# **Phase 2: Detect Trending Topics Per Category**

## Steps to Complete Phase 2 – Trend Detection

### **Preprocessing**

Created a Preprocessing Script File: **preprocess\_articles.py** inside the preprocessing/ folder. Implemented a **clean\_text()** function that lowercases the text, removes punctuation, tokenizes words, removes stopwords, and lemmatizes each word using NLTK. This function is applied to each article by combining the title and description fields into one.

The script processes all CSV files in data/raw/, applies the cleaning function, adds a new cleaned\_text column, and saves the cleaned files to data/processed/ with the same filenames.

### **Tokenization & Vectorization**

Implemented TF-IDF (TfidfVectorizer) to convert the cleaned article text into numerical vectors.

Each vector represents how important each word is across the articles in a given category.

This allows to identify the most significant terms in each topic group.

### **Topic Extraction / Trend Detection**

Using TF-IDF scores, extracted the top 10 highest-scoring keywords per category as representative "trending terms."

These keywords were printed per category, providing insight into which concepts were most heavily discussed in that category’s articles.

### **Ranking & Selection**

For each top keyword in a category, scanned the articles to find those that contained that keyword.

Then selected the first 3 matching articles per category, considering them as representative examples of the trending topics.

These articles are stored in the data/trends/ folder, one CSV per category, with fields: title, description, publishedAt, url, category, keyword.