Nicholas Chakmakas and Jeffrey Alhassan - DS4900 Final Report

To view the webpage, which contains a condensed version of this report and the interactive visualizations, visit:
https://pages.github.ccs.neu.edu/njchakmakas/DS4900-Senior-Project/

Repo link: https://github.ccs.neu.edu/njchakmakas/DS4900-Senior-Project

Introduction

For our project, we wanted to compare sentiment over the past year with overall covid cases, to see if there is any correlation between the two variables. To go about doing this, we are handling the data at a city level. We are tracking for the following cities:

- Boston
- Chicago
- Miami
- New York City

By taking from various cities, we are sampling from multiple parts of the country. Covid was handled differently in parts of the country, and this may translate to a difference in overall sentiment that we can trace and measure over time. In addition, New York City was the epicenter of the virus during the beginning of the pandemic, so this could have a tangible impact on the sentiment of Reddit posts during that time frame of the initial months of the outbreak in the US.

Comparing sentiment trends to covid case numbers could result in a correlation that can be tracked in the future and result in a way to determine how covid numbers will change with a changing sentiment.

Part I: Data Collection / Manipulation

Initial method:

Data was collected from a third party Reddit API, pushshift.io (https://pushshift.io/). Using this API, we constructed queries for the data that we wanted using the following parameters, structured as such:

- **Query=**”subreddit={sub}&before={before}&after={after}”
- *After***:** Gets posts after the given timestamp

- *Before:* Gets posts before the given timestamp
- *Subreddit:* Gets posts from only the given subreddit

We collected for each city's respective subreddit for the date range of 3/1/2020 - 3/1/2021 to get an entire year's worth of data. Using pushshift's API calls directly is what we've done up to this point. We use the *requests* package to get the json from the endpoint, and return the data field within that json to get the relevant data. Since each query to the API can only return 100 rows, we built a workaround that advances the timestamp ranges of the query and iterates over calls to the API until the entire year is covered. However, we are perfecting data collection and recently found that psaw (Pushshift API Wrapper for Python) will get us every post for the given time range; currently, the direct API calls only collect a few posts per day, totalling in about 1300 rows per subreddit. This data was then converted to a dataframe and stored as a CSV.

Updated Method:

To get all rows of data, we transitioned from using direct calls using requests to using psaw (https://pypi.org/project/psaw/), a Python wrapper for the pushshift API. The data we received using this method was significantly more substantial than our previous one, with each dataset going from containing about 1300 rows to over 20,000 in most cases.

For covid data for each city, we will be using data provided by the New York Times on Github: https://github.com/nytimes/covid-19-data. This data has granularity at the national, state, and county levels. We will be using the county levels to emulate the city level. In addition, we will be calculating the daily total number of cases based off of the cumulative number of cases, which is what is provided in the dataset.

There were a few odd findings in the covid 19 data from NYT. There were a few days where there were no cases reported in some datasets (such as 12/25 or 1/1) or in some cases, days where the total number of cumulative cases went down (resulting in a negative number for daily cases). Since there were only a few rows with this issue, these days were discarded from the dataset since they are clearly incorrect. Finally, to smooth out the outliers in the data, we opted to compute the daily average on a week-by-week basis, starting on every Sunday for that week.

Part II: Initial Sentiment Analysis with TextBlob

Once we had the data collected and properly formatted, we began sentiment analysis. Initially, we utilized *TextBlob*, a python framework that categorizes text into a polarity in

