In [44]:
```
import pandas as pd
```

Overview of the Movies Dataset (2,850 Records)

This dataset contains detailed information about movies, including movie ID, description, language, release date, ratings, writers, directors, cast, genres, and movie titles. It will be used as the base data for building the movie recommendation system.

In [45]:
```
df = pd.read_csv('C:/Users/Aditya Sharma/Desktop/DUCAT/ML-DL/ML_Dataset/movies_content.csv')
df
```

Out[45]:

	movie_id	description	language	released	rating	writer	director	cast	genre	name
0	tt5286444	Neerja is the story of the courageous Neerja B...	["Hindi"]	2016-02-19T00:00:00.000Z	7.9	["Saiwyn Quadras", "Sanyukta Shaikh Chawla"]	["Ram Madhvani"]	["Sonam Kapoor", "Shabana Azmi", "Yogendra Ti...	["Biography", "Drama", "Thriller"]	Neerja
1	tt4434004	A story that revolves around drug abuse in the...	["Hindi", "Panjabi"]	2016-06-17T00:00:00.000Z	7.9	["Sudip Sharma", "Abhishek Chaubey"]	["Abhishek Chaubey"]	["Alia Bhatt", "Shahid Kapoor", "Diljit Dosan...	["Crime", "Drama", "Thriller"]	Udta Punjab
2	tt0248126	Yashvardhan Raichand lives a very wealthy life...	["Hindi", "English", "Urdu"]	2001-12-14T00:00:00.000Z	7.5	["Karan Johar", "Sheena Parikh"]	["Karan Johar"]	["Amitabh Bachchan", "Jaya Bhaduri", "Shah Ru...	["Drama", "Musical", "Romance"]	Kabhi Khushi Kabhie Gham...
3	tt0347304	Naina, an introverted, perpetually depressed g...	["Hindi", "Urdu", "Gujarati", "Panjabi"]	2003-11-28T00:00:00.000Z	8.0	["Niranjan Iyengar", "Karan Johar"]	["Nikkhil Advani"]	["Shah Rukh Khan", "Preity Zinta", "Saif Ali ...	["Comedy", "Drama", "Romance"]	Kal Ho Naa Ho
4	tt3043252	'Parched' is a story about women set in the he...	["Hindi", "English"]	2016-06-17T00:00:00.000Z	7.6	["Supratik Sen", "Leena Yadav"]	["Leena Yadav"]	["Tannishtha Chatterjee", "Radhika Apte", "Le...	["Drama"]	Parched
...	...	...	...	...	...	...	...	...	...	...
2845	tt0308793	NaN	["Kannada"]	NaN	7.7	[]	["V. Ravichandran"]	["V. Ravichandran", "Kushboo", "Anant Nag", "...	["Crime"]	Ranadheera
2846	tt3006508	NaN	["Kannada"]	2013-02-22	7.7	[]	["Nagashekar"]	["Chethan Kumar", "Nithya	["Romance"]	Myna

	movie_id	description	language	released	rating	writer	director	cast	genre	name
								Menon", "Sarath Kum...		
2847	tt5348692	The plot is simple. Ranganna (Rangayana Raghu)...	["Kannada"]	2013-11-29	7.7	["Pc Shekhar"]	["Pc Shekhar"]	["Rangayana Raghu", "Sadhu Kokila", "Avinash"]	["Comedy", "Drama"]	Chaddi Dosth
2848	tt3171754	NaN	["Kannada"]	2013-12-27	7.6	[]	["Manju Swaraj"]	["Ganesh", "Amulya", "Anant Nag", "Tara", "Av...	["Comedy", "Romance"]	Shravani Subramanya
2849	tt0231455	NaN	["Kannada"]	NaN	7.5	[]	["S.R. Puttana Kanagal"]	[]	[]	Dharmasere

2850 rows × 10 columns

Selecting Important Columns for the Recommendation System

This step filters the dataset to retain only the essential features required for building the content-based recommendation model. The selected columns are `movie_id`, `name`, `description`, `language`, `director`, and `genre`.

In [46]:
```
df=df[['movie_id','name','description','language','director','genre']]
df
```

Out[46]:

	movie_id	name	description	language	director	genre
0	tt5286444	Neerja	Neerja is the story of the courageous Neerja B...	["Hindi"]	["Ram Madhvani"]	["Biography", "Drama", "Thriller"]
1	tt4434004	Udta Punjab	A story that revolves around drug abuse in the...	["Hindi", "Panjabi"]	["Abhishek Chaubey"]	["Crime", "Drama", "Thriller"]
2	tt0248126	Kabhi Khushi Kabhie Gham...	Yashvardhan Raichand lives a very wealthy life...	["Hindi", "English", "Urdu"]	["Karan Johar"]	["Drama", "Musical", "Romance"]
3	tt0347304	Kal Ho Naa Ho	Naina, an introverted, perpetually depressed g...	["Hindi", "Urdu", "Gujarati", "Panjabi"]	["Nikkhil Advani"]	["Comedy", "Drama", "Romance"]
4	tt3043252	Parched	'Parched' is a story about women set in the he...	["Hindi", "English"]	["Leena Yadav"]	["Drama"]
...	...	...	...	...	...	...
2845	tt0308793	Ranadheera	NaN	["Kannada"]	["V. Ravichandran"]	["Crime"]
2846	tt3006508	Myna	NaN	["Kannada"]	["Nagashekar"]	["Romance"]
2847	tt5348692	Chaddi Dosth	The plot is simple. Ranganna (Rangayana Raghu)...	["Kannada"]	["Pc Shekhar"]	["Comedy", "Drama"]
2848	tt3171754	Shravani Subramanya	NaN	["Kannada"]	["Manju Swaraj"]	["Comedy", "Romance"]
2849	tt0231455	Dharmasere	NaN	["Kannada"]	["S.R. Puttana Kanagal"]	[]

2850 rows × 6 columns

Checking for Duplicate Records in the Dataset

This step prints the number of duplicate rows in the processed dataframe. A result of `0` indicates that the dataset contains no duplicate entries.

In [47]:
```
print(df.duplicated().sum())
```

```
0
```

Removing Missing Values from the Dataset

Rows containing any missing values are removed to ensure clean and reliable input for the recommendation system. After filtering, 1,752 rows remain from the original 2,850 records.

In [48]:
```
df=df.dropna()
df
```

Out[48]:

	movie_id	name	description	language	director	genre
0	tt5286444	Neerja	Neerja is the story of the courageous Neerja B...	["Hindi"]	["Ram Madhvani"]	["Biography", "Drama", "Thriller"]
1	tt4434004	Udta Punjab	A story that revolves around drug abuse in the...	["Hindi", "Panjabi"]	["Abhishek Chaubey"]	["Crime", "Drama", "Thriller"]
2	tt0248126	Kabhi Khushi Kabhie Gham...	Yashvardhan Raichand lives a very wealthy life...	["Hindi", "English", "Urdu"]	["Karan Johar"]	["Drama", "Musical", "Romance"]
3	tt0347304	Kal Ho Naa Ho	Naina, an introverted, perpetually depressed g...	["Hindi", "Urdu", "Gujarati", "Panjabi"]	["Nikkhil Advani"]	["Comedy", "Drama", "Romance"]
4	tt3043252	Parched	'Parched' is a story about women set in the he...	["Hindi", "English"]	["Leena Yadav"]	["Drama"]
...	...	...	...	...	...	...
2838	tt5872120	Run Antony	Antony is depressed and wants to commit suicid...	["Kannada"]	["Raghu Shastry"]	["Action", "Romance", "Thriller"]
2841	tt3666724	Jolly Days	A bubbly tale, the story is about four pairs o...	["Kannada"]	["M.D. Sridhar"]	["Drama"]
2842	tt5652478	The Great Story of Sodabuddi	The film tells the story of Sodabuddi, who spe...	["Kannada"]	["Jyothirao Mohith"]	["Crime", "Drama", "Romance"]
2844	tt5508936	Vamshi	Vamshi, under the eye of his mother is living ...	["Kannada"]	["Prakash"]	["Action", "Drama", "Family"]
2847	tt5348692	Chaddi Dosth	The plot is simple. Ranganna (Rangayana Raghu)...	["Kannada"]	["Pc Shekhar"]	["Comedy", "Drama"]

1752 rows × 6 columns

In [49]:
```
df=df.reset_index(drop=True)
```

Creating a Combined Feature Column for Text Processing

A new column named `details` is created by concatenating the movie description, language, director, and genre fields. This combined text will be used later for vectorization and similarity computation in the recommendation model.

