# *New York Times Comments*

New York Times has a wide audience and plays a prominent role in shaping people's opinion and outlook on current affairs and in setting the tone of the public discourse, especially in the USA. The comment section in the articles is very active and it gives a glimpse of readers' take on the matters concerning the articles. The data contains information about the comments made on the articles published in New York Times in Jan-May 2017 and Jan-April 2018. The month-wise data is given in two csv files - one each for the articles on which comments were made and for the comments themselves. The csv files for comments contain over 2 million comments in total with 34 features and those for articles contain 16 features about more than 9,000 articles. Given the constraints of our computational resources, our analysis (models) is focused on a curated dataset encompassing comments and articles from January 2017. This deliberate temporal limitation allows us to manage the scope of our project effectively while still gleaning valuable insights from a substantial volume of user interactions and content.

The data set is rich in information containing comments' texts, that are largely very well written, along with contextual information such as section/topic of the article, as well as features indicating how well the comment was received by the readers such as *editorsSelection* and *recommendations*. This data can serve the purpose of understanding and analyzing the public mood.

At the heart of our project lies a sophisticated endeavor: to predict user engagement rates within the bustling ecosystem of The New York Times comments section. By harnessing the nuanced patterns of past user interactions, we aim to construct a predictive model that not only serves as a barometer for reader sentiment but also shapes a more engaging, dynamic, and responsive discussion platform.

Our selection of engagement rate prediction as the focal point of our project is indeed a departure from the common analytical approaches applied to The New York Times comments. Typically, studies and models concentrate on more visible and conventional classification problems, such as sentiment analysis, topic modeling, or identifying the prevalence of certain opinions. These are valuable in their own right, but they do not directly address the predictive intricacies of engagement.

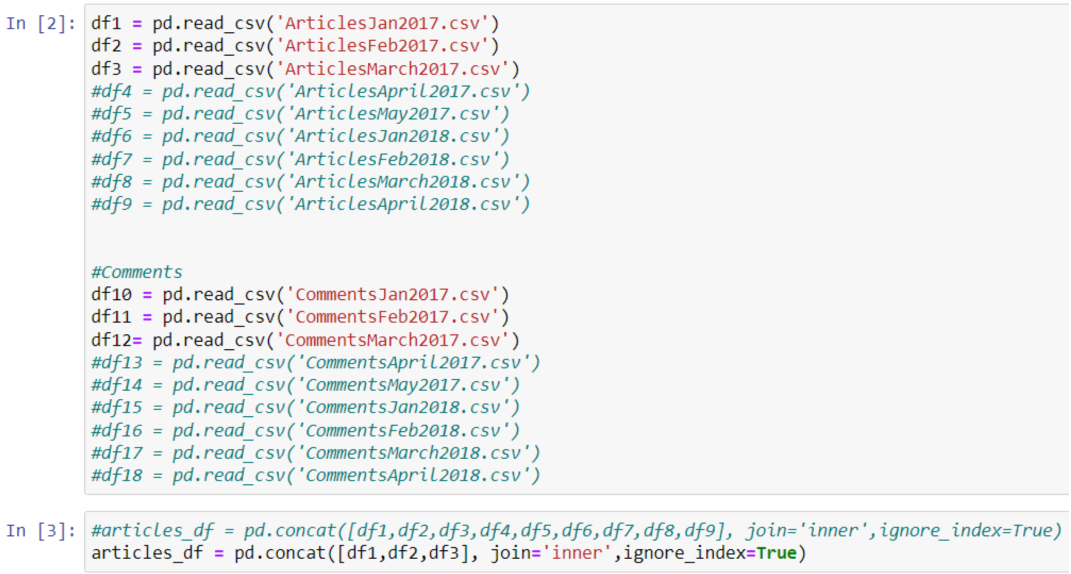
Engagement rate prediction is unique because it requires us to forecast a quantifiable outcome based on a range of less tangible factors, such as the interplay of reader sentiments, the timing of comments, and the evolving context of the news cycle. It is a challenge that goes beyond categorizing past behavior; it is about anticipating future interactions and the collective pulse of the readership.

This approach deviates from the norm because it does not simply seek to understand what sort of comments are made or what the prevailing sentiments are. Instead, it endeavors to map out the likelihood of user engagement in a forward-looking manner, thus embracing the complexity of predictive modeling. Engagement, in this sense, is not just a metric; it's a multifaceted outcome that is influenced by dynamic content, user behavior, and temporal factors.

The potential impact of a successful solution in this domain is substantial. For the publishers, it equates to an unparalleled understanding of reader engagement, enabling a data-driven approach to content creation and community management that can significantly amplify reader satisfaction and loyalty. Advertisers and strategists could leverage this insight to tailor their campaigns with unprecedented precision, thus maximizing the efficacy of their outreach efforts.

Yet, the road ahead is fraught with challenges. The digital footprints of user behavior are complex, influenced by a multitude of factors that range from the content of the article to the global news climate. Ensuring privacy and ethical use of data is paramount; it's a delicate balance to maintain while striving for analytical depth. Furthermore, the evolving nature of language, sentiment, and social norms adds layers of complexity to an already intricate task. Adapting to these shifts is not merely a technical challenge but an ongoing commitment to learning and evolution.

Loading the data:

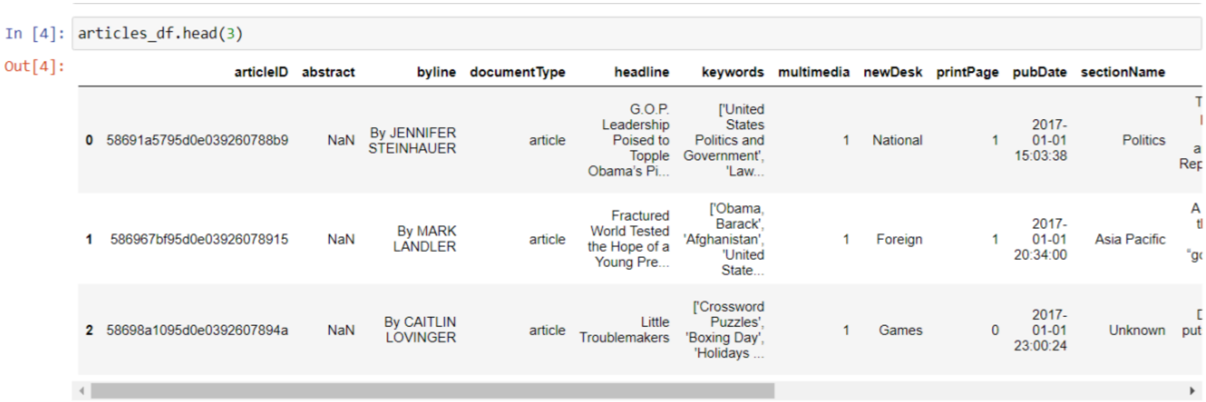


```
In [2]: df1 = pd.read_csv('ArticlesJan2017.csv')
        df2 = pd.read_csv('ArticlesFeb2017.csv')
        df3 = pd.read_csv('ArticlesMarch2017.csv')
        #df4 = pd.read_csv('ArticlesApril2017.csv')
        #df5 = pd.read_csv('ArticlesMay2017.csv')
        #df6 = pd.read_csv('ArticlesJan2018.csv')
        #df7 = pd.read_csv('ArticlesFeb2018.csv')
        #df8 = pd.read_csv('ArticlesMarch2018.csv')
        #df9 = pd.read_csv('ArticlesApril2018.csv')


        #Comments
        df10 = pd.read_csv('CommentsJan2017.csv')
        df11 = pd.read_csv('CommentsFeb2017.csv')
        df12= pd.read_csv('CommentsMarch2017.csv')
        #df13 = pd.read_csv('CommentsApril2017.csv')
        #df14 = pd.read_csv('CommentsMay2017.csv')
        #df15 = pd.read_csv('CommentsJan2018.csv')
        #df16 = pd.read_csv('CommentsFeb2018.csv')
        #df17 = pd.read_csv('CommentsMarch2018.csv')
        #df18 = pd.read_csv('CommentsApril2018.csv')
```

```
In [3]: #articles_df = pd.concat([df1,df2,df3,df4,df5,df6,df7,df8,df9], join='inner',ignore_index=True)
        articles_df = pd.concat([df1,df2,df3], join='inner',ignore_index=True)
```

Below is the article dataset in tabular form

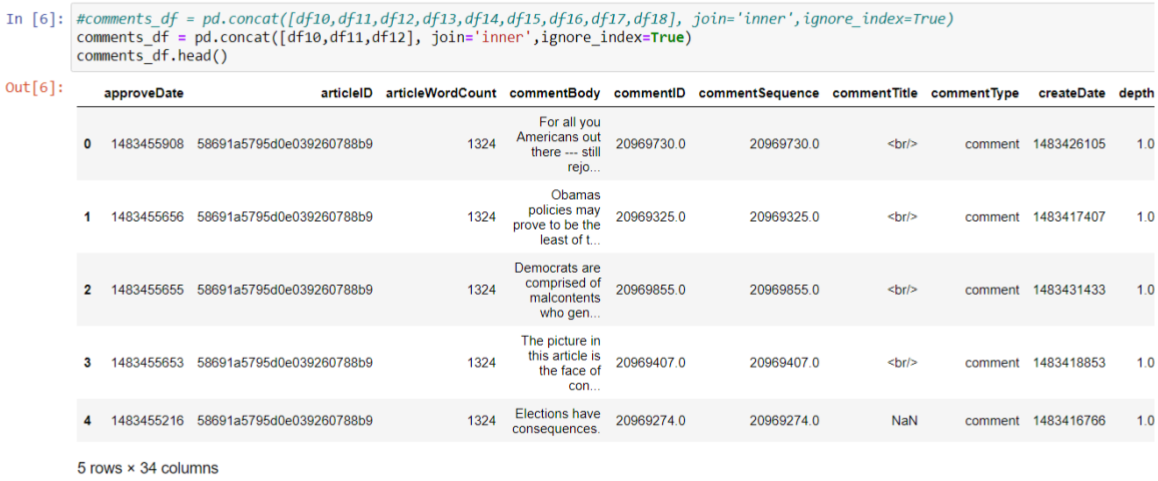


```
In [4]: articles_df.head(3)
```

Out[4]:

|   | articleID | abstract | byline | documentType | headline | keywords | multimedia | newDesk | printPage | pubDate | sectionName |   |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 58691a5795d0e039260788b9 | NaN | By JENNIFER STEINHAUER | article | G.O.P. Leadership Poised to Topple Obama's Pi... | ['United States Politics and Government', 'Law... | 1 | National | 1 | 2017-01-01 15:03:38 | Politics | T I a Rep |
| **1** | 586967bf95d0e03926078915 | NaN | By MARK LANDLER | article | Fractured World Tested the Hope of a Young Pre... | ['Obama, Barack', 'Afghanistan', 'United State... | 1 | Foreign | 1 | 2017-01-01 20:34:00 | Asia Pacific | A tl "gc |
| **2** | 58698a1095d0e0392607894a | NaN | By CAITLIN LOVINGER | article | Little Troublemakers | ['Crossword Puzzles', 'Boxing Day', 'Holidays ... | 1 | Games | 0 | 2017-01-01 23:00:24 | Unknown | D put |

And below is the comments dataset in tabular form.



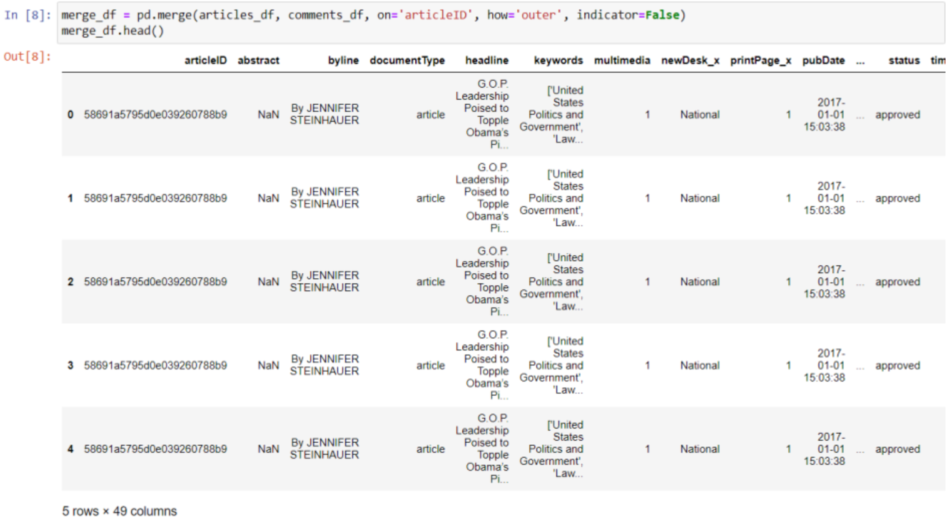
```
In [6]: #comments_df = pd.concat([df10,df11,df12,df13,df14,df15,df16,df17,df18], join='inner',ignore_index=True)
        comments_df = pd.concat([df10,df11,df12], join='inner',ignore_index=True)
        comments_df.head()
```

Out[6]:

|   | approveDate | articleID | articleWordCount | commentBody | commentID | commentSequence | commentTitle | commentType | createDate | depth |
|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 1483455908 | 58691a5795d0e039260788b9 | 1324 | For all you Americans out there --- still rejo... | 20969730.0 | 20969730.0 | <br/> | comment | 1483426105 | 1.0 |
| **1** | 1483455656 | 58691a5795d0e039260788b9 | 1324 | Obamas policies may prove to be the least of t... | 20969325.0 | 20969325.0 | <br/> | comment | 1483417407 | 1.0 |
| **2** | 1483455655 | 58691a5795d0e039260788b9 | 1324 | Democrats are comprised of malcontents who gen... | 20969855.0 | 20969855.0 | <br/> | comment | 1483431433 | 1.0 |
| **3** | 1483455653 | 58691a5795d0e039260788b9 | 1324 | The picture in this article is the face of con... | 20969407.0 | 20969407.0 | <br/> | comment | 1483418853 | 1.0 |
| **4** | 1483455216 | 58691a5795d0e039260788b9 | 1324 | Elections have consequences. | 20969274.0 | 20969274.0 | NaN | comment | 1483416766 | 1.0 |

5 rows × 34 columns

This is articles and comments data frames merged.