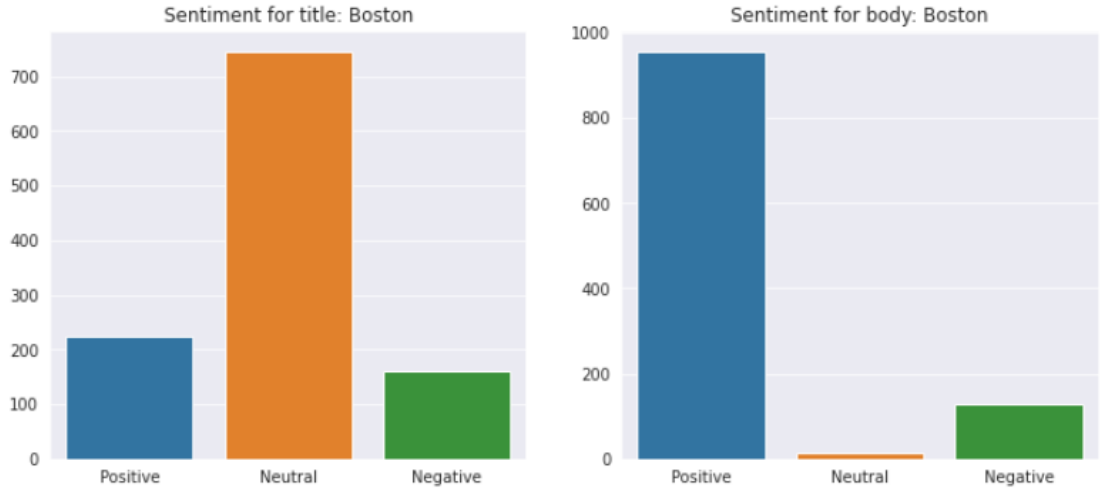
a range [-1, 1], with negative values associating to a negative sentiment, a zero value associating to a neutral sentiment, and a positive value associated with a positive sentiment.

There are two major parts of a reddit post that we would like to analyze: the **title** of the post, and the **body** of the post. By performing sentiment analysis on both of these components, we can determine the overall sentiment of it, and we plan on taking the average sentiment over time and analyzing its change.

Typically, the title is kept short with the details and full sentences occurring in the body. Some examples of titles versus bodies:

Title	*Body*
Mid-Cycle Recap	Hello everyone! I just found out this subreddit existed and I definitely feel comfortable here, since r/LSA is so heavily T14 focused. Iâ€™m applying to two T14s as long shots, but here is my mid-cycle recap!\n\nStats: 3.69/165/non-URM/T3 softs\n\nAccepted: American\n\nUR: GW, Georgetown, George Mason (interviewed)\n\nComplete: University of MN, University of Maryland, Emory\n\nApplying January: Boston College, Boston University, UPenn
Humoring the Harden Enthusiasts. Looping in a 3rd team for Kemba might be a path.	53.3 mil in outgoing salary. Harden himself only makes 41.2 mil. So even if Houston also sent back 35 year old PJ Tucker (7.9), a vet that could actually help us, we're still only taking back 49.2 mil in salary. Which, if you've been following along with the TPE deep-dives, would actually increase our current "hard" cap from 22 mil…[*body shortened due to length*]

However, our preliminary results were not promising and it was difficult to determine the source of the issue, since we were dealing with a model that we did not train ourselves. Our preliminary results for our cities were detecting the bodies of our posts as overly positive. One example is below; all cities follow the same pattern of having very positive sentiments in their subreddit post bodies:



Part III: Updated Sentiment Analysis

To make our own model, we first started with finding data that is pre-labelled with sentiment so that we could use that for training, as opposed to manually labelling data ourselves. Initially, we wanted to have our model track sentiments in three categories: *positive, negative* and *neutral.* To achieve this goal, we were working with an Amazon reviews dataset with each review labelled 1 through 5 stars; intuitively, we chose any review less than 3 stars as negative, more than 3 stars as positive, and 3 stars exactly as neutral. However, this model performed poorly on our test set in the neutral category, even after experimenting with the sentiment thresholds. Most neutral sentiments were getting labelled as positive or negative, when it is expected that most sentiments are neutral and less are explicitly positive or negative.

