# SIADS 696: Milestone II Project Report
# Racial Bias Detection in News Articles

Amanda Fear (amafear@umich.edu), Ejaz Alam (ejazalam@umich.edu), Nikolay Jamgaryan (nikolayj@umich.edu)

## Introduction:

News media plays a pivotal role in shaping public opinion and perceptions of societal issues, including race. How news articles discuss and portray racial dynamics has deep implications for how different racial groups are viewed and treated within society. As information dissemination accelerates, the need to identify and address racial bias in news reporting becomes increasingly urgent. This project employs Natural Language Processing (NLP) techniques to detect and quantify racial bias in news articles, with the aim of contributing to fair and inclusive reporting practices.

This project utilizes both supervised and unsupervised learning methods to analyze news articles. Supervised learning approaches, including BERT (Bidirectional Encoder Representations from Transformers), Long Short-Term Memory (LSTM), Support Vector Machines (SVM), and Stochastic Gradient Descent (SGD) with Logistic Regression, are applied to classify articles based on the presence and degree of racial bias. BERT, a cutting-edge deep learning model, is particularly adept at NLP tasks, while SVM and SGD serve as classic machine learning benchmarks for text classification.

In the unsupervised learning portion, Latent Dirichlet Allocation (LDA) with TF-IDF is used for topic modeling to uncover latent themes in the text data.  Additionally, K-means clustering with word embeddings is employed to categorize similar articles, potentially revealing patterns indicative of racial bias.

The project's main findings include the superior performance of the BERT model in supervised learning, achieving an accuracy score of 85.8% and an F1 score of 0.84. In unsupervised learning, the LDA model identified 5 distinct topics related to racial discussions in news articles, and the K-means model determined that 17 clusters best represented the data.

## Related Work:

Our project draws inspiration from existing studies that leverage machine learning for bias detection, yet it differs by focusing on news articles and employing a broader range of techniques.  The first study by Mozafari et al., titled **"Hate Speech Detection and Racial Bias Mitigation in Social Media Based on BERT Model"**, utilizes BERT for identifying hate speech and racial bias on Twitter[13].  Our project innovates by shifting the focus to news articles, presenting distinct linguistic and contextual challenges, and by exploring additional classification methods beyond BERT.  Also, Mozafari's work primarily involves statistical models and query optimization. In contrast, our project aims to use both supervised and unsupervised learning approaches to derive metrics specific to bias detection (Precision, Recall, F1-score, etc.).

Gupta et al.'s **"PoliBERT: Classifying political social media messages with BERT"** employs BERT to categorize political content on social media, demonstrating its superiority over traditional methods[6]. Our project diverges by concentrating on the detection of racial bias within news articles, a domain that requires a nuanced understanding of journalistic language and conventions.

The third study, **"Measuring racial bias within the Dutch public news outlet's coverage"**, by de Boer, examines the portrayal of non-Western ethnic groups in Dutch news media using sentiment analysis and word embeddings[5]. Our project builds on these techniques, applying them to English-language news sources and incorporating the BERT model for enhanced analysis.

## Data Source:

For this project, our primary data source was the **“Navigating News Narratives: A Media Bias Analysis Dataset"** [14]. The dataset combines news data from various reputable sources covering a broad range of news topics and biases such as hate speech, toxicity, sexism, ageism, gender, racism, and more.

**Location and Format:** The dataset can be downloaded from Zenodo.org in CSV format.

**Important Variables:**

- **text**: The main content of the news article.
- **dimension**: Descriptive category of the text.
- **biased_words**: A compilation of words regarded as biased.
- **aspect**: Specific sub-topic within the main content.
- **label**: Indicates the level of bias (highly biased, slightly biased, neutral).
- **sentiment**: Positive, negative, or neutral sentiment.

**Size and Records:** The dataset contains over 3.7 million rows and 6 columns (960 MB). Since our focus is on racial bias, we selected a subset for our analysis which resulted in 41,769 records.

**Initial Preprocessing:**

1. The dataset was filtered to include only rows where the 'aspect' column is 'Racial'.
2. After removing duplicate records to ensure data quality and avoid redundancy, there were 39,830 unique records.
3. Row-wise deletion of missing values was performed, targeting the ‘text’ column due to its significance in our analysis.

Some exploratory data analysis is shown below:

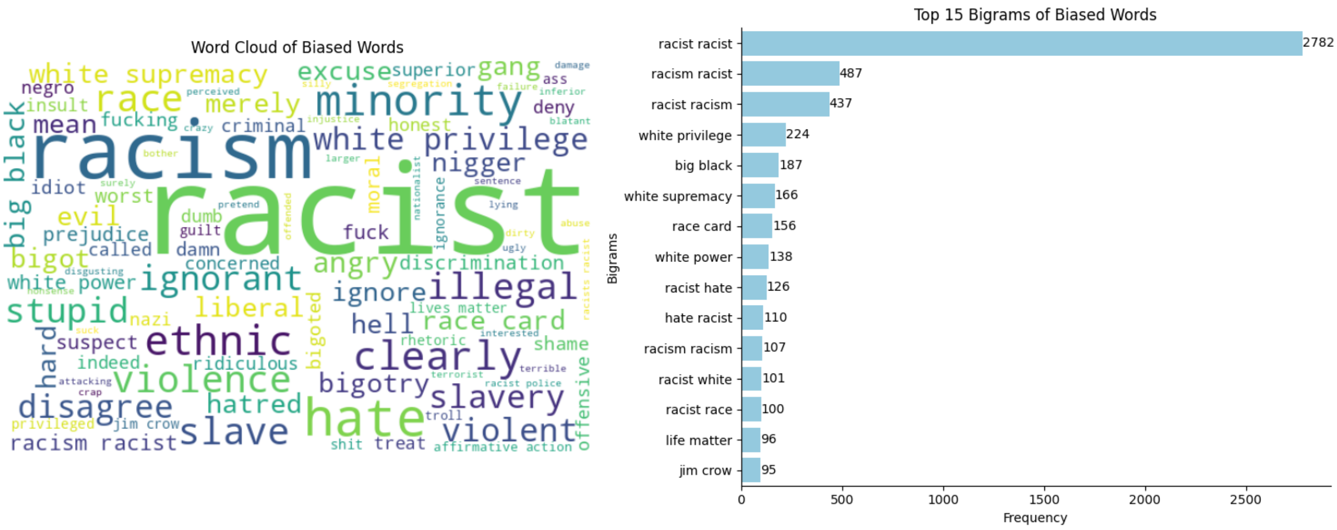


**Figure 1: Word Cloud & Top 15 Bigrams of Biased Words.**

The above figure shows some of the most notable biased words found in the dataset. Examples of some of those words are ‘racist’, ‘racism’, ‘hate’, ‘minority’, ‘ethnic’, ‘illegal’, ‘slave’, ‘white privilege’, ‘white supremacy, and so on.

