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:

   - Transcripts with missing or extremely short content were removed to eliminate non-informative instances.
   - The text was lowercased, stripped of punctuation, digits, and common stopwords, then lemmatized to reduce words to their base forms—effectively minimizing noise and improving feature quality.

   Answer (b) - Feature computation:

   - max_df=0.85, Filters out very common words that appear in over 85% of talks, as they add little value.
   - min_df= 5, Ignores rare words that appear in fewer than 5 talks to keep the focus on relevant terms.
   - ngram_range=(1, 1), Ensure that only single words are used as features in the feature extraction process.
   - stop_words='english', Removes common English words like "the" and "is" that don't add much meaning.
   - max_features= 10000 Limits to the 10,000 most useful terms to ensure efficiency without losing key information.

   Answer (c) - Number of features: 10000 Features

   - A total of 10,000 features were extracted to retain a rich yet manageable representation of the most informative terms across the corpus.

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 selected because it is efficient for high-dimensional data like TF-IDF vectors and works well when the number of clusters can be estimated. It's also widely used for topic discovery in text mining due to its simplicity and scalability.

Answer (b) - Number of clusters:

- **Eight clusters** were extracted using the **Elbow Method**, where a slight flattening of the inertia curve suggested a reasonable trade-off between compact clusters and distinct separation.

Answer (c) - Evaluation:

- The **Silhouette Score** was used to assess how well-separated the clusters are. The score was **0.0105**, indicating that the clustering structure is weak and the clusters are not well-defined.

Answer (d) - Interpretation:

- Cluster 1: Focuses on EU policy-making and climate-related discussions involving member states and the European Commission.
- Cluster 2: Highlights gender equality and women's rights, including issues like violence, abortion, and sexual abuse.
- Cluster 3: Covers legal and institutional matters in the EU, such as human rights, democracy, and parliamentary governance.
- Cluster 4: Centers on the European energy crisis, including gas prices, electricity, and the shift to renewables.
- Cluster 5: Discusses public health, youth well-being, mental health, and access to care and education.
- Cluster 6: Relates to voting processes, agenda points, and internal parliamentary procedures.
- Cluster 7: Involves formal communications such as statements, written discussions, and session scheduling.
- Cluster 8: Addresses the Ukraine–Russia war, including European support for Ukraine and condemnation of Russia.

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:

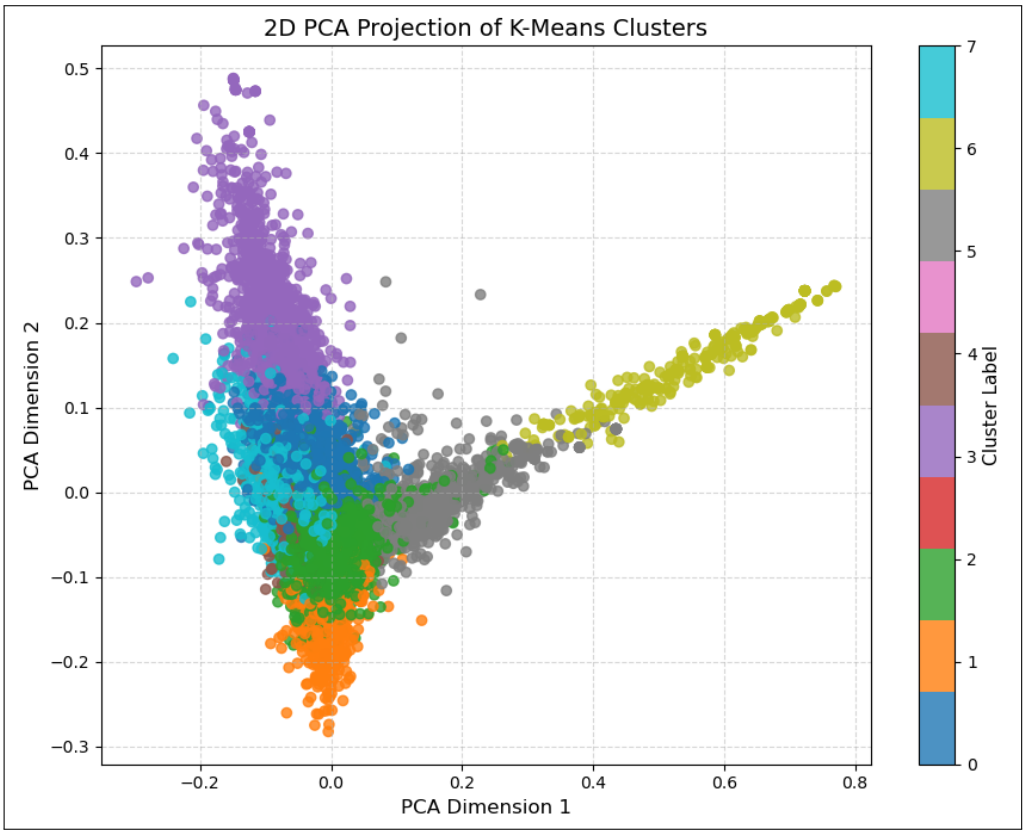

Figure 1: 2D PCA Projection of K-Means Clusters

Answer (b) - Cluster separation:

- The clusters are somewhat distinct but still overlap in places, indicating that while the topics are generally separable, there is some thematic blending between them.

Answer (c) - Interpretation:

- PCA-1 This axis seems to capture the routine business and formal activities of the European Parliament, like voting sessions, agenda items, and written statements.
- PCA-2 This dimension reflects the core policy debates, especially around urgent issues like energy prices, the climate crisis, and the broader economic situation.

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:

- The white depth values in the **'white_depth_of_win'** column were grouped into five levels to create a new **'win depth level'** feature, and 'white_depth_of_win' column was removed.
- One-Hot Encoding was then used to convert the categorical columns—**white_king_file**, **white_rook_file**, and **black_king_file** —into binary features, followed by standardizing the dataset with StandardScaler.

Answer (b) - Model choice:

- Tested Logistic Regression, Decision Tree, and Random Forest; Decision Tree performed best based on train and test accuracy.
- **Decision Tree Classifier** suits this task well as they handle categorical inputs and capture position-based patterns effectively.

Answer (c) - Evaluation setup:

- The model was evaluated using **accuracy** with 4-fold **cross-validation** to ensure stable performance across different data splits, and further assessed using a **confusion matrix** and **classification report** for a detailed view of class-wise performance.

Answer (d) - Hyperparameters:

- I used GridSearchCV to tune the Decision Tree Classifier for the best performance.
- The optimal hyperparameters found were: criterion='entropy', splitter='best', max_depth=None, min_samples_split=2, min_samples_leaf=1, with no restrictions on features or leaf nodes

Answer (e) - Results:

Table 1: Confusion Matrix for Win Depth Level Classification

	Pred 0	**Pred 1**	**Pred 2**	**Pred 3**	**Pred 4**
Actual 0	389	6	45	61	48
Actual 1	11	106	11	2	0
Actual 2	38	6	533	55	4
Actual 3	59	2	52	1874	85
Actual 4	48	0	4	87	2086

Table 2: Classification Report for Win Depth Level Classification

	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