To solve this issue, we utilized the IMDB movie dataset (https://www.kaggle.com/lakshmi25npathi/sentiment-analysis-of-imdb-movie-reviews) with sentiments labelled positive or negative only, and used the probability of a positive sentiment to determine whether a post was positive, negative or neutral. By using the probability instead of a discrete binary label, this allowed us to set the range for what a neutral post should be. Our bounds were the following:

- *Negative*: $p <= 0.4$
- *Neutral*: $.4 < p < .6$

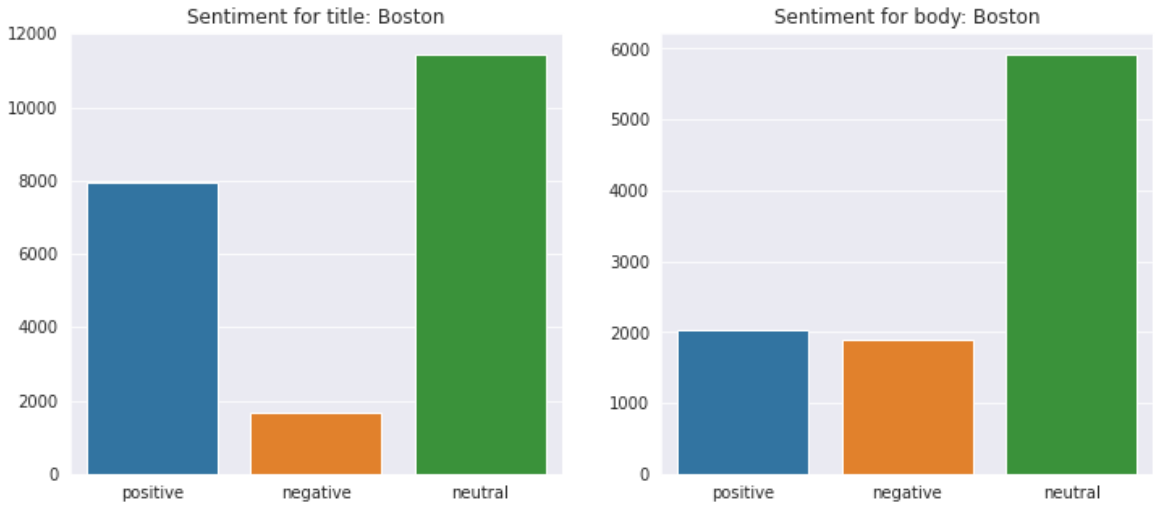
- *Positive*: p >= .6

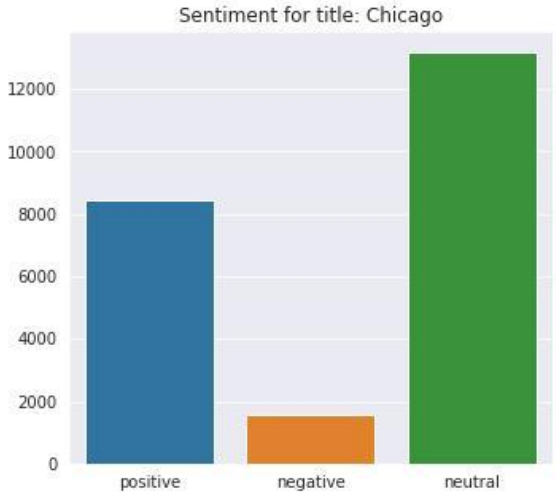
To actually train our model, we first had to vectorize our text into a matrix of features so our model could interpret the text data. Once we vectorized our text, we used Logistic Regression as our model of choice due to our data technically having a binary output, even though we will be utilizing the probabilities instead of the actual output values themselves.

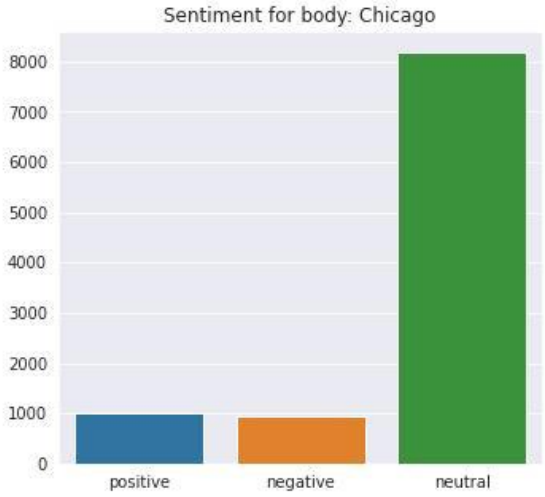
Our model performed well when we switched to binary data, with over 80% accuracy and recall for both labels:

