Sentiment Analysis

Movies Reviews

Table of Contents

List of Figures

List of Tables

1 Objectives

This study aims to analyze movie reviews using text mining techniques, specifically Support Vector Classifiers (SVC) and Naive Bayes algorithms, and visualize the results using Uniform Manifold Approximation and Projection (UMAP) to explore trends and patterns in the data. The study will collect movie reviews from online sources using OpenAI's ChatGPT(Aydın and Karaarslan, 2023). The results will then be used to train the SVC and Naive Bayes Classifiers (NBC) to predict the sentiment of movie reviews. Finally, UMAP will be used to visualize the results in a lower-dimensional space to explore patterns and trends in the data.

Two datasets were generated using this generative AI tool namely, "Labelled.csv" and "Unlabelled.csv". The first dataset, "Labelled.csv" contains 1140 rows of positive and negative movie reviews. Using this dataset, this study evaluates the performance of the two classification models, SVC and NBC. The best performing classification algorithm will then be deployed on the second dataset, "Unlabelled.csv" which contains 30 rows of positive and negative movie reviews.

The findings of this study could be valuable to the movie industry, as they can use the sentiment analysis of movie reviews to identify areas for improvement in their productions, marketing, and public relations. The UMAP visualization could provide insights into trends and patterns in the movie review data that could be used to improve the industry's understanding of audience preferences and inform decision-making.

2 Data Preparation

To begin the sentiment analysis, the dataset, "Labelled.csv", the target column, "Sentiment" required some basic data preprocessing. The strings ('positive' and 'negative') are first converted to numerical values (1 and 0). Then, the dataset is checked for any null values or NaNs.

Next, the target column distribution is checked to see if the dataset is balanced or not. There are 598 rows of positive (1) reviews and 542 rows of negative (0) reviews. The balance of the dataset is okay to adopt the models' accuracies as a performance metric.

Thereafter, a cleaner helper function is used to parse the dataset and prepare it for analysis. In summary, the helper function achieves the following:

- Removing HTML entities from the text using the BeautifulSoup library.
- Substituting hashtags, mentions, URLs, and other non-alphabetic characters with whitespace using regular expressions.
- Tokenizing the text into individual words using the NLTK library.
- Converting all words to lowercase.
- Removing stop words (commonly used words such as "a", "an", "the", etc.) using the NLTK library.
- Lemmatizing the remaining words (converting them to their base form) using the WordNetLemmatizer from the NLTK library.

The resulting output of this function is a list of cleaned and lemmatized words from the original text, which can be used for further analysis or machine learning tasks.

A final step in the data preparation stage is join the words in each review together into a single string with whitespace separating each word. The resulting 'cleaned_reviews' column would contain a list of strings, where each string represents a cleaned and lemmatized movie review that can be used as input to a machine learning model.

3 Term Frequency-Inverse Document Frequency Matrix (TASK 1)

Term Frequency-Inverse Document Frequency (TF-IDF) matrix is a statistical tool used in natural language processing to represent the importance of words in a document or corpus. In simpler terms, it helps to identify the most significant words in a document based on

how frequently they appear in the text and how unique they are to that particular document (Zhang and Ge, 2019).

The term frequency (TF) measures how often a word appears in a document, while the inverse document frequency (IDF) measures how unique the word is to that particular document compared to other documents in the corpus. These two metrics are combined to create a matrix that assigns a score to each word in a document, with higher scores indicating greater importance. This matrix can then be used to compare and analyze documents based on the significance of their individual words (Harish and Revanasiddappa, 2017).

3.1 Creation of Document-Term or TF-IDF

To create a TF-IDF matrix for the movie review dataset, the TfidfVectorizer object, a class from the scikit-learn library (*sklearn.feature_extraction.text.TfidfVectorizer*, no date) is used to transform the text corpus into a TF-IDF matrix. The constructor of the TfidfVectorizer takes several parameters, which can be tuned to optimize the performance of the classifiers. In this study, two hyperparameters were tuning during the simulations to evaluate the classifiers performance. The two hyperparameters are:

- **min_df**: This parameter, a float value between 0 and 1, specifies the minimum document frequency that a term must have in order to be included in the matrix.
- **ngram_range**: This parameter, a tuple, specifies the range of n-grams (contiguous sequences of words) to be considered or extracted.

In this study, these hyperparameters were tuned and the effect of the tuning were analyzed as well as the model accuracy under a range of experimental simulations. These are described in the next section.

3.1.1 Simulation 1

In this case, the min_df hyperparameter was set to 0.0264 and the ngram_range parameter was set to (1,3). In this case, any term that appears in less than 2.64% of the documents will be ignored. In other words, each ngram (unigram, bigram, & trigram) must be present

in at least 30 documents for it to be considered as a token. The ngram_range setting here means that both unigrams (single words), bigrams (two-word phrases) and trigrams (three-word phrases) will be considered.

The SVC model records an accuracy of 0.9210 whereas the NBC model achieves an accuracy of 0.8912. The SVC model records approximately 0.03 performance improvement gain over the NBC.

The TF-IDF matrix shape in this setting is (1140, 67). This indicates that there are 1140 movie reviews in Labelled.csv (the corpus), and 67 unique terms (or words) in the vocabulary. Each document is represented as a row in the matrix, and each term is represented as a column. The entries in the matrix represent the TF-IDF weights of the corresponding term in the corresponding document. For example, the entry in row i and column j represents the TF-IDF weight of term j in document i. This means that the TF-IDF weight of term j is high in document i if the term appears frequently in the document but infrequently in the rest of the corpus.

3.1.2 Simulation 2

In this case, the min_df hyperparameter is still set to 0.0264. However, the ngram_range parameter was set to (1,2). In this case, each ngram (unigram & bigram) must be present in at least 30 documents for it to be considered as a token.

The SVC model and NBC accuracies are unchanged. The SVC and NBC model record accuracies of 0.9210 and 0.8912 respectively. The TF-IDF matrix shape in this setting remains as (1140, 67).

3.1.3 Simulation 3

In this case, the min_df hyperparameter is still set to 0.0264 with the ngram_range parameter was set to (1,1). The value (1,1) specifies that only unigrams (single words) should be considered.