## Feature Engineering:

The feature engineering pipeline for our project involved several key steps to transform the raw text data into a structured format suitable for machine learning models.  Below is an outline of the major steps undertaken:

1. **Text Preprocessing:** Convert text to lowercase, Remove special characters, Tokenize the text into words, Eliminate stopwords, and Apply lemmatization to reduce words to their base forms.
2. **Categorical Label Encoding:** Encode categorical labels (‘Neutral’,’ Slightly Biased’, ‘Highly Biased’) into numeric values using scikit-learn’s LabelEncoder.
3. **Feature Selection and Transformation:** Retain ‘Text’ after preprocessing for further analysis, and Derive ‘Sentiment’ through sentiment analysis techniques to indicate the sentiment of the text.
4. **Model Specific Feature Engineering:**
    a. The BERT model used the DistilBERT tokenizer to convert preprocessed text into input features.
    b. The SGD, SVM, and LDA models apply TF-IDF using tfidfVectorizer to transform text into numerical feature vectors that reflect the importance of words in the document collection.
    c. The LSTM model tokenized the text and padded the sequences to a maximum of 100 tokens to ensure uniform input sizes.
    d. The K-means clustering trained a Word2Vec model on the preprocessed text to generate word embeddings, then created document vectors by averaging these embeddings for each document.

Throughout this pipeline, the data was systematically cleaned, reduced, and enhanced to facilitate the detection of racial bias in news articles. The final dataset was optimized for both supervised and unsupervised machine learning methods, with a focus on features that would be the most indicative of bias. For a detailed list of features used in our models, please refer to **Appendix C**.

## Supervised Learning:

### Motivation:

This project aims to experiment with state-of-the-art neural networks (BERT, LSTM) and classic machine learning models (SVM, SGD) to compare their quality in predicting racial bias in news text. The goals for this part were to understand the maximum quality that can be reached without using production-scale resources and to deeply analyze the model that will show above-average quality.

We explored four different learning models. After data preprocessing and feature engineering, our workflow included model training, hyperparameter tuning, evaluation, and further detailed analysis on the best model.

### Methods Description:

1. **Linear Support Vector Machine (non-probabilistic, linear):**
   SVM was chosen because of its robustness to high-dimensional data and its effectiveness with sparse data.
2. **Stochastic Gradient Descent (SGD) with Logistic Regression (probabilistic, linear):**
   SGD was chosen because of its ability to adapt to classification tasks with a suitable loss function. It is appropriate for text data due to its efficiency with large datasets.

3. **Long Short-Term Memory (LSTM) neural network (non-linear, sequential):** LSTM was chosen due to its ability to capture long-term dependencies and context within text sequences.
4. **BERT (Bidirectional Encoder Representations from Transformers):** BERT was chosen because it is based on the transformer architecture and was pretrained on a large corpus of text, which allows it to show great results out of the box. BERT's prior training on the masked language modeling task (MLM) and next sentence prediction task (NSP) allows it to capture the nuances of the language.

### Hyperparameter Tuning

We performed hyperparameter tuning to determine the best model from each of the above models explored. We used RandomizedSearchCV for SGD and SVM to explore different hyperparameters, optimizing for the 'f1_macro' score across 5-fold cross-validation.

For SGD, the learning rate ('alpha') and the maximum number of iterations ('max_iter') were the primary parameters that were tuned. For SVM, the regularization parameter ('C') was tuned.

For LSTM, we used a combination of manual method and early stopping due to limited amount of time and computational resources. We manually set values for key parameters such as vocabulary size, embedding dimension, number of LSTM units, and dropout rate and used early stopping to determine the optimal number of training epochs.

All models' quality metrics were calculated using cross-validation with five folds. The only exception to this rule was the quality metrics of the BERT model, which was trained on a sample of 80,000. Because training was taking too long, we used three folds instead of five, and the primary hyperparameters to tune for BERT were the learning rate, batch size, and size of the training sample.

## Supervised Evaluation:

Overall Results

During training and evaluation of the model's quality on the text bias prediction task, we used mean metrics (accuracy, weighted precision, weighted recall, and weighted F1 score) across 5-fold cross-validation along with standard deviations. Since our project deals with multi-class classification problems (Neutral, Slightly Biased, Highly Biased), we used these metrics because we can have a comprehensive view of our model's performance. This helps us understand not only how our model is correct overall but also how well it identifies bias and how reliable it is when it flags content as biased.

We decided to choose the best model based on the weighted F1 score.

The formula for weighted F1 score: $F1_{weighted} = 2 * \frac{Precision_{weighted} * Recall_{weighted}}{Precision_{weighted} + Recall_{weighted}}$

Weighted precision and weighted recall for multiclass classification are calculated for each class separately on a one-vs-all basis and then averaged over all classes with the same weight of $\frac{1}{n}$ for each of $n$ classes. This metric allows us to take into consideration both types of errors for each class (false negatives and false positives) and class imbalance.

The table below shows the performance of each model with the F1 score presented as mean ± standard deviation across 5 folds.

**Table 1: Model Performance (Supervised)**

| | BERT | LSTM | SVM | SGD |
|---|---|---|---|---|
| **F1 Score (mean with s.d.)** | 0.84 ± 0.0154 | 0.674 ± 0.008 | 0.649 ± 0.002 | 0.600 ± 0.003 |

We can observe from the above table that BERT followed by LSTM performed well when compared to classic ML algorithms. In addition, we can also notice low standard deviation for all models which indicates a consistent performance across folds.

Due to time and computational resources limitations, instead of performing deeper analysis on the best model (BERT), the following deeper analyses were performed on other models such as LSTM and SVM.

Feature Importance Analysis

We performed feature importance analysis to analyze the importance of features that contributed the most to our SVM model's performance. Our method involved using permutation importance to assess the impact by shuffling the feature's values randomly and observing its impact on the model's F1 score.

We observed the results for the top ten features and visualized them below.

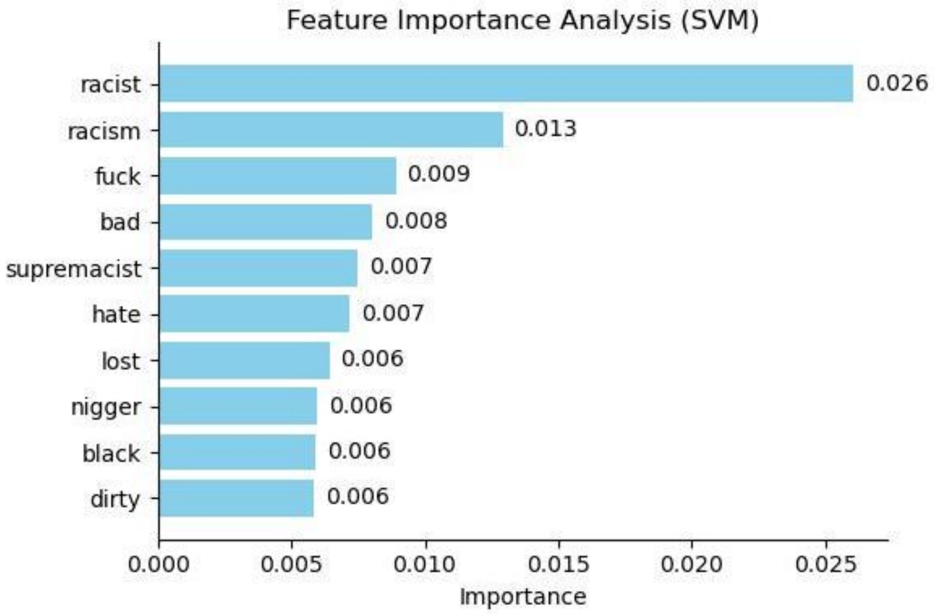


**Figure 2: Feature Importance Analysis (SVM)**

From the above visualization, we can observe some of the most essential words contributing to our model's performance. For example, 'racist' and 'racism' were the top two features that together if they were removed, would cause a combined 0.039 drop (0.026 and 0.013 respectively) in the F1 score. In other words, mentions of 'racist' and 'racism' were pretty strong indicators of bias. In addition, we can also observe mentions of hate speech such as 'fuck', 'hate', 'nigger' and racial slurs that indicate that mentions of these were also key indicators in detecting bias.

