**Assignment-1**

Name:Dasari Abhiram

REG NO.: 41733021

Branch: CSE-DS

University: Sathyabama University

Aim: To recommend the best anime for a new anime viewer.

Algorithm:

1. Import all the required libraries.
2. Take the dataset, remove all the duplicates.
3. Print the information of the dataframe.
4. Rename the columns.
5. Remove the null values.
6. Plot a graph on anime vs ranking.
7. Plot a graph on anime vs total members.
8. Plot a graph on category vs frequency, to find the most viewed category of anime.
9. Print the best recommended anime for a new viewer.

Code:

I/p-1:

import pandas as pd

import numpy as np

I/p-2:

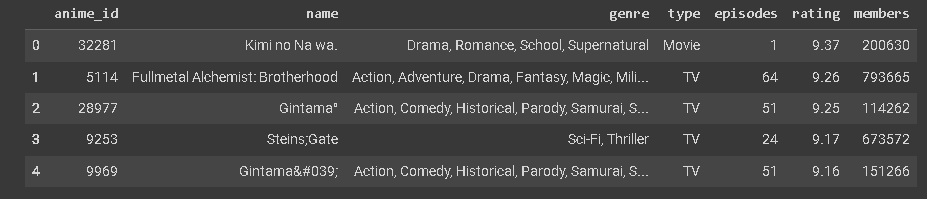
anime=pd.read\_csv('/content/anime.csv' ,sep=',')

classification=pd.read\_csv('/content/rating.csv' ,sep=',')

I/p-3:

anime.head()

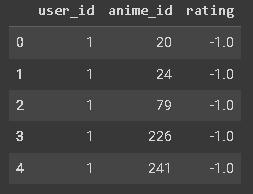
O/p-1:



I/p-4:

classification.head()

O/p-2:



I/p-5:

  print(anime.shape)

print(classification.shape)

O/p-3:

(12294, 7)

(315535, 3)

I/p-6:

print(anime.dtypes)

print(classification.dtypes)

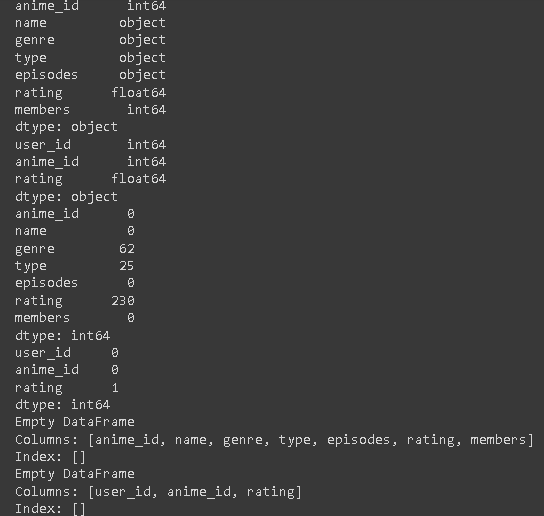
print(anime.isna() .sum())

print(classification.isna() .sum())

print(anime.loc[anime.duplicated()])

print(classification.loc[classification.duplicated()])

O/p-4:



I/p-7:

anime.rename(columns={'name':'Name','genre':'Genre','type':'Type','episodes':'Episodes','rating':'classification','members':'Members'},inplace=True)

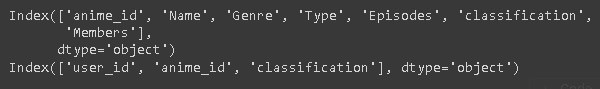
classification.rename(columns={'rating':'classification'},inplace=True)

print(anime.columns)

print(classification.columns)

classification.drop\_duplicates(inplace=True)

O/p-5:



I/p-8:

anime['classification'].fillna(anime['classification'].median() , inplace=True)

anime['Genre'].fillna(anime['Genre'].mode()[0],inplace=True)

anime['Type'].fillna(anime['Type'].mode()[0],inplace=True)

I/p-9:

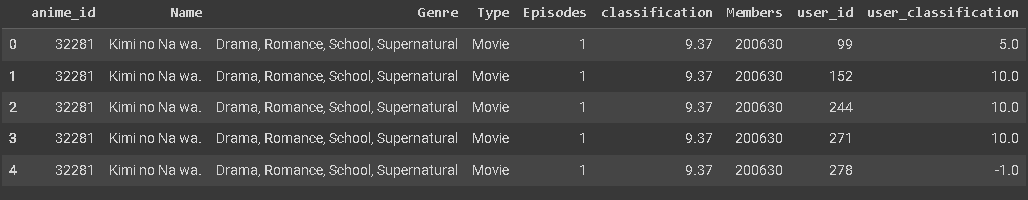
df=pd.merge(anime,classification,on='anime\_id')

df.rename(columns={'classification\_y':'user\_classification','classification\_x':'classification'},inplace=True)

df.shape

df.head()

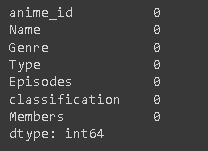
O/p-6:



I/p-10:

anime.isnull().sum()

O/p-7:



