# Equity Price Prediction with LSTM

Anthony Chi

**March 23rd, 2018**

**Project Proposal**

**Data Examination**

The goal here is to examine the available data, and to achieve the following objectives:

Compare Data Sources: many data sources are becoming unavailable, turned into paid services, or only available on proprietary platforms. Many webpages were inactivated or unable to be scrapped, and APIs were also inactivated. I was only able to access three data sources: Quandl API, NASDAQ, and finance.yahoo.com.

After I compared the three data sources, there are visible differences in Adjusted Closing and Adjusted Volume. There were also occasional missing data from Quandl API; the NASDAQ maximum historical data only spanned 10 years. Therefore, I decided to soley rely on finance.yahoo.com’s data.

Using the available features, including open, close, high, low, volume and adj. close, I generated additional features to explore the following:

* What am I trying to predict? Specific price, change in price, simple direction, ordinal direction? This will determine if this is a classification (bi or multi) or a regression problem.
* Are there any relationship between the independent and dependent features?
* Are the features distribution normal? Do I need to make any transformation?
* Are there any missing or outliers that I need to take care of?
* Do I need to use scalers, or other feature engineering techniques such as PCA?
* Is there a model that can be applied to all equities? Or do I need to train for each equity?

**Modeling and Optimization**

I will use two benchmark optimized models, which are XG Boost and Lasso regression. The goal is to beat these benchmark models with RNN, specifically LSTM, which should have builtin long-term and short-term memory that would help us outperform the benchmarks.

During the LSTM exercise, I will attempt to explore how the hyperparameters such as window, epoch, batch size, and LSTM constructs will affect the model performances. This will surely help me tune the model for better performances.

Note this is a regression and a classification exercise.

Also, this exercise may also be computationally expensive. If it is proven to be the case, I may need to use AWS resources to get through this project.

# Findings and questions

# Questions:

# Predicting next day price may not be the best thing to do. May want to predict price in 5 days, or even much more ahead?

# Should I predict volume?

# Should I include other features to predict price?

# Does dropout improve performance?

# Does multiple layers improve performance?

# Should we actually try to predict price? Or should we predict a range? For example, -1~1% is neutral, 1~5% is good, 5%+ is excellent, etc.

# Findings:

# Best results is with window 15 (about 3 weeks), epoch 2000, and batch-size 50-250.

# There may be upward bias, meaning all that we have seen is upward trend (most of the time).

# When selecting hyperparameters, training predict time, testing predict time, training error eval time, and testing error eval time are negligible.

# The larger the window, the longer the training time. The higher the epoch, and the smaller the batch size, the longer the training time.

# Best training error achieved is with window 15, batch size 10, epoch 500: 2.5616e-06

# Best testing error achieved is with window 10, batch size 500, epoch 2000: 7.58083e-05

## I. Definition

### Project Overview

### Problem Statement

To complete the exercise, here’s the steps that I took:

* Explore data –
* Clean data –
  + Outliers:
  + Missing data:
* Prepare data –
* Feature treatments –
* Model selection –

### Metrics

## II. Analysis

### Data Exploration

### Exploratory Visualization

**Target**

**Outliers**

**Missing Data**

**Delete Feature**

**Feature Types Transformed**

**One-hot Features**

**New Features created**

**Features Transformed**

**PCA**

### Algorithms and Techniques

### Benchmark

**III. Methodology**

**Data Preprocessing**

**Target**

**Outliers**

**Missing Data**

**Delete Feature**

.

**Feature Types Transformed**

**One-hot Features**

**New Features created**

**Features Transformed**

**PCA**

### Implementation

### Refinement

### **DecisionTreeRegressor (benchmark)**

### **KNeighborsRegressor:**

## IV. Results

### Model Evaluation and Validation

### Justification

## V. Conclusion

### Free-Form Visualization

### Reflection

### Citation and Sources

**Relevant Files and Folders**