BA820 A1 - **Group 03** - Professor Mohannad Elhamod

Group Members: **Pengru Lin, Mauro Wang, Lyushen Song, Ashley Mercado, Leonardo Trucios**

Date: March 2nd, 2024

Github Link: <https://github.com/mwangcy/BA820>

**Final Project Deliverable**

**Problem Statement and Motivation:**

In today’s world, we learn and hear what’s happening through online media, and news reports, such as CNN. While these events are highly factual, it is not uncommon to tell that reports and social media purposefully tweak the wording of the news to attract readers or to serve political influences. The goal of this project is to practice our ability to analyze these platforms using Natural Language Processing to understand if the news report platform stays true to be informative, or influences readers towards other directions.

**Dataset:**

Our group will utilize Kaggle as the primary data source (Link: [CNN Articles, Raw data (kaggle.com)](https://www.kaggle.com/datasets/hadasu92/cnn-articles-after-basic-cleaning)). The dataset's information originates from the CNN Site Map. However, data extraction was conducted through a specific web crawling process targeting all articles published from 2011 to 2022. This web crawler was tailored to extract only CNN SiteMap articles via URLs provided from the main file. It goes across these URLs to a specified depth, extracting detailed information for each article. This dataset comprises 37,949 rows and 10 columns, containing specific information about articles such as author, dates, section, URL, among others.

**Proposed Methodology:**

We tokenized the text to break it down into manageable units for analysis. Additionally, we will remove stop words, as they do not contribute meaningful information to our analysis. Further data cleaning will involve stemming to improve the performance of sentiment analysis. Although we utilized a Bag of Words approach to train a sentiment prediction model, it would be time-consuming as it necessitates manual data labeling, which we lack without a target variable. Moreover, the package we intended to use for analysis did not require a Bag of Words approach for predictions. With the data cleaned, we utilized the **NLTK** sentiment analyzer to analyze the sentiment of our text. We maintained separate columns for the main column, headlines, descriptions, keywords, and second headlines to identify inconsistencies in the prediction model, subsequently averaging their scores into a new column. With these steps completed, we then conducted a market basket analysis to determine if certain article sections tend to exhibit higher-than-average positive sentiment or plot them on a timeline to gauge the general sentiment of CNN over the years.

**Project Execution**

**Cleaning**

The raw dataset contained 37,949 entries with 10 columns, with each row representing a unique article. Irrelevant columns like “Index” and “URL” were dropped upon loading. The analysis focused on relevant columns, particularly “Author”, where entries with “By” or blank were replaced with “Anonymous” for consistency since we didn’t have information on them. Missing values were addressed, with only 9 instances found in the 'Article text' column, representing 0.02% of the dataset, and thus were dropped. Due to computational constraints, undersampling techniques were employed for the analysis, conducted on a 20% data subset.

**Pre-Processing**

After generating the sample, we opted to standardize the text by converting all strings to lowercase and eliminating trailing white spaces across all text columns. Following this, we converted the “Data published” column into a “datetime” data type for easier manipulation. Subsequently, in our string cleaning process, we excluded all numbers and punctuation marks except for “!” and “?”, recognizing their importance for sentiment analysis. Numbers and most punctuation marks were deemed irrelevant for sentiment analysis and could introduce unwanted noise. Lastly, to enhance tokenization for Bag of Words (BoW) and TF-IDF approaches, we introduced a new function incorporating stemming, aiming to minimize column count and facilitate sentence comparisons. We also set a minimum frequency for these models to weed out outliers and errors in the corpus. We opted for TF-IDF over Bag-of-Words (BoW) because TF-IDF assigns higher weights to words that are less frequent in the entire document collection but appear frequently within a specific corpus. This was particularly beneficial for tasks like sentiment analysis, where distinguishing the sentiment-carrying words is crucial.

**Analytical Roadmap:**

We utilized the NLTK sentiment analyzer (SentimentIntensityAnalyzer) and Vector Comparison with cosine similarity to analyze the sentiment of our corpus. SentimentIntensityAnalyzer is a pre-built sentiment analysis tool and works well for analyzing the overall sentiment of the text. On the other hand, Vector Comparison captures sentiment based on word relationships without training a model, ideal for initial exploration and unsupervised learning scenarios. To identify potential inconsistencies, sentiment scores were maintained for individual text sections (headlines, descriptions, keywords, and second headlines) before being averaged. Afterward, we used TextBlob to obtain sentiment with another sentiment analysis tool to obtain a broader prediction. With these steps completed, we conducted a market basket analysis to determine if certain article sections tend to exhibit higher-than-average positive sentiment or plot them on a timeline to gauge the general sentiment of CNN over the years (Appendix Chart 1).