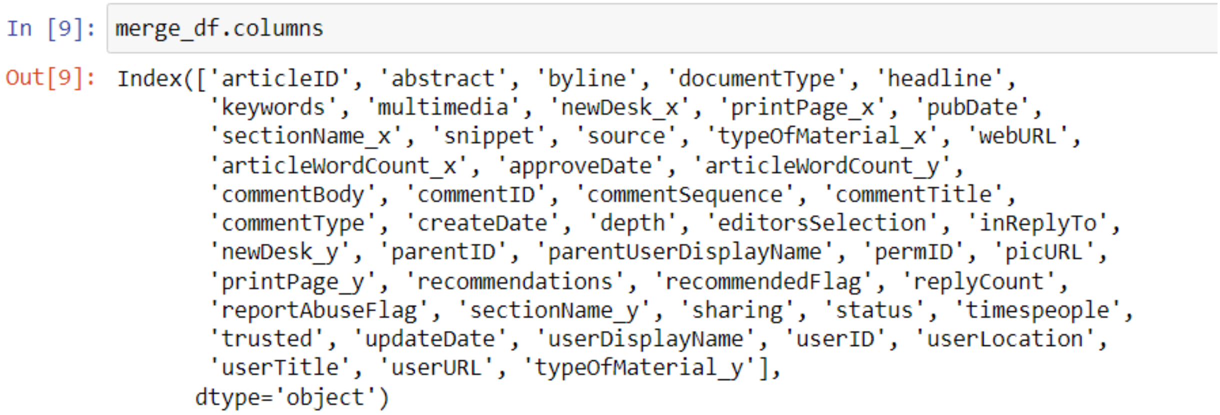


```
In [8]: merge_df = pd.merge(articles_df, comments_df, on='articleID', how='outer', indicator=False)
        merge_df.head()
```

Out[8]:

|   | articleID | abstract | byline | documentType | headline | keywords | multimedia | newDesk_x | printPage_x | pubDate | ... | status | tim |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 58691a5795d0e039260788b9 | NaN | By JENNIFER STEINHAUER | article | G.O.P. Leadership Poised to Topple Obama's Pi... | ['United States Politics and Government', 'Law... | 1 | National | 1 | 2017-01-01 15:03:38 | ... | approved |   |
| **1** | 58691a5795d0e039260788b9 | NaN | By JENNIFER STEINHAUER | article | G.O.P. Leadership Poised to Topple Obama's Pi... | ['United States Politics and Government', 'Law... | 1 | National | 1 | 2017-01-01 15:03:38 | ... | approved |   |
| **2** | 58691a5795d0e039260788b9 | NaN | By JENNIFER STEINHAUER | article | G.O.P. Leadership Poised to Topple Obama's Pi... | ['United States Politics and Government', 'Law... | 1 | National | 1 | 2017-01-01 15:03:38 | ... | approved |   |
| **3** | 58691a5795d0e039260788b9 | NaN | By JENNIFER STEINHAUER | article | G.O.P. Leadership Poised to Topple Obama's Pi... | ['United States Politics and Government', 'Law... | 1 | National | 1 | 2017-01-01 15:03:38 | ... | approved |   |
| **4** | 58691a5795d0e039260788b9 | NaN | By JENNIFER STEINHAUER | article | G.O.P. Leadership Poised to Topple Obama's Pi... | ['United States Politics and Government', 'Law... | 1 | National | 1 | 2017-01-01 15:03:38 | ... | approved |   |

5 rows × 49 columns



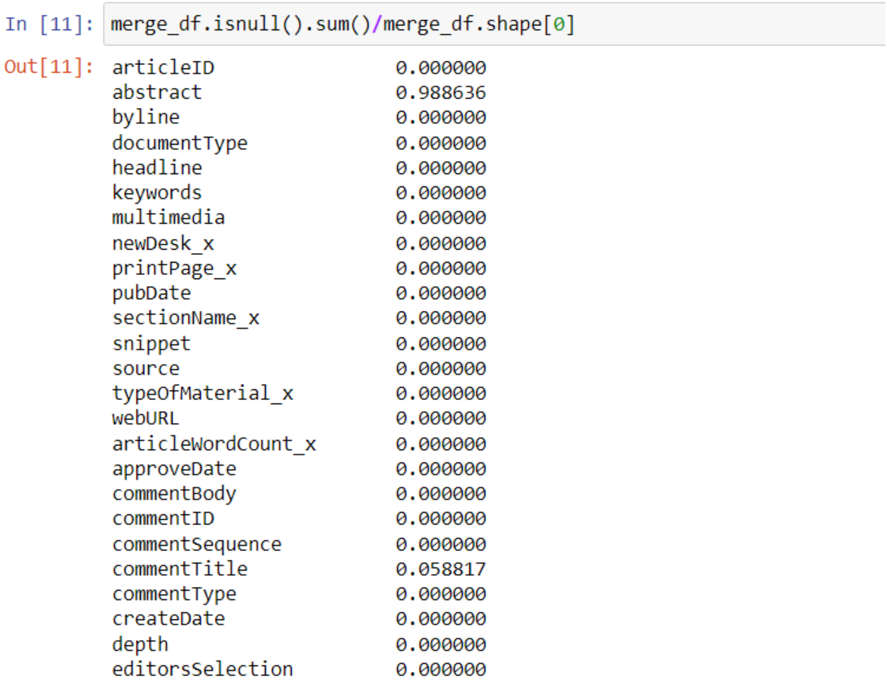
```
In [9]:  merge_df.columns

Out[9]:  Index(['articleID', 'abstract', 'byline', 'documentType', 'headline',
                'keywords', 'multimedia', 'newDesk_x', 'printPage_x', 'pubDate',
                'sectionName_x', 'snippet', 'source', 'typeOfMaterial_x', 'webURL',
                'articleWordCount_x', 'approveDate', 'articleWordCount_y',
                'commentBody', 'commentID', 'commentSequence', 'commentTitle',
                'commentType', 'createDate', 'depth', 'editorsSelection', 'inReplyTo',
                'newDesk_y', 'parentID', 'parentUserDisplayName', 'permID', 'picURL',
                'printPage_y', 'recommendations', 'recommendedFlag', 'replyCount',
                'reportAbuseFlag', 'sectionName_y', 'sharing', 'status', 'timespeople',
                'trusted', 'updateDate', 'userDisplayName', 'userID', 'userLocation',
                'userTitle', 'userURL', 'typeOfMaterial_y'],
               dtype='object')
```

**Task 2 – EDA:**

Acknowledging the limitations imposed by our computational capacity, we concentrated our exploratory data analysis (EDA) on a more expansive dataset, comprising comments and articles spanning three months. This extended timeframe provides us with a richer, more nuanced understanding of user behavior and content engagement, offering a solid foundation for our predictive modeling despite the initial constraint to January 2017 for model training purposes.

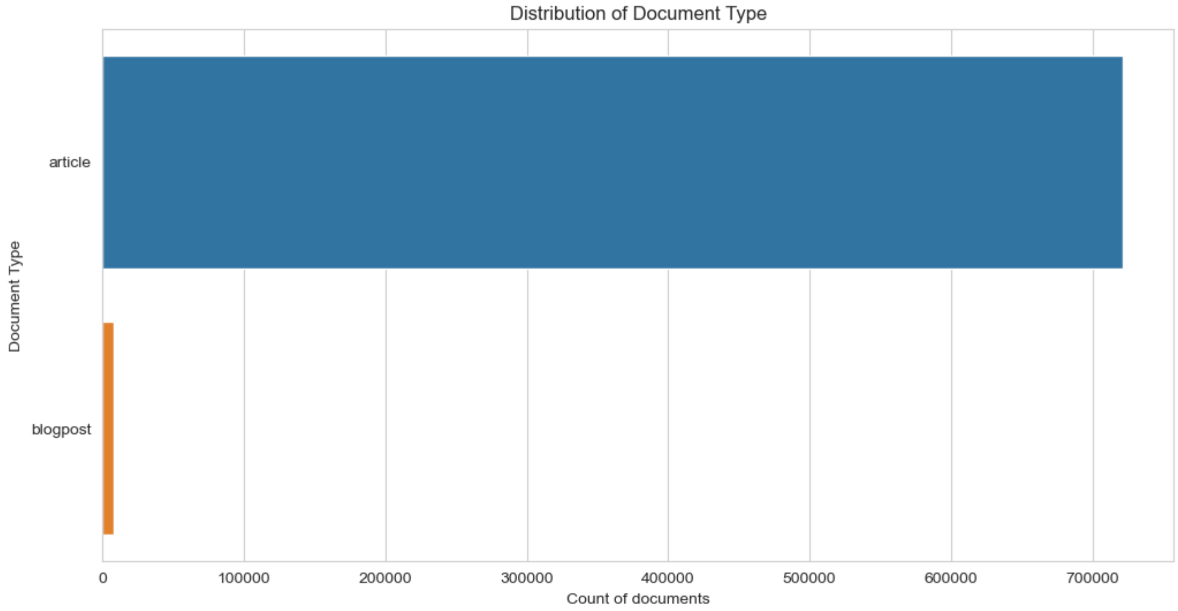
**Checking for NA values**



```
In [11]:  merge_df.isnull().sum()/merge_df.shape[0]

Out[11]:  articleID             0.000000
          abstract              0.988636
          byline                0.000000
          documentType          0.000000
          headline              0.000000
          keywords              0.000000
          multimedia            0.000000
          newDesk_x             0.000000
          printPage_x           0.000000
          pubDate               0.000000
          sectionName_x         0.000000
          snippet               0.000000
          source                0.000000
          typeOfMaterial_x      0.000000
          webURL                0.000000
          articleWordCount_x    0.000000
          approveDate           0.000000
          commentBody           0.000000
          commentID             0.000000
          commentSequence       0.000000
          commentTitle          0.058817
          commentType           0.000000
          createDate            0.000000
          depth                 0.000000
          editorsSelection      0.000000
```

Checking for null values and removing the unnecessary columns since given this extensive missing data, it is not advisable to impute the values. Doing so could introduce biases, lead to overfitting in predictive models, and compromise the accuracy of any analysis.



The provided count plot visualizes the distribution of document types in terms of the number of articles. In this plot, it's evident that articles have a larger count compared to other document types, such as blog posts. This difference in counts can be attributed to several factors:

1. **Formality and Purpose:**

   Articles are often associated with more formal and informative writing. They tend to be used for reporting news, providing in-depth analysis, or presenting research findings. This formality often leads to a higher count, as articles are created for various purposes and by different authors, such as journalists, researchers, or experts in their fields.

   On the other hand, blog posts are typically more informal and may have a personal or opinionated tone. They are often authored by individuals or bloggers who express their thoughts and experiences. Blog posts may serve a narrower range of purposes and have a smaller count in the dataset.

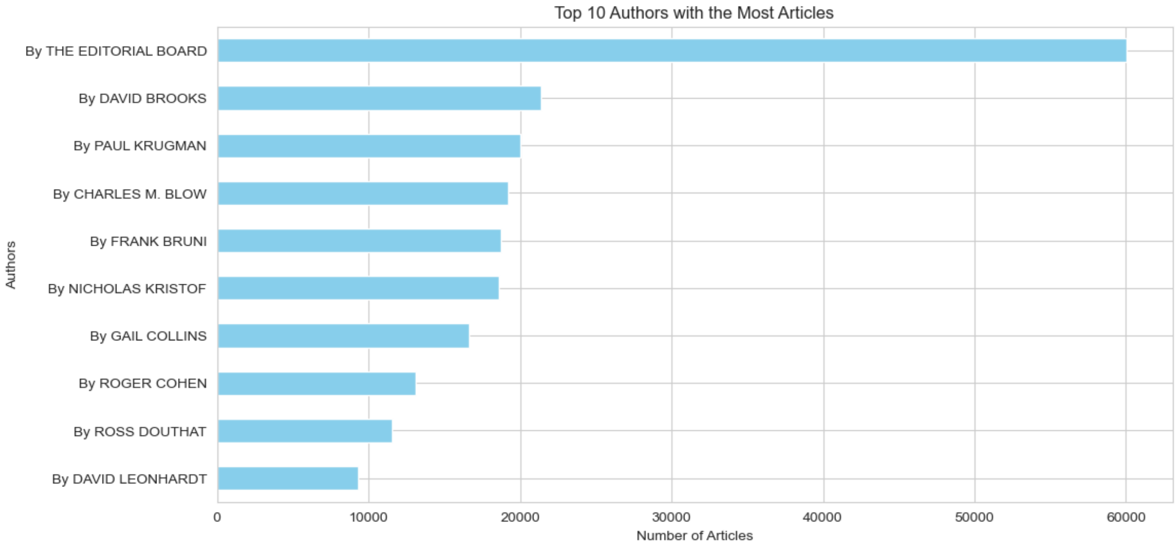
2. **Audience and Platform:**

Articles are commonly found in established and reputable publications, including newspapers, magazines, academic journals, and news websites. These platforms attract a broad readership and contribute to the higher count of articles.

