Questions (1), (2), and (3) are related to each other and you may iterate over those three questions together to improve your results. In all three questions you will be working with the dataset "debates_2022.csv" (available in TeachCenter), which includes transcripts of all talks in the European parliament in 2022 with some additional metadata. All talk transcripts are in English. Your goal in questions (1), (2), and (3) is to extract the most important topics of these talks by clustering the talks.

1. *Feature Engineering.* Extract the features from the talk transcripts by computing tf-idf scores for words. You can use TfIdfVectorizer. Read the documentation of the vectorizer carefully and decide on the parameters you want to use to obtain most informative features. Before the feature extraction decide whether you need preprocessing including (among others) removal of non-informative instances.

   (a) Describe preprocessing steps if any. **Max. two sentences.**

   (b) Describe the parameters that you set for the vectorizer. Explain your reasoning? **Max. one sentence per parameter.**

   (c) How many features did you extract? Why? **Max. one sentence.**

   **Answer (a)** - Preprocessing:

   - All talks with less than 10 words are removed, since they are very short talk and non-informative.

   **Answer (b)** - Feature computation:

   - Max 2000 features: limit the volcabulary to most frequent words.
   - Stop words: English_stop_words plus some custom stop words to remove non-descriptive words
   - max_df = 0.85: Ignore very common words since they wont help us distinguish different topics
   - min_df = 5: Ignore very specific terms that only show up in less than 5 talks.
   - ngram_range = (1,2): allows the vectorizer to extract features for both individual words (unigrams) and two-word phrases (bigrams), capturing more contextual meaning.
   - lower case all words: same words with different capital letters still have same meaning, and should be treated as same words.

   **Answer (c)** - Number of features:

   - I extracted 2000 features, since too high might produce too much noise, and too low might not include important words that help us identify the topics

2. *Clustering.* Using the features that you extracted implement a clustering method of your choice. Use an appropriate evaluation metric to evaluate the quality of your clustering result.

   (a) What is your clustering algorithm and why? **Max. two sentences.**

   (b) How many clusters did you extract? How did you decide on the number of clusters. **Max. one sentence.**

   (c) Which evaluation metric did you use to evaluate your results. What is your evaluation score? **Max. two sentences.**

   (d) Interpret your clusters, e.g., by looking into ten most important words in each cluster. **Max. one sentence per cluster.**

**Answer (a)** - Clustering algorithm:

- I used K-means algorithm
- It is conceptually simple and computationally efficient.
- K-means tries to separate data points into groups, which is useful for us to identify the topics of the debates

**Answer (b)** - Number of clusters:

- I tried a range of different K from 1 to 20, and with 20 clusters i archived the highest silhouette score, so it was chosen as the final k =20

**Answer (c)** - Evaluation:

- I used Silhouette to evaluate each of the number of clusters k. With k = 20 i got the highest score of 0.0194

**Answer (d)** - Interpretation:

- Cluster 1: Dominated by procedural terms , likely noise and not any actual topic
- Cluster 2: Focused on human rights with a global perspective
- Cluster 3: Specific to China, Taiwan, and Hong Kong, with a focus on human rights and trade
- Cluster 4: Focused on democracy and media freedom
- Cluster 5: Energy policy and pricing
- Cluster 6: EU enlargement and Western Balkans, with Moldova and Georgia
- Cluster 7: Humanitarian issues, including children, refugees, and Ukraine
- Cluster 8: Social and economic policy, including budget and cohesion
- Cluster 9: Cultural heritage, with a focus on Ukraine and Azerbaijan
- Cluster 10: Digital policy and taxation
- Cluster 11: Health issues, including mental health and cancer
- Cluster 12: Ukraine-Russia conflict
- Cluster 13: Youth issues, including mental health and education
- Cluster 14: Foreign policy and security, focusing on Africa and Turkey
- Cluster 15: Climate change and environmental policy
- Cluster 16: Dominated by procedural and conversational terms, because these data points are noise and not valid topic
- Cluster 17: Rule of law, focused on Hungary and Poland
- Cluster 18: Gender issues and women's rights
- Cluster 19: Schengen area expansion
- Cluster 20: Agriculture and food security

In summary, we found 18 topics and 2 clusters (cluster 1 and 16) that are noise and not any valid topics

3. *Dimensionality Reduction for Visualization.* Perform dimensionality reduction with PCA on the features that you extracted previously. Use your clustering results and plot data points in 2D PCA space with clusters as colors for your data points.

   (a) Your plot.

   (b) Are clusters well separated in your plot? **Max. one sentence.**

   (c) Interpret the PCA dimensions that you used for visualization. **Max. one sentence per dimension.**

**Answer (a)** - Your plot:

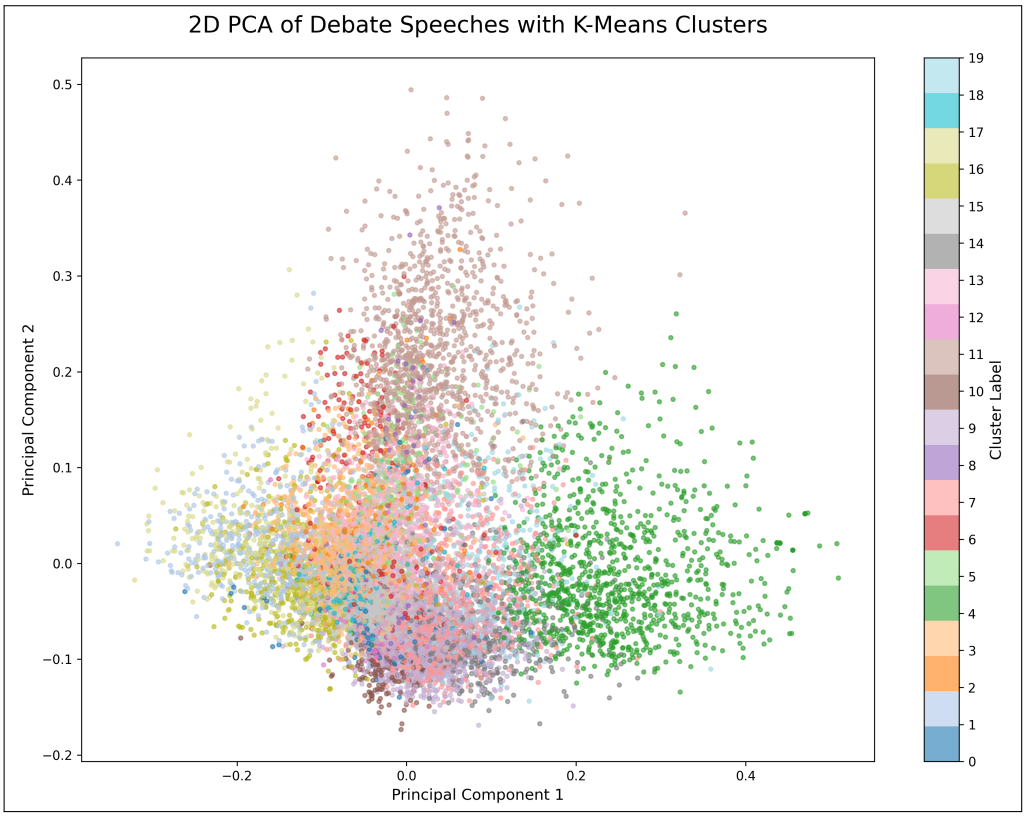


Figure 1: Cluster quality vs. number of clusters

**Answer (b)** - Cluster separation:

- All cluster are not well separated, especially some clusters heavily overlapping each other.

**Answer (c)** - Interpretation:

- PCA-1: This dimension captures the largest portion of variance in the TF-IDF vectorized debate texts, likely representing the most common themes or terms across all debates
- PCA-2: This dimension accounts for the second-largest variance, reflecting secondary thematic differences orthogonal to the first component.

4. *Classification.* Given the dataset "king_rook_vs_king.csv" (available in TeachCenter), with data on chess endgames featuring the white king and a white rook against the black king, implement a classifier of your choice to predict whether the white will win. Each endgame is described by the rank and file positions of the white king, the white rook, and the black king (six features in total). The target variable is the depth of white win (a categorical variable with either draw or zero, one, ..., sixteen indicating that the white wins in that many moves). Transform the target variable to obtain the win depth levels as:

   - draw: 0
   - zero, one, two, three, four: 1
   - five, six, seven, eight: 2
   - nine, ten, eleven, twelve: 3
   - thirteen, fourteen, fifteen, sixteen: 4.

   Use this new variable as your classification target. Evaluate your classifier by a metric of your choice. If your model has hyperparameters cross-validate.

   (a) Describe preprocessing and feature transformations steps if you made any. **Max. two sentences.**

   (b) What is your model and why? **Max. two sentences.**

   (c) Describe your evaluation setup. **Max. one sentence.**

   (d) Describe hyperparameter optimization if any. Give the final values of hyperparameters. **Max. two sentences.**

   (e) Give your evaluation results as text or a table.

   **Answer (a)** - Preprocessing & feature transformations:

   - Use ordinal encoding to transform the features from 'a', 'b', ... to numbers 1, 2, .... The labels are transformed from string to numbers from 0 to 4 as described by the task

   **Answer (b)** - Model choice:

   - My model is LightGBM , which is a model built on the strength of Decision Tree. It has high performance, speed, and ability to model complex, non-linear relationships, which is exactly what this chess endgame problem requires.

   **Answer (c)** - Evaluation setup:

   - I use cross validation since the dataset is imbalanced, with the main metric being Balanced Accuracy Score (average of recall of each class). I also include a classification report for the final model

   **Answer (d)** - Hyperparameters:

   - I use Optuna library to help optimizing the number of estimators, the learning rate, number of leaves, max depth, alpha and lambda (for L1 and L2 regularization), feature fraction and subsample.
   - Final hyperparameters: 'n_estimators': 1582, 'learning_rate': 0.08, 'num_leaves': 67, 'max_depth': 18, 'reg_alpha': 0.0, 'reg_lambda': 0.0, 'colsample_bytree': 0.61, 'subsample': 0.84

   **Answer (e)** - Results:

   - Balanced accuracy: 0.9761
   - Classification report:

Table 1: Classification Report

| **Class** | **Precision** | **Recall** | **F1-score** | **Support** |
|---|---|---|---|---|
| 0 | 1.00 | 0.99 | 0.99 | 559 |
| 1 | 0.97 | 0.96 | 0.96 | 126 |
| 2 | 0.96 | 0.96 | 0.96 | 636 |
| 3 | 0.98 | 0.98 | 0.98 | 2030 |
| 4 | 0.99 | 0.99 | 0.99 | 2261 |
| **Accuracy** | | | 0.98 | 5612 |
| **Macro Avg** | 0.98 | 0.98 | 0.98 | 5612 |
| **Weighted Avg** | 0.98 | 0.98 | 0.98 | 5612 |