On the other hand, one could also argue that such high importance of terms such as 'racist' and 'racism' may be a potential risk of overfitting as the model might struggle to detect bias if the content doesn't have these specific terms.

## Sensitivity Analysis

We performed sensitivity analysis on our LSTM model by focusing on the impact of dropout rates while keeping the other parameters constant. We chose different ranges of dropout rates because it is an important hyperparameter that prevents overfitting and can significantly impact a model's performance.

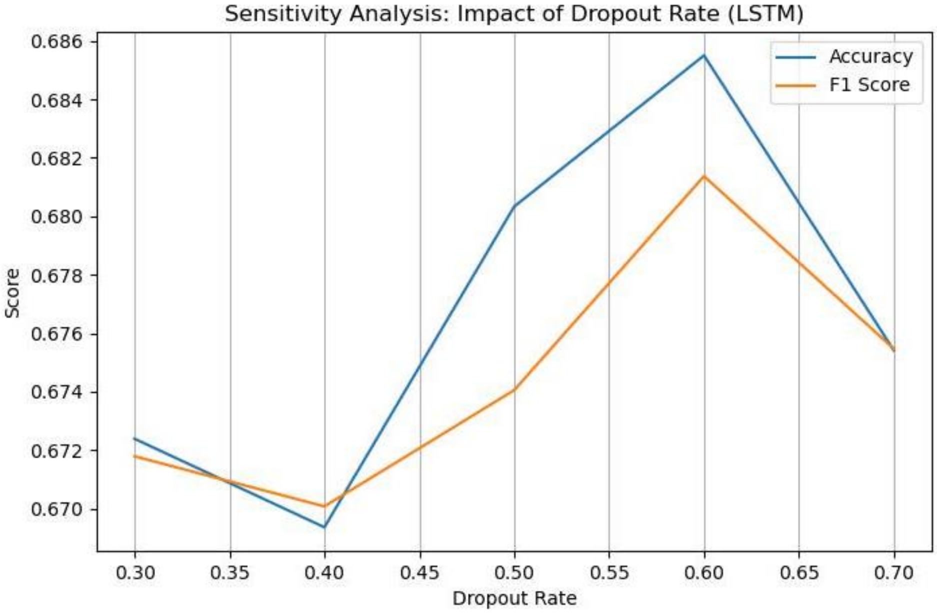


**Figure 3: Sensitivity Analysis: Impact of Dropout Rate (LSTM)**

From the above visualization, we can see that both 'Accuracy' and 'F1 score' are slightly sensitive to changes in the dropout rate at a local optimum. Changing the dropout rate from the local optimum at 0.6 by ± 0.1 gives us an average drop of 0.004 for both metrics. This indicates that the quality of the LSTM model is relatively stable to changes in the dropout rate at the point of 0.6 and likely to generalize well to unseen data.

## Tradeoff Analysis

Given the evaluation results and metrics used, we observed some tradeoffs as mentioned below:

1. The more obvious tradeoff was between a model's complexity and performance. BERT outperformed the other models such as LSTM, SVM, and SGD, but this was more apparent due to higher computational resources and longer training time requirements compared to classic ML algorithms.

2. The second tradeoff was between speed and accuracy, as simpler models such as SGD take little time to train and predict. However, they don't achieve the same accuracy as more advanced models such as BERT and LSTM.

Below is a deeper analysis of two specific tradeoffs between LSTM’s training data size and accuracy and between its training time and accuracy.

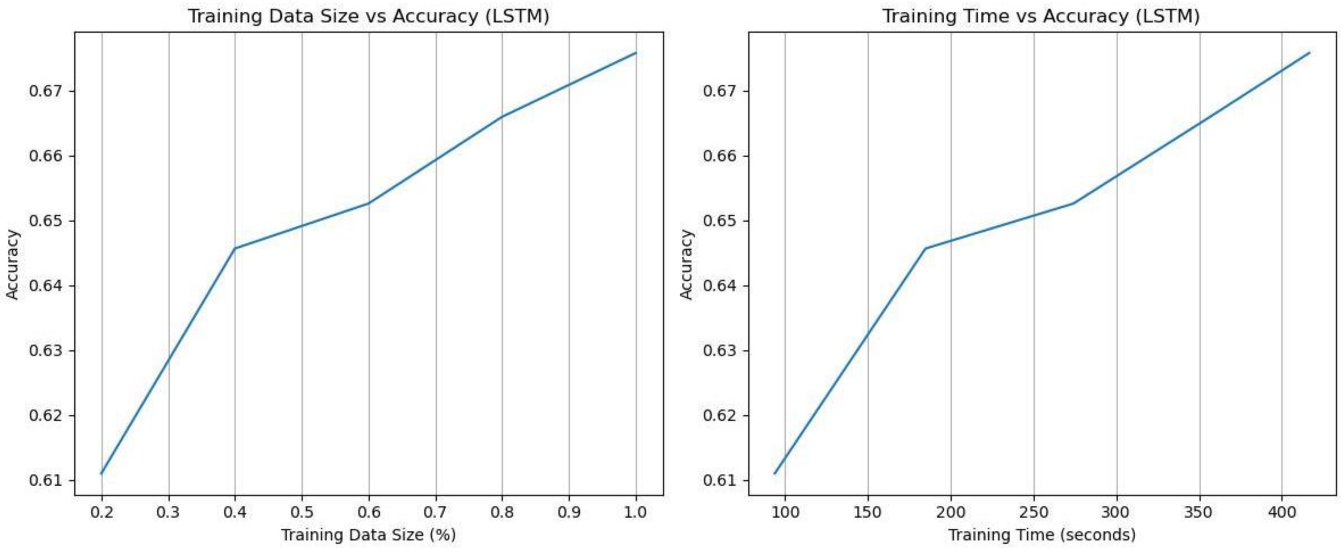


**Figure 4: Training Data Size vs. Accuracy (LSTM) & Training Time vs Accuracy (LSTM)**

From the above visualization, we can observe that both training data size and training time are positively correlated with LSTM's accuracy.