Blog posts are typically published on personal or specialized blogs. While they have their own audience, the reach of individual blogs can vary, leading to a smaller count compared to articles from widely recognized sources.

### 3. Overlap in Document Types:

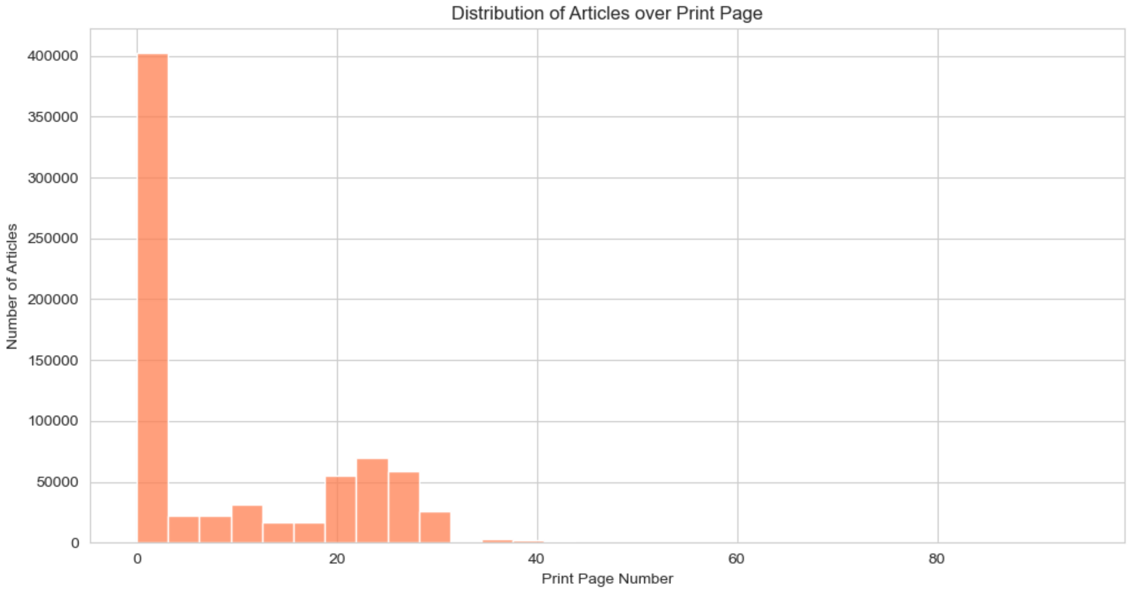
It's possible that domain-specific content or content from experts in a particular field may be featured in both articles and blog posts. This overlap can contribute to a larger count of articles, as they encompass a wider range of topics and authors.



The image shows the top 10 authors with the most articles published in the given period. The x-axis shows the number of articles published, and the y-axis shows the author's name.

The image shows that the most prolific author is The Editorial Board, with over 60,000 articles published. The other authors in the top 10 are also highly productive, with each having published over 20,000 articles.

The image also shows that the top 10 authors are a diverse group, representing a variety of different genres and perspectives. For example, The Editorial Board is a collective of writers who contribute to a news organization, while David Brooks is a conservative political commentator. Paul Krugman is a liberal economist, while Charles M. Blow is a black columnist.

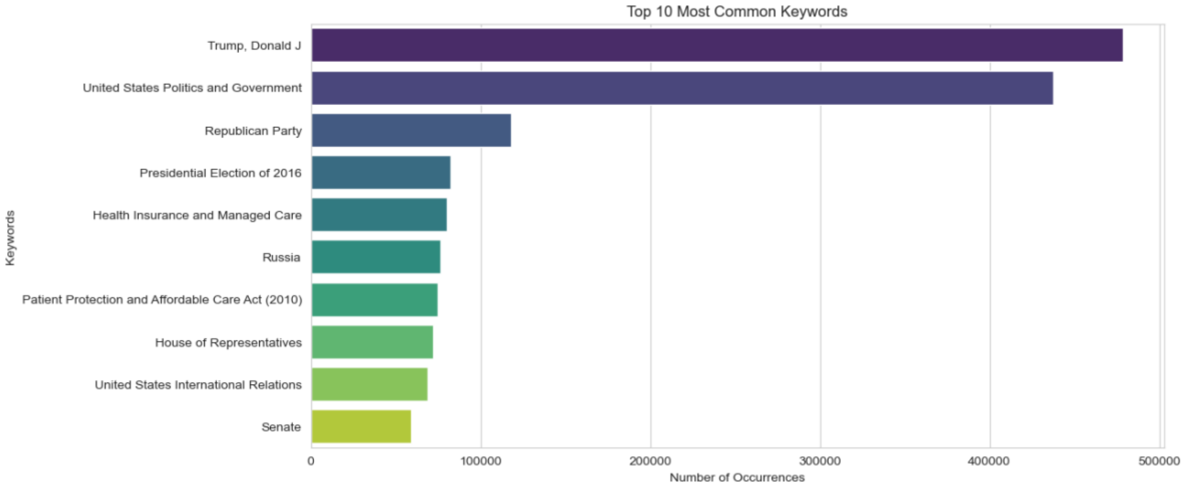


The graph shows the distribution of articles over print pages. The x-axis shows the print page number, and the y-axis shows the number of articles. The graph shows that the number of articles increases as the print page number increases.

One possible explanation for this is that newspapers typically start with the most important articles on the front page, and then move to less important articles on subsequent pages. This is because newspapers want to grab readers' attention with the most important news first.

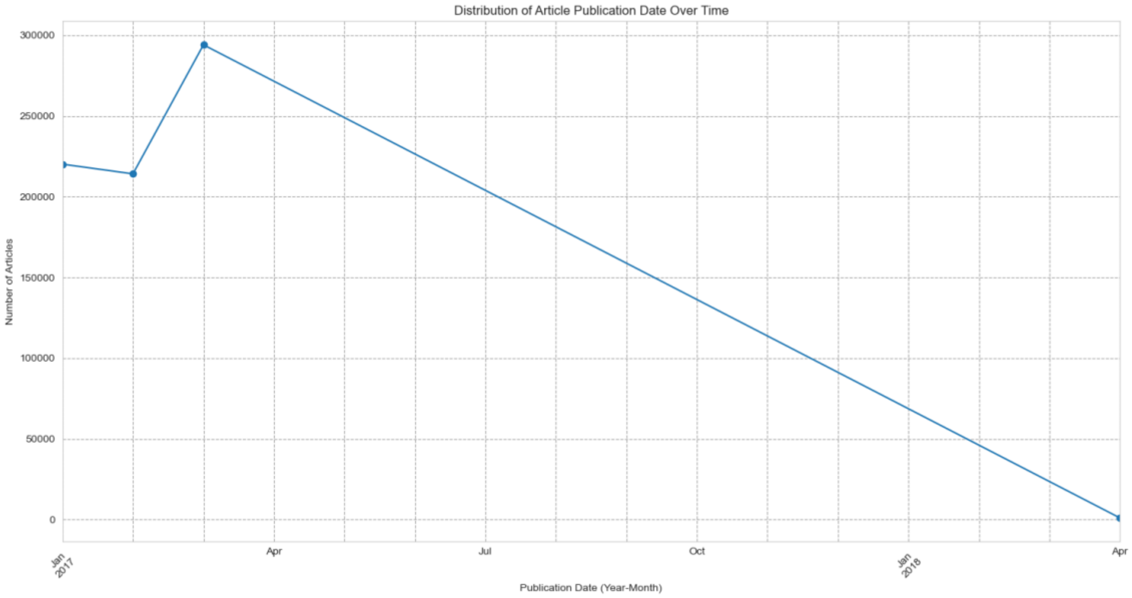
Another possible explanation for the graph is that newspapers may have different sections for different types of articles. For example, the front page may focus on news and current events, while the back pages may focus on sports, entertainment, and lifestyle. This would mean that there are more articles on the front page because it is the most visible section of the newspaper.

Finally, it is also possible that the graph is simply a reflection of the fact that newspapers are getting shorter. In recent years, there has been a trend towards online news consumption, and newspapers have been losing readers. This has led to newspapers reducing the number of pages they print. As a result, there is less space for articles, and the number of articles per page has increased.



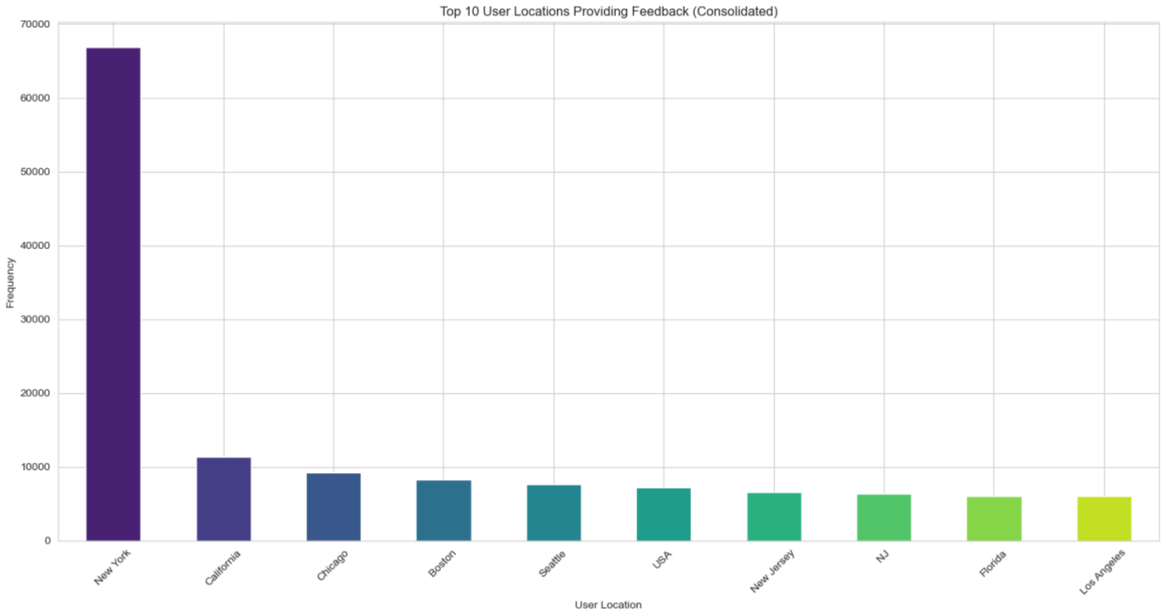
The image shows the top 10 most common keywords in the United States published in New York Times, in Jan-May 2017 and Jan-April 2018. The keywords are listed in order of popularity, with the most popular keyword at the top.

The image shows the top 10 most common keywords in the United States published in The New York Times from January to May 2017 and January to April 2018. The keywords are listed in order of popularity, with the most popular keyword at the top. The keyword with the most occurrences is 'Donald J Trump.' This is largely due to his election as the President of the United States of America in November 2016, with his presidential inauguration taking place on January 20, 2017. The second most frequently used keyword is 'United States Politics and Government. The Presidential Elections in 2016 overall was the biggest subject of discussion.



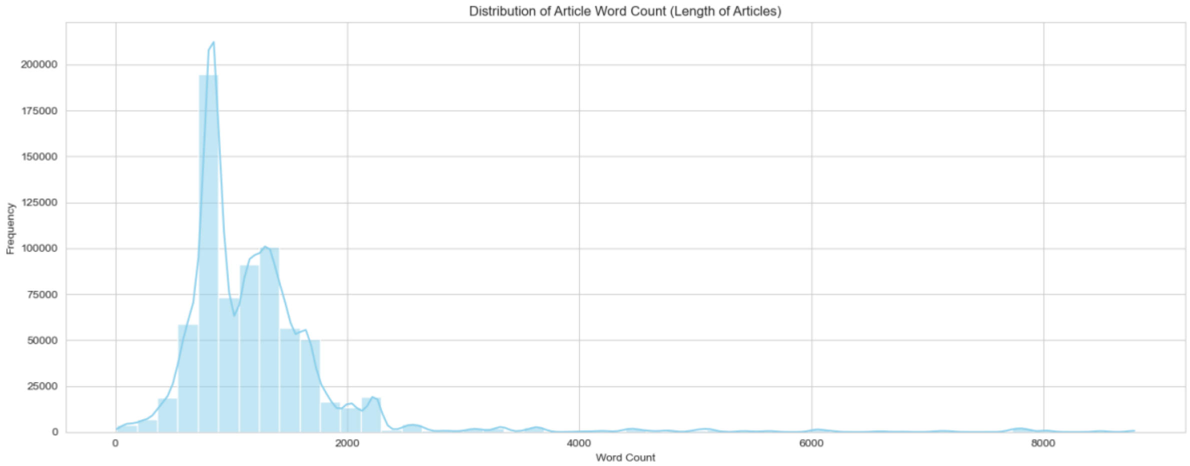
The figure reveals an intriguing pattern in the publication dates of articles, painting a picture of two distinct peaks amidst a gradual decline.

