Week 5: Feature Engineering, Data Augmentation, and Dimension Reduction

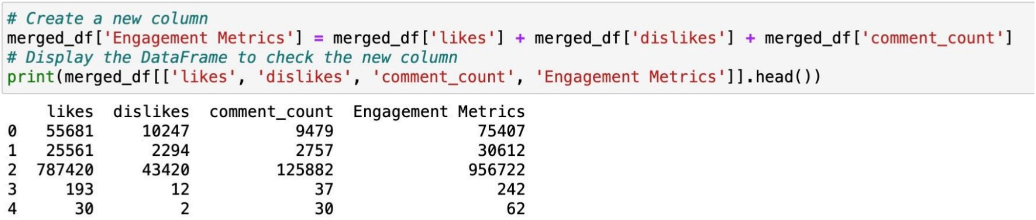
Yujia Cao, Wendi Yuan, Yuhan Zhao

For our project, we initially gathered five datasets from different countries and regions: China, France, Germany, Great Britain, and the United States. These datasets represent diverse perspectives but pose a language challenge, as three of them contain non-English data. While we initially used all five datasets for exploratory data analysis(EDA) to understand our data's structure and distribution, language inconsistencies have become a significant obstacle in preparing for model development and future analysis.

So far, we've completed feature engineering, data augmentation, and dimensionality reduction. However, due to limitations with translation accuracy using available Python packages, we cannot effectively preprocess the non-English datasets. As a result, we decided to focus on the English datasets (Great Britain and the United States) to avoid language conflicts during modeling and text analysis.

Once the model is established, we can revisit the initial non-English datasets and apply a more precise translation process for future iterations, ultimately enabling multilingual expansion of our analysis. This stepwise approach will help ensure the integrity of our model and better prepare us for effective text analysis and prediction.

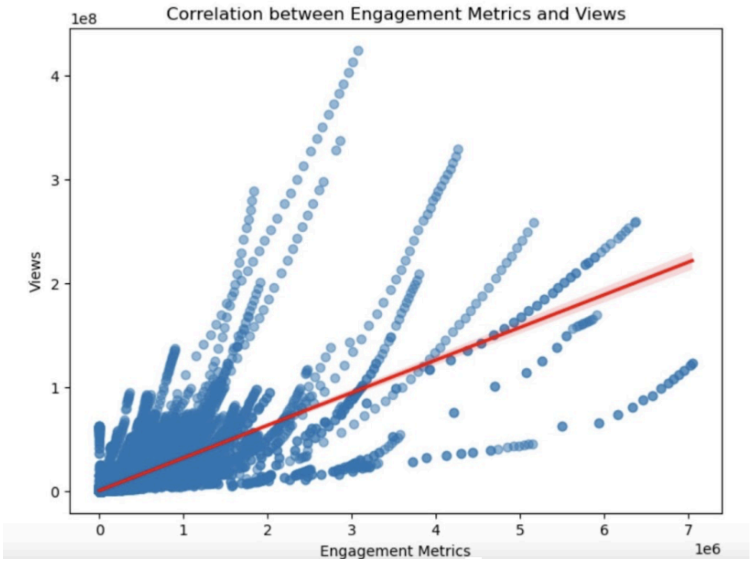
The engagement metrics for YouTube videos represent valuable insights into audience interaction, which plays a crucial role in determining a video's popularity. By summing up the *likes*, *dislikes*, and *comment_count*, a new engagement metric was created, providing a comprehensive view of audience engagement. In our analysis, videos with higher engagement scores tend to have a larger number of views, indicating a positive relationship between these metrics and video visibility.



