Katie Dillon
Maya Genson
Eva Smith
SI 206: Winter 2019
April 29, 2019

Newsies' Final Project Report

Original Project Goals:
The plan was to find varying data from 3 different APIs in relation to politics. We wanted to see how we could compare and contrast the sites in number of articles, words used, sentiment, etc.

Goals Actually Achieved:
- Found certain terms (example: politics, impeach, etc.) within data accessed with the API keys and then processed that data to generate meaningful charts.
- Found themes in our data and created visualizations to show common words and topics
- We were able to analyze the sentiment of content from varying news sources (data from newsapi), finding the average polarity scores.
- Visually showed how reddit user scores and number of comments are related.
- Calculated how many article headlines contained particular search terms.

Problems Faced:
- NYTimes API was a challenge to figure out how to limit the searches and rows added to tables.
- Figuring out what data we could actually pull and how we could use it was also a challenge.
- We had to be aware of whether the content we wanted to use for calculations existed.
- We learned how to have a better flow in working with our shared Github repository; it was a rocky process at times.

Calculations Files (There are three files with the written calculations)

News Outlet and Average Polarity Score:
Yahoo.com, -0.00703333333333
Sciencemag.org, 0.4215
CNN, 0.187575
The New York Times, 0.0608344444444
Engadget, 0.200611111111
Mashable, 0.194926388889
Wired, 0.45553
The Times of India, -0.3182
Entrepreneur.com, -0.0716666666667
Gizmodo.com, -0.327663333333
BBC News, -0.2324
TechCrunch, 0.290414

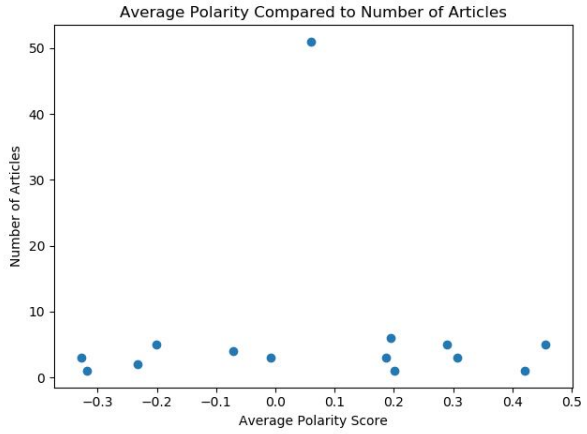
The Verge, -0.2003
Lifehacker.com, 0.306666666667

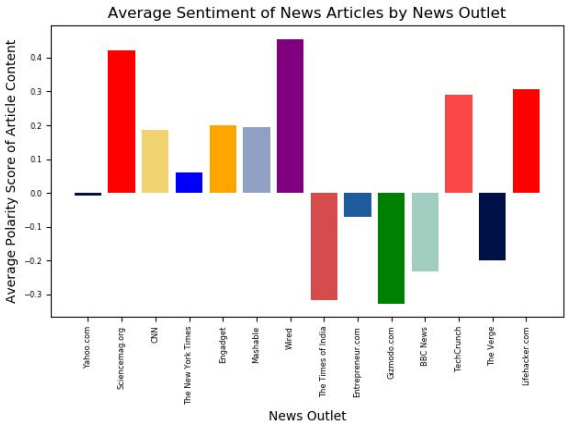
NYT:
HEADLINE COUNT:

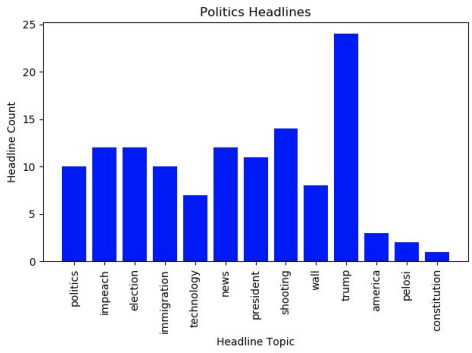
POLITICS: 9
IMPEACH: 8
TRUMP: 5
AMERICA: 2
PELOSI: 2

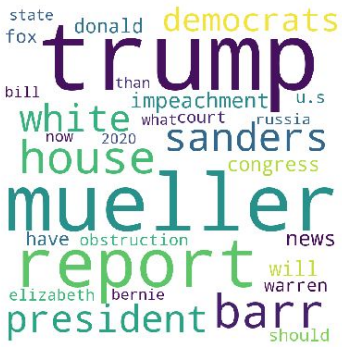
Word occurrences in Reddit posts used to create word cloud
==========
trump: 581
mueller: 400
report: 281
barr: 108
house: 102
sanders: 74
president: 69
white: 68
democrats: 64
impeachment: 64
news: 61
will: 58
congress: 53
donald: 51
have: 48
fox: 47
warren: 44
should: 43
u.s: 42
court: 42
obstruction: 41
elizabeth: 37
now: 36
…(many more rows within the text file)

