Sentimental Analysis on News Articles of Varying Political Biases

By: Archita Srivastava, George Cheng

Instructor: Steven Bergner

2024.04.11

Background

News agencies often use various tactics to influence viewers toward their political stance. These methods include appealing to emotions, omitting important context, and sometimes disseminating false information. As global polarisation increases, it is increasingly crucial to understand how news outlets manipulate readers' emotions and perceptions.

Question

Hence, this project aims to explore potential media bias by questioning whether news outlets from different political standings `(Left, Right and Centre)` use sentimental language, namely positive or negative, to report on topics with suspected polarisation such as Abortion, Immigration, etc, which could influence an emotional reaction amongst readers.

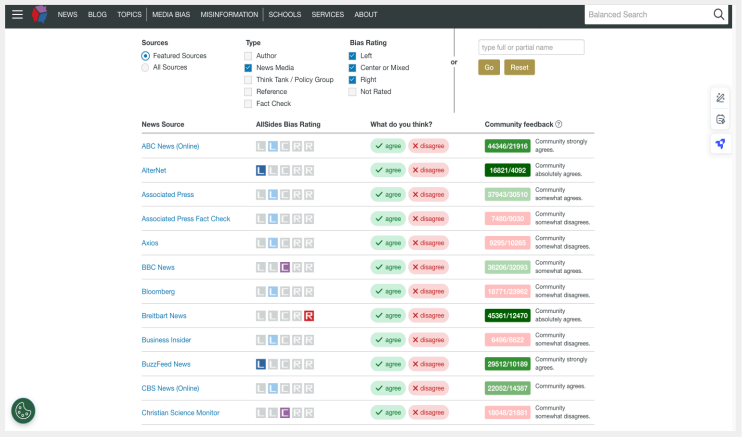
Methodology

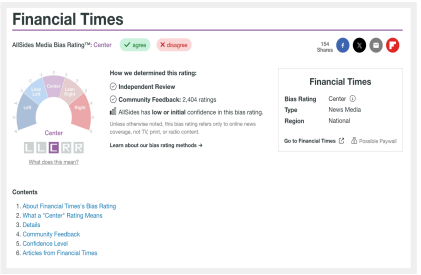
AllSides

Don't be fooled by media bias & misinformation.

In order to answer this question, we leveraged a website called `AllSides.com` which aims to mitigate media bias and misinformation by presenting information and ideas from all sides of the political spectrum.

It uses a tool called `Allsides Media Bias Rating` to rate the Political stance ranging from `Left to Right`, of all the major News sources such as `ABC News`, `BBC News`, `Bloomberg`, and so on.





Given the scope of our project, we utilised the `Balanced Search tool` on the website to collect articles from various news outlets covering specific topics. Initially, we focused on topics such as `Defense, Security, and Foreign Policy; Energy, Sustainability, and Environment; Gun Control, Gun Rights, and Violence in America; Healthcare and Public Health`, among others. Our choice of these topics was driven by our personal interests and the anticipated polarisation surrounding each subject. For example, we chose `Energy and Sustainability`

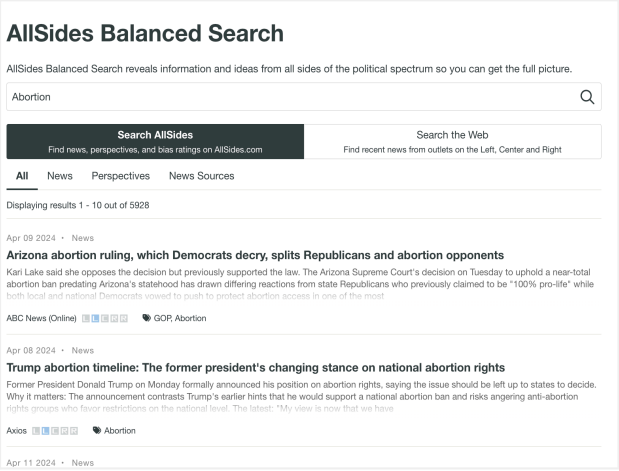
because it is not only a compelling topic but also one where we expect news outlets with differing political orientations to exhibit distinct sentiments in their coverage.

However, due to time constraints and a limited number of articles available on certain topics, we had to narrow down our list of topics to the following:

1. `Environment and Sustainability (combined Topics - Sustainability and Environment)`
2. `Public Healthcare (combined Topics - Healthcare and Public Health)`
3. `Abortion`
4. `Immigration`

Dataset Creation

The `Balanced Search tool` on AllSides generated a list of articles sorted by dates across multiple pages, all relevant to our chosen topic. Each article entry included the headline, the news channel that reported the article, political bias associated with that news channel, tags associated with that article, a URL to the article on AllSides, and a URL to the original article on the news channel's website. This detailed level of information was instrumental in our decision to use AllSides as the primary source for our data collection.



For example, when searching for "`Abortion`", the tool would display all articles related to the topic. These articles were listed across multiple pages, enabling us to scrape content from any specified number of pages. In this particular search, there were `593` pages available. Our dataset creation script was designed to allow users to specify how many pages they wanted to retrieve, facilitating targeted and efficient data collection.

Web Scraping

We implemented a web scraping script to collect news articles from the AllSides website using the URL format: `https://www.allsides.com/search?search={topic}&item_bundle=1&sort_by=node_created`. Here, `{topic}` is a placeholder where users can enter a specific news category (for instance, "`abortion`"). This script also provides users the flexibility to choose both the topic and the number of pages they want to scrape for each search.

To execute the scraping process, users can specify the topic and the desired number of pages, then run the script using the command `python news_scraping.py <topic>`. This action triggers the script to scrape data according to the specified parameters and

subsequently generates a CSV file named `<topic>.csv`. The resulting CSV file extracts the following columns: `Published Date, Title, AllSides Content, AllSides URL, News Channel, News Channel URL, Bias, Tags`

Feature Engineering

To enhance the comprehensiveness of our dataset after gathering all news articles and their relevant information from `AllSides`, we decided to expand it by adding additional columns. This expansion was necessary to align the dataset more closely with the specific requirements for answering the questions posed at the beginning of the project. It was accomplished using the following series of scripts:

news_sources.py

This script utilises the `newspaper3k` package to enhance our dataset by adding a "`News Channel Content`" column. The package allows the creation of an Article Object, which simplifies content retrieval. By passing the actual `News Channel URL` into the Article Object, the script automatically fetches the full text from the corresponding news channel's website. This eliminates the need for manual scraping of each news article's content. To run the script, users can use the command `python news_sources.py <topic>.csv <topic>_news.csv`. This command processes `<topic>.csv` and outputs a new file, `<topic>_news.csv`, which includes all previous data plus the newly added column containing the full text of the articles.

data_cleaning.ipynb

After executing the `news_sources.py` script separately for each of the six topics — "`abortion,`" "`immigration,`" "`sustainability,`" "`environment,`" "`public health,`" and "`healthcare`" — to add "`News Channel Content`" column to the respective datasets, this Jupyter notebook then consolidates these individual CSV files into a single dataset and extract the csv file - `news_data_all.csv`. The notebook also explores the 3 different hugging face Text Summarization models on the `News Channel Content` for further exploration.

summary.py

After testing the different summarization models from Hugging Face in `data_cleaning.ipynb`, this script loads all the models - `facebook/bart-large-cnn`, `sshleifer/distilbart-cnn-12-6`, and `Falconsai/text_summarization` and then processes the "`News Channel Content`" to produce 3 distinct summaries. As a result, it adds 3 new columns - `Distil_Summary`", "`Falcons_Summary`", and "`Bart_Summary`" to the final dataset. After executing the script, a csv file called `final_summ.csv` is obtained.

Data Analysis

Once our complete dataset was compiled, we conducted sentiment analysis on the "`News Channel Content`", “`Distil_Summary`”, ”`Falcons_Summary`”, and ”`Bart_Summary`”. After this, our project branched out into two different sections:

1. **`Implemented Classification models`** to assess whether the models could accurately predict the Topic and Political Bias based on the article content and the associated sentiment scores.
2. **`Performed exploratory data analysis`** to evaluate and categorise the range of sentiments expressed, reflecting the diverse political viewpoints represented in the news articles.

Sentiment Analysis

labeler.py

This script calculates the sentiment scores of the following columns - "`News Channel Content`", “`Distil_Summary`”, ”`Falcons_Summary`”, and ”`Bart_Summary`”, using: `nltk.sentiment.vader.SentimentIntensityAnalyzer()` and a slightly modified version of `sentiment_pipeline_model_cardiffnlp`. After executing the script, a csv file called `labeled.csv` is obtained (which is used in both the sections as follows).

Section 1: Model Training - Predicting Bias and Topic

model.ipynb

After obtaining sentiment scores for the article content and its summaries in our dataset, we proceeded to develop classification models (`XGBoostClassifier` and `LogisticRegression`). Our goal was to assess whether using the original article content or any of its three associated summaries would affect the models' performance in predicting two attributes: `Topic` and `Bias`.

Input Features

To evaluate the performance based on the actual articles and their summaries, we organised the final dataset into four distinct subsets:

1. **Article Content Data** - comprises the full text of the `news articles` along with its corresponding sentiment scores from both the `nltk's SentimentIntensityAnalyzer` and the modified `cardiffnlp` sentiment analysis model.
2. **Distil Summary Data** - comprises summaries of news articles created using the `sshleifer/distilbart-cnn-12-6` model, accompanied by its corresponding sentiment scores from both the `nltk's SentimentIntensityAnalyzer` and the modified `cardiffnlp` sentiment analysis model.

3. **Bart Summary Data** - comprises summaries of news articles created using the `facebook/bart-large-cnn` model, accompanied by its corresponding sentiment scores from both the `nltk's SentimentIntensityAnalyzer` and the modified `cardiffnlp` sentiment analysis model.
4. **Falcon Summary Data** - comprises summaries of news articles created using the `Falconsai/text_summarization` model, accompanied by its corresponding sentiment scores from both the `nltk's SentimentIntensityAnalyzer` and the modified `cardiffnlp` sentiment analysis model.

All the datasets also contain the assigned target labels for `Bias` and `Topic`

Feature Engineering

For all the datasets, we utilized the `SentenceTransformer model('paraphrase-MiniLM-L6-v2')` to create vector embeddings of the text input features. These embeddings were then combined with the two sentiment scores (NLTK and Cadiffnlp) derived from our sentiment analysis. This approach of merging textual embeddings with sentiment metrics facilitated our predictive modelling capabilities.

Output Labels

For each dataset, we trained 2 separate pairs of classification models (`XGBoost Classifier and Logistic Regression`) to predict `'Bias'` and `'Topic'`. The `'Bias'` model categorises entries as `Center, Left, or Right`, while the `'Topic'` model classifies content into one of four categories: `Immigration, Abortion, Public Healthcare, Environment and Sustainability`.

To prepare for these classification tasks, the predicted features—`'Bias'` and `'Topic'`—were transformed into numerical labels using the `LabelEncoder()` function. This encoding step allowed us to convert categorical text data into a format that could be effectively processed for fitting the machine learning models.

Results - Topic Prediction

	Article Content Data	Distil Summary Data	Bart Summary Data	Falcon Summary Data
XGBoostClassifier	79.37%	71.75%	73.97%	75.87%
Logistic Regression	77.78%	71.11%	71.11%	69.84%

Conclusion: Based on this table, it is clear that utilising the full `news articles` leads to improved outcomes for both classification models when it comes to predicting the `Topic` Label. However, the `XGBoost Classifier Model` generally demonstrates better performance in predicting the Topic Labels when compared to `Logistic Regression`.

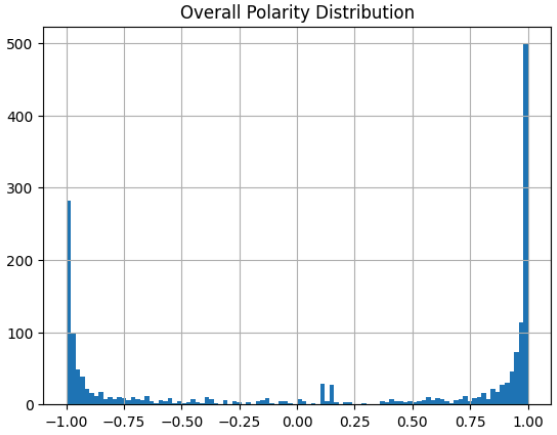
Results - Bias Prediction

	Article Content Data	Distil Summary Data	Bart Summary Data	Falcon Summary Data
XGBoostClassifier	65.08%	58.41%	66.03%	61.27%
Logistic Regression	55.87%	49.84%	52.70%	47.30%

Conclusion: This table indicates that the `XGBoost Classifier Model` yields the most accurate bias predictions when applied to article content summarised by `facebook/bart-large-cnn`. Generally, `XGBoost Classifier` outperforms `Logistic Regression` in predicting biases but the overall accuracy scores for all the models seem to be relatively on the lower end.

