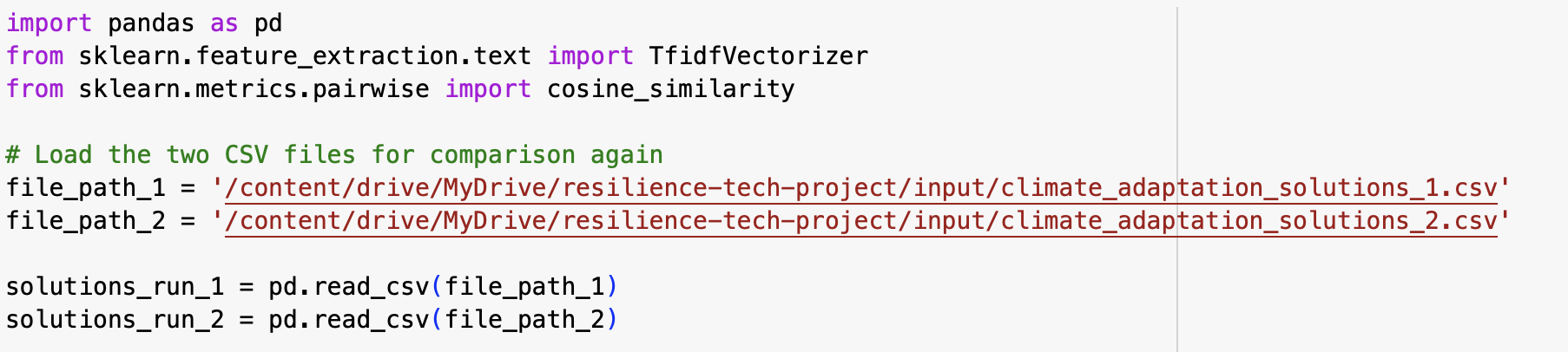
Identifying similar solutions based on TFIDF (Natural Language Processing).

This script compares two sets of climate adaptation solutions (`solutions\_run\_1` and `solutions\_run\_2`) to identify unique and common solutions based on similarity in their names (and optionally, technologies).:

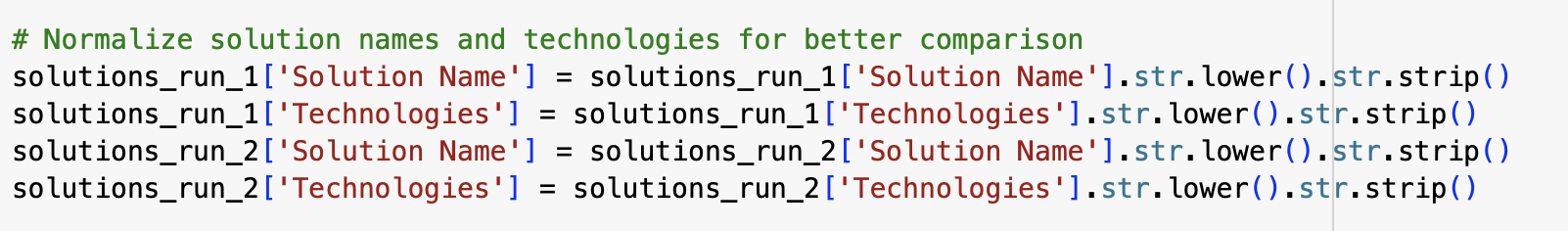
Script notebook: <https://drive.google.com/file/d/1k3geSVTHNF1MrLMGX1be1tUglWtOgWJ2/view?usp=share_link>

Script Summary:

1. Load Data: Two CSV files are loaded as dataframes. We compare two runs.



2. Text Normalization: The `Solution Name` and `Technologies` columns are converted to lowercase and stripped of extra spaces to standardize text for comparison.

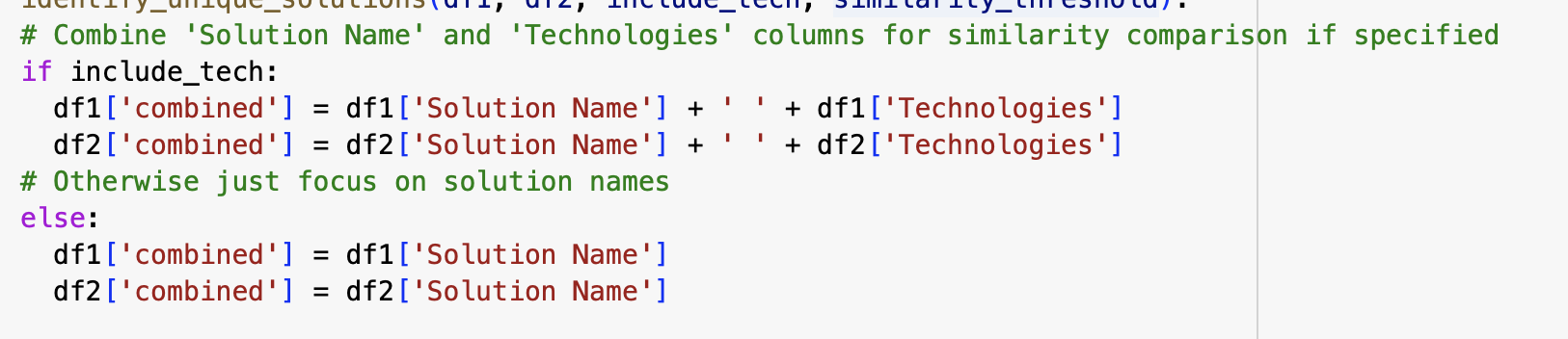


3. TF-IDF Vectorization: The script combines `Solution Name` (and optionally, `Technologies`) into a single field, creating a vector representation of each solution. TF-IDF (Term Frequency-Inverse Document Frequency) assigns higher weights to terms that are frequent in individual solutions but rare across the datasets, highlighting distinguishing words.

def identify\_unique\_solutions(df1, df2, include\_tech, similarity\_threshold)

**TF-IDF (Term Frequency-Inverse Document Frequency)** is a technique to measure the importance of words in a document relative to a collection of documents. It assigns higher weights to terms that appear frequently in a single document but rarely across others, helping capture distinctive words for comparison. TF-IDF vectors are then used to calculate cosine similarity, a metric that measures the angle between two text vectors, where closer angles (higher similarity scores) indicate more similar texts.

Two files are categorized as unique or similar based on similarity\_threshold. If we use both Technology and Solution name columns, it makes sense to set a lower similarity\_threshold to account for noisy data in the technologies description.



The function *identify\_unique\_solutions* takes two datasets, combines the Solution Name and Technologies columns (if include\_tech = True) or only considers Solution Name (if include\_tech = False) and computes cosine similarity based on their TF-IDF vectors and similarity threshold. It uses a threshold to filter unique solutions, outputting only those entries from each dataset that don't exceed the similarity threshold when compared to the other dataset.

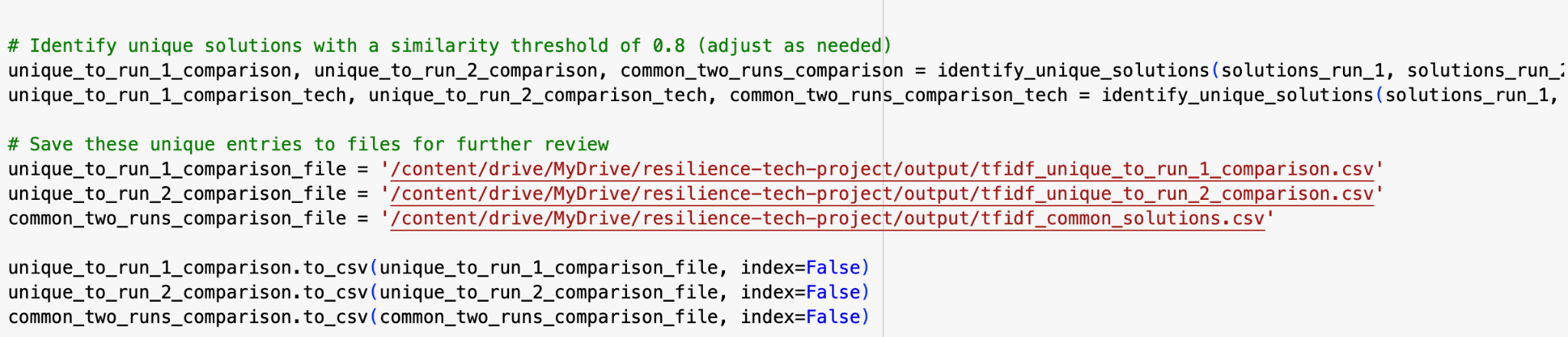
4. Cosine Similarity Calculation: Using the TF-IDF vectors, cosine similarity is computed between each pair of solutions from the two datasets. Cosine similarity, which measures the angle between text vectors, ranges from 0 (completely different) to 1 (identical).

5. Threshold-Based Filtering:

- Unique Solutions: Solutions with similarity scores below a specified threshold (e.g., 0.5) are considered unique to each dataset.

- Common Solutions: Solutions exceeding the similarity threshold are flagged as common.

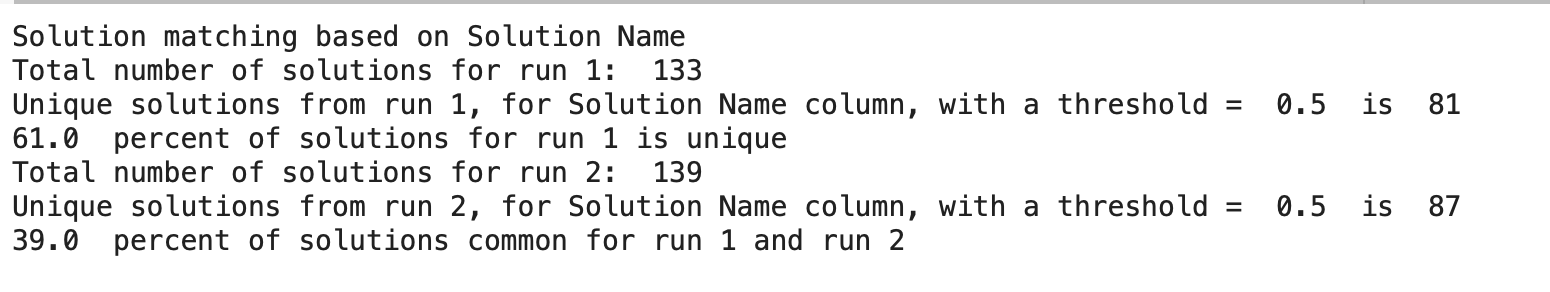


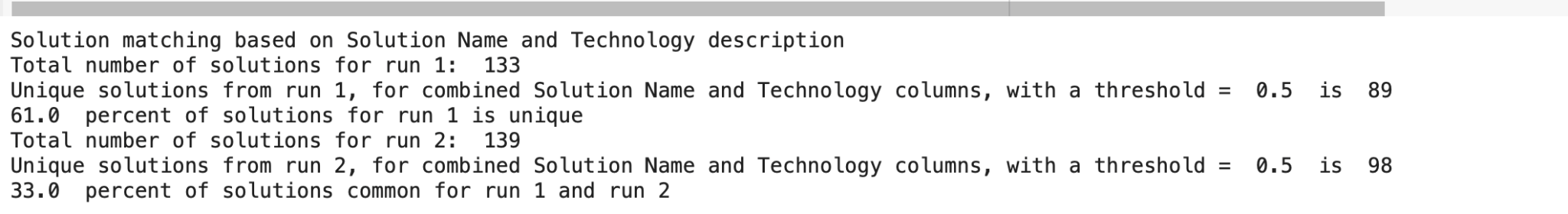
6. Save Results: The identified unique and common solutions are saved as separate CSV files for further analysis.