Both model accuracies increase in this setting. The SVC and NBC model record increased accuracies of 0.9237 and 0.8930 respectively. The TF-IDF matrix shape in this setting changes to (1140, 57). There are now 57 unique terms (or words) in the vocabulary.

3.1.4 Simulation 4

In this case, the min_df hyperparameter was set to 0.0527 and the ngram_range parameter was set to (1,3). In this case, any term that appears in less than 5.27% of the documents will be ignored. In other words, each ngram (unigram, bigram, & trigram) must be present in at least 60 documents for it to be considered as a token.

Both models record a drop in classification accuracies. The SVC model records an accuracy of 0.7860 whereas the NBC model achieves an accuracy of 0.7737. . The TF-IDF matrix shape in this setting changes to (1140, 17). There are now only 17 unique terms (or words) in the vocabulary.

3.1.5 Simulation 5

The min_df hyperparameter is still set to 0.0527. However, the ngram_range parameter was set to (1,2). In this case, each ngram (unigram & bigram) must be present in at least 60 documents for it to be considered as a token.

The SVC model and NBC accuracies are unchanged. The SVC and NBC model record accuracies of 0.7860 and 0.7737 respectively. The TF-IDF matrix shape in this setting remains as (1140, 17).

3.1.6 Simulation 6

The min_df hyperparameter is still set to 0.0527. However, the ngram_range parameter was set to (1,1). Only unigrams will be considered.

The SVC and NBC model record accuracies of 0.7886 and 0.7842 respectively. The TF-IDF matrix shape in this setting changes to (1140, 16).

The results of the six (6) simulations are presented in Table 1 below.

Table 1: Results from TF-IDF Matrix and Hyperparameter Tuning

Simulation #	min_df	ngram_range	Vocabulary Size	SVC Accuracy	NBC Accuracy	TF-IDF Shape
1	0.0264	(1,3)	30	0.9210	0.8912	(1140,67)
2	0.0264	(1,2)	30	0.9210	0.8912	(1140,67)
3	0.0264	(1,1)	30	0.9237	0.8930	(1140,57)
4	0.0527	(1,3)	60	0.7860	0.7737	(1140,17)
5	0.0527	(1,2)	60	0.7860	0.7737	(1140,17)
6	0.0527	(1,1)	60	0.7886	0.7842	(1140,16)

4 Model Deployment

4.1 Justification for Model Evaluation Metric

Accuracy was chosen as the metric for model evaluation. This is because it had been observed during the data preparation that the dataset under evaluation is a balanced dataset where the number of instances in each class is roughly equal. In the case of balanced datasets, where there is no significant class imbalance, accuracy provides a good indication of the overall performance of the classifier. For instance, let's suppose the task at hand is a binary classification problem with 50% of the instances belonging to class A and the other 50% belonging to class B. If a classifier achieves an accuracy of 80%, it means that it has correctly classified 80% of the instances in both classes, which is a good indication of its overall performance.

4.2 Limitations of the Accuracy Metric

Accuracy may not be the best metric to use in cases of imbalanced datasets, where the number of instances in one class is significantly larger than the other. In such cases, a classifier that always predicts the majority class will achieve high accuracy, even though it

may not be performing well on the minority class. In such cases, metrics such as precision, recall, and F1 score may be more appropriate.

4.3 Deployment of the SVC model on Unlabelled.csv

The best performing SVC model was deployed on the Unlabelled dataset. First, the "Sentiment" column is removed to ensure that there is no target column. Next, the list of features in the TF-IDF model is exported to a csv file, "vocabulary.csv". Exporting the features of a TF-IDF model is important when predicting an unlabelled dataset because it allows you to use the same vocabulary and feature representation that was used in training the model.

The TF-IDF model represents each document as a vector of features, where each feature corresponds to a unique term in the vocabulary. These features capture the importance of each term in the document relative to the entire corpus. When the model is trained, it calculates the IDF (inverse document frequency) values for each term in the corpus, which are used to weight the importance of each term in a document.

To predict an unlabelled dataset using the same TF-IDF model, you need to represent each document in the dataset as a vector of the same features that were used in the training data. This requires using the same vocabulary and IDFvalues that were used during training. By exporting the features of the TF-IDF model, you can ensure that the same vocabulary and IDF values are used during prediction, which helps ensure that the model is making accurate predictions based on the same feature representation that it was trained on.

After performing the data cleaning as was done on the Labelled dataset, the Unlabelled dataset, the deployed model is used to transform the Unlabelled data into a TF-IDF matrix representation. The resulting predictions are saved in "predicted_sentiment.csv".

4.4 Prediction Results of the Unlabelled.csv

The results of the predictions on the Unlabelled dataset is compared with the actual Unlabelled dataset containing the target column, "Sentiment". The actual and predicted sentiments are both saved in the csv file, "sentiment_pred_actual.csv". The accuracy using

the deployed model with the unlabelled dataset is 0.86. This is a very satisfactory performance given that these reviews were trained using the vocabulary from a different corpus. The confusion matrix from the predictions is shown below.

ACTUAL	Positive (1)	**13 (TP)**	**2 (FN)**
	Negative (0)	**2 (FP)**	**13 (TN)**
		Positive (1)	Negative (0)
		PREDICTED	

5 Visualization of Labelled Data Using UMAP (TASK 2)

UMAP is a technique used to visualize high-dimensional data in a way that is easier to understand. One of the important things that UMAP does is to identify the relationships between different data points and preserve them in the lower-dimensional representation of the data.(Wu *et al.*, 2022).

One of the key hyperparameters of UMAP is **n_neighbors,** which determines the size of the local neighborhood that is used to compute the local structure of the data. For example, if n_neighbors is set to 10, then UMAP will look at the 10 closest data points to each point to understand how they are related. By looking at these relationships, UMAP can create a lower-dimensional representation of the data that preserves the important relationships between data points. Choosing the right value for n_neighbors is important because if the value is too small, important relationships may be missed, and if the value is too large, the visualization may be dominated by more general patterns in the data (Khan, Walker and Zeiler, 2023).