Training Data Size vs Accuracy (LSTM):

In this plot, we can observe a significant increase in the model's accuracy as the training data size increases from 20% to 60%, but the rate of improvement in accuracy slows beyond 60%. At 100%, it peaks at its highest accuracy. This indicates a clear tradeoff where more data is definitely important for a model's training and better accuracy, but the benefit starts decreasing after a certain point.

Training Time vs Accuracy (LSTM):

In this plot, we can see a similar pattern to the training data size. As training time increases, so does the model's accuracy. Initially, the model's accuracy increased significantly but slowed beyond 250 seconds of training and peaked at over 400 seconds. This also highlights the tradeoff between time and performance, as more training time does improve a model's performance, but after an optimal point, the rate of improvement slows.

Overall, the tradeoff analysis highlights the importance of carefully considering the specific requirements when selecting a model and fine-tuning it.

## Failure Analysis:

To analyze failures of the LSTM model and propose recommendations for improvement we decided to analyze three specific examples where our model failed:

**Example 1:**

| | |
|---|---|
| **Sentence** | bloor lansdowne gentrified dread think like |
| **True label** | Highly Biased |
| **Model's prediction** | Neutral |

Our model might not have recognized 'gentrified' as a biased term. Some of the possible reasons could be that the sentence is short without proper context and the lack of explicit racial references might have played a part which in itself could be a challenge for the model.

**Example 2:**

| | |
|---|---|
| **Sentence** | blame n n main people blame enslavement black african feel worst european pay enslavement african |
| **True label** | Slightly Biased |
| **Model’s prediction** | Highly Biased |

In this scenario, the presence of multiple racial terms and references might have led our model to overestimate the level of bias. This is similar to what we observed in our feature ablation section, that possibly our model is too sensitive and over-relying on the presence of explicit racial terms or references.

**Example 3:**

| | |
|---|---|
| **Sentence** | twitter done everyone else blacking look bit foolish |
| **True label** | Slightly Biased |
| **Model’s prediction** | Neutral |

This example highlights that our model struggles with less common terms that could be interpreted as racial bias depending on the context.

Below are our suggestions for future improvements:

1. As we observed in the tradeoff analysis section, the LSTM model’s quality was positively correlated with the size of the training data. One possible solution is to include other categories of bias and train the model on them to allow it to better classify between different categories of bias.
2. Implement strategies that could help improve the model’s performance to detect more subtle forms of bias without the use of explicit racial terms or references. This could involve including more words or phrases to capture a broad range of bias indicators and possibly use synthetic examples of subtle bias to detect less obvious cases.

## Unsupervised Learning:

### Motivation:

In this section of the project, we delved into the world of unsupervised learning to uncover hidden patterns and structures within a dataset related to racial bias. Our motivation was to explore how race was discussed in news articles, seeking to understand potential biases and the underlying themes within the data. We employed two powerful techniques: Latent Dirichlet Allocation (LDA) for topic modeling and K-means clustering for grouping similar documents.

To address our specific research questions:

- Were there distinct topics within these articles?
- Could we identify underlying themes related to race, bias, or specific events?
- Was it possible to group articles based on their content similarities?
- Did certain clusters exhibit common biases or patterns?

We leveraged various visualization techniques to interpret the results of our models. Visualizations such as word clouds, parallel coordinates plots, and silhouette plots helped us assess the quality of our topic modeling and clustering, ensuring that our findings were robust and meaningful.

In addition to these analyses, we performed sensitivity analyses on both the LDA and K-means models. This involved adjusting key parameters to understand how the models' outputs changed with different configurations. By doing so, we ensured that our conclusions were not sensitive to the particular choices of model parameters.

### Methods Description:

1. **LDA:** We first preprocessed the text data using TF-IDF(Term Frequency-Inverse Document Frequency) to weigh the importance of words in the documents. This step is crucial for the LDA model to understand the semantic structure of the text. We then trained the LDA model using the Gensim library, which provides efficient tools for topic modeling. To find the best LDA model, we defined a grid of hyperparameters to explore, including the number of topics, learning decay, and the maximum number of iterations. Using ParameterGrid from scikit-learn and parallel processing with joblib, we trained multiple LDA models with different combinations of these parameters and selected the one with the highest coherence score. Finally, we used visualization tools like pyLDAvis and MatPlot to interpret the topics discovered by LDA.
2. **K-Means Clustering:** We used K-means clustering to group the documents into clusters where each document belongs to the cluster with the nearest mean, serving as a prototype cluster. We then trained a Word2Vec model to generate word embeddings. These embeddings capture the semantic relationships between words and serve as features for clustering. We used the Davies-Bouldin index to find the optimal number of clusters, which is an internal evaluation method that evaluates the clustering algorithm's performance. Hyperparameter tuning was implemented by varying the number of clusters and selecting the one that minimizes the Davies_bouldin index, indicating better cluster performance.

### Unsupervised Evaluation:

To evaluate the unsupervised learning models, we performed several analyses. For LDA, we assessed the coherence score of the topics, which indicates the quality of the topics. We also performed sensitivity analysis by varying the number of topics around the optimal value found during hyperparameter tuning and evaluating the coherence score for each model.

For K-means, we used the Davies-Bouldin index to evaluate the clustering performance. We also performed sensitivity analysis by varying the number of clusters around the optimal value and calculating the Davies-Bouldin Index for each model.  Additionally, we used visualization techniques such as PCA, t-SNE, parallel coordinates plots, and silhouette plots to interpret the clusters and assess the clustering quality.

**Table 2: Model Performance (Unsupervised)**

| Model | Best Hyperparameters | Evaluation Metric | Score |
|---|---|---|---|
| **LDA with TF-IDF** | n_topics=5 | Coherence Score | 0.4846 |
| **K-means with Word Embeddings** | n_clusters=17 | Davies-Bouldin Index | 1.7539 |

### Main Findings:

Our unsupervised learning analysis yielded significant insights into the structure and content of the news articles related to racial bias.  The coherence scores for different numbers of topics indicated that the model with 5 topics (n-components: 5) had the highest coherence score of 0.4846, suggesting that this model best captured the underlying themes in the dataset.  This model was chosen as our best LDA model due to its superior ability to coherently represent the main topics within our dataset.  The LDA model, which was used for topic modeling, revealed five distinct themes within the dataset:

**Table 3: Key Words by Topic**

| Topic | Description | Key Words |
|---|---|---|
| Topic 1 | Discrimination, inequality, and social justice | black, white, racism, discrimination |
| Topic 2 | Identity and experiences of people of color | culture, history, community, heritage |
| Topic 3 | Impact of racial bias on specific groups | police, crime, poverty, education |
| Topic 4 | Role of media in perpetuating racial bias | news, media, coverage, reporting |
| Topic 5 | Importance of diversity and inclusions | Diversity, inclusion, representation, equality |

The topics identified by the LDA model provided a glimpse into the prevalent narratives and potentially uncovered instances of racial bias.  By examining these themes, we could gain a better understanding of how race is discussed in the news media.

