You should limit the length of your answers as indicated in the questions!
Not following these limits will result in deduction of points!

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:

   - Tokenization and Lowercasing: The text is split into individual words, and all words are converted to lowercase to ensure uniformity.
   - Removal of Stop Words: Commonly used words like "the", "and", "is", etc., are removed since they do not carry significant meaning in topic modeling.

   Answer (b) - Feature computation:

   - max_df=0.90: Words that appear in more than 90% of the talks are ignored.
   - min_df= 10: Words that appear in fewer than 10 talks are ignored, reducing noise from infrequent terms that may not provide much value for clustering
   - ngram_range=(1, 1):To ensure that only single words are used as features in the feature extraction process.
   - stop_words='english': To remove common English stopwords
   - max_features= 10000: To limit the feature set to the top 10,000 most important words, helping to reduce the dimensionality and focus on the most relevant terms.

   Answer (c) - Number of features: 7689

   - The 7,689 features were extracted because, after filtering out very common and rare words and focusing on the most relevant ones, these 7,689 unique words remained as the key features for the model.

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:

- **K-Means Clustering** was chosen because it is known to be simple and scalable for large datasets. It is also considered effective when the number of clusters can be estimated in advance and works well with TF-IDF vectors due to their high dimensionality and sparsity.

Answer (b) - Number of clusters: 7 Clusters

- After testing cluster sizes from 1 to 10, 7 clusters were selected based on the Elbow Method, where the reduction in inertia began to level off—indicating a suitable balance between cluster cohesion and separation.

Answer (c) - Evaluation:

- The Silhouette Score was used to assess the quality of clustering, as it reflects how well each data point fits within its assigned cluster compared to others.
- The resulting score of 0.01038 indicates that the clusters are poorly defined, with significant overlap between them.

Answer (d) - Interpretation:

- Cluster 1: Focuses on parliamentary procedures and voting sessions, including agendas, points of order, and committee matters.
- Cluster 2: Covers EU policy discussions on climate, governance, and the roles of member states and the European Commission.
- Cluster 3: Centers on the Ukraine–Russia war, highlighting support for Ukraine and geopolitical tensions with Russia.
- Cluster 4: Addresses gender equality and women's rights, with emphasis on issues like abortion, violence, and discrimination.
- Cluster 5: Involves general political discourse on EU law, human rights, and the functioning of the European Parliament.
- Cluster 6: Reflects administrative and procedural communications such as statements, written discussions, and session closures.
- Cluster 7: Focuses on the European energy crisis, particularly gas prices, market conditions, and renewable energy solutions

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:

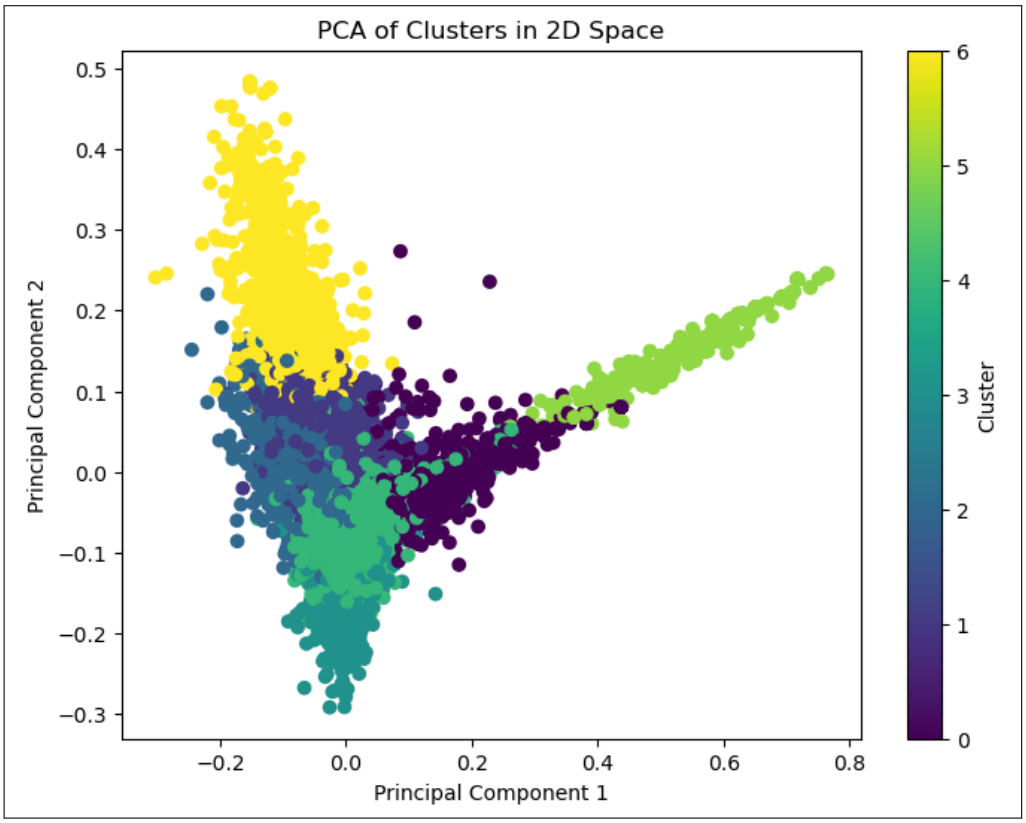

Figure 1: PCA of Clusters in 2D Space

Answer (b) - Cluster separation:

- Clusters show moderate separation with some overlap, suggesting distinguishable but not fully isolated thematic groupings.

Answer (c) - Interpretation:

- PCA-1 This dimension reflects parliamentary procedure and session management, as it is dominated by terms related to voting, agenda items, debates, and scheduling (e.g., "vote", "agenda", "minute", "session", "discussion").
- PCA-2 This dimension emphasizes policy discussions related to energy, climate, and crisis management, indicated by words such as "energy", "crisis", "gas", "renewable", and "climate".

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:

   - One-Hot Encoding was applied to the white_king_file, white_rook_file, and black_king_file columns to convert categorical values into binary features.
   - Transformed the white_depth_of_win column into win_depth_level by grouping win depths into 5 levels, then standardized the dataset using StandardScaler.

   Answer (b) - Model choice:

   - The **Decision Tree Classifier** was chosen because it performed better than both the Logistic Regression and Random Forest classifiers in terms of training and testing accuracy.
   - Its ability to capture complex patterns and provide clear decision rules made it a good fit for this task.

   Answer (c) - Evaluation setup:

   - The classifier was evaluated using accuracy and 4-fold cross-validation to ensure generalization, and the results included training and test accuracy, along with confusion metrics and a classification report.

   Answer (d) - Hyperparameters:

   - The best hyperparameters for the model were set to **class_weight=None**, meaning no special weighting was applied to the classes, and **criterion='entropy'**, which uses information gain to decide splits.
   - **max_depth=None** allows the tree to grow until all leaves are pure, while **min_samples_split=2** and **min_samples_leaf=1** ensure that nodes are split as long as there are at least 2 samples, and each leaf can have as few as 1 sample.

   Answer (e) - Results:

   - Classification Report:

	Precision	**Recall**	**F1-score**	**Support**
0	0.71	0.71	0.71	549
1	0.88	0.82	0.85	130
2	0.83	0.84	0.83	636
3	0.90	0.90	0.90	2072
4	0.94	0.94	0.94	2225
Accuracy			0.89	5612
Macro Avg	0.85	0.84	0.85	5612
Weighted Avg	0.89	0.89	0.89	5612