Section 2: Exploratory Data Analysis on Bias Distribution for different topics

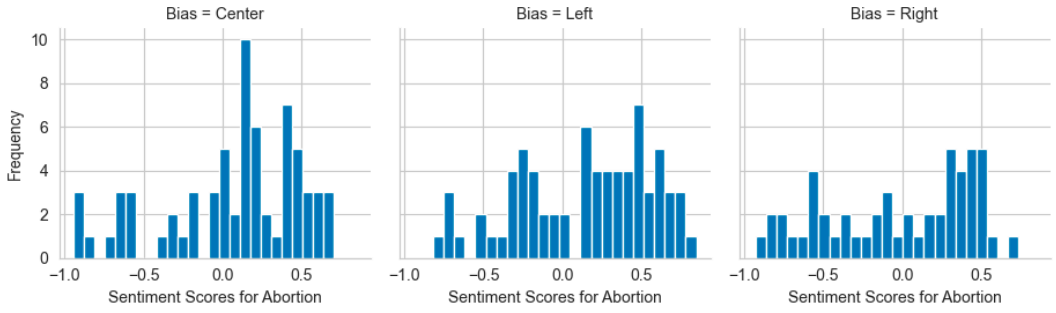
Overall polarity scores:

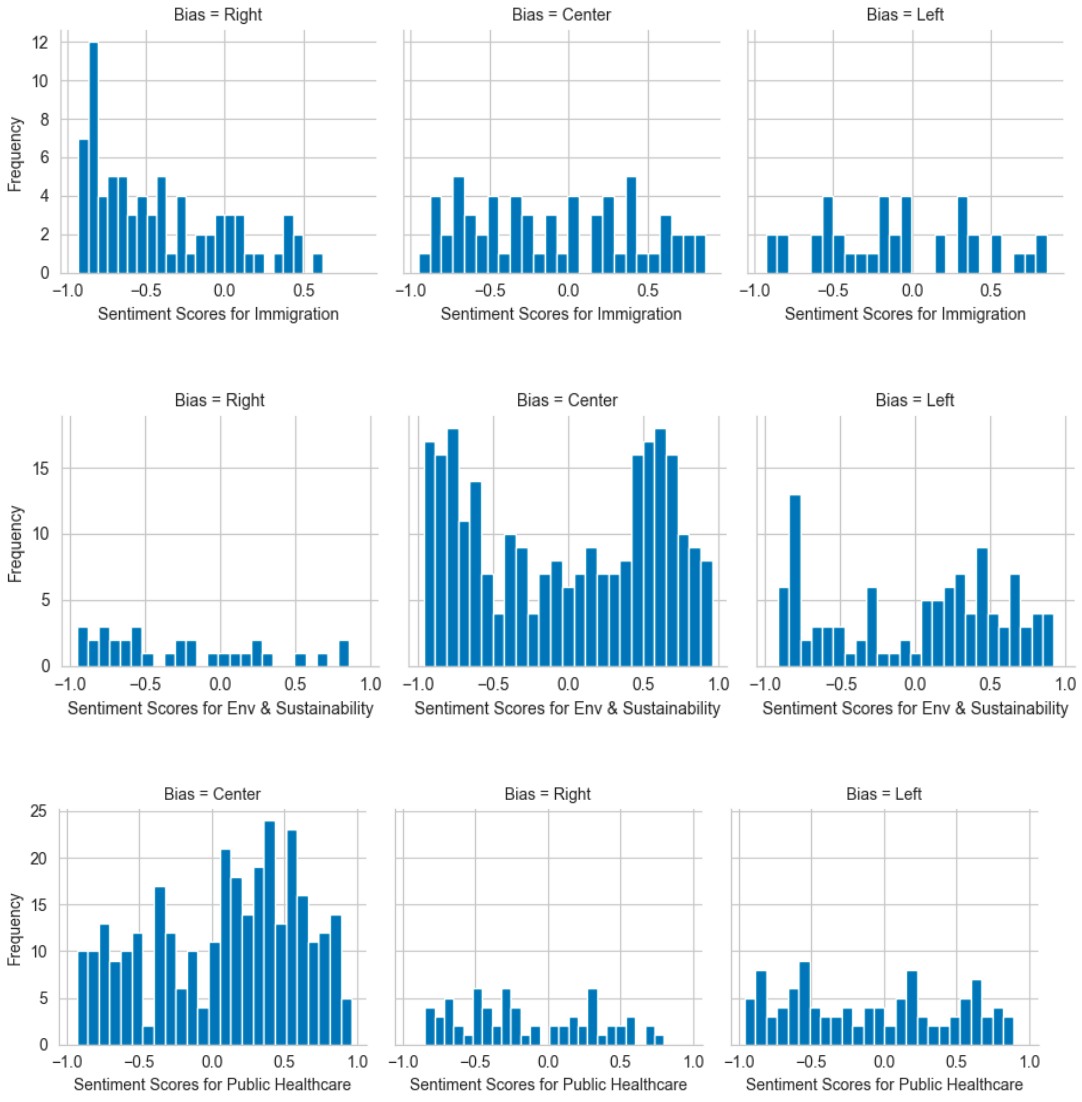


Overall sentiment was distributed near the edges. It appears that news articles rarely have neutral sentiments, most sway the reading to feeling either positively or negatively about a topic.

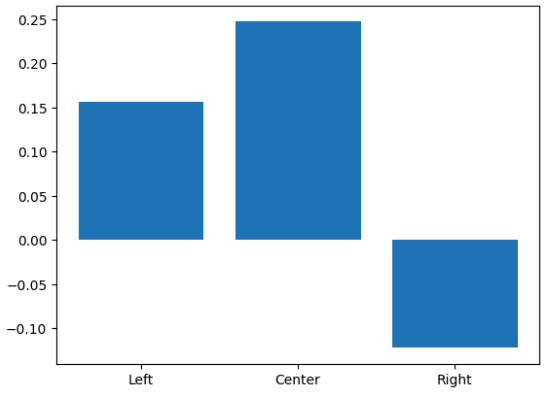
Sentiment Distribution based on average NLTK Scores

The following graphs provide the distribution of the average sentiment scores calculated by `NLTK` (more visualisations in results.ipynb) and categorised based on our `4` topics - `Abortion, Immigration, Environment and Sustainability, and Public Healthcare`

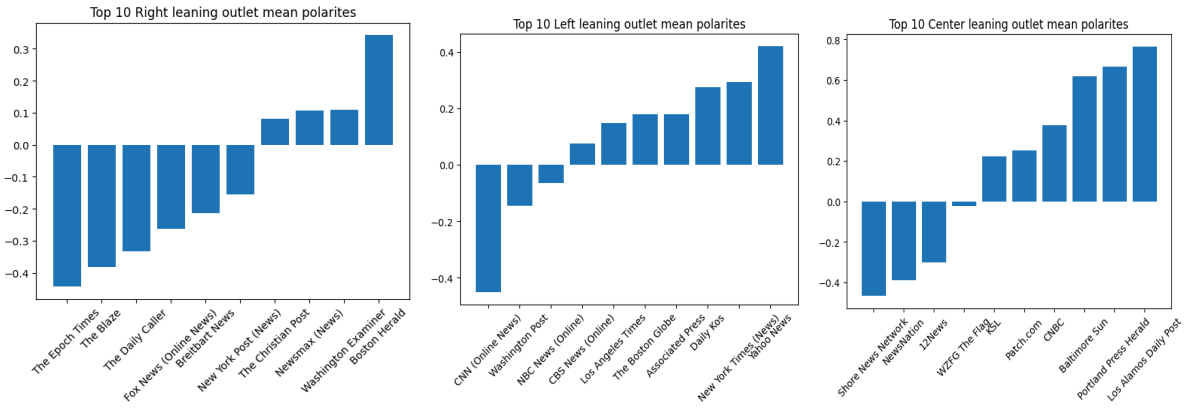




Mean Sentiment for top 10 news outlets:



This was the average polarity score aggregated on the political bias. While the magnitude of this score is low we can see the data weakly suggests central aligning news articles are more likely to introduce positive sentiment into their news articles, along with left leaning outlets to a weaker extent. Average right polarity is negative which suggests a right leaning news article tends to be written more negatively. Of course this is aggregated on many news outlets, so this is susceptible to outliers on any side.



Only 3 or 4 of the top news outlets for centre and left aligning news outlets had a mean sentiment of less than 0. The highest sentiment scoring, centre aligning news outlets, almost reached a score of 0.8. While the right side news outlets are more negative overall, with even the higher sentiment news articles scoring less sentiment than seen on the centre or left aligning news outlets.

From here draw the conclusions that left leaning and centre news articles are dominated by more news stations that tend to write high sentiment scored articles. The top right aligning news is more negative, but more evenly distributed than the left or centre aligning news outlets.

