Capstone proposal

Domain Background

Investment firms, hedge funds and even individuals have been using financial models to better understand market behavior and make profitable investments and trades. A wealth of information is available in the form of historical stock prices and company performance data, suitable for machine learning algorithms to process. There are many factors to use to predict the movement of stock prices. Yet, none of them are 100% perfect. Furthermore, the one thing that stands in the way of a good investment strategy is human emotions. There is a tendency to exit a profitable position too early or hold on to a losing position because we have invested in them.

There have been numerous attempts and research done and implemented using machine learning for trading. [1] is a fine example from the stanford ML course as a guidance. We can also follow some of the methods used in Udacity's Machine learning for trading by Tucker Balch Most of the trading algorithms are quite complex and requires numerous data sources or huge computing power. We want to be able to translate simple statistical measures and commonly understood risk factors into a decent stock predictor for the common man to implement trades.

Problem Statement

We want to be able to predict from data a near future value of a stock. This can help a trader make decision. With the current trend in the particular stock, what can be the expected value of the stock in a day's time? We would experiment with a few factors, such as a stock's adjusted close, simple moving average, Boillinger band value, P/E ratio. We can focus on a few measurements such as Cumulative returns, daily returns, standard deviation, sharpe ratio to evaluate our algorithm and its performance.

Datasets and Inputs

Our data sets will be from Yahoo finance and be read by our algorithm as csv files to build the model for a particular stock. Some of the measures that are not available in Yahoo, we may easily compute them using the data.

We will train our model using Apple's stock. The reason being this stock has a high daily volume of trades. Our aim is to build a model that can predict Apple's stock price the next day using the daily stock price movements over the past two weeks. However, we can experiment our model on other liquid stocks such as Microsoft's. We will be looking at data from the past five years.

Solution statement

With the input of two weeks' data, we would like to output the price of the stock in one week's time. Stock prices are clearly observable in the market. We will also be using recurrent neural networks or more specifically the Long Short Term Memory(LSTM) which is well suited for time series data. We will also be training on a polynomial regressor(deg = 2).

Benchmark model

We can compare our model to a naïve base line model that outputs a stock price on whatever the input is today or i.e. no change in stock price. Suppose everyone is buying and/or selling at random, then the expected stock price tomorrow should be the same as today's.

Evaluation metrics

This is a time-series problem and so care must be taken with our algorithms. For example, using k-fold cross validation without careful manipulation would result in peeking into the future which we must avoid. We can use the root mean square error as the evaluation metric for the overall performance of the model. The root mean square is defined as the square root of the mean of the square of all error.

$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$$

We can also compare the percentage difference between the predicted price and the actual price, for example +/- 5%.

Project Design

Using the data from Yahoo finance, our first step would be to look for any missing data figures. We can use forward fill or backward fill method for empty trade dates. Thereafter, we shall compute the desired statistical and risk measures for our model. This includes, the daily returns, a simple moving average, P/E ratio etc. from our data. These shall form our predictive factors. We will clearly describe our output which is change in price. From this we can easily out put the expected stock price.

Before we can go on, we can use a five day moving average instead of the actual stock price movements. This would help to filter out noise.

We will split our data into training and testing set. On our training set, we will take two week's data and predict the stock price one day from the end date. Our forecast is one day and our lookback is 2 weeks. Comparing out prediction with the actual stock price will give us our error and from there we will look to minimze this error using the machine learning algorithms such as RNN and our quadratic regressor. Once we have our output we will combine the projections with equal weightage.

References

[1]http://cs229.stanford.edu/proj2013/DaiZhang-MachineLearningInStockPriceTrendForecasting.pdf

[2] Udacity's Machine learning for trading by Tucker Balch

Review

- The problem input and output are clearly defined, also it can be indirectly inferred that you are dealing with regression as you mention that you want to predict the price.
- You can do your own "roll forward" cross-validation method (it is not that hard) or you can use the TimeSeriesSplit from sklearn, which does exactly what you need for your evaluation :)
- Root mean squared error is a reasonable metric for this problem
- I don't see how you could use boosting for this problem, I work a lot with time series and data stream mining (kind of a time series renegaded simbling) and often boosting is not a good choice. My advice is that you stick to your plan of using LSTM in here.