**Personalized Movie Recommendations through clustering and NLP**

Group 9

Yu-Chin (Alyssa) Chen, Shuo-Ming (Chris) Kuo, Pei-Hsin (Bonny) Yang, Yu-Chun (Lila) Su

With the abundance of movies available, users often face difficulty in finding content that aligns with their tastes. However, we believe that movies favored by the same audience or movies with high box office revenue will share certain common characteristics to some extent. Leveraging this information can help platforms better select movies for release and assist production companies in creating successful films. Our primary objective is to find the patterns and develop a recommendation system that employs NLP techniques on movie overviews. Additionally, we aim to incorporate other relevant variables for clustering to enhance the accuracy and effectiveness of our recommendation system.

**1. EDA & Pre-processing**

In the preprocessing step, we took 10 crucial steps to clean our dataset, ensuring its suitability for subsequent analysis. For more details of each step please refer to Table 1 in the appendix. This included checking for missing values, removing unused columns and null values, splitting the 'Date' column, setting a specific year period, and filtering the data based on language and other criteria. Additionally, we enriched the dataset by adding a 'Continent' column and transformed categorical variables into dummy variables. We normalized numeric columns for uniform scaling and arrived at a final dataset ready for comprehensive analysis.

**2. Data Analysis**

**Challenge:**

A significant challenge we faced was the high prevalence of zero values in the revenue and budget features, with nearly 50% of entries being zero. We decided to exclude them from our analysis to prevent potential bias these values could introduce to our outcome. This incomplete data limited our ability to delve deeply into pricing aspects. Additionally, given the large size of our dataset, comprising over 900,000 entries, we filtered it to focus on movies from the past decade, resulting in the loss of insights into older films. Despite our efforts to use t-SNE for dimensional reduction, we were unable to obtain results even after an hour of processing. Due to time constraints, we discontinued this approach.

**Clustering Analysis:**

For our clustering analysis, we employed PCA along with a scree plot to determine the optimal number of components that effectively capture the most variance in the dataset. This process helped in reducing the dimensionality of the data while retaining as much information as possible. The cumulative explained variance ratio plot revealed that a minimum of 9 components is necessary to capture at least 85% of the variance, indicating the key dimensions for representing the dataset effectively (Figure 2). Furthermore, we attempted to employ t-SNE to further analyze the data. However, despite trying the number of components ranging from 3 to 9, the computation ran for 60 minutes without yielding results. Consequently, we retained the results from PCA for our analysis.

Additionally, we utilized k-means clustering to identify inherent patterns and groupings within the data. To determine the optimal number of clusters for k-means, we constructed an Elbow plot, which depicts the relationship between the number of clusters and the within-cluster sum of squares. The optimal number of clusters for our dataset would be 5.

**NLP Analysis:**

For our NLP analysis, we used the "overview" column of our dataset. We utilized techniques such as n-gram analysis, TF-IDF, word2vec, and GloVe. By comparing the results obtained from these techniques, we continued to iterate and improve our NLP techniques to enhance the accuracy and relevance of our movie recommendations. Our goal is to fine-tune our algorithms and explore additional tokenization approaches to deliver even more precise suggestions to users, ensuring a satisfying and personalized movie-watching experience.

Additionally, parameter tuning was conducted in the Word2Vec model to optimize its performance. We adjusted the vector\_size parameter, exploring values of 100 and 300 to refine the dimensionality of single-word embeddings and capture more semantic relationships. Further adjustments included changing the number of windows from 3 to 5, expanding the context window of word embeddings to encompass broader contextual information. Ultimately, the final parameters were selected based on computational efficiency and the resultant performance, followed by a comparative analysis with other NLP models.

* **3. Results**

**Clustering Result:**

Employing k-means clustering on the dataset of 33,688 movies resulted in the segmentation of the data into 5 clusters. The distribution of movies across these clusters is as follows: 7,104 movies belong to Cluster Zero, 5,796 movies belong to Cluster One, 12,982 movies belong to Cluster Two, 358 movies belong to Cluster Three, and 7,448 movies belong to Cluster Four. This segmentation offers initial insights into the inherent structure and patterns within the dataset. The result is shown in Figure 3.