**Results:**

We performed sentiment analysis using different methodologies. Firstly, we analyze with NLTK's SentimentIntensityAnalyzer, which is applied to all five columns. The results are then averaged to reduce noise. We then use the SentimentIntensityAnalyzer to identify very positive, negative, and neutral sentences. This approach enables us to conduct vector comparison by harnessing Word2Vec and applying cosine similarity. In doing so, we gain an unsupervised understanding of the sentiment expressed in each row. Afterwards, we use TextBlob to obtain sentiment with another sentiment analysis tool to obtain a broader prediction. Given the three predictions, we can identify instances where the predictions are inconsistent. We then take the mode for these predictions and set 0 when all predictions are different. Finally, we can perform market basket analysis using the Author, category, and section data to discover an interesting pattern

The preliminary results from the second deliverable shown in Appendix chart 5 revealed a skewed negative sentiment in the articles published by CNN. However, with the introduction of neutral sentiment, we have seen improvement. We observed that neutral sentiment predominates, which is not surprising, considering that news outlets strive to be factual and unbiased when sharing news, meaning that CNN is doing a great job at being an unbiased news source (Appendix Chart 2). From Word2Vec, the results showed a very similar sentiment (Appendix Chart 3). Furthermore, we have deployed one more tool for sentiment analysis, known as TextBlob (Appendix Chart 4). We decided to use voting to average the predictions and find the mode among the three predicted sentiments. Lastly, we have utilized t-SNE and PCA for visualization to better understand whether the labels are clustered together.

In conclusion, after our thorough examination of the sentiment analysis, we confirmed that most of the news on CNN is not positively or negatively biased, but tends to be neutral on average. In t-SNE, we can see that the few instances of positive and negative sentiment appear near the border of the sphere we obtained, whereas for PCA, we see that they are concentrated towards the center of the graph. This observation suggests that the labeling may exhibit signs of utilizing actual patterns to predict sentiment (Appendix Chart 6 and 7).

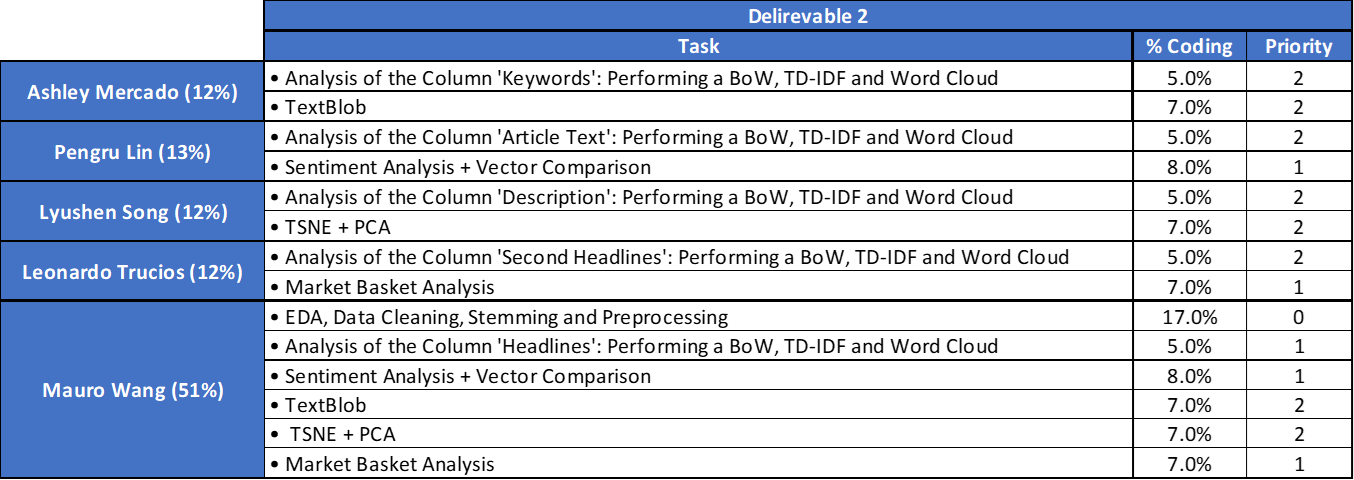
**Challenges:**

While our team was able to navigate through every challenge we encountered during the analysis, some of these challenges were worth documenting. One challenge we encountered involved encountering blank or unnamed entries in key columns, particularly in the “Author” column. To maintain consistency, we opted to substitute these instances with the string “Anonymous”. We also observed 9 missing values in the “Article text” column, and we were able to drop them as they only constituted 0.02% of the dataset. Ultimately, we had issues with computational constraints, hence we decided to use undersampling techniques. We conducted the undersampling on only 20% of the data subset, which enabled us to obtain a reasonably sized sample while mitigating constraints and conserving time.

**Business Applications:**

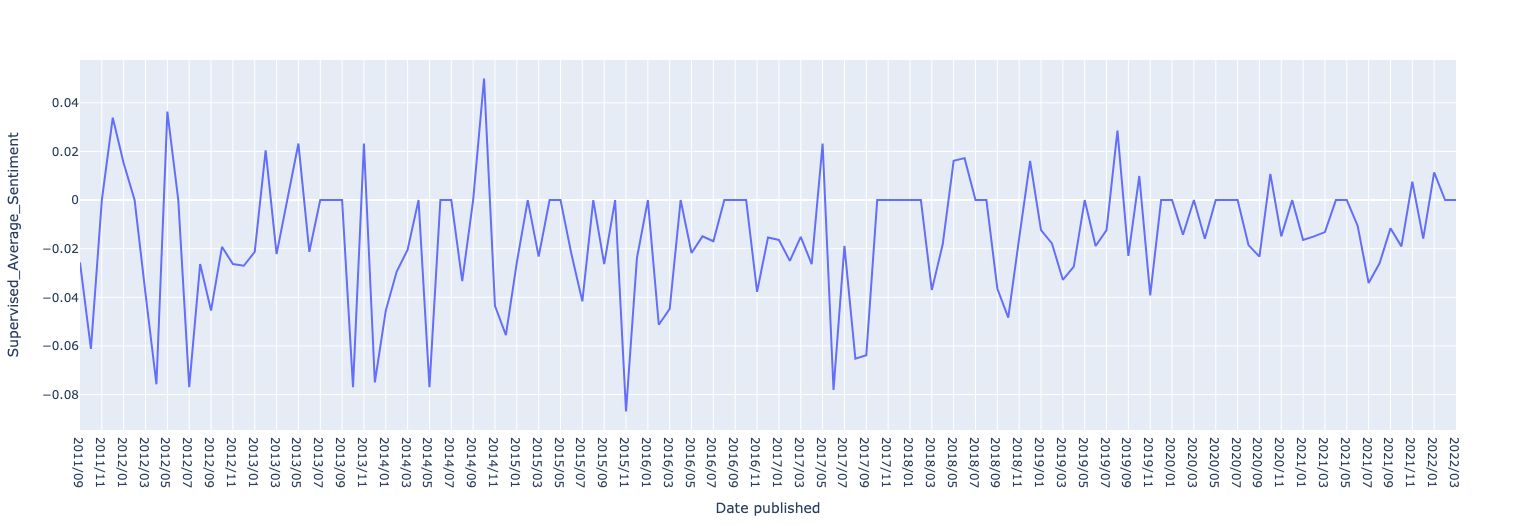
Understanding the tone and sentiment of news media companies like CNN provides valuable insights for optimizing content delivery. This enables news providers to maintain factual and neutral reporting, fostering trust with their audience. By impartially analyzing sentiment, biases can be minimized, resulting in a more balanced portrayal of news. Additionally, extending sentiment analysis into financial applications allows investors to gauge market sentiment towards specific entities, aiding in informed investment decisions. Furthermore, sentiment analysis finds application in politics, enabling voters to understand if newspapers show unbiased sentiment.

