

Machine Learning Engineer Nanodegree

Capstone Proposal

Oscar Martinez Rico

May 13th, 2017

Investment and Trading Capstone Project Proposal

Domain Background

As I joined my current employer, I was awarded some stocks. I find myself always trying to decide if it's a good idea to keep my stocks or sell them based on the current stock price. Having a software system that helps me determine if it is the right time to sell or keep a particular stock would be very helpful. There is available historical stock information that could be used combined with machine learning algorithms to make profitable investment decisions.

Problem Statement

Building a stock price predictor that takes daily trading data over a certain date range as input, and outputs projected estimates for given query dates. By looking at the historical data of a given stock as an input, the stock predictor application will train the model to predict the Adjusted Close value for the given stock in the future. Having such a software system will help people like me to make more educated decisions managing their stock portfolio.

Datasets and Inputs

The stock predictor application will extract the historic stock information from Yahoo! Finance using the python library "yahoo-finance 1.4.0".

- <https://pypi.python.org/pypi/yahoo-finance>

The inputs will 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). The stock predictor application will predict the Adjusted Close price.

Solution Statement

The stock predictor application should be able to predict the stock price of a given stock by doing the following:

- 1. The application should receive the stock name and date range as input.*
- 2. The application will fetch historical data of the given stock using the Yahoo Finance API.*
- 3. Split historical data collected to have two datasets, training data and test data.*
- 4. Training the model using the historical data.*
- 5. Validate the prediction results using the test dataset.*
- 6. Tweak the model, if necessary, to get better scores results and repeat steps 4 to 5.*
- 7. Predict the stock price using the training model and date range given as input.*
- 8. Show the results to users.*

Benchmark Model

If any of the algorithms, used to predict the stock price (Linear Regression, Support Vector Regression, Nearest Neighbors Regression) obtains a score of more than 50% will predicting a stock price, we can assume that this model can be useful to predict stocks prices using unseen data accurately.

Evaluation Metrics

The model should predict 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.

Project Design

- 1. To obtaining the historical data I thinking to try the following:*
 - ☐ *Create the datasets in real-time calling:*

❑ `stock.get_historical('2014-04-25', '2014-04-29')`

- ❑ Obtain manually the datasets and attach the corresponding files in the application.
- ❑ Split the historical dataset into test and train data.

2. I'm planning to use sklearn and explore the following algorithms:

- ❑ Linear Regression
- ❑ Support Vector Regression.
- ❑ Nearest Neighbors Regression

3. Using each of the algorithms I will train and test each model to identify which one provides a better score; this model will have more changes to accurately predict stocks prices using unseen data.

4. Implement a more user-friendly interface that lets users specify stock they are interested in and provides predictions at some pre-defined intervals.