I, Poonam Parag Thakur, understand that this is an individual take-home exam. I understand and agree that I will not work with any other humans or AI mechanisms to complete this exam. I agree on my honor to only do my own work, not to share or request any assistance on this exam, and to use only the resources permitted which include Dr. Gates code, sites, and resources, my own code, web resources that are not AI-based, anything on Canvas, and any books/articles, etc. I understand and agree that sharing on an exam is unethical and unfair to my fellow classmates. I also agree that if sharing occurs, I will accept a 0 grade. I also agree to submit my Exam before the deadline. The absolute last moment to submit is May 5 at 12 noon MT. I understand that it is impossible to submit after this even by one second 😊

PoonamPThakur Signature 04-25-2023 Date

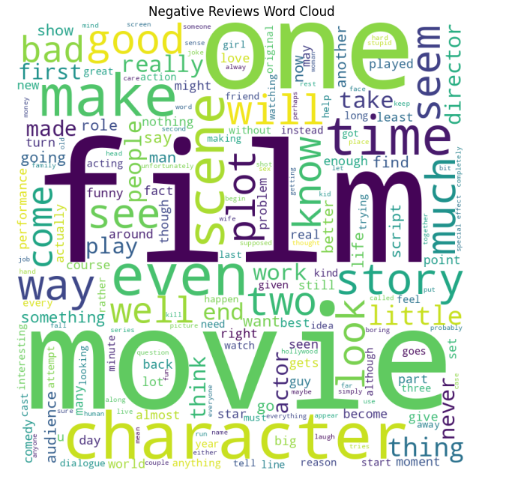
## **Part 1: Introduction**

The given dataset contains a collection of movie reviews and corresponding user sentiments. The overall topic of this dataset is to explore the relationship between movie reviews and user sentiment to gain insights into the factors that influence people's opinions about movies. The stakeholders for this data could be movie production companies, distributors, theaters, and online streaming platforms that are keen on understanding the preferences and opinions of their audience. The information gained from this data could help these stakeholders make informed decisions about movie production, marketing, and distribution.

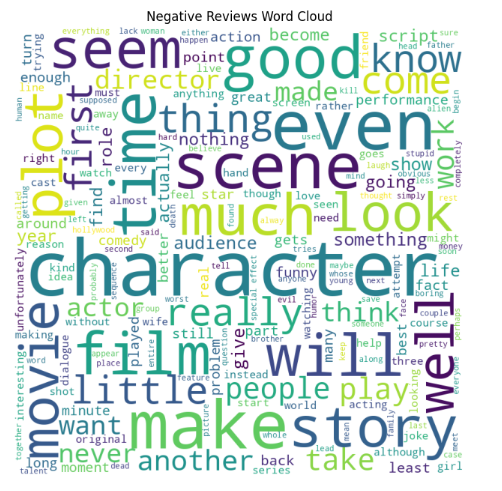
The goals for this dataset are to identify the key features that make a movie popular among audiences, to predict the sentiment of users based on the movie review, and to determine the factors that contribute to positive or negative user sentiment. By analyzing this data, stakeholders can gain a deeper understanding of audience preferences and identify areas that require improvement in the movie industry. The predictions derived from this dataset could help the stakeholders optimize their marketing campaigns and tailor their offerings to meet the needs of their target audience. This dataset could provide valuable insights into the movie industry and help stakeholders make data-driven decisions.

## **Part 2: Data Cleaning**

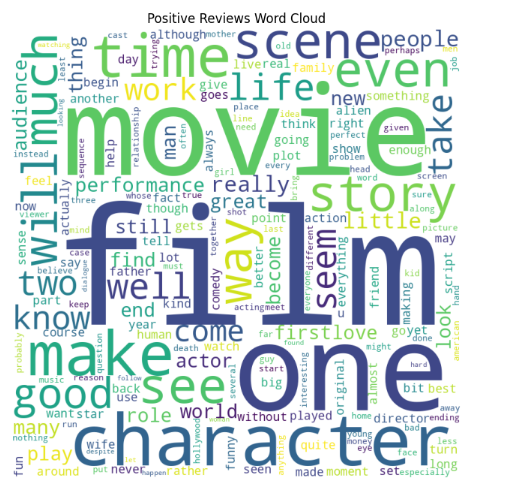
### Word cloud for “neg” data before cleaning



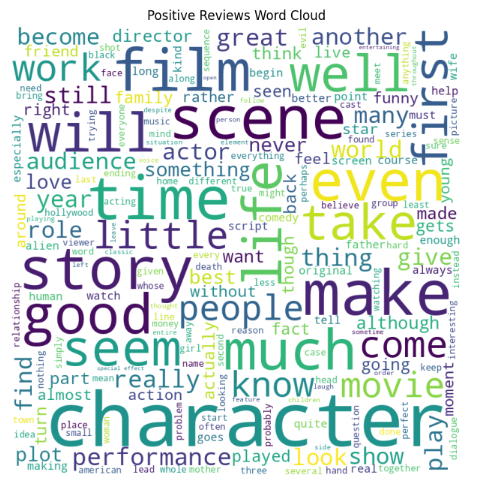
### Wordcloud for “neg” after cleaning



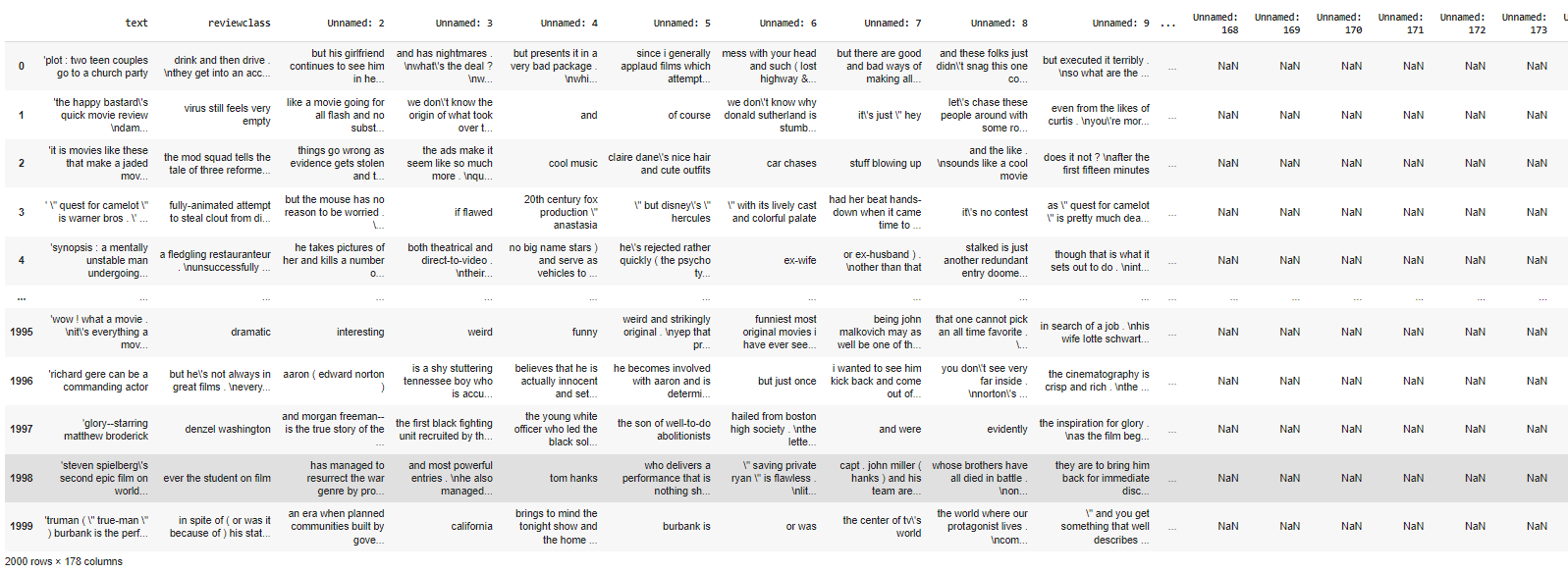
### Word cloud for “pos” data before cleaning

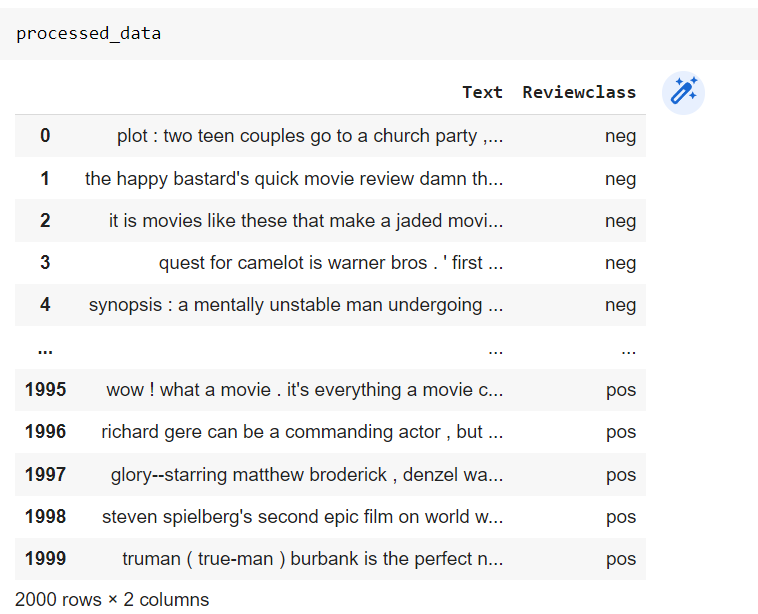


### Wordcloud for “pos” after cleaning

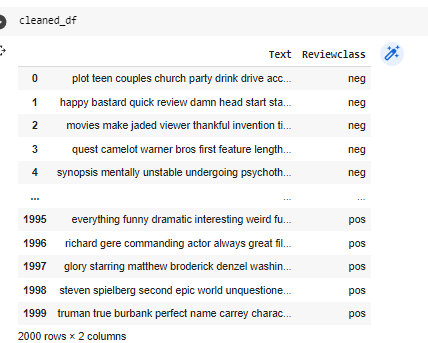


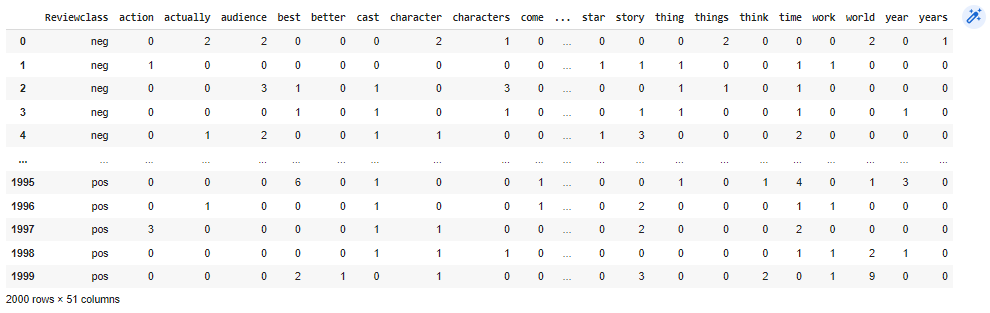
### “before cleaning” data frame



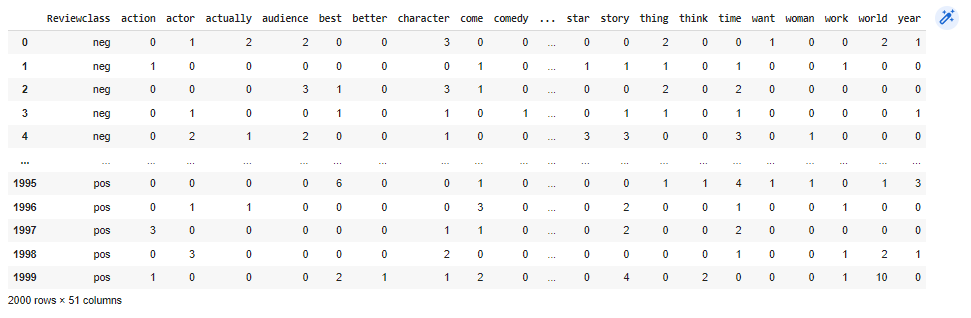


### “After cleaned” data frame





### Second Data Frame - Lemmatized DF

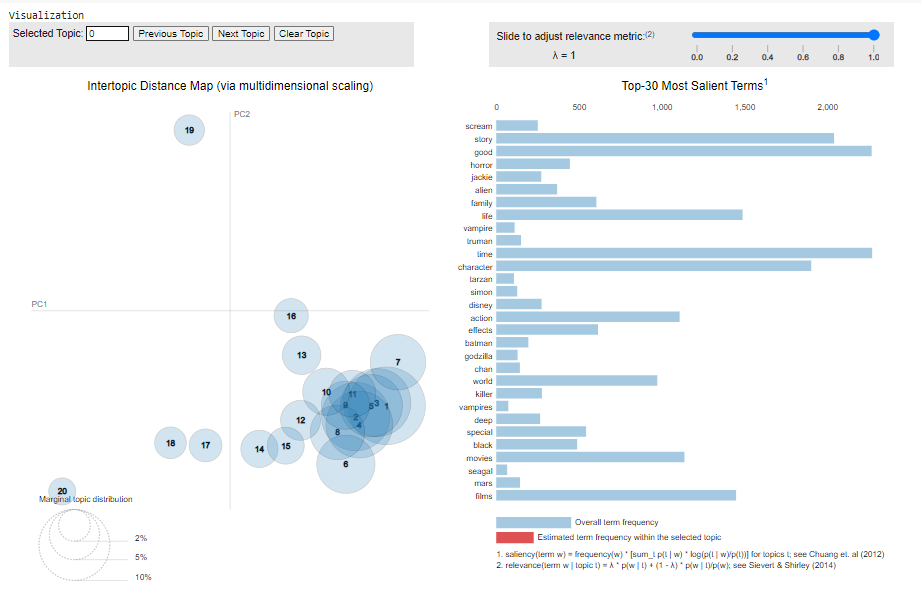


### Third Data Frame - 30 features data frame

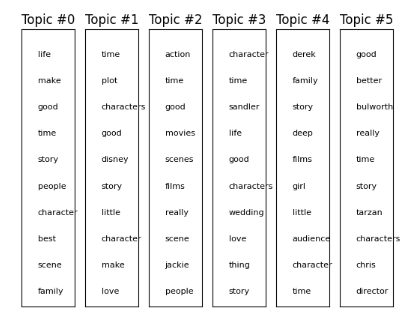


## **Part 3: Latent Dirichlet Allocation**

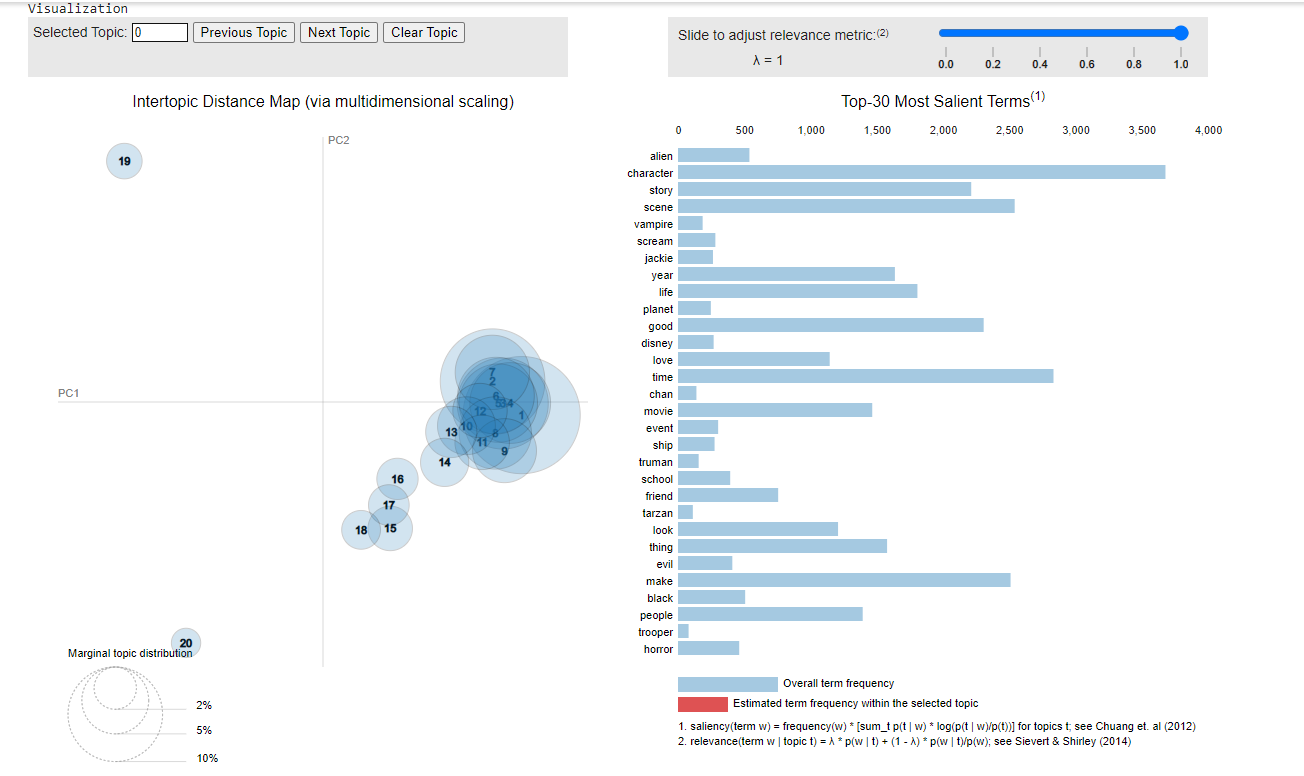
### First Dataframe - Cleaned Dataframe



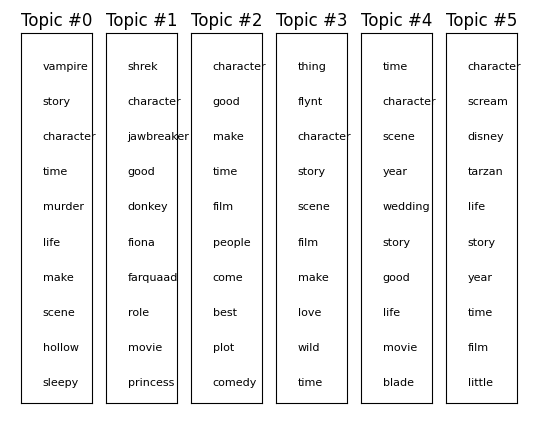
Words like good, glory, time, character, life, films, action, and movies are the Most Salient Terms with the highest frequencies. Words ‘life, good, time’ belong to many topics.



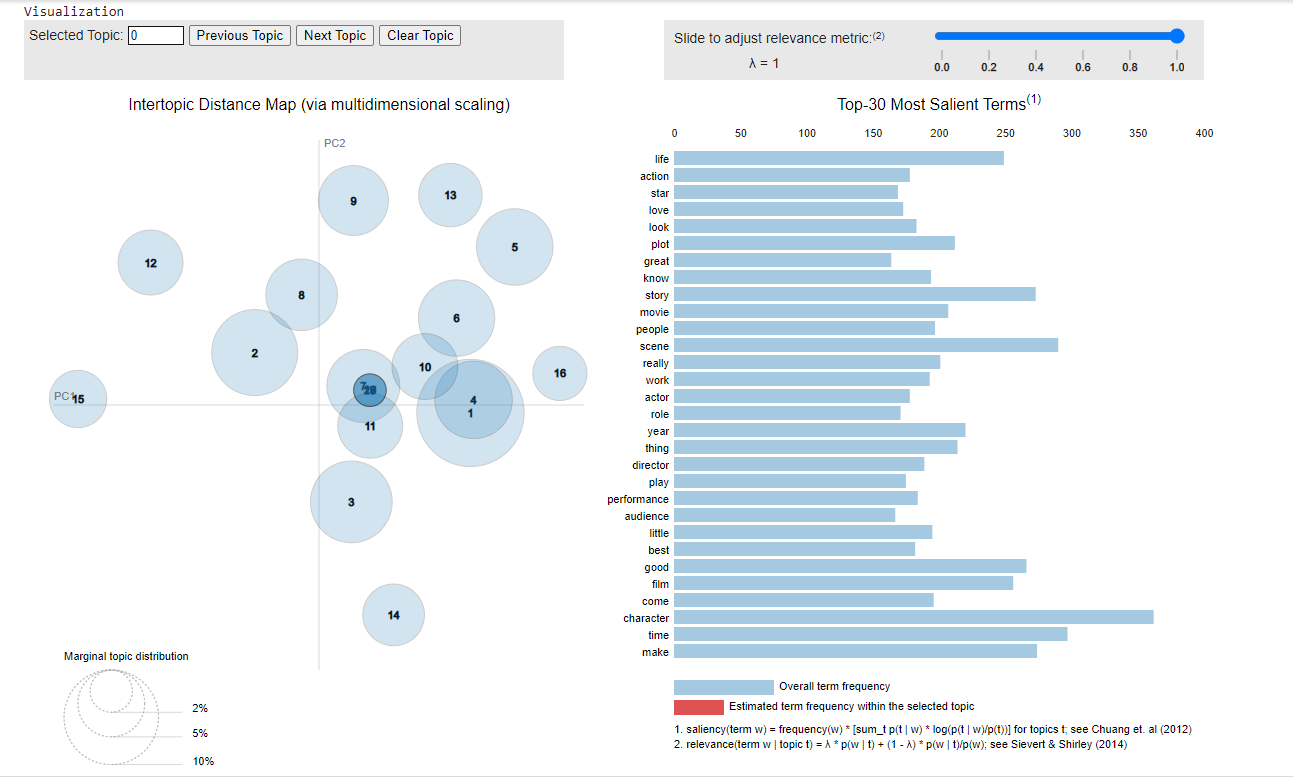
### Second Data Frame - Lemmatized DF



Words like character, time, scene, make, planet and story are the Most Salient Terms with the highest frequencies. Words ‘make, movie’ belong to many topics.



