**Netflix Data: Cleaning, Analysis and Visualization**

# Project Overview

The main goal of this project is to explore and analyze the Netflix dataset by performing data cleaning, preprocessing, and visualization. This helps uncover insights about content distribution, trends over time, and user consumption patterns on the platform.

# Tools Used

Programming Languages : Python

Database: SQL

Spreadsheet Software: Excel

# About Dataset

# Netflix is a popular streaming service that offers a vast catalog of movies, TV shows, and original contents. This dataset is a cleaned version of the original version which can be found [here](https://www.kaggle.com/datasets/shivamb/netflix-shows). The data consist of contents added to Netflix from 2008 to 2021. The oldest content is as old as 1925 and the newest as 2021. This dataset will be cleaned with PostgreSQL and visualized with Tableau. The purpose of this dataset is to test my data cleaning and visualization skills. The cleaned data can be found below and the Tableau dashboard can be found [here](https://public.tableau.com/app/profile/abdulrasaq.ariyo/viz/NetflixTVShowsMovies_16615029026580/NetflixDashboard) .

# Data Cleaning

We are going to:

1. Treat the Nulls
2. Treat the duplicates
3. Populate missing rows
4. Drop unneeded columns
5. Split columns

Extra steps and more explanation on the process will be explained through the code comments

**Example: You can get the basic idea how you can create a project from here**

**Netflix Data: Cleaning, Analysis, and Visualization (Beginner ML Project)**

This project involves loading, cleaning, analyzing, and visualizing data from a Netflix dataset. We'll use Python libraries like Pandas, Matplotlib, and Seaborn to work through the project. The goal is to explore the dataset, derive insights, and prepare for potential machine learning tasks.

**Step 1: Import Required Libraries**

import pandas as pd import numpy as np import matplotlib.pyplot as plt import seaborn as sns from wordcloud import WordCloud

**Step 2: Load the Dataset**

Assume we have a dataset named netflix\_titles.csv.

# Load the dataset data = pd.read\_csv('netflix\_titles.csv')

# Display the first few rows of the dataset print(data.head())

**Step 3: Data Cleaning**

Identify and handle missing data, correct data types, and drop duplicates.

# Check for missing values print(data.isnull().sum())

# Drop duplicates if any data.drop\_duplicates(inplace=True)

# Drop rows with missing critical information data.dropna(subset=['director', 'cast', 'country'],

inplace=True)

# Convert 'date\_added' to datetime data['date\_added'] = pd.to\_datetime(data['date\_added']) # Show data types to confirm changes print(data.dtypes)

**Step 4: Exploratory Data Analysis (EDA)**

**1. Content Type Distribution (Movies vs. TV Shows)** # Count the number of Movies and TV Shows type\_counts = data['type'].value\_counts()

# Plot the distribution plt.figure(figsize=(8, 6)) sns.barplot(x=type\_counts.index, y=type\_counts.values, palette='Set2') plt.title('Distribution of Content by Type') plt.xlabel('Type') plt.ylabel('Count') plt.show()

1. **Most Common Genres**

# Split the 'listed\_in' column and count genres data['genres'] = data['listed\_in'].apply(lambda x: x.split(',

')) all\_genres = sum(data['genres'], []) genre\_counts = pd.Series(all\_genres).value\_counts().head(10)

# Plot the most common genres plt.figure(figsize=(10, 6)) sns.barplot(x=genre\_counts.values, y=genre\_counts.index, palette='Set3') plt.title('Most Common Genres on Netflix') plt.xlabel('Count') plt.ylabel('Genre') plt.show()

1. **Content Added Over Time**

# Extract year and month from 'date\_added' data['year\_added'] = data['date\_added'].dt.year data['month\_added'] = data['date\_added'].dt.month

# Plot content added over the years plt.figure(figsize=(12, 6)) sns.countplot(x='year\_added', data=data, palette='coolwarm') plt.title('Content Added Over Time') plt.xlabel('Year') plt.ylabel('Count') plt.xticks(rotation=45) plt.show()

1. **Top 10 Directors with the Most Titles**

# Count titles by director top\_directors = data['director'].value\_counts().head(10)

# Plot top directors plt.figure(figsize=(10, 6)) sns.barplot(x=top\_directors.values, y=top\_directors.index, palette='Blues\_d') plt.title('Top 10 Directors with the Most Titles') plt.xlabel('Number of Titles') plt.ylabel('Director') plt.show()

