STOCK SENTIMENT VISUALIZATION

KEVIN CRISPIE & WILLIAM LU

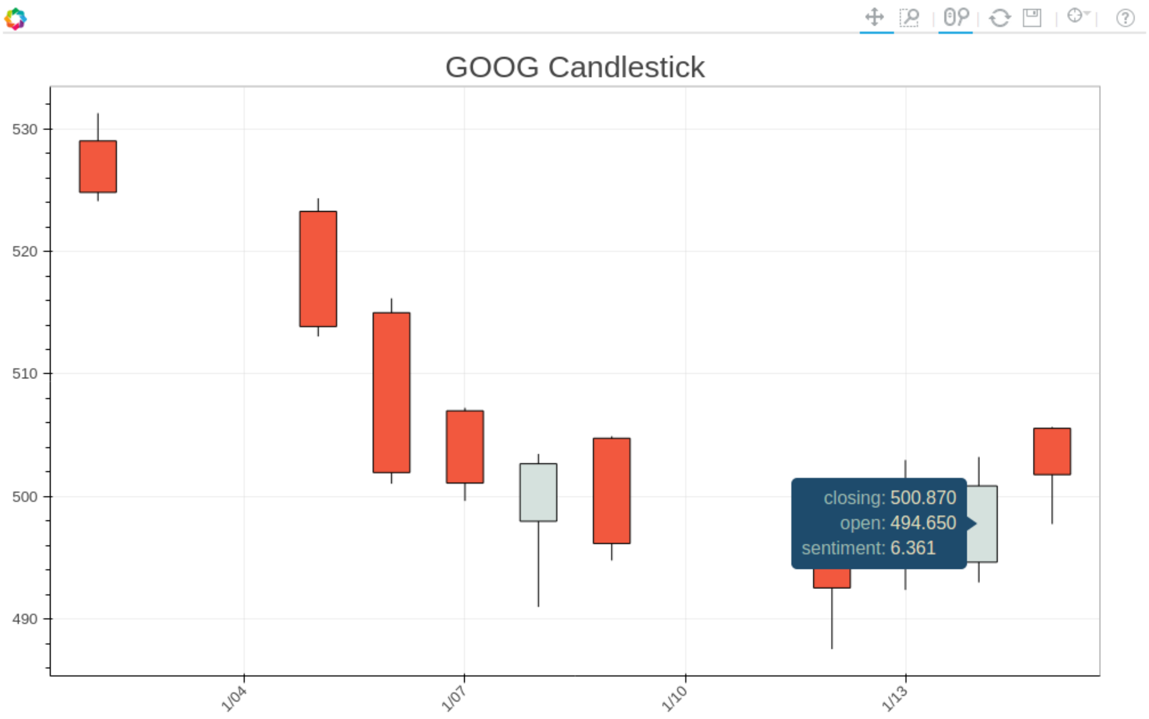
1. PROJECT OVERVIEW

We set out to correlate sentiment data of stocks and stock price, with an interactive element being tooltips on a candlestick chart showing users, per day, open/close prices and average sentiment.

2. RESULTS

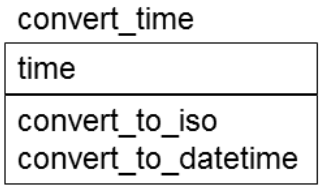
Our visualization first takes in user input of a stock ticker and date range and graphs a candlestick chart for that stock in that date range. The user can then interact with the chart by zooming in and out, panning around, etc. In addition, if the user mouses over a candlestick, a tooltip will appear. The tooltip will display open and close price data as well as average sentiment data for that day. Sentiment data is scaled like this:

Sentiment Value	Sentiment
4+	Very Good
1-3	Good
0	Neutral
-1 to -3	Bad
-4+	Very Bad



The data show that, unsurprisingly, there is a correlation (but not necessarily causation) between sentiment and stock price – on days when stock prices increase, unsurprisingly, sentiment is usually positive. If we had more time, we would probably be able to do a more fine grained analysis of our data and determine whether stock prices increase/decrease in response to sentiment or vice versa.

3. IMPLEMENTATION



Our code is not very object oriented for this project because there was no need to write object oriented code and we felt that the form our code is currently in is most efficient for this application. In addition, all the example code we found was not explicitly object oriented.

We process two main types of data: stock data and sentiment data. Historical stock data is pulled from the internet using the module ystockquote, which then returns the data in a dictionary. We then convert that dictionary into a dataframe that is then plotted by Bokeh. For tooltips to display properly, we further convert the dataframe into column data. Sentiment data is extracted from a csv file that we downloaded from SentDex, a company that runs sentiment analysis on stocks. That data is then also converted into a dataframe and then into column data in order for Bokeh to properly display it in a tooltip.

One major design decision that we made was to use the Bokeh module, a module that supposedly helps to “quickly and easily create interactive plots.” We originally chose Bokeh because we thought it would help us easily create interactive plots and would provide a lot of core functionality for our visualization.

4. REFLECTION

It turns out that this was a horrible design decision – a more accurate description of Bokeh’s function would be to “attempt to quickly and easily create interactive plots, but only after an intense amount of pain and suffering as well as an immense investment of time and energy.” Bokeh is documented, but poorly, with extremely buggy pre-alpha standard code.

On top of making a terrible decision to use Bokeh, we inaccurately scoped our project. We started out thinking that we would create a sentiment analyzer that would analyze financial articles for certain stock tickers and return a sentiment value for that stock. We scrapped that portion of the project due to lack of time. We also wanted to create an extremely interactive plot beyond the tooltips that we have implemented – the user would be able to pick from a list of stocks (or input a ticker for any stock they would want to look at). Additionally, sentiment data would be graphed as a bar graph where a traditional volume graph on a stock chart would be. This would allow the user to visually correlate sentiment data with stock rise and fall. Because of Bokeh’s poor documentation and frustrating problems with implementation, we also gave up on that dream in favor of having working code.

We both decided to work in parallel for optimum efficiency and it actually worked really well, especially after we figured out how to use branches in Git to avoid merge conflicts.