Machine Learning Engineer Nanodegree

Capstone Proposal

Dillon Pietsch
July 5th, 2017

Proposal

Domain Background

The first stock exchange in the world was the London Stock Exchange, which opened up in 1698. Not soon after, the NYSE opened its doors in 1792, creating one of the cornerstones of the modern day capitalist society. Many other exchanges opened up since, but one of the most important advances occurred in 1971, when the NASDAQ was created. It was an important turning point since it was the world's first electric stock exchange, and the first to allow online trading. Since then, technology has become more integrated into the stock exchanges. Many human traders have been left at the wayside as computers can more effectively execute the trading. Besides trading on the floor, computers have become more prominent (with the rise of Artificial Intelligence) in hedge funds and managing portfolios for millions of people. Take Aidyia, a hedge fund run entirely by AI, for example. Their AI executes trades on its own, without any human intervention. Other hedge funds are using AI in some capacity in order to give better returns.

As the benefits of AI are clear, continued research and use of new technologies is increasing. Big banks likes JP Morgan Chase are developing AI algorithms to get more money out of market. All of this investment shows hope and promise for the future of Artificial Intelligence's applications with respect to stock trading. Looking through the list of possible capstone projects and seeing a stock predictor I immediately jumped on the idea. My dad was a broker and because of his profession I've had a deep interest in the field along with the potential applications of AI for stocks. While I realize that I'm not going to have a fantastic algorithm and predictor, I'll learn more about AI, stocks, and the market, which is a great takeaway for this project.

Problem Statement

Since the stock market has been created, people have wanted to know what was going to happen next. The uncertainty of the future is clearly shown by how difficult it is to predict stock prices. The problem to be solved is, taking into account past stock data, use machine learning and domain knowledge of stock price factors, predict the price of a stock into a certain amount of time into the future.

Datasets and Inputs

The dataset used for this project is of the publicly traded companies from the NASDAQ and NYSE, accessed through the yahoo-finance module. The module is imported as 'yahoo-finance'.

The data from the NYSE and NASDAQ are great choices to get data from because it's where the vast collection of publicly traded companies are traded and the exchanges are going to have accurate historical data on the stocks. The stock data can go far back into the past in order to train the ML algorithm well enough to make more accurate predictions compared to guessing. The current and past stock prices will provide the framework for feature engineering to make sure that the data being used is relevant to predicting stock prices.

Dataset Characteristics

```
Open: The opening price of the stock which is the first price at the opening of the
exchange.
Close: The closing price of the stock when the exchange closes.
Adj_close: CLosing price adjusted for stock splits or other actions that change the
value of the stock.
Date: Gives the calendar date of the stock price.
Low: Gives the low for the day of trading.
High: Highest price for the given day of trading.
Symbol: Unique stock identifier.
Volume: Number of shares traded on the given day.
```

Solution Statement

Using the vast amounts of historical price data, a solution to the problem would be to use supervised learning algorithms to predict the price of a stock to a certain degree of accuracy. Supervised learning is ideal because all of the data is labeled and the type of answer is known -- the price of the stock sometimes into the future.

Benchmark Model

The benchmark model will be a random forest regression algorithm. This algorithm won't be optimised to the data, which will show what a baseline algorithm will predict compared to optimized algorithms.

Evaluation Metrics

R Squared will be used as the evaluation metric to determine the accuracy of the predictions. For R^2, the best score will be closer to 1.0, while worse scores will be closer to 0.0. It shows how well the predictions are to the actual values. The value is found by subtracting the quotient of the sum of squared errors by the total sum of squares.

Project Design

Gnerally speaking, this project will have a very simple UI interface to allow users to input stock information which will then output the predicted adjusted close price of the stock on the screen. It will be a very simple interface that allows for easily predicting a stock price x amount of days into the future.

Data Analysis

I expect that the data from the yahoo-finance module will likely be intact and there won't need to be much manipulation with the values. However some feature engineering and data cleaning may need to take place with regards to normalizing features like 'Volume' and filling in missing values should there be any. For the most part I think the data will be good straight out of the gate.

PCA

Some features are needed in order to organize the data, but don't have any instrinsic meaning (like the 'Date'). However, some features may seem look to be helpful but actually have no relevance. In this case, it's best to use PCA (Principal Component Analysis) to find the importance each feature has towards the stock price. PCA, shown through visualization techniques in Python, should give an idea to the more important features in the dataset.

Supervised Learning Algorithms

This project calls for supervised learning algorithms that return use regression, since it's trying to predict a value rather than classify.

Algorithms that will be tested:

```
K Nearest Neighbors (KNN)
Support Vector Machines (SVM)
Decision Tree Regressor
Gradient Boosting Regressor
```

Train/Test Split and Cross Validation

For each algorithm, the data will be tested for accuracy in multiple ways. The will be a 70/30 train/test split with the data, along with k-fold cross-validation. This will help ensure a lot of variety with the data used and make sure there are no biases that I or the computer can use to take advantage and 'cheat' to get a better prediction.

UI

The user interface will be simple. IT will have a text field for the stock ticker and the ability to choose a date range for the historic price data, along with how far out into the future the user wants reported back. Once the algorithm predicts the stock price, it will appear on the GUI.

References

Domain Background


```
http://www.nanalyze.com/2017/02/artificial-intelligence-stock-trading/
https://www.wired.com/2016/01/the-rise-of-the-artificially-intelligent-hedge-fund/
http://www.wisestockbuyer.com/2012/06/when-did-the-stock-market-begin/
http://www.bbc.com/news/business-34264380
```

Datasets and Inputs

```
https://pypi.python.org/pypi/yahoo-finance
```

Benchmark Model

```
http://scikitlearn.org/stable/modules/generated/sklearn.ensemble.RandomForestRegressor.html
```

Evaluation Metrics

```
http://scikit-learn.org/stable/modules/generated/sklearn.metrics.r2_score.html
http://www.hedgefund-index.com/d_rsquared.asp
```