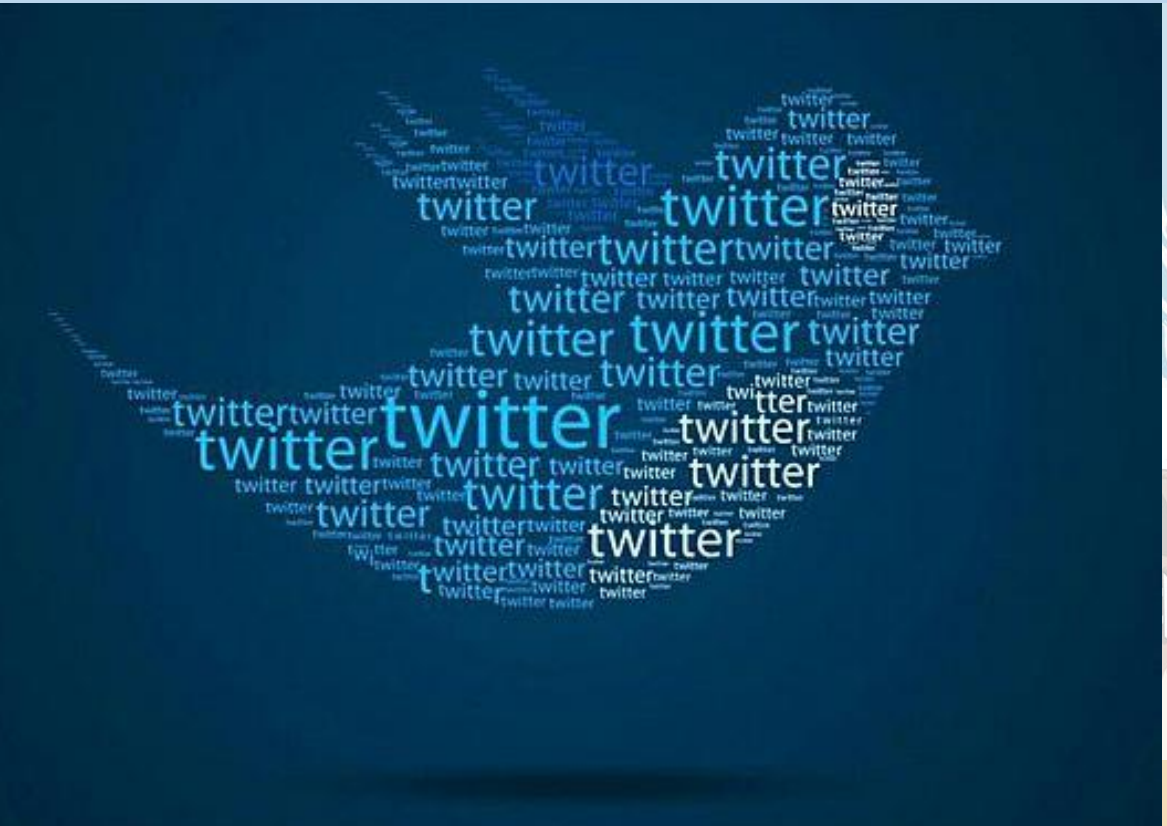
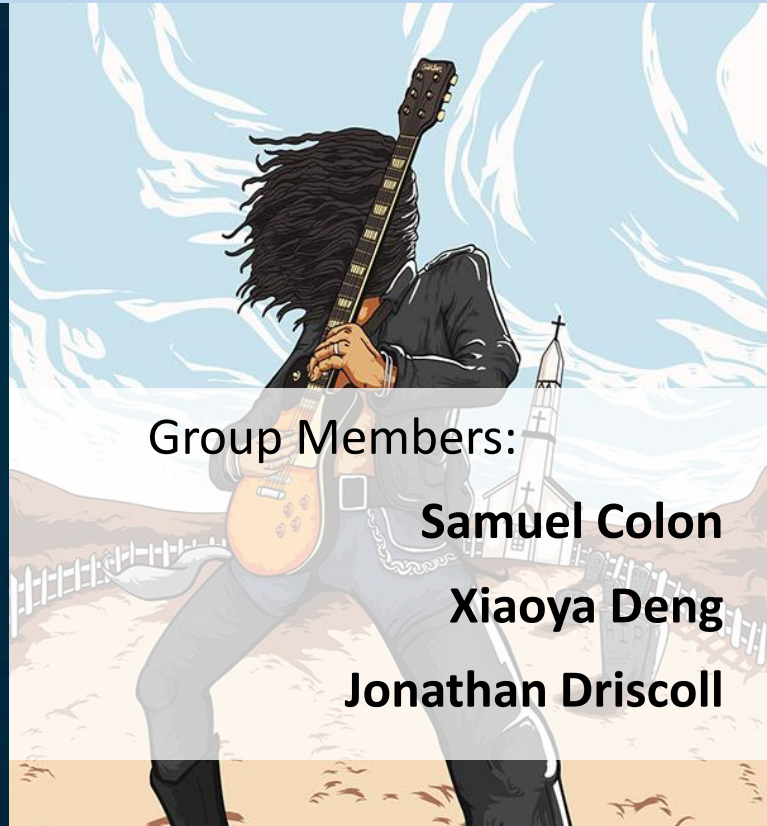
November_Rain Presentation