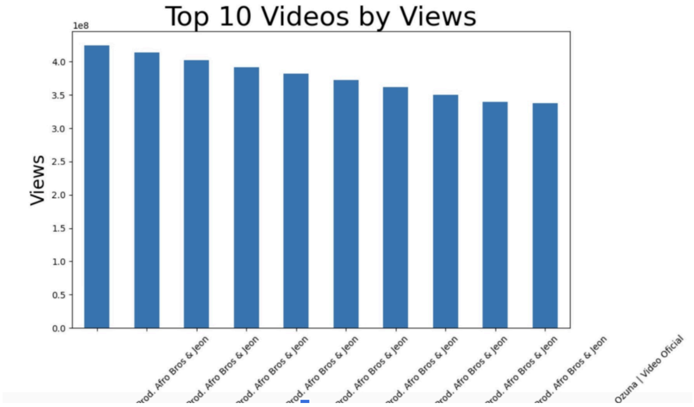
```python
# Create a new column
merged_df['Engagement Metrics'] = merged_df['likes'] + merged_df['dislikes'] + merged_df['comment_count']
# Display the DataFrame to check the new column
print(merged_df[['likes', 'dislikes', 'comment_count', 'Engagement Metrics']].head())
```

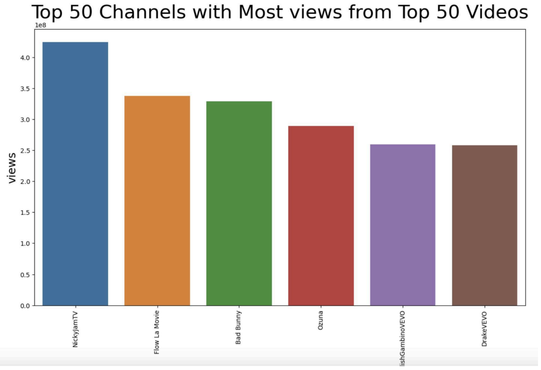
```
    likes  dislikes  comment_count  Engagement Metrics
0   55681     10247           9479               75407
1   25561      2294           2757               30612
2  787420     43420         125882              956722
3     193        12             37                 242
4      30         2             30                  62
```

A scatter plot with a regression line was generated to visually explore the correlation between engagement metrics and *views*. The plot clearly illustrates that as engagement metrics increase, the number of views also rises, albeit with some variability. This suggests that while higher engagement generally drives more views, there are other factors at play that influence a video's performance, such as content relevance, timing, and audience preferences.

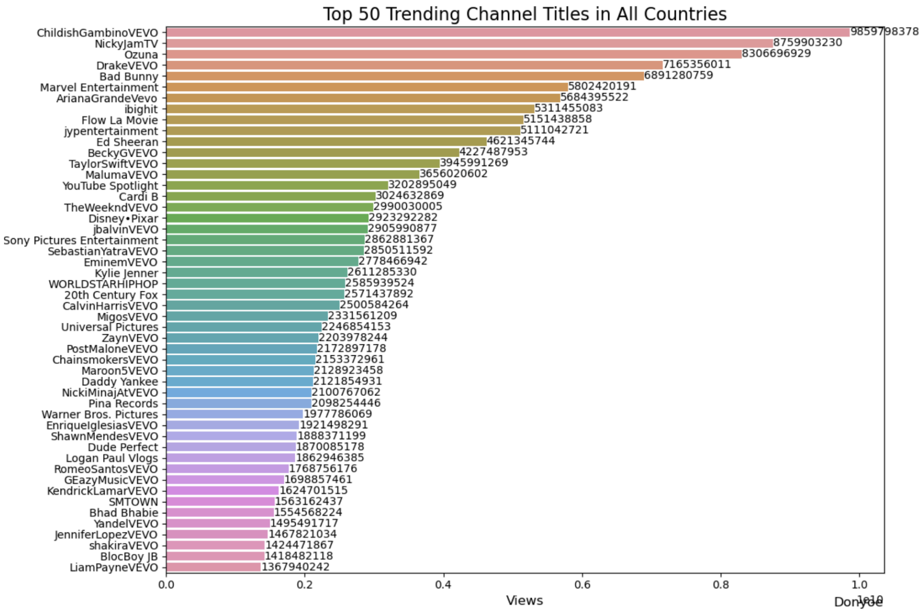


For analysis the top 10 videos by *views* revealed a strong link between high engagement and viewership. Videos that performed exceptionally well in terms of views also had some of the highest engagement metrics, underscoring the importance of fostering audience interaction. Similarly, examining the top 50 videos by *views* demonstrated a consistent pattern where popular artists and content creators, such as Nicky Jam and Bad Bunny, benefited from a strong and engaged fan base.