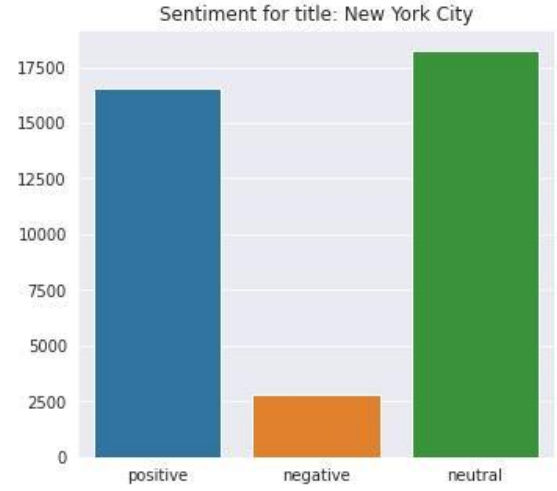
	precision	recall	f1-score	support
negative	0.87	0.81	0.84	489
positive	0.83	0.88	0.86	511
accuracy			0.85	1000
macro avg	0.85	0.85	0.85	1000
weighted avg	0.85	0.85	0.85	1000

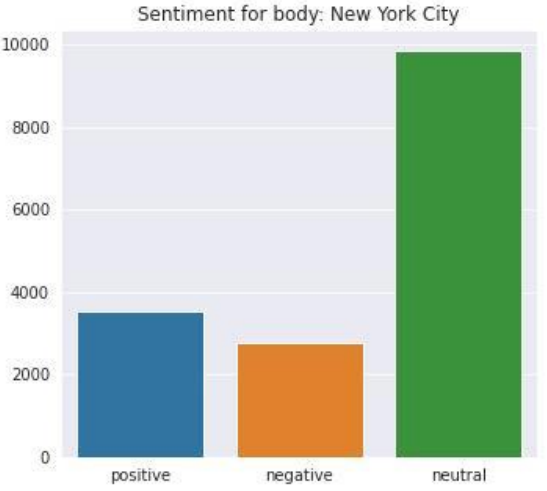
Our results were more along the lines of what we had expected, with most sentiments being marked in the neutral category for all cities. Below are total counts for both the body and title for each city's subreddit:

