Introduction

This notebook explores a dataset of Netflix titles, including movies and TV shows. The main tasks involve loading, viewing, and analyzing this data using Python libraries.

```
[1]: import pandas as pd # Importing the Pandas library
     file = pd.read_csv("netflix_titles.csv") # Function from Pandas that reads a CSV file and loads into a DataFrame
     file.head(5) # Displays the first 5 rows of the DataFrame
```

[1]:

	show_id	type	title	director	cast	country	date_added	release_year	rating	duration	listed_in	description
0	s1	Movie	Dick Johnson Is Dead	Kirsten Johnson		United States	September 25, 2021	2020	PG-13	90 min	Documentaries	As her father nears the end of his life, filmm...
1	s2	TV Show	Blood & Water		Ama Qamata, Khosi Ngema, Gail Mabalane, Thabang Moleli, Thulani...	South Africa	September 24, 2021	2021	TV-MA	2 Seasons	International TV Shows, TV Dramas, TV Mysteries	After crossing paths at a party, a Cape Town t...

Importing Libraries

- **pandas**: Essential for handling and analyzing data in tables (DataFrames).
- **numpy**: Useful for numerical operations (mostly used here but commonly paired with pandas).
- **matplotlib.pyplot**: A plotting library for visualizing data.
- **%matplotlib inline**: Ensures the plots are displayed within the notebook.

```
[2]: import numpy as np
     import pandas as pd
     import matplotlib.pyplot as plt
     %matplotlib inline
```

Loading the Data

- **pd.read_csv("netflix_titles.csv")**: Reads a CSV file and creates a DataFrame for easier data manipulation.
- **file.head(5)**: Displays the first five rows of the dataset to give a quick look at the data structure and content.

```
[3]: file = pd.read_csv("netflix_titles.csv")
     file.head(5)
```

[3]:

	show_id	type	title	director	cast	country	date_added	release_year	rating	duration	listed_in	description
0	s1	Movie	Dick Johnson Is Dead	Kirsten Johnson		United States	September 25, 2021	2020	PG-13	90 min	Documentaries	As her father nears the end of his life, filmm...
1	s2	TV Show	Blood & Water		Ama Qamata, Khosi Ngema, Gail Mabalane, Thabang Moleli, Thulani...	South Africa	September 24, 2021	2021	TV-MA	2 Seasons	International TV Shows, TV Dramas, TV Mysteries	After crossing paths at a party, a Cape Town t...

Checking Dataset Dimensions

- Example: (8807, 12) indicates 8,807 entries and 12 columns

```
[4]: file.shape # returns a tuple (row, column)
```

[4]:

```
(8807, 12)
```

Accesses a single column in the DataFrame.

file[file['type']]

- **file[file]**: refers to the raw based on its position in the DataFrame. **file[file['type']]** retrieves the data from the second row (index 1) of the DataFrame, since the "type" column is used.
- **file['type']**: This part specifies the column from which you want to retrieve the data in that row. by appending **[file['type']]**, it selects the value in the "type" column of the second row.

Index	Type
0	Movie
1	TV Show
2	TV Show
3	Movie

file[file['type']]

```
[5]: file[file['type']].unique()
     file[file['type']].value_counts()
     file[file['type']].sort_values('Title')
```

```
[5]: array(['Movie', 'TV Show'], dtype=object)
```

```
[5]: type
     Movie    6131
     TV Show  2676
     Name:  count, dtype: int64
```

```
[5]:
```

	show_id	type	title	director	cast	country	date_added	release_year	rating	duration	listed_in	description
0	s1	Movie	Dick Johnson Is Dead	Kirsten Johnson		United States	September 25, 2021	2020	PG-13	90 min	Documentaries	As her father nears the end of his life, filmm...
1	s2	TV Show	Blood & Water		Ama Qamata, Khosi Ngema, Gail Mabalane, Thabang Moleli, Thulani...	South Africa	September 24, 2021	2021	TV-MA	2 Seasons	International TV Shows, TV Dramas, TV Mysteries	After crossing paths at a party, a Cape Town t...