Another important hyperparameter in UMAP is the **min_dist** parameter, which determines how tightly the data points are packed together in the lower-dimensional representation. For example, if min_dist is set to a small value, then UMAP will try to preserve the

distances between the data points as much as possible, resulting in a more tightly packed visualization.

5.1 UMAP Visualization on Labelled Dataset with n_neighbors=30, min_dist=0.3

Setting the UMAP hyperparameters, n_neighbors and min_dist as follows results to clear formation of clusters as shown in Figure 1 below.

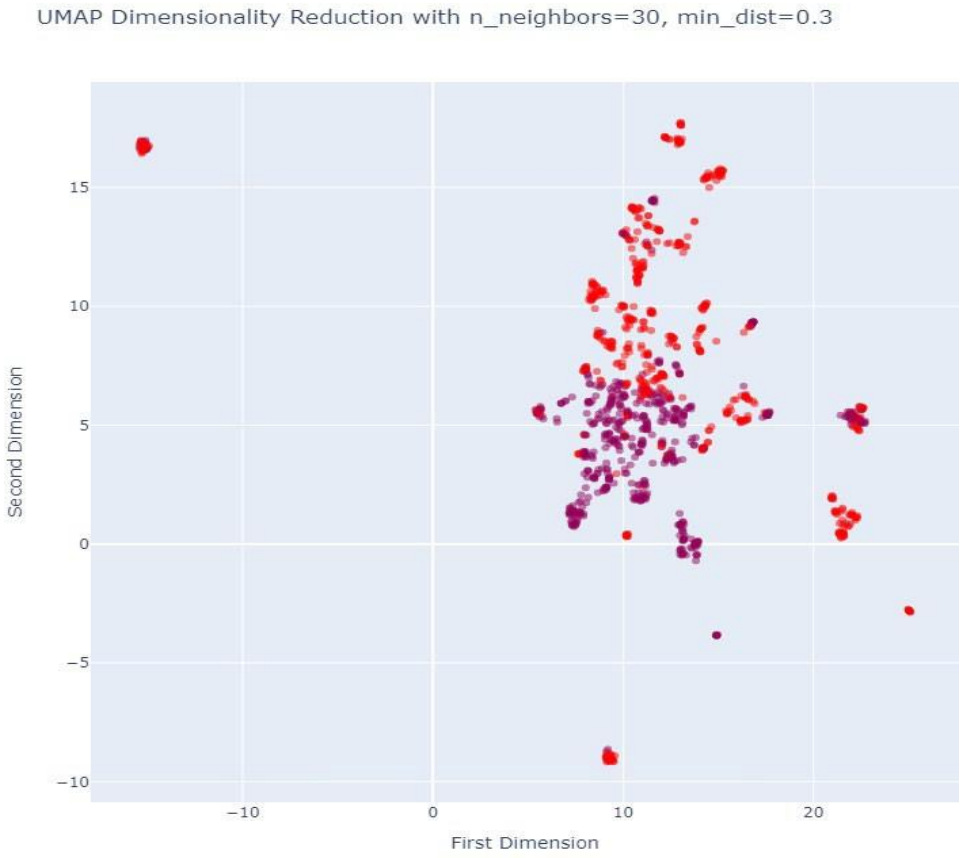


Figure 1: UMAP Formation of Clusters with n_neighbors=30, min_dist=0.3

We can clearly see some of the clusters in the annotated figure shown in Figure 2. There are at least 2 main clusters and about 5 sub-clusters as indicated in the plot.

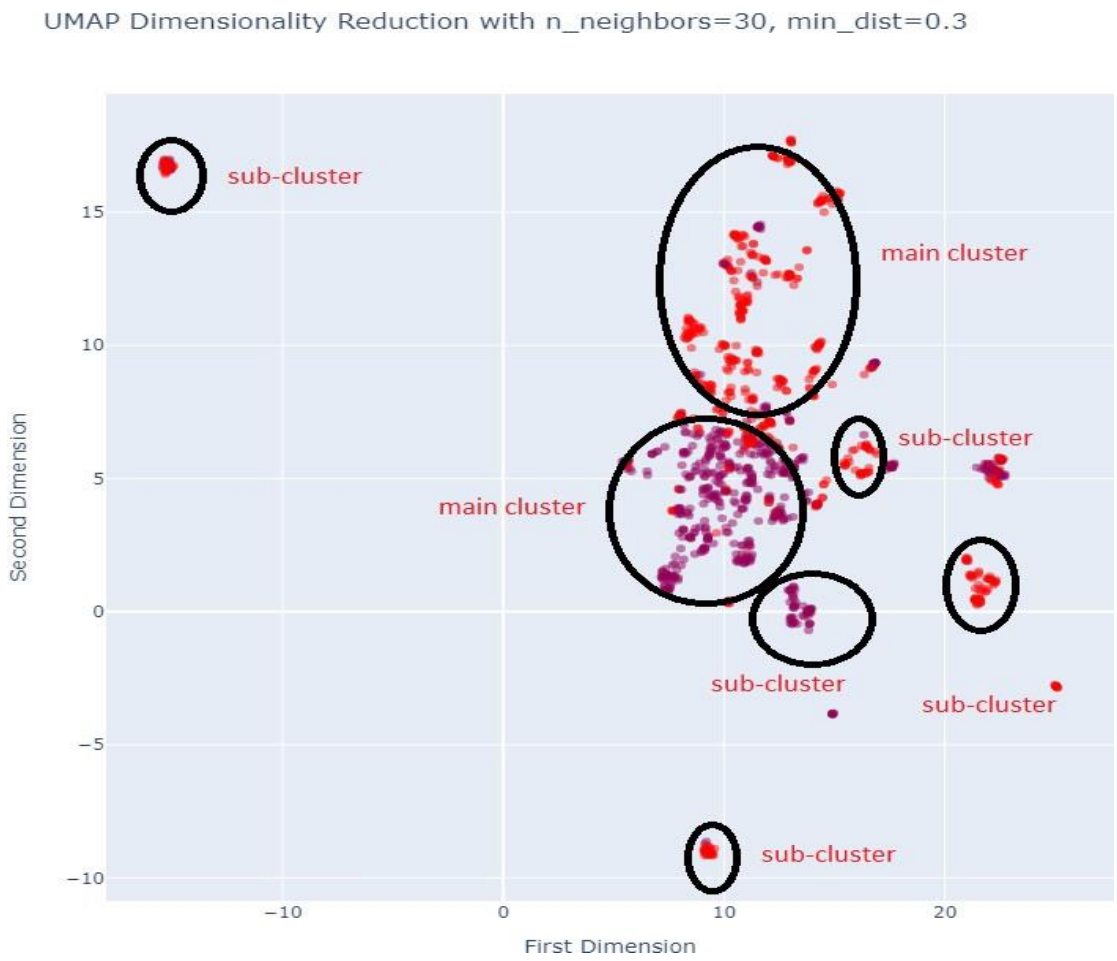


Figure 2: Formation of Clusters and sub-clusters

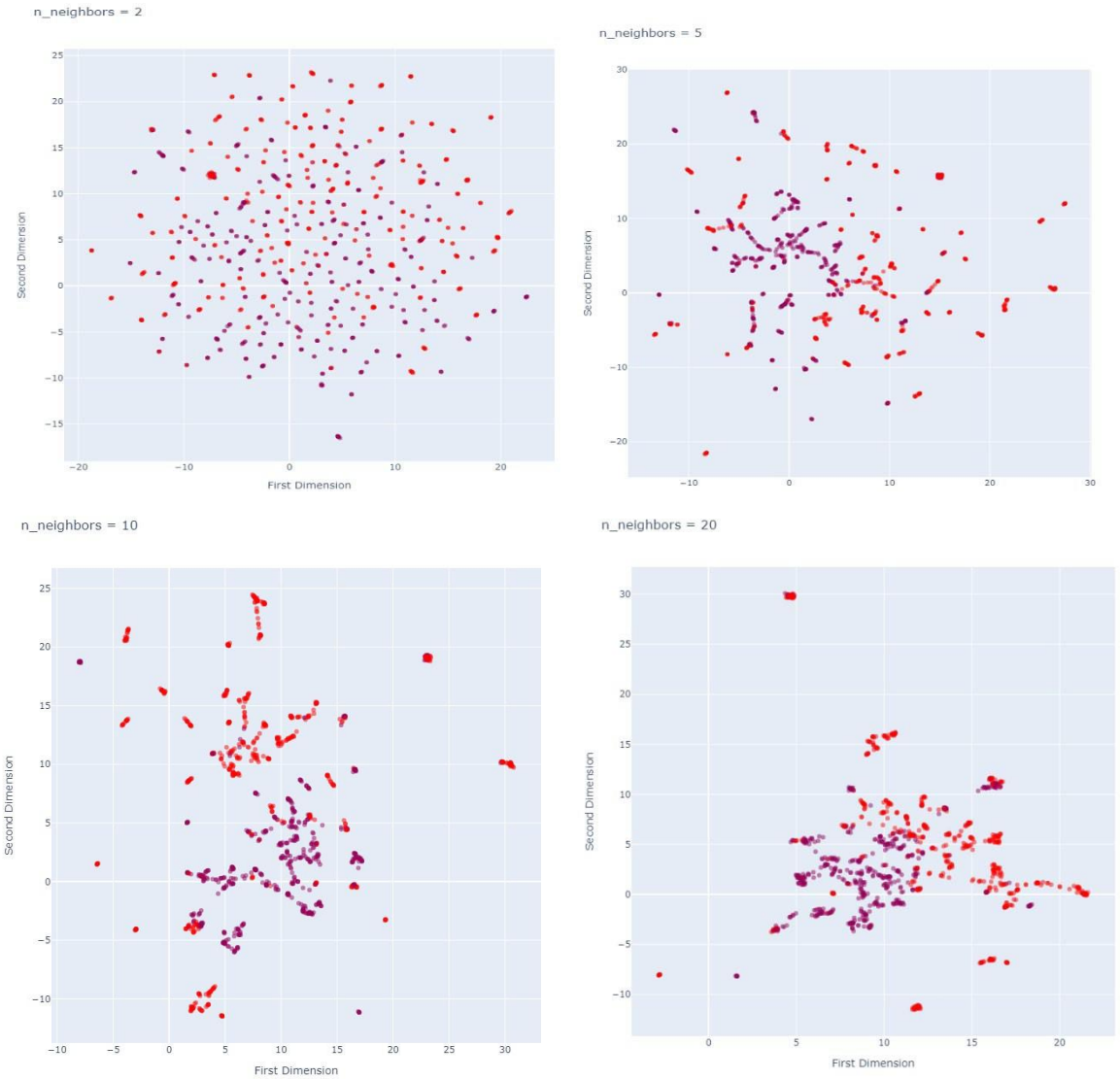
5.2 UMAP Hyperparameters Tuning of n_neighbors and min_dist

To evaluate the impact of the two hyperparameters, n_neighbors and min_dist, on the Labelled dataset, we will first write two similar short utility functions: **explore_umap_n_neighbors** and **explore_umap_min_dist** that can fit the data with

UMAP given a set of parameter choices, and plot the result. The resulting plots are saved for analysis.

5.2.1 Tuning of n_neighbors

In tuning the n_neighbors hyperparameter, the default value is set to 30. We then look at values ranging from 2 (which is a very local view of the manifold) up to 285 (a quarter of the data). The values used are (2, 5, 10, 20, 30, 50, 100, 200, 285).