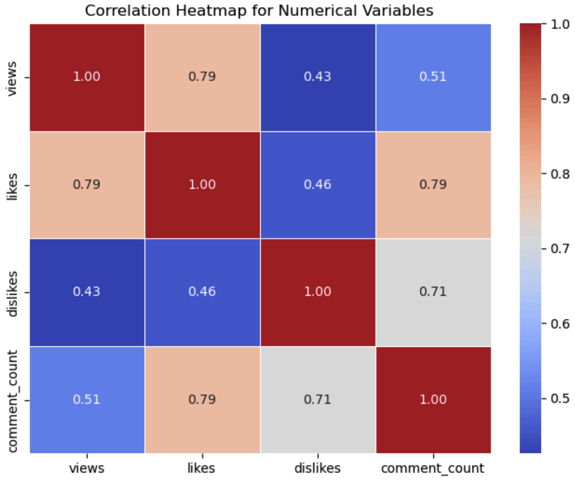




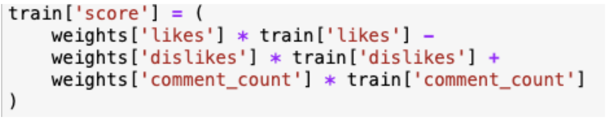
When exploring the top trending 50 channels by *views*, it became evident that channels with large audiences and high engagement tend to dominate viewership across regions. The findings from this analysis reinforce the notion that content creators who encourage more likes, comments, and even dislikes can significantly boost their chances of attracting more viewers. This correlation suggests that maximizing user engagement should be a key focus for creators looking to enhance their content's performance on platforms like YouTube.



We computed the correlation matrix for key numerical variables to assess their relationships. The correlation between *likes* and *views* is 0.78, indicating a strong positive relationship where videos with more likes generally receive more views. The correlation between *comment_count* and *views* is 0.50, suggesting a moderate positive correlation; thus, videos with more comments tend to attract more views. *dislikes* and *views* have a weaker positive correlation of 0.41, showing that even disliked videos can still garner high view counts.

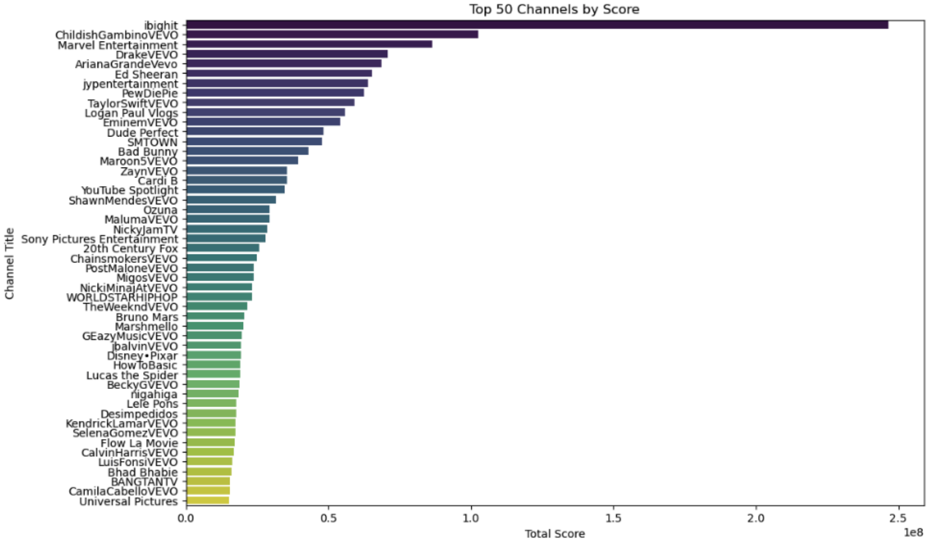


To better evaluate video performance, we calculated a composite score based on engagement metrics: *likes*, *dislikes*, and *comment_counts*. We normalized the absolute values of these metrics to ensure proportional contribution to the score, resulting in the following weights: *likes* (46.04%), *dislikes* (24.44%), and *comment_counts* (29.49%).

