CS 122 Project Proposal

Will Griffin, Joyce Liu, Steven Lymperis, Alexander Kahn

We plan to collect stock return time series data and merge this with additional information scraped from the web. For example, we might merge the returns on Apple stock with the returns on other tech stocks as well as Google trends data and scraped news articles about the iPhone. We still aren't sure what sort of data we want to include, but we plan to combine several data sources that we think might be correlated with stock returns.

We will then split the data into a train and test set and try to select the best 5 or best 10 models in the train set using a greedy algorithm and potentially information criterion like AIC or BIC. We are thinking about using not only linear models, but also using Python's machine learning libraries to try to fit LSTM models, for example. We would then pick the "best" model according to performance in the train set.

We would like this to work seamlessly with any major stock ticker such that we could input just the ticker and have our program spit out a model that it thinks is most appropriate. We could then ultimately test if this data mining approach generates any excess returns for the average ticker relative to risk.

Basic sketch of steps:

Data acquisition

- Learn to use Python's yfinance library to pull market data from Yahoo finance to start, will expand to other data sources later
- Write a function that will pull a selection of tickers using the yfinance library as well as other APIs or web scraping and save it as a csv file. We want this to work with potentially a very large amount of data.

Model selection

- Load data as a Pandas data frame and split 70-30 into train and test sets

- Write an algorithm to fit linear models (OLS regression, ARMA) as well as non-linear models (can use Python's TensorFlow library to fit LSTM models) and select the best models based on AIC or BIC

Model evaluation

- For each time point in the test set simulate a trade for each model and compare the Sharpe ratios of the model returns adjusted for some fixed trading costs, select the best model
- Output the best model's summary and its performance in the test set

Bonus if we have enough time

- Try to fit models to forecast not just returns but also volatilities