Machine Learning Engineer Nanodegree
Capstone Project

Mauricio Domingues Aroldi

June 9, 2017

Implementation of a Stock Predictor

Domain Background

Machine Learning has been used for many years in stock trading, specially with high-frequency trading and hedge funds. The huge amount of data produced requires fast analysis and process to hope having some advantage over the market. Many algorithms and models were made but as this is an always-changing field, no solution remains forever "optimal", meaning that it requires continuous improvements. [1, 2, 3, 4, 5, 6, 7, 8]

This project aims to predict stock prices using Machine Learning algorithms, indicating which of them one should buy or sell shares. For this I will explore different supervised learning algorithms and features to find the best results. Predicting stock prices is a very hard task, as there are many factors that influence its price and which fall outside the scope of this study. Nonetheless, it is still a very interesting field of study, as it involves math, statistics and behavioral analysis.

Problem Statement

It is a regression problem as what I want to predict is the Adjusted Close price of a stock, or multiple stocks, and prices are continuous. The input data will be extracted and processed, and joined with the time series data, becoming the features necessary to create one or many models capable to return the desired result. This result is a number that, compared to the previous Adjusted Close price (from the day or days before) will give us the tendency of the movement, pointing if it will get more expensive or cheap, therefore indicating to buy or sell the shares of the specific stock.

Datasets and Inputs

The data used in this problem will be extracted from Yahoo! Finance, and the obtained features are:

Column	Type	Meaning
Date	YYYY-MM-DD	date
Open	float	Price of the stock when the market opened on that day, in US dollars.
High	float	Maximum price of the stock during the day, in US dollars.
Low	float	Minimum price of the stock during the day, in US dollars.
Close	float	Price of the stock when the market closed on that day, in US dollars.
Adj Close	float	Closing price of a stock on any given day of trading that has been amended to include any distributions and corporate actions that occurred at any time prior to the next day's open.
Volume	integer	The number of shares of that stock traded on that day.

Above there is an example of the data provided in a CSV file downloaded from Yahoo! Finance, a few trading days of Tesla Motors ("TSLA"):

Date	Open	High	Low	Close	Adj Close	Volume
2017-05-10	321.559998	325.500000	318.119995	325.220001	325.220001	5741600
2017-05-11	323.399994	326.000000	319.600006	323.100006	323.100006	4753800
2017-05-12	325.480011	327.000000	321.529999	324.809998	324.809998	4121600

The values of Adjusted Close, that will be our target, while the rest will be used as parameters or to form new parameters, if necessary. The size of the file/table will vary as it depends on the date range chosen to train the model. To train the model, I will be able to input the ticker symbols and a date range - compatible with the availability of the historical data. I propose the use of a small set of ticker symbols from three different sectors (e.g. Energy, Technology and Cyclical Consumer Goods & Services) to create a portfolio and compensate temporary instability of a whole sector.

Solution Statement

To solve this problem, I propose the implementation of three different models - parametric, instance based and ensemble - so that I can evaluate and make use of the best to create the user interface. Our first goal is to implement a day-trader model, that predicts the Adjusted Close prices of the following day, but I will also evaluate the predictions for different time ranges as 7, 14 and 28 days.

Benchmark Model

The benchmark will be made comparing the results of each model against the S&P 500 index, also available through Yahoo! Finance API.

Evaluation Metrics

The performance of each model will be made measuring the difference between the actual stock price and the predicted Adjusted Close price, this is, the root mean squared percentage error. With these percentages it is possible to better evaluate the accuracy of each model. The performance can also be evaluated with backtesting, where we compare the results of the models with the historical data of the same period.

Project Design

The project will be developed in Python 2, using Jupyter notebook as IDE. The main Python libraries in use are: Numpy, Matplotlib, Scikit-learn, Pandas. Other libraries can also be used as needed.

The first analysis to be made with the data is to verify its completeness, as the chosen date range may select a period where a stock has not been traded, or may have some gaps in its middle. I intend to pick nine ticker symbols, three companies in each of the three sectors mentioned before, to form a portfolio.

I will use three different algorithms to test which has the best performance compared to the S&P 500 index. [9] These algorithms are: Linear Regression, KNN and Ensemble.[10] I will start with the basic settings, and according with the results, tune each one to get the best responses. This may include tools as Grid Search and Cross Validation, as an example.[11] If not enough, there is the possibility to use the features to generate another ones (or even extract new ones, as EPS - earnings per share) as momentum, SMA (Simple Moving Average) or Bollinger bands®.[12] Another solution to try against possible low accuracy is to use all of the models together, taking the mean of their results to have a more balanced one, as suggested by professor Tucker Balch in his Machine Learning for Trading classes.[13]

To generate each model, the features (X) will be paired with n-days-ahead target (y) - Adjusted Close prices. N-days-ahead meaning that it depends the number of days we want to predict and that it must be related with a future date to create an honest model.

Having the model ready, it is possible to input the date range and the prediction interval (n-days-ahead) and generate an order. Actually it will be a suggestion for buy and sell for each of the stocks, as it may be a tough task (to not say doubtful) to point out how many shares should be traded.

References

[1] A. C. Andersen and S. Mikelsen, "A novel algorithmic trading framework applying evolution and machine learning for portfolio optimization," Master's thesis, Norwegian University of Science and Technology, 2012.

[2] L. Fievet and D. Sornette, "Decision trees unearth return sign correlation in the s&p 500," 10 2016.

[3] S. Ganguli and J. Dunnmon, "Machine learning for better models for predicting bond prices," 05 2017.

[4] L. Troiano, E. Mejuto, and P. Kriplani, "On feature reduction using deep learning for trend prediction in finance," 04 2017.

[5] D. Hendricks and S. J. Roberts, "Optimal client recommendation for market makers in illiquid financial products," 04 2017.

[6] Y. Hilpisch, *Python for Finance*. O'Reilly Media, 1st ed., December 2014.

[7] J. Hallgren and T. Koski, "Testing for causality in continuous time bayesian network models of high-frequency data," 01 2016.

[8] N. Milosevic, "Equity forecast: Predicting long term stock price movement using machine learning," 03 2016.

[9] "https://finance.yahoo.com/quote/^GSPC?p=^GSPC," June 2017.

[10] "http://scikit-learn.org/stable/index.html," June 2017.

[11] "http://scikit-learn.org/stable/model_selection.html#modelselection," June 2017.

[12] "http://www.investopedia.com/," June 2017.

[13] "https://udacity.com/course/machine-learning-for-trading–ud501/."