In addition to topic modeling, we employed K-means clustering to group similar documents based on their content.  The optimal number of clusters was determined to be 17, as it minimized the Davies-Bouldin index, indicating better clustering performance.  This clustering allowed us to see how the articles naturally aggregate, which can provide further evidence of bias if certain clusters disproportionately represent specific racial narratives.

For our cluster visualizations, we use PCA and t-SNE. The PCA and t-SNE plots reduced the high-dimensional document vectors to a 2D space, enabling us to visualize the clusters and observe the distribution and separation of the clusters. PCA seeks to find the orthogonal projection of the data onto a lower dimensional space that maximizes the variance, while t-SNE is particularly well-suited for the visualization of high-dimensional datasets, as it preserves the local structure of the data.

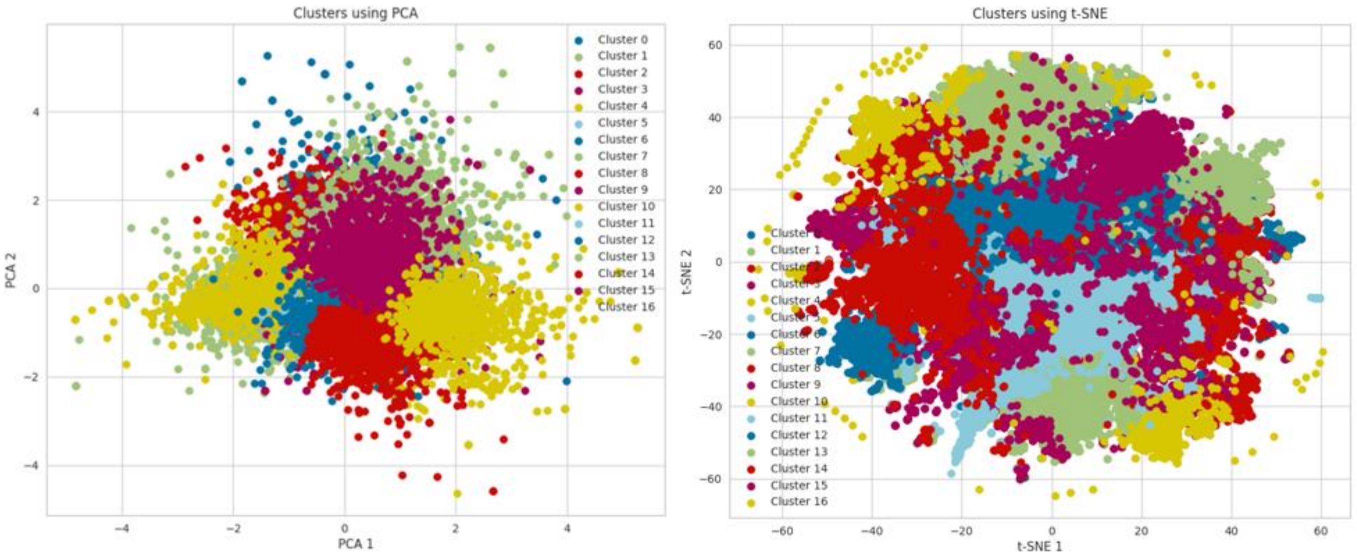


**Figure 5: Dimensionality Reduction Technique Comparison PCA vs t-SNE**

Although both PCA and t-SNE are dimensionality reduction techniques, as we can see from the results, they have different strengths and weaknesses. PCA preserves global data structure by capturing maximum variance in linear combinations of features, but it often results in overlapping clusters due to its linear nature. In contrast, t-SNE excels at revealing distinct clusters by maintaining local data structure, making it effective for visualizing non-linear relationships [4]. However, t-SNE is computationally intensive and sensitive to parameter settings. While PCA is faster and simpler, t-SNE provides clearer cluster visualizations and is the better dimensionality reduction technique for our project.

Another plot used to visualize our clusters was a Silhouette plot. Silhouette Plots visually assess the quality of clysters by showing how well-separated individual clusters are within a dataset. Positive silhouette coefficients indicate that data points within a cluster are more similar to each other than to points in other clusters, while negative coefficients suggest potential overlap or misclassifications. Silhouette plots can help determine the optimal number of clusters, identify poorly matched data points, and guide decisions on refining cluster assignments [3].

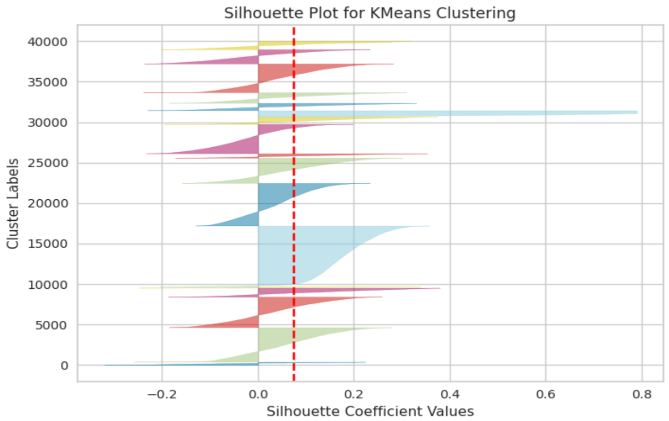


**Figure 6: Silhouette Plot**

Our Silhouette plot illustrates the distribution of the silhouette coefficient values across different clusters obtained from K-means clustering. Most of the data falls within the range of approximately 0.2 to just above 0.6 on the x-axis. This indicates that many data points are well-matched to their clusters and have positive coefficients. The thickness of silhouette plots provides insights into separation. Most clusters have good separation, as their silhouette plots do not cross significantly into negative values. However, some clusters have portions that dip below zero, suggesting potential overlap or misassignment. Overall, clusters exhibit moderate separation indicating the majority of the data points are well-assigned although there are areas for improvement. Addressing some of the overlapping regions by checking for outliers or more in-depth hyperparameter tuning could enhance the cluster quality.

**Sensitivity Analysis:**

Before we delve into the sensitivity analysis, let's understand why it's important. Sensitivity analysis helps us determine how robust our models are to changes in hyperparameters.  By analyzing the coherence scores for LDA and Davies-Boldin scores for K-means across a range of values around the optimal parameters, we can gain insights into the stability of our models and the confidence we can have in our results.