From the preliminary results, it's evident that Cluster Zero has the shortest average runtime, about 15 minutes, indicating that it is primarily composed of short videos. Cluster Three stands out with the highest vote count and popularity among all clusters, indicating it likely contains some highly acclaimed movies. Notably, Cluster One exhibits the lowest average vote rating, implying that movies in this cluster might not be highly recommended.

**NLP Result:**

For the movie overviews, we utilized natural language processing (NLP) techniques to enhance the precision of movie recommendations for users. Specifically, we employed TF-IDF to determine which words are most important in each description. Additionally, we utilized Word2Vec to uncover connections between words, revealing hidden meanings within the overview text. Based on the results from these models, we provided recommendations for five movies that best match users' preferences and interests. We used "Avengers: Endgame" and "Dune" as examples. Notably, the top 5 recommended movies varied slightly between models. The detailed results are provided in the appendix (Table 2 to Table 5). We also tested the model performance with genres by filtering romance movies and thriller movies to observe their cosine similarity in different models. Word2Vec performs better in this scenario. We expected to get relatively low scores since these two genres are very different, but the GloVe model gives us a high score, with the average above 0.85. Therefore, we consider this a poor performance that requires optimization or replacement.

**4. Insights**

By classifying all films into five clusters, it is recommended to provide diverse content, with each cluster potentially representing a different genre, style or theme, reflecting the different preferences and interests of the audience. Additionally, differences in votes and popularity between clusters indicate varying levels of audience engagement and acceptance. Understanding the reasons for a movie's popularity can provide valuable insights into audience preferences and trends. Taken together, these insights provide an in-depth understanding of corpus composition and audience dynamics, providing valuable opportunities to further explore and refine movie recommendation systems and content curation strategies.

For NLP analysis, we utilized techniques such as n-gram analysis, TF-IDF, Word2Vec, and GloVe to gain a comprehensive understanding of movie overviews. Each technique offers unique insights into the significance and context of words within the text. For example, in the case of the movie "Dune," both n-gram analysis and TF-IDF highlight its frequency and importance within the movie overviews. The top 5 movie overviews may contain specific keywords, such as "young man", "planet", and "space". In contrast, Word2Vec and GloVe delve deeper into the meaning and semantic relationships of words. The top 5 movie overviews identified through Word2Vec and GloVe may mention themes, concepts, or contexts, such as "space", "universe", and "planet", offering a deeper understanding of their usage within the text. By integrating these techniques, we can extract comprehensive insights from movie overviews, thereby enhancing the accuracy and relevance of movie recommendations for users.

**5. Conclusion**

The project aims to employ unsupervised machine learning to craft personalized viewing suggestions based on users' watch histories, using keywords from movie descriptions to refine recommendations. This enhances viewer satisfaction and loyalty, as well as helps users discover similar or new content more easily. For the platform, better recommendations can boost engagement, lengthen viewing times, and improve retention, while also informing content decisions and marketing, driving revenue and market competitiveness. For filmmakers, it broadens audience reach and guides more strategic content creation and promotion, potentially increasing box office success. In essence, the system serves audiences, platforms, and creators by promoting quality content discovery and enjoyment.

**6. Future Work**

We will also try to acquire additional features such as revenue and budget to enhance our system, allowing us to better capture the underlying structure and characteristics of the movie dataset. This will result in a more robust clustering algorithm. Additionally, we decide to implement mechanisms for collecting user feedback and preferences, such as ratings, reviews, or explicit user input, to continuously update and refine our recommendation system.