### Third Data Frame - 30 features data frame



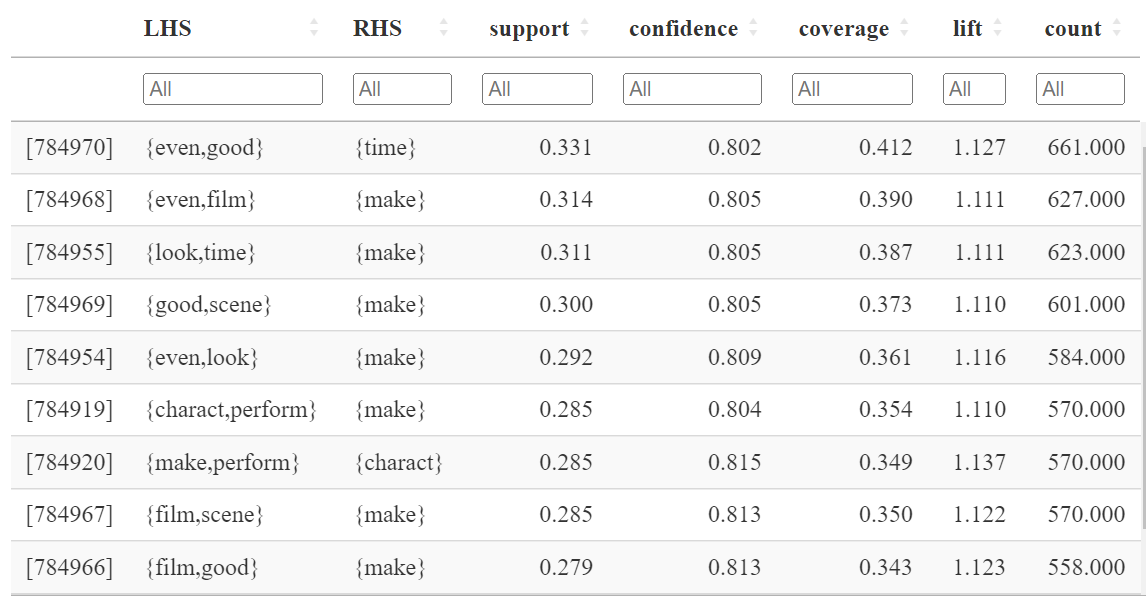
Words like character, time, scene, make, good and story are the Most Salient Terms with the highest frequencies. Words ‘life, love, year’ belong to many topics.



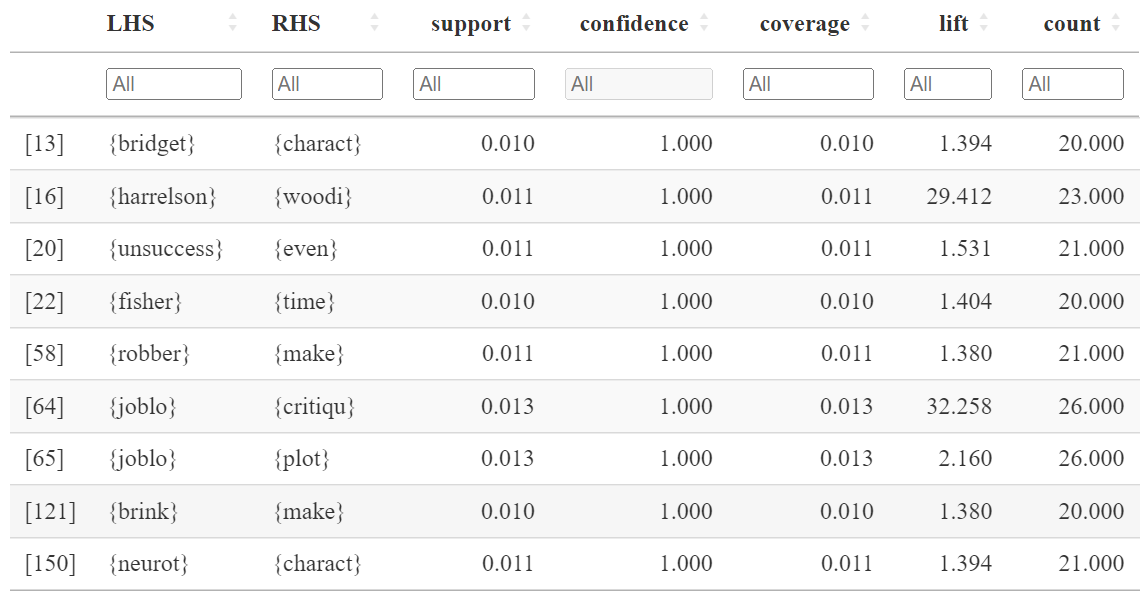
## **Part 4: Association Rule Mining**

### First Dataframe - Cleaned Dataframe

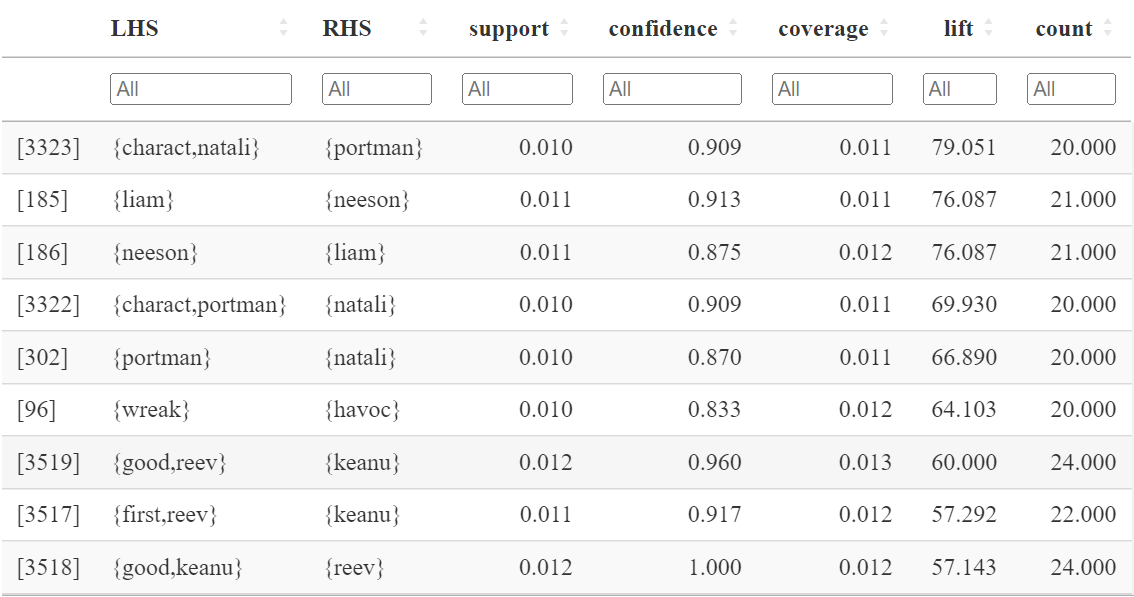
#### top 10 rules for support



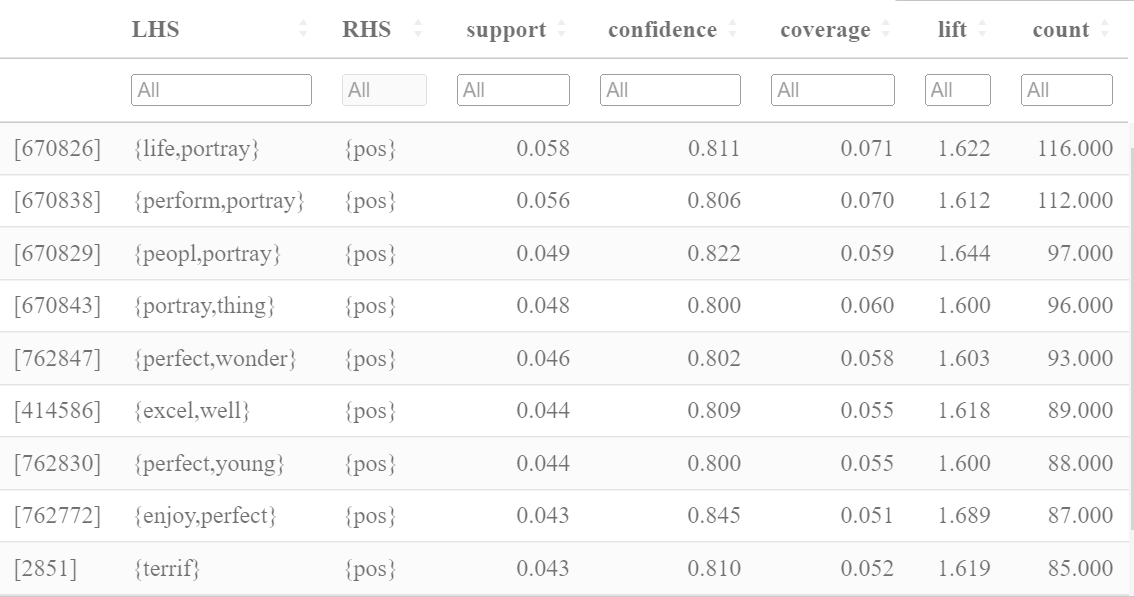
#### the top 10 rules for confidence



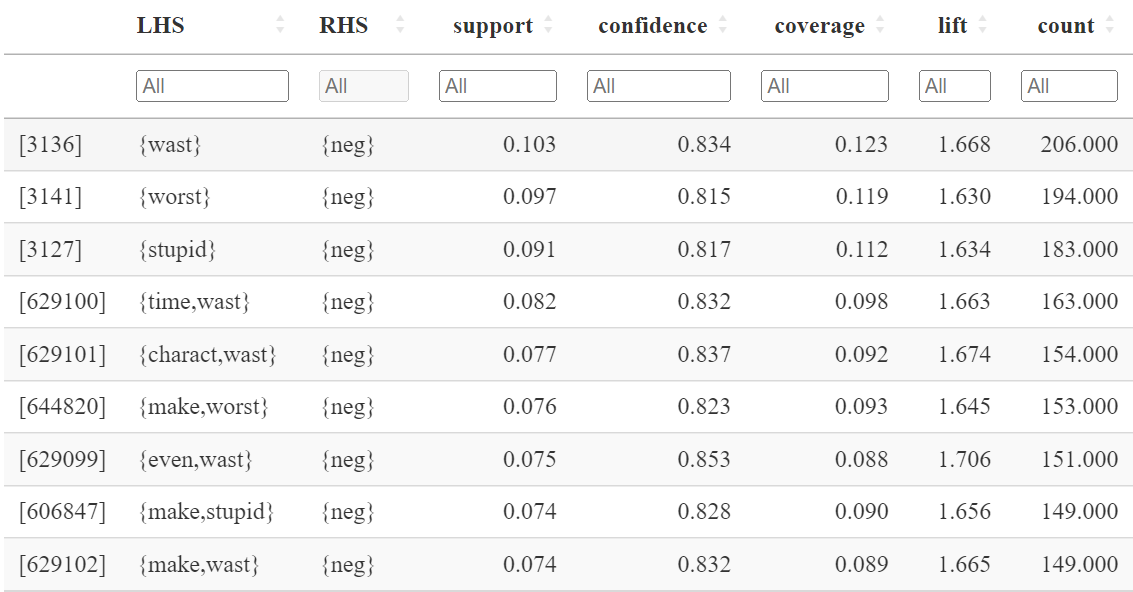
#### the top 10 rules for lift.