In January 2017, the number of articles reached a remarkable high of approximately 230,000, indicating a surge in scholarly output. This was followed by an even sharper spike in March 2017, suggesting a period of intense research activity. However, this burst of productivity was not sustained, and the subsequent months witnessed a steady decrease in article submissions. Overall, the figure highlights the dynamic nature of article publication, with periods of intense activity followed by more subdued phases.



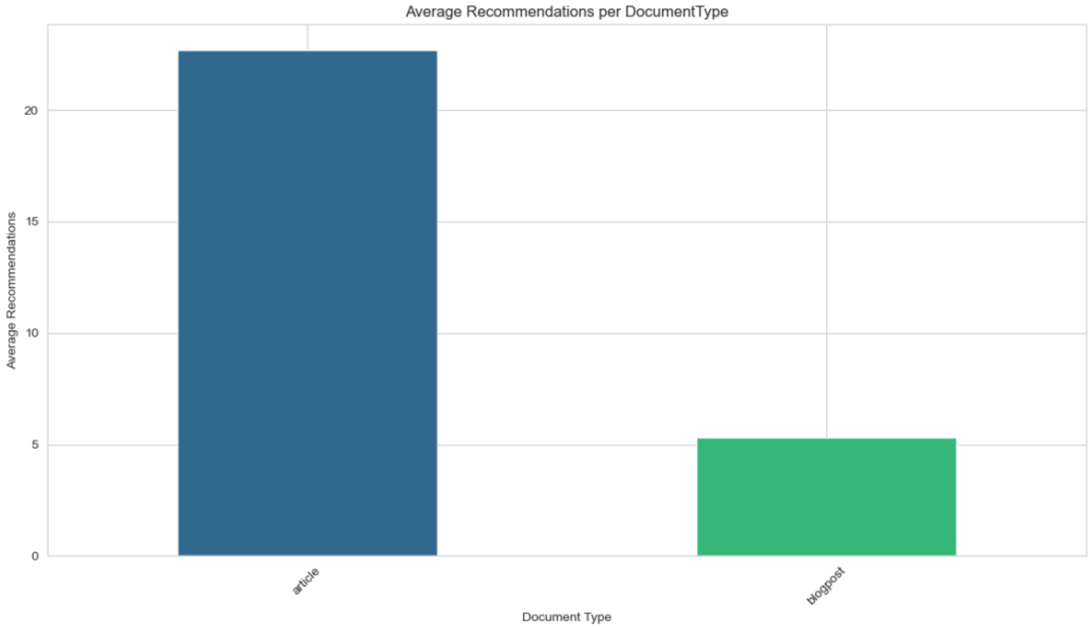
The image depicts a bar graph that showcases the top 10 locations in the United States that have the highest number of users providing feedback. The graph indicates that the top 10 user locations, based on frequency considered, are New York, California, Chicago, Boston, Seattle, USA, New Jersey, Florida, Los Angeles, and Washington.

The highest frequency of feedback is from New York, which accounts for approximately 67,000 of all feedback provided. California comes in second place with 11,000, and Chicago follows with 8,000. Boston, Seattle, and USA are next, with 7,600, 7,400, and 7,200 respectively. New Jersey, with 6,800, Florida with 6,600, and Los Angeles with 6,500 round out the top 10 locations.



The provided image illustrates the distribution of article word count, which indicates the number of words in each article within a particular dataset. The graph shows that most articles fall within the 0-2000 words range, with a peak around 500 words. This suggests that a significant number of articles are of shorter length, encompassing sufficient details without becoming excessively lengthy.

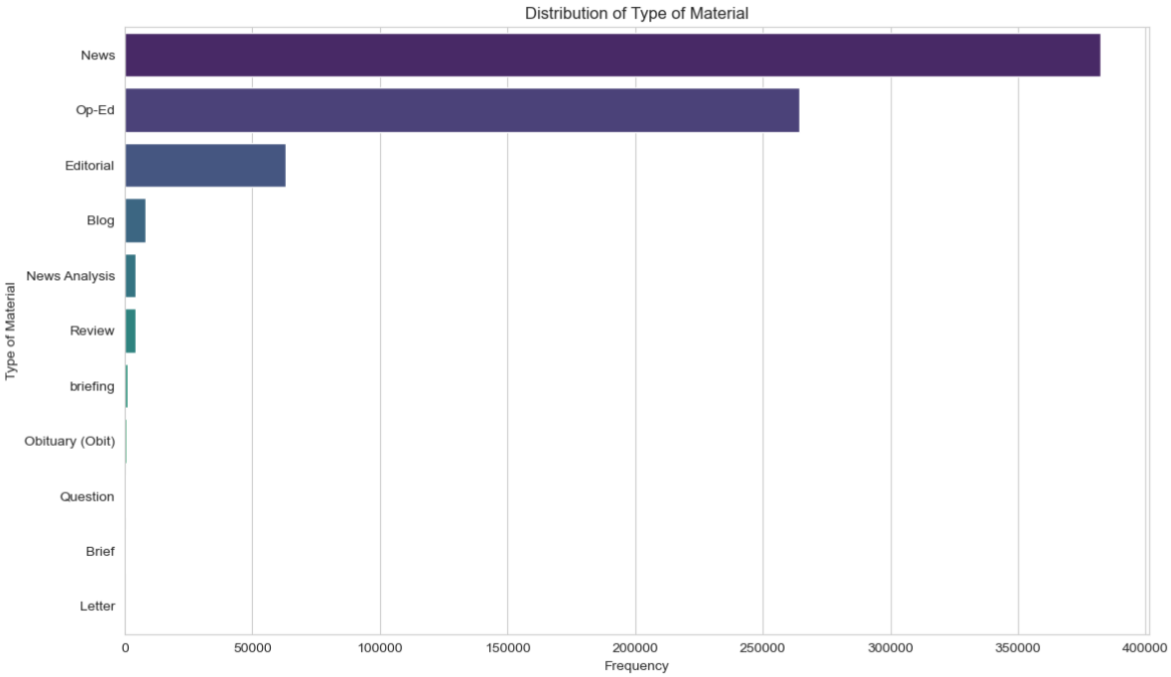
The distribution also reveals a tail extending towards longer word counts, indicating that a smaller proportion of articles are considerably more extensive. This may include in-depth analyses, comprehensive research papers, or articles that delve into complex topics.



The provided image depicts a bar graph that compares the average number of recommended articles for two different document types: articles and blog posts.

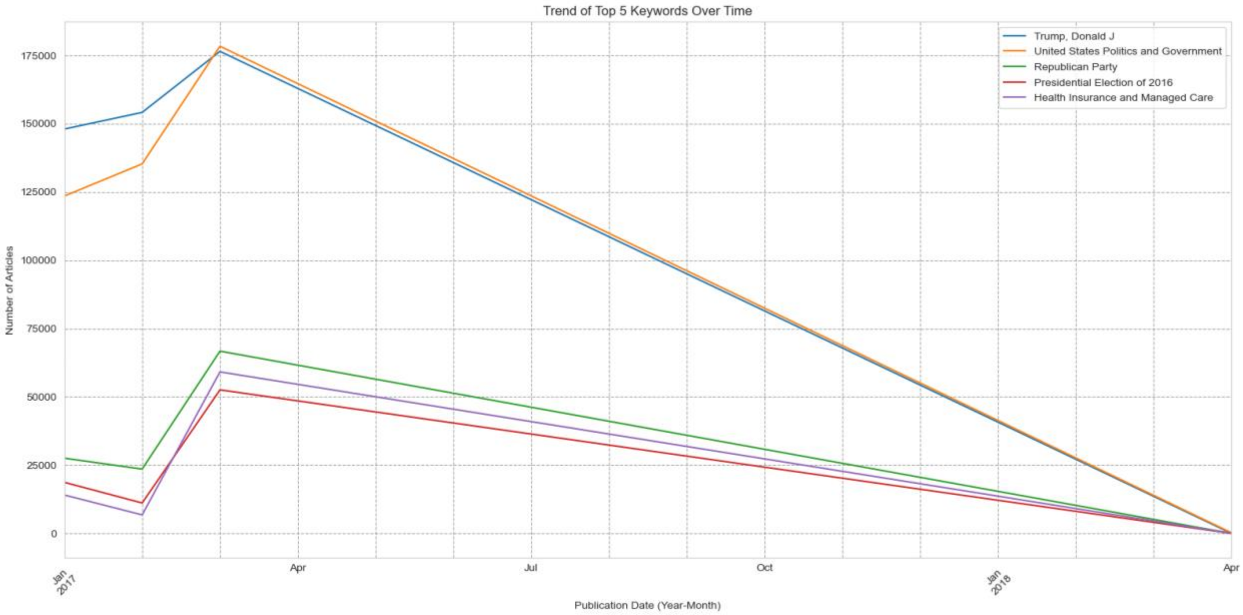
The graph shows that articles, on average, receive more recommendations than blog posts. Specifically, articles receive an average of 20 recommendations, while blog posts receive an average of 15 recommendations. This suggests that articles are generally considered to be more valuable and informative than blog posts and are therefore more likely to be recommended to users.

The graph also reveals that the distribution of recommendations for both article and blog post types is consistent. There is a slight skew towards the lower values, indicating that most recommendations fall within the 10-20 range. However, there is also a small number of outliers with higher recommendation counts, suggesting that some articles and blog posts are particularly popular and receive a high number of recommendations.

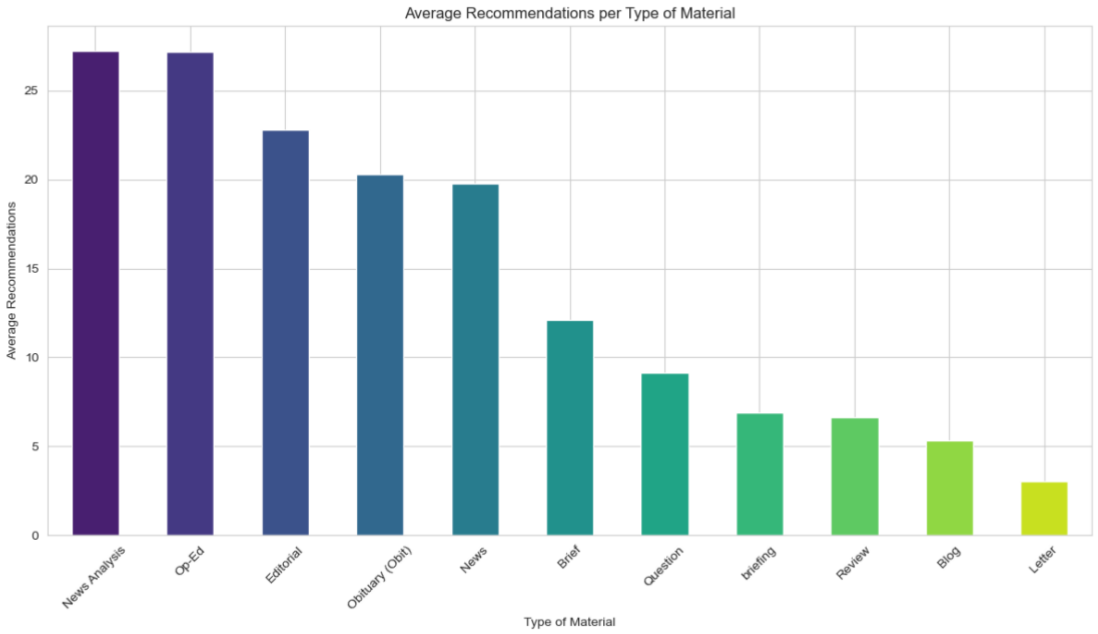


The provided visual representation highlights the distribution of various types of materials. Notably, "News" content takes the lead in terms of frequency, surpassing all other content categories.

Within this spectrum, "Op-ed" which is characterized as an essay in a newspaper or magazine conveying the writer's opinion and typically authored by someone external to the publication, is a substantial category with an approximate frequency of 25,000. Following closely behind are "Editorial" pieces, and a diverse array of other content types, including blogs, reviews, briefings, obituaries, questions, and letters, among others, collectively contributing to the overall content distribution.