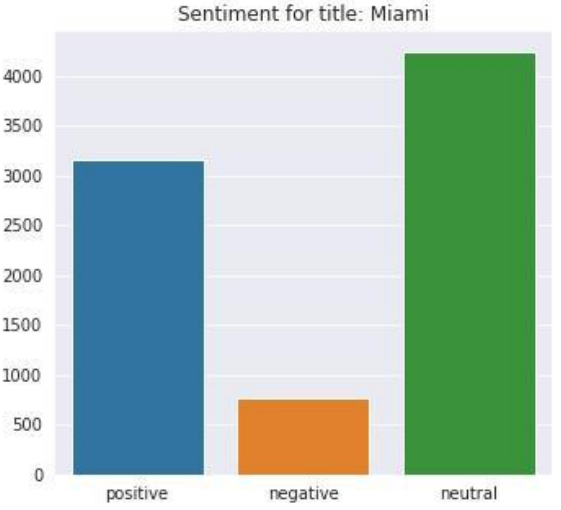


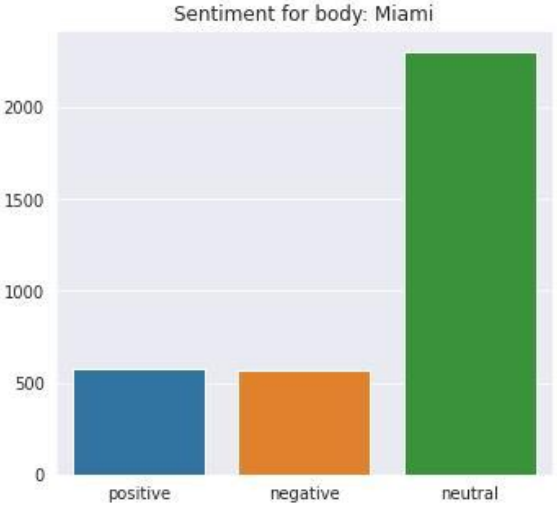

Sentiment for title: Chicago
12000
10000
8000
6000
4000
2000
0
positive
negative
neutral


Sentiment for body: Chicago
8000
7000
6000
5000
4000
3000
2000
1000
0
positive
negative
neutral


Sentiment for title: New York City
17500
15000
12500
10000
7500
5000
2500
0
positive
negative
neutral

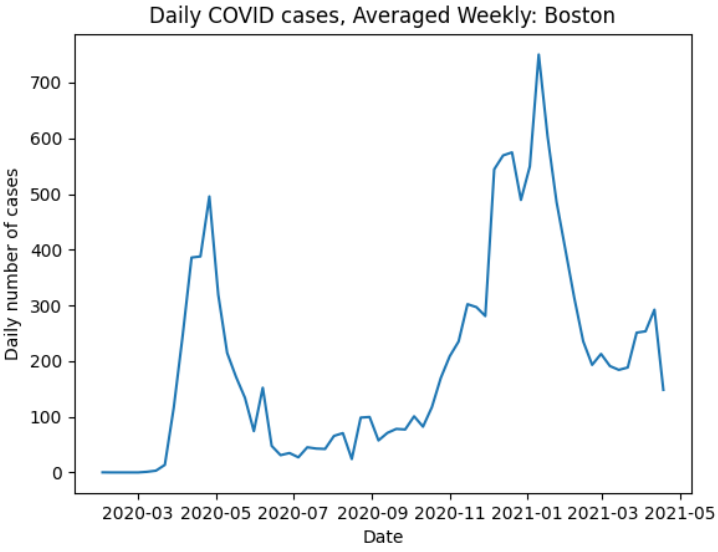

Sentiment for body: New York City
10000
8000
6000
4000
2000
0
positive
negative
neutral

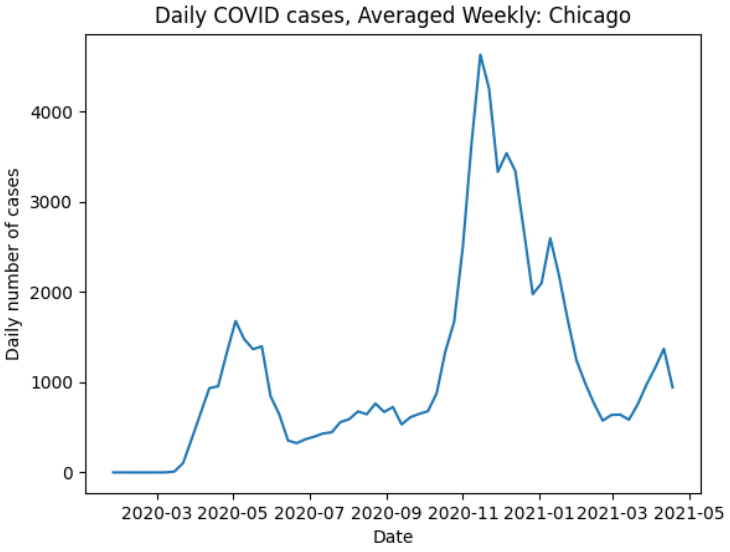

Sentiment for title: Miami
4000
3500
3000
2500
2000
1500
1000
500
0
positive
negative
neutral

