SAMPLE CODE

```
import re

import numpy as np

import pandas as pd

import seaborn as sns

import matplotlib.pyplot as plt

import random

%cd C:\Users\Pranav Gopal N V\Downloads\archive (9)

t=pd.read_csv("tv_shows.csv")

t.head()

t.dtypes

import re

t['IMDb'].isnull().sum()

t['IMDb']= t['IMDb'].replace(np.nan, 0)

t.head()

t["Age"]=t["Age"].str.replace("+","",regex=False)

t['Age']=pd.to_numeric(t['Age'],errors='coerce')

t["Rotten Tomatoes"]=t["Rotten Tomatoes"].str.replace("/",".0")

t['Rotten Tomatoes']=pd.to_numeric(t['Rotten Tomatoes'],errors='coerce')

t["IMDb"]=t["IMDb"].str.replace("/","")

t['IMDb']=pd.to_numeric(t['IMDb'],errors='coerce')

t.head()

plt.subplots(figsize=(4,6))
```

```
sns.histplot(t["Year"],kde=False, color="blue")


print("TV Shows with highest IMDb ratings are= ")

print((t.sort_values("IMDb",ascending=False).head(20))['Title'])

print("TV Shows with highest Rotten Tomatoes scores are= ")

print((t.sort_values("Rotten Tomatoes",ascending=False).head(20))['Title'])

ratings=t[["Title",'IMDb',"Rotten Tomatoes"]]

ratings.head()

len(ratings)

ratings.info()

ratings=ratings.dropna()

ratings["IMDb"]=ratings["IMDb"]*10

ratings.head()

X=ratings[["IMDb","Rotten Tomatoes"]]

X.head()

plt.figure(figsize=(10,6))

sns.scatterplot(x = 'IMDb',y = 'Rotten Tomatoes',  data = X  ,s = 70 )

plt.xlabel('IMDb rating (multiplied by 10)')

plt.ylabel('Rotten Tomatoes')

plt.title('IMDb rating (multiplied by 10) vs Rotten Tomatoes Score')

plt.show()

from sklearn.cluster import KMeans

wcss=[]

for i in range(1,11):
    km=KMeans(n_clusters=i)
    km.fit(X)
    wcss.append(km.inertia_)

plt.figure(figsize=(12,6))
plt.plot(range(1,11),wcss)
plt.plot(range(1,11),wcss, linewidth=2, color="red", marker ="8")
plt.xlabel("K Value")
plt.xticks(np.arange(1,11,1))
plt.ylabel("WCSS")
plt.show()
km=KMeans(n_clusters=4)
km.fit(X)
y=km.predict(X)
ratings["label"]=y
ratings.head()
plt.figure(figsize=(10,6))
sns.scatterplot(x = 'IMDb',y = 'Rotten Tomatoes',hue="label",
                palette=['green','orange','red','blue'], legend='full',data = ratings  ,s = 60 )
plt.xlabel('IMDb rating(Multiplied by 10)')
plt.ylabel('Rotten Tomatoes score')
plt.title('IMDb rating(Multiplied by 10) vs Rotten Tomatoes score')
plt.show()
print('Number of Cluster 0 TV Shows are=')
print(len(ratings[ratings["label"]==0]))

print("------------------------------------------")

print('Number of Cluster 1 TV Shows are=')

print(len(ratings[ratings["label"]==1]))

print("------------------------------------------")

print('Number of Cluster 2 TV Shows are=')

print(len(ratings[ratings["label"]==2]))

print("------------------------------------------")

print('Number of Cluster 3 TV Shows are=')

print(len(ratings[ratings["label"]==3]))

print("------------------------------------------")

print('TV Shows in cluster 0')

print(ratings[ratings["label"]==0]["Title"].values)

print('TV Shows in cluster 0')

for title in ratings[ratings["label"] == 0]["Title"].values:

    print(title)

print('TV Shows in cluster 1')

for title in ratings[ratings["label"] == 1]["Title"].values:

    print(title)

print('TV Shows in cluster 2')

for title in ratings[ratings["label"] == 2]["Title"].values:

    print(title)
print('TV Shows in cluster 3')

for title in ratings[ratings["label"] == 3]["Title"].values:

    print(title)
```