Part 1 – Cluster Analysis

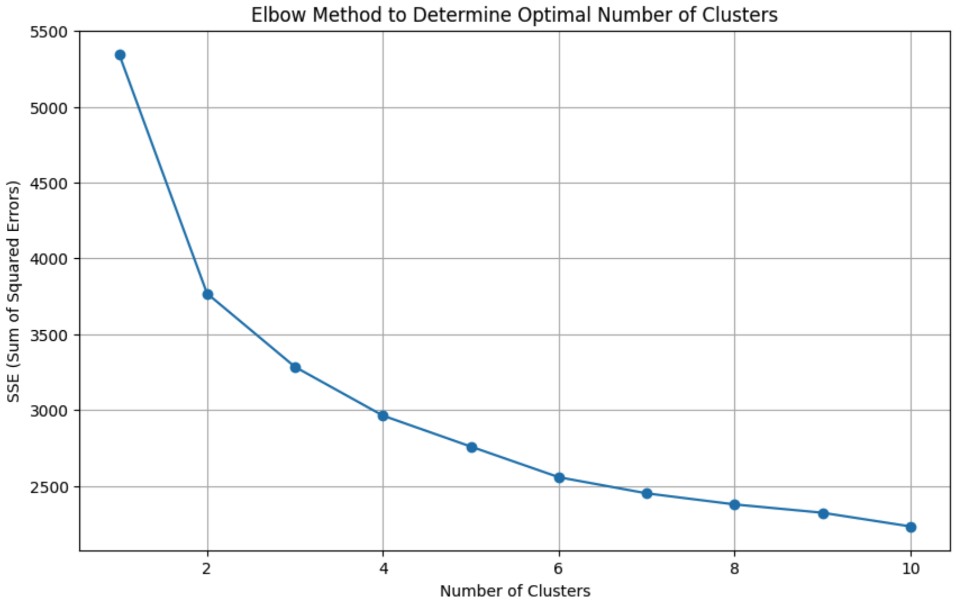
The primary objective of this part of the analysis was to apply clustering techniques to the IMDB dataset. The goal was to group movies that share common characteristics into clusters.

Preprocess Dataset:
In this step, we handle null values by imputing both categorical and numerical variables. And later we scaled the numerical data for better clustering.

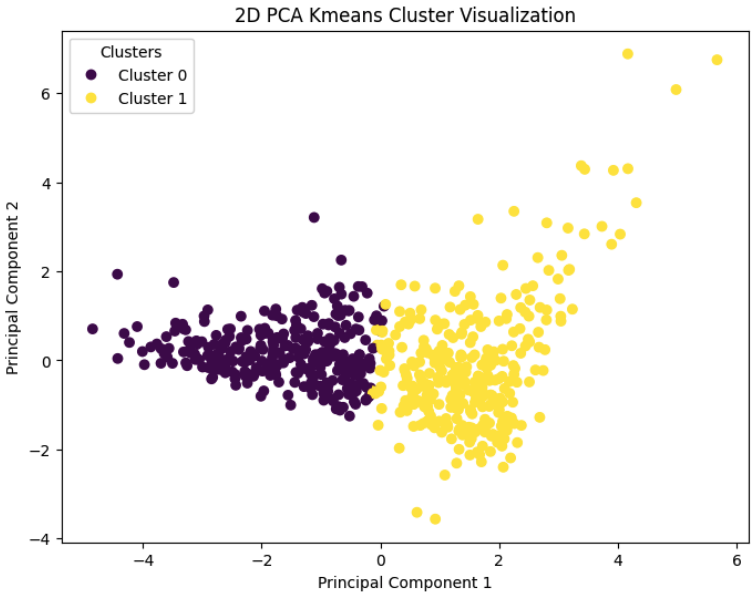
KMeans Clustering:

The Elbow Method was utilized to determine the optimal number of clusters for the KMeans algorithm.
We plotted the Sum of Squared Errors (SSE) against a range of possible cluster numbers and identified the 'elbow point', where the rate of decrease in SSE sharply changes.