**5. Word Cloud of Movie Titles** # Generate word cloud movie\_titles = data[data['type'] == 'Movie']['title']

wordcloud = WordCloud(width=800, height=400,

background\_color='black').generate(' '.join(movie\_titles))

# Plot word cloud plt.figure(figsize=(10, 6)) plt.imshow(wordcloud, interpolation='bilinear') plt.axis('off')

plt.show()

**Step 5: Conclusion and Insights**

In this project, we:

1. **Cleaned the data** by handling missing values, removing duplicates, and converting data types.
2. **Explored the data** through various visualizations such as bar plots and word clouds.
3. **Analyzed content trends** over time, identified popular genres, and highlighted top directors.

**Step 6: Next Steps**

1. **Feature Engineering**: Create new features, such as counting the number of genres per movie or extracting the duration in minutes.
2. **Machine Learning**: Use the cleaned and processed data to build models for recommendations or trend predictions.
3. **Advanced Visualization**: Use interactive plots or dashboards for more detailed analysis.

This project is a foundational exercise that introduces essential data analysis techniques, paving the way for more advanced projects.

**Sample code:**

[1]:

**import**

**pandas**

**as**

**pd**

**import**

**numpy**

**as**

**np**

**import**

**matplotlib**

**.**

**pyplot**

**as**

**plt**

**import**

**seaborn**

**as**

**sns**

[17]:

data

=

pd

.

read\_csv(

"

netflix1.csv

"

)

data

.

head()

[17]:

show\_id

type

title

director \

|  |  |  |
| --- | --- | --- |
| 0 | s1 Movie | Dick Johnson Is Dead Kirsten Johnson |
| 1 | s3 TV Show | Ganglands Julien Leclercq |
| 2 | s6 TV Show | Midnight Mass Mike Flanagan |

1. s14 Movie Confessions of an Invisible Girl Bruno Garotti
2. s8 Movie Sankofa Haile Gerima

country date\_added release\_year rating duration \

|  |  |
| --- | --- |
| 0 United States 9/25/2021 | 2020 PG-13 90 min |
| 1 France 9/24/2021 | 2021 TV-MA 1 Season |
| 2 United States 9/24/2021 | 2021 TV-MA 1 Season |
| 3 Brazil 9/22/2021 | 2021 TV-PG 91 min |
| 4 United States 9/24/2021 | 1993 TV-MA 125 min  listed\_in |
| 0 | Documentaries |

1. Crime TV Shows, International TV Shows, TV Act…
2. TV Dramas, TV Horror, TV Mysteries
3. Children & Family Movies, Comedies
4. Dramas, Independent Movies, International Movies

[19]:

data

.

info()

<class 'pandas.core.frame.DataFrame'> RangeIndex: 8790 entries, 0 to 8789 Data columns (total 10 columns):

|  |  |  |
| --- | --- | --- |
| # Column | Non-Null Count Dtype | |
| --- ------ | -------------- ----- | |
| 0 show\_id | 8790 non-null object | |
| 1 type | 8790 non-null object | |
| 2 title 8790 non-null | | object |
| 3 director 8790 non-null | | object |
| 4 country 8790 non-null | | object |
| 5 date\_added 8790 non-null | | object |
| 6 release\_year 8790 non-null | | int64 |
| 7 rating 8790 non-null | | object |
| 8 duration 8790 non-null | | object |
| 9 listed\_in 8790 non-null  dtypes: int64(1), object(9) memory usage: 686.8+ KB | | object |

[21]:

data

.

shape

[21]: (8790, 10)

[23]:

data

=

data

.

drop\_duplicates()

[25]:

data[

'

type

'

]

.

value\_counts()

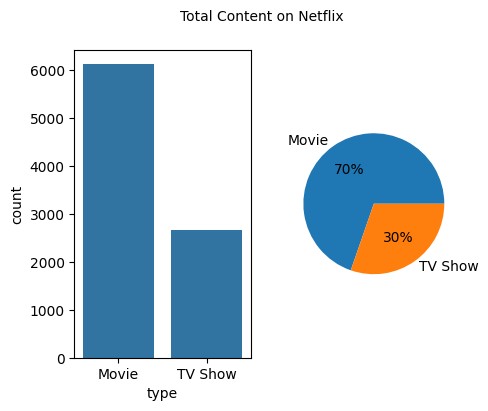
[25]: type

Movie 6126 TV Show 2664

Name: count, dtype: int64

[39]: freq=data['type'].value\_counts() fig, axes=plt.subplots(1,2, figsize=(5, 4)) sns.countplot(data, x=data['type'], ax=axes[0]) plt.pie(freq, labels=['Movie', 'TV Show'], autopct='**%.0f%%**') plt.suptitle('Total Content on Netflix', fontsize=10)