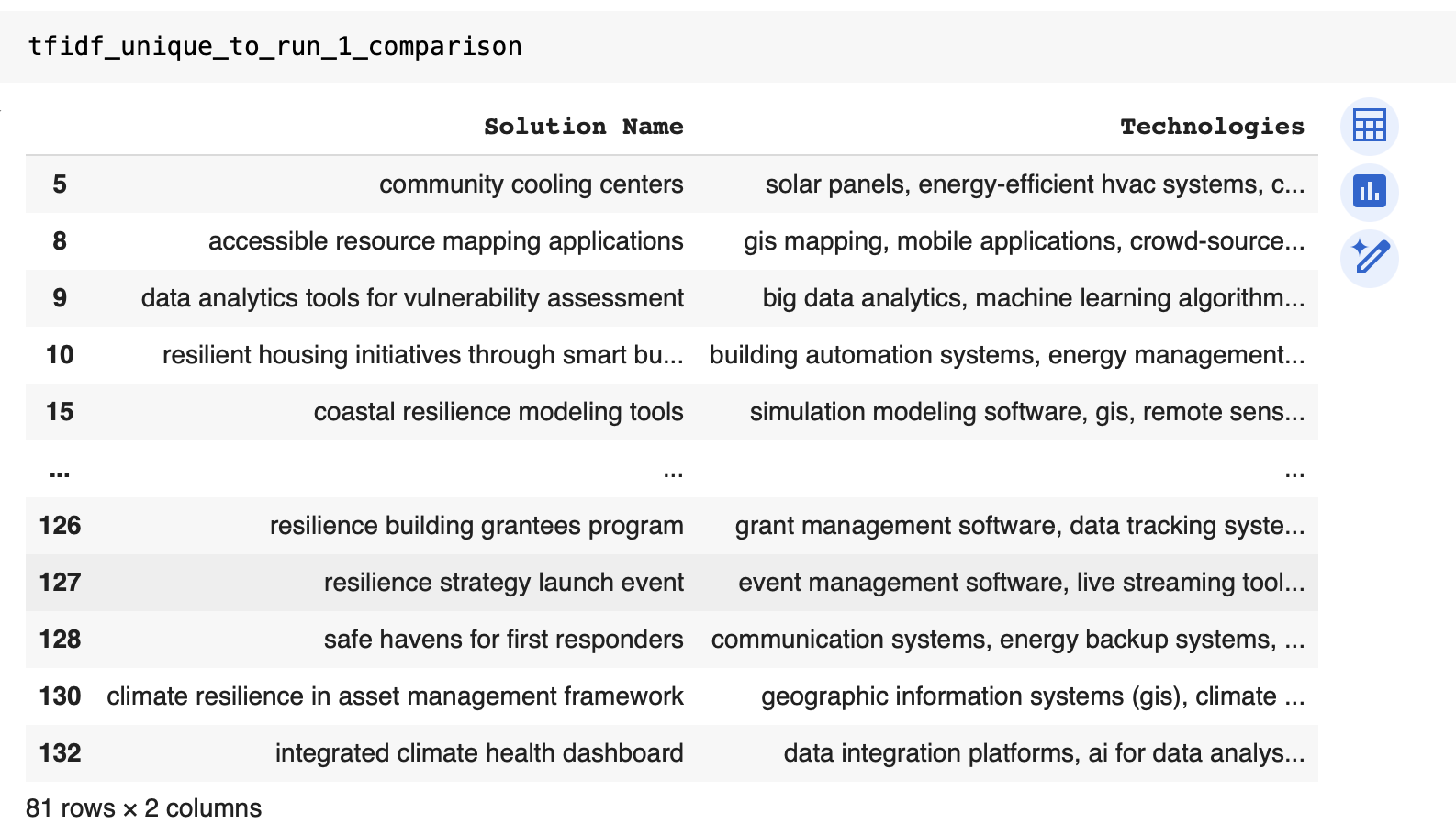
TF-IDF Purpose:

TF-IDF assigns a higher weight to distinctive words (those frequent in one solution but rare across all). This representation, paired with cosine similarity, helps identify solutions that are similar but not identical, even if they use different wording.

Results:







**Summary of Output Files**

- Unique to Run 1: Contains a total of 89 unique solutions. Sample solutions include:

- "Snow and ice management technologies" using autonomous vehicles, GPS technology, and IoT sensors.