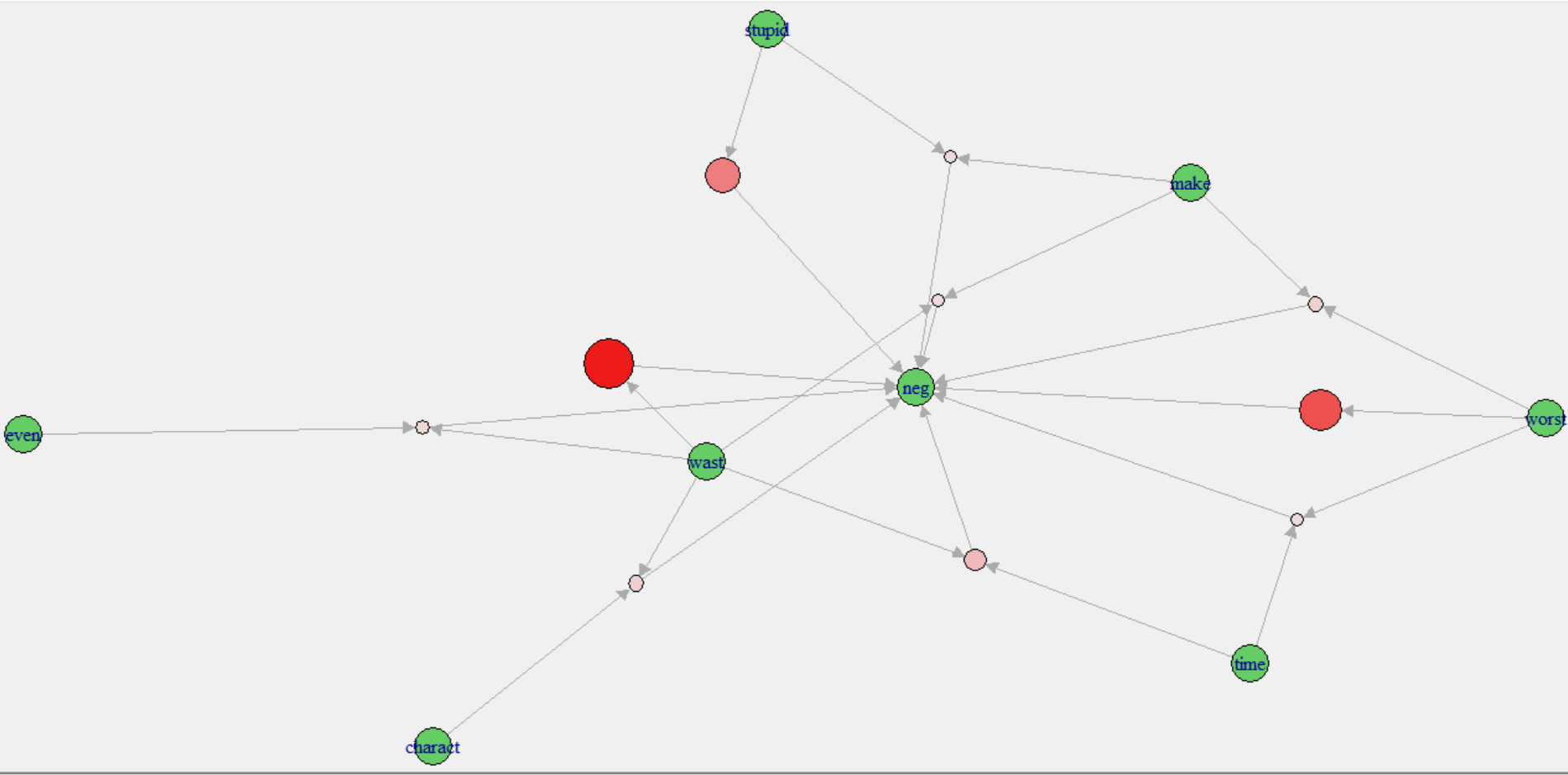
#### top 10 rules with "pos" as the right-hand side by support



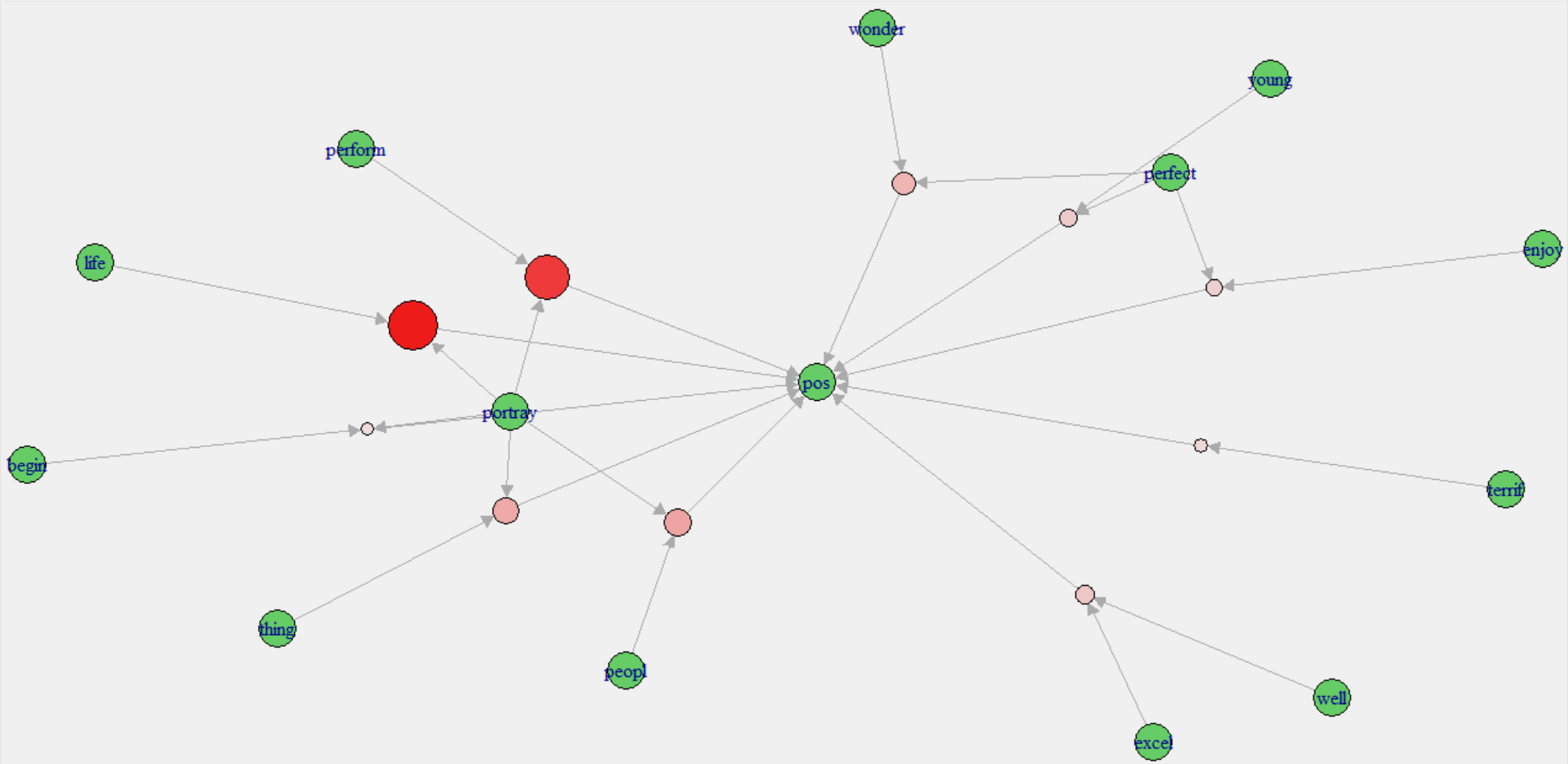
#### top 10 rules with "neg" as the right-hand side by support



#### network visualization for top 10 negative rules with support



#### network visualization for top 10 positive rules with support

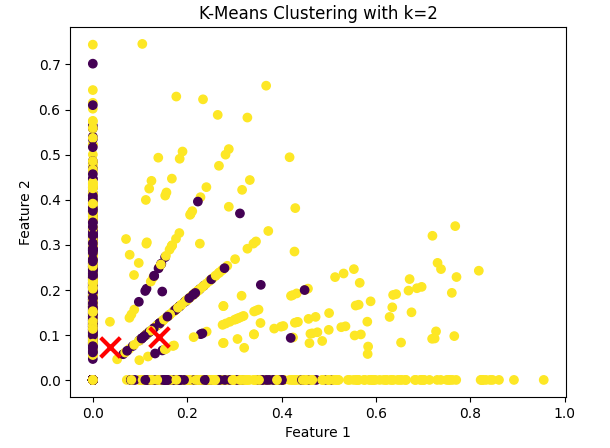


Words like perform, begin, excel, well, wonder, life, enjoy, perfect, and young are highly associated with positive reviews, and Words like stupid, wast, and worst are highly associated with negative reviews. The top 10 rules for support indicate the frequency with which each combination of items appears together in the dataset. Higher support values suggest that the items are frequently found together in the same review. The first rule in the top 10 rules (support) with LHS "even, good" and RHS "time" has a support of 0.331. This means that 3.3% of all transactions contain both "even" and "good", and among those transactions, 80.2% of them also contain "time". The lift value of 1.12 indicates that "even" and "good" occur together 1.12 times more frequently than if they were independent. The confidence of 0.802 shows that among all transactions that contain "even" and "good", 80.2% of them also contain "time". The coverage of 0.412 means that "even, good" appears in 41.2% of all transactions, and the count of 661 shows that this rule appears in 661 transactions.

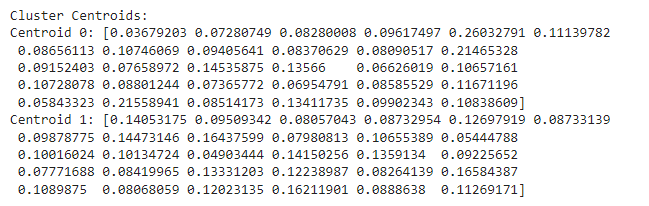
## **Part 5: Clustering**

### For the third data frame - 30 features data frame

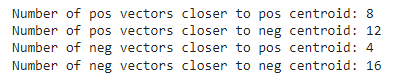
#### Visualization for k – means with k = 2



#### centroids for the two clusters



#### Euclidean distance between your 20 pos and 20 neg test vectors and the two centroids



The centroid for cluster 0 shows that the most important features (or words) for this cluster are "film", "character", "story", "acting", "good", "time", "make", "like", "great", "see", "well", "one", "really", "love", "perform", "life", "get", "also", "much", "play", "director", "world", "end", "could", "best", "find", "never", "us", "give", and "way". These words indicate that cluster 0 represents reviews with positive sentiments.

The centroid for cluster 1 shows that the most important features (or words) for this cluster are "film", "story", "character", "make", "like", "time", "one", "see", "acting", "good", "really", "well", "get", "much", "even", "director", "plot", "never", "end", "bad", "way", "scene", "people", "think", "great", "could", "say", "us", "go", and "watch". These words indicate that cluster 1 represents reviews with negative sentiments.