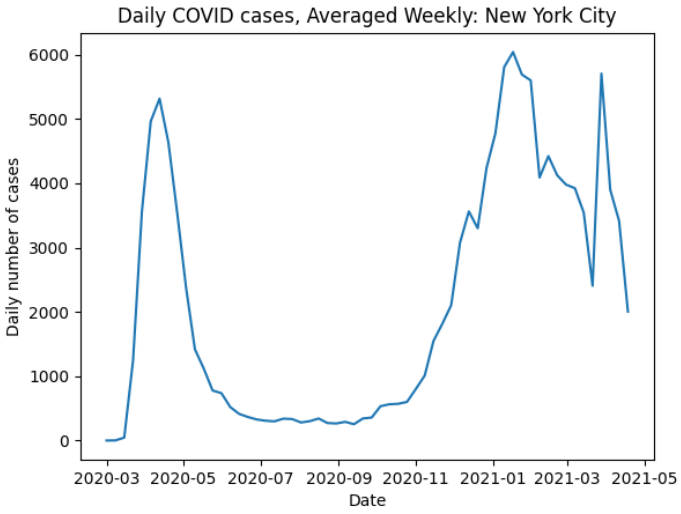

Sentiment for body: Miami
2000
1500
1000
500
0
positive
negative
neutral

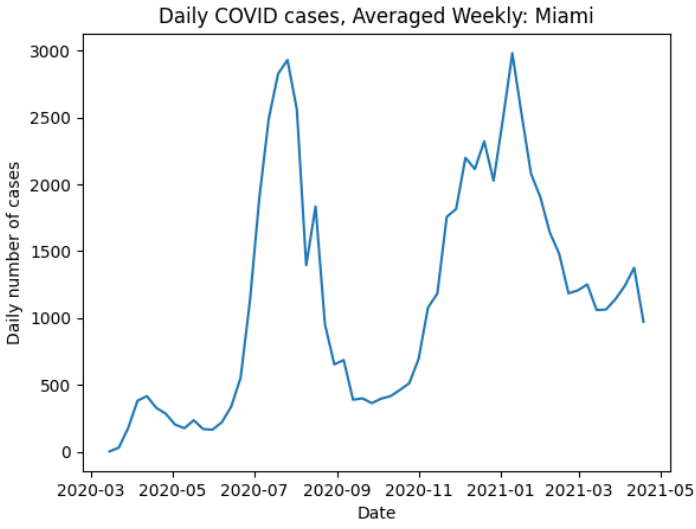
Part IV: Visualizations

One visualization that we felt was important was showing average covid cases and average sentiment over time. When plotting covid cases, first we wanted to observe trends at the city levels themselves. To smooth out the graph in the presence of outliers, we opted to plot the daily average number of cases per week for each city.