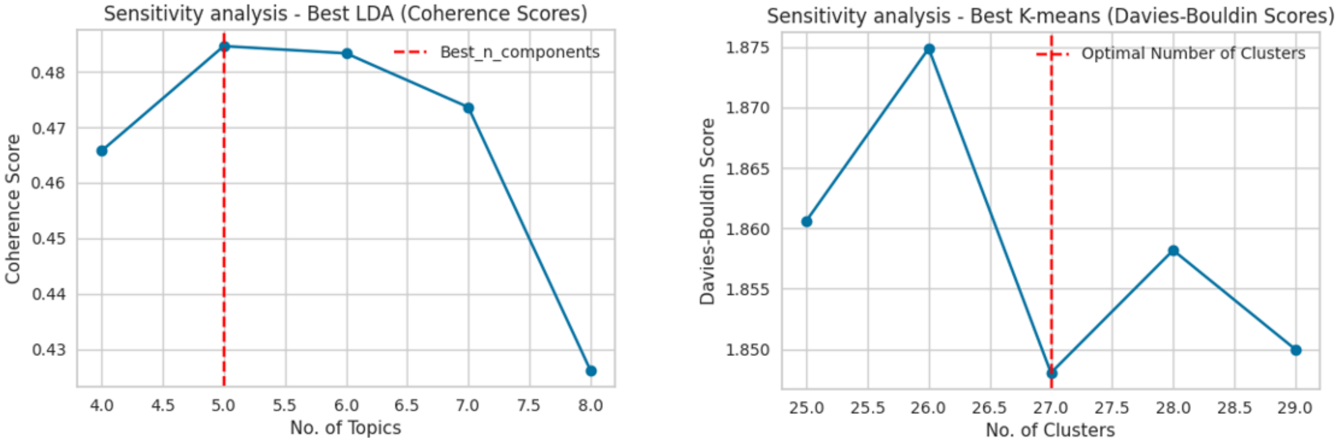


**Figure 7: Sensitivity Analysis of Best LDA and Best K-means**

The LDA sensitivity analysis plot shows the coherence scores for different numbers of topics, ranging from 4 to 8., with 5 being the optimal number. The x-axis represents the number of topics, and the y-axis shows the coherence score, which ranges from 0.42 to 0.49. The graph line starts near 0.465, peaks at approximately 0.489 with 5 topics, and gradually declines to 0.424 as the number of topics increases. This indicates that the model performs best at 5 topics. The results suggest that too few topics might not capture the full range of themes in the data, while too many topics could lead to overfitting, resulting in less and less meaningful groupings. The sensitivity analysis thus confirms that 5 topics represent the best balance for capturing the underlying themes in the dataset with the LDA model.

The K-means sensitivity analysis plot depicts the Davies-Bouldin Scores against different numbers of clusters.  The x-axis represents the number of clusters ranging from 15 to 19, while the y-axis represents the Davies-Bouldin Score ranging from 1.75 to 1.88.  The graph forms a 'V' shape, with the lowest score occurring at the optimal k = 17, indicating the best separation between clusters. As the number of clusters increases beyond this point, the score gradually rises. This suggests that using 17 clusters results in the most distinct and well-defined groups within the data. The Davies-Bouldin score measures the ratio of within-cluster distances to between-cluster distances, with a lower score indicating better clustering performance.  Therefore, the sensitivity analysis confirms that 17 clusters represent the optimal choice for K-means clustering in this dataset.

## Discussion:

### Supervised

Even though LSTM performed better than the traditional and more simpler models such as SVM and SGD, we were surprised to see that the performance was not as substantial as we initially thought given LSTM's capability of capturing context within text sequences. In addition, the ablation analysis showed that certain individual words such as 'racist' and 'racism' had a significant impact on our model's performance. This implies that the presence of these terms had a profound impact in bias categorization, but it was more pronounced than we expected. Lastly, it was quite impressive to see that SVM had a relatively good performance compared to LSTM even though the former is rather simpler in comparison with the latter.

One of the challenges we faced was achieving the best hyperparameter for each of the models, particularly BERT and LSTM due to time and computational resources limitations. We addressed it by implementing a combination of manual tuning to balance efficiency and effectiveness.

With more time and computational resources, we would fine-tune the BERT model, which has shown state-of-the-art performance on various NLP tasks, including text classification and automate pipelines for experimenting with BERT (DVC, MLFlow). In addition, with more time, we could explore further with advanced feature engineering techniques such as using pre-trained word embeddings like GloVe or FastText or add attention mechanisms in our neural network models.

### Unsupervised

The unsupervised learning phase yielded some surprising and encouraging results. The LDA model, despite its unsupervised nature, identified five distinct and well-defined themes related to racial bias, suggesting that consistent patterns exist in how race is discussed across a diverse set of news sources. Additionally, the robustness of both LDA and K-means models demonstrated through sensitivity analysis, instilled confidence in the findings. The models performed consistently well across a range of parameter values, indicating that the results are not overly sensitive to specific hyperparameter choices.

Challenges were encountered, particularly regarding potential biases within the data. To mitigate this, we carefully selected a diverse representative dataset and employed techniques like TF-IDF, which helps account for word frequencies and minimizes the impact of biased language patterns. Computational costs posed another hurdle, as both clustering and topic modeling are computationally intensive. We addressed this by leveraging optimized libraries like Gensim for LDA and scikit-learn for K-means, and by employing parallel processing to train multiple models simultaneously, speeding up the process.

With more time and resources, we could have expanded the research significantly. Analyzing a larger dataset, potentially gathered through web scraping, would have provided more robust and generalizable results. We could have explored advanced clustering techniques, such as hierarchical or density-based clustering, for additional insights. Additionally, incorporating publication dates into the analysis could have revealed how discussions of racial bias have evolved in the news media over time. Finally, exploring cross-cultural comparisons by analyzing news articles from different countries and cultures would have provided a broader understanding of how race and bias are discussed globally.

## Ethical Considerations:

Our project explored racial bias in news articles using both supervised and unsupervised learning methods. We were mindful of the ethical implications of working with this sensitive topic, taking steps to minimize potential harm, and promote responsible data analysis.

A key ethical concern is the potential for perpetuating existing biases. If the training data reflects societal prejudices, the model might learn and reinforce these biases, leading to discriminatory outcomes. In supervised learning, we addressed data bias by employing data augmentation techniques to balance the dataset and by carefully choosing and engineering features to minimize bias in the model's input. We also focused on using interpretable models, allowing us to understand why the model made specific predictions and identify potential biases in its reasoning. To further mitigate potential misinterpretations, we emphasized the importance of contextual understanding and human oversight in validating model findings.

Unsupervised learning also presents ethical challenges. For example, clustering algorithms could inadvertently group articles based on existing biases, further reinforcing those prejudices. We addressed these concerns by using a diverse set of features and visualizations to counter potential clustering biases, ensuring that our analysis did not inadvertently reinforce existing prejudices. We carefully interpreted the topics identified by LDA, considering potential biases in the language used within each topic. We acknowledged the potential impact of our findings on journalists, readers, and policymakers.

## Statement of Work:

| Ejaz Alam | Nikolay Jamgaryan | Amanda Fear |
|---|---|---|
| **Data Processing and Exploration:** All<br>**Unsupervised:** Sensitivity Analysis<br>**Supervised:** SGD, Linear SVM, LSTM, Feature Ablation Analysis, Sensitivity Analysis, Tradeoff Analysis, Failure Analysis, Visualizations<br>**General Project:** Research, Collaboration, Report Writing<br>**Additional:** Standup Templates, Final Project Template | **Supervised**: Naive Bayes, SVM, and BERT, Sensitivity Analysis and, Failure Analysis description, Tables<br>**General Project:** Research, Collaboration, Report writing | **Unsupervised:** Topic modeling with LDA, K-means clustering, PCA and t-SNE analysis, Sensitivity Analysis, Visualizations, Comments, and Insights<br>**General Project**: Research, Collaboration, Report writing |

## Appendix A – References:

[1] ASA Statistical Computing & Graphics Sections. (2014, September). LDAvis: A method for visualizing and interpreting topic models [Video file]. YouTube. https://www.youtube.com/watch?v=IksL96ls4o0. Accessed May 28, 2024

[2] Awan-Ur-Rahman. (2020, April 12). Latent Dirichlet Allocation (LDA): A Guide to Probabilistic Modeling Approach for Topic Discovery. Towards Data Science. https://towardsdatascience.com/latent-dirichlet-allocation-lda-a-guide-to-probabilistic-modeling-approach-for-topic-discovery-8cb97c08da3c. Accessed May 22, 2024

[3] Benjamin, B., & Ziegler, M. (n.d.). Silhouette Visualizer. In Scikit-Yellowbrick Documentation. Retrieved [June 1, 2024], from https://www.scikit-yb.org/en/develop/api/cluster/silhouette.html. Accessed June 15, 2024

[4] Babu, R. (2020, March 17). Understanding PCA and T-SNE intuitively. Analytics Vidhya (Medium). https://towardsdatascience.com/understanding-pca-and-t-sne-intuitively-126000205e7. Accessed June 5, 2024

[5] de Boer, D. (2022). Measuring racial bias within the Dutch public news outlet's coverage. Master Thesis, Applied Data Science, Utrecht University. https://studenttheses.uu.nl/bitstream/handle/20.500.12932/42457/Thesis_ADS_DMdeBoer_final.pdf?sequence=1. Accessed May 10, 2024

[6] Gupta, S., Bolden, S.E., Kachhadia, J., Korsunska, A., & Stromer-Galley, J. (2020). PoliBERT: Classifying political social media messages with BERT. Paper presented at the Social, Cultural and Behavioral Modeling (SBP-BRIMS 2020) conference. Washington, DC, October 18-21, 2020. Accessed May 12, 2024

[7] Hartigan, J. A., & Wong, M. A. (1979). Algorithm AS 136: A K-Means Clustering Algorithm. Journal of the Royal Statistical Society. Series C (Applied Statistics), 28(1), 100-108. Accessed June 6, 2024

[8] Machine Learning Plus. (n.d.). Gensim Overview. Machine Learning Plus. Retrieved [June 4, 2024], from https://www.machinelearningplus.com/nlp/gensim-tutorial/

[11] McInnes, L., Healy, J., & Melville, J. (2018). UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction. arXiv preprint arXiv:1802.03426.

[12] Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient Estimation of Word Representations in Vector Space. arXiv preprint arXiv:1301.3781.

[13] Mozafari, M., Farahbakhsh, R., & Crespi, N. (2020). Hate speech detection and racial bias mitigation in social media based on BERT model. PLOS ONE, 15(8), e0237861. https://doi.org/10.1371/journal.pone.0237861. Accessed May 10, 2024

[14] Raza, S. (2023). Navigating News Narratives: A Media Bias Analysis Dataset [Data set]. Zenodo. https://zenodo.org/records/10231028. Accessed June 6, 2024

All notebooks, charts, and experiment results can be found in the GitHub repository

This is an acknowledgment of the use of GenAI for assistance in the coding portion for debugging and cleaning purposes.

# Appendix B – Data Schema:

## Data Source:

Navigating News Narratives: A Media Bias Analysis Dataset

## Description:

The “Navigating News Narratives: A Media Bias Analysis Dataset” is a comprehensive dataset designed to address the urgent need for tools to detect and analyze media bias. It covers a broad spectrum of biases, making it a valuable asset in the field of media studies and artificial intelligence.

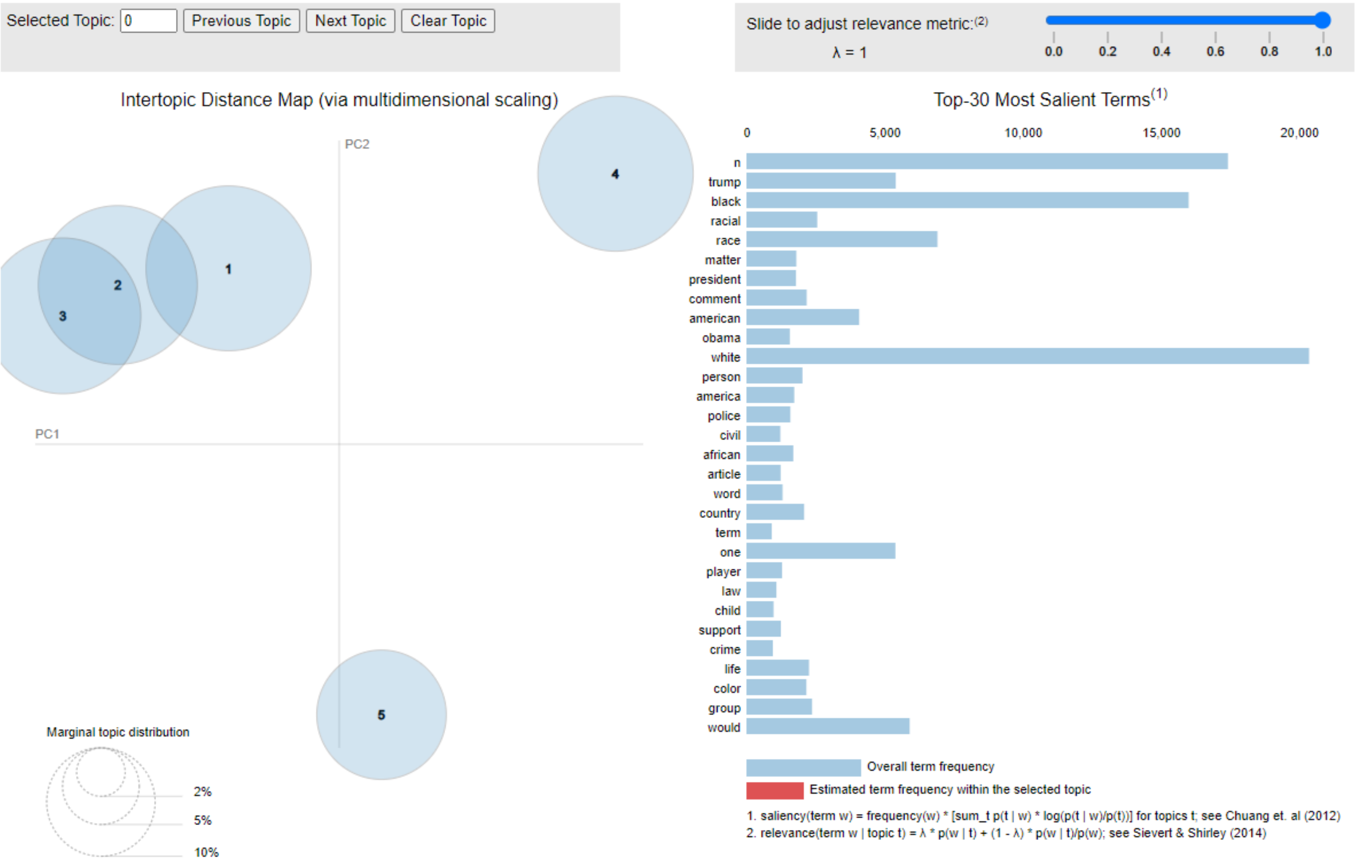
| Column Name | Description | Data Type | Original | Supervised | Unsupervised |
|---|---|---|---|---|---|
| text | The main content | String | Y | N | N |
| dimension | Descriptive category of text | String | Y | N | N |
| biased_words | A compilation of words regarded as biased | String | Y | N | N |
| aspect | Specific sub-topic within the main content | String | Y | N | N |
| label | Indicates the presence (True) or absence (False) of bias | String | Y | N | N |
| sentiment | Classified as ‘Positive’, ‘Negative’, or ‘neutral’ | String | Y | N | N |
| preprocessed_text | The main content after being processed for analysis | String | N | Y | Y |
| cluster | Cluster preprocessed_text was in | int64 | N | N | Y |
| doc_vectors | Numerical representation of text document | object | N | N | Y |
| label_encoded | Numerical representation of label column | int32 | N | Y | N |

