CHANDIGARH UNIVERSITY
Discover. Learn. Empower.

Name: Johnson Kumar **Uid:** 232bcs12654

Topic: Book Recommender Using NLP: Scrape book summaries and recommend similar books.

Code:

import pandas as pd Untitled-2

```python
import pandas as pd
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.metrics.pairwise import cosine_similarity

# Load the dataset from GitHub
url = 'https://raw.githubusercontent.com/zygmuntz/goodbooks-10k/master/books.csv'
df = pd.read_csv(url)

# Show column names (for debugging)
print("Available Columns in the Dataset:", df.columns)

# Use 'original_title' as the substitute for summary/description
if 'original_title' in df.columns:
    df = df[['title', 'authors', 'average_rating', 'original_title']]
else:
    print("Column 'original_title' not found in the dataset.")
    exit()

# Drop rows with missing values
df = df.dropna().reset_index(drop=True)

# Vectorize titles using TF-IDF
tfidf = TfidfVectorizer(stop_words='english', max_features=5000)
tfidf_matrix = tfidf.fit_transform(df['original_title'])

# Compute cosine similarity matrix
cosine_sim = cosine_similarity(tfidf_matrix, tfidf_matrix)

# Reverse index mapping
indices = pd.Series(df.index, index=df['title']).drop_duplicates()

```

```python

# Recommendation function
def recommend(title, num_recommendations=5):
    # Try to match the input with existing titles (case-insensitive)
    matches = df[df['title'].str.contains(title, case=False, na=False)]
    if matches.empty:
        print(f"No books found matching: {title}")
        return None
    idx = matches.index[0]  # First matching book
    sim_scores = list(enumerate(cosine_sim[idx]))
    sim_scores = sorted(sim_scores, key=lambda x: x[1], reverse=True)
    sim_scores = sim_scores[1:num_recommendations+1]
    book_indices = [i[0] for i in sim_scores]
    return df[['title', 'authors', 'average_rating']].iloc[book_indices]

# 🔹 Let user enter a book title
user_input = input("Enter a book title to get recommendations: ")
recommended_books = recommend(user_input)

# 🔹 Show recommendations
if recommended_books is not None:
    print("\nRecommended Books:\n", recommended_books)
```

Output:

```
OUTPUT  DEBUG CONSOLE  TERMINAL  PROBLEMS  PORTS

PS D:\Code\Python> python -u "d:\Code\Python\tempCodeRunnerFile.python"
Available Columns in the Dataset: Index(['book_id', 'goodreads_book_id', 'best_book_id', 'work_id',
       'books_count', 'isbn', 'isbn13', 'authors', 'original_publication_year',
       'original_title', 'title', 'language_code', 'average_rating',
       'ratings_count', 'work_ratings_count', 'work_text_reviews_count',
       'ratings_1', 'ratings_2', 'ratings_3', 'ratings_4', 'ratings_5',
       'image_url', 'small_image_url'],
      dtype='object')
Enter a book title to get recommendations: Twilight

Recommended Books:
                                      title                    authors  average_rating
1492  Crossroads of Twilight (Wheel of Time, #10)          Robert Jordan            3.82
4095                Twilight (The Mediator, #6)              Meg Cabot            4.26
6735                  The Servants of Twilight  Leigh Nichols, Dean Koontz            3.81
6836    Tempt Me at Twilight (The Hathaways, #3)  Lisa Kleypas, Rosalyn Landor            4.18
7932   Twilight (Warriors: The New Prophecy, #5)                Erin Hunter            4.30
```

```
OUTPUT   DEBUG CONSOLE   TERMINAL   PROBLEMS   PORTS

PS D:\Code\Python> python -u "d:\Code\Python\tempCodeRunnerFile.python"
Available Columns in the Dataset: Index(['book_id', 'goodreads_book_id', 'best_book_id', 'work_id',
       'books_count', 'isbn', 'isbn13', 'authors', 'original_publication_year',
       'original_title', 'title', 'language_code', 'average_rating',
       'ratings_count', 'work_ratings_count', 'work_text_reviews_count',
       'ratings_1', 'ratings_2', 'ratings_3', 'ratings_4', 'ratings_5',
       'image_url', 'small_image_url'],
      dtype='object')
Enter a book title to get recommendations: Johnson kumar
No books found matching: Johnson kumar
PS D:\Code\Python>
```

```
OUTPUT   DEBUG CONSOLE   TERMINAL   PROBLEMS   PORTS

PS D:\Code\Python> python -u "d:\Code\Python\tempCodeRunnerFile.python"
Available Columns in the Dataset: Index(['book_id', 'goodreads_book_id', 'best_book_id', 'work_id',
       'books_count', 'isbn', 'isbn13', 'authors', 'original_publication_year',
       'original_title', 'title', 'language_code', 'average_rating',
       'ratings_count', 'work_ratings_count', 'work_text_reviews_count',
       'ratings_1', 'ratings_2', 'ratings_3', 'ratings_4', 'ratings_5',
       'image_url', 'small_image_url'],
      dtype='object')
Enter a book title to get recommendations: A Game of Thrones

Recommended Books:
                                          title                                        authors  average_rating
3178      A Game of Thrones: The Graphic Novel, Vol. 1  Daniel Abraham, George R.R. Martin, Tommy Patt...            4.48
839                                The Westing Game                                  Ellen Raskin            4.03
4731  The Game: Penetrating the Secret Society of Pi...                                Neil Strauss            3.73
6712                            The Damnation Game                                   Clive Barker            3.82
7238               The Vor Game (Vorkosigan Saga, #6)                           Lois McMaster Bujold            4.28
PS D:\Code\Python>
```