Text as Data: Supervised & Unsupervised Learning Homework

Objective

In this assignment, you will apply the techniques covered in class to perform **Classification** (Supervised Learning) and **Topic Modeling** (Unsupervised Learning) on real-world text datasets.

Deadline

The deadline is **22.12.2025**.

Part 1: Supervised Learning - Sentiment Analysis

Dataset: You can use any dataset you want. Or the `reviews.csv` (Rotten Tomatoes Movie Reviews) dataset we used in class.

Tasks:

1. **Preprocessing**
   - Perform standard preprocessing: e.g., lowercasing, removing punctuation, and removing stop words. Justify your choice.
   - Split the dataset into training (80%) and testing (20%) sets.
2. **Vectorization**
   - Convert the text data into numerical features using a method of your choice (e.g., `CountVectorizer`, `TfidfVectorizer`).
   - You can limit the vocabulary size (e.g., `max_features=5000`) to keep the model manageable.
3. **Model Selection & Training**
   - Choose a supervised classification model (e.g., Logistic Regression, Naive Bayes, Support Vector Machine, etc.).
   - Train your chosen model on the training set.
   - *Briefly explain:* Why did you choose this particular model?
4. **Cross-Validation**
   - Perform **K-Fold Cross-Validation** (e.g., $k = 5$ or $k = 10$) on your training data.
   - Report the average performance metric (e.g., Accuracy or F1-Score) across the folds. Does the model perform consistently?
5. **Evaluation**

- Predict class on the *held-out* test set.
- Calculate and report the **Accuracy**, **Precision**, **Recall**, and **F1-Score**.
- Plot the **Confusion Matrix** to visualize the performance.

6. **Interpretability**
   - Extract the top 20 positive (loading on class 1) and top 20 negative (loading on class 2) features (words) based on the model's coefficients or feature importance.
   - *Bonus:* Select one misclassified text from the test set and use **LIME** or **SHAP** to explain why the model made that prediction.

Part 2: Unsupervised Learning - Topic Modeling

Dataset: Again, any dataset you wish, or the `sotu.csv` (State of the Union Addresses) we used in class.

Tasks:

1. **Preprocessing**
   - Preprocess the text (similar to Part 1, but consider removing common political stop words or context-specific noise).
2. **Topic Modeling with LDA & Hyperparameter Tuning**
   - Use `CountVectorizer` to create a Document-Term Matrix (LDA typically works better with raw counts).
   - Run the LDA algorithm multiple times, experimenting with the following:
     (a) **Number of Topics (k):** Experiment with at least three different values (e.g., $k \in \{5, 10, 20\}$).
     (b) **Hyperparameters α and β:** Experiment with different settings for α (doc-topic prior) and β (topic-word prior, often called `eta` in libraries like scikit-learn). Compare default settings vs. specific high/low values.
3. **Topic Analysis & Comparison**
   - For your experimental runs, display the top 10 words for the generated topics.
   - **Compare the results:**
     – How does the coherence of the topics change as k increases?
     – How do changes in α (document sparsity) and β (topic sparsity) affect the resulting topic distributions?
     – Which configuration provides the most interpretable results?
   - Assign human-readable labels to the topics from your "best" model.

Submission

Submit a single Jupyter Notebook (`.ipynb`) containing your code, visualizations, and textual answers to the analysis questions. Please submit your file to my email: petro.tolochko@univie.ac.at.