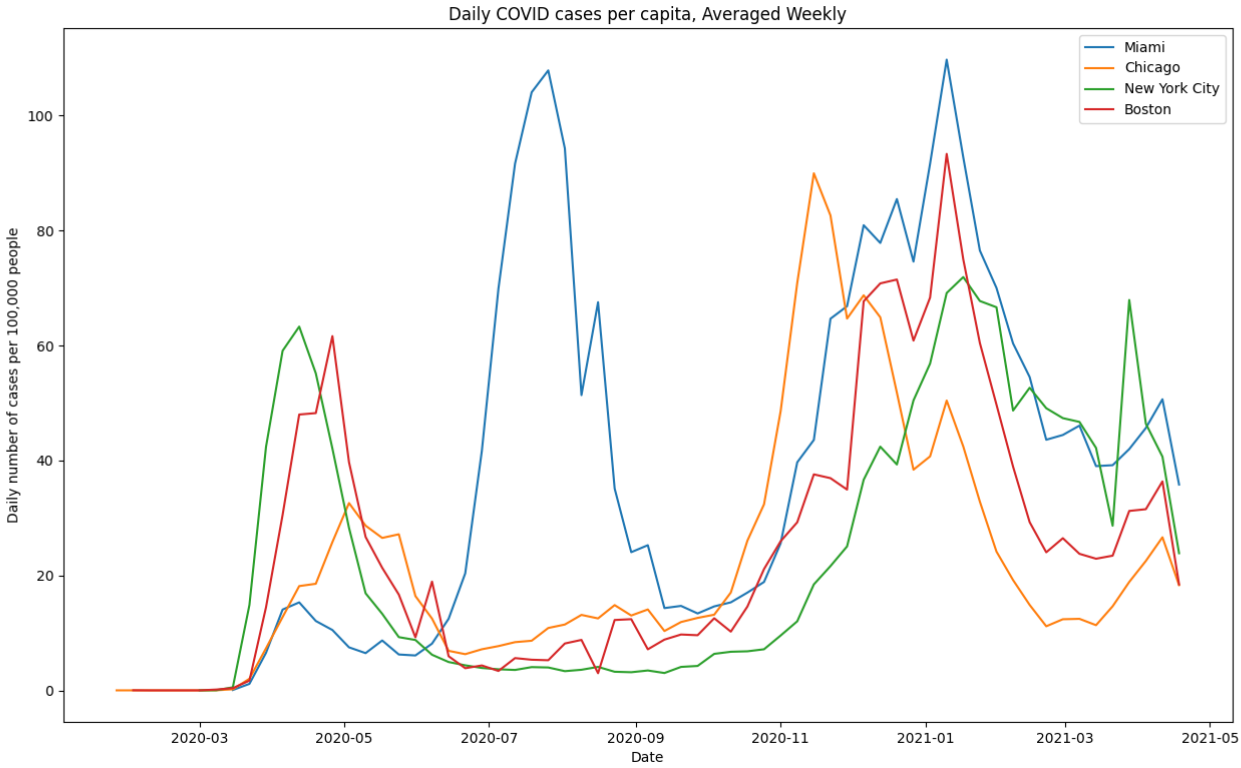






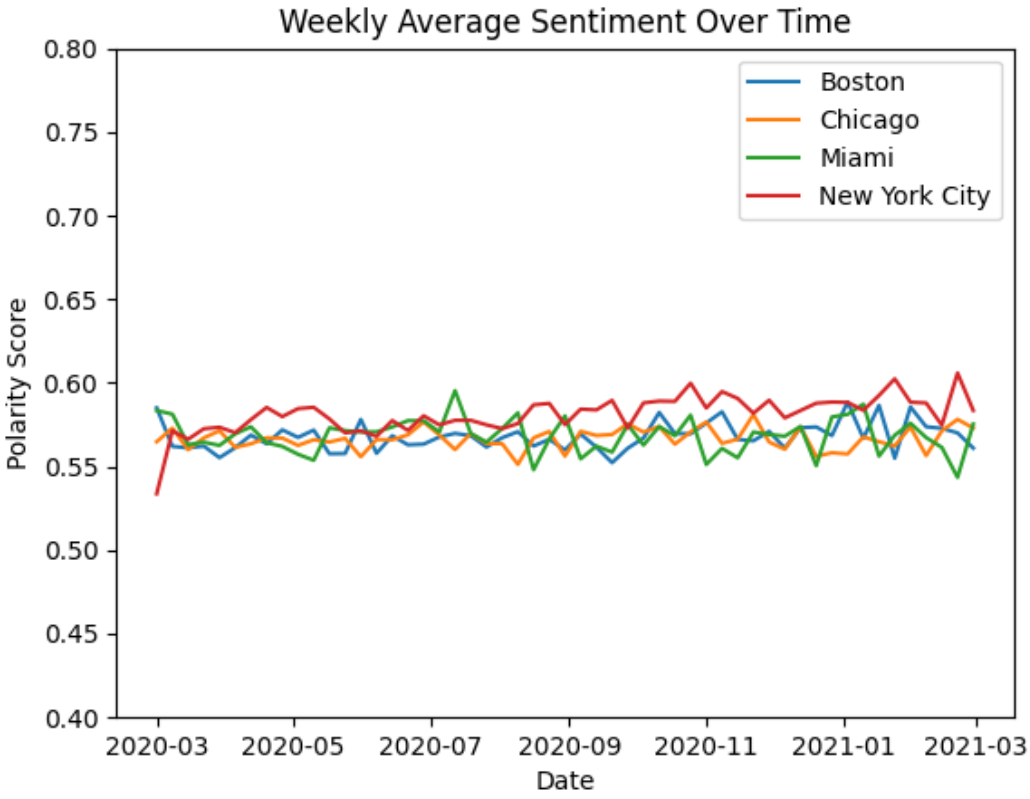
Each graph has an interesting trend, with some unexpected surprises as well. We see the initial peak in New York, and smaller initial peaks in Chicago and Boston. In Miami,