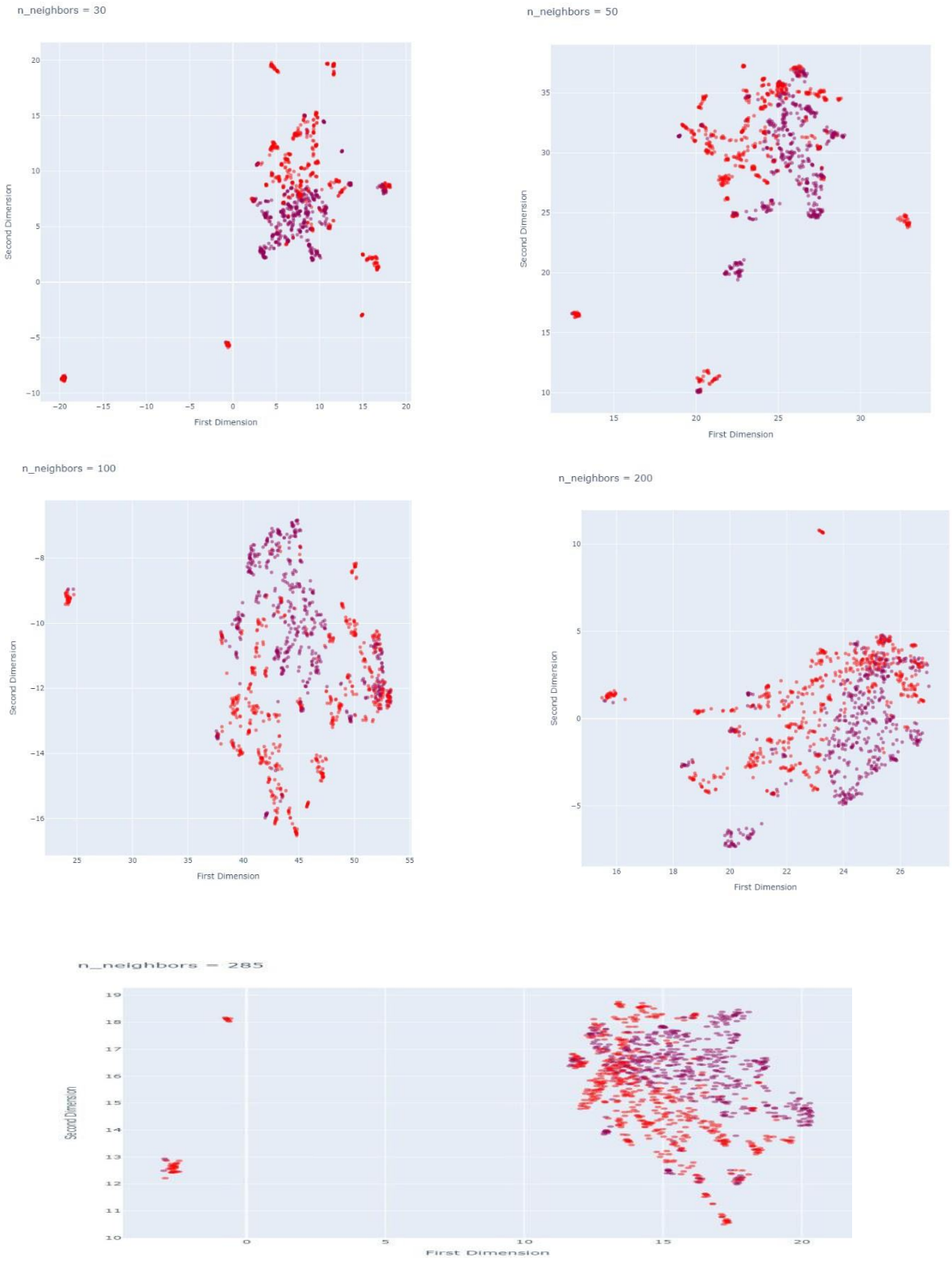


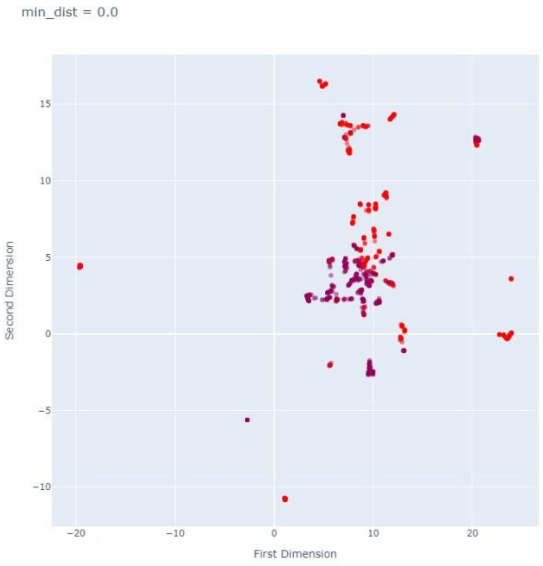
Figure 3: Tuning the n_neighbors parameter

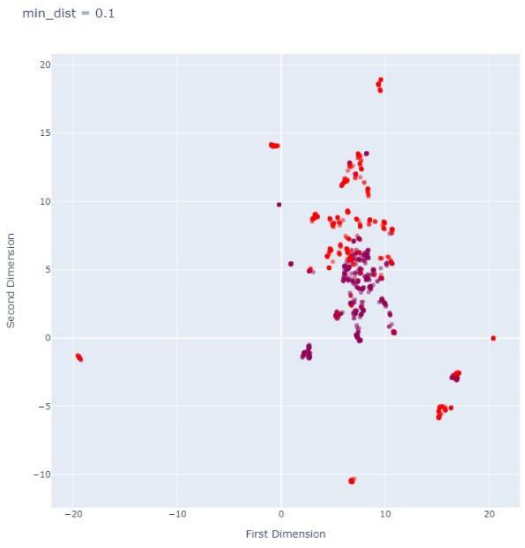
We can see from the plots that when setting the value of **n_neighbors** to 2 in UMAP, it is observed that the algorithm connects small chains together, however, it fails to capture the larger connections due to its narrow/local viewpoint. Consequently, numerous isolated components and individual data points remain. This outcome is a reflection of the data's disconnection and scattered nature when observed at a fine level of detail.

