# Clustering of Text based on News Groups with textual and visual explanations

Group T03 : Shreeya Channappa Yogesh, Mallika Manam, Siddhi Belgamwar

Advanced Topics in Machine Learning Semester Project SoSe24

OTTO VON GUERICKE UNIVERSITÄT MAGDEBURG

## Motivation and problem statement

Text clustering, which is an unsupervised learning task, involves grouping of similar documents without prior knowledge of the categories that helps in understanding underlying structure of data and can be used in various applications like information retrieval, document organization and topic modelling. Numerous techniques have been developed in recent times that deliver remarkable performance in the said task. However, performing text clustering also presents the challenge of understanding intricate working of cluster formations. Efficient models acting as a black box lack transparency in their reasoning and decision process. This project aims to employ advanced techniques like Doc2Vec (to generate document embeddings), Latent Dirichlet Allocation and Self-Organizing Maps for clustering. Using the Explainable AI (XAI) methods like Shapely, we sought to develop a pipeline which will explain the clustering outcome and its decision process.

## Dataset

The 20 Newsgroups dataset, organized into 20 categories with around 1,000 documents each, covers topics like sports, politics, technology, and science. It presents challenges such as high dimensionality, class imbalance, and noise. For effective text clustering, Doc2Vec and Latent Dirichlet Allocation (LDA) are used as features. Doc2Vec generates fixed-length, dense, low-dimensional continuous vectors capturing context and semantics, while LDA discovers latent topics, offering sparse, interpretable topic distributions and reduces the dimensionality of the document-term matrix creating a balanced feature set for clustering. Combining these methods provides a rich feature set that captures both semantic and topical information, leading to improved clustering performance and meaningful results. After pre-processing, hyper-parameter tuning (the best vector size found was 200) and merging the two features, final dataset consisted of 222 columns and 18846 rows.

Hyperparameter tuning for the Doc2Vec model is performed by evaluating different combinations of vector size, window size, and minimum word count to find the best set of parameters that yield the highest similarity scores for the documents. The evaluation_doc2vec function takes a data frame, a list of parameter combinations, and a processed corpus of documents. It iterates over each parameter combination, trains a Doc2Vec model with these parameters, and computes a score based on how accurately the model identifies similar documents using the most_similar function. Using this approach, we have determined the hyperparameter values as vector_size – 200, window size – 5, min_count - 3.

We employ KNN algorithm to select subset of data instead of using entire dataset to perform clustering.

## Clustering and Explanations

Text document clustering involves two key steps: representing pre-processed text in a structured format and clustering the documents. A combination of Latent Dirichlet Allocation (LDA) and Doc2Vec feature representations is used for document representation. Self-Organizing Maps (SOMs), a popular unsupervised neural network, are employed for clustering. SOMs effectively analyze high-dimensional patterns in machine learning by implementing a nonlinear projection of high-dimensional space onto a low-dimensional map. The nodes on this map correspond to clusters of the input samples. Compared to other text clustering methods, SOMs provide a visual representation of the similarity between documents within the low-dimensional map. It leverages its powerful visualization capabilities, effective handling of high-dimensional data, and unsupervised learning nature to facilitate intuitive understanding of text data clusters and provide a robust method for discovering natural groupings in the data without relying on ground truth labels.

To generate the explanation, once the clusters are formed from the SOMs, SHapley Additive exPlanations (SHAP) are applied to interpret the results. SHAP values provide a method for explaining the outputs of any machine learning model using a game-theoretic approach that assesses each feature's contribution to the final prediction. In this context, each feature is given an importance value that indicates its influence on the model's output. SHAP values show how each feature impacts each prediction, the relative importance of each feature, and the model's dependence on feature interactions.

## Implementation

To implement clustering using SOMs, the 'minisom' python package was used. Minisom is a minimalistic and Numpy based implementation of the SOMs. The minisom can be trained by providing appropriate parameters such as the learning rate, number of iterations, the dimensions of the map, etc. The 'som.winner' method is used to get the winner neuron for each data point. The weights are initialized randomly in the source code of the package. Clusters are shown using scatterplot.

To generate explanations, we use 'shap' library in python. To interpret the clustering results, we train a RandomForestClassifier as a surrogate model to predict the SOM cluster labels based on the combined Doc2Vec and LDA features. SHAP is then applied to this surrogate model to identify the contribution of each feature to the clustering decisions. This method provides insights into the features driving the clustering, transforming the unsupervised task into a pseudo-supervised one for interpretability. The contribution is shown using a summary plot and bar pot for 1 cluster.