**7. Use of AI**

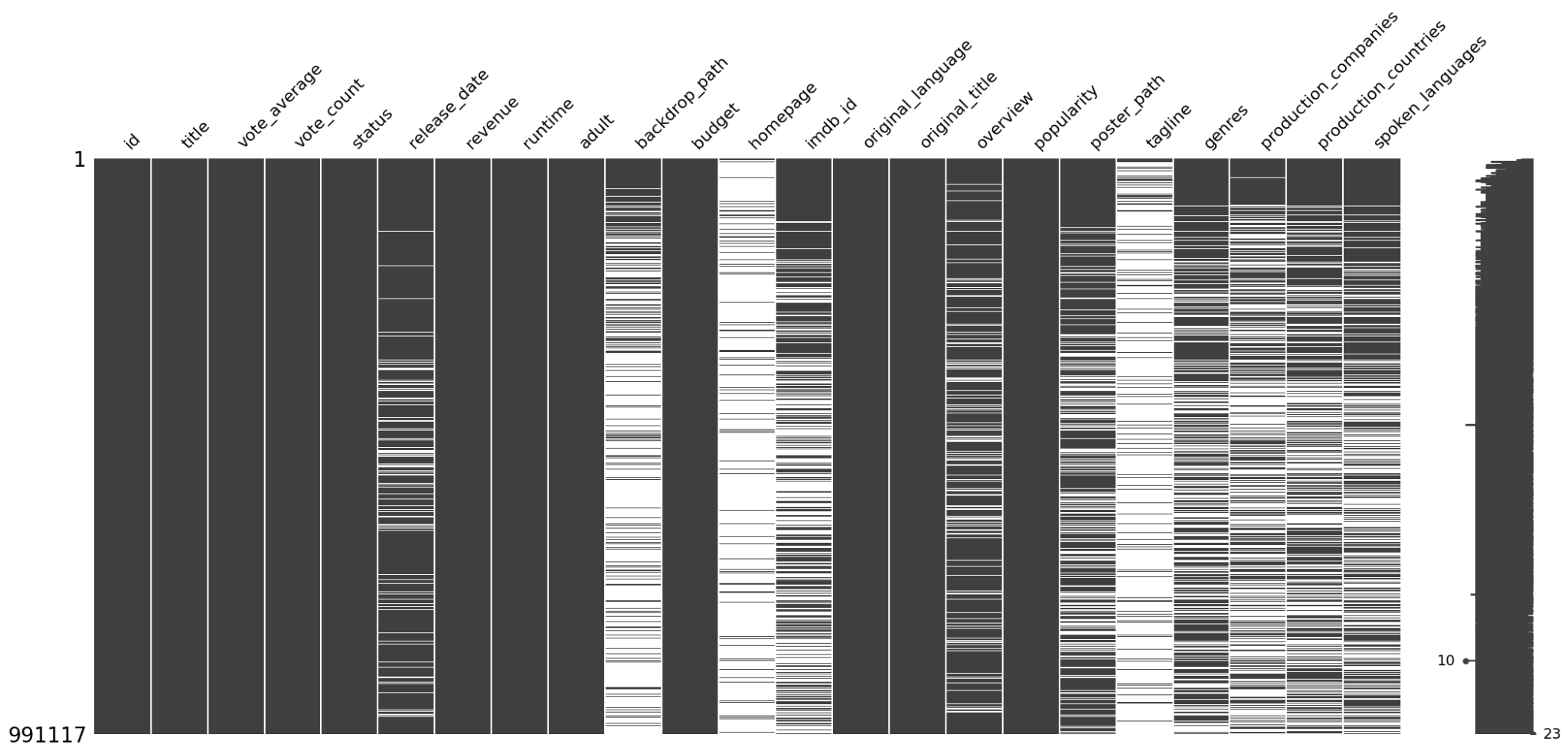
We use chatGPT to refine our proposal’s sentences and correct the code.