The number of positive and negative vectors closer to each centroid also provides insights into the sentiment of the movies in each cluster. In this case, there are eight positive vectors closer to the positive centroid (Centroid 1) and four negative vectors closer to the positive centroid, indicating that the movies in Centroid 1 are more likely to have positive sentiments. Similarly, there are 12 negative vectors closer to the negative centroid (Centroid 0) and 16 positive vectors closer to the negative centroid, indicating that the movies in Centroid 0 are more likely to have negative sentiments.

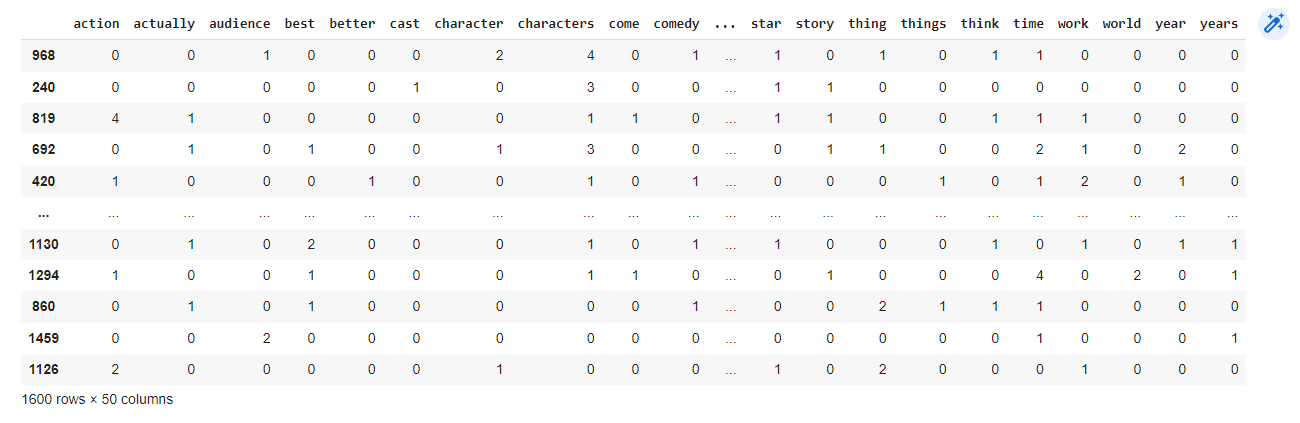
A creative visualization to represent this output could be a scatter plot with two different colored clusters (yellow and purple) representing the two centroids, where the size of the markers represents the number of positive or negative vectors closer to each centroid.