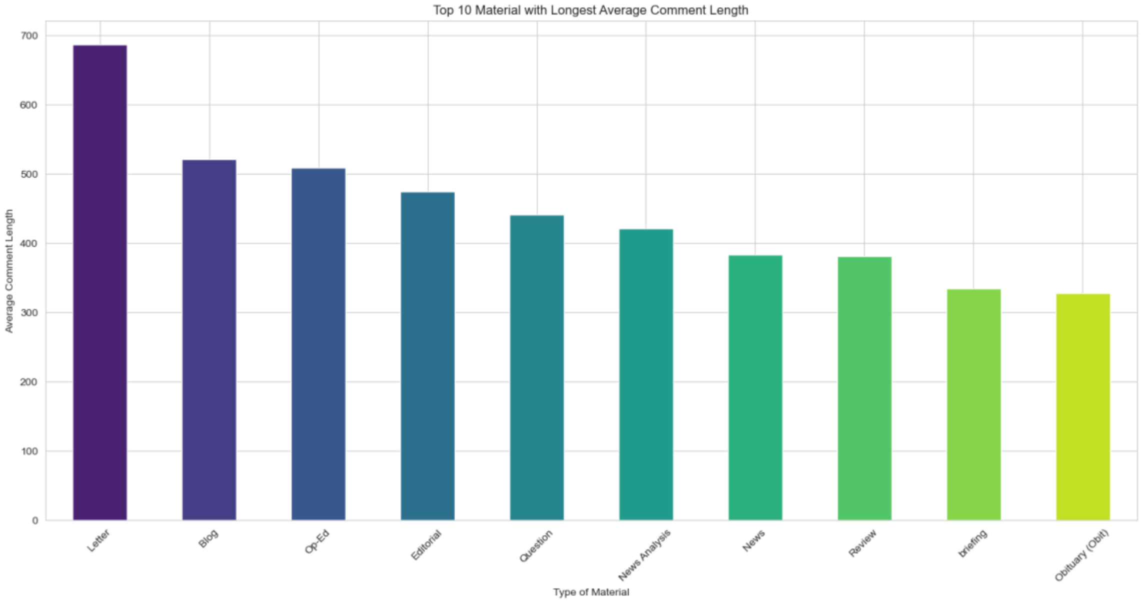
The graph shows the trend of the top 5 keywords over time. The keywords are Trump, Donald J, United States Politics and Government, Republican Party, Presidential Election of 2016, and Health Insurance and Managed Care. The graph shows that the search volume for these keywords peaked in March 2017, post the presidential election. The search volume for the keyword "Trump" was particularly high, reaching a peak of over 175,000 searches per day. The search volume for the other keywords also increased significantly during the election, but not as much as the volume for the keyword "Trump".



The figure above depicts a bar graph that represents the average recommendation per type of material for different types of materials. The graph indicates that news analysis receives the highest average recommendation, followed by op-ed, editorial, obituary, news brief, question and briefing, review, blog, and letter.

News analysis is generally in-depth and objective evaluations of current events, offering insights and perspectives on the significance and implications of happenings. Op-eds, on the other hand, are opinion pieces that express personal viewpoints or commentary on current affairs. Editorials, typically representing the stance of the publication, offer informed commentary and analysis on important issues. Obituaries provide tributes to deceased individuals, highlighting their accomplishments and contributions to society. News briefs summarize key points of news articles in concise form. Question and briefing materials, often in the form of question-and-answer sessions or interviews, provide in-depth information about specific topics. Reviews assess the merits or drawbacks of products, services, or experiences. Blogs offer personal reflections, commentary, and analyses on various topics. Letters, whether written to editors or individuals, express personal opinions or provide feedback.

This graph suggests that news analysis, op-eds, editorials, obituaries, and question and briefing materials tend to receive more recommendations compared to other material types. This may be attributed to their informative, thought-provoking, or timely nature, appealing to a broader audience.

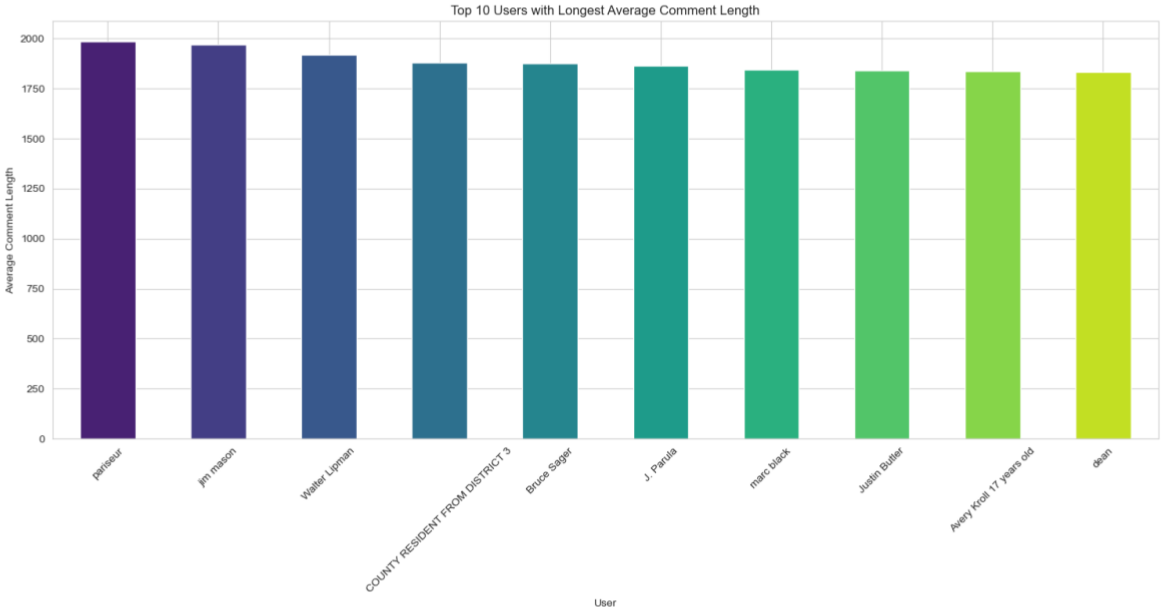


The bar graph illustrated compares the average comment length for different types of materials. The graph indicates that letters have the longest average comment length, followed by blogs, op-eds, editorials, questions, news analysis, news, reviews, briefings, and obituaries.

Letters generally convey personal experiences, thoughts, or opinions, fostering deeper engagement and discussion. Blogs also encourage open dialogue and commentary on various topics. Op-eds and editorials, with their strong opinions and stances on current issues, often spark debates and discussions.

The graph suggests that materials that evoke personal connection, evoke strong opinions, or delve into in-depth analysis tend to generate longer comment threads. This could be attributed to the inherent curiosity and engagement that these types of content inspire among readers.

Overall, the graph highlights the varying levels of engagement elicited by different types of materials. Understanding this pattern can be useful for content creators and publishers to tailor their strategies to optimize audience interaction and discussion.



The graph depicts the average comment length for the top 10 users on a website with the longest average comment length. The graph indicates that the top 10 users have average comment lengths ranging from 750 characters to 2000 characters. The user with the longest average comment length is *pariseur*, with a comment length of 2000 characters. The user with the shortest average comment length is Avery Kroll 17 years old, with a comment length of 500 characters.

This graph suggests that some users on the website are more likely to leave long comments than others. This could be due to a number of factors, such as the user's level of engagement with the website, their interest in the topics being discussed, or their personality.

**Task 3:**

The data set is rich in information containing comments' texts, that are largely very well written, along with contextual information such as section/topic of the article, as well as features indicating how well the comment was received by the readers such as ***editorsSelection*** and ***recommendations***. This data can serve the purpose of understanding and analyzing the public mood. Our main goal is to predict how deeply a user will engage with a new article. To achieve this, we're leveraging their previous interactions with a variety of articles as valuable insights into their preferences and behavior.

Few observations to be made on this dataset are:

1. **Article Count Dominance**:

    - The dataset reveals a notable prevalence of articles over other document types, such as blog posts. This prominence may stem from diverse factors, including the formality and purpose inherent in articles, the targeted audience, the platform's influence, and potential overlaps in document types.
    - These varied elements contribute to the higher count of articles, highlighting the dataset's richness in capturing a wide spectrum of textual content.

2. **Peak Engagement Amidst Presidential Elections**:

    - A discernible spike in user engagement is evident during the period encompassing the Presidential Elections. Notably, the keyword 'Trump' emerges as the most frequently used term during this heightened engagement phase.
    - This observation underscores the dataset's responsiveness to major political events, emphasizing the strong connection between topical relevance, user interest, and interaction levels. Understanding these spikes can provide valuable insights into user behavior during pivotal moments in public affairs.

Our problem is as intriguing as it is challenging: Can we predict the engagement rate of users based on their past behavior within the comments section? Traditional analytics have skirted around the edges of this question, focusing on the more immediate, visible layers of data. However, we've observed through exploratory data analysis (EDA) that there's a deeper narrative woven into the fabric of these digital discussions. Patterns emerge, trends become apparent, and the potential to forecast engagement from this rich tapestry of interaction becomes a tangible goal.

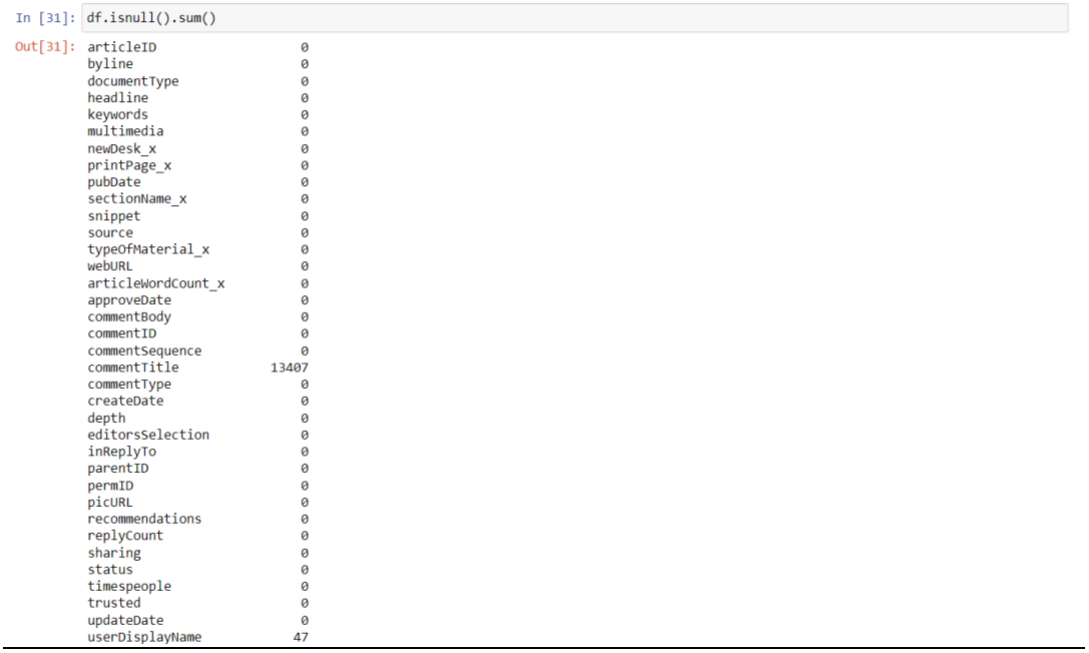
Through EDA, we've unearthed compelling observations. We've seen how certain topics drive more interaction, how the time of day influences the quantity and quality of comments, and how the ebb and flow of the news cycle plays a crucial role in user participation. These insights have not only justified our choice but have also fueled our motivation. They underscore the significance of our endeavor, highlighting the potential benefits not just for The New York Times but for the broader landscape of digital journalism and audience engagement.

The impact of our work promises to be profound. With a successful predictive model, content creators can craft stories that resonate more deeply with their audience. Editors can foster a more robust and engaging community, and strategists can make informed decisions that help adapt to the ever-changing tides of public discourse.

**Task 4:**

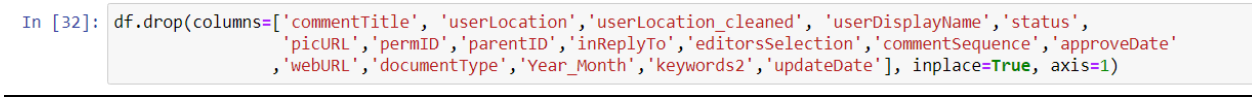
**Data Preprocessing**

1. **Data Cleaning:**



```
In [31]: df.isnull().sum()

Out[31]: articleID                0
         byline                   0
         documentType             0
         headline                 0
         keywords                 0
         multimedia               0
         newDesk_x                0
         printPage_x              0
         pubDate                  0
         sectionName_x            0
         snippet                  0
         source                   0
         typeOfMaterial_x         0
         webURL                   0
         articleWordCount_x       0
         approveDate              0
         commentBody              0
         commentID                0
         commentSequence          0
         commentTitle         13407
         commentType              0
         createDate               0
         depth                    0
         editorsSelection         0
         inReplyTo                0
         parentID                 0
         permID                   0
         picURL                   0
         recommendations          0
         replyCount               0
         sharing                  0
         status                   0
         timespeople              0
         trusted                  0
         updateDate               0
         userDisplayName         47
```



```
In [32]: df.drop(columns=['commentTitle', 'userLocation','userLocation_cleaned', 'userDisplayName','status',
                          'picURL','permID','parentID','inReplyTo','editorsSelection','commentSequence','approveDate'
                         ,'webURL','documentType','Year_Month','keywords2','updateDate'], inplace=True, axis=1)
```

After eliminating *commentTitle*, *userLocation_cleaned*, *userDisplayName*, *userLocation* as well as the status, permID, parentID and other irrelevant columns, our dataset is now streamlined to 25 columns. We opted to remove the columns with null values, since it has many missing values. Moreover, many activities shared identical orientation values, which could potentially mislead our models.

- Addressing Data Inconsistencies

Initially, we carefully scrutinized our dataset for completeness and consistency. During this stage, columns with excessive missing values were eliminated. This decision was predicated on the understanding that such columns would not contribute meaningfully to predictive accuracy and could, in fact, introduce bias. Moreover, we excised redundant columns that exhibited no significant correlation with our target variable to streamline the dataset and focus on high-impact data.

Data Transformation & Normalization

To better comprehend the interaction between the timing of article publication and user comments, we engineered a new feature: 'Average Comment Hour'. This transformation is key to unlocking potential patterns in user engagement across different times of the day. Moreover, we initiated a process of normalization for our continuous variables, ensuring that all features contributed equally to the analysis and model training. This negates the disproportionate influence that numerically larger features could exert on the learning algorithm.