there are spikes in the summer, and for all cities, covid case numbers have been up during the winter months and are just starting to make their way down. But, when we compare all cities to each other on a per capita basis, there's even more information to extract here.



The above graph shows the average daily covid cases per week for each city, per 100,000 people (Population numbers for each county obtained from 2020 Census Data). We see that the initial outbreak in NYC was actually not the worst we've seen, with a higher percentage of cases in the winter months for almost all the cities. Also, Miami's peak in the summer was a surprise to us, since news coverage of Covid-19 was mostly focused on the earlier months, with NYC at the forefront. It is starting to look like covid case numbers are finally going down after a tough winter.

As for sentiment, the average sentiment seems to be relatively steady throughout the pandemic. This is understandable, since the majority of posts are falling in the neutral category anyways, with very little change:



Interestingly, New York's sentiment was lowest during the first week of the pandemic, and slowly but consistently became the highest average sentiment, especially during the winter months.

Something to also consider is the high volume of posts we collected, which would make it more likely to follow a trend of staying towards neutral, since it's understandable that many people creating reddit posts do not use strong, suggestive language and would be detected as neutral under our model.

Part V: Building Interactive Visualizations

To build interactive visualizations, we opted to use D3, since we could host a static webpage on Github and convert our data into the format that would make it usable for D3. After choosing a template from html5-templates.com, we then could build upon that with our D3 visualizations. With our sentiment polarity scores over time, and our average daily covid cases for each city, we were able to construct an interactive graph that allows you to see the values of each on a given day. We did this by creating two graphs on a single SVG object, with the same X-scale for each graph. Once that was created, we also built a vertical line to easily track where you are in the graphs, since it can be difficult to do so when two graphs are stacked on, as well as a dialogue box with exact values. D3 was tricky to work with, so this proved to be a more difficult task than it seems. However, we felt that it was the best method to use so we could host the web page the way we liked. In addition to the main line graphs, we also included a dynamic bar graph for the total sentiment counts.

One issue we faced when creating the line graph was the cutting off of the square when you reach the end of the graph; that is something that we are aware of and weren't able to quite figure out how to fix in time.

Part VI: Conclusion

Covid-19 has been difficult on everyone for over a year now. People are exhausted and ready to resume their normal lives, and we were hoping that some of this would be reflected in our sentiment data. Unfortunately, we didn't find sentiment trends that were conclusive, since the majority of posts fall under the neutral category. However, we must also remember that the sentiment of Reddit posts could be different than the representation of how we feel as a whole. Perhaps we are more reserved when posting online, leaving only a handful of people who voice their opinions strongly.

An interesting aspect of the project that we found was the covid case data from the New York Times on its own. It is interesting to see how different areas dealt with the pandemic, especially since during the summer months when people were starting to go outside and take their attentions elsewhere; it revealed that in some cases, like Miami, some places had it worse when we weren't aware of it. And, from this project, we found that in the winter months, we're actually seeing more confirmed cases than in the initial stages. This could be due to the increased level of testing, but is still an intriguing story from our data.

Overall, this project was a great learning experience, from dealing with working through the Reddit API, to building a model, to creating a final visualization to wrap up our findings. To build upon this project, one could use data from other sources of the web to compare sentiments across different social media platforms; this could also be done without the lens of Covid-19, to see if some platforms are more “negative” or “positive” than others.