## **Part 6: Naïve Bayes, Decision Trees, and SVMs**

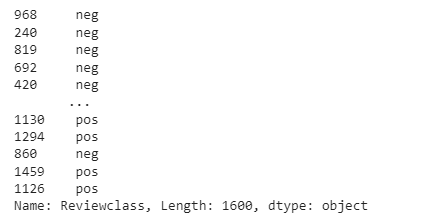
### Train and Test Data splitting

#### First Dataframe - Cleaned Dataframe

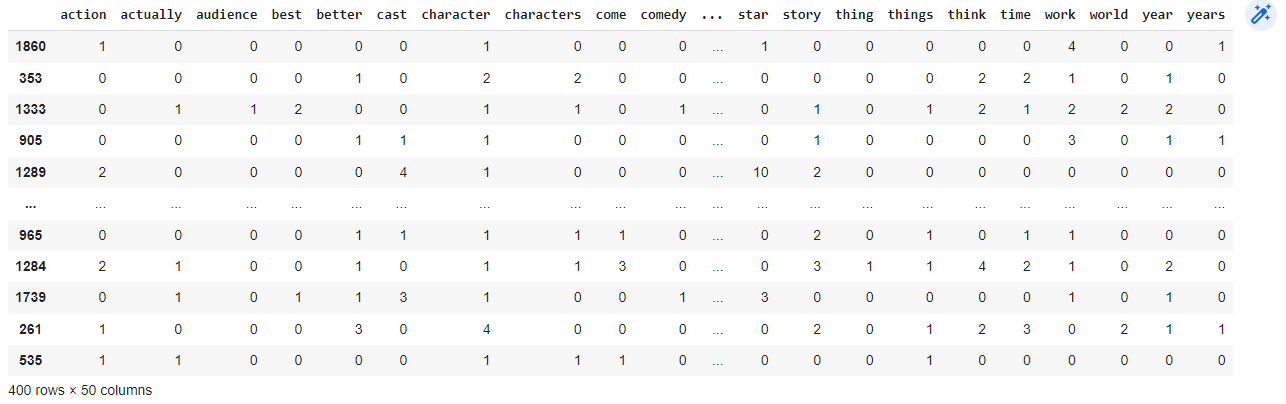
* + 1. Part 1: Training Data



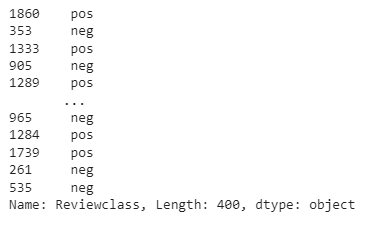
* + 1. Part 2: Training Labels



* + 1. Part 3: Testing Data

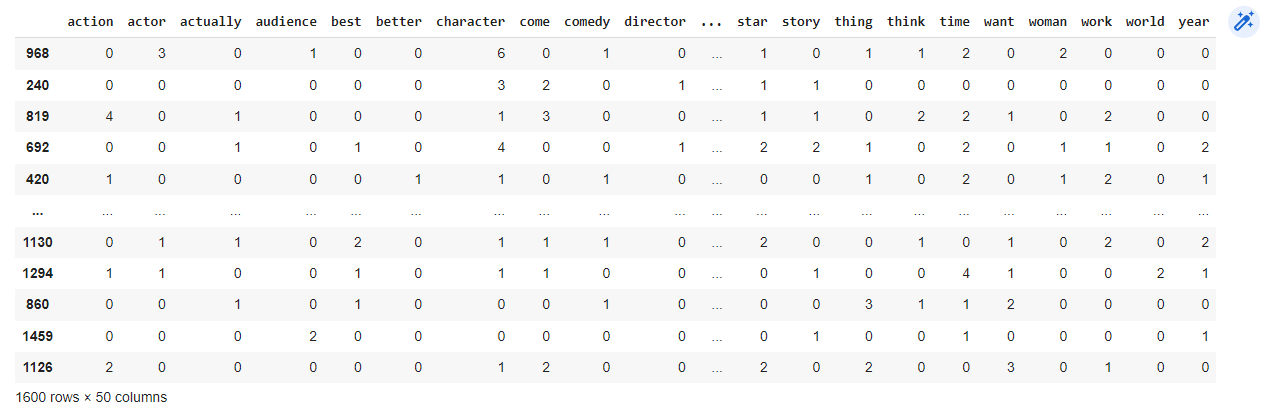


* + 1. Part 4: Testing Labels

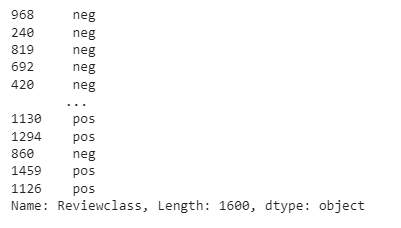


#### Second Data Frame - Lemmatized DF

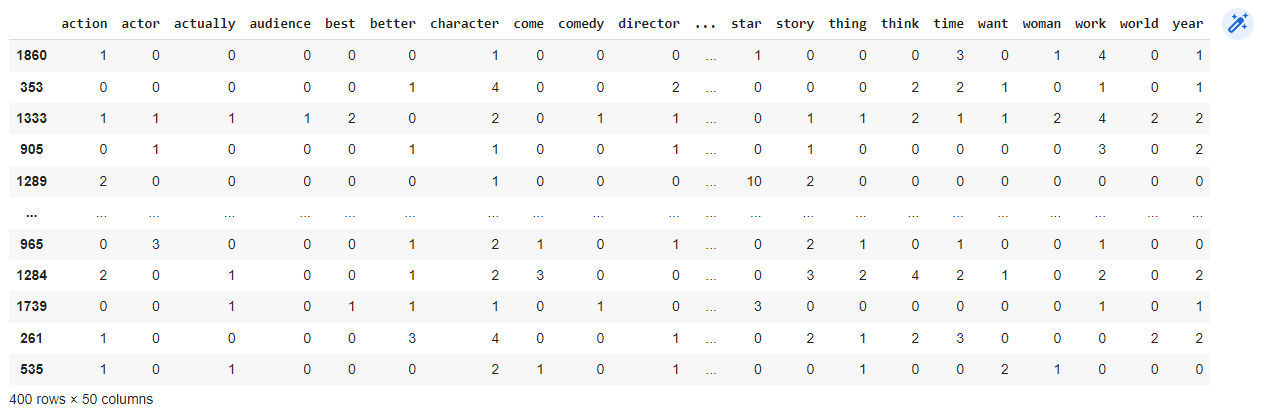
* + 1. Part 1: Training Data



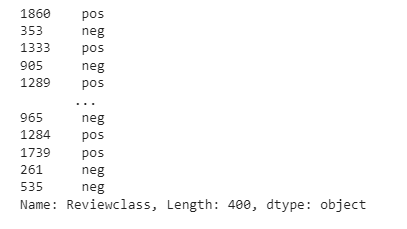
* + 1. Part 2: Training Labels



* + 1. Part 3: Testing Data

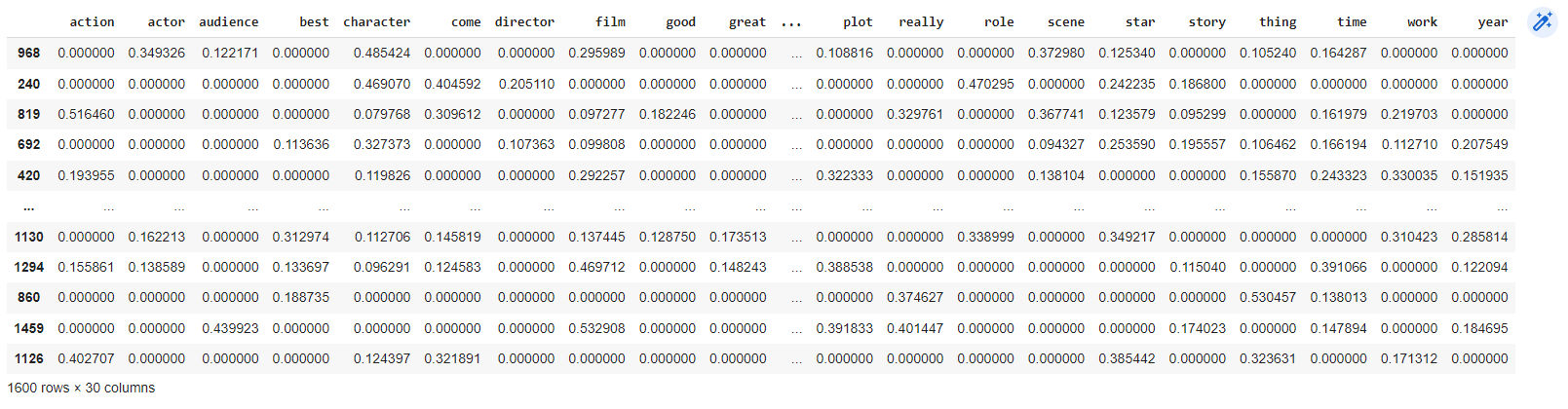


* + 1. Part 4: Testing Labels

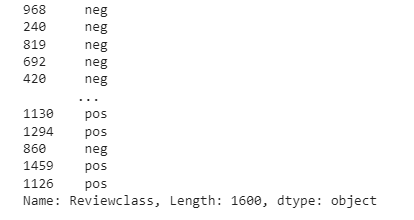


#### For the third data frame - 30 features data frame

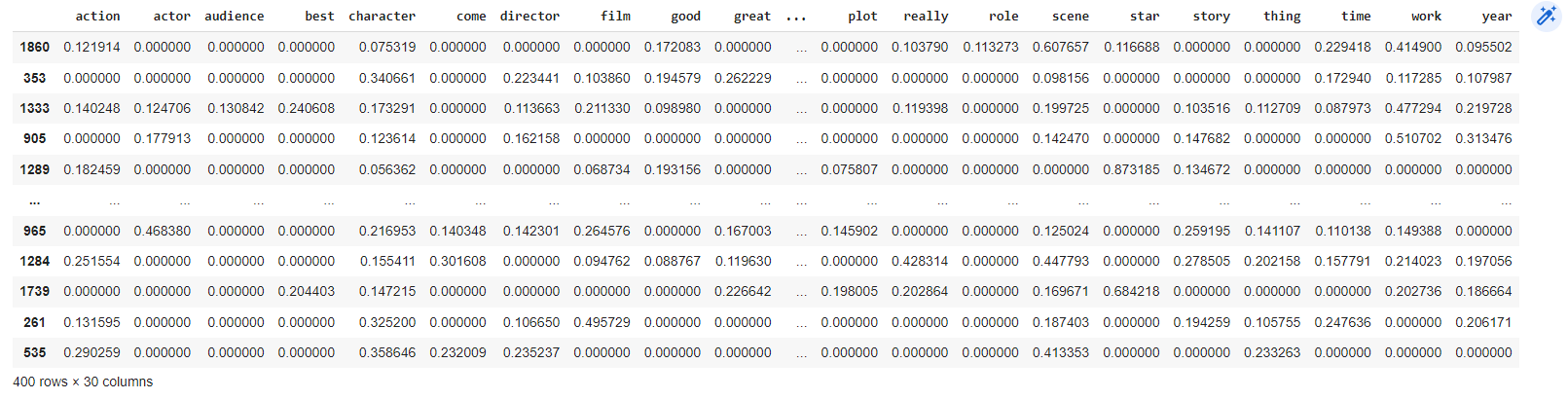
* + 1. Part 1: Training Data



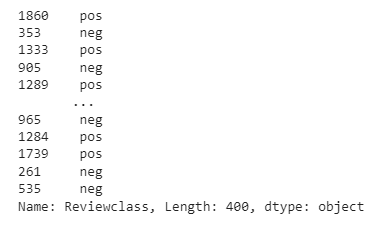
* + 1. Part 2: Training Labels



* + 1. Part 3: Testing Data



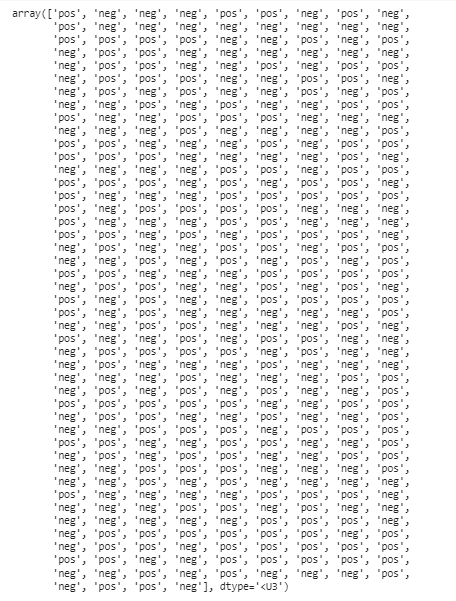
* + 1. Part 4: Testing Labels



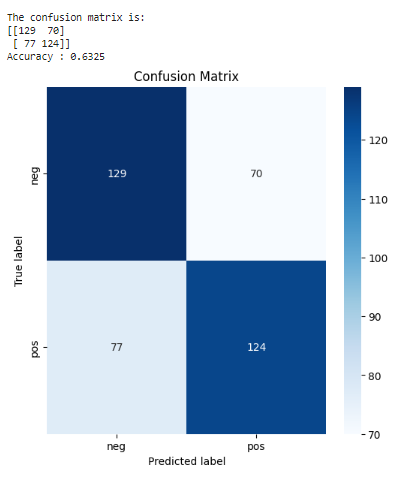
### Naïve Bayes (NB) and Results

#### First Dataframe - Cleaned Dataframe

* + 1. Predicted labels



* + 1. Confusion matrix and Accuracy

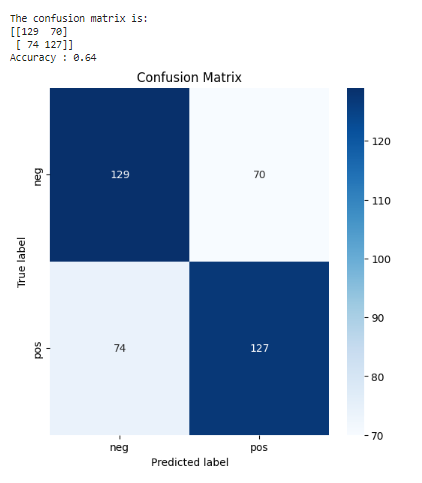


#### Second Data Frame - Lemmatized DF

* + 1. Predicted labels

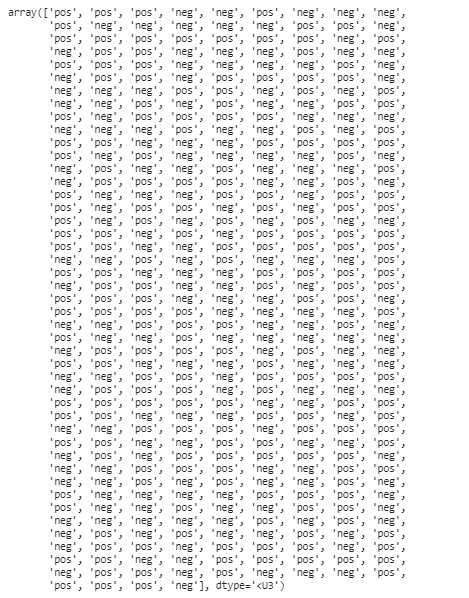


* + 1. Confusion matrix and Accuracy

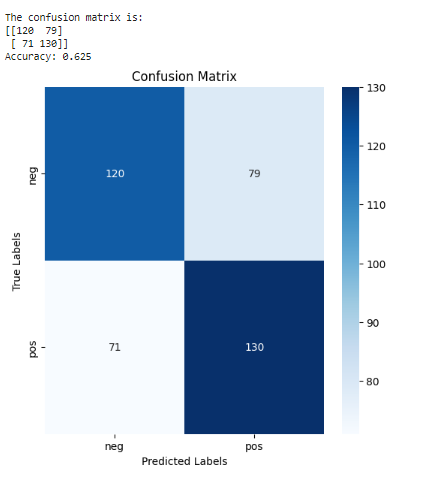


#### For the third data frame - 30 features data frame

* + 1. Predicted labels



* + 1. Confusion matrix and Accuracy



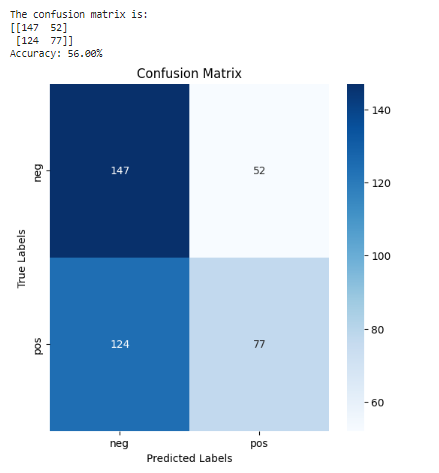
### Decision Trees and Results

#### First Dataframe - Cleaned Dataframe

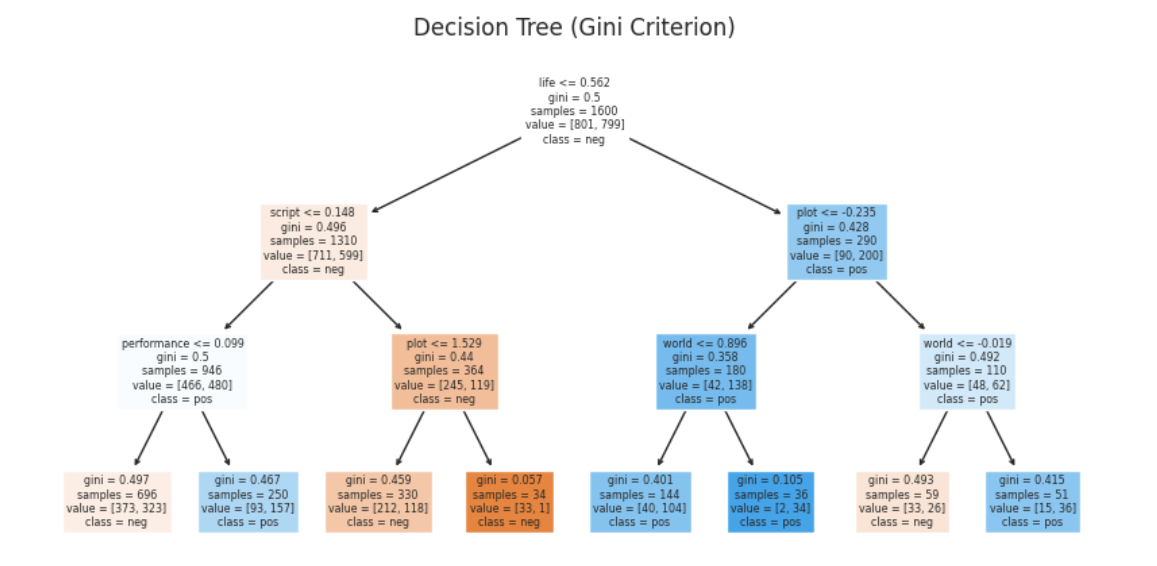
* + 1. Predicted labels



* + 1. Confusion matrix and Accuracy



* + 1. DT (at least 3 levels)

