Code Summary

1. Data Loading and Preprocessing:

- The code loads the dataset from a CSV file containing climate adaptation solutions.

- It uses `nltk` to tokenize the descriptions, convert them to lowercase, remove non-alphanumeric words, and eliminate common stopwords (like "the," "is," etc.).

2. Generating n-grams:

- The function `generate\_ngrams` creates n-grams (in this case, bigrams) from the cleaned descriptions.

- Bigrams represent pairs of consecutive words, useful for identifying word combinations that frequently appear together (e.g., "climate resilience").

3. Topic Modeling with `sklearn`:

- The code then performs topic modeling using Latent Dirichlet Allocation (LDA) from `sklearn`.

- It vectorizes the descriptions into a document-term matrix, then applies LDA to identify topics.

- The top 10 words for each topic are printed to give insights into each topic's theme.

4. Topic Modeling with `gensim`:

- The code also applies topic modeling using `gensim`'s LDA, which allows us to work with custom tokenized text and dictionaries.

- Gensim builds a more customizable model, where topics are generated with meaningful word distributions.

- Each topic is displayed with its most representative words, providing a human-readable summary of the underlying themes in the descriptions.

Output and Customization

- The code outputs sample bigrams, lists of keywords for each topic, and the primary themes.

- Key parameters (number of topics, n-gram size) can be adjusted to refine the output based on dataset characteristics and analysis needs.