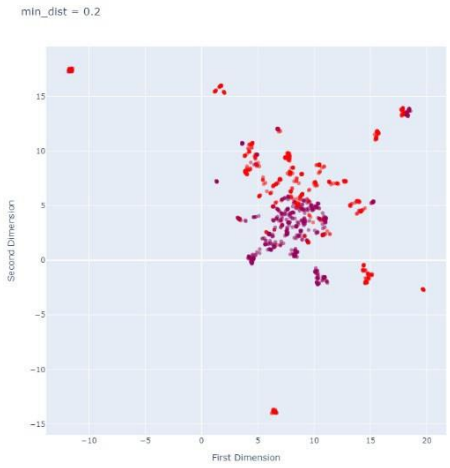
Increasing the value of n_neighbors in UMAP allows for a better understanding of the data's overall structure. This results in UMAP being able to connect more components together and provide a more comprehensive view of the data's broader structure. By the time n_neighbors reaches **100**, the overall view of the data is fairly accurate and it becomes evident how different colors interrelate with each other across the entire dataset.

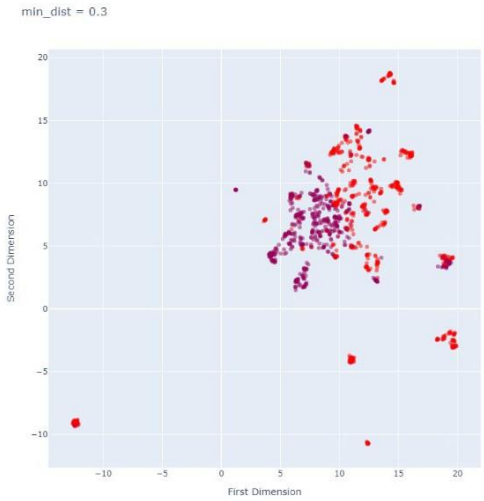
5.2.2 Tuning of min_dist

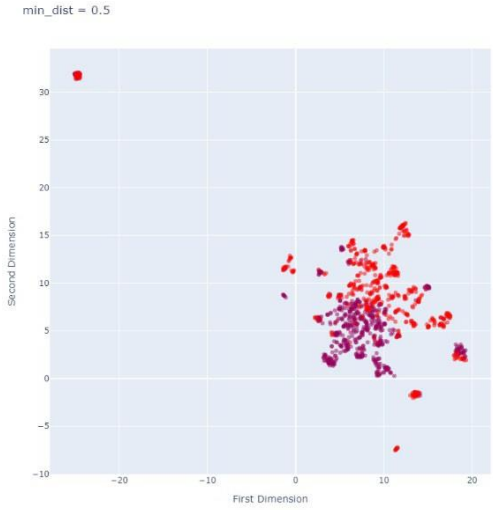
The default min_dist value is set to 0.3. We will tune the min_dist parameter from 0.0 to 0.9. The values used are (0.0, 0.1, 0.2, 0.3, 0.5, 0.6, 0.7, 0.8, 0.9). The resulting plots are shown below.

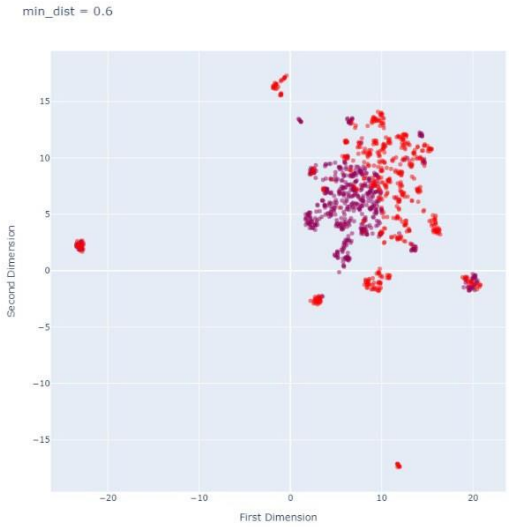


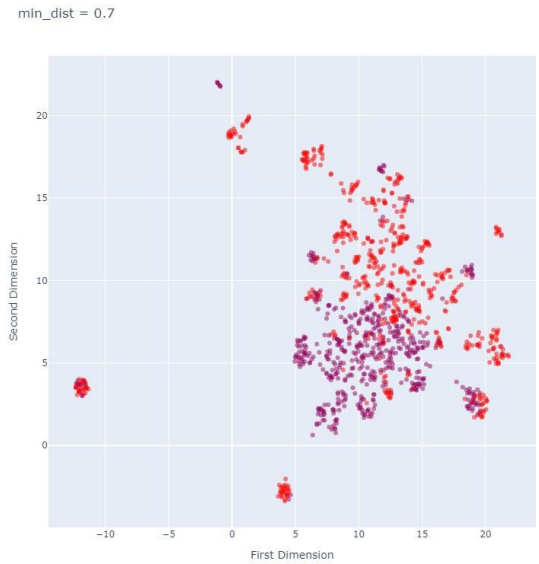


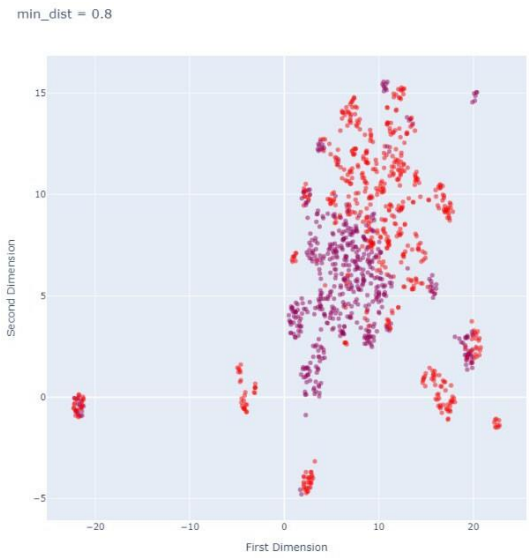

min_dist = 0.2
Second Dimension
First Dimension


min_dist = 0.3
Second Dimension
First Dimension


min_dist = 0.5
Second Dimension
First Dimension


min_dist = 0.6
Second Dimension
First Dimension


min_dist = 0.7
Second Dimension
First Dimension


min_dist = 0.8
Second Dimension
First Dimension

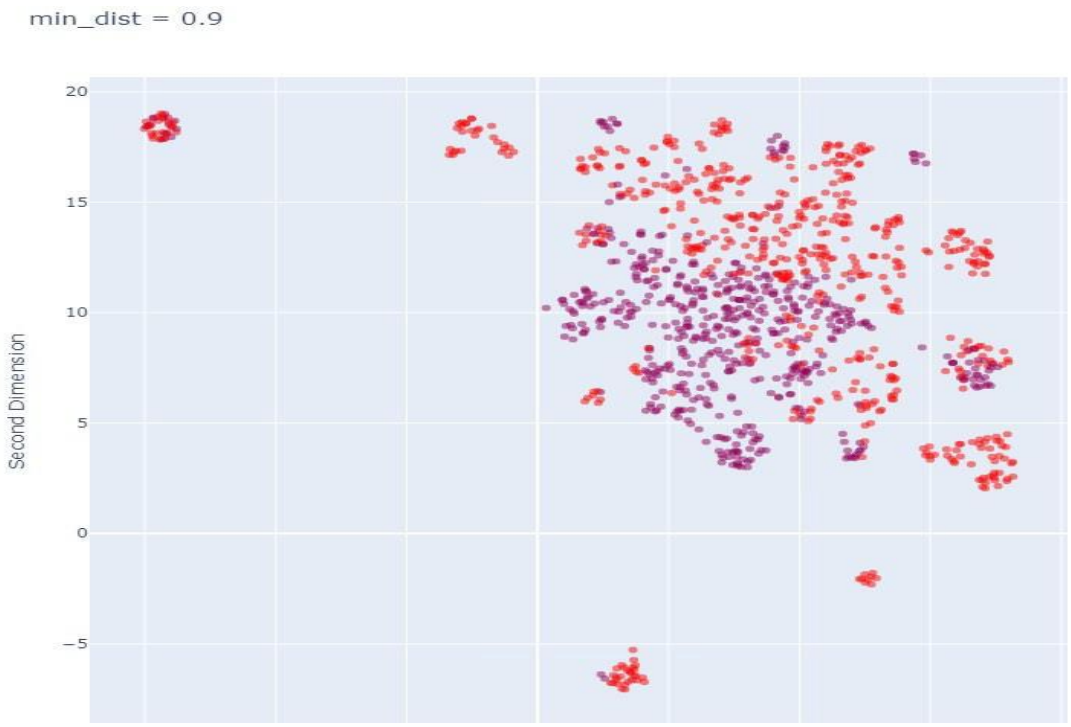


Figure 4: Tuning the min_dist parameter

The plot observations show that setting min_dist to 0.0 in UMAP helps identify small connected components, clumps, and strings in the data, and highlights these features in the resulting embedding. However, as min_dist is increased up to about **0.8**, these structures become more diffused and generalized, resulting in a broader view of the data while losing the finer details of the topological structure.

5.3 Visualizing Clusters and Sub-clusters

Let us analyze the plot with an optimal n_neighbors value of 100 shown in Figure 3. We can observe the two main clusters formed in the annotated illustrated in Figure 5. The

negative reviews have been largely clustered in the top of the figure. The positive reviews have been largely bound together in the southbound zone or bottom of the plot.

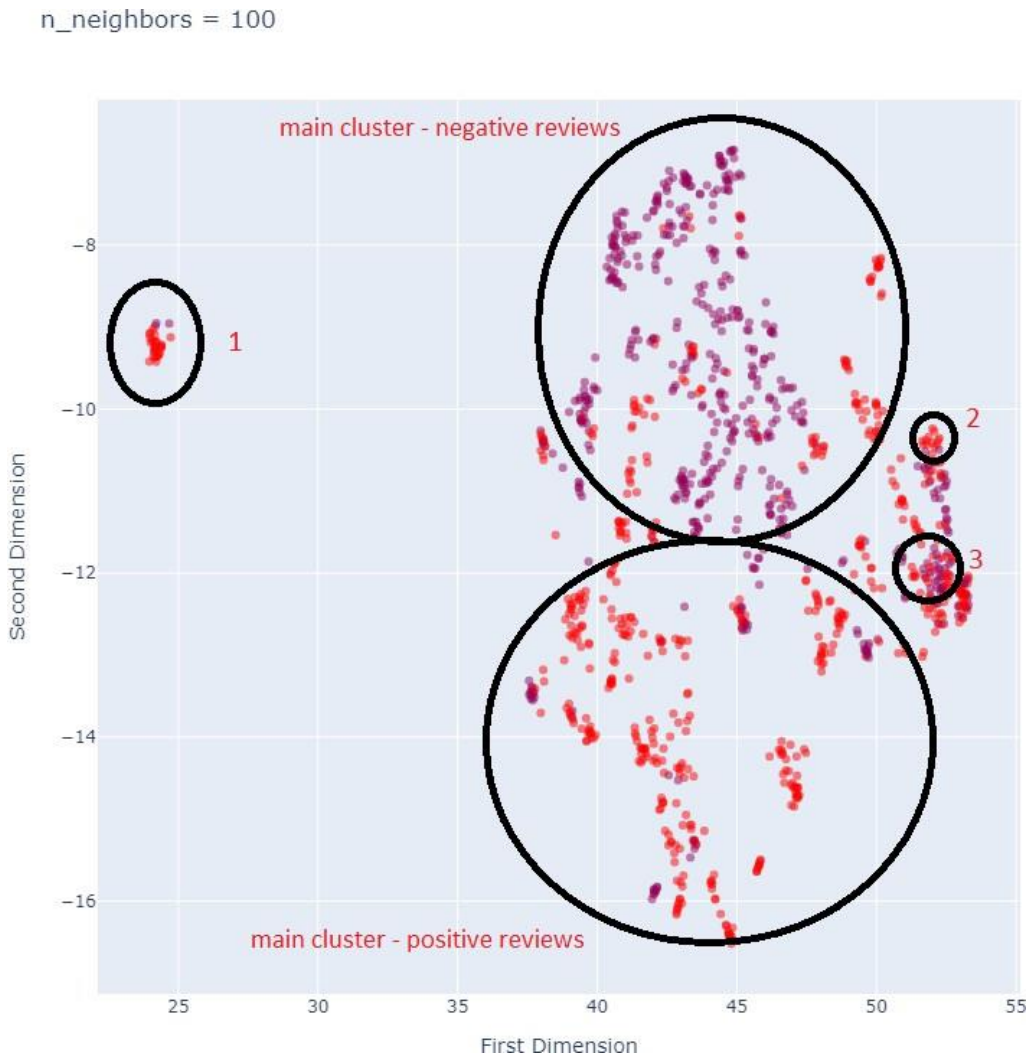


Figure 5: Visualizing Clusters and Sub-clusters with an optimal value of 100 n_neighbors

Some reviews clustered in the <u>negative cluster</u> of the plot include:

- "I found this movie to be completely forgettable. It didn't leave any lasting impression on me."
- "I found this movie to be dull and uninspiring. It didn't spark any emotion or excitement."
- "I found this movie to be incredibly confusing. The plot was convoluted and hard to follow."