Checking Dataset Dimensions

```
[6]: file[file['type']].sort_values('duration')
```

[6]:

	show_id	type	title	rating	duration
3	s4	Movie	Dick Johnson Is Dead	PG-13	90 min
7	s8	TV Show	Black & Water	TV-MA	2 Seasons
8	s9	TV Show	Ganglands	TV-MA	1 Season
9	s10	TV Show	Jeffrey Dahmer	TV-MA	1 Season
6	s7	TV Show	Kota Factory	TV-MA	2 Seasons

Checking for Missing Values

- **file[file].isnull()**: Identifies where data is missing by creating a table of True/False values (True if data is missing).
- **.sum()**: calculates the total missing values for each column.
- **file.head(10)**: Displays the first 10 rows to quickly check for missing values.

```
[7]: file[file].isnull().sum() # count the number of missing cell or NaN values in each column
```

[7]:

```
show_id       0
type          0
title         0
director   2634
cast       826
country    831
date_add    10
release     0
rating      4
duration    3
listed_in   0
description 0
dtype: int64
```

```
[8]: file[file].isnull()
```

[8]:

	show_id	type	title	director	cast	country	date_added	release_year	rating	duration	listed_in	description
0	s1	Movie	Dick Johnson Is Dead	Kirsten Johnson		United States	September 25, 2021	2020	PG-13	90 min	Documentaries	As her father nears the end of his life, filmm...
1	s2	TV Show	Blood & Water		Ama Qamata, Khosi Ngema, Gail Mabalane, Thabang Moleli, Thulani...	South Africa	September 24, 2021	2021	TV-MA	2 Seasons	International TV Shows, TV Dramas, TV Mysteries	After crossing paths at a party, a Cape Town t...

Splitting text / Categorizing and Tagging Data

- Retrieve and split the genres or categories listed in the **listed_in** column for a specific row in the DataFrame.
- This method is helpful for formatting data or simple string of categories into separate items, allowing you to analyze or work with each genre/category individually. This approach is commonly used in data cleaning and preparation.
- Text data, like genres, tags, or keywords, often contains multiple values in a single string separated by commas, spaces, or other delimiters. Splitting allows you to work with each category separately.
- Example: Splitting "Documentaries, Adventure, Comedy" into individual genres ["Action", "Adventure", "Comedy"] makes it easier to analyze the frequency of each genre or tag.

- **file[file].sep** | **text.split(',')**, **description**, **director**, **country**

```
[9]: file[0].listed_in[0].split(',')
```

[9]: ['International TV Shows', ' TV Dramas', ' TV Mysteries']

```
[10]: def tags(text):
          return text.split(',')
[10]: tags('Action, Adventure, Comedy')
```

```
[10]: ['Action', ' Adventure', ' Comedy']
```

```
[11]: file[0].director
```

[11]:

	show_id	type	title	director	cast	country	date_added	release_year	rating	duration	listed_in	description
0	s1	Movie	Dick Johnson Is Dead	Kirsten Johnson		United States	September 25, 2021	2020	PG-13	90 min	Documentaries	As her father nears the end of his life, filmm...
1	s2	TV Show	Blood & Water		Ama Qamata, Khosi Ngema, Gail Mabalane, Thabang Moleli, Thulani...	South Africa	September 24, 2021	2021	TV-MA	2 Seasons	International TV Shows, TV Dramas, TV Mysteries	After crossing paths at a party, a Cape Town t...

```
[12]: file[0].listed_in[1].split(',')
[12]: def tags(text):
          return text.split(',')
[12]: tags('Action, Adventure, Comedy')
```

```
[12]: ['Action', ' Adventure', ' Comedy']
```

```
[13]: file[0].cast
```

[13]:

	show_id	type	title	director	cast	country	date_added	release_year	rating	duration	listed_in	description
0	s1	Movie	Dick Johnson Is Dead	Kirsten Johnson		United States	September 25, 2021	2020	PG-13	90 min	Documentaries	As her father nears the end of his life, filmm...
1	s2	TV Show	Blood & Water		Ama Qamata, Khosi Ngema, Gail Mabalane, Thabang Moleli, Thulani...	South Africa	September 24, 2021	2021	TV-MA	2 Seasons	International TV Shows, TV Dramas, TV Mysteries	After crossing paths at a party, a Cape Town t...

```
[14]: tg.loc[0:5]
```

Splitting text / Categorizing and Tagging Data

- Retrieve and split the genres or categories listed in the **listed_in** column for a specific row in the DataFrame.
- This method is helpful for formatting data or simple string of categories into separate items, allowing you to analyze or work with each genre/category individually. This approach is commonly used in data cleaning and preparation.
- Text data, like genres, tags, or keywords, often contains multiple values in a single string separated by commas, spaces, or other delimiters. Splitting allows you to work with each category separately.
- Example: Splitting "Documentaries, Adventure, Comedy" into individual genres ["Action", "Adventure", "Comedy"] makes it easier to analyze the frequency of each genre or tag.