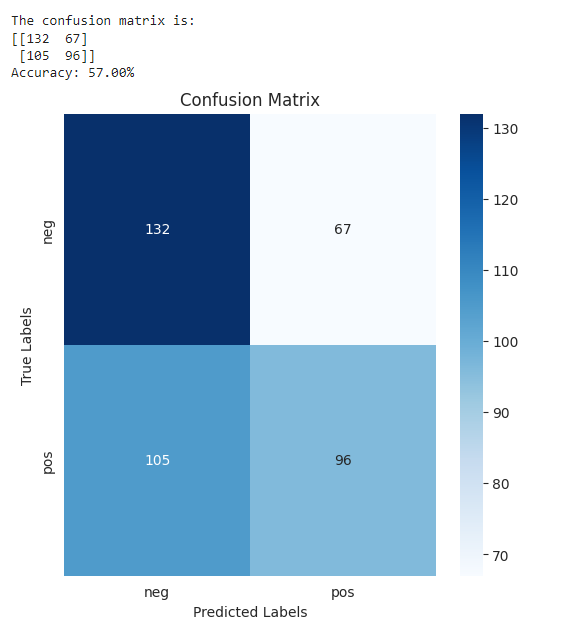


#### Second Data Frame - Lemmatized DF

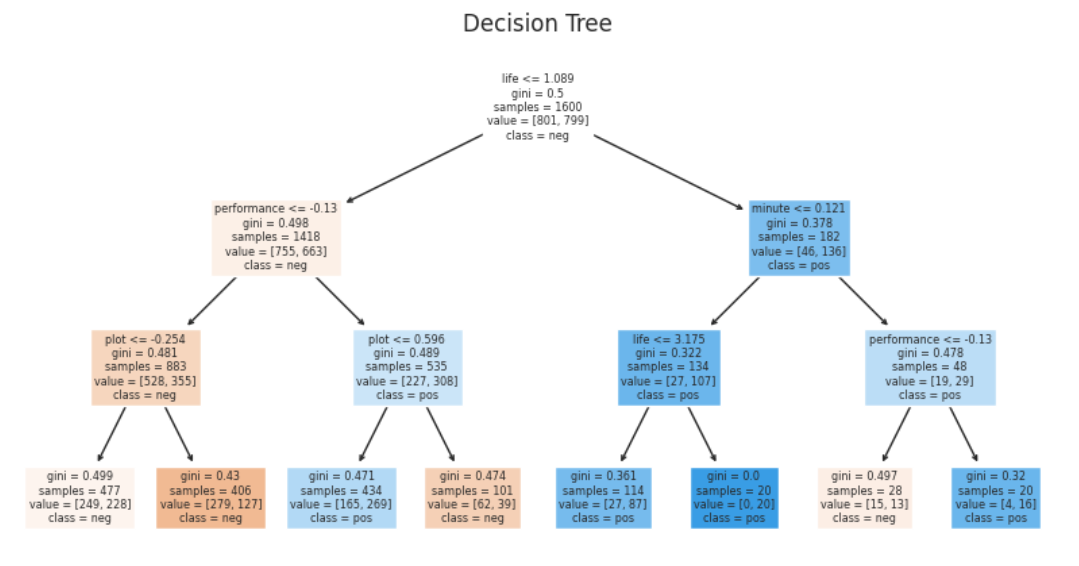
* + 1. Predicted labels



* + 1. Confusion matrix and Accuracy

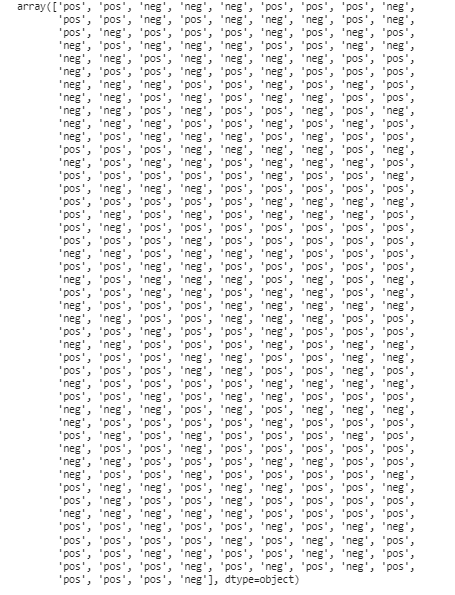


* + 1. DT (at least 3 levels)

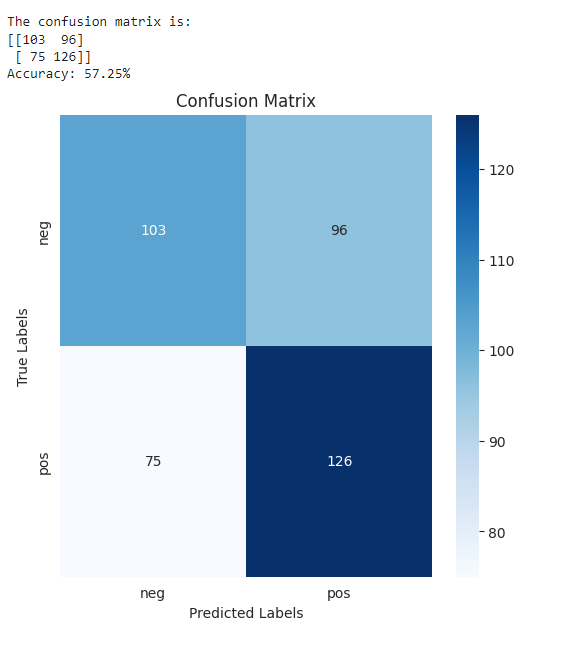


#### For the third data frame - 30 features data frame

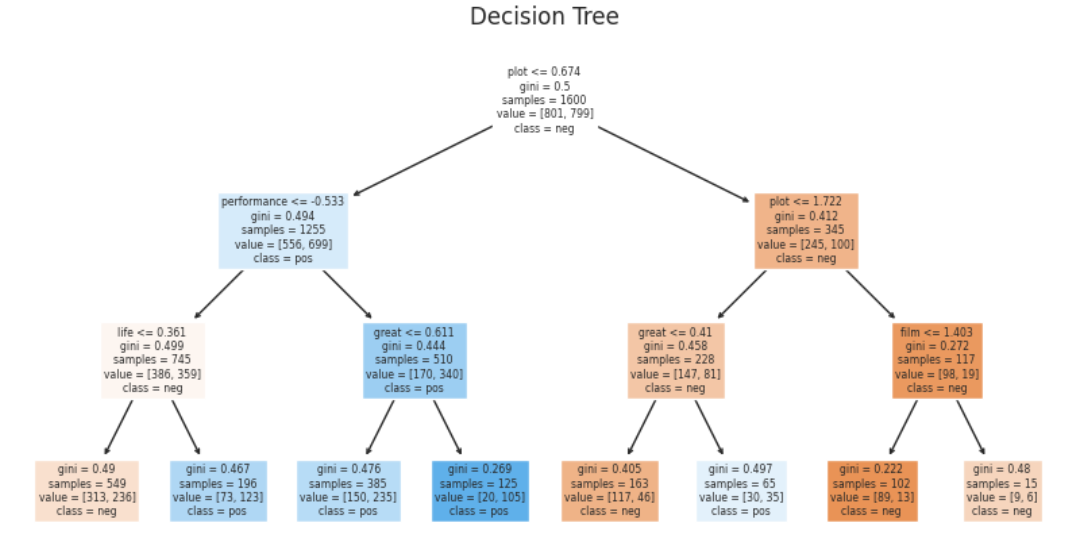
* + 1. Predicted labels



* + 1. Confusion matrix and Accuracy



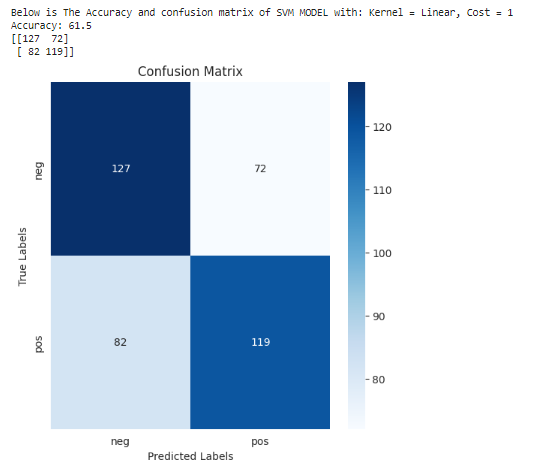
* + 1. DT (at least 3 levels)



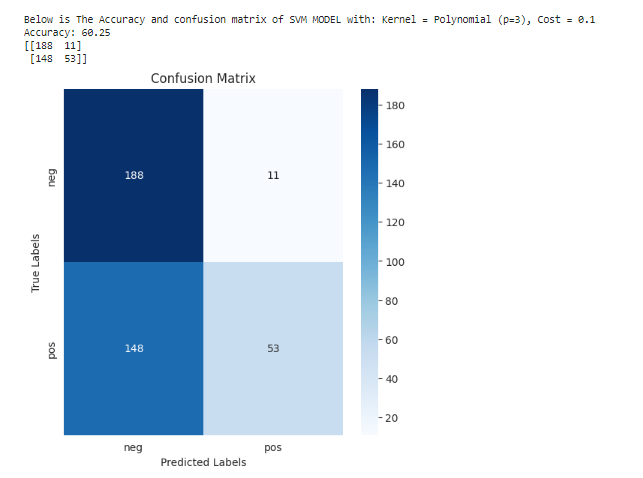
### Support Vector Machines (SVMs) and Results

#### First Dataframe - Cleaned Dataframe

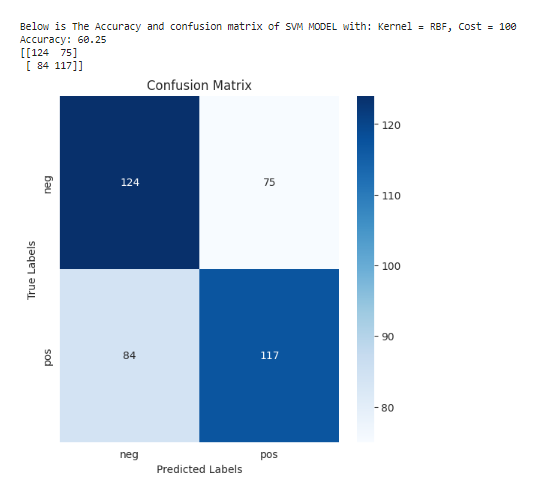
* + 1. 'Kernel = Linear, Cost = 1'



* + 1. 'Kernel = Polynomial (p=3), Cost = 0.1'

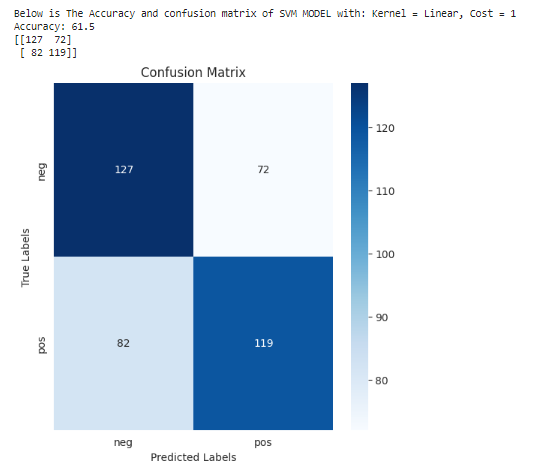


* + 1. 'Kernel = RBF, Cost = 100'