There is a strong similarity in the vocabulary in the above reviews. This is because these reviews share common negative sentiment or theme. This clustering can be useful in identifying patterns in negative tweets and analyzing the reasons behind negative sentiment.

More so, the UMAP plot clusters these three reviews together as negative reviews because they all contain negative sentiment and language about the movie. Specifically, they all use phrases such as "forgettable," "dull," "uninspiring," "confusing," and "convoluted plot" to describe their negative experiences with the movie. These phrases suggest that the reviewers did not enjoy the movie and found it to be lacking in key areas such as excitement, emotion, and coherence. Overall, UMAP is able to group these reviews together based on the similarity of their negative sentiment and language, indicating that they are likely to be negative reviews of the same movie.

Some reviews clustered in the positive cluster of the plot include:

- "I absolutely loved this movie! The acting was superb, the story was compelling and the soundtrack was amazing."
- "I loved this movie! The characters were well-developed and the dialogue was witty and entertaining."
- "I loved everything about this movie. The story, the acting the music - all of it was amazing."

As was the case for the negative reviews clustered together, these three reviews are clustered as positive reviews by the UMAP plot because they all contain positive sentiment and praise for the movie. Specifically, they all use phrases such as "loved this movie," "amazing," "superb acting," "compelling story," "well-developed characters," "witty dialogue," and "entertaining." These phrases suggest that the reviewers enjoyed the movie and found it to be of high quality.

Overall, UMAP is able to group these reviews together based on the similarity of their positive sentiment and language, indicating that they are likely to be positive reviews of the same movie.

Some reviews clustered sub-clusters 1&2 of the plot include:

- “This movie was so inspiring. It left me feeling motivated and uplifted.”
- “The story in this movie was heartwarming and inspiring. It left me feeling uplifted.”
- “The special effects in this movie were stunning. They added an extra layer of immersion to the already captivating story.”

The UMAP plot clusters these two reviews together as positive reviews because they both contain similar positive sentiment and language about the movie. Both reviews use phrases such as "inspiring," "heartwarming," and "uplifting" to describe their emotional response to the movie. These phrases indicate that the reviewers found the movie to be emotionally moving and impactful in a positive way. Additionally, both reviews mention the story of the movie, which suggests that the plot was a key factor in eliciting the emotional response from the reviewers. UMAP is able to group these reviews together based on the similarity of their positive sentiment and language, indicating that they are likely to be positive reviews of the same inspiring and uplifting movie.

Finally, some reviews clustered <u>sub-cluster 3</u> of the plot include:

- “The plot was convoluted and confusing. I had a hard time keeping up.”
- “I couldn't even make it through the first 10 minutes, it was that bad.”

It is evident that the UMAP plot clusters these two reviews together as negative reviews because they both contain negative sentiment and language about the movie.

The first review mentions that the plot was "convoluted and confusing" and that the reviewer had a hard time keeping up, indicating that the movie did not provide a satisfying viewing experience. The second review goes even further by stating that the movie was so bad that the reviewer could not even make it through the first 10 minutes, implying that the movie was not worth watching at all. Both reviews use negative language to describe their experiences with the movie, which suggests that they did not enjoy it. Finally, UMAP is able to group these reviews together based on the similarity of their negative sentiment and language, indicating that they are likely to be negative reviews of the same movie.

6 Conclusion

In this study, machine learning algorithms were used to perform text analysis on a dataset of movie reviews, with the goal of clustering similar reviews together and identifying common themes and sentiment. The dataset was then visualized using UMAP, a dimensionality reduction technique that allowed for the exploration and clustering of high-dimensional text data in a lower-dimensional space.

The results of the study demonstrated that UMAP was an effective method for clustering reviews based on their sentiment and language, allowing for the identification of positive and negative reviews of the same movie. This has important implications for the movie industry, as it can be used to better understand and analyze customer feedback and preferences, and make data-driven decisions about which movies to produce and how to market them.

Overall, the study highlights the importance of text analysis and machine learning techniques in understanding customer sentiment and feedback, and their potential to drive decision-making in the movie industry and other industries as well.

References

Aydın, Ö. and Karaarslan, E. (2023) 'Is ChatGPT Leading Generative AI? What is Beyond Expectations?', *What is beyond expectations* [Preprint].

Harish, B.S. and Revanasiddappa, M.B. (2017) 'A comprehensive survey on various feature selection methods to categorize text documents', *International Journal of Computer Applications*, 164(8), pp. 1–7.

Khan, W., Walker, S. and Zeiler, W. (2023) 'A bottom-up framework for analysing city-scale energy data using high dimension reduction techniques', *Sustainable Cities and Society*, 89, p. 104323.

sklearn.feature_extraction.text.TfidfVectorizer (no date) *scikit-learn*. Available at: https://scikit-learn/stable/modules/generated/sklearn.feature_extraction.text.TfidfVectorizer.html (Accessed: 20 April 2023).

Wu, L. *et al.* (2022) 'Deep clustering and visualization for end-to-end high-dimensional data analysis', *IEEE Transactions on Neural Networks and Learning Systems* [Preprint].

Zhang, T. and Ge, S.S. (2019) 'An improved TF-IDF algorithm based on class discriminative strength for text categorization on desensitized data', in *Proceedings of the 2019 3rd international conference on innovation in artificial intelligence*, pp. 39–44.