**Machine Learning Engineer Nanodegree**

**Capstone Proposal**

Kun Zheng

May 13th, 2018

1. 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.

I picked up this suggested problem area from the capstone description.

In this section, provide brief details on the background information of the domain from which the project is proposed. Historical information relevant to the project should be included. It should be clear how or why a problem in the domain can or should be solved. Related academic research should be appropriately cited in this section, including why that research is relevant. Additionally, a discussion of your personal motivation for investigating a particular problem in the domain is encouraged but not required.

1. Problem Statement

In this section, clearly describe the problem that is to be solved. The problem described should be well defined and should have at least one relevant potential solution. Additionally, describe the problem thoroughly such that it is clear that the problem is quantifiable (the problem can be expressed in mathematical or logical terms) , measurable (the problem can be measured by some metric and clearly observed), and replicable (the problem can be reproduced and occurs more than once).

1. Datasets and Inputs

I found the S&P 500 companies historical prices on Kaggle but it appears the dataset is a little bit outdated and spans from 2010 to the end 2016.

Alpha Vantage providers real-time and historical data for stocks, ETFs, mutual funds, forex, digital/crypto-currency, and more for free.

https://www.alphavantage.co

https://github.com/RomelTorres/alpha\_vantage

A python wrapper for Alpha Vantage API for financial data

In this section, the dataset(s) and/or input(s) being considered for the project should be thoroughly described, such as how they relate to the problem and why they should be used. Information such as how the dataset or input is (was) obtained, and the characteristics of the dataset or input, should be included with relevant references and citations as necessary It should be clear how the dataset(s) or input(s) will be used in the project and whether their use is appropriate given the context of the problem.

1. Solution Statement

In this section, clearly describe a solution to the problem. The solution should be applicable to the project domain and appropriate for the dataset(s) or input(s) given. Additionally, describe the solution thoroughly such that it is clear that the solution is quantifiable (the solution can be expressed in mathematical or logical terms) , measurable (the solution can be measured by some metric and clearly observed), and replicable (the solution can be reproduced and occurs more than once).

1. Benchmark Model

In this section, provide the details for a benchmark model or result that relates to the domain, problem statement, and intended solution. Ideally, the benchmark model or result contextualizes existing methods or known information in the domain and problem given, which could then be objectively compared to the solution. Describe how the benchmark model or result is measurable (can be measured by some metric and clearly observed) with thorough detail.

1. Evaluation Metrics

In this section, propose at least one evaluation metric that can be used to quantify the performance of both the benchmark model and the solution model. The evaluation metric(s) you propose should be appropriate given the context of the data, the problem statement, and the intended solution. Describe how the evaluation metric(s) are derived and provide an example of their mathematical representations (if applicable). Complex evaluation metrics should be clearly defined and quantifiable (can be expressed in mathematical or logical terms).

1. Project Design

In this final section, summarize a theoretical workflow for approaching a solution given the problem. Provide thorough discussion for what strategies you may consider employing, what analysis of the data might be required before being used, or which algorithms will be considered for your implementation. The workflow and discussion that you provide should align with the qualities of the previous sections. Additionally, you are encouraged to include small visualizations, pseudocode, or diagrams to aid in describing the project design, but it is not required. The discussion should clearly outline your intended workflow of the capstone project.

Investment and Trading Capstone Project

Build a Stock Price Indicator

Description

For this project, your task is to build a stock price predictor that takes daily trading data over a certain date range as input, and outputs projected estimates for given query dates. Note that 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); your system only needs to predict the Adjusted Close price.

You are free to choose what form your project takes (a simple script, a web app/service, Android/iOS app, etc.), and any additions/modifications you want to make to the project (e.g. suggesting what trades to make). Make sure you document your intended features in your report.

Setup

Recommended setup for a simple script or web app/service:

* Python
* NumPy, SciPy, Pandas
* (Optional) Python API for data access (see below)

Android/iOS app:

* Android/iOS SDK
* Native library/API for data access (see below)

Data

There are several open sources for historical stock price data which you are free to use:

* [Yahoo! Finance](https://www.google.com/url?q=http://finance.yahoo.com/&sa=D&ust=1526285409807000): You can directly query for a stock through the web API, or download a dump of .csv files and use them.
* [Bloomberg API](https://www.google.com/url?q=http://www.bloomberglabs.com/api/libraries/&sa=D&ust=1526285409807000): Multiple APIs available, including Python.
* [Quandl](https://www.google.com/url?q=https://www.quandl.com/home-v3&sa=D&ust=1526285409807000): Also multiple APIs, including Python.

Look for an API endpoint/library function that lets you obtain daily stock values such as Open, High, Low, Close, Volume and Adjusted Close. Remember that Adjusted Close is what you are trying to predict.

Tasks

Implement stock predictor

For your core stock predictor, implement:

* A training interface that accepts a data range (start\_date, end\_date) and a list of ticker symbols (e.g. GOOG, AAPL), and builds a model of stock behavior. Your code should read the desired historical prices from the data source of your choice.
* A query interface that accepts a list of dates and a list of ticker symbols, and outputs the predicted stock prices for each of those stocks on the given dates. Note that the query dates passed in must be after the training date range, and ticker symbols must be a subset of the ones trained on.

Test and measure performance

A basic run of the core system would involve one call to the training interface, and one or more calls to the query interface. Implement a train-test cycle to measure the performance of your model. Use it to test prediction 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.

(Note: Pick the training period accordingly so that you have ground truth data for that many days in the future.)

Build user interface

Once you’re iterated on your stock predictor a few times, and it is giving results you are happy with (say, predicted stock value 7 days out is within +/- 5% of actual value, on average), implement a more user-friendly interface that lets you specify stock(s) you are interested in and provides predictions at some pre-defined intervals.

You can extend the system to suggest good stocks to buy or sell, and when. You could also maintain a portfolio of stocks for the user to make these suggestions more concrete. Document these enhancements in your report, with diagrams, screenshots, etc.