Capstone proposal

December 3, 2018

1 Machine Learning Capstone Project

1.1 Capstone Proposal

Sverre Lillelien
November 28th, 2018

1.2 Proposal

1.2.1 Domain Background

Automation of stock trading has been among the most popular uses of advances in hardware and machine learning, and has been a topic of interest to academics (http://cs229.stanford.edu/proj2012/BernalFokPidaparthi-FinancialMarketTimeSeriesPredictionwithRecurrentNeural.pdf) and practitioners alike. Using algorithms to attempt to identify over- or undervalued companies traded on the stock market is a strategy employed by many investment funds and banks. The stock market is a natural candidate for such analysis because of the vast amount of data, even real-time time, generated by stock markets and companies. For a human it is impossible to filter through all of this information in a meaningful way, so perhaps machines can help us do this dirty work for us, and provide some new insight.

Personally I have doubts regarding the value of such an approach, as there are so many variables impacting stock prices, and so much randomness, that predicting stock prices with any degree of certainty and consistency seems impossible. However, this approach is employed by such a vast number of different funds and analysts that it would be a useful exercise in its own right to attempt to apply some machine learning techniques to this field to better understand its strengths and limitations. With luck, even finding some small insight which could help us filter through some of the noise in the stock market might give us an edge and narrow our focus when selecting stocks, would be valuable.

1.2.2 Problem Statement

The problem I wish to analyze in this project is whether we can predict the future price of a stock. Specifically, I will label stocks as either buy, sell or hold based on changes in its price within a period of time, and treat this as a classification problem.

The scope of this project will be limited due to time constraints, so I will initially attempt to predict future prices based on historical prices. This is unlikely to provide great insight in itself, but might be a useful starting point to elaborate further on, and expand the analysis.

1.2.3 Datasets and Inputs

My data set consists of five years of stock data for SP 500 companies. The data includes opening prices, as well as high, low and closing prices. The data set can be found here: https://www.kaggle.com/camnugent/sandp500/home

The data set contains 619 000 rows of data, one for each date for each company starting from February 2003 and going till Februar 2018. I will focus on closing prices, and pivot the dataframe to sort using date as the index, and give each company its own column.

I will be using cross validation to split the data into training and test sets.

I also have data sets containing financial figures for the same companies, which may or may not be used, depending on how the project develops and the time I have available.

1.2.4 Solution Statement

"Solving" the problem is measured by obtaining future stock prices in USD, and classifying the stock as either a buy or sell, or to simply keep holding the stock. As such it is quantifiable and easy to understand.

To help me solve this problem I will attempt to use supervised learnining, more specifically SVM, k-nearest neighbors and random forest classfier.

1.2.5 Benchmark Model

I am a bit unsure of the appropriate benchmark model in this case, but the stock prices are given, and can be checked against the predictions. As a benchmark I'll use an out-of-the-box SVM, which we can hopefully outperform.

1.2.6 Evaluation Metrics

My plan is to test different models and try to experiment with model parameters to find good fits. As I have landed on a classification problem I will most likely use AUC as my metric. However, these metrics may be subject to change as the project takes shape.

1.2.7 Project Design

Coming up with a capstone proposal I have spent a lot of time thinking about different problems, starting to work on some, quickly discarding others. Simply coming up with a problem for the project that is relevant, viable and realistic to complete has been challenging. I have found it especially hard not to focus on all of the different reasons why potential projects might be too complicated, or all the difficulties that can arise trying to apply ML-techniques to problems.

In the process of writing this proposal, I have gone through several data sets trying to find the most suitable ones. At first I attempted to create my own data set via Quandl's API (https://www.quandl.com), and I tried to improve upon a data set downloaded using Yahoo Finance some time ago, only to find that they no longer offer the same services as before.

My task now is preparing the data set for further analysis, and preprocessing it for machine learning. Since pricing data itself will vary from company to company, and depend on the number of shares issued, one way to normalize the data will be to look at percentage change in the prices. I will use these percentage changes as my features determining the labels: buy, sell or hold. The hypothesis is that some positive percentage change in price over a given period of time will indicate a buy, and a negative change indicates a sell. Choosing the appropriate time period and acceptable percentage change is important, as we do not want to "buy high and sell low", but if we have a fairly short time period and a modest price change, it seems plausible this could give us an edge.

I will visualize some stocks prices to look at different companies and trends in general, and see if we can learn something from looking at specific sectors or companies over time.

I will then attempt to apply different machine learning models. Neural networks are widely used to predict stock prices. If I succeed in labeling the stocks as either buy or sell using some metrics like percent change in price over a period, I want to try and classify using random forest. Ideally, this labeling would incorporate financial data from the accounts as well, but this might be beyond the scope of this project. Using SVM and k-nearest neighbors I've already mentioned above, and given your feedback I also want to try XGBoost and possibly LightGBM, depending on complexity.