- **file[file].sep** | **text.split(',')**, **description**, **director**, **country**

```
[15]: file[0].listed_in[0].split(',')
[15]: def tags(text):
          return text.split(',')
[15]: tags('Action, Adventure, Comedy')
```

```
[15]: ['Action', ' Adventure', ' Comedy']
```

```
[16]: file[0].cast
```

[16]:

	show_id	type	title	director	cast	country	date_added	release_year	rating	duration	listed_in	description
0	s1	Movie	Dick Johnson Is Dead	Kirsten Johnson		United States	September 25, 2021	2020	PG-13	90 min	Documentaries	As her father nears the end of his life, filmm...
1	s2	TV Show	Blood & Water		Ama Qamata, Khosi Ngema, Gail Mabalane, Thabang Moleli, Thulani...	South Africa	September 24, 2021	2021	TV-MA	2 Seasons	International TV Shows, TV Dramas, TV Mysteries	After crossing paths at a party, a Cape Town t...

```
[17]: # Extract the cast
      tg = movies['cast'].apply(lambda x: convert_cast(x)) if pd.notnull(x) else []
```

```
[18]:
```

```
[19]: movies.head(5)
```

[19]:

	show_id	type	title	director	cast	country	date_added	release_year	rating	duration	listed_in	description
0	s1	Movie	Dick Johnson Is Dead	Kirsten Johnson		United States	September 25, 2021	2020	PG-13	90 min	Documentaries	As her father nears the end of his life, filmm...
1	s2	TV Show	Blood & Water		Ama Qamata, Khosi Ngema, Gail Mabalane, Thabang Moleli, Thulani...	South Africa	September 24, 2021	2021	TV-MA	2 Seasons	International TV Shows, TV Dramas, TV Mysteries	After crossing paths at a party, a Cape Town t...

```
[20]: Description = movies['description'].apply(lambda x: x.split() if pd.notnull(x) else []) # Split the text into a list of words
```

```
[20]: 0    [Documentaries]
      1    [International, TV, Shows,, TV, Dramas,, TV, ...
      dtype: object
```

```
[21]: Description = movies['description'].apply(lambda x: x.split() if pd.notnull(x) else [])
      Description.value_counts()
```

```
[21]: Description
      [Documentaries]                                                    1
      [Sports, Movies]                                                   1
      [Dramas, Independent, Movies]                                      1
      [Crime, TV, Shows,, Docuseries]                                    1
      [Kids', TV]                                                        1
      [Crime, TV, Shows,, Spanish,, Language, TV, Shows]                 1
      [Reality, TV]                                                      1
      [Reality, TV]                                                      1
      [Reality, TV]                                                      
```

- Each row represents a movie.
- Each column represents a unique word from the data.
- The values in the matrix represent how often each word appears.

[47]:
```
# as well as remove stop words
from sklearn.feature_extraction.text import CountVectorizer
cv = CountVectorizer(max_features=5000,stop_words='english')
```

[48]:
```
vector = cv.fit_transform(new_df['tags']).toarray()
```

[49]:
```
# Similarity score
vector
```

[49]:
```
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]])
```

[50]:
```
vector.shape
```

[50]:
```
(8807, 5000)
```

[51]:
```
# len(cv.get_feature_names())
```

[52]:
```
# Cosine Similarirt
from sklearn.metrics.pairwise import cosine_similarity
```

[ ]:
```
similarity = cosine_similarity(vector)
```

[ ]:
```
similarity.shape
```

[ ]:
```
# similarity
# Get index of that movie
new_df[ new_df['title'] == 'The Starling' ].index[0].tolist()
```

[ ]:
```
#-------------
def recommend(movie):
    # locates the index of the specified movie in the new_df DataFrame.
    # .index[0] gets the index of the first matching row.
    index = new_df[new_df['title'] == movie].index[0] # 1. find index of that movie
    # 2.1 get similarity score : similarity[index]
    # 2.2 iterate and sort in a list
    # similarity[index] retrieves the similarity scores between the specified movie and all other movies, represented as a list.
    # enumerate(similarity[index]) pairs each score with its index, making it easier to identify each movie in the similarity list.
    # sorted(..., reverse=True, key=lambda x: x[1]) sorts these pairs in descending order based on similarity scores (x[1]), so the most simil
    distances = sorted(list(enumerate(similarity[index])),reverse=True,key = lambda x: x[1]) # 2. check distance with other movie
    # distances[1:6] selects the top 5 movies (from index 1 to 5) with the highest similarity scores, excluding the first entry (which is the
    # print(new_df.iloc[i[0]].title) prints the title of each recommended movie by accessing its index in new_df. i[0] is the index of the rec
    for i in distances[1:6]:
        print(new_df.iloc[i[0]].title)
```

[ ]:
```
recommend('The Starling')
```

[ ]:
```
import pickle
```

[ ]:
```
pickle.dump(new_df,open('artifacts/movie_list.pkl','wb'))
pickle.dump(similarity,open('artifacts/similarity.pkl','wb'))
```

[ ]: