



New chat



```
e movie_index = get_index_from_title(movie_user_likes)

similar_movies = list(enumerate(cosine_sim[movie_index]))

## Step 7: Get a list of similar movies in descending order of similarity score
sorted_similar_movies = sorted(similar_movies, key=lambda x: x[1], reverse=True)
```



The code you've provided is implementing a movie recommendation system using a technique called collaborative filtering. The `get_index_from_title()` function retrieves the index of a movie in the dataset based on its title, and `cosine_sim` is a matrix of similarity scores between all pairs of movies in the dataset.

The `list(enumerate(cosine_sim[movie_index]))` creates a list of tuples where each tuple contains an index and the similarity score of a movie to the movie that the user likes.

`sorted_similar_movies = sorted(similar_movies, key=lambda x: x[1], reverse=True)` is sorting the list of tuple of (index, score) in descending order according to the similarity score.

So the final result is a list of tuples, where each tuple contains the index of a movie and its similarity score to the movie that the user likes, sorted in descending order of similarity score.



e ENUMERATE



`enumerate()` is a built-in Python function that allows you to loop over a collection (e.g., list, tuple, or string) and retrieve both the index and the value of each element. The basic syntax for using `enumerate()` is as follows:

Copy code

```
enumerate(iterable, start=0)
```

where `iterable` is the collection you want to loop over, and `start` is an optional argument that allows you to specify the index at which to start the enumeration. The default value of `start` is 0.

When you call `enumerate()` on a collection, it returns an enumerate object, which is an iterator that generates tuples containing the index and value of each element in the collection. Here is an example:

Copy code

```
>>> words = ["hello", "world"]
>>> for i, word in enumerate(words):
...     print(i, word)
...
0 hello
1 world
```

```
e def combine_features(row):
    try:
        return row['keywords'] + " " + row['cast'] + " " + row["genres"] + " " + row["director"]
    except:
        print ("Error:", row)

df["combined_features"] = df.apply(combine_features,axis=1)
```



