CS211 Quiz 3 Review

True/False

1. Vector embeddings always produce the same output length regardless of input data size.
2. Semantic similarity can only be calculated between pieces of text of the same length.
3. FLUX requires at least one text prompt to generate images.
4. In semantic search, exact keyword matching is more effective than using embeddings.
5. The quality of vector embeddings depends on the size of the training dataset used to create them.

Multiple Choice

6. Which statement about vector embeddings is correct?

   a) They can only be used with numerical data
   b) They transform data into fixed-length numerical representations
   c) They always increase the dimensionality of the data
   d) They can only be used for text processing

7. What is a key benefit of semantic search using embeddings?

   a) It requires less computational power than traditional search
   b) It can find related content even with different vocabulary
   c) It only works with short text fragments
   d) It provides faster indexing than keyword-based search

8. When working with FLUX for image generation, which is true?

   a) It can only generate artistic illustrations
   b) It works without any user input
   c) It uses natural language prompts to create images
   d) It only works with black and white images

9. Why do we calculate cosine similarity between embeddings?

   a) To measure the semantic relationship between items
   b) To reduce processing time
   c) To save storage space
   d) To validate the embedding process

DataFrames

10. Consider the following code:

    What will be printed from this code?

    ```
    import pandas as pd
    data = {
        'student': ['Emma', 'James', 'Sofia'],
        'score': [85, 92, 88],
        'subject': ['Math', 'Science', 'Math']
    }
    df = pd.DataFrame(data)
    print("Original DataFrame:")
    print(df)
    df = df.drop('score', axis=1)
    df.iloc[2, 1] = 'Art'
    print("\nModified DataFrame:")
    print(df)
    ```

11. Given the following DataFrame:

    ```
    import pandas as pd
    data = {
        'student': ['Emma', 'James', 'Sofia'],
        'score': [85, 92, 88]
    }
    df = pd.DataFrame(data)
    ```

    Write code to add a new column called ‘status’ with values [‘Fail’, ‘Pass’, ‘Pass’] and print the resulting DataFrame.