Political Bias Detector – Prototype Pipeline (Text → % Left / Center / Right)

Goal: Prototype that outputs % leaning (Left, Center, Right) from plain text input.

Scope: Fast, practical prototype (not production-perfect).

Pipeline:

1) Data Collection
- Use GDELT 2.0 Global Knowledge Graph (GKG) or any news dataset with text + source metadata.
- Derive labels from outlet-level bias lists (e.g., left/center/right). Start with ~5k–10k articles.

2) Label Strategy
- Map outlet bias → article label (Left=0, Center=1, Right=2).
- Balance classes; deduplicate identical or near-identical texts.

3) Preprocessing
- Clean text (remove HTML, boilerplate, excessive whitespace). Lowercase.
- Keep one language (e.g., English); filter/translate others. Train/Val/Test = 80/10/10.

4) Text Representation
- Option A (better): BERT embeddings from a pretrained model.
- Option B (faster): TF-IDF n-grams. Keep a fixed vocabulary.

5) Model (Prototype)
- Simple: Logistic Regression or Random Forest on embeddings.
- Stronger: Fine-tuned BERT classifier (3–5 epochs; small batch).

6) Training
- Multi-class cross-entropy. Use class weights if imbalanced.
- Early stopping on validation loss; save best model.

7) Evaluation
- Accuracy, Macro-F1, ROC-AUC per class. Confusion matrix for errors.
- Probability calibration check (optional).

8) Inference Pipeline
- User text → preprocess → embed → model → probabilities (% Left/Center/Right).
- Present calibrated percentages; optionally pick argmax as label.

9) Explainability (Optional)
- Use SHAP/LIME or attention attributions to highlight influential terms.

10) Packaging & Deployment (Prototype)
- Save vectorizer/tokenizer + model.
- Wrap in a simple API/UI for interactive testing.