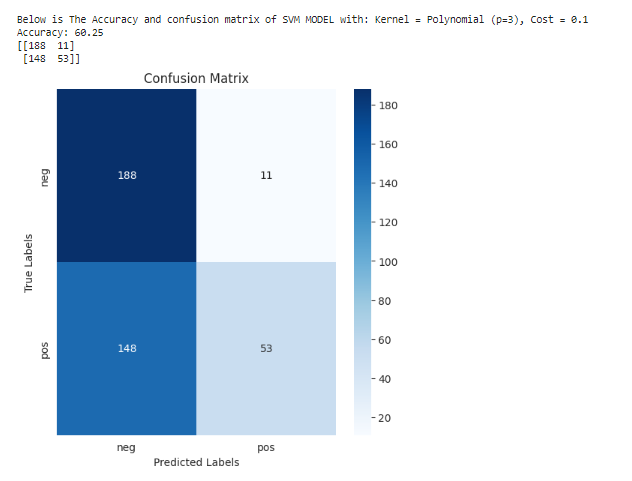


#### Second Data Frame - Lemmatized DF

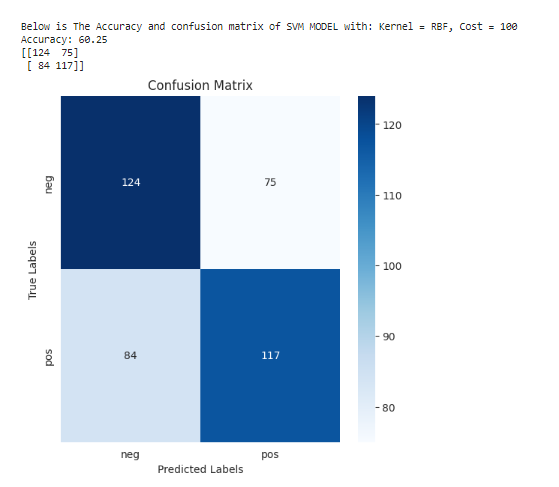
* + 1. 'Kernel = Linear, Cost = 1'



* + 1. 'Kernel = Polynomial (p=3), Cost = 0.1'

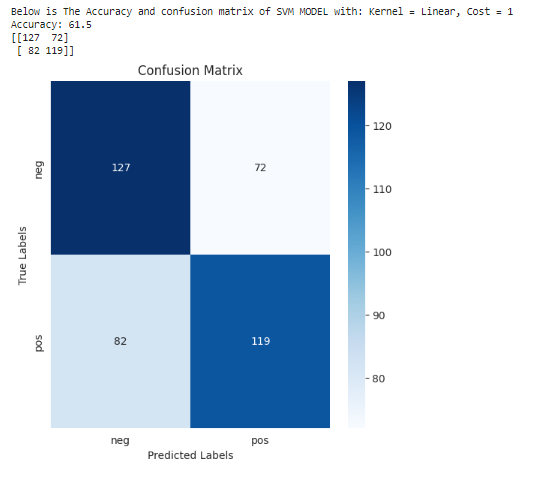


* + 1. 'Kernel = RBF, Cost = 100'

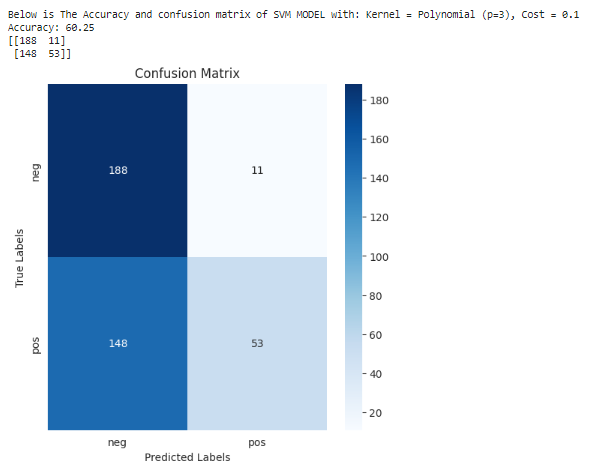


#### For the third data frame - 30 features data frame

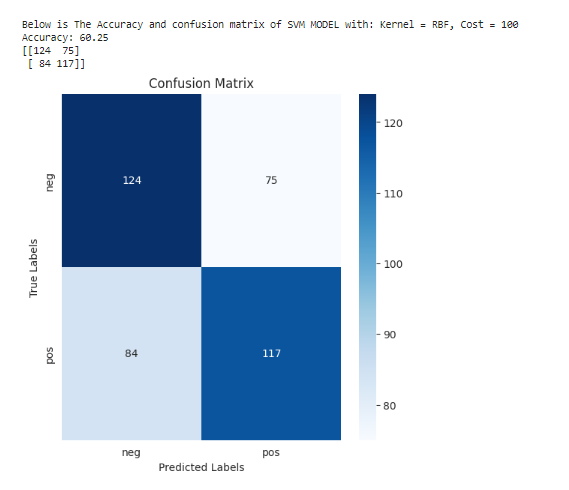
* + 1. 'Kernel = Linear, Cost = 1'



* + 1. 'Kernel = Polynomial (p=3), Cost = 0.1'



* + 1. 'Kernel = RBF, Cost = 100'



### Comparin all model results for all three dataframes and related parameters

| **Dataframe** | **Model** | **Params** | **Accuracy** |
| --- | --- | --- | --- |
| 1  (Cleaned DF) | Naïve Bayes (NB) | - | 63.25 |
| 1  (Cleaned DF) | Decision Trees (DT) | max\_depth=3, min\_samples\_split=5, min\_samples\_leaf=3 | 56.00 |
| 1  (Cleaned DF) | Support Vector Machines (SVMs) | Kernel = Linear, Cost = 1 | 61.5 |
| 1  (Cleaned DF) | Support Vector Machines (SVMs) | Kernel = Polynomial (p=3), Cost = 0.1 | 60.25 |
| 1  (Cleaned DF) | Support Vector Machines (SVMs) | Kernel = RBF, Cost = 100 | 60.25 |
| **2  (Lemmatized DF)** | **Naïve Bayes (NB)** | **-** | **64.00** |
| 2  (Lemmatized DF) | Decision Trees (DT) | max\_depth=3, min\_samples\_split=5, min\_samples\_leaf=3 | 57.00 |
| 2  (Lemmatized DF) | Support Vector Machines (SVMs) | Kernel = Linear, Cost = 1 | 61.5 |
| 2  (Lemmatized DF) | Support Vector Machines (SVMs) | Kernel = Polynomial (p=3), Cost = 0.1 | 60.25 |
| 2  (Lemmatized DF) | Support Vector Machines (SVMs) | Kernel = RBF, Cost = 100 | 60.25 |
| 3  (30 features DF) | Naïve Bayes (NB) | - | 62.5 |
| 3  (30 features DF) | Decision Trees (DT) | max\_depth=3, min\_samples\_split=5, min\_samples\_leaf=3 | 57.25 |
| 3  (30 features DF) | Support Vector Machines (SVMs) | Kernel = Linear, Cost = 1 | 61.5 |
| 3  (30 features DF) | Support Vector Machines (SVMs) | Kernel = Polynomial (p=3), Cost = 0.1 | 60.25 |
| 3  (30 features DF) | Support Vector Machines (SVMs) | Kernel = RBF, Cost = 100 | 60.25 |

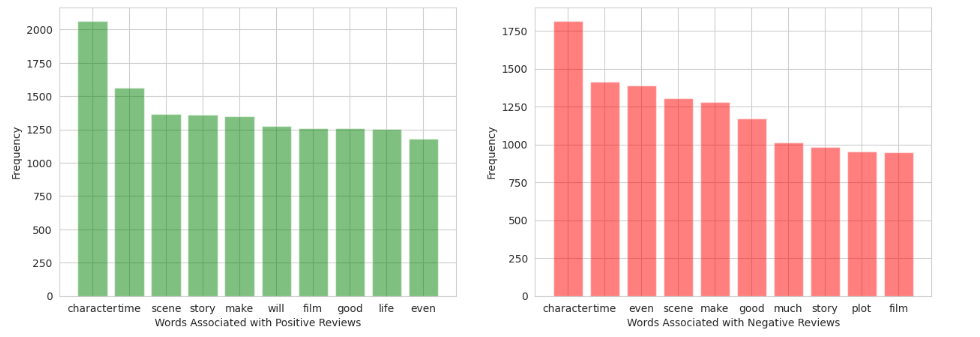
Based on the table, it appears that the Naïve Bayes model generally performed the best across all three data frames, with the highest accuracy being 64% on the Lemmatized DF. The Decision Trees model had the lowest accuracy among the three models, with a maximum of 57.25% accuracy on the 30 features DF. Regarding parameter tuning, the best parameters varied depending on the model and the specific data frame. For example, the SVM model with a linear kernel and a cost of 1 had the highest accuracy for both the cleaned and lemmatized data frames, while the SVM model with an RBF kernel and a cost of 100 had the highest accuracy for the 30 features DF. Overall, it seems that the choice of preprocessing method and feature selection had a greater impact on accuracy than the choice of model or parameters.

## **Part 7: Conclusions**

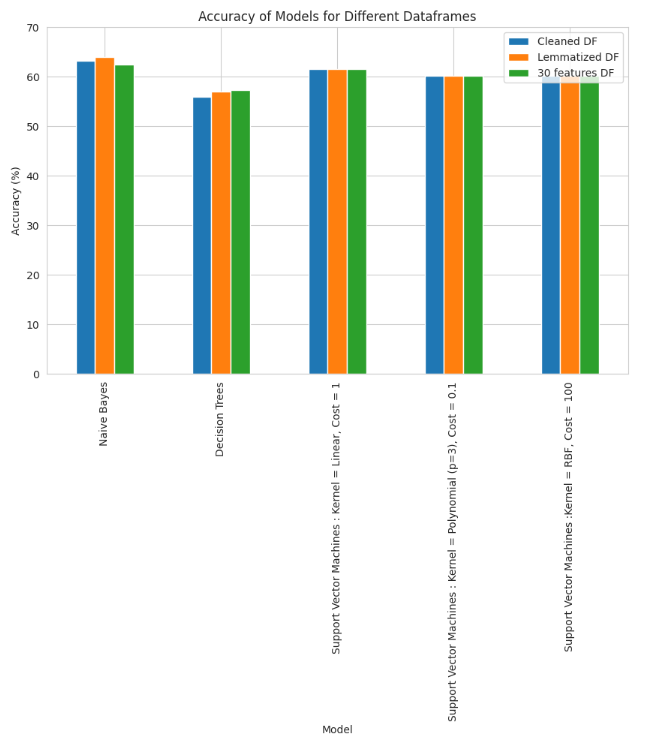
### 10 bullet points of key take-home messages

* 1. We used machine learning algorithms to predict the sentiment of movie reviews with up to 64% accuracy.
  2. Text preprocessing techniques like cleaning, lemmatization, and feature selection can significantly improve model performance.
  3. Naive Bayes algorithm showed the best performance across all three data frames.
  4. The Decision Trees and Support Vector Machines models had lower accuracies, ranging from 56% to 61.5%.
  5. Positive and negative reviews contain different words, indicating distinct language patterns for expressing sentiments.
  6. Common words like "movie," "film," and "character" are not strong indicators of sentiment and can be removed to improve model accuracy.
  7. Certain words like "scene," "story," "make," and "good" have a strong association with positive or negative sentiment and can be used to predict reviews.
  8. Network visualizations show the relationships between frequently co-occurring words, indicating their potential significance in sentiment analysis.
  9. Machine learning models can quickly analyze large volumes of text data, providing insights into customer sentiment and preferences.
  10. Further analysis, such as topic modeling or aspect-based sentiment analysis, can provide deeper insights into specific aspects of customer sentiment.

### “take-home message” visualizations



Character word is associated with both positive and negative reviews.



It clearly illustrates that Naive Bayes has the highest accuracy among all models.