3. Specific Data Type Processing:

a. Text Data Processing:

Our text data underwent a rigorous cleansing process. We constructed a function that not only lowercases all text but also purges punctuation and digits. This was a strategic pivot from our initial intention to include numerical data, which we later deemed superfluous noise. Subsequently, we tokenized words, expunged common stopwords—those offering little to no semantic weight—and applied lemmatization to consolidate different inflected forms of a word. This meticulous refinement is aimed at producing a more structured and meaningful set of textual features for subsequent analysis.

b. Engagement Rate Calculation:

We established a weighted scoring system to quantify user engagement based on various interaction metrics, such as comment length, sharing frequency, reply count, and comment depth. This composite score offers a nuanced view of engagement, capturing not just the quantity but the quality of user interactions with content.

c. Polarity Columns:

By computing the sentiment polarity of comments and article snippets, we seek to understand the sentiment bias in user engagement. This is grounded in the hypothesis that a user's historical sentiment patterns could significantly influence their engagement with certain topics, thereby allowing us to predict future interactions more accurately.

d. Deduplication:

Duplicate data points, an unintended consequence of feature engineering, were identified and removed to ensure the integrity of our dataset. This step is crucial to prevent overfitting and to preserve the model's ability to generalize from our data.

4. Pre-Modeling Data Preparation:

a. Data Splitting:

We adhered to a stratified split of 80/10/10 for our train, validation, and test sets, respectively. This strategy ensures a fair distribution of data across all subsets and helps prevent information leakage, particularly when applying techniques such as TF-IDF vectorization for text features.

b. Feature Encoding & Dimensionality Reduction:

We employed binary encoding for categorical variables to reduce dimensionality while maintaining essential information. Furthermore, we leveraged techniques like Truncated SVD and PCA, which are indispensable for managing high-dimensional data and expediting computational performance.

c. Standardization and Integration:

Standardizing our dense feature set facilitated model training by aligning the data onto a common scale. Thereafter, we fused our dense and sparse matrices, creating a harmonized input space for the learning algorithm.

d. Target Scaling:

The target variable underwent min-max scaling to ensure that our model's evaluation metrics, such as Mean Squared Error (MSE), could be interpreted on a consistent and intuitive scale.

By enriching the explanation of each preprocessing step, we lend greater transparency to the methodology, justifying the rationale behind our decisions and setting the stage for more reliable model training and evaluation.

**Task 5 & 6:**

Our model development journey commenced with a preselection of eight distinctive models, chosen with meticulous consideration for both their complexity and computational efficiency. We initiated this process with a foundational baseline model, which served as a critical benchmark for subsequent evaluations. In addition to the baseline model, our ensemble included the following predictive algorithms: Linear Regression, Ridge Regression, Elastic Net Regression, Stochastic Gradient Descent (SGD) Regressor, Decision Tree, Random Forest, XGBoost, and a Feedforward Neural Network, specifically a Multilayer Perceptron (MLP).

The rationale behind our selection was to ensure a diverse representation of both linear and non-linear models, intentionally omitting more computationally demanding alternatives like K-Nearest Neighbors (KNN) or Support Vector Regression (SVR) to maintain a focus on efficiency.

Our initial experimentation was confined to the Linear Regression Model, valued for its computational speed and interpretability. The aim was to dissect the intricate relationship between the model's performance and the number of dimensions yielded by the TF-IDF vectorization, subsequently reduced through Principal Component Analysis (PCA) and Truncated Singular Value Decomposition (SVD). This experiment was rooted in the quest to discover an optimal equilibrium - a point where we could minimize the dimensionality without a substantial sacrifice in model accuracy, thereby addressing the inherent trade-off between performance and computational expense. As dimensions proliferate, so typically does model performance, yet at the cost of increased computational resources.

For the empirical investigation, we evaluated the model's performance over three months' worth of data comprising comments and articles. The dataset underwent a TF-IDF transformation, followed by dimensionality reduction through PCA or Truncated SVD. The resulting Mean Squared Error (MSE) was our metric of choice for gauging performance.

The outcomes from our meticulous trials are summarized below:

- With the maximum TF-IDF features set to 1500, and after reducing dimensionality to 50, the model produced an MSE of 0.033589206176013006.
- Escalating the TF-IDF features to 3000 (3*1000) with a dimensional reduction to 100 slightly improved the MSE to 0.03344259310082224.
- Further incrementing TF-IDF features to 4500 (3*1500) and reducing dimensions to 250 continued the trend of improvement, yielding an MSE of 0.03330188485513363.
- At a TF-IDF feature count of 7500 (3*2500) and dimensionality truncated to 350, the model achieved an MSE of 0.03320980426994299, indicating a pattern where increased features paired with optimal dimensionality reduction enhanced performance.
- Interestingly, a return to 3000 TF-IDF features with the same dimensionality reduction to 350 demonstrated a comparable MSE of 0.03321426505951219, suggesting a possible plateau in performance gains relative to dimensionality settings.

Conversely, our analysis also illuminated that overextension in the number of features to 10500 (3*3500), with the same dimensional reduction, did not translate to significant performance gains, as evidenced by an MSE of 0.03324902820483597. However, when dimensionality was further reduced to 450 and then 700, the MSE marginally decreased, hinting at a nuanced interplay between feature count and dimensional reduction.

An interesting case emerged when applying PCA to a TF-IDF feature count of 10500 with reduced dimensions to 700, the MSE observed was 0.03309416720423065. When compared to the PCA's application to the same number of features but with a dimensionality reduction to 100, the MSE deteriorated to 0.03354232817921276. This demonstrates that PCA's effectiveness is contingent upon the interrelationship between the original feature space and the reduced dimensional space.

For a more computationally manageable subset of data—comments and articles from January 2017 alone—a TF-IDF feature set of 1500 (3*500) and a dimensionality reduction to 100 post-PCA application, a promising MSE of 0.032 was recorded.

Interpreting these results, we deduced a nuanced balance between feature richness and dimensionality reduction that is crucial for optimizing model performance while maintaining computational feasibility. The linear regression analysis has elucidated that beyond a certain threshold, increasing the dimensionality does not guarantee a proportional enhancement in model accuracy, and may indeed reach a plateau. This plateau suggests an optimal feature count and dimensional space where the model performs best before the law of diminishing returns sets in.

Building upon the aforementioned analysis, we ventured into a series of trial and error experiments with an objective to probe the limits of dimensionality reduction. Our goal was to ascertain whether we could markedly decrease the number of dimensions without incurring a significant penalty on model performance. To this end, we aggressively trimmed down the feature dimensions to a mere 10, applying PCA to condense them further into 9 principal features that continued to encapsulate the essence of the text data.

Remarkably, the outcome of this bold reduction was counterintuitive: the performance of the Linear Regression model not only remained robust but exhibited a slight improvement. This observation introduces a compelling argument against the common presumption that model efficacy is invariably tied to the richness of features. It suggests that, beyond a certain point, the quality of features—how well they capture the underlying patterns—can be more impactful than their quantity.

However, it's imperative to exercise caution in generalizing this result. The unique behavior of Linear Regression in this context may not mirror the performance of other models under similar conditions. Each algorithm has its own architectural nuances and might respond differently to such a drastic reduction in dimensionality. Therefore, while the enhanced performance with minimal features provides an intriguing precedent, it cannot be taken as a universal predictor of success across all models.

Nonetheless, this discovery provides a valuable point of departure as we pivot towards experimenting with other models in our selection. It empowers us to explore and possibly employ other techniques to amplify model performance, instead of focusing solely on the feature count. This could entail leveraging ensemble methods, feature engineering, or advanced regularization techniques that may help in accentuating model accuracy.

As we proceed, this newfound insight will serve as a guiding principle, informing our strategy for model training and optimization. It encourages a more holistic approach that weighs the intricacies of each model against the backdrop of computational efficiency and model effectiveness, setting the stage for sophisticated and streamlined machine learning solutions.

**Pre-processing step: Results of TF-IDF (3*500), TruncatedSVD (10 components). And PCA (90%):**

Baseline Model: 0.03422904657706901

Linear Regression: 0.03308574699356936

Ridge Regression: 0.033085712521023415

Elastic Net Regression: 0.03422904657706901

SGD Regressor: 0.033286184111119

Decision Tree: 0.04626249202187169

Random Forest: 0.03012410693967352

XGBoost Regressor: 0.029659867563546858

Feedforward Neural Network (MLP): 0.028664758243256174

Ensemble methods and the neural network outperformed the simpler linear models and the baseline, indicating that our dataset likely benefits from models capable of capturing more complex patterns. The Decision Tree model might be too simple or not well-tuned compared to the ensemble methods that aggregate many trees. The regularization in Ridge Regression and Elastic Net didn't yield improvements over basic Linear Regression, which might imply that feature selection or model complexity isn't the main issue with the simpler models.

**Boosting the performance:**

**Word2Vec (Pre-processing step):**

Linear Regression:

Mean Absolute Error (MAE): 0.14137755765970597

Mean Squared Error (MSE): 0.03319878302384105

Root Mean Squared Error (RMSE): 0.1822053320400944

R-squared (R2): 0.029948272477434923

Ridge Regression:

Mean Absolute Error (MAE): 0.14137760839692687

Mean Squared Error (MSE): 0.033198781093278895

Root Mean Squared Error (RMSE): 0.18220532674232906

R-squared (R2): 0.02994832888748933

Elastic net Regression:

Mean Absolute Error (MAE): 0.14426567323800496

Mean Squared Error (MSE): 0.03422904657706901

Root Mean Squared Error (RMSE): 0.18501093637152646

R-squared (R2): -0.0001555099080401412

SGD Regressor:

Mean Absolute Error (MAE): 0.14073779170285694

Mean Squared Error (MSE): 0.033302759489739674

Root Mean Squared Error (RMSE): 0.18249043670762496

R-squared (R2): 0.02691013248615215

Decision Tree:

Mean Absolute Error (MAE): 0.15547635173148502

Mean Squared Error (MSE): 0.047980302360818475

Root Mean Squared Error (RMSE): 0.2190440648838002

R-squared (R2): -0.4019602814579908

Random Forest:

Mean Absolute Error (MAE): 0.1261835880812586

Mean Squared Error (MSE): 0.030005859705672933

Root Mean Squared Error (RMSE): 0.1732219954442072

R-squared (R2): 0.1232438845006768

XGBoost:

Mean Absolute Error (MAE): 0.1348576615936283

Mean Squared Error (MSE): 0.030430841070656613

Root Mean Squared Error (RMSE): 0.17444437815721267

R-squared (R2): 0.11082614295360671

Feedforward Neural Network (MLP):

Mean Absolute Error (MAE): 0.12915807115680142

Mean Squared Error (MSE): 0.02798258874373174

Root Mean Squared Error (RMSE): 0.16727997113740706

R-squared (R2): 0.18236284348383713

- Linear Regression and Ridge Regression show very similar results across all metrics, with a low R-squared value, indicating limited explanatory power of the model.
- Elastic Net Regression has slightly higher errors and a negative R-squared value, suggesting a poor fit.
- The SGD Regressor has marginally lower errors than Linear and Ridge Regression but still shows a low R-squared value.
- The Decision Tree model shows significantly higher errors and a negative R-squared value, which is worse than the baseline, indicating a poor model fit.
- Random Forest and XGBoost models show improved error metrics and positive R-squared values, suggesting better performance and model fit than the previously mentioned models.
- The Feedforward Neural Network (MLP) has the lowest errors and the highest R-squared value among the models, indicating the best performance in terms of the metrics provided.

**GloVe (Pre-processing step):**

Linear Regression:

Mean Absolute Error (MAE): 0.14125807455206588

Mean Squared Error (MSE): 0.03316457219994881

Root Mean Squared Error (RMSE): 0.182111427977348

R-squared (R2): 0.030947895529665703

Ridge Regression:

Mean Absolute Error (MAE): 0.14125805208695438

Mean Squared Error (MSE): 0.033164544329586376

Root Mean Squared Error (RMSE): 0.18211135145725094

R-squared (R2): 0.03094870988762588

Elastic net Regression:

Mean Absolute Error (MAE): 0.14426567323800496

Mean Squared Error (MSE): 0.03422904657706901

Root Mean Squared Error (RMSE): 0.18501093637152646

R-squared (R2): -0.0001555099080401412

SGD Regressor:

Mean Absolute Error (MAE): 0.13960120257534545

Mean Squared Error (MSE): 0.033235681588011344

Root Mean Squared Error (RMSE): 0.18230655936639073

R-squared (R2): 0.02887011500730141

Decision Tree:

Mean Absolute Error (MAE): 0.15307126930775317

Mean Squared Error (MSE): 0.046652195784355345

Root Mean Squared Error (RMSE): 0.21599119376575365

R-squared (R2): -0.3631536758692573

Random Forest:

Mean Absolute Error (MAE): 0.12573372387995654

Mean Squared Error (MSE): 0.02985875852440708

Root Mean Squared Error (RMSE): 0.17279687070200977

R-squared (R2): 0.12754210696579182

XGBoost Regressor:

Mean Absolute Error (MAE): 0.13477561949615996

Mean Squared Error (MSE): 0.030398423709320586

Root Mean Squared Error (RMSE): 0.1743514373594912

R-squared (R2): 0.11177336193278298

Feedforward Neural Network (MLP):

Mean Absolute Error (MAE): 0.12606680737022838

Mean Squared Error (MSE): 0.0280717564155882

Root Mean Squared Error (RMSE): 0.16754628141378788

R-squared (R2): 0.1797574089996451

- Linear Regression and Ridge Regression have almost identical error metrics, with a slight improvement in R-squared compared to the Word2Vec pre-processing, indicating a small increase in the models' explanatory power.
- Elastic Net Regression has unchanged errors from the Word2Vec results and a negative R-squared value, suggesting it does not fit this dataset well.
- The SGD Regressor shows a slight improvement in error metrics over Linear and Ridge Regression, but its R-squared value remains low.
- The Decision Tree has high error metrics and a substantially negative R-squared value, which implies a poor fit and possible overfitting.
- Random Forest displays lower errors and a higher R-squared value than the simpler models, indicating better predictive performance.
- The XGBoost Regressor also shows lower errors and a positive R-squared value, suggesting it performs well, though not as well as the Random Forest.
- The Feedforward Neural Network (MLP) has the lowest MAE and MSE, with a substantially positive R-squared value, making it the best-performing model among those tested with GloVe pre-processing.