[39]: Text(0.5, 0.98, 'Total Content on Netflix')



[41]:

data

.

info()

<class 'pandas.core.frame.DataFrame'> RangeIndex: 8790 entries, 0 to 8789 Data columns (total 10 columns):

|  |  |
| --- | --- |
| # Column Non-Null Count Dtype  --- ------ -------------- ----- | |
| 0 show\_id 8790 non-null | object |
| 1 type 8790 non-null | object |
| 2 title 8790 non-null | object |
| 3 director 8790 non-null | object |
| 4 country 8790 non-null | object |
| 5 date\_added 8790 non-null | object |
| 6 release\_year 8790 non-null | int64 |
| 7 rating 8790 non-null | object |
| 8 duration 8790 non-null | object |
| 9 listed\_in 8790 non-null dtypes: int64(1), object(9) memory usage: 686.8+ KB | object |

[43]:

data[

'

rating

'

]

.

value\_counts()

|  |  |
| --- | --- |
| [43]: rating |  |
| TV-MA | 3205 |
| TV-14 | 2157 |
| TV-PG | 861 |
| R | 799 |
| PG-13 | 490 |
| TV-Y7 | 333 |
| TV-Y | 306 |
| PG | 287 |
| TV-G | 220 |
| NR | 79 |
| G | 41 |
| TV-Y7-FV | 6 |
| NC-17 | 3 |
| UR | 3 |

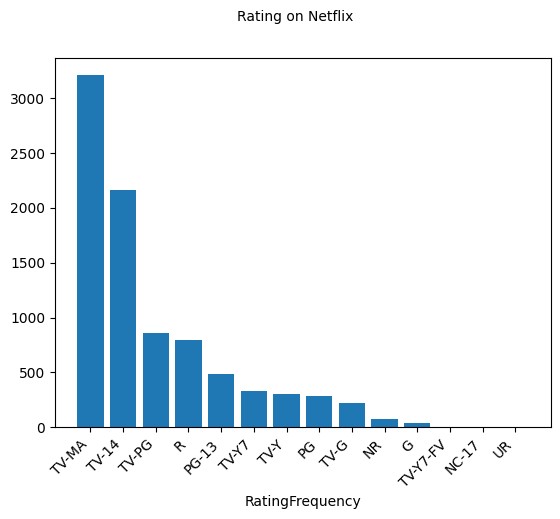
Name: count, dtype: int64

[61]: ratings=data['rating'].value\_counts().reset\_index().sort\_values(by='count',␣

↪ascending=**False**) plt.bar(ratings['rating'],ratings['count']) plt.xticks(rotation=45, ha='right') plt.xlabel('RatingFrequency')

plt.suptitle('Rating on Netflix',fontsize=10)

[61]: Text(0.5, 0.98, 'Rating on Netflix')



[83]:

plt

.

pie(ratings[

'

count

'

][:

8

]

,

labels

=

ratings[

'

rating

'

][:

8

]

,

autopct

=

'

**%.0f**

**%%**

'

)

plt

.

suptitle(

'

Rating on Netflix

'

,fontsize

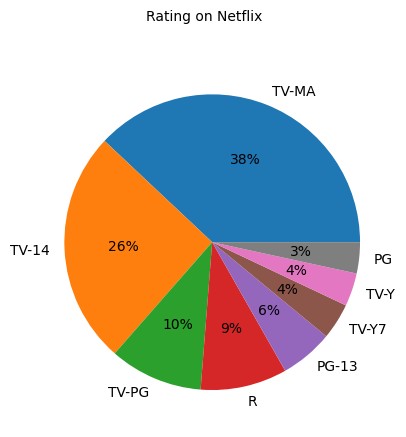
=

10

)

[83]:

Text(0.5, 0.98, 'Rating on Netflix')



[87]:

data[

'

date added

'

]

=

pd

.

to\_datetime(data[

'

date\_added

'

])

[89]:

data

.

describe()

|  |  |
| --- | --- |
| [89]: | release\_year date added |
| count | 8790.000000 8790 |
| mean | 2014.183163 2019-05-17 21:44:01.638225408 |
| min | 1925.000000 2008-01-01 00:00:00 |
| 25% | 2013.000000 2018-04-06 00:00:00 |
| 50% | 2017.000000 2019-07-03 00:00:00 |
| 75% | 2019.000000 2020-08-19 18:00:00 |
| max | 2021.000000 2021-09-25 00:00:00 |
| std | 8.825466 NaN |

[91]:

data[

'

country

'

]

.

value\_counts()

|  |  |
| --- | --- |
| [91]: country |  |
| United States | 3240 |
| India | 1057 |
| United Kingdom | 638 |
| Pakistan | 421 |

|  |  |
| --- | --- |
| Not Given | 287  … |
| Iran | 1 |
| West Germany | 1 |
| Greece | 1 |
| Zimbabwe | 1 |
| Soviet Union | 1 |

Name: count, Length: 86, dtype: int64

[95]:

top\_ten\_countries

=

data[

'

country

'

]

.

value\_counts()

.

reset\_index()

.

␣

↪

sort\_values(by

=

'

count

'

, ascending

=

**False**

)[:

10

]

plt

.

figure(figsize

=

(

10

,

6

))

plt

.

bar(top\_ten\_countries[

'

country

'

]

,

top\_ten\_countries[

'

count

'

])

plt

.

xticks(rotation

=

45

, ha

=

'

right

'

)

plt

.

xlabel(

"

Country

"

)

plt

.

ylabel(

"

Frequency

"

)

plt

.

suptitle(

"

Top 10 countries with most content on Netflix

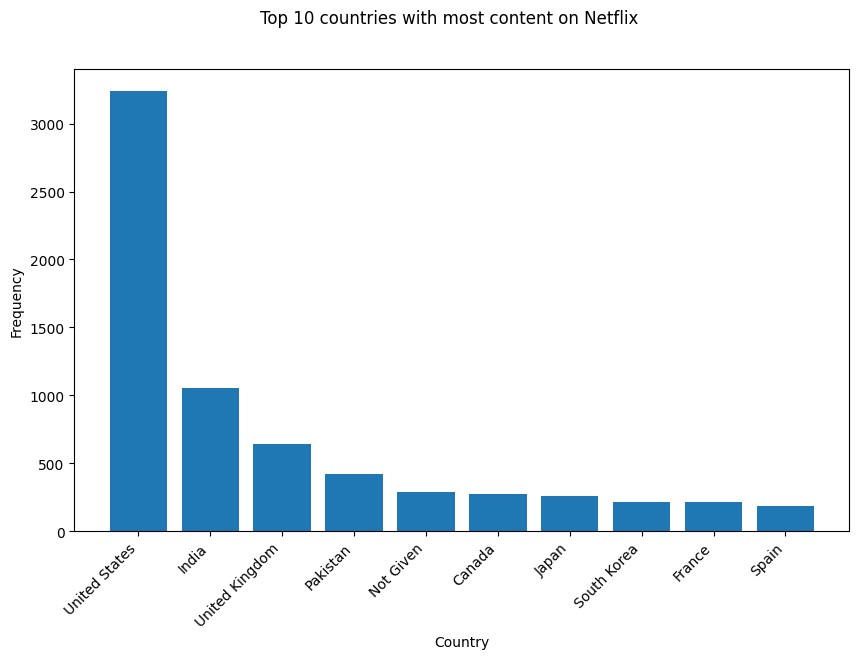
"

)

plt

.

show()



[123]: data['year']=data['date added'].dt.year data['month']=data['date added'].dt.month data['day']=data['date added'].dt.day

[147]: monthly\_movie\_release=data[data['type']=='Movie']['month'].value\_counts().

↪sort\_index() monthly\_series\_release=data[data['type']=='TV Show']['month'].value\_counts().

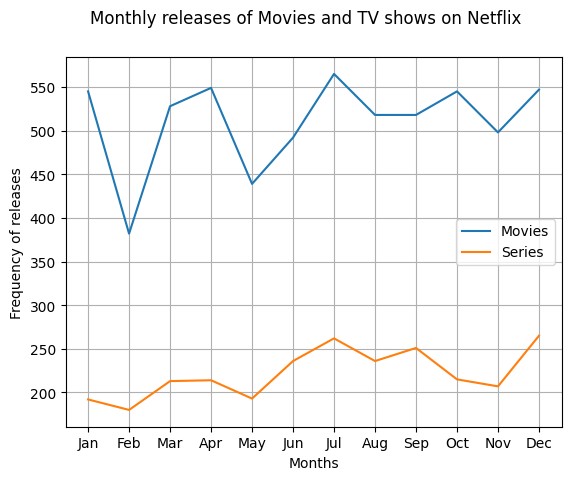
↪sort\_index() plt.plot(monthly\_movie\_release.index, monthly\_movie\_release.values,␣

↪label='Movies') plt.plot(monthly\_series\_release.index, monthly\_series\_release.values,␣

↪label='Series') plt.xlabel("Months") plt.ylabel("Frequency of releases") plt.xticks(range(1, 13),

['Jan', 'Feb', 'Mar', 'Apr', 'May','Jun', 'Jul', 'Aug', 'Sep', 'Oct', 'Nov',␣

↪'Dec']) plt.legend() plt.grid(**True**) plt.suptitle("Monthly releases of Movies and TV shows on Netflix") plt.show()



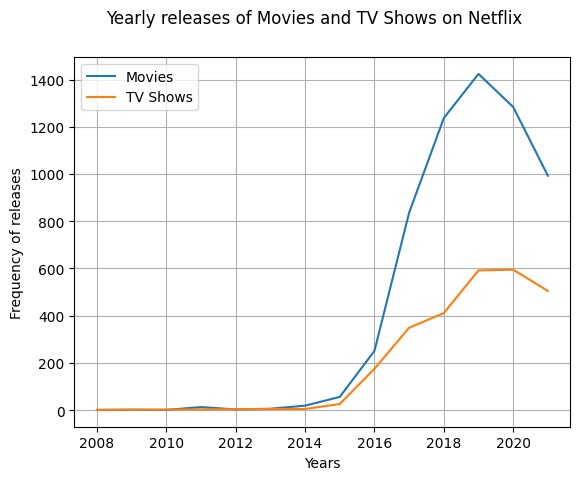
[145]: yearly\_movie\_releases=data[data['type']=='Movie']['year'].value\_counts().

↪sort\_index() yearly\_series\_releases=data[data['type']=='TV Show']['year'].value\_counts().

↪sort\_index() plt.plot(yearly\_movie\_releases.index,yearly\_movie\_releases.

↪values,label='Movies') plt.plot(yearly\_series\_releases.index,yearly\_series\_releases.values,label='TV␣

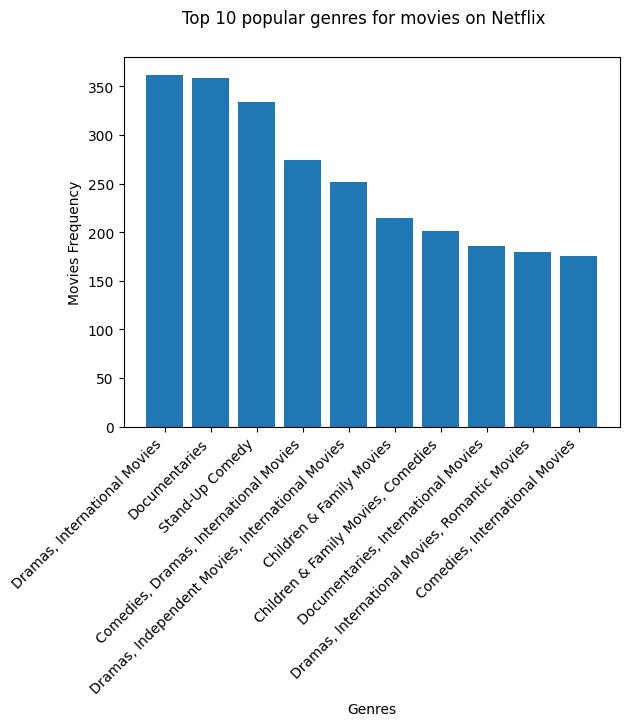
↪Shows') plt.xlabel("Years") plt.ylabel("Frequency of releases") plt.grid(**True**) plt.suptitle("Yearly releases of Movies and TV Shows on Netflix") plt.legend() plt.show()



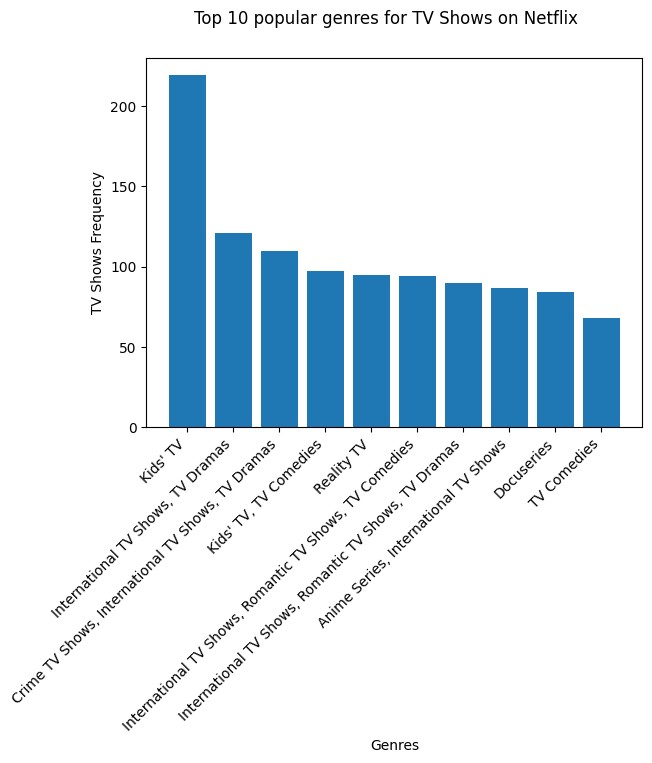
[155]: popular\_movie\_genre=data[data['type']=='Movie'].groupby("listed\_in").size().

↪sort\_values(ascending=**False**)[:10] popular\_series\_genre=data[data['type']=='TV Show'].groupby("listed\_in").size().

↪sort\_values(ascending=**False**)[:10] plt.bar(popular\_movie\_genre.index, popular\_movie\_genre.values) plt.xticks(rotation=45, ha='right') plt.xlabel("Genres") plt.ylabel("Movies Frequency") plt.suptitle("Top 10 popular genres for movies on Netflix") plt.show()



[157]: plt.bar(popular\_series\_genre.index,popular\_series\_genre.values) plt.xticks(rotation=45, ha='right') plt.xlabel("Genres") plt.ylabel("TV Shows Frequency") plt.suptitle("Top 10 popular genres for TV Shows on Netflix") plt.show()



[161]: directors=data['director'].value\_counts().reset\_index().

↪sort\_values(by='count',ascending=**False**)[1:15] plt.bar(directors['director'], directors['count']) plt.xticks(rotation=45, ha='right')

[161]: ([0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13],

[Text(0, 0, 'Rajiv Chilaka'),

Text(1, 0, 'Alastair Fothergill'),

Text(2, 0, 'Raúl Campos, Jan Suter'),

Text(3, 0, 'Suhas Kadav'),

Text(4, 0, 'Marcus Raboy'),

Text(5, 0, 'Jay Karas'),

Text(6, 0, 'Cathy Garcia-Molina'),

Text(7, 0, 'Youssef Chahine'),

Text(8, 0, 'Jay Chapman'),

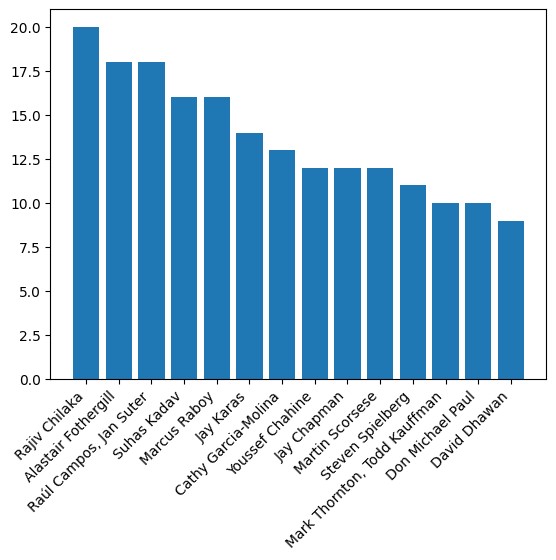
Text(9, 0, 'Martin Scorsese'),

Text(10, 0, 'Steven Spielberg'),

Text(11, 0, 'Mark Thornton, Todd Kauffman'),

Text(12, 0, 'Don Michael Paul'),

Text(13, 0, 'David Dhawan')])



[Reference link](https://github.com/Hema0708/Netflix-Data-Cleaning-Analysis-and-Visualization)