**Appendix**

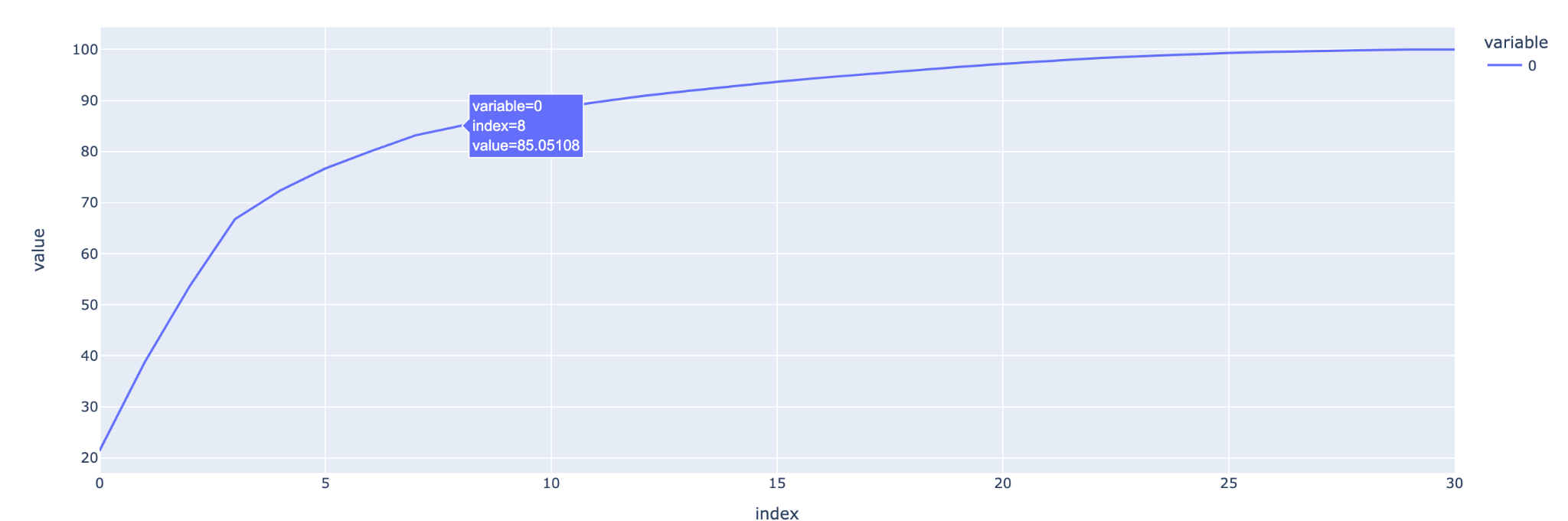
**Table 1. Data Preprocessing Steps**

| **Steps** | **Explanation** |
| --- | --- |
| **1. Checking the missing value** | In the beginning, we generated a missing value chart (Figure 1) to provide a comprehensive overview of the data distribution, aiding us in identifying areas requiring attention. |
| **2. Drop Unused Columns** | We systematically pruned the dataset by removing redundant columns and those plagued by substantial missing values, defined as exceeding 50%. These excised columns included 'id', 'backdrop\_path', 'homepage', 'imdb\_id', 'poster\_path', 'original\_title', 'tagline', 'production\_companies', 'revenue', 'budget', and 'spoken\_languages', thereby streamlining the dataset for enhanced clarity and efficiency. |
| **3. Drop Null Values** | To protect data integrity, any rows containing null values ​​are also excluded from consideration |
| **4. Split 'Date' Column** | After preliminary refinement, we extract time information from the "release\_date" field. |
| **5. Set Year Period** | We are interested in films released from 2014 to 2023, covering a ten-year time frame. By narrowing the scope to this time period, we aimed to capture a contemporary and relevant subset of films for analysis. |
| **6. Filter Data** | We used filtering to ensure that only released films were included. Additionally, limiting our dataset to "English” language films. We also excluded movies with zero votes, as popularity and public acceptance are important factors in determining the suitability of recommendations. Moreover, since negative runtime values are nonsensical, we exclude movies with negative or zero runtime durations. |
| **7. Add 'Continent' Column** | To enrich our data set with geographic insights, we converted the "Country of Production" information into the more informative "Continent" field. |
| **8. Get Dummy Variables** | We transformed categorical variables such as 'genres' and 'continent' into dummy variables. This transformation resulted in 19 variables for genres and 7 variables for continents, providing granular insights into movie categorization and global distribution patterns. |
| **9. Normalize Numeric Columns** | For the other continuous variables, we standardized them to ensure uniform scaling across the dataset. |
| **10. Get Final Dataset** | As a cumulation of these preprocessing steps, our final dataset included 33,688 rows for comprehensive analysis and insightful exploration. |