## Appendix C - List of Features:

We added links to the complete list of features as separate CSV files below and these files will also be submitted with the report.

1. TF-IDF features can be downloaded from here: 'tfidf_features.csv'. It contains all the features used with 'Feature' and 'TF-IDF Score' as the columns.
2. Word2Vec vocabulary can be downloaded from here: 'word2vec_features.csv'. It contains all the words included with 'Word' and 'Frequency' as the columns.
3. LSTM input features can be downloaded from here: 'lstm_features.csv'. It contains all the words used as input features with 'Word' and 'Index' as the columns.

# Appendix D - Additional Graphs:



1. saliency(term w) = frequency(w) * [sum_t p(t | w) * log(p(t | w)/p(t))] for topics t; see Chuang et. al (2012)
2. relevance(term w | topic t) = λ * p(w | t) + (1 - λ) * p(w | t)/p(w); see Sievert & Shirley (2014)

**Figure: LDA Interactive visualization using pyLDAvis**

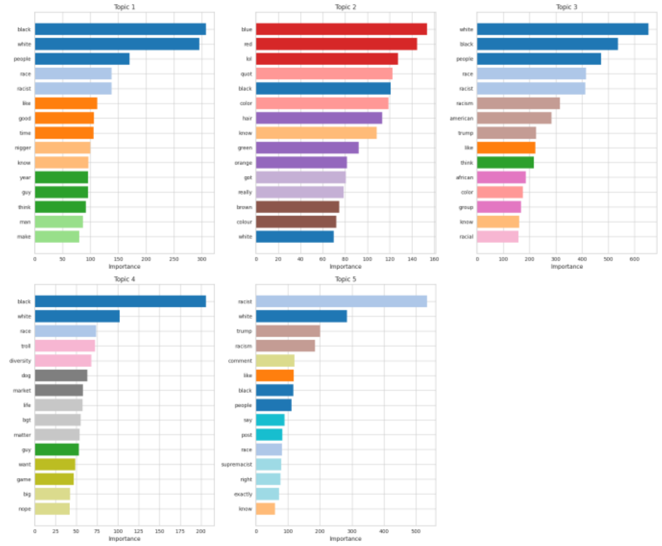


**Figure Caption: Top Words in Each Topic - LDA**

This figure presents bar charts illustrating the top 10 words for each of the five topics discovered by the Latent Dirichlet Allocation (LDA) model. Each bar chart corresponds to a distinct topic, with the x-axis representing the importance of each word within that topic. The importance is quantified by the word's weight in the LDA model, which reflects how strongly the word is associated with the topic.

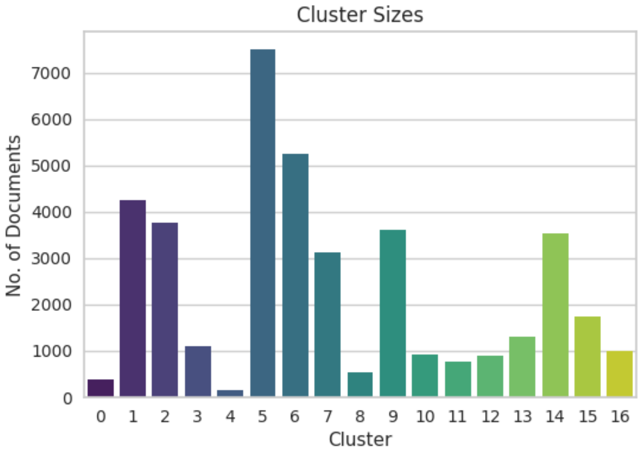


**Figure Caption: Visualizing Cluster Sizes**

This bar plot illustrates the distribution of document counts across 17 distinct clusters. Each bar corresponds to one of the clusters, labeled from 0 to 16. The height of each bar represents the number of documents within that cluster. Notably, Cluster 5 stands out with a significantly larger document count compared to other clusters.

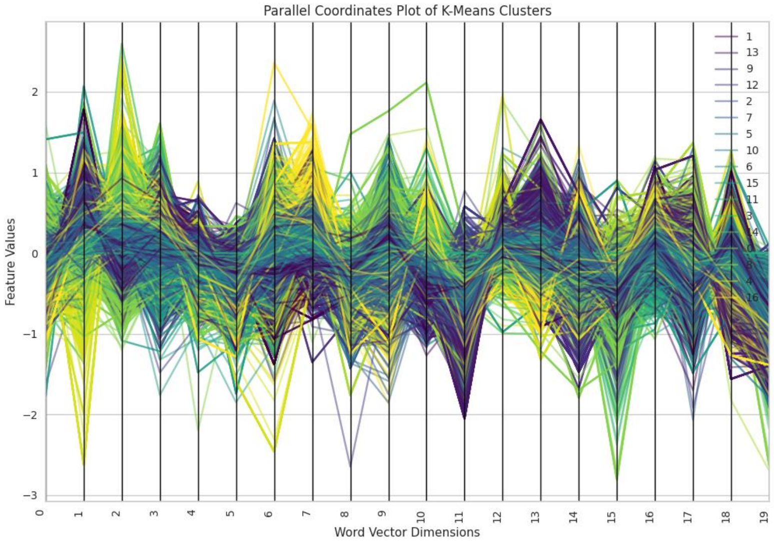


**Figure Caption: Parallel Coordinate Plot**

This Parallel Coordinate Plot represents multi-dimensional data using parallel axes, where each line corresponds to an observation (data point). The vertical axes represent different features derived from word vectors. These word vectors capture semantic meaning, context, or similarity. The colored lines intersect these axes, indicating the values of each feature for a given observation. The distinct colors represent clusters identified by the K-means algorithm. For instance, Cluster 5 exhibits similar paths across the axes, suggesting similarity within that group. In contrast, other clusters show more variation, indicating a wider range of characteristics.