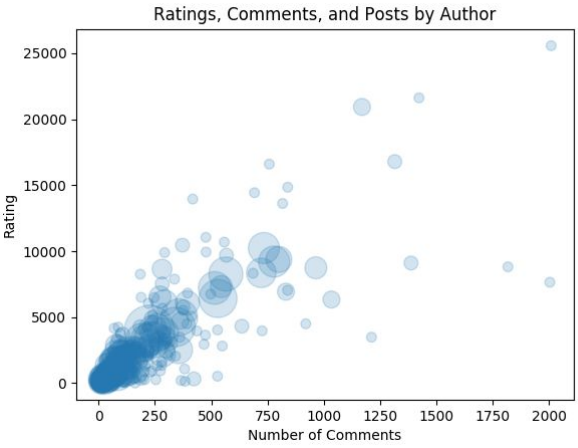
Visualizations:

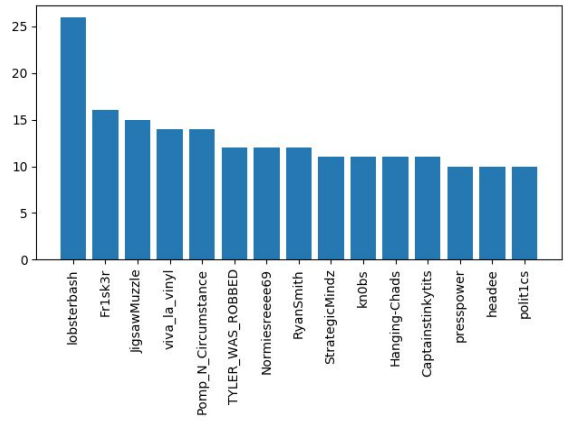












Instructions for Running the Code:

- If the database is empty, you must run the three files that will populate the three tables: reddit.py, news.py, and NYTimes.py, which will pull data when you have the appropriate keys.

- If the tables are populated, you can then run the visualization files to complete necessary calculations before generating the final visualizations.
- Any other instructions are within the python files.

Functions Explained (includes input and output):

- Reddit API:
    - reddit.py
        - **scrape_reddit_politics()**: This function connects to the Reddit API and collects data from the politics subreddit. Then, it connects to our SQLite database and adds data to our database if that piece of data does not already exist. The data collected includes submission ID, author, title, content, link, score, and number of comments for each post. This function also updates data in the database if the score or number of comments have changed since the data was initially added. For clarity when collecting data, this function also keeps track of and prints out the number of posts that were successfully added and the number of posts that were successfully updated.
    - vis.py
        - Class: RedditStats
            - **__init__(self)**: This function connects to the SQLite database and selects all rows. Rows are stored as a list of tuples in a member variable self.data.
            - **write_to_file(self, filename, data)**: This function takes in self, a filename, and a dictionary called data. It creates a file if one does not already exists and then writes the values of the dictionary to the file in descending order.
            - **generate_word_cloud(self, write=False)**: This function creates a dictionary that maps words to number of occurrences and uses that dictionary to create a word cloud. The function also ignores some common words like “and”, “it”, “a” and more so that the more relevant words show up in the word cloud. It also uses natural language processing to ignore punctuation and symbols. If the write flag is set to True, the write_to_file function is called.
            - **most_common_authors(self)**: This function goes through the authors on the politics subreddit and finds the 15 authors who have made the most posts. The function then plots these authors on a bar graph.
            - **authors_numPosts_ratings(self, ratingOffset=0, commentsOffset=0)**: This function creates a scatter plot using three different variables: number of posts, average rating, and number of comments. The default arguments allow the user to run the function and eliminate outliers if desired. The function creates three dictionaries that map authors to ratings, posts, and

comments. Then, the function combines this data and plots one point for each unique author. The size of the point corresponds to the number of posts that author has made, the distance down the x-axis corresponds to the average number of comments received on that author's posts, and the distance up the y-axis corresponds to the average rating received by that author on their posts.

