Movie Recommender Program - Cosine Similarity

Import Libraries and Data

In [3]:

```
# Import libraries
import pandas as pd
from sklearn.metrics.pairwise import cosine_similarity
```

In [4]:

```
# Import data set
movies_df = pd.read_csv('movies.csv')
```

Prepare Data Set

In [6]:

```
# Remove Year from title column
movies_df['Title'] = movies_df['title'].str[:-6]

# Remove white spaces
movies_df['Title'] = movies_df['Title'].str.strip()
```

In [7]:

```
# Obtain year from title column and create new column
movies_df['Year'] = movies_df['title'].str[-6:]

# Drop parentheses
movies_df['Year'] = movies_df['Year'].str.replace(r'[()]','', regex=True)
```

In [8]:

```
# Drop original 'title' column
movies_df.drop('title', axis=1, inplace=True)
```

In [9]:

```
# Ensure 'genres' is formatted as a string
movies_df['genres'] = movies_df['genres'].astype(str)

# Obtain dummy variables from 'genres' column
movies_df = movies_df.join(movies_df['genres'].str.get_dummies(sep='|'))
```

In [10]:

```
# Drop 'genres'column
movies_df.drop('genres', axis=1, inplace=True)
```

Create Recommender Program

In [12]:

```
# Extract genre columns
genre_columns = movies_df.columns[4:]

# Create matrix
genre_matrix = movies_df[genre_columns]
```

In [13]:

```
# Compute cosine similarity
similarity_matrix = cosine_similarity(genre_matrix)
```

In [14]:

```
# Create recommender function
def recommend_movies(movie_title, movies_df, similarity_matrix):
    # Find the index of the movie
    movie_index = movies_df[movies_df['Title'].str.lower() == movie_title.lower()].index

    # Add message if movie is not found
    if len(movie_index) == 0:
        return "Movie not found. Please check the title and try again."

    # Extract movie index (row number)
    movie_index = movie_index[0]

    # Get similarity scores
    similarity_scores = list(enumerate(similarity_matrix[movie_index]))

    # Sort movies by similarity score, exclude title that was typed by user
    similar_movies = sorted(similarity_scores, key=lambda x: x[1], reverse=True)[1:11]

    # Retrieve movie titles
    recommended_titles = [movies_df.iloc[i[0]]['Title'] for i in similar_movies]

    return recommended_titles
```

Test Recommender Program

In [16]:

```
# Ask for user input
print("""
Are you looking for something to watch? Type a movie you
like and we'll suggest 10 similar movies
""")
user_input = input("Name of favorite movie: ")

# Run recommendation function
movies = recommend_movies(user_input, movies_df, similarity_matrix)

# Handle wrong input and print top 10 movies if successful
if isinstance(movies, str):
    print(movies)
else:
    print(f"Top {len(movies)} similar movies to {user_input} are:")
    for id, movie in enumerate(movies, start=1):
        print(f"{id}. {movie}")
```

```
Are you looking for something to watch? Type a movie you
like and we'll suggest 10 similar movies

Top 10 similar movies to Avatar are:
1. Spider-Man 2
2. Superman Returns
3. Star Trek
4. Transformers: Revenge of the Fallen
5. Avatar
6. Tron: Legacy
7. Avengers, The
8. John Carter
9. Amazing Spider-Man, The
10. Oblivion
```