**Dimensionality Reduction Technique (Sparse Random Projection) without PCA:**

Linear Regression:

Mean Absolute Error (MAE): 0.14128765160804224

Mean Squared Error (MSE): 0.03315355984691981

Root Mean Squared Error (RMSE): 0.18208119026115743

R-squared (R2): 0.03126967094149913

Ridge Regression:

```
Mean Absolute Error (MAE): 0.14128774284413417
Mean Squared Error (MSE): 0.03315358871526659
Root Mean Squared Error (RMSE): 0.18208126953442133
R-squared (R2): 0.031268827422938905

Elastic Net Regression:
Mean Absolute Error (MAE): 0.14426567323800496
Mean Squared Error (MSE): 0.03422904657706901
Root Mean Squared Error (RMSE): 0.18501093637152646
R-squared (R2): -0.0001555099080401412

SGD Regressor:
Mean Absolute Error (MAE): 0.14124403872683167
Mean Squared Error (MSE): 0.03328588387595303
Root Mean Squared Error (RMSE): 0.18244419386747562
R-squared (R2): 0.02740322942572926

Decision Tree:
Mean Absolute Error (MAE): 0.15347900508445933
Mean Squared Error (MSE): 0.04694106193118954
Root Mean Squared Error (RMSE): 0.21665886072623372
R-squared (R2): -0.3715941992630838

Random Forest:
Mean Absolute Error (MAE): 0.12555134314185296
Mean Squared Error (MSE): 0.02971530269515182
Root Mean Squared Error (RMSE): 0.17238127130042816
```

R-squared (R2): 0.13173381408024565

XGBoost Regressor:

Mean Absolute Error (MAE): 0.1348644946660921

Mean Squared Error (MSE): 0.0304485141101609

Root Mean Squared Error (RMSE): 0.17449502603272363

R-squared (R2): 0.11030974563598817

Linear Regression and Ridge Regression have virtually identical error metrics, with a slight improvement in R-squared, indicating a modest increase in the proportion of variance explained by the models.

Elastic Net Regression shows no improvement compared to the previous two pre-processing techniques, with the same error metrics and a non-positive R-squared, indicating a poor fit for the data.

The SGD Regressor has a slight increase in the Mean Squared Error and a decrease in R-squared compared to Linear and Ridge Regression, suggesting it's not capturing the data's variance as effectively.

The Decision Tree model has high error metrics and a substantially negative R-squared value, again indicating a poor model fit and possibly overfitting or not capturing the underlying patterns in the data.

Random Forest shows a lower error across all metrics and the highest R-squared value among the models tested with this technique, suggesting it's the most effective at predicting the target variable.

The XGBoost Regressor also performs well, with error metrics only slightly higher than Random Forest, and a positive R-squared, but not quite as high, indicating good but slightly inferior predictive performance compared to the Random Forest model.

**Preprocessing: Removing Encoded columns:**

Linear Regression:

Mean Absolute Error (MAE): 0.1431442081149577

Mean Squared Error (MSE): 0.03385009972023313

Root Mean Squared Error (RMSE): 0.18398396593245056

R-squared (R2): 0.010917126484953688

Ridge Regression:

Mean Absolute Error (MAE): 0.1413653452196044

Mean Squared Error (MSE): 0.03319228626814251

Root Mean Squared Error (RMSE): 0.182187503051506

R-squared (R2): 0.030138104408446886

Elastic net Regression:

Mean Absolute Error (MAE): 0.14426567323800496

Mean Squared Error (MSE): 0.03422904657706901

Root Mean Squared Error (RMSE): 0.18501093637152646

R-squared (R2): -0.0001555099080401412

SGD Regressor:

Mean Absolute Error (MAE): 0.14277858078929356

Mean Squared Error (MSE): 0.033903888021993435

Root Mean Squared Error (RMSE): 0.18413008451090612

R-squared (R2): 0.009345459384818677

Decision Tree:

Mean Absolute Error (MAE): 0.15276496538907697

Mean Squared Error (MSE): 0.04636376028440191

Root Mean Squared Error (RMSE): 0.21532245652602497

R-squared (R2): -0.35472573576049693

Random Forest:

Mean Absolute Error (MAE): 0.12556755421095642

Mean Squared Error (MSE): 0.029839505798047207

Root Mean Squared Error (RMSE): 0.17274115258978448

R-squared (R2): 0.1281046619381081

XGBoost Regressor:

Mean Absolute Error (MAE): 0.13321878010416238

Mean Squared Error (MSE): 0.029836170945922322

Root Mean Squared Error (RMSE): 0.17273149957643025

R-squared (R2): 0.12820210463841597

Feedforward Neural Network (MLP):

Mean Absolute Error (MAE): 0.13495099544956976

Mean Squared Error (MSE): 0.029918923734567757

Root Mean Squared Error (RMSE): 0.17297087539400313

R-squared (R2): 0.12578410981236565

- Linear Regression shows increased errors and a very low R-squared value, indicating minimal predictive accuracy.
- Ridge Regression offers a slight improvement over Linear Regression with slightly better error metrics and a marginally higher R-squared value, suggesting a small improvement in fit.
- Elastic Net Regression remains consistent with its prior performance, with unchanged error metrics and a negative R-squared value, continuing to suggest a poor fit.
- The SGD Regressor shows increased errors and a low R-squared value similar to Linear Regression, indicating a similarly poor predictive performance.
- The Decision Tree has high error metrics and a significantly negative R-squared value, suggesting it is performing worse than random chance and may be highly overfitting to the data.
- Random Forest shows the lowest error metrics among the models and a positive R-squared value, indicating it has the best fit among the tested models.
- The XGBoost Regressor also performs well, with error metrics and an R-squared value close to those of the Random Forest, indicating good predictive performance.

- The Feedforward Neural Network (MLP) has slightly higher errors than the Random Forest and XGBoost but still maintains a positive R-squared value, suggesting decent model performance but not quite as strong as the two ensemble methods.

**Pre-processing step-Word2Vec (300 dimensions):**

Linear Regression:

Mean Absolute Error (MAE): 0.14136538025640746

Mean Squared Error (MSE): 0.03319232142327917

Root Mean Squared Error (RMSE): 0.18218759953212835

R-squared (R2): 0.03013707719305092

Ridge Regression:

Mean Absolute Error (MAE): 0.1413653452196044

Mean Squared Error (MSE): 0.03319228626814251

Root Mean Squared Error (RMSE): 0.182187503051506

R-squared (R2): 0.030138104408446886

Elastic net Regression:

Mean Absolute Error (MAE): 0.14426567323800496

Mean Squared Error (MSE): 0.03422904657706901

Root Mean Squared Error (RMSE): 0.18501093637152646

R-squared (R2): -0.0001555099080401412

SGD Regressor:

Mean Absolute Error (MAE): 0.14148805424991828

Mean Squared Error (MSE): 0.03333325241681559

Root Mean Squared Error (RMSE): 0.18257396423591069

R-squared (R2): 0.026019144507286374

Decision Tree:

Mean Absolute Error (MAE): 0.15324394320576287

Mean Squared Error (MSE): 0.04702746314253031

Root Mean Squared Error (RMSE): 0.21685816365202928

R-squared (R2): -0.37411879916365764

Random Forest:

Mean Absolute Error (MAE): 0.12495210760622787

Mean Squared Error (MSE): 0.029458013293024724

Root Mean Squared Error (RMSE): 0.17163336882152236

R-squared (R2): 0.13925168088962248

XGBoost Regressor:

Mean Absolute Error (MAE): 0.13484059214645316

Mean Squared Error (MSE): 0.030416561570892647

Root Mean Squared Error (RMSE): 0.17440344483665637

R-squared (R2): 0.1112433827483379

Feedforward Neural Network (MLP):

Mean Absolute Error (MAE): 0.13512788462147182

Mean Squared Error (MSE): 0.02940967904710448

Root Mean Squared Error (RMSE): 0.17149250434670454

R-squared (R2): 0.14066398322371

- Linear Regression and Ridge Regression have nearly identical error metrics and a low R-squared value, suggesting only a small portion of the variance in the data is being captured by these models.

- Elastic Net Regression consistently shows no improvement, with error metrics similar to previous results and a negative R-squared value, indicating a poor fit.
- The SGD Regressor has slightly higher errors compared to Linear and Ridge Regression and a low R-squared value, indicating a weaker fit to the data.
- The Decision Tree model performs poorly with high error metrics and a significantly negative R-squared value, suggesting that it might be overfitting the data.
- Random Forest shows the best performance with the lowest error metrics and the highest R-squared value, indicating it is the most robust model for the data.
- The XGBoost Regressor also has good performance, with slightly higher errors than the Random Forest but still maintaining a positive R-squared value, suggesting effective predictive capability.
- The Feedforward Neural Network (MLP) has low error metrics and a positive R-squared value comparable to the Random Forest, indicating strong performance and a good fit to the data.

**Word2Vec (300 dimensions) along with** Min/Max scaling:

Linear Regression:

Mean Absolute Error (MAE): 0.1413165237126753

Mean Squared Error (MSE): 0.033176441007742065

Root Mean Squared Error (RMSE): 0.1821440117262768

R-squared (R2): 0.03060109494076413

Ridge Regression:

Mean Absolute Error (MAE): 0.14131662309765294

Mean Squared Error (MSE): 0.033176455452253756

Root Mean Squared Error (RMSE): 0.18214405137762188

R-squared (R2): 0.030600672879404045

Elastic net Regression:

Mean Absolute Error (MAE): 0.14426567323800496

Mean Squared Error (MSE): 0.03422904657706901

```
Root Mean Squared Error (RMSE): 0.18501093637152646

R-squared (R2): -0.0001555099080401412


SGD Regressor:

Mean Absolute Error (MAE): 0.1409335156089808

Mean Squared Error (MSE): 0.03323671525628683

Root Mean Squared Error (RMSE): 0.18230939431715204

R-squared (R2): 0.028839911740645596


Decision Tree:

Mean Absolute Error (MAE): 0.1528177904596685

Mean Squared Error (MSE): 0.0466468754481455

Root Mean Squared Error (RMSE): 0.2159788773193932

R-squared (R2): -0.36299821832348966


Random Forest:

Mean Absolute Error (MAE): 0.1254066407814728

Mean Squared Error (MSE): 0.02982655747258012

Root Mean Squared Error (RMSE): 0.17270366953999594

R-squared (R2): 0.12848300549000435


XGBoost Regressor:

Mean Absolute Error (MAE): 0.13484408517736893

Mean Squared Error (MSE): 0.03044367522600166

Root Mean Squared Error (RMSE): 0.17448116008899545

R-squared (R2): 0.11045113539520257
```

Feedforward Neural Network (MLP):

Mean Absolute Error (MAE): 0.13676244728518389

Mean Squared Error (MSE): 0.03196969497564949

Root Mean Squared Error (RMSE): 0.17880071301773237

R-squared (R2): 0.06586160651649742

- Linear Regression and Ridge Regression present very close error metrics with a slight improvement in the R-squared value compared to the results without Min/Max scaling, indicating a modest enhancement in model fit.
- Elastic Net Regression has unchanged high error metrics and a negative R-squared value from previous runs, indicating it is not a good fit for the data.
- The SGD Regressor sees a minimal improvement in errors compared to Linear and Ridge Regression, but its R-squared value remains low, indicating a slightly better but still limited fit.
- The Decision Tree model exhibits high errors and a significantly negative R-squared value, suggesting it is not modeling the underlying data well and may be overfitting.
- Random Forest outperforms other models with the lowest error metrics and the highest R-squared value, signifying a robust performance and a good fit to the data.
- The XGBoost Regressor also has low error metrics and a positive R-squared value, indicating it performs well, though not as optimally as the Random Forest.
- The Feedforward Neural Network (MLP) shows higher error metrics and a reduced R-squared value compared to its performance without Min/Max scaling, suggesting that this data normalization may not be beneficial for the neural network in this specific case.