Group Members:

Samuel Colon

Xiaoya Deng

Jonathan Driscoll

Project Process

November_Rain

Step 1

- Read .json files into Spark RDD

Step 2

- Build a base model – to describe the closing price of a company's stock as a function of relevant word frequency

Step 3

- Improve the base model using sentiment analysis

Where We Are

November_Rain

- In the process of **building a simple model** which describe the closing price of a company's stock as a function of how many times that company has been mentioned on twitter;

- A simple scatterplot of price vs. Twitter mentions per day

- Potential problems:

  - Only 6 days of data
  - More granular financial data is not cheaply available

  → We do not anticipate this model being very helpful
  → But it is a good starting point!

What We Have

November_Rain

- Code to **unzip** a directory of .json files;
- A **dictionary** of words and their corresponding ‘sentiment value’ based on the McLougran data set;
  - About the data set. It has over 8,000 English words: positive, negative, or neutral in sentiment
- Code to return the **number of times** a specific word is mentioned and to find the stock ticker data using the yahoo_finance package;

- ✓ We now have all the working parts to create our base model!

Challenges We Have

November_Rain

- **Scalability issues**:
  - 100GB data is a lot for our laptops
  - The entire training data set
  - We have compromised by reading 30 – 60 json files at a time (one hour of twitter data)
  - ✓ **Possible solution**: to push the entire training set to AWS and run our training data there.

Challenges We Have

November_Rain

- **General programming issues**: (in particular type conversion issues)
  - We must determine the most efficient strategy with regards to data manipulation
  - Our approach: to deal exclusively with the data as either an RDD object or a pyspark dataframe, with lots of alternation between them
  - We have found: manipulation and transformation of these data types is not ideal
  - ✓ **Possible solution**: to perform any cleaning/filtering of the files while they are still in json format, since these can be read as strings or dictionaries, which are much more flexible data types.
  - ✓ We are not yet sure what effect this approach will have on scalability.

Where We Are Going

November_Rain

- **Our general approach:**

1) Filter out all irrelevant tweets and any superfluous information contained within them(non-english, does not mention one of our companies, etc), and store as a new .json (or bundle of jsons)

2) Pass a function that analyzes each tweet one at a time. This function will compare every word in the tweet to the sentiment dictionary we have created

3) Calculate a 'daily sentiment score' by tallying all scores from tweets mentioning a company. Plot closing price as a function of this daily sentiment score. Compare to base model.

4) If there is a correlation between price and sentiment score, use same approach on a test dataset to predict future closing price based on that day's 'daily sentiment score'

Where We Are Going

November_Rain

- If time permits we will fine tune our NLP approach. Although the loughran database is a great start, it is not extensive and categorizing words as either completely positive or negative is a crude approach to NLP.
- It would be great to apply some more advanced NLP techniques to our data.