TECH ODYSSEY

ML TEAM - WEEK 4

Hey guys, Welcome to the **TechOdessey ML event** conducted by team DCS. Here you will be provided with a series of tasks you must complete within the given time. Ready to dip your toes in ml? Let's get started !!

Task 4:

This week's main aim is to **get started with a Content-Based Filtering System.**
The content of the movie (summary, cast, crew, keyword, tagline, etc.) is employed in this recommender system to determine its resemblance to other movies. The films that are most likely to be similar are then recommended.

1. **PLOT-BASED** recommender system
We shall be computing pairwise similarity scores based on the movie plots and recommending based on the same

1.1 Read about Term Frequency-Inverse Document Frequency (TF-IDF) vectors

1.2 Write down the formulas for the same, also mention the intuition behind the math in TF_IDF vectors

1.3 Construct a TF IDF matrix on the overview column

1.3.1 Import TfidfVectorizer from sklearn.feature_extraction

1.3.2 Define the vector objects with stop_words in English

1.3.3 Read about the concept of stop_words

1.3.3 Clean NAN values in the overview column

1.3.4 use the fit_tranform method on the overview column

1.4 Constructing Similarity score

1.4.1 Understand and write a short para on the various similarity scores(eg Cosine score,euclidean score etc)

1.4.2 Compute Cosine Score

$$\text{similarity} = \cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\|\|\mathbf{B}\|} = \frac{\sum\limits_{i=1}^{n} A_i B_i}{\sqrt{\sum\limits_{i=1}^{n} A_i^2} \sqrt{\sum\limits_{i=1}^{n} B_i^2}},$$

1.4.2.1 You can either do it the hard way using Numpy or use the Linear Kernel function from sklearn.

1.5 Construct a function that takes in movie titles as input and outputs the top 10 movies with high cosine score

1.5.1 Construct a map or dictionary with movie titles and Id’s

1.5.2 Calculate cosine score pairwise

1.5.3 Sort and Display the titles with the highest similarity score.

TECH ODYSSEY

THANK YOU!

DEVELOPER COMMUNITY SASTRA