

```

In [9]: import pandas as pd
        from sklearn.feature_extraction.text import TfidfVectorizer
        from sklearn.metrics.pairwise import cosine_similarity

In [10]: df = pd.read_csv('https://raw.githubusercontent.com/rashida048/Some-NLP-Projects/main/movies.csv')

In [11]: df['overview'] = df['overview'].fillna('')

In [12]: tfidf = TfidfVectorizer(stop_words='english')

In [13]: tfidf_matrix = tfidf.fit_transform(df['overview'])

In [14]: cosine_sim = cosine_similarity(tfidf_matrix, tfidf_matrix)

In [15]: def get_recommendations(title):
        idx = df[df['title'] == title].index[0]
        sim_scores = list(enumerate(cosine_sim[idx]))
        sim_scores = sorted(sim_scores, key=lambda x: x[1], reverse=True)
        sim_scores = sim_scores[1:11]
        movie_indices = [i[0] for i in sim_scores]
        return df['title'].iloc[movie_indices]

In [17]: print(get_recommendations('The Dark Knight Rises'))

65                The Dark Knight
299                Batman Forever
428                Batman Returns
1359               Batman
3854  Batman: The Dark Knight Returns, Part 2
119                Batman Begins
2507               Slow Burn
9      Batman v Superman: Dawn of Justice
1181               JFK
210                Batman & Robin
Name: title, dtype: object

In [18]: print(get_recommendations('Batman'))

3      The Dark Knight Rises
119   Batman Begins
65    The Dark Knight
428   Batman Returns
210   Batman & Robin
299   Batman Forever
9     Batman v Superman: Dawn of Justice
3857  Wicked Blood
1524   George and the Dragon
3389   Chairman of the Board
Name: title, dtype: object

In [24]: print(get_recommendations('Batman Forever'))

```

```
3          The Dark Knight Rises
119         Batman Begins
65         The Dark Knight
428        Batman Returns
210        Batman & Robin
3854       Batman: The Dark Knight Returns, Part 2
1359       Batman
4343       Cry_Wolf
174        The Incredible Hulk
9          Batman v Superman: Dawn of Justice
Name: title, dtype: object
```

In []: