaded a dump of .csv files and used them.

- S&P 500: https://finance.yahoo.com/quote/SPY/history?p=SPY
- Microsoft Corporation: https://finance.yahoo.com/quote/MSFT/history?p=MSFT
- AMZN : https://finance.yahoo.com/quote/AMZN/history?p=AMZN
- DIS : https://finance.yahoo.com/quote/DIS/history?p=DIS

S&P 500 stock market, Microsoft Corporation, Amazon.com, Inc. (AMZN) and The Walt Disney Company (DIS) were chosen because of their sizes and historic data available for training. In order to predict future stock prices, models would need current and historic data to determine data behaviour over time or similar characteristics.

Daily stock contains the following values Date, Open, High, Low, Close, Volume and Adjusted Close. Open is the price on the open day of the stock market. High is the highest price the stock reached that day and then it's Low. Close is the price of the last trade when the market closed that day. Volume is the number of shares/contracts traded in a security during a given trading day and the Date is the calendar date of the reported values, the adjusted close is usually the after-hours price and the true open price adjusted from the close price posted.

Tagne Nguifo Eric Hermann

Historic stock data will be used for data pre-processing, statistical analysis, feature selection and engineering, model training and finally the trained model will be used to predict future stock prices.

Adjusted Close is what we will be trying to predict and prediction will only be carried out on S&P 500.the others will be about statistical analysis.

Solution Statement

For the core stock predictor, we would implement:

- A training interface that accepts a data range (start_date, end_date) and a list of ticker symbols (e.g. GOOG, AAPL), and builds a model of stock behaviour. My code should read the desired historical prices from the data source of my choice.
- A query interface that accepts a list of dates and a list of ticker symbols, and outputs the predicted stock prices for each of those stocks on the given dates. Note that the query dates passed in must be after the training date range, and ticker symbols must be a subset of the ones trained on.

A proposed solution would be to use Deep Recurrent Neural Network (Long Short term Memory) model trained on historic data to predict, within a margin of error, the future closing price of a stock. This knowledge would put the owner of said information in an advantageous position, allowing him or her to invest in a company's stock increase or decrease in value.

Benchmark Model

The benchmark model to be used in this project is linear regression. It is an approach for modelling the relationship between a scalar dependent variable *y* and one or more explanatory variables (or independent variables) denoted *X*. The case of one explanatory variable is called *simple linear regression*. For more than one explanatory variable, the process is called *multiple linear regression*[2]

Evaluation Metrics

The benchmark model and solution model will be evaluated, when appropriate, with Mean Squared Error and R^2 score metrics.

Mean squared error [3]

The mean_squared_error function computes mean square error, a risk metric corresponding to the expected value of the squared (quadratic) error loss or loss.

If $\hat{y}_i$ is the predicted value of the i-th sample, and y_i is the corresponding true value, then the mean squared error (MSE) estimated over n_{samples} is defined as

$$\text{MSE}(y, \hat{y}) = \frac{1}{n_{\text{samples}}} \sum_{i=0}^{n_{\text{samples}}-1} (y_i - \hat{y}_i)^2.$$

R^2 score, the coefficient of determination[4]

The r2_score function computes R^2, the coefficient of determination. It provides a measure of how well future samples are likely to be predicted by the model. Best possible score is 1.0 and it can be negative (because the model can be arbitrarily worse). A constant model that always predicts the expected value of y, disregarding the input features, would get a R^2 score of 0.0.

If $\hat{y}_i$ is the predicted value of the i-th sample and y_i is the corresponding true value, then the score R^2 estimated over n_{samples} is defined as

$$R^2(y, \hat{y}) = 1 - \frac{\sum_{i=0}^{n_{\text{samples}}-1} (y_i - \hat{y}_i)^2}{\sum_{i=0}^{n_{\text{samples}}-1} (y_i - \bar{y})^2}$$

Where

$$\bar{y} = \frac{1}{n_{\text{samples}}} \sum_{i=0}^{n_{\text{samples}}-1} y_i$$

Project Design

The approach for this problem will, as a whole, take on the view of a product and have not just a predictive model but a more user friendly interface for someone to easily interact with and request predictions. Additionally, the system will provide its own historical predictions and its final accuracy. Below are the associated processes for both the Modelling and Product workflows:

Modelling Workflow – Deep Recurrent Neural Network

- Data gathering + collection
- Data pre-processing
- visualizations
- Feature reduction / selection
- Dimensionality reduction
- Train/test data split
- Final model
- Test dataset
- Predictions

- Model evaluation

Test and measure performance

A basic run of the core system would involve one call to the training interface, and one or more calls to the query interface. Implement a train-test cycle to measure the performance of my model. Use it to test prediction accuracy for query dates at different intervals after the training end date, e.g. the day immediately after training end date, 7 days later, 14 days, 28 days, etc.

(Note: Pick the training period accordingly so that you have ground truth data for that many days in the future.)

Build user interface

Once you're iterated on your stock predictor a few times, and it is giving results you are happy with (say, predicted stock value 7 days out is within +/- 5% of actual value, on average), implement a more user-friendly interface that lets you specify stock(s) you are interested in and provides predictions at some pre-defined intervals.

References

1 https://en.wikipedia.org/wiki/Stock_market_prediction

2 https://en.wikipedia.org/wiki/Linear_regression

3 http://scikit-learn.org/stable/modules/model_evaluation.html#mean-squared-error

4 http://scikit-learn.org/stable/modules/model_evaluation.html#r2-score-the-coefficient-of-determination