In [7]:

```
import numpy as np
import pandas as pd
import ast
import pickle
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.metrics.pairwise import cosine_similarity

dataset = pd.read_csv(r"movies_metadata.csv", encoding='ISO-8859-1', on_bad_line
movies = dataset[['belongs_to_collection', 'original_title', 'tagline', 'overvie

def show(text):
    mov = []
    if pd.isna(text) or text == '':
        return None
    try:
        parsed_data = ast.literal_eval(text)
        if isinstance(parsed_data, dict):
            mov.append(parsed_data.get('name', None))
        elif isinstance(parsed_data, list):
            for i in parsed_data:
                if isinstance(i, dict):
                    mov.append(i.get('name', None))
    except (ValueError, SyntaxError, TypeError):
        return None
    return mov if mov else None

movies.dropna(inplace=True)
movies['belongs_to_collection'] = movies['belongs_to_collection'].apply(show)
movies['genres'] = movies['genres'].apply(show)

def space(text):
    rem = []
    if text is None:
        return None
    for i in text:
        rem.append(i.replace(" ", ""))
    return rem

movies['belongs_to_collection'] = movies['belongs_to_collection'].apply(space)
movies['genres'] = movies['genres'].apply(space)

def safe_join(text):
    return ' '.join([i for i in text if i]) if text else ''

movies['tags'] = (movies['belongs_to_collection'].apply(safe_join) + ' ' +
                  movies['genres'].apply(safe_join) + ' ' +
                  movies['original_title'].fillna('') + ' ' +
                  movies['tagline'].fillna('') + ' ' +
                  movies['overview'].fillna(''))

new = movies.drop(columns=['overview', 'genres', 'tagline', 'belongs_to_collecti
new['tags'] = new['tags'].apply(lambda x: " ".join(x.split()))
new['tags'] = new['tags'].apply(lambda x: x.lower())

cv = CountVectorizer(max_features=5000, stop_words='english')
vector = cv.fit_transform(new['tags']).toarray()

similarity = cosine_similarity(vector)
```

```
pickle.dump(new, open('movie_list.pkl', 'wb'))
pickle.dump(similarity, open('similarity.pkl', 'wb'))
def recommend(movie):
    movie = movie.lower()
    matched_movies = new[new['tags'].str.lower().str.contains(movie)]

    if matched_movies.empty:
        print(f"Movie '{movie}' not found in the dataset.")
        return

    index = matched_movies.index[0]

    distances = sorted(list(enumerate(similarity[index])), reverse=True, key=lam

    print(f"Movies similar to '{movie}':")
    for i in distances[1:6]:
        print(new.iloc[i[0]].original_title)

recommend('father')
new.head()
```

```
C:\Users\Dell\AppData\Local\Temp\ipykernel_15008\1873446660.py:8: DtypeWarning: C
olumns (16,23,25,26) have mixed types. Specify dtype option on import or set low_
memory=False.
  dataset = pd.read_csv(r"movies_metadata.csv", encoding='ISO-8859-1', on_bad_lin
es='skip')
C:\Users\Dell\AppData\Local\Temp\ipykernel_15008\1873446660.py:27: SettingWithCop
yWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stabl
e/user_guide/indexing.html#returning-a-view-versus-a-copy
  movies.dropna(inplace=True)
C:\Users\Dell\AppData\Local\Temp\ipykernel_15008\1873446660.py:28: SettingWithCop
yWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stabl
e/user_guide/indexing.html#returning-a-view-versus-a-copy
  movies['belongs_to_collection'] = movies['belongs_to_collection'].apply(show)
C:\Users\Dell\AppData\Local\Temp\ipykernel_15008\1873446660.py:29: SettingWithCop
yWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stabl
e/user_guide/indexing.html#returning-a-view-versus-a-copy
  movies['genres'] = movies['genres'].apply(show)
C:\Users\Dell\AppData\Local\Temp\ipykernel_15008\1873446660.py:39: SettingWithCop
yWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stabl
e/user_guide/indexing.html#returning-a-view-versus-a-copy
  movies['belongs_to_collection'] = movies['belongs_to_collection'].apply(space)
C:\Users\Dell\AppData\Local\Temp\ipykernel_15008\1873446660.py:40: SettingWithCop
yWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stabl
e/user_guide/indexing.html#returning-a-view-versus-a-copy
  movies['genres'] = movies['genres'].apply(space)
C:\Users\Dell\AppData\Local\Temp\ipykernel_15008\1873446660.py:45: SettingWithCop
yWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stabl
e/user_guide/indexing.html#returning-a-view-versus-a-copy
  movies['tags'] = (movies['belongs_to_collection'].apply(safe_join) + ' ' +
```

```
Movies similar to 'father':
The Exterminator
Ace Ventura: Pet Detective
Utvandrarna
Darkman
How to Train Your Dragon 2
```

Out[7]:

	original_title	id	tags
2	Grumpier Old Men	15602	grumpyoldmencollection romance comedy grumpier...
4	Father of the Bride Part II	11862	fatherofthebridecollection comedy father of th...
9	GoldenEye	710	jamesbondcollection adventure action thriller ...
12	Balto	21032	baltocollection family animation adventure bal...
18	Ace Ventura: When Nature Calls	9273	aceventuracollection crime comedy adventure ac...

In [ ]:

In [ ]:

In [ ]: