Code Understanding Report

Generated: 2025-05-05 23:25:19

This report presents automated insights based on large language models and code analysis tools.

File: app.py

Summary

- This code defines a function named vectorize that takes in one input, Text. It uses the TfidfVectorizer from the sklearn.feature_extraction.text module to transform the text into a vector format. It then returns the resulting array.

Function Definitions:

TfidfVectorizer() - This function initializes the TfidfVectorizer object which is a class from sklearn.feature_extraction.text that converts text into a matrix of token counts.

fit*transform(Text) - This method is used to fit the vectorizer on the - This function calculates the cosine similarity between two input documents. The function first calls a built-in Python function 'cosine*similarity' to calculate the cosine similarity between the two input documents, which is then returned by the function. The 'cosine_similarity' function calculates the cosine similarity between two vectors.

Note: 'cosine_similarity' is a standard method for calculating the cosine similarity between vectors in Python.

This is a simple example, in a real-world scenario you would likely have a lot more complexities, and perhaps the cosine_similarity - This function uses the cosine similarity between the text vectors of two students as a measure of plagiarism. This is achieved by comparing the similarity score obtained from the cosine similarity function. If the similarity score is above a certain threshold (e.g., 0.7), it means that the text vectors of the two students are plagiarised.

The function takes two parameters, a list of student vectors (s_vectors) and a list to store the plagiarism results. The function is then looping through each pair of students, calculating the cosine similarity between their text vectors,

Docstring

- ### Code: def vectorize(Text): return TfidfVectorizer().fit_transform(Text).toarray()

Docstring:

This function takes a list of text documents and returns a matrix of TF-IDF features.

Parameters: - Text: A list of strings, where each string is a document.

Returns: - A sparse - ### Code: def similarity(doc1, doc2): return cosine_similarity([doc1, doc2])

Docstring:

The function `similarity` takes two documents as input and returns the cosine similarity between them.

The `cosine_similarity` function is a part of the `sklearn.metrics.pairwise` module, - ### Code: def check*plagiarism(): global s*vectors for student*a, text*vector*a in s*vectors: new*vectors = s*vectors.copy() current*index = new*vectors.index((student*a, text*vector*a)) del *new*vectors[current*index] for student*b, text*vector*b in new*vectors: sim*score = similarity(text*vector*a, text*vector*b)[0][1] student*pair = sorted((student*a, student*b)) score = (student*pair[0], student*pair[1], sim*score) plagiarism*results.add(score) return plagiarism*results

Docstring:

This function checks for plagiarism in a list of text vectors. It does this by comparing each text vector with every other text vector in the list. The similarity between the two vectors is calculated using the `similarity` function,

Code Quality

Tool: `pylint`
Issues: 0`

```
text ************* Module tmpmcj3j01f C:
\Users\nmoha\AppData\Local\Temp\tmpmcj3j01f.py:21:8: W0602: Using
global for 's_vectors' but no assignment is done (global-
variable-not-assigned)
```

Conclusion

The code is defining a function named `vectorize` that transforms a text into a vector format using TfidfVectorizer from sklearn.feature_extraction.text. The output is an array. The function `similarity` is another function that takes two documents as input and returns the cosine similarity between them.