**Hyperparameter Tuning using GridSearchCV or Randomized SearchCV:**

Ridge Regression:

Mean Absolute Error (MAE): 0.14131877518067681

Mean Squared Error (MSE): 0.0331752567090597

Root Mean Squared Error (RMSE): 0.1821407607018805

R-squared (R2): 0.030635699552081008

Best Alpha: 100

```
Elastic Net Regression:
Mean Absolute Error (MAE): 0.1413211635298035
Mean Squared Error (MSE): 0.03317387897069658
Root Mean Squared Error (RMSE): 0.1821369785922029
R-squared (R2): 0.030675956373490854
(0.001, 0.01)

SGD Regressor:
Mean Absolute Error (MAE): 0.14218661784982625
Mean Squared Error (MSE): 0.03330335844096654
Root Mean Squared Error (RMSE): 0.1824920777485054
R-squared (R2): 0.02689263143281828
{'alpha': 0.0001, 'l1_ratio': 0.15, 'penalty': 'elasticnet'}
Decision Tree:
Mean Absolute Error (MAE): 0.11582533121660821
Mean Squared Error (MSE): 0.02478064254565098
Root Mean Squared Error (RMSE): 0.15741868550350363
R-squared (R2): 0.27592209951596325
{'max_depth': 10,
 'max_features': None,
 'min_samples_leaf': 2,
 'min_samples_split': 5}

Random Forest Regressor:
Mean Absolute Error (MAE): 0.12558428314331907
Mean Squared Error (MSE): 0.027272737751495708
```

Root Mean Squared Error (RMSE): 0.16514459649499802

R-squared (R2): 0.20310433213442014

```
{'n_estimators': 100,
 'min_samples_split': 10,
 'min_samples_leaf': 1,
 'max_features': 'sqrt',
 'max_depth': 10}
```

XGBoost Regressor:

Mean Absolute Error (MAE): 0.11514103454034047

Mean Squared Error (MSE): 0.02408457515145539

Root Mean Squared Error (RMSE): 0.1551920589187971

R-squared (R2): 0.29626083837053707

Best Parameters: {'subsample': 1.0, 'n_estimators': 200, 'max_depth': 9, 'learning_rate': 0.1, 'gamma': 0.1, 'colsample_bytree': 1.0}

Feedforward Neural Network (MLP):

Mean Absolute Error (MAE): 0.13029475698929924

Mean Squared Error (MSE): 0.0319006012659399

Root Mean Squared Error (RMSE): 0.17860739420847027

R-squared (R2): 0.06788048993209106

Best Parameters: {'activation': 'relu', 'alpha': 0.5505878444394529, 'batch_size': 98, 'hidden_layer_sizes': (50,), 'learning_rate_init': 0.1, 'solver': 'sgd'}

- Ridge Regression shows a slight improvement in all error metrics and R-squared value after tuning. The best alpha parameter found is 100, suggesting that a higher penalty on the coefficients aids in reducing overfitting and improving the model's prediction.
- Elastic Net Regression also shows a minor enhancement in error metrics and the R-squared value, with the best parameters indicating a combination of L1 and L2 regularization strengths at (0.001, 0.01).

- The SGD Regressor doesn't show significant improvement post-tuning with its error metrics and R-squared remaining similar to previous models. The best parameters indicate a preference for a mix of L1 and L2 regularization with a small alpha value.
- The Decision Tree model displays a substantial improvement with the lowest MAE and RMSE among all models and a notably higher R-squared value, indicating that hyperparameter tuning has significantly enhanced the model's fit to the data. The best parameters suggest a relatively shallow tree with constraints on leaf and split samples, preventing overfitting.
- Random Forest Regressor also improves, with reduced error metrics and a higher R-squared value compared to its untuned counterpart. The best parameters indicate a preference for a limited number of features to consider at each split and a shallow tree depth, which helps in generalizing the model.
- The XGBoost Regressor shows the most significant improvement, with the lowest errors and the highest R-squared value, indicating a very good fit to the data. The best parameters suggest a high number of estimators and a subsample ratio of 1, indicating that using the full sample for each tree and a moderate tree depth helps in achieving a robust model.
- The Feedforward Neural Network (MLP) shows moderate error metrics and an R-squared value that is improved from the non-tuned version but not as high as the ensemble methods. The best parameters suggest using the ReLU activation function and a specific configuration for the network's architecture and learning rate.

**Cross-Validation After tuning:**

Linear Regression:

Cross-validation MSE scores: [0.03369283 0.03378097 0.03424263 0.03485176 0.03388293]

Mean MSE: 0.03409022376327775

Standard Deviation of MSE: 0.0004241907563055569

Mean RMSE: 0.1846318267773916

Standard Deviation of RMSE: 0.0011455584319571604

Ridge Regression:

Cross-validation MSE scores: [0.03369201 0.03378086 0.03424427 0.03485175 0.03388186]

Mean MSE: 0.03409014778194535

```
Standard Deviation of MSE: 0.0004245812569252899

Mean RMSE: 0.1846316144072183

Standard Deviation of RMSE: 0.0011466225750459296


Elastic Net Regression:

Cross-validation MSE scores: [0.03369164 0.03378049 0.0342462  0.03485205 0.03388131]

Mean MSE: 0.034090337685301356

Standard Deviation of MSE: 0.00042500727600230297

Mean RMSE: 0.184632121502747

Standard Deviation of RMSE: 0.0011477781563631465


SGD Regressor:

Cross-validation MSE scores: [0.03385413 0.03390526 0.03429964 0.03500758 0.03405968]

Mean MSE: 0.03422525945804981

Standard Deviation of MSE: 0.0004207414415816008

Mean RMSE: 0.18499722857303005

Standard Deviation of RMSE: 0.0011335247451574912


Decision Tree:

Cross-validation MSE scores: [0.02493834 0.02555986 0.02591051 0.02617537 0.02566647]

Mean MSE: 0.02565011027665599

Standard Deviation of MSE: 0.0004144234666021647

Mean RMSE: 0.16015126754962117

Standard Deviation of RMSE: 0.001296834185873496


Random Forest:

Cross-validation MSE scores: [0.02774249 0.02787916 0.02835181 0.02883147 0.02794844]
```

```
Mean MSE: 0.02815067480103283

Standard Deviation of MSE: 0.00039627603613779316

Mean RMSE: 0.1677774932068668

Standard Deviation of RMSE: 0.0011779534169936792


XGBoost Regressor:

Cross-validation MSE scores: [0.024469   0.02464194 0.02503735 0.02556886 0.02476491]

Mean MSE: 0.024896413180714096

Standard Deviation of MSE: 0.00038386869896391305

Mean RMSE: 0.15778130988416622

Standard Deviation of RMSE: 0.0012130259481174588


Feedforward Neural Network (MLP):

Cross-validation MSE scores: [0.03148915 0.03190672 0.03204212 0.03263707 0.03182462]

Mean MSE: 0.031979936245984705

Standard Deviation of MSE: 0.0003757786233116315

Mean RMSE: 0.17882627379378252

Standard Deviation of RMSE: 0.0010488312618449208
```

- Linear Regression showed a mean MSE of 0.0340902 with a small standard deviation, indicating consistent performance across folds, but not a particularly low error.
- Ridge Regression had a very similar mean MSE to Linear Regression, with 0.0340901 and a comparable standard deviation, suggesting that regularization did not significantly change the performance in cross-validation.
- Elastic Net Regression also had a mean MSE close to Linear Regression and Ridge Regression, at 0.0340903, with the smallest variation among the three, although it still did not improve performance markedly.
- The SGD Regressor had a slightly higher mean MSE of 0.0342252, with a low standard deviation, which implies stable but not improved performance over the other linear models.

- The Decision Tree showed a significant improvement with a mean MSE of 0.0256501, which is substantially lower than the linear models, and a relatively low standard deviation, indicating better and consistent performance.
- Random Forest had a mean MSE of 0.0281506, which is higher than the Decision Tree but lower than the linear models, also with low variance, showing it has a good and consistent performance across different data splits.
- The XGBoost Regressor presented the lowest mean MSE of 0.0248964, outperforming all other models in terms of both the mean error and consistency, as reflected by the low standard deviation.
- The Feedforward Neural Network (MLP) had a mean MSE of 0.0319799, which is better than the linear models and worse than the ensemble methods, with a low standard deviation indicating stable performance across folds.

**Final Testing:**

Linear Regression:

Mean Absolute Error (MAE): 0.141118082200026

Mean Squared Error (MSE): 0.03334696853376951

Root Mean Squared Error (RMSE): 0.1826115235514164

R-squared (R2): 0.03185744026953374

Ridge Regression:

Mean Absolute Error (MAE): 0.14112942011459323

Mean Squared Error (MSE): 0.033350582527265256

Root Mean Squared Error (RMSE): 0.18262141858847022

R-squared (R2): 0.03175251736146567

Elastic Net Regressor:

Mean Absolute Error (MAE): 0.14114432858913212

Mean Squared Error (MSE): 0.03335481078420815

Root Mean Squared Error (RMSE): 0.1826329947851925

R-squared (R2): 0.03162976091073322

```
SGD Regressor:
Mean Absolute Error (MAE): 0.14219830858001697
Mean Squared Error (MSE): 0.03354360621831967
Root Mean Squared Error (RMSE): 0.1831491365481139
R-squared (R2): 0.026148576176922678

Decision Tree:
Mean Absolute Error (MAE): 0.11768512315617571
Mean Squared Error (MSE): 0.025347123442875364
Root Mean Squared Error (RMSE): 0.1592077995667152
R-squared (R2): 0.2641121501962348

Random Forest:
Mean Absolute Error (MAE): 0.1266658588567239
Mean Squared Error (MSE): 0.027810493555506674
Root Mean Squared Error (RMSE): 0.16676478511816178
R-squared (R2): 0.19259460148738317

XGBoost:
Mean Absolute Error (MAE): 0.11674170593125628
Mean Squared Error (MSE): 0.024711373791846257
Root Mean Squared Error (RMSE): 0.15719851714264438
R-squared (R2): 0.2825694889456859

MLP:
Mean Absolute Error (MAE): 0.1376038957570324
```

Mean Squared Error (MSE): 0.03126517256461259

Root Mean Squared Error (RMSE): 0.17681960458221985

R-squared (R2): 0.09229697546671778

- Linear Regression has a modest MAE and a low R-squared value, indicating a limited fit to the test data.
- Ridge Regression shows very similar performance to Linear Regression with marginally higher error metrics, suggesting that regularization did not have a significant impact on the test data.
- Elastic Net Regressor also performs similarly to Linear and Ridge Regression with slightly higher errors and a low R-squared value, indicating a comparable fit.
- The SGD Regressor has the highest errors and the lowest R-squared value among the linear models, suggesting it has the weakest fit on the test data.
- The Decision Tree model has notably better performance with a substantially lower MAE and MSE, and a higher R-squared value, indicating a stronger fit compared to the linear models.
- Random Forest also shows improved performance over the linear models with lower errors and a higher R-squared value, indicating a good fit to the test data.
- The XGBoost model has the lowest errors across all metrics and the highest R-squared value, suggesting it has the best fit and predictive power on the test data among all the models.
- The MLP (Feedforward Neural Network) has moderate error metrics and a low R-squared value compared to the ensemble models, indicating a better fit than the linear models but not as strong as the ensemble methods.

| **Model** | **TF-IDF & PCA (RMSE)** | **Word2Vec** | **GloVe** | **Sparse Random Projection** | **Word2Vec 300** | **Word2Vec 300 with Min/Max** | **Hyperparameter Tuning** | **Cross Validation (mean RMSE)** | **Final Test** |
|---|---|---|---|---|---|---|---|---|---|
| Linear Regression | 0.1819 | 0.182205 | 0.182111 | 0.182081 | 0.182188 | 0.182144 | N/A | 0.184632 | 0.182612 |
| Ridge Regression | 0.1819 | 0.182205 | 0.182111 | 0.182081 | 0.182188 | 0.182144 | 0.182141 | 0.184632 | 0.182621 |
| Elastic Net | 0.1850 | 0.185011 | 0.185011 | 0.185011 | 0.185011 | 0.185011 | 0.182137 | 0.184632 | 0.182633 |

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| Regression | | | | | | | | | |
| SGD Regressor | 0.1824 | 0.182490 | 0.182307 | 0.182444 | 0.182574 | 0.182309 | 0.182492 | 0.184997 | 0.183149 |
| Decision Tree | 0.2151 | 0.219044 | 0.215991 | 0.216659 | 0.216858 | 0.215979 | 0.157419 | 0.160151 | 0.159208 |
| Random Forest | 0.1736 | 0.173222 | 0.172797 | 0.172381 | 0.171633 | 0.172704 | 0.165145 | 0.167777 | 0.166765 |
| XGBoost | 0.1722 | 0.174444 | 0.174351 | 0.174495 | 0.174403 | 0.174481 | 0.155192 | 0.157781 | 0.157199 |
| Feedforward Neural Network (MLP) | 0.1693 | 0.167280 | 0.167546 | 0.171493 | 0.171493 | 0.178801 | 0.178607 | 0.178826 | 0.17681 |

**Summary:**

Model XGBoost with Hyperparameter Optimization using GridSearchCV or RandomizedSearchCV after preprocessing step Word2Vec (300 dimensions) was the most effective because it consistently showed the lowest Mean Squared Error (MSE) and the highest R-squared (R2) values across different evaluations, including cross-validation and final testing. This indicates that it had the best fit to the data, capturing the underlying patterns with greater precision than other models. The optimization process allowed it to fine-tune parameters such as 'subsample', 'n_estimators', 'max_depth', 'learning_rate', and 'gamma', which contributed to its superior performance.

On the other hand, the SGD Regressor with the same hyperparameter optimization was the least effective, particularly after the final testing phase, where it showed the highest errors and the lowest R-squared value among the linear models. This suggests that despite the optimization, the SGD Regressor was unable to capture the complexity of the data as effectively as the other models.