Toronto Metropolitan University

Machine Learning - CPS803

Assignment 4

Recommendation System Using Clustering By Text Similarity

Peter Lee - 501088502

Due date: Nov 27, 2024

Background

The following paper used data “movies_metadata.csv” from “The Movies Dataset” published in Kaggle by Rounak Banik [2]. This data has 45466 rows and 24 attributes ranging from movie budget, IMDB ID, and spoken languages. The goal of this paper is to create a movie recommendation system using only the “overview” attribute which is a quick summary of the movie. The interesting part of this data is that every movie entry has an “imdb_id”. The website IMDB is a site that reviews movies and recommends other movies which the user may be interested in. The website URL is always formatted as…

“https://www.imdb.com/title/<imdb_id>/”

Thus, this paper will suggest a new form of recommendation system to sites like IMDB which uses text to explain the content of their data. Therefore, by utilizing the URL format and the “imdb_id” attribute, users can visit other IMDB related sites based on text similarity clustering.

For machine learning, the concepts needed are different clustering methods like k-means and agglomerative clustering as well as the elbow method for model evaluation and hyperparameter tuning. This paper also uses preprocessing methods like PCA for dimensionality reduction and simple management of NaN and duplicate data. As for the NLP concepts, the data preprocessing includes tokenization, stemming, contractions, stopword removal, character removal, and vectorization [4].

As a note for this paper, due to the restrictions on the machine learning assignment for a 5 minute limit on runtime for the script, overall results for the scripts will be poor.

Methods

Data Preprocessing

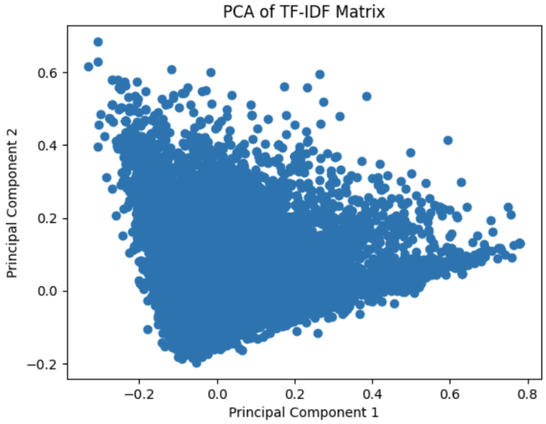
The first step is data retrieval, exploration, and preprocessing. The preprocessing step will ignore NLP for now. As already mentioned, there are 45466 rows and 24 attributes. However, this paper will only be using two attributes, “imdb_id” for IMDB’s URL path and “overview” for a summary of the movie. In the preprocessing stage, it is important that all data has a frequency of one. This means that there are no duplicates to the movie data entries. After finishing this preprocessing stage, only 44282 rows remain.

Natural Language Processing

The next step is natural language processing. Firstly, based on research for NLP, the use of English only for the overview is preferred due to how different languages may overpower the data as a unique sentence [3]. This was fixed by removing entries that are not english through the detection of characters that are not a part of ASCII. The other steps on NLP are removing extra spaces, removing special characters and numbers, lowercasing all characters, expanding contractions, removing stopwords, and stemming words. The main idea for all of these steps is to tokenize the overview text by turning a string of characters into understandable words with great significance as a feature [1, 5]. These tokens are then passed to a vectorization algorithm which determines the frequency of each word in the dataset. Vectorization is the process of converting words into vectors. For this paper, TF-IDF was used. TF-IDF or Term Frequency-Inverse Document Frequency calculates the frequency of words but includes a weighting aspect where uncommon words are more significant as a feature and thus has a higher weight. After NLP, the remaining number of rows is 39308. That is around 86% of the original data of 45466 rows that remains.