We found for our dataset optimum cluster is 2.



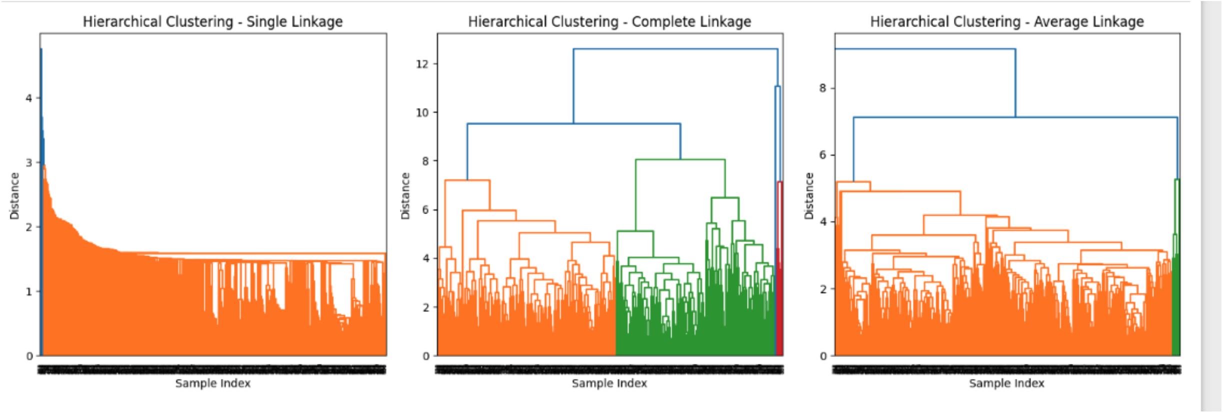
After determining the optimal number of clusters, the KMeans clustering algorithm was applied to the dataset and we visualized the clusters by doing PCA.



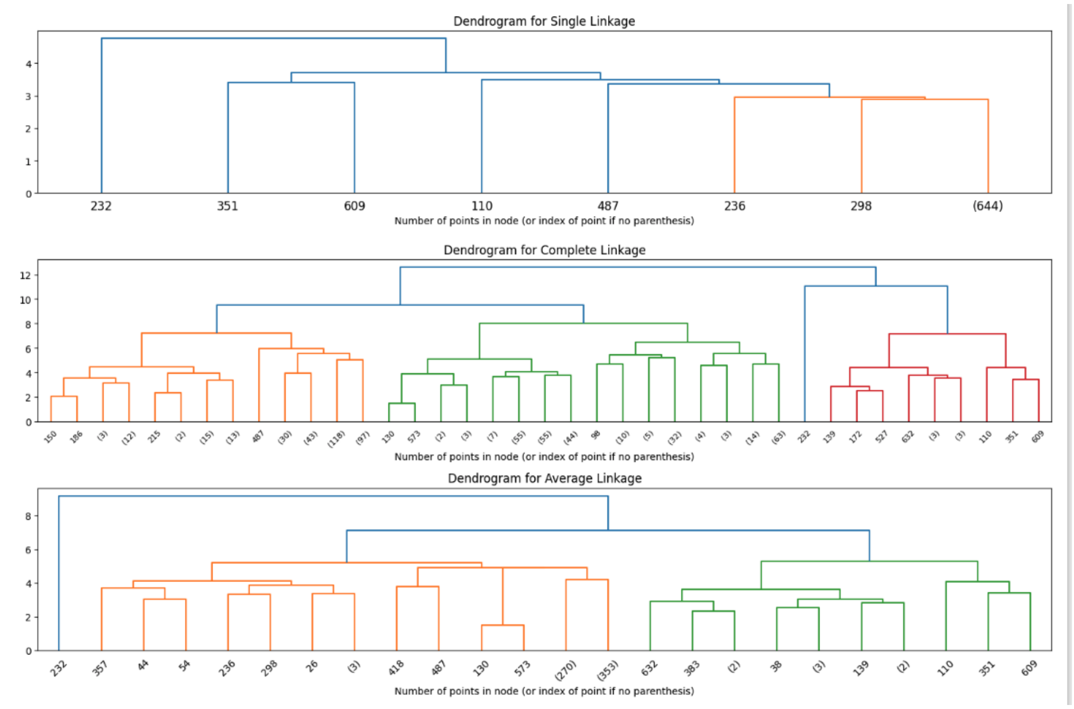
Hierarchical Clustering:

Different linkage methods, namely single, complete, and average, were employed in hierarchical clustering.

Dendrograms were generated for each linkage method to visualize the hierarchical structure of the movie clusters.



To simply the plot we plot up to level 5.



Part 2 – TEXT MINING

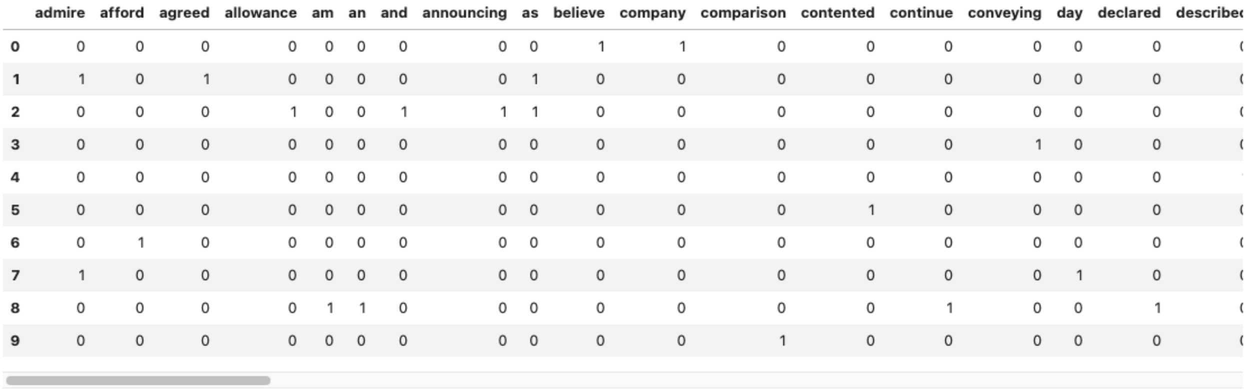
Count Vectorization

This process involves converting the text data into a matrix of token counts. Each row in the matrix represents a document from the dataset, and each column represents a unique word in the dataset. The value in each cell is the frequency of the word in the corresponding document.

TF-IDF Vectorization

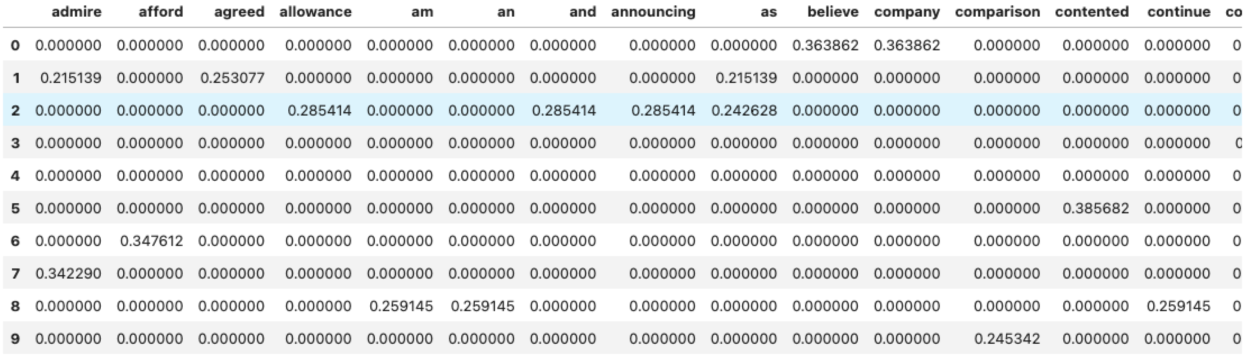
TF-IDF stands for Term Frequency-Inverse Document Frequency. This process involves converting the text data into a matrix where each row represents a document, and each column represents a unique word. The value in each cell is the TF-IDF score of the word in the corresponding document. TF-IDF increases the importance of words that are unique to a document and decreases the importance of words that are common across multiple documents.