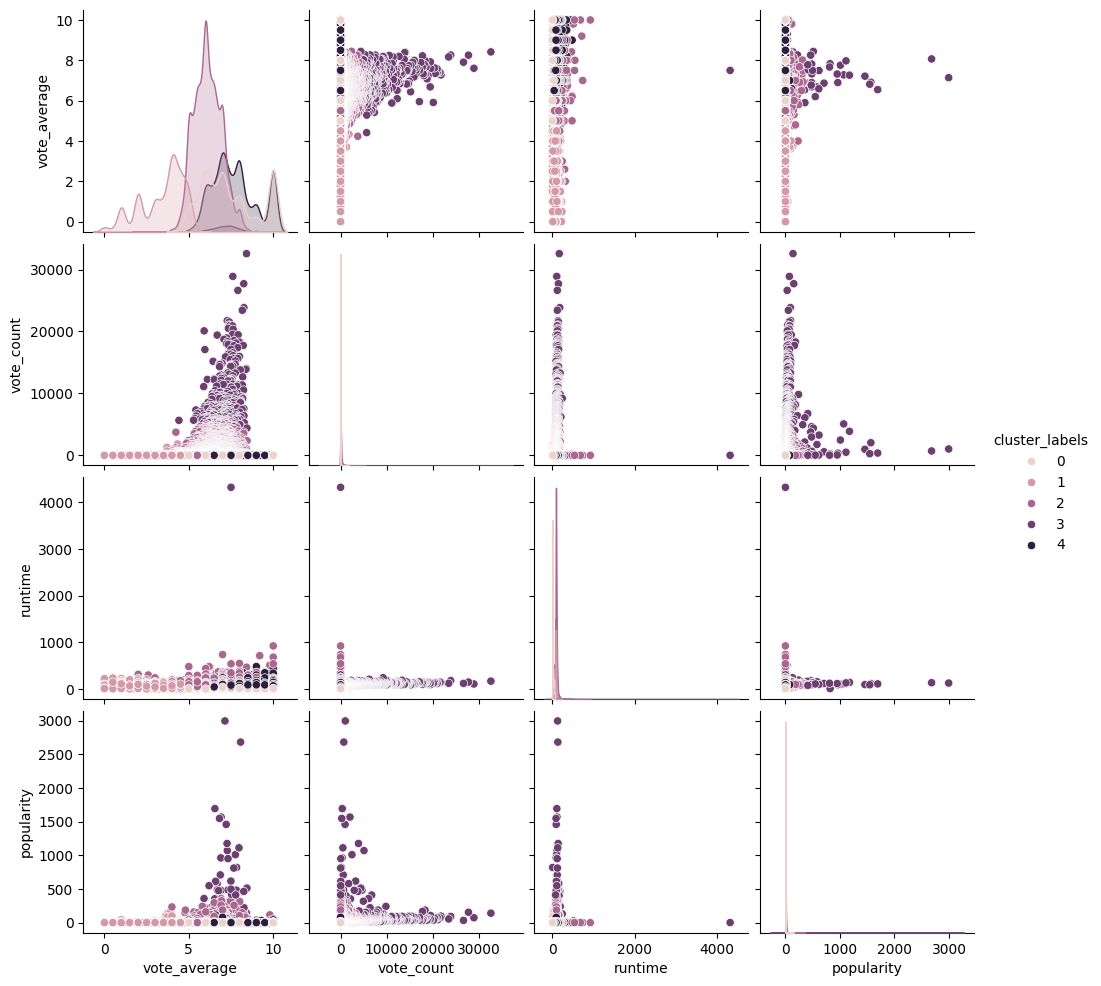
**Figure 1. Check null value plot**

****

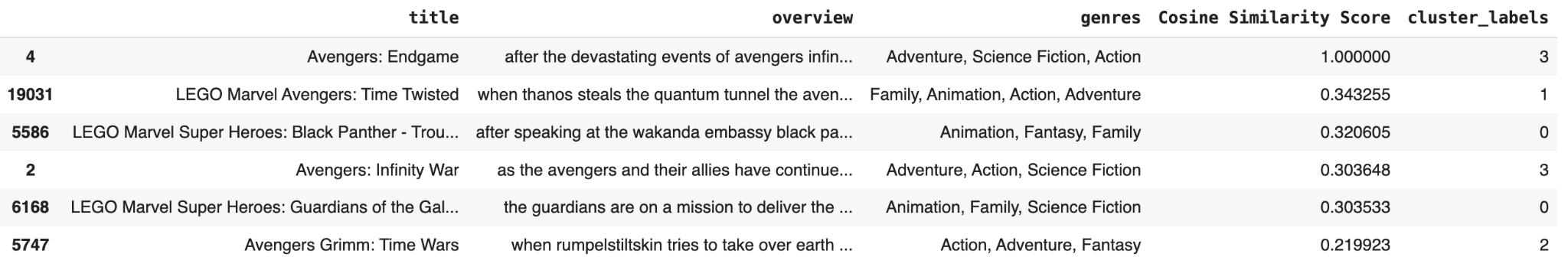
**Figure 2. Cumulative explained variance ratio plot of PCA**

****

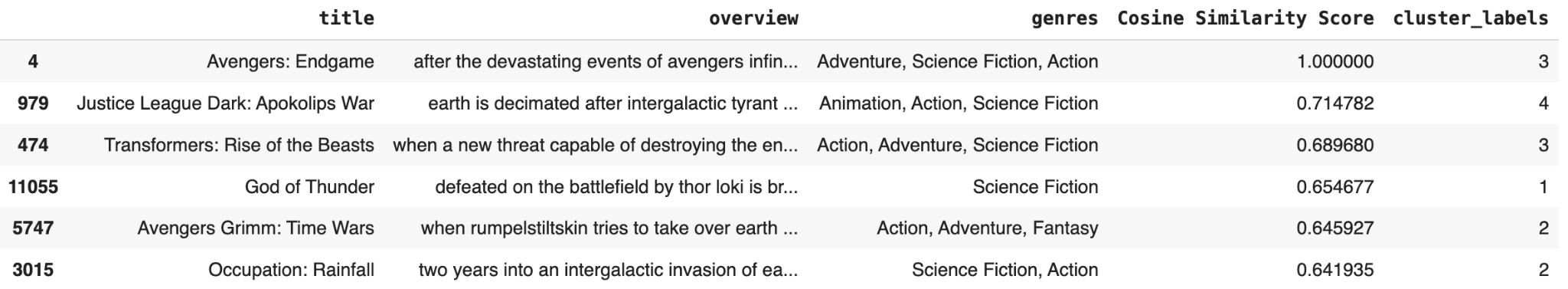
**Figure 3. Pair Plot after Clustering**

****

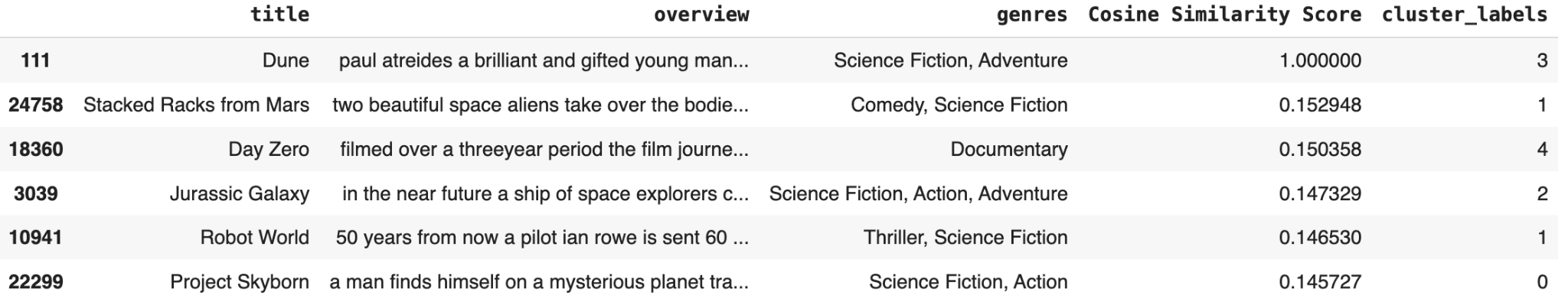
**Table 2. The TF-IDF result for Avengers: Endgame**

****

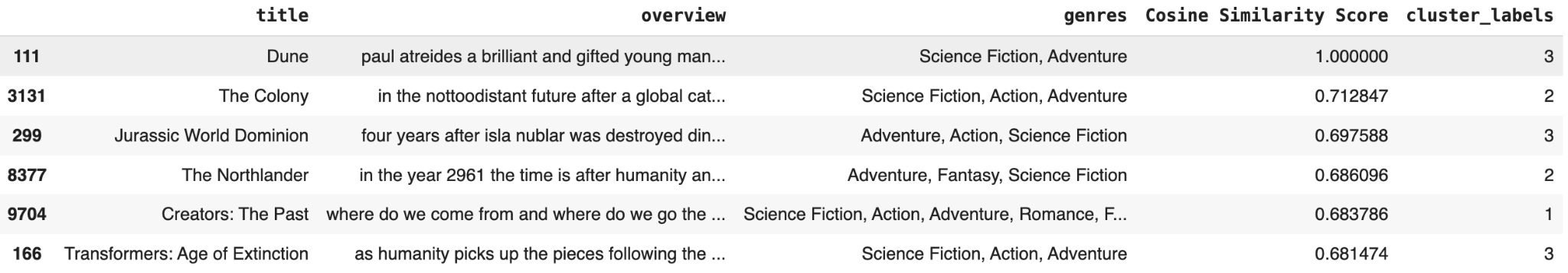
**Table 3. The Word2Vec result for Avengers: Endgame**

****

**Table 4. The TF-IDF result for Dune**

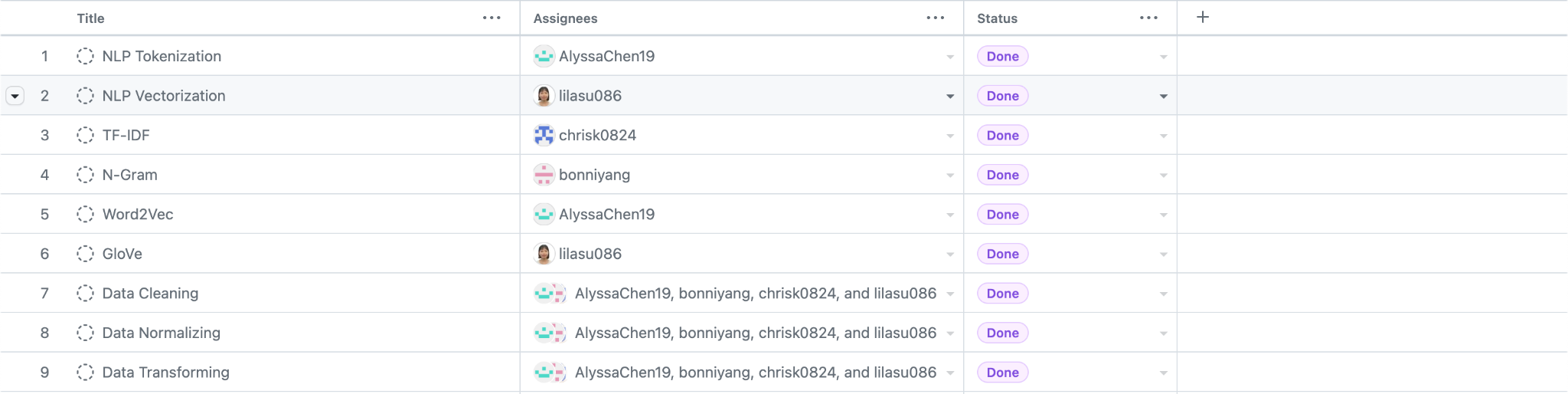
****

**Table 5. The Word2Vec result for Dune**

****

**Coding Contribution**

<https://docs.google.com/spreadsheets/d/1Tju_26ckC3_1ctQHKhLu2pGL1KIdugHw-OlMD6j-oGY/edit?usp=sharing>

****

****

**Github Link**

<https://github.com/bonniyang/ba820.git>