**Coding Contribution:** <https://github.com/users/mwangcy/projects/1/views/1>

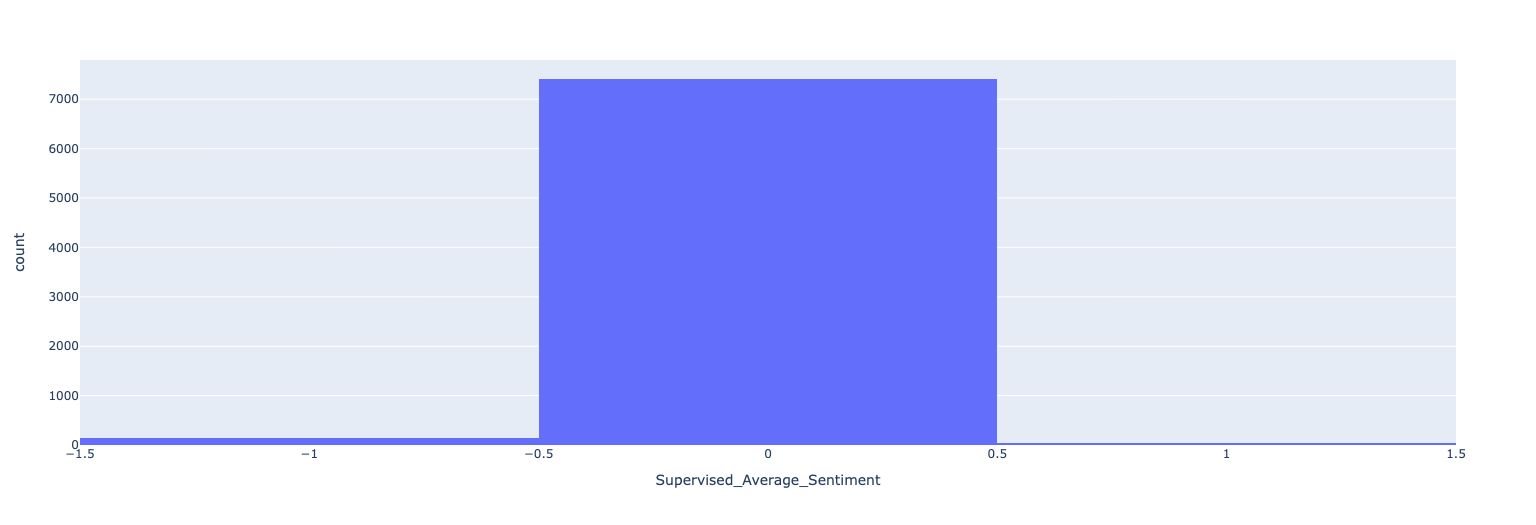
****

**Appendix**

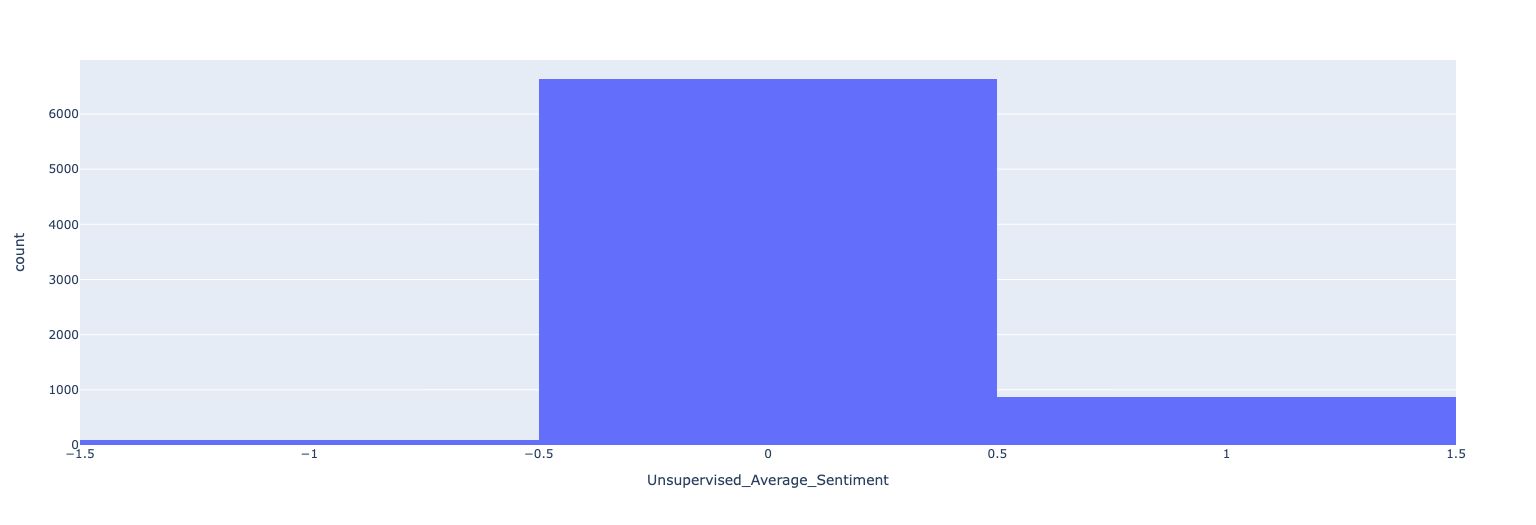
**Chart 1:** Sentiment Analysis Timeline - SentimentIntensityAnalyzer

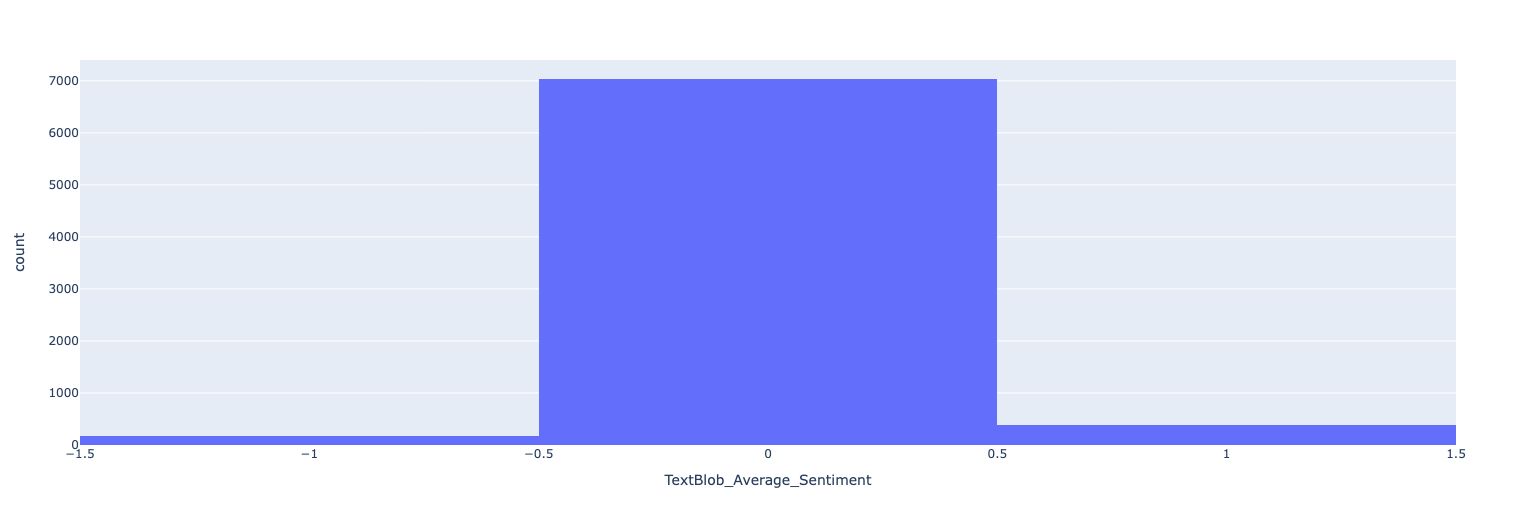


**Chart 2:**  Sentiment Analysis Results - SentimentIntensityAnalyzer Algorithm

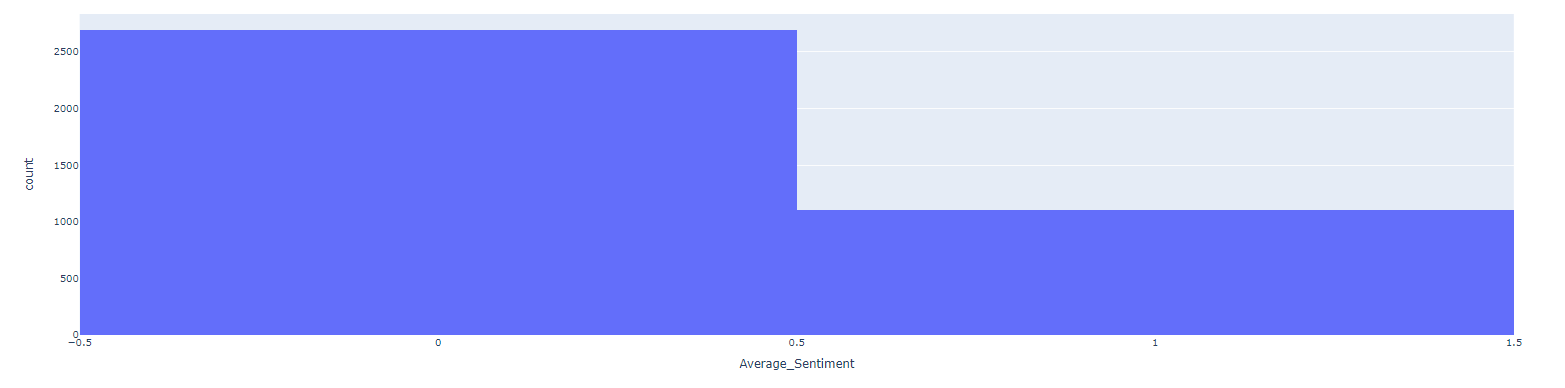


**Chart 3:** Sentiment Analysis Results - Word2vec Algorithm

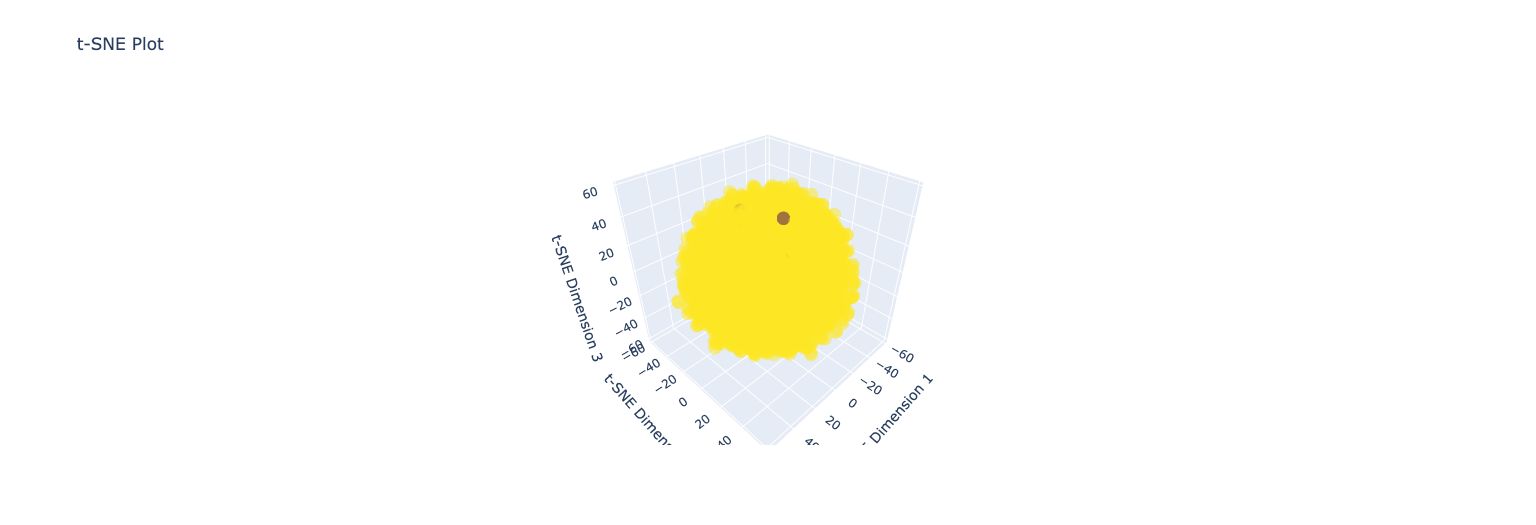


**Chart 4:** Sentiment Analysis Results - Text Blo

**Chart 5:** Preliminary Results



**Chart 6:** t-SNE Analysis



**Chart 7:** PCA Analysis