A Movie Recommendation system

In [1]:
```
# Import relevant module
import pandas as pd #Data processing
from sklearn.feature_extraction.text import TfidfVectorizer #Creates vectors
from sklearn.metrics.pairwise import cosine_similarity #Measures similarity in text analysis
```

In [2]:
```
# Import the csv file
df = pd.read_csv('C:/Users/Sanayak/Desktop/TMDb_updated.CSV')
df.head() #Returns first 5 entries of the Dataset
```

Out[2]:

	Unnamed: 0	title	overview	original_language	vote_count	vote_average
0	0	Ad Astra	The near future, a time when both hope and har...	en	2853	5.9
1	1	Bloodshot	After he and his wife are murdered, marine Ray...	en	1349	7.2
2	2	Bad Boys for Life	Marcus and Mike are forced to confront new thr...	en	2530	7.1
3	3	Ant-Man	Armed with the astonishing ability to shrink i...	en	13611	7.1
4	4	Percy Jackson: Sea of Monsters	In their quest to confront the ultimate evil, ...	en	3542	5.9

In [3]:
```
# Create a new Dataframe with relevant information
movies_df = df[['title', 'overview']]
print(movies_df.sample(5)) #Prints 5 randaom entries from the dataset
```

```
                   title                                           overview
1987      Cool Hand Luke  When petty criminal Luke Jackson is sentenced ...
4151  The Limehouse Golem  A series of murders has shaken the community t...
6644      Manner Teacher  Deputy Hyeon-moo from Isang Plans doesn't know...
6323    The Song of Names  A man searching for his childhood best friend ...
8423       Madly in Love  Cristina is a rich, spoiled princess visiting ...
```

In [4]:
```
movies_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10000 entries, 0 to 9999
Data columns (total 2 columns):
 #   Column    Non-Null Count  Dtype 
---  ------    --------------  ----- 
 0   title     10000 non-null  object
 1   overview  9970 non-null   object
dtypes: object(2)
memory usage: 156.4+ KB
```

In [5]:

```
# Get sum of null entries
movies_df['overview'].isna().sum()
```

Out[5]:

```
30
```

In [6]:

```
# Handle missing values
movies_df['overview'] = movies_df['overview'].fillna('')
movies_df['overview'].isna().sum()
```

```
C:\Users\Sanayak\AppData\Local\Temp\ipykernel_3312\134415056.py:2: SettingWithCopyWarning: 
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/stable/user_guide/indexing.html#returning
-a-view-versus-a-copy
  movies_df['overview'] = movies_df['overview'].fillna('')
```

Out[6]:

```
0
```

In [7]:

```
#Instantiate the TfidfVectorizer
tfid_vectorizer = TfidfVectorizer(stop_words="english")
X = tfid_vectorizer.fit_transform(movies_df['overview'])
```

In [8]:

```
print(X.shape)
```

```
(10000, 28709)
```

In [9]:

```
# Calculate the cosine similarity matrix
cosine_sim = cosine_similarity(X,X)
indices = pd.Series(movies_df.index, index=movies_df['title'])
```

In [10]:

```
# Define a funtion 'recommend_movies'
def recommend_movies(movie_title, cosine_sim=cosine_sim):
    # Get the index of the given movie
    idx = indices[movie_title]

    # Get pairwise similarity scores for this movie with all others
    sim_scores = list(enumerate(cosine_sim[idx]))

    # Sort the movies by similarity scores in descending order
    sim_scores = sorted(sim_scores, key=lambda x: x[1], reverse=True)

    # Get the top 10 similar movies (excluding the input movie)
    sim_scores = sim_scores[1:11]

    # Get the movie indices for the top 10 similar movies
    movie_indices = [i[0] for i in sim_scores]

    # Return the top 10 similar movies
    return movies_df['title'].iloc[movie_indices]
```

In [11]:

```
movie_to_recommend = "Bloodshot"
recommended_movies = recommend_movies(movie_to_recommend)
print(f"Movies similar to \"{movie_to_recommend}\":")
print(recommended_movies)
```

```
Movies similar to "Bloodshot":
830                Fractured
1677                     JFK
6969            Frozen River
1357    Escape Plan 2: Hades
3861           Looking Glass
3737            Uptown Girls
5364       Sweet and Lowdown
6708            The Thin Man
399                  Memento
6795                Shopgirl
Name: title, dtype: object
```

In [12]:

```
movie_to_recommend = "Diary of a Wimpy Kid"
recommended_movies = recommend_movies(movie_to_recommend)
```

```python
print(f"Movies similar to \"{movie_to_recommend}\":")
print(recommended_movies)
```

```
Movies similar to "Diary of a Wimpy Kid":
3781    Diary of a Wimpy Kid: Rodrick Rules
7077         Diary of a Wimpy Kid: Dog Days
7262               Welcome to the Dollhouse
5701          All The Freckles In The World
3638    Diary of a Wimpy Kid: The Long Haul
9039                           Mrs. Miniver
5247                            Brotherhood
3441                        A Grand Day Out
5560                         Little Fockers
7434                           The Outcasts
Name: title, dtype: object
```

In [13]:

```python
movie_to_recommend = "Ant-Man"
recommended_movies = recommend_movies(movie_to_recommend)
print(f"Movies similar to \"{movie_to_recommend}\":")
print(recommended_movies)
```

```
Movies similar to "Ant-Man":
175             Ant-Man and the Wasp
1012                       Rocky III
3929          It Could Happen to You
3695                The Ghost Writer
3509                        Triple 9
2315                Thick as Thieves
2720              Lying and Stealing
4704                         Armored
6233                  Swiss Army Man
9961    The Incredible Shrinking Man
Name: title, dtype: object
```

PREPARED AND DEVELOPED BY: VICTOR INIOBONG ITINAH