

Semester Project Code and Data Source Link and snapshot:

Data Source Link: <https://www.kaggle.com/datasets/shivamb/netflix-shows>

The screenshot shows the Kaggle dataset page for 'Netflix Movies and TV Shows'. The page includes a sidebar with navigation options like Home, Competitions, Datasets, Models, Code, Discussions, Learn, and More. The main content area displays the dataset title, a search bar, and a 'Download (1 MB)' button. Below this, there are tabs for 'Data Card', 'Code (1464)', and 'Discussion (75)'. A section titled 'How would you describe this dataset?' shows various tags like 'Well-documented 49', 'Well-maintained 20', 'Clean data 45', 'Original 16', 'High-quality notebooks 12', and 'Other'. The 'Data Explorer' section on the right shows the file 'netflix_titles.csv' (3.4 MB) with a 'Detail' tab selected. The 'About this file' section states: 'All TV Shows and Movies meta data on Netflix. Updated every month.' Below this, a table lists the columns: 'show_id', 'type', 'title', 'director', and 'cast', each with a brief description.

Code:

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
```

Created on Sun Dec 3 08:27:14 2023

```
@author: bidyabhattacharai
"""
```

Importing Necessary Libraries

```
import pandas as pd
import seaborn as sns
import warnings
warnings.filterwarnings('ignore')
```

```
import matplotlib.pyplot as plt
```

Loading the dataset

```

df = pd.read_csv('netflix_titles.csv')
df.head()

##### Displaying Column Information #####
df.columns, df.columns.__len__()

##### Checking the Null values in the dataset #####
df.count()

##### Creating the copy of dataframe #####

df_copy = df.copy()

##### Removing rows with null values #####
df_copy.dropna(inplace=True)
df_copy.count()
df_copy.info()

##### Visualizing the count of movies and TV shows #####
c = df_copy.copy().value_counts('type').reset_index()
c.columns = ['Type', 'count']
print(c)

##### plotting a pie chart for the distribution of movies and TV shows #####
plt.pie(c['count'], labels=c['Type'], autopct='%1.1f%%')
plt.show()

##### Extracting release year information #####

top_release=df_copy[['release_year']]

##### sorting and visualizing the count of releases by year #####
#####

top_release = top_release.sort_values(by='release_year', ascending=False)
top_release
plt.figure(figsize=(8, 12))
sns.countplot(data=top_release, y='release_year')
plt.xlabel('Count')

```

```
plt.ylabel('release_year')
plt.title('top release')
plt.show()
```

```
##### creating dataframe of movies#####
```

```
net_movies=df_copy[df_copy['type']=='Movie']
```

```
##### Visualizing the count of movies by rating #####
```

```
sns.countplot(data=net_movies, x='rating', )
plt.title('Movies Rating')
plt.show()
```

```
##### Extracting and visualizing the top movie ratings #####
```

```
rating = net_movies.value_counts('rating')
print(rating)
rating[:5].plot(kind='pie')
plt.title('Top 5 Ratings of movies')
plt.show()
```

```
##### Extracting and visualizing the top movie genres #####
```

```
genre = net_movies.value_counts('listed_in')
print(genre)
genre[:5].plot(kind='bar')
plt.title('Top 5 Genre of movies')
plt.show()
```

```
##### movie count released by years #####
```

```
a = net_movies.copy().value_counts('release_year', ascending=False).reset_index()
a.columns = ['release_year', 'count']
print(a)
sns.scatterplot(data=a, x='release_year', y='count')
plt.title('movie count release by years')
plt.show()
```

```
##### creating dataframe of tv shows separately #####
```

```
TV_shows=df_copy[df_copy['type']=='TV Show']
```

```
##### count rated tv shows #####
```

```
sns.countplot(data=TV_shows, x='rating', )  
plt.title('TV shows rating')  
plt.show()
```

```
##### Extracting and visualizing the top movie ratings #####
```

```
rating = TV_shows.value_counts('rating')  
print(rating)  
rating[:5].plot(kind='pie')  
plt.title('Top 5 Ratings of TV_shows')  
plt.show()
```

```
##### TV_shows count released by years #####
```

```
a = TV_shows.copy().value_counts('release_year', ascending=False).reset_index()  
a.columns = ['release_year', 'count']  
print(a)  
sns.scatterplot(data=a, x='release_year', y='count')  
plt.show()
```

```
#####
```

```
##### removing 'min' from the movie duration column #####
```

```
net_movies['duration']=net_movies['duration'].str.replace(' min','')  
net_movies['duration']=net_movies['duration'].astype(str).astype(int)  
net_movies['duration']
```

```
#####Plotting a kernel density estimate for movie time duration#####
```

```
net_movies['duration'].plot(kind='kde')  
plt.title('Movie Time Duration')
```

```
#####Extracting the number of seasons from TV shows data#####
```

```
features=['title','duration']  
durations= TV_shows[features]
```

```
durations['no_of_seasons']=durations['duration'].str.replace(' Season','')
```

```
durations['no_of_seasons']=durations['no_of_seasons'].str.replace('s','')

durations['no_of_seasons']=durations['no_of_seasons'].astype(str).astype(int)

t=['title','no_of_seasons']
top=durations[t]

##### Sorting and visualizing the top 20 TV shows
##### by number of seasons#####

top=top.sort_values(by='no_of_seasons', ascending=False)

top20=top[0:20]
top20.plot(kind='bar',x='title',y='no_of_seasons')
```