In [50]:
```
df['details']=df['description']+' '+df['language']+' '+df['director']+' '+df['genre']
df
```

Out[50]:

	movie_id	name	description	language	director	genre	details
0	tt5286444	Neerja	Neerja is the story of the courageous Neerja B...	["Hindi"]	["Ram Madhvani"]	["Biography", "Drama", "Thriller"]	Neerja is the story of the courageous Neerja B...
1	tt4434004	Udta Punjab	A story that revolves around drug abuse in the...	["Hindi", "Panjabi"]	["Abhishek Chaubey"]	["Crime", "Drama", "Thriller"]	A story that revolves around drug abuse in the...
2	tt0248126	Kabhi Khushi Kabhie Gham...	Yashvardhan Raichand lives a very wealthy life...	["Hindi", "English", "Urdu"]	["Karan Johar"]	["Drama", "Musical", "Romance"]	Yashvardhan Raichand lives a very wealthy life...
3	tt0347304	Kal Ho Naa Ho	Naina, an introverted, perpetually depressed g...	["Hindi", "Urdu", "Gujarati", "Panjabi"]	["Nikkhil Advani"]	["Comedy", "Drama", "Romance"]	Naina, an introverted, perpetually depressed g...
4	tt3043252	Parched	'Parched' is a story about women set in the he...	["Hindi", "English"]	["Leena Yadav"]	["Drama"]	'Parched' is a story about women set in the he...
...	...	...	...	...	...	...	...
1747	tt5872120	Run Antony	Antony is depressed and wants to commit suicid...	["Kannada"]	["Raghu Shastry"]	["Action", "Romance", "Thriller"]	Antony is depressed and wants to commit suicid...
1748	tt3666724	Jolly Days	A bubbly tale, the story is about four pairs o...	["Kannada"]	["M.D. Sridhar"]	["Drama"]	A bubbly tale, the story is about four pairs o...
1749	tt5652478	The Great Story of Sodabuddi	The film tells the story of Sodabuddi, who spe...	["Kannada"]	["Jyothirao Mohith"]	["Crime", "Drama", "Romance"]	The film tells the story of Sodabuddi, who spe...
1750	tt5508936	Vamshi	Vamshi, under the eye of his mother is living ...	["Kannada"]	["Prakash"]	["Action", "Drama", "Family"]	Vamshi, under the eye of his mother is living ...
1751	tt5348692	Chaddi Dosth	The plot is simple. Ranganna (Rangayana Raghu)...	["Kannada"]	["Pc Shekhar"]	["Comedy", "Drama"]	The plot is simple. Ranganna (Rangayana Raghu)...

1752 rows × 7 columns

Dropping Unnecessary Columns After Feature Combination

After creating the `details` column, the original fields (`description` , `language` , `director` , and `genre`) are removed from the dataframe to keep only the essential columns for further processing.

In [52]:
```
df=df.drop(['description','language','director','genre'],axis=1)
df
```

Out[52]:

	movie_id	name	details
0	tt5286444	Neerja	Neerja is the story of the courageous Neerja B...
1	tt4434004	Udta Punjab	A story that revolves around drug abuse in the...
2	tt0248126	Kabhi Khushi Kabhie Gham...	Yashvardhan Raichand lives a very wealthy life...
3	tt0347304	Kal Ho Naa Ho	Naina, an introverted, perpetually depressed g...
4	tt3043252	Parched	'Parched' is a story about women set in the he...
...	...	...	...
1747	tt5872120	Run Antony	Antony is depressed and wants to commit suicid...
1748	tt3666724	Jolly Days	A bubbly tale, the story is about four pairs o...
1749	tt5652478	The Great Story of Sodabuddi	The film tells the story of Sodabuddi, who spe...
1750	tt5508936	Vamshi	Vamshi, under the eye of his mother is living ...
1751	tt5348692	Chaddi Dosth	The plot is simple. Ranganna (Rangayana Raghu)...

1752 rows × 3 columns

Vectorizing Text Features and Computing Cosine Similarity

In this step, the combined `details` text is converted into numerical feature vectors using a vectorization technique (such as CountVectorizer or TF-IDF). These vectors are then used to calculate cosine similarity between movies to identify how closely they relate to each other.

In [53]:
```
from sklearn.feature_extraction.text import TfidfVectorizer
```

Generating TF-IDF Vectors Without Stop Words

TF-IDF vectorization is applied to the combined `details` column to convert the text into numerical feature vectors. The parameter `stop_words` is not used intentionally, as including all words may preserve useful context for movie descriptions and improve similarity detection.

In [54]:
```
tv=TfidfVectorizer(lowercase=True)
vectors=tv.fit_transform(df.details).toarray()
vectors
```

Out[54]:
```
array([[0., 0., 0., ..., 0., 0., 0.],
       [0., 0., 0., ..., 0., 0., 0.],
       [0., 0., 0., ..., 0., 0., 0.],
       ...,
       [0., 0., 0., ..., 0., 0., 0.],
       [0., 0., 0., ..., 0., 0., 0.],
       [0., 0., 0., ..., 0., 0., 0.]])
```

In [55]:
```
tv.get_feature_names_out()
```

Out[55]:
```
array(['000', '10', '100', ..., 'zoya', 'zubair', 'zubeida'], dtype=object)
```

In [56]:
```
from sklearn.neighbors import NearestNeighbors
```

Training the Nearest Neighbors Model Using Cosine Similarity

A Nearest Neighbors model is created with cosine distance as the similarity metric. The model is then fitted on the TF-IDF vectors so that similar movies can be efficiently retrieved based on their feature similarity.

In [57]:
```
model=NearestNeighbors(metric='cosine')
model.fit(vectors)
```

Out[57]:
```
NearestNeighbors(metric='cosine')
```

Finding Movies Similar to "Kal Ho Naa Ho"

To recommend movies similar to "Kal Ho Naa Ho", the following steps are performed:

1. Extract the `details` of the movie "Kal Ho Naa Ho" and convert it into a TF-IDF vector.
2. Pass this vector to the trained Nearest Neighbors model.
3. The model computes cosine similarity and returns the indices of the most similar movies based on the vector representation.

Interpreting the Similarity Results for "Kal Ho Naa Ho"

The Nearest Neighbors model returns two arrays:

1. **Cosine Distances:** `[[0.0, 0.8476, 0.8542, 0.8557, 0.8601]]`
    - A distance of `0.0` indicates the movie itself.
    - Smaller values indicate higher similarity; in this case, `0.8476` to `0.8601` are the closest matches.
2. **Indices of Similar Movies:** `[[3, 105, 146, 1038, 1287]]`
    - These are the row indices of the top 5 movies most similar to "Kal Ho Naa Ho".
    - Using these indices, we can retrieve the movie names from the dataframe for recommendations.

Generating Movie Recommendations for "Parched"

The recommendation system finds movies similar to "Parched" using the following steps:

1. Locate the index of "Parched" in the dataframe and extract its TF-IDF vector.
2. Pass the vector to the trained Nearest Neighbors model to compute cosine similarity.
3. Retrieve and display the top 4 most similar movies (excluding the original movie):

- Iraivi
- Walkaway
- Bhobhar - The Live Ash
- Thenkasipattanam

In [81]:
```
movie_name = 'Parched'
index=df[df.name==movie_name].index[0]
test_vector=vectors[index]
score,indexes=model.kneighbors([test_vector],n_neighbors=5)

for n in df.iloc[indexes[0][1:]].name.values:
    print(n)
```

```
Iraivi
Walkaway
Bhobhar - The Live Ash
Thenkasipattanam
```

Fetching Movie Poster from OMDb API

Using the movie's `movie_id` (`tt5286444`), a request is sent to the OMDb API to retrieve additional movie information. The `Poster` field from the JSON response provides the URL of the movie's poster image, which can be used for display in the recommendation system.

In [82]:
```
movie_id ='tt5286444'
url= f'http://www.omdbapi.com/?i={movie_id}&apikey=1a71d6fd'

import requests
resp = requests.get(url)
resp.json()['Poster']
```

Out[82]: 'https://m.media-amazon.com/images/M/MV5BYjg1OGMyZjktYzBhYS00NTkwLWE3NTUtZGEwZjc5Njc1NDk1XkEyXkFqcGc@._V1_SX300.jpg'

Saving Processed Data and Model Using Joblib

The cleaned dataframe, TF-IDF vectors, and the trained Nearest Neighbors model are saved using `joblib`. This allows the recommendation system to load the preprocessed data and model later without recomputing, improving efficiency:

- `df.pkl` → Processed movie dataframe
- `vectors.pkl` → TF-IDF vectors of movie details
- `model.pkl` → Trained Nearest Neighbors model

In [83]:
```
import joblib
joblib.dump(df,"df.pkl",compress=3)
joblib.dump(vectors,"vectors. pkl",compress=3)
joblib.dump(model,"model.pkl",compress=3)
```

Out[83]: ['model.pkl']

In [ ]: