

# Netflix\_titles

June 14, 2025

## 0.0.1 Step 1: Dataset Load

```
[2]: # import libraries
import pandas as pd

df = pd.read_csv(r'C:\Users\scpl\OneDrive\Desktop\IT\Data_
↳Science\Datasets\netflix_titles.csv')
```

```
[3]: # View first 5 rows
print("First 5 rows of the dataset:")
print(df.head())
```

First 5 rows of the dataset:

|   | show_id | type    | title                 | director \      |
|---|---------|---------|-----------------------|-----------------|
| 0 | s1      | Movie   | Dick Johnson Is Dead  | Kirsten Johnson |
| 1 | s2      | TV Show | Blood & Water         | NaN             |
| 2 | s3      | TV Show | Ganglands             | Julien Leclercq |
| 3 | s4      | TV Show | Jailbirds New Orleans | NaN             |
| 4 | s5      | TV Show | Kota Factory          | NaN             |

|   | cast  | country \     |
|---|---|---------------|
| 0 | NaN   | United States |
| 1 | Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban... | South Africa  |
| 2 | Sami Bouajila, Tracy Gotoas, Samuel Jouy, Nabi... | NaN           |
| 3 | NaN   | NaN           |
| 4 | Mayur More, Jitendra Kumar, Ranjan Raj, Alam K... | India         |

|   | date_added         | release_year | rating | duration \ |
|---|--------------------|--------------|--------|------------|
| 0 | September 25, 2021 | 2020         | PG-13  | 90 min     |
| 1 | September 24, 2021 | 2021         | TV-MA  | 2 Seasons  |
| 2 | September 24, 2021 | 2021         | TV-MA  | 1 Season   |
| 3 | September 24, 2021 | 2021         | TV-MA  | 1 Season   |
| 4 | September 24, 2021 | 2021         | TV-MA  | 2 Seasons  |

|   | listed_in \                                       |
|---|---|
| 0 | Documentaries                                     |
| 1 | International TV Shows, TV Dramas, TV Mysteries   |
| 2 | Crime TV Shows, International TV Shows, TV Act... |
| 3 | Docuseries, Reality TV                            |

4 International TV Shows, Romantic TV Shows, TV ...

```
description
0 As her father nears the end of his life, filmm...
1 After crossing paths at a party, a Cape Town t...
2 To protect his family from a powerful drug lor...
3 Feuds, flirtations and toilet talk go down amo...
4 In a city of coaching centers known to train I...
```

## 0.0.2 Step 2: Dataset Summary (Shape, Columns, Info)

```
[35]: # Shape of dataset: (rows, columns)
print("Shape of dataset:")
print(df.shape)
```

Shape of dataset:  
(8807, 12)

```
[36]: # Column names
print("Column Names:")
print(df.columns.tolist())
```

Column Names:  
['show\_id', 'type', 'title', 'director', 'cast', 'country', 'date\_added',  
'release\_year', 'rating', 'duration', 'listed\_in', 'description']

```
[37]: # Data info (types + null values)
print("Dataset Info:")
print(df.info())
```

Dataset Info:  
<class 'pandas.core.frame.DataFrame'>  
RangeIndex: 8807 entries, 0 to 8806  
Data columns (total 12 columns):

| #  | Column       | Non-Null Count | Dtype  |
|----|--------------|----------------|--------|
| 0  | show_id      | 8807 non-null  | object |
| 1  | type         | 8807 non-null  | object |
| 2  | title        | 8807 non-null  | object |
| 3  | director     | 6173 non-null  | object |
| 4  | cast         | 7982 non-null  | object |
| 5  | country      | 7976 non-null  | object |
| 6  | date_added   | 8797 non-null  | object |
| 7  | release_year | 8807 non-null  | int64  |
| 8  | rating       | 8803 non-null  | object |
| 9  | duration     | 8804 non-null  | object |
| 10 | listed_in    | 8807 non-null  | object |
| 11 | description  | 8807 non-null  | object |

dtypes: int64(1), object(11)

memory usage: 825.8+ KB  
None

### 0.0.3 Step 3: Unique Values in 'Type' Column using List, Set, Loop

```
[4]: # Convert 'type' column to a list
type_list = df['type'].tolist()

unique_types = set(type_list)

print(" Total number of titles:", len(type_list))
print(" Unique content types in the dataset are:")

for content_type in unique_types:
    print(" -", content_type)
```

Total number of titles: 8807  
Unique content types in the dataset are:  
- TV Show  
- Movie

```
[43]: # Get unique types using set
unique_types = set(type_list)

print(" Unique content types in the dataset are:")
for content in unique_types:
    print(" -", content)

print(" Total unique types found:", len(unique_types))
```

Unique content types in the dataset are:  
- Movie  
- TV Show  
Total unique types found: 2

```
[40]: # Display unique content types
print("Unique content types:")
for content in unique_types:
    print(content)
```

Unique content types:  
Movie  
TV Show

#### 0.0.4 Step 4: Count Each Type (Movies/TV Shows) using Dictionary + Loop

```
[44]: # Count content types manually
type_count = {}

for t in type_list:
    if t in type_count:
        type_count[t] += 1
    else:
        type_count[t] = 1

# Show the result
print("Count of content types:")
print(type_count)
```

Count of content types:  
{'Movie': 6131, 'TV Show': 2676}

#### 0.0.5 Step 5: Function to Filter Content by Country

```
[45]: # Create a function to get content from a specific country
def get_content_by_country(country):
    result = df[df['country'] == country]
    return result[['title', 'type', 'release_year']]

# Example: Content from India
indian_content = get_content_by_country('India')
print("Top 5 Indian titles:")
print(indian_content.head())
```

Top 5 Indian titles:

|    | title                         | type    | release_year |
|----|-------------------------------|---------|--------------|
| 4  | Kota Factory                  | TV Show | 2021         |
| 24 | Jeans                         | Movie   | 1998         |
| 39 | Chhota Bheem                  | TV Show | 2021         |
| 50 | Dharmakshetra                 | TV Show | 2014         |
| 66 | Raja Rasoi Aur Anya Kahaniyan | TV Show | 2014         |

#### 0.0.6 Step 6: Find All Movies Released in a Particular Year

```
[46]: # Function to get movies by year
def get_movies_by_year(year):
    result = df[(df['release_year'] == year) & (df['type'] == 'Movie')]
    return result[['title', 'country']]

# Example usage
print("Movies released in 2020:")
```

```
print(get_movies_by_year(2020).head())
```

Movies released in 2020:

|     | title   | country       |
|-----|---|---------------|
| 0   | Dick Johnson Is Dead                              | United States |
| 16  | Europe's Most Dangerous Man: Otto Skorzeny in ... | NaN           |
| 78  | Tughlaq Durbar                                    | NaN           |
| 84  | Omo Ghetto: the Saga                              | Nigeria       |
| 103 | Shadow Parties                                    | NaN           |

### 0.0.7 Step 7: Sort Dataset by 'date\_added'

```
[49]: # Clean extra whitespace from 'date_added' column
df['date_added'] = df['date_added'].str.strip()

# Convert to datetime format safely
df['date_added'] = pd.to_datetime(df['date_added'], errors='coerce')

# Sort by latest added content
latest_added = df.sort_values(by='date_added', ascending=False)

# Show top 5 recently added titles
print("Recently added titles on Netflix:")
print(latest_added[['title', 'date_added']].head())
```

Recently added titles on Netflix:

|    | title                               | date_added |
|----|-------------------------------------|------------|
| 0  | Dick Johnson Is Dead                | 2021-09-25 |
| 6  | My Little Pony: A New Generation    | 2021-09-24 |
| 10 | Vendetta: Truth, Lies and The Mafia | 2021-09-24 |
| 9  | The Starling                        | 2021-09-24 |
| 8  | The Great British Baking Show       | 2021-09-24 |

### 0.0.8 Step 8: Most Common Countries – Top 5

```
[50]: # Top 5 countries by content count
top_countries = df['country'].value_counts().head(5)

print("Top 5 content-producing countries:")
print(top_countries)
```

Top 5 content-producing countries:

| country        |      |
|----------------|------|
| United States  | 2818 |
| India          | 972  |
| United Kingdom | 419  |
| Japan          | 245  |
| South Korea    | 199  |

Name: count, dtype: int64

### 0.0.9 Step 9: Create a Dictionary – Count Content Types

```
[51]: # Create a dictionary to count each content type
type_counts = {}

# Loop through the 'type' column and count occurrences
for content_type in df['type']:
    if content_type in type_counts:
        type_counts[content_type] += 1
    else:
        type_counts[content_type] = 1

# Print the result
print("Content Type Distribution:")
print(type_counts)
```

```
Content Type Distribution:
{'Movie': 6131, 'TV Show': 2676}
```

### 0.0.10 Step 10: Write a Function – Summarize Type Counts Nicely

```
[54]: # Define a function to display dictionary data in a clean way
def display_content_summary(count_dict):
    print("\n Content Summary Report:")
    total = sum(count_dict.values())
    for key, value in count_dict.items():
        percent = (value / total) * 100
        print(f"• {key}: {value} titles ({percent:.2f}%)")
    print(f"Total Titles: {total}")

# Call the function with your dictionary
display_content_summary(type_counts)
```

```
Content Summary Report:
• Movie: 6131 titles (69.62%)
• TV Show: 2676 titles (30.38%)
Total Titles: 8807
```

```
[ ]:
```

### 0.0.11 Conclusion :

This project demonstrates a strong foundation in Python and basic data analysis by applying core skills like variables, loops, lists, sets, dictionaries, and functions to a real-world Netflix dataset. I successfully extracted meaningful insights—such as content distribution between movies and TV shows, unique content types, and recent titles added. Through this hands-on work, I’ve strengthened my problem-solving abilities and gained confidence in writing clean, logical, and scalable code.

[ ]:

[ ]:

[ ]: