Neil Hinnant (nrhinnan) & Duncan Smith (duncan26)

Predicting the Present: Applying Google Trends Data toward Market Prediction

Vision

The stock market is a valuable tool for allocating capital to companies that need it while offering investors the potential for a healthy return on their investment. In the modern era, stock traders have heavily invested in high-speed computing technology to leverage ever-faster trading latencies to gain a competitive advantage over the market. The market for high-speed trading devices is saturated, and the market responds extremely quickly to any changes within itself. Predicting the stock market using real time stock price data is extremely difficult. The market reacts unpredictably, because traders are unpredictable. The market creates or destroys billions of dollars based on swings of public opinion concerning the actions of certain companies (or the market itself) on a daily basis. It is impossible for the common person to track stock price trends quickly enough to buy in before the market has already compensated for any shift. However, what if we could predict these opinion-based shifts in stock price before they happen? We propose to use Google Trends data as an aggregator of public opinion to attempt to predict significant swings in a market *before they happen*. The mechanism is the following: We expect that google searches for a stock or company will spike just prior to significant valuation changes (both negative and positive). We believe there is evidence that Trends data may allow us to catch these 'opinion bubbles' as they are forming, and allow a savvy trader to capitalize on opinion swings as they form.

Software Architecture

There are two primary components to this program, a GUI frontend and a Java (or other language) backend to handle procuring the Trends data. The GUI must display graphs of the search term frequency and allow users to submit terms they want to graph. Ideally, we would support graph manipulation, and allow users to graphs multiple Trends against one another. In this fashion, users could

pattern match to find appropriate times to buy into stocks. The backend has to grab the Trends data from Google and parse it (data is available from Google in CSV form).

It would be worth considering functionality to support displaying a stock price graph superimposed on the Trends data graph. This lets the user match trends spikes historically to stock price fluctuations. It is currently unclear how to get data from exchanges such as the NASDAQ or Dow Jones. This is a question that must be addressed in the design phase.

Long term (and most likely outside the scope of this class), we would like to include prediction algorithms that would allow the application to suggest courses of action to the user, based on the user's portfolio.

Challenges & Risks

This program is relatively simple from a programmatic point of view. The difficulty comes in acquiring the data. It is unclear how often we may request data from Google Trends, and there is no official Trends API. Additionally, this project falls into relatively uncharted territory, we do not know definitively how accurate a predictor Google Trends can be for stock pricing. We believe it to be very accurate at predicting market drops, but may or may not be accurate for predicting spikes. Another concern is the legality of using Trends data in this manner.

This project requires at least one person with GUI experience, and preferably as many as possible. This is a visual product, at its core.