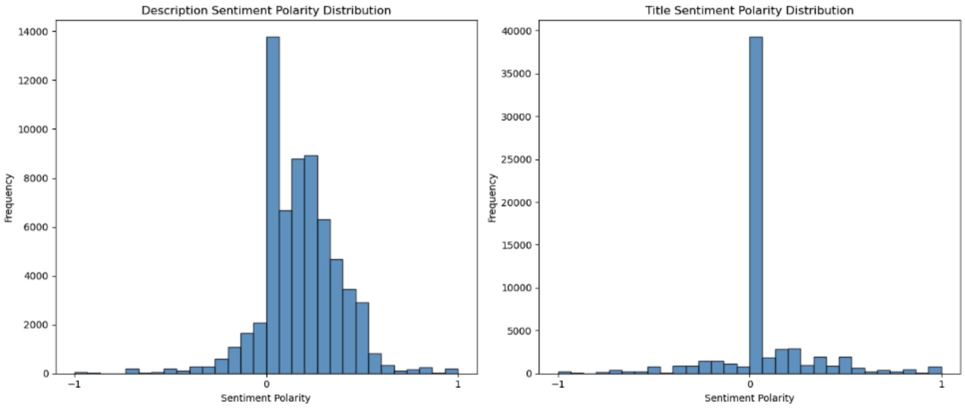


```
train['score'] = (
    weights['likes'] * train['likes'] -
    weights['dislikes'] * train['dislikes'] +
    weights['comment_count'] * train['comment_count']
)
```

This scoring formula rewards videos with high engagement in *likes* and *comment_count* while penalizing high *dislikes* counts. The scores rank the videos in descending order, highlighting the top performers based on audience engagement.

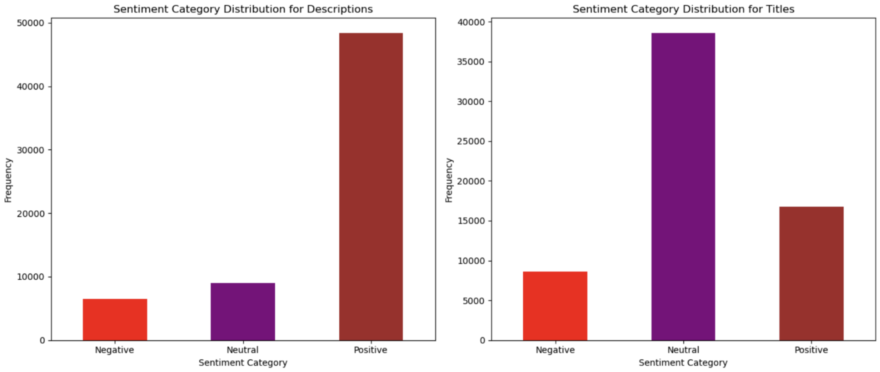


The next step for our project is sentiment analysis of *description* and *title*. Those content provides valuable insights into how content creators utilize different textual elements to communicate with their audience. By applying the TextBlob library, we were able to calculate polarity scores, which measure sentiment on a scale from -1 (most negative) to 1 (most positive). The average *description* sentiment score was 0.1717, indicating a moderately positive trend, while the average *title* sentiment score was only 0.0478, suggesting that *title* remains largely neutral in tone. This stark difference raises critical questions about the roles each of these components play in engaging users.



The histograms illustrating sentiment polarity distribution offer further clarity. For *description*, the data demonstrates a wider spread of sentiment values, with a significant concentration of slightly positive sentiment around the 0.1 to 0.2 range, confirming the more expressive and potentially engaging nature of descriptions. In contrast, title sentiment polarity is highly condensed around zero, with very few extreme positive or negative values. This finding could suggest that content creators may deliberately keep titles factual or neutral to increase clarity, while reserving emotional engagement for the *description* field. Approximately 14,000 data points for descriptions cluster around the 0.0 mark, with a long tail extending toward positive sentiment. In titles, the overwhelming number of data points fall directly at 0, showing that creators seldom use emotionally charged language in titles.

When sentiment was categorized into "positive," "neutral," and "negative," the differences between *titles* and *description* became even more apparent. Over 450,000 descriptions were categorized as positive, making up the majority of description content, with a relatively small portion labeled as neutral or negative. Conversely, in *title*, the neutral category dominates, with over 350,000 titles falling into this class, followed by negative sentiment titles, which make up a sizable portion, while only a small fraction of titles are classified as positive. This significant disparity suggests a strategic approach: creators are more likely to use neutral or even negative language in titles, potentially to elicit curiosity or controversy, while positive sentiment is more frequently reserved for descriptions to build excitement or convey enthusiasm.

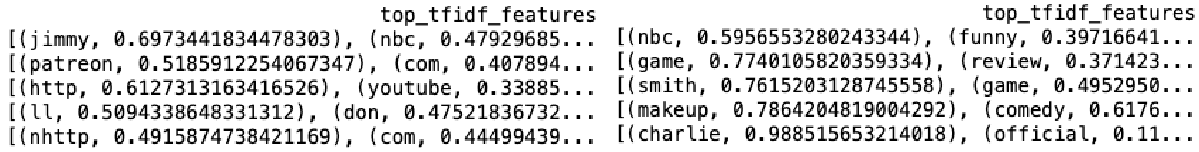


These findings lead to critical questions about how sentiment influences content performance. With *description* skewing heavily positive and *title* remaining neutral or negative, it's possible that creators are experimenting with different emotional tones in distinct parts of their content to influence engagement metrics such as click-through rates and watch times. The heavy reliance on neutral and negative sentiment in titles could be intentional, as viewers might be more inclined to click on content that presents a more grounded or intriguing tone, rather than overt positivity. On the other hand, the predominance of positive descriptions could reflect an attempt to emotionally engage viewers once they've already clicked, offering them a more inviting and positive view of the content they are about to consume. These patterns point toward a complex dynamic between sentiment in titles and descriptions, where different strategies are employed to maximize audience engagement at various stages of the viewer experience.

Another feature we have implemented is the use of Term Frequency-Inverse Document Frequency (TF-IDF) for our dataset. This technique transforms our text fields into a format that emphasizes the significance of specific words relative to each document, which can be valuable for models predicting engagement, sentiment, and trends. TF-IDF is a statistical measure that reflects the importance of a word within a document in a collection, balancing the frequency of a word in a particular document with its overall occurrence across the entire dataset.

In our analysis, we convert the textual columns into numerical vectors using TF-IDF, enabling us to incorporate text as features in our machine learning models. We specifically selected the *description* and *title* columns because, after evaluating other textual data, we found these two fields contained the most compelling content for viewer engagement. Furthermore, these columns include complete sentences, making them more relevant than isolated words.

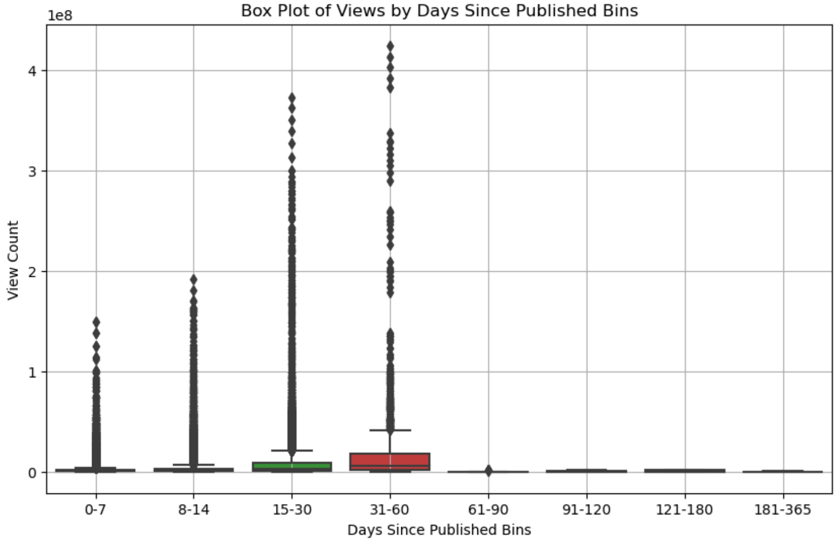
The output from the TF-IDF process is a sparse matrix, where each word represents a feature and the corresponding values are their TF-IDF scores. These scores help identify terms that are distinct across the entire dataset, with higher TF-IDF values indicating words that are more unique and relevant to specific videos. Both the *description* and *title* columns undergo separate TF-IDF transformations, with the results stored as new columns featuring the top TF-IDF attributes. Below is an example of the highest TF-IDF scores within the entire dataset for the *description* (left) and *title* (right):



```
                         top_tfidf_features                           top_tfidf_features
[(jimmy, 0.6973441834478303), (nbc, 0.47929685... [(nbc, 0.5956553280243344), (funny, 0.39716641...
[(patreon, 0.5185912254067347), (com, 0.407894... [(game, 0.7740105820359334), (review, 0.371423...
[(http, 0.6127313163416526), (youtube, 0.33885... [(smith, 0.7615203128745558), (game, 0.4952950...
[(ll, 0.5094338648331312), (don, 0.47521836732... [(makeup, 0.7864204819004292), (comedy, 0.6176...
[(nhttp, 0.4915874738421169), (com, 0.44499439... [(charlie, 0.988515653214018), (official, 0.11...
```

The purpose of creating the feature *days_since_published* and binning it into specific time blocks is to understand how the time lag between a YouTube video's *publish date* and its *trending date* affects its view count. This time lag reflects the period from when the video was uploaded until it started trending, giving insights into the video's lifecycle and popularity growth pattern.

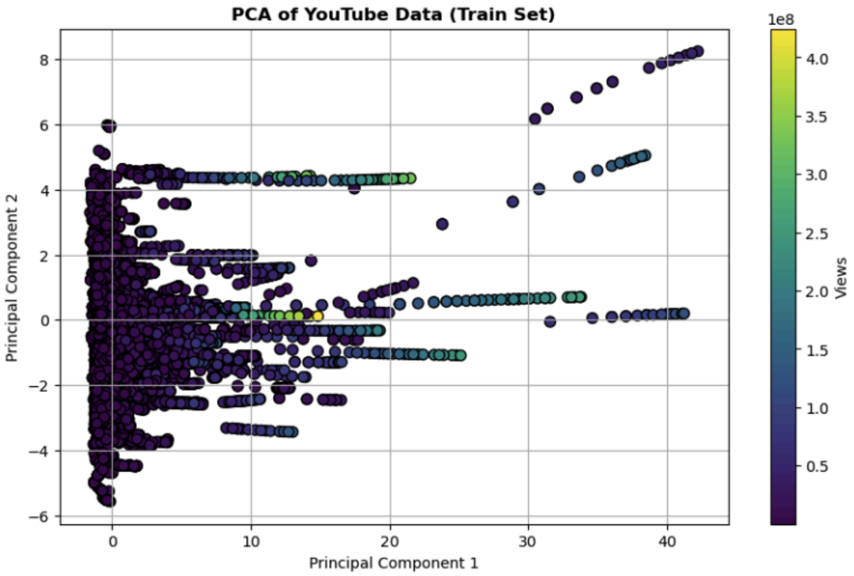
By using bins ['0-7', '8-14', '15-30', '31-60', '61-90', '91-120', '121-180', '181-365']:We are able to categorize videos into distinct time intervals based on how quickly they gain traction. The maximum days of the time lag is 365, if we expand the time lag longer, the trend result may not be that correct since a year has passed after a video was published, and other irrelevant factors may influence it.