- "Community cooling centers" with solar panels, energy-efficient HVAC systems, and community engagement platforms.

- "Emergency communication networks" utilizing mobile applications, cloud computing, and satellite communication systems.

- Unique to Run 2: Contains a total of 98 unique solutions. Sample solutions include:

- "Sustainable urban mobility solutions" involving electric vehicles, transportation sharing apps, and infrastructure for cycling and walking.

- "Fire weather monitoring system" with satellite imaging, meteorological sensors, and alert systems.

- "Landslide monitoring system" using ground-penetrating radar, satellite imagery, and real-time monitoring systems.

- Common Solutions: Includes 44 solutions common to both runs. Sample solutions include:

- "Resilient coastal infrastructure" with structural engineering technologies and real-time monitoring systems.

- "Wildfire risk assessment tools" utilizing remote sensing, machine learning, and GIS.

- "Community engagement platforms" leveraging online forums, mobile applications, and social media tools.

**Summary of Technology-Based Output Files**

- Unique to Run 1 (Tech): Contains 89 unique solutions based on technology. Sample solutions include:

- "Snow and ice management technologies" using autonomous vehicles, GPS technology, and IoT sensors.

- "Community cooling centers" with solar panels, energy-efficient HVAC systems, and community engagement platforms.

- "Emergency communication networks" utilizing mobile applications, cloud computing, and satellite communication systems.

- Unique to Run 2 (Tech): Contains 98 unique solutions based on technology. Sample solutions include:

- "Sustainable urban mobility solutions" involving electric vehicles, mobile apps for transportation sharing, and infrastructure for cycling and walking.

- "Fire weather monitoring system" with satellite imaging, meteorological sensors, and alert systems.

- "Landslide monitoring system" using ground-penetrating radar, satellite imagery, and real-time monitoring systems.

- Common Tech Solutions: Includes 44 solutions common across both runs based on technology. Sample solutions include:

- "Resilient coastal infrastructure" with structural engineering technologies and real-time monitoring systems.

- "Wildfire risk assessment tools" utilizing remote sensing, machine learning, and GIS.

- "Community engagement platforms" leveraging online forums, mobile applications, and social media tools.

**Summary analysis:**

1. Category Differences:

- Run 1: Solutions are often more specific to certain scenarios (e.g., "Snow and Ice Management Technologies," "Community Cooling Centers").

- Run 2: Some solutions encompass broader themes, combining multiple aspects (e.g., "Sustainable Urban Mobility Solutions," which addresses transportation as a whole).

2. Solution Name Variability:

- Some solutions appear in both runs with minor differences in naming. For instance:

- "Wildfire Risk Assessment Tools" in Run 1 versus "Fire Weather Monitoring System" in Run 2.

- This naming variability suggests slight inconsistencies in how similar solutions are labeled, potentially due to different keywords or terminologies.

3. Technologies Field:

- Both runs include similar technologies, such as "machine learning," "remote sensing," and "satellite imagery."

- Run 2 tends to list a broader range of technologies for some solutions (e.g., "Data analytics, satellite imagery" vs. Run 1 listing "Data analytics" alone). This indicates that Run 2 may capture more comprehensive technology descriptions.

4. Description Differences:

- Descriptions between runs are generally similar but may vary in wording. For example:

- Run 1 might describe, "Using IoT sensors for monitoring," while Run 2 uses, "Deploying IoT and real-time sensors."

- Although these differences don’t change the meaning significantly, they introduce variability that could affect how solutions are matched.

5. Unique Solutions:

- Each run has unique solutions not found in the other, such as:

- Run 1: "Emergency Communication Networks" and "Community Cooling Centers."

- Run 2: "Sustainable Urban Mobility Solutions" and "Landslide Monitoring System."

- These unique solutions highlight the tool’s capability to detect different solution nuances in each run, possibly based on varied extraction methods or data sources.

Summary: Although there is overlap between the two runs, differences in naming, categorization, and phrasing lead to some inconsistencies in identifying climate adaptation solutions. Harmonizing terminology and categorization could help in reducing discrepancies, creating a more uniform dataset across multiple extraction runs.