- Newsapi:
    - News.py:
        - news_scrape(): this function creates a connection to the *Final.sqlite* database file, and creates the *News* table. It searches the newsapi for everything it has for a particular search query, which is set to 'politics'. Then it takes the data that is returned from the api and sets pieces of the data to created variables, which are then inserted and written into the appropriate *News* table columns. It is keeping track of how many rows are added using the num_items variable.
    - News_viz.py:
        - Class: NewsSentiment
            - Init method (takes in self) to set up connection to the database and select all items in the *New* table. This data is set equal to a reusable variable in other class methods. There are also other variables created to be used in other class methods as there were going to be a number of necessary calculations: self.raw_sia_dict, self.outlet_counts_dict, self.avg_sia_dict, self.total_articles
            - articles_per_outlet method: takes in self, outputs a counts per outlet dictionary (self.outlet_counts_dict)
                - Keys are news_outlet and the values are the number of items from that particular outlet that are being analyzed
            - content_sentiment_calculator method: takes in self, outputs raw total sentiment scores per news outlet
                - It counts the number of sentences in each row's content column.
                - It checks that the content column actually contains a string and makes sure it is not a useless piece of information ("Chat with us…")
                - It splits the content by sentences to get a better analysis
                - Each sentence for an article is passed through the nltk sentiment analyzer and a raw polarity score is calculated. These scores are added together and then divided by the total number of sentences for that particular row to calculate the average sentiment score. These averages are put into a dictionary where the keys are the news_outlet and the values are the added average scores.

        - avg_sentiment_per_outlet method: takes in self, outputs dictionary of average sentiment scores (polarity) for each news outlet in the table (self.avg_sia_dict)
        - sentiment_chart method: takes in self, outputs bar chart of average polarity scores for each news outlet; it uses the self.avg_sia_dict keys and values
- NY Times:
    - NYTimes.py
        - Def get_dict(): This function connects to the NYT api content and turns the returned data into a json dictionary.
        - Def scrape_nyt(): This function takes in the json dictionary that is returned from get_data and loops through the dictionary to check if the keys have “response” or “docs” or none, as we found that some did and some didn’t. If they didn’t it returned a key_error. If we use the loop we can still get the results we need without returning an error. This function will return a dictionary.
        - Def politics_data(): This function takes in the dictionary from scrape_nyt and creates a database with the tables: url, headline, date and source. If it scrapes more than 20, the function will break. Otherwise, it will go through the dictionary and add data to the database and then execute the same database.
        - Def visual_nyt(): This function calculates how many times specific terms appear in each scraped headline, and put that information into a dictionary. We’ve set the code to change all terms to lowercase, so that it will count terms that are both upper and lowercase. This data will be used for a visualization.
        - Def calculations_file(): This function will take the calculations and headlines from visual_nyt and write that into a file, which will list how many of each headline there was.
    - NYT_viz.py
        - Created a class in order to organize all of the method as well as being able to call previous methods later on.
        - __init__(self): This method connects with the Final.sqlite file and retrieves the information collected within the database.
        - Def get_dict(self): This method creates a dictionary, and then searches within the data in the database as well as searching the terms. If the search yields positive results, then that term is added to the dictionary and will increment as more headlines for that term is found.
        - Def bar_chart(self): This method creates a bar chart based on the results of the dictionary from get_dict.

Resources:

Date	Issue Description	Location of Resource	Result (issue solved?)
4/27/19	bar chart labels were cut off at bottom	https://stackoverflow.com/questions/6774086/why-is-my-xlabel-cut-off-in-my-matplotlib-plot	plt.tight_layout() solved the issue
4/27/2019	Wanted to make the font size smaller on x and y ticks	https://stackoverflow.com/questions/6390393/matplotlib-make-tick-labels-font-size-smaller	Yes, issue resolved
	Needed to know how to get the data from the news API	https://newsapi.org/docs	Yes, it helped to know what parameters were possible.
4/23/19	Needed to download nltk and libraries	https://www.nltk.org/data.html	Helped, but issues with download.
4/24/19	issues with nltk libraries download certification error	https://stackoverflow.com/questions/38916452/nltk-download-ssl-certificate-verify-failed/39142816	Yes, found code to work around the issue; successful download of nltk libraries
	Wanted to use some natural language processing to help parse unwanted words for the wordcloud	https://spacy.io/	Installed spaCy, a natural language processing module, and used it to remove symbols and unwanted words
4/27/19	Find out how to use the NYT API	NYT Developer https://developer.nytimes.com/docs/articlesearch-product/1/overview	Showed how I need to format code to search within the api.
4/29/19	Checking documentation of bar chart visualization from a dictionary	https://plot.ly/matplotlib/bar-charts/	It did clear some things up.