Limitations

Dataset Duplication

In our initial plan, we aimed to construct a large dataset of `80,000` rows by merging all the topic-related datasets from `Allsides.com` in `data_cleaning.ipynb`. However, due to the extensive time required to run summarizer models on the article content, we adjusted our approach to manage processing times. We set a data cap at `4,000` rows for topics like `abortion` and `immigration`, and `2,000` rows for `public health`, `healthcare`, `sustainability`, and `environment`.

During the later stages of our data analysis, we ended up discovering numerous duplicates within our dataset. Now, by this time, we had already compiled `test.csv`, which included all sentiment scores for duplicate entries. So, after identifying this issue, we created a more refined dataset, `labeled.csv`, which only contains non-duplicated data. As a result of this, however, the total number of rows in our data reduced drastically, leading to significant shifts in our model's performance and the outcomes of our analyses.

For example, when using `test.csv` for predictions, our model showed very high accuracy, likely inflated by the duplicate rows. However, the model's performance noticeably changed when we switched to using the non-duplicated `labeled.csv` for training. Despite these shifts, the overall trend in our comparisons remained consistent: the `XGBoostClassifier` consistently outperformed `Logistic Regression` when predicting both - Topic and Label.

Topic Prediction using test.csv

	Article Content Data	Distil Summary Data	Bart Summary Data	Falcon Summary Data
XGBoostClassifier	97.19%	96.27%	96.00%	95.79%
Logistic Regression	96.60%	95.95%	95.73%	95.79%

↓

Topic Prediction using labeled.csv

	Article Content Data	Distil Summary Data	Bart Summary Data	Falcon Summary Data
XGBoostClassifier	79.37%	71.75%	73.97%	75.87%
Logistic Regression	77.78%	71.11%	71.11%	69.84%

Bias Prediction using labeled.csv

	Article Content Data	Distil Summary Data	Bart Summary Data	Falcon Summary Data
XGBoostClassifier	98.70%	98.22%	98.76%	99.03%
Logistic Regression	95.46%	93.74%	96.17%	91.52%

↓

Bias Prediction using labeled.csv

	Article Content Data	Distil Summary Data	Bart Summary Data	Falcon Summary Data
XGBoostClassifier	65.08%	58.41%	66.03%	61.27%
Logistic Regression	55.87%	49.84%	52.70%	47.30%

If we had some more time, we would probably do the following:

- Identify the issues with dataset duplication
- Explore better summarization models
- Explore better methods to produce vector embeddings of the text input features
- Produce better and more elaborate visualisations

Accomplishment Statements

Archita

- **`Implemented Web Scraping System:`** Developed and executed a robust web scraping script (`news_scraping.py`) that effectively collected detailed information on news articles from `AllSides.com` across multiple pages, enabling topic-specific searches and pagination, successfully handling multiple pages per topic, and

capturing essential details like publication date, headline, political bias, and source URLs.

- **Enhanced Data Enrichment Techniques:** Utilised the `newspaper3k` package to automatically extract and append the full content of news articles from their original sources, significantly reducing manual data entry.
- **Streamlined Data Integration and Cleaning:** Orchestrated a comprehensive data cleaning process using `data_cleaning.ipynb`, which consolidated data from six different topics into a single, clean dataset (`news_data_all.csv`).
- **Automated Content Summarization:** Initiated the development of a summarization tool (summary.py) employing state-of-the-art NLP models from Hugging Face to generate concise summaries of extensive news content.
- **Implemented Machine Learning Classification Models:** Trained a logistic regression and an XGBoost Classifier model aimed at predicting bias and topic given the article content and its associated sentiment scores.
- **Identified Project Limitations**
- **Data Analysis:** Performed data analysis on the labelled dataset to attain the sentiment distribution for different topics

George

- **Preprocessed text data:** Cleaned up and transformed data. That includes removing stop words, tokenizing text and lemmatization. This resulted in clean data ready for the nltk model to use.
- **Model exploration:** Surveyed many sentimental analysis models for their suitability for our use. Models were selected based on their compatibility of output and feasibility of implementation.
- **Implemented sentimental analysis labeler:** Developed a program that takes in a dataset as input and outputs the data with a polarity score from -1 to 1 as a column. This program uses a few different models.
- **Data Analysis:** Performed data analysis on the labelled data. Drew conclusions based on patterns seen on the data visualisations.