

```
In [28]: import numpy as np
import pandas as pd
from sklearn.feature_extraction import text
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.metrics.pairwise import cosine_similarity
```

```
In [29]: data = pd.read_csv("C:\\Users\\91805\\Downloads\\archive (7).zip")  
print(data.head())
```

	Show Id	Title \
0	cc1b6ed9-cf9e-4057-8303-34577fb54477	(Un)Well
1	e2ef4e91-fb25-42ab-b485-be8e3b23dedb	#Alive
2	b01b73b7-81f6-47a7-86d8-acb63080d525	#AnneFrank - Parallel Stories
3	b6611af0-f53c-4a08-9ffa-9716dc57eb9c	#blackAF
4	7f2d4170-bab8-4d75-adc2-197f7124c070	#cats_the_mewvie

	Description \
0	This docuseries takes a deep dive into the luc...
1	As a grisly virus rampages a city, a lone man ...
2	Through her diary, Anne Frank's story is retol...
3	Kenya Barris and his family navigate relations...
4	This pawesome documentary explores how our fel...

	Director \
0	NaN
1	Cho Il
2	Sabina Fedeli, Anna Migotto
3	NaN
4	Michael Margolis

	Genres \
0	Reality TV
1	Horror Movies, International Movies, Thrillers
2	Documentaries, International Movies
3	TV Comedies
4	Documentaries, International Movies

	Cast	Production Country \
0	NaN	United States
1	Yoo Ah-in, Park Shin-hye	South Korea
2	Helen Mirren, Gengher Gatti	Italy
3	Kenya Barris, Rashida Jones, Iman Benson, Genn...	United States
4	NaN	Canada

	Release Date	Rating	Duration	Imdb Score	Content Type	Date Added
0	2020.0	TV-MA	1 Season	6.6/10	TV Show	NaN
1	2020.0	TV-MA	99 min	6.2/10	Movie	September 8, 2020
2	2019.0	TV-14	95 min	6.4/10	Movie	July 1, 2020
3	2020.0	TV-MA	1 Season	6.6/10	TV Show	NaN
4	2020.0	TV-14	90 min	5.1/10	Movie	February 5, 2020

```
In [30]: print(data.isnull().sum())
```

```
Show Id          0
Title            0
Description       0
Director        2064
Genres           0
Cast            530
Production Country 559
Release Date      3
Rating           4
Duration         3
Imdb Score       608
Content Type      0
Date Added      1335
dtype: int64
```

```
In [31]: data = data[["Title", "Description", "Content Type", "Genres"]]
print(data.head())
```

```
      Title \
0      (Un)Well
1      #Alive
2  #AnneFrank - Parallel Stories
3      #blackAF
4  #cats_the_mewvie

      Description Content Type \
0  This docuseries takes a deep dive into the luc...  TV Show
1  As a grisly virus rampages a city, a lone man ...   Movie
2  Through her diary, Anne Frank's story is retol...   Movie
3  Kenya Barris and his family navigate relations...  TV Show
4  This pawesome documentary explores how our fel...   Movie

      Genres
0      Reality TV
1  Horror Movies, International Movies, Thrillers
2      Documentaries, International Movies
3      TV Comedies
4      Documentaries, International Movies
```

```
In [32]: data=data.dropna()
```

```
In [33]: import nltk
import re
```

```
In [34]: nltk.download('stopwords')
stemmer=nltk.SnowballStemmer('english')
from nltk.corpus import stopwords
import string
stopword=set(stopwords.words('english'))
```

```
[nltk_data] Downloading package stopwords to
[nltk_data] C:\Users\91805\AppData\Roaming\nltk_data...
[nltk_data] Package stopwords is already up-to-date!
```

```
In [35]: def clean(text):
    text=str(text).lower()
    text=re.sub('\[.*?\]', '', text)
    text = re.sub('https?://\S+|www\.\S+', '', text)
    text = re.sub('<.*?>+', '', text)
    text = re.sub('[%s]' % re.escape(string.punctuation), '', text)
    text = re.sub('\n', '', text)
    text = re.sub('\w*\d\w*', '', text)
    text = [word for word in text.split(' ') if word not in stopword]
    text=" ".join(text)
    text = [stemmer.stem(word) for word in text.split(' ')]
    text=" ".join(text)
    return text
```

```
In [36]: data['Title']=data['Title'].apply(clean)
```

```
In [37]: print(data.Title.sample(10))
```

```
5942                zero chill
4622            angri bird movi
3346                nang nak
3329            naa bangaaru talli
4729            confess killer
3882    rebellcomedi straight outta zoo
521            baggio divin ponytail
5077            next karat kid
1273                derek
4621                angel
Name: Title, dtype: object
```

```
In [40]: from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.metrics.pairwise import cosine_similarity

# Assuming 'data' is your DataFrame and 'Genres' is the column containing genre information
features = data['Genres'].tolist()

tfidf = TfidfVectorizer(stop_words='english')
tfidf_matrix = tfidf.fit_transform(features)
similarity = cosine_similarity(tfidf_matrix)
```

```
In [42]: indices=pd.Series(data.index,
                        index=data['Title']).drop_duplicates()
```

```
In [43]: def netFlix_recommendation(title,similarity=similarity):
    index=indices[title]
    similarity_scores=list(enumerate(similarity[index]))
    similarity_scores=sorted(similarity_scores,key=lambda x: x[1],reverse=True)
    similarity_scores=similarity_scores[0:10]
    movieindices=[i[0] for i in similarity_scores]
    return data['Title'].iloc[movieindices]
```

```
In [44]: print(netFlix_recommendation('girlfriend'))
```

```
3          blackaf  
285        washington  
417      arrest develop  
434    astronomi club sketch show  
451    aunti donna big ol hous fun  
656          big mouth  
752        bojack horseman  
805          brew brother  
935          champion  
937        chappell show  
Name: Title, dtype: object
```

```
In [ ]:
```