Feature Subset Selection By Dimensionality Reduction Using PCA

For the final preprocessing stage, sampling and PCA was used to prevent overfitting and mainly to reduce the runtime of the script to five minutes. PCA is one of the ways that will minimize the manipulation of the data while still being able to reduce time and space needed for the training. It works by reducing the number of features to a number of principal components. In this case, I have reduced the components to 2 features compared to 41247 features from TF-IDF. Since the dimension is two, it is possible to visualize the data using a scatterplot as shown below.

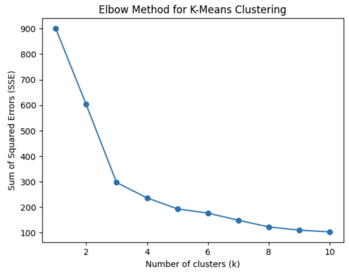


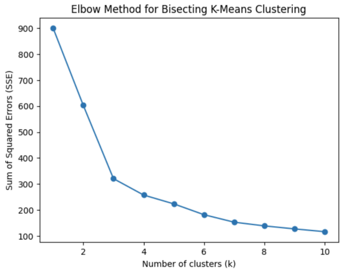
Normally, PCA does not affect the data too much but due to the massive difference in the number of features from 41247 to 2, this will greatly reduce the time needed to run the script but will also affect the accuracy of the clusters.

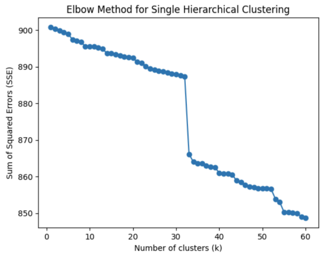
Sampling with TF-IDF Parameters
As for sampling, the subset of the original vectorization output has been tuned using the "min_df" parameter of TfidfVectorizer in sklearn. In this case, min_df was set to five which reduces the number of features to 10100. min_df when represented as an integer, means that the term must appear in a minimum number of movies in order to be considered as a feature.

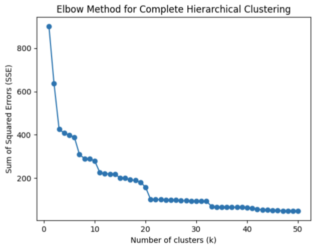
Results

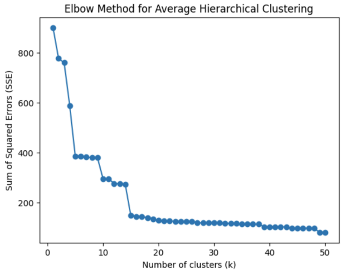
The following clustering techniques used are k-means, bisecting k-means, and agglomerative hierarchy clustering for the linkage of single, complete, average, and centroid. Each clustering method is evaluated with the unsupervised approach using the sum of squared error (SSE) and hyperparameters for each cluster are optimized using the elbow method.

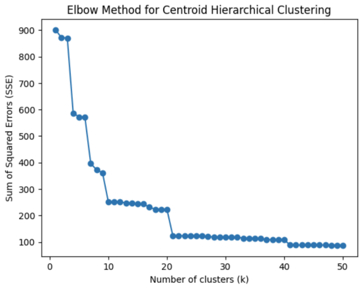






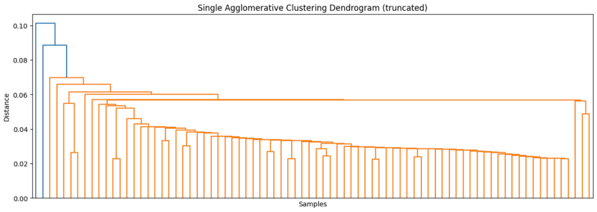


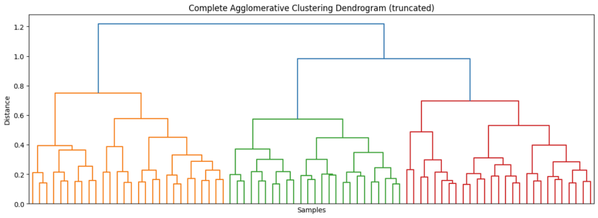


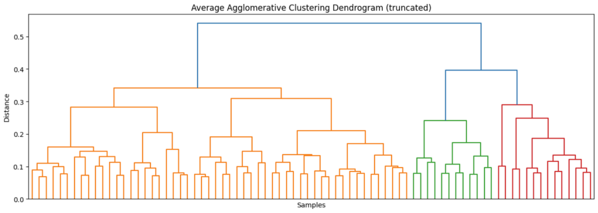


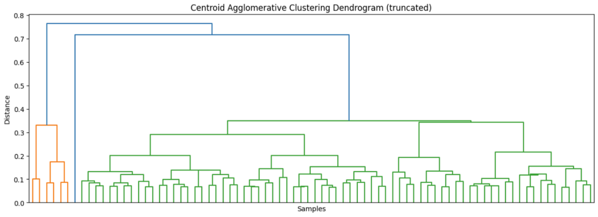
Based on the elbow method graphs shown, the optimal k values for k-means and bisecting k-means is 3. For each linkage of agglomerative clustering is k=32 for single, k=21 for complete, k=15 for average, and k=21 for centroid.

For more details on the structure of the agglomerative clustering models, these are the following dendrograms.









Overall, the results for the elbow method shows the different SSE measures for every used clustering method in this study. Thus we can compare the performance of the clustering methods based on whichever has the lowest SSE score. The steepness of the elbow can also determine if the SSE measurement is valid compared to other clusters. Most likely, having more clusters will be better since more clusters means more categories in which different movies can be related to one another. But having too many clusters may overfit the data. Based on the results for the elbow method shown above, single agglomerative clustering has the worst performance of a SSE value of around 865. The next worst is k-means and bisecting k-means with a score of nearly 300. Afterwards is complete and average linkage for agglomerative clustering of around 150 for the SSE score. Finally the best clustering method is centroid agglomerative clustering with a SSE value of around 110 which is the lowest.

Conclusion

To conclude, based on the SSE values, clustering using centroid agglomerative clustering is the best method while the simple agglomerative clustering. By using this model, users can find other movies that may be related to the current movie that they are reviewing at IMDB. The script at the very end includes a way to search other movies in the same cluster. When using centroid agglomerative clustering for the recommendation system, using a cluster index of 5 which is the same cluster index as the movie “Toy Story” will recommend the imdb id of tt0113497 which is the movie “Jumanji”. But there are also some pretty bad results like tt0114576 which is the movie “Sudden Death”, an R rated movie. Due to the lack of training and preserving of data, the results of the clusters are quite poor. However, if there was not a 5 minute runtime limit, it may be possible to improve the results of the recommendation system. So in the future, this model can be fitted properly compared to the current results where the data is greatly under fitted.

References

1. Arslan, Erhan. "Natural Language Processing: Vectorization Techniques — Step 6." Medium. Last modified July 5, 2024. https://medium.com/@erhan_arslan/natural-language-processing-vectorization-techniques-step-6-d44b53575a2c.

2. Banik, Rounak. "The Movies Dataset." Kaggle. Last modified 2017. https://www.kaggle.com/datasets/rounakbanik/the-movies-dataset.

3. Inc, Ghatifernado. "NLP for Multilingual Communication: Breaking Language Barriers." Medium. Last modified July 23, 2024. https://medium.com/@ghatifernado.inc/nlp-for-multilingual-communication-breaking-language-barriers-6d0aac46c22b.

4. Jose, Basil K. "Data Preprocessing | Natural Language Processing." Medium. Last modified May 4, 2021. https://basilkjose.medium.com/data-preprocessing-natural-language-competition-processing-dcbbf9d014e8.

5. Sharma, Padmesh. "TEXT CLASSIFICATION USING ML." Medium. Last modified September 22, 2023. https://medium.com/@padmesh2224/text-classification-using-ml-17b756558d49.