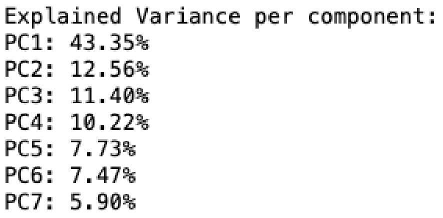
From the plot, it's clear to observe that most of the videos get viral or receive view counts in 60 days after they were published, and *views* trend increase from 7 days to 60 days. We want to see if videos that become popular within the first week tend to achieve high view counts, indicating rapid virality. We also notice that It's common for most videos to gain momentum later after being published. By analyzing the view counts in the 30-60 day window, we could identify the trend that a substantial portion of videos receive their most views during this period. This allows us to compare the views across different time frames and understand how quickly or slowly videos become popular. And a box plot of *views* by these bins will visually reveal whether videos tend to go viral quickly (within the first week or two) or whether some take longer (up to a year) to reach their peak popularity.Ultimately, this feature helps us clarify the relationship between time lag and video success, allowing us to see if there is a standard time frame for when most YouTube videos go viral, and if videos published within certain periods tend to accumulate higher views. This time-based insight is valuable for modeling and can be incorporated into further predictions about video virality.

After we process our features, we reduce the dimensionality of data by applying Principal Component Analysis (PCA). There are over 15 predictor variables after we do the feature engineering and combination, before we did the PCA, we firstly identify and drop the non-numerical variables including: *publish_time, title, channel_title, tags, description, location, trending_date, trending_day_of_week, day_of_week.* These are original non-numerical variables before we do the feature analysis, but after we do the sentiment analysis and synthesize new features, there are several more variables like *description_sentiment*, *title_sentiment* added up to check if there exist missing values. After we check variables in train data, we fill the missing values in test data to be 0 to keep the dataset size uniformity. Then, we standardize variables including the target variables to make sure that their data type is the same so PCA works more efficiently. After we prepare these for further analysis, there are no missing values among the dataset and data points are complete.

We choose the setting for *n_component* for PCA is 95%, which is the base for retaining components that could capture 95% of the variance in the dataset, and we don’t want to reduce the dimension of the dataset to only 2 components. Even though variables in the dataset are not too complicated to understand , their relationship is not easy to capture. A 95% variance retention may lead to retaining more dimensions, which could be necessary to capture these relationships effectively. Also, By setting a threshold of 95% variance, we intend to still keep a significant portion of the original dataset's information instead of dropping larger portions of components. This helps maintain the plentiful characteristics of our dataset, which could be essential for us to implement downstream tasks such as model experiments like classification or regression.



In the graph, the x-axis represents the first principal component (PC1), and the y-axis represents the second principal component (PC2). These axes represent new dimensions derived from the original features that capture 95% variance in the dataset. Each point corresponds to an observation in the dataset,projected onto the new PCA dimensions. We obtain four results from the PCA analysis. We get the variance statistics from the graph from PC1 to PC7.



```
Explained Variance per component:
PC1: 43.35%
PC2: 12.56%
PC3: 11.40%
PC4: 10.22%
PC5: 7.73%
PC6: 7.47%
PC7: 5.90%
```

PC1 is 0.4335, and it means that the first principal component captures 43.35% of the total variance in the data. This means some part of the dataset's variability is captured through a single space, but PC1 is not enough to capture an essential part of the dataset. And PC2 is 0.1256, PC3 is 0.1140, PC4 is 0.1022.We do a simple calculator here by adding PC1 to PC4 and get 0.7753，which indicates these four components represent the most crucial components for explaining the dataset. After PC4, the subsequent component explains the diminishing amount of the variance. PC5 is 0.0773, PC6 is 0.0747 and PC7 is 0.0599, they explain less than 10% of the variance compared to the first 4 components, while representing less important patterns of the data. If adding seven components together, the result is 0.982 means 98% of the total variance was explained. PC1, PC2, PC3 and PC4 capture most of the variance of the dataset, and the rest of principal components helps to retain some patterns that the former four components may miss.