Author: Natalia Vera
Date: March 25, 2025
Course: ASD section E - Unit 3 Project

I. How are you going to differentiate between items that have the same value for every label?

1. Data Generation (Preventing Identical Items):

- **Constraint during generation:** When generating the dataset, I would implement a check to ensure no two items have identical label vectors. This could involve comparing each newly generated item against all previously generated items. If a match is found, the new item would be regenerated.
- **Adding a unique identifier:** Even if I enforce uniqueness on the labels, it is useful to add a unique identifier column, such as a numerical ID or the movie title itself. This allows for unambiguous identification of each item, regardless of label similarities.
- **Introduce more varied features:** If, after generation, I find that many items are too similar, I could expand the number of binary classifications or add categorical classifications. The more varied the features, the less likely duplicate entries will occur.

2. Game Logic (Handling Potential Duplicates):

- **Displaying all remaining options:** If the algorithm reaches a point where only identical-label items remain, it could display all of them and ask the user to select the correct one.

II. How will you pick the first question to ask? In each subtree, how will you pick the next question to ask?

1. Question Selection:

- **Split Ratio:**
  - I can look at the ratio of 1's and 0's for each feature. A question that splits the dataset roughly in half is generally better than one that creates very uneven subsets.
  - I would calculate the difference between the number of items that have a 1 for a feature and the number of items that have a 0. The feature with the smallest difference would be chosen
  - Repeat for each subtree to find the next question to ask.

- **Remaining Items:**

  - The question selection will change based on the remaining items. If only a few items remain, the question selection should focus on those items, and the labels that best differentiate them.

- **Decision Tree Implementation:**
    - This process naturally lends itself to a decision tree structure.
    - Each node in the tree represents a question, and each branch represents a possible answer (Yes/No).
    - The algorithm traverses the tree based on the user's answers, recursively selecting the next question at each node.