Here is the result of Count Vectorizer:



	admire	afford	agreed	allowance	am	an	and	announcing	as	believe	company	comparison	contented	continue	conveying	day	declared	describe
0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0
1	1	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
2	0	0	0	1	0	0	1	1	1	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
5	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
6	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
8	0	0	0	0	1	1	0	0	0	0	0	0	0	1	0	0	1	0
9	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0

Here is the result for TF-IDF Vectorizer



	admire	afford	agreed	allowance	am	an	and	announcing	as	believe	company	comparison	contented	continue	co
0	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.363862	0.363862	0.000000	0.000000	0.000000	0
1	0.215139	0.000000	0.253077	0.000000	0.000000	0.000000	0.000000	0.000000	0.215139	0.000000	0.000000	0.000000	0.000000	0.000000	0
2	0.000000	0.000000	0.000000	0.285414	0.000000	0.000000	0.285414	0.285414	0.242628	0.000000	0.000000	0.000000	0.000000	0.000000	0
3	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0
4	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0
5	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.385682	0.000000	0
6	0.000000	0.347612	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0
7	0.342290	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0
8	0.000000	0.000000	0.000000	0.000000	0.259145	0.259145	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.259145	0
9	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.245342	0.000000	0.000000	0

Usage of TF-IDF

- In a large text corpus, TF-IDF can identify the most relevant words in each document and highlight keywords that are unique to that document.

- It can also be used as information retrieval systems, such as search engines. When a user inputs a query, the system can rank documents by the TF-IDF scores of the query terms to retrieve the most relevant documents.

- In machine learning, TF-IDF is used as a feature extraction technique to convert text data into a format that algorithms can work with. By representing documents through their TF-IDF vectors, it becomes possible to apply classification or clustering algorithms, like KMeans or hierarchical clustering, to group or classify documents based on their content.

- By considering the inverse document frequency, TF-IDF naturally filters out common words that appear in many documents and are less informative (like 'the', 'is', 'and', etc.) or stopwords. This helps in focusing on words that are more unique and descriptive of each document.

Part 3 – ARTIFICIAL NEURAL NETWORK (ANN)

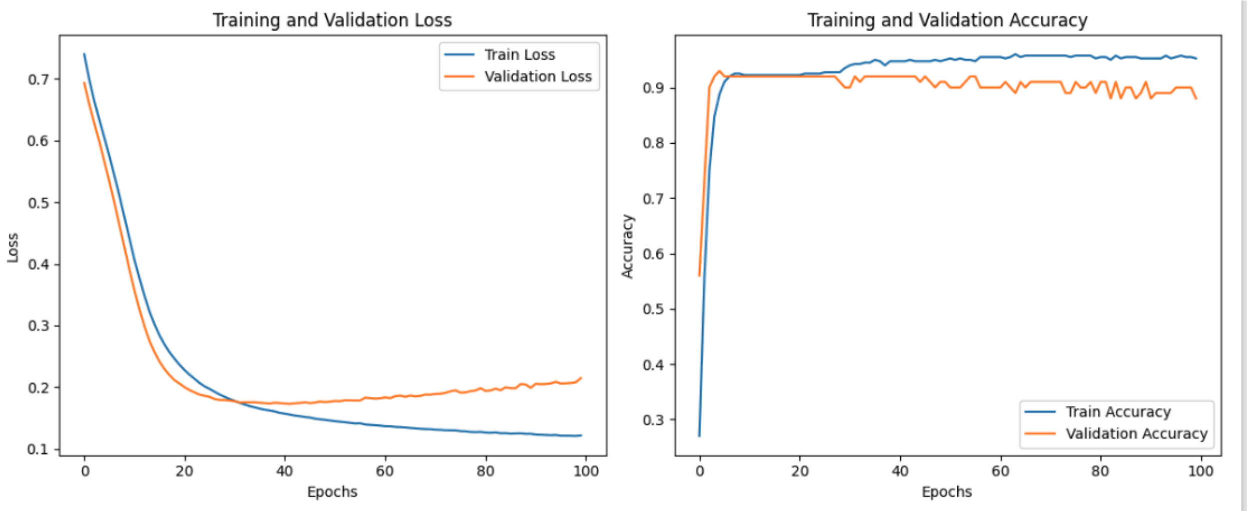
Preprocessing:

- We scaled the numerical data for better classification.
- We then convert the label column to categorical by converting the probability to the binary label.

We split the dataset to 80:20.

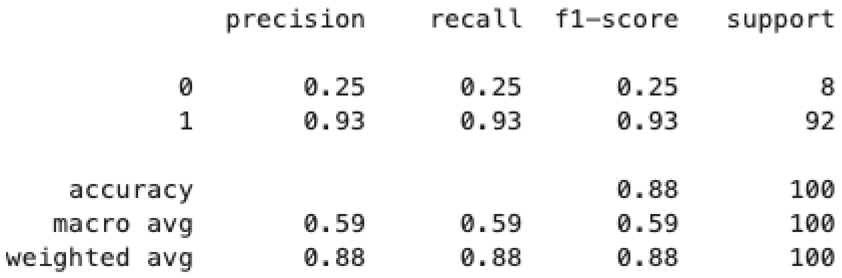
Modeling:

- We then define our tensorflow NN model.
- We visualize the loss and accuracy for training and testing set.



We found the model has decent accuracy metrics which is 0.88 on the test set.

Classification report:

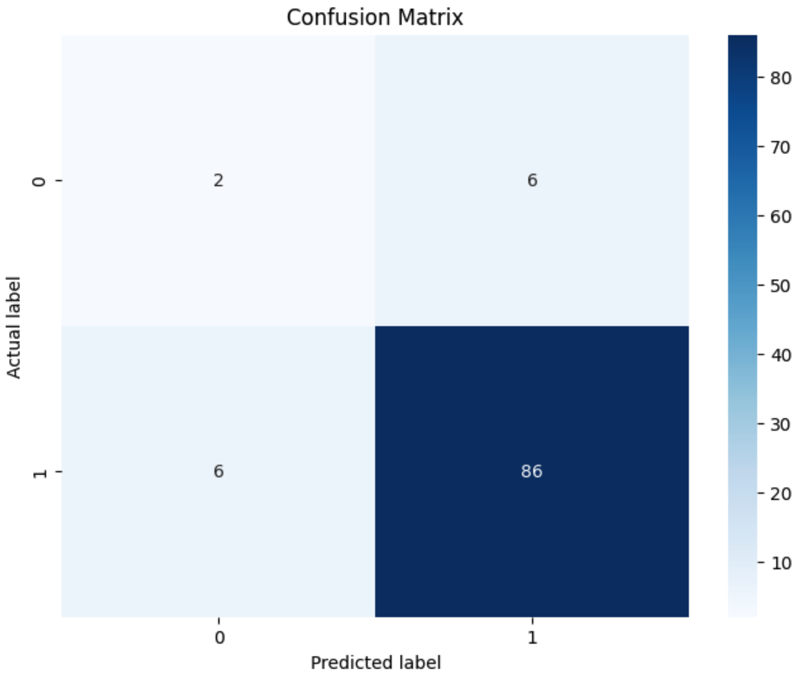


```
              precision    recall  f1-score   support

           0       0.25      0.25      0.25         8
           1       0.93      0.93      0.93        92

    accuracy                           0.88       100
   macro avg       0.59      0.59      0.59       100
weighted avg       0.88      0.88      0.88       100
```

Confusion Matrix:


Confusion Matrix
Actual label
Predicted label