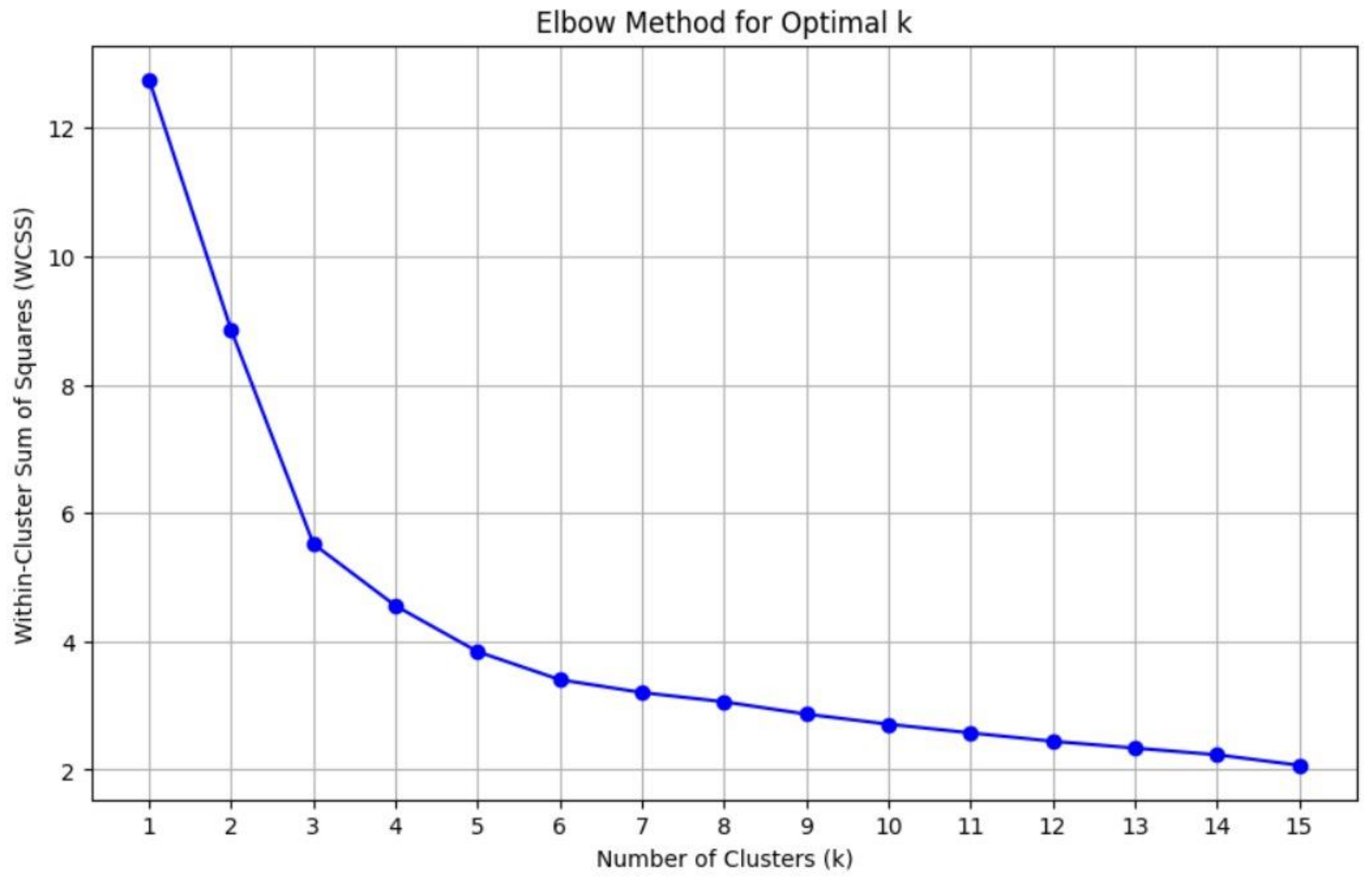

Figure 1: Elbow method

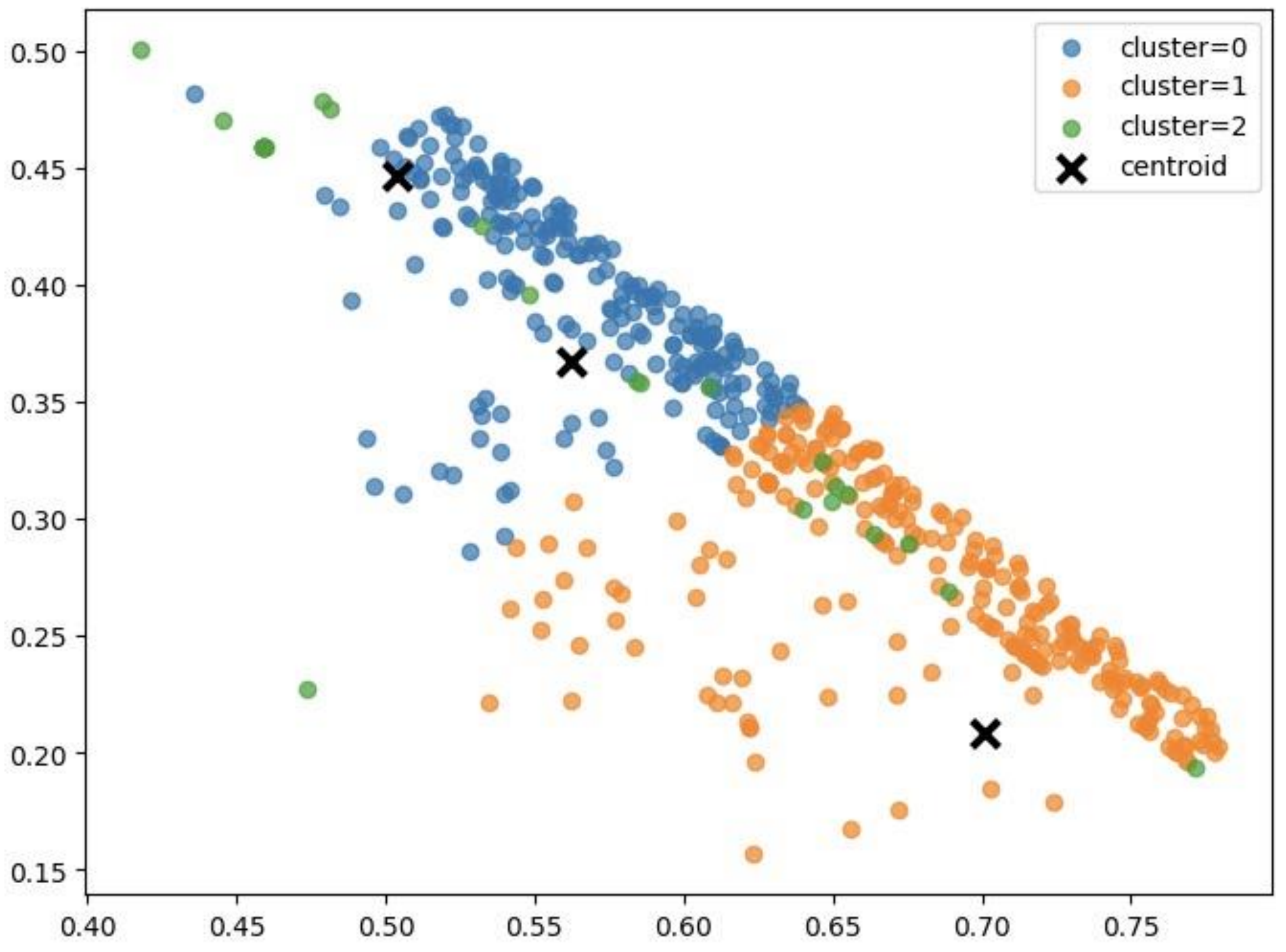

Figure 2: Clustering by SOM

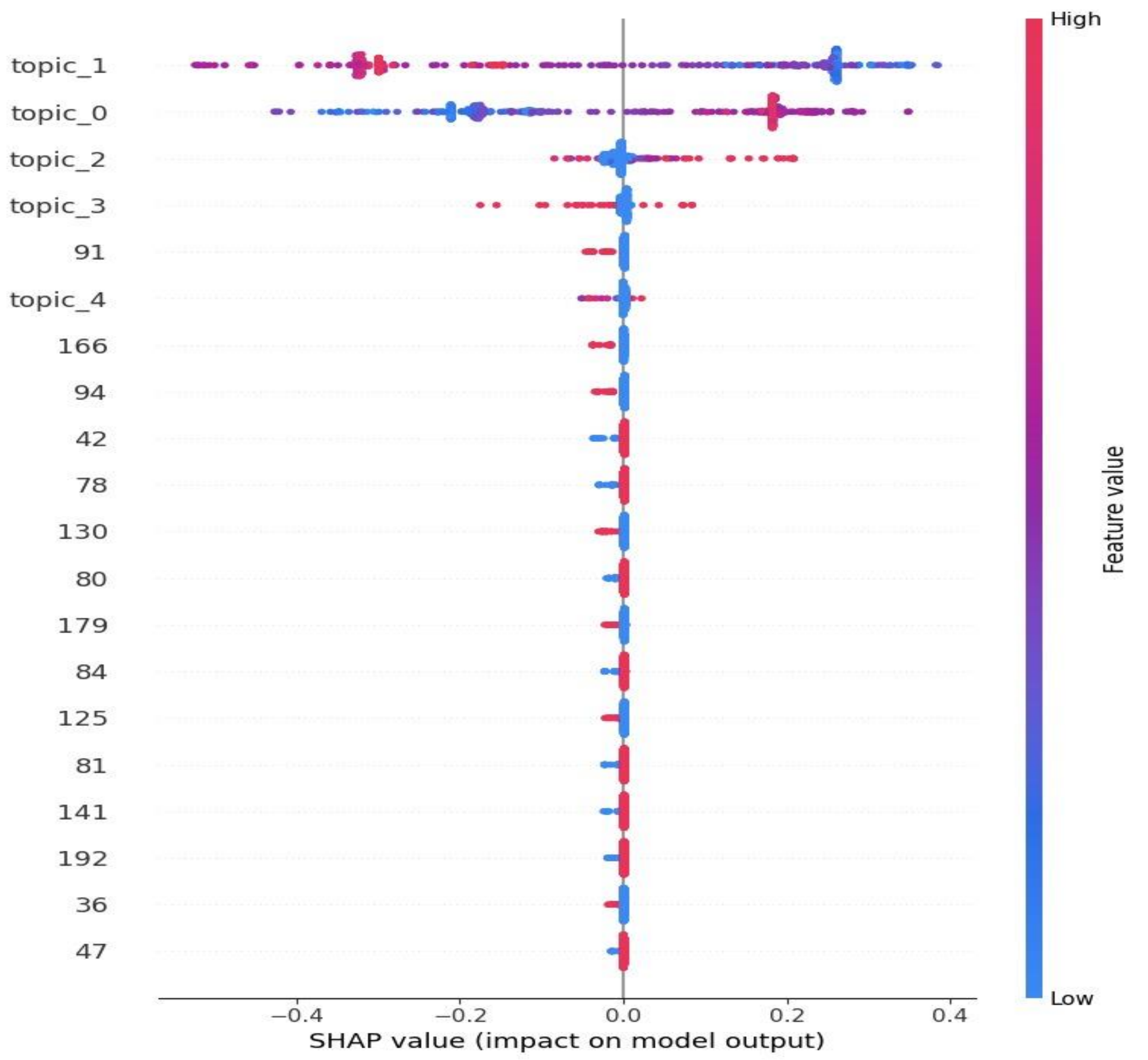

Figure 3: Shap values summary plot

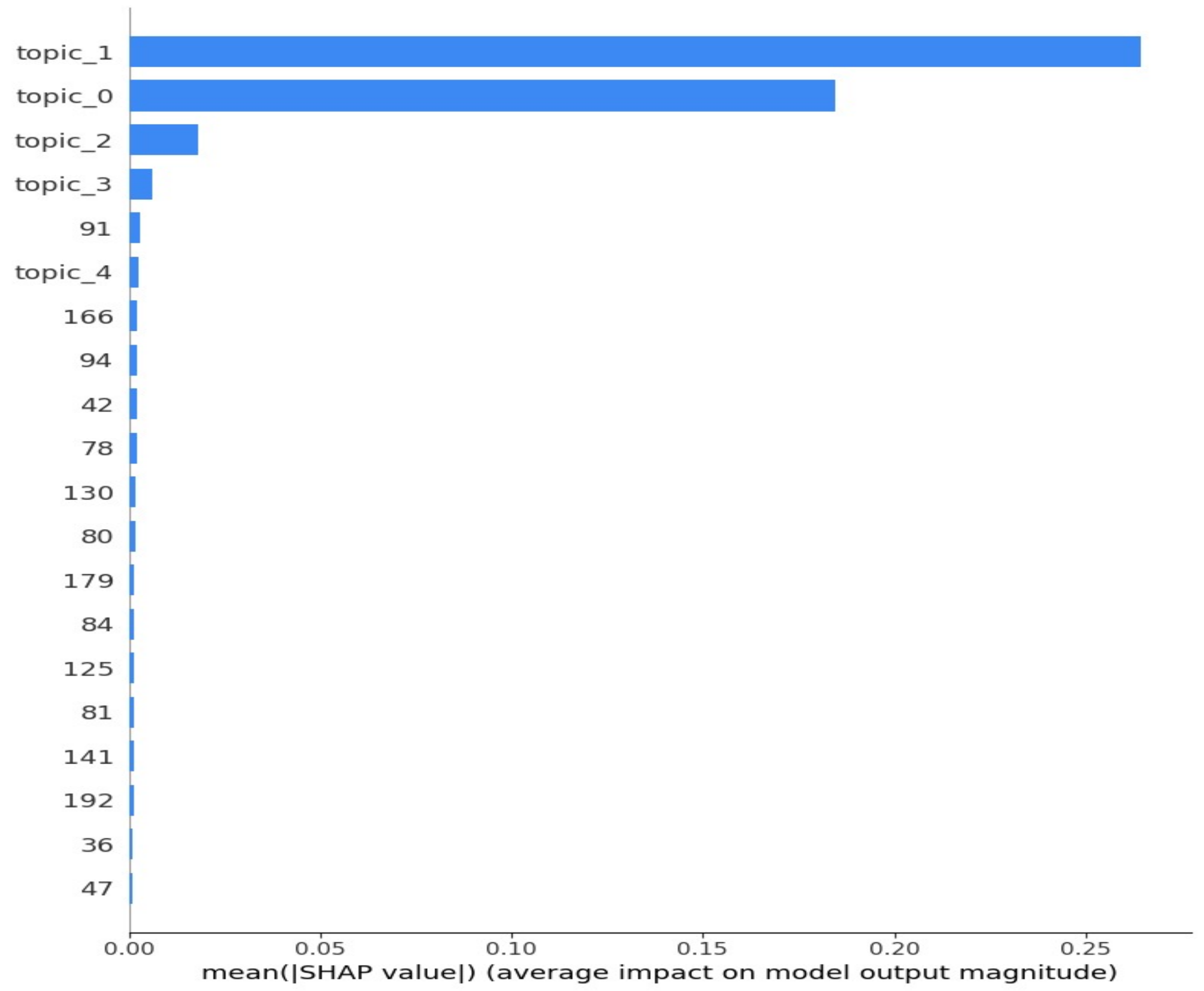

Figure 4: Shap values bar plot

## Evaluation

We evaluated the results clustering based on SOM using Silhouette score and Calinsky-Harabasz Index (CH Index). Because we don't have true labels, we cannot use metrics like Normalized Mutual Information (NMI) or Adjusted Rand Index (ARI)After employing the Elbow method across a range of k values from 1 to 21, we determined that k=3 is optimal for clustering our data when certain values of nearest neighbors and the data point are considered. This conclusion is bolstered by a Silhouette score of approximately 47% and a Calinski-Harabasz (CH) Index score of 1440.55. We also evaluated k=2, which yielded a lower CH Index compared to k=3, further reinforcing our confidence in selecting k=3 as the appropriate number of clusters.Due to the random initialization of weights in SOM, the Calinski-Harabasz (CH) Index score can vary for a fixed set of data. of our text

## Conclusion

Overall, we could see that topic_1, topic_0, and topic_2 features for a particular document are of high importance. The evaluation results indicate that model outperformed for 3 clusters, 500 neighbors with highest CH Index and a Silhouette score of 47%. In our analysis, we found that certain high-priority features such as the ones mentioned above for the same document remained consistent even when parameters such as the number of nearest neighbors and k values changed, indicating their inherent importance.In our project, the

MiniSom package by default initializes weights using random values. We could improve this initialization

process by using the centroids obtained from Hierarchical Clustering as the initial weights for the SOM. This approach might aim to enhance the quality and performance of the Self-Organizing Map (SOM).

## References

- Understanding Self-Organising Map Neural Network with Python Code: https://towardsdatascience.com/understanding-self-organising-map-neural-network-with-python-code-7a77f501e985
- Maha Fraj, Mohamed Aymen Ben Hajkacem, and Nadia Essoussi (2020): Self-Organizing Map for Multi-view Text Clustering https://doi.org/10.1007/978-3-030-59065-9_30
- Minisom github repository :https://github.com/JustGlowing/minisom
- How to make clustering explainable
- https://towardsdatascience.com/how-to-make-clustering-explainable-1582390476cc
- An Introduction to SHAP Values and Machine Learning Interpretability https://www.datacamp.com/tutorial/introduction-to-shap-values-machine-learning-interpretability
- Shap https://shap.readthedocs.io/en/latest/
- Tf-idf and doc2vec hyperparameters tuning https://medium.com/betacom/hyperparameters-tuning-tf-idf-and-doc2vec-models-73dd418b4d