I/p-11:

from sklearn.preprocessing import StandardScaler

numerical\_columns = ['classification', 'Members']

scaler = StandardScaler()

df[numerical\_columns]=scaler.fit\_transform(df[numerical\_columns])

I/p-12:

  from sklearn.preprocessing import LabelEncoder

  categorical\_columns=['Name','Genre','Type','user\_classification']

  label\_encoder=LabelEncoder()

  for col in categorical\_columns:

    df[col]=label\_encoder.fit\_transform(df[col])

I/p-13:

import plotly.express as px

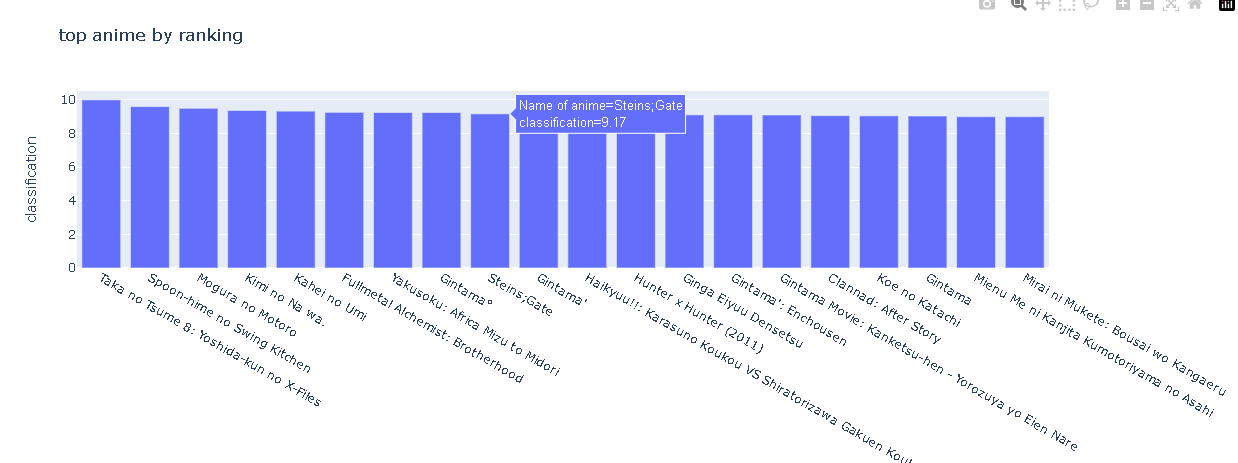
Top\_anime\_classification=anime.sort\_values(by='classification', ascending=False )

fig=px.bar(x=Top\_anime\_classification['Name'][:20],y=Top\_anime\_classification['classification'][:20],

           labels={'x':'Name of anime','y':'classification'},title='top anime by ranking')

fig.show()

O/p-8:



I/p-14:

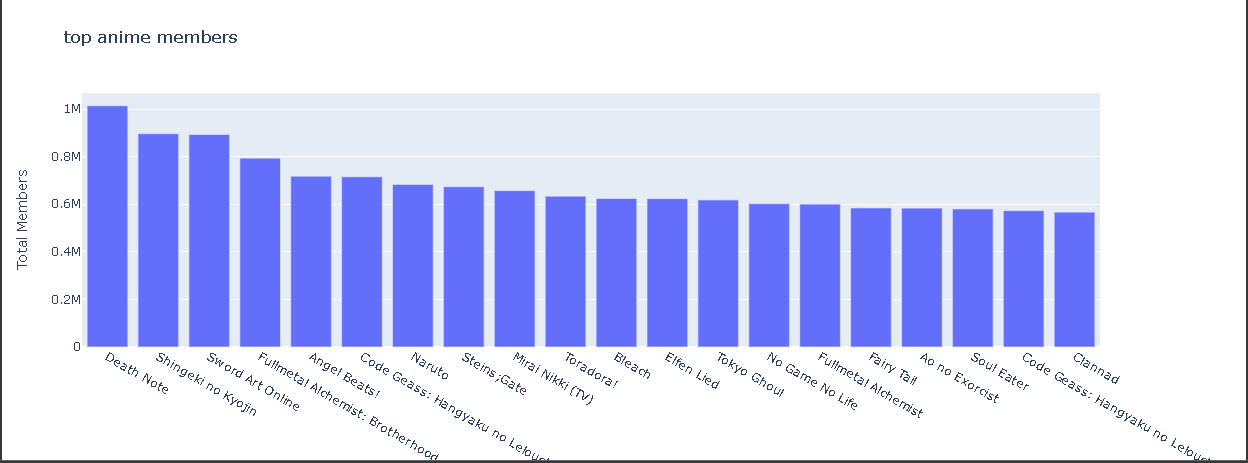
Top\_anime\_members=anime.sort\_values (by='Members', ascending=False)

fig1=px.bar(x=Top\_anime\_members['Name'][:20], y=Top\_anime\_members['Members'][:20],

labels={'x':'Name of anime', 'y': 'Total Members'},title='top anime members')

fig1.show()

O/p-9:



I/p-15:

fig2=px.bar(

x=anime.value\_counts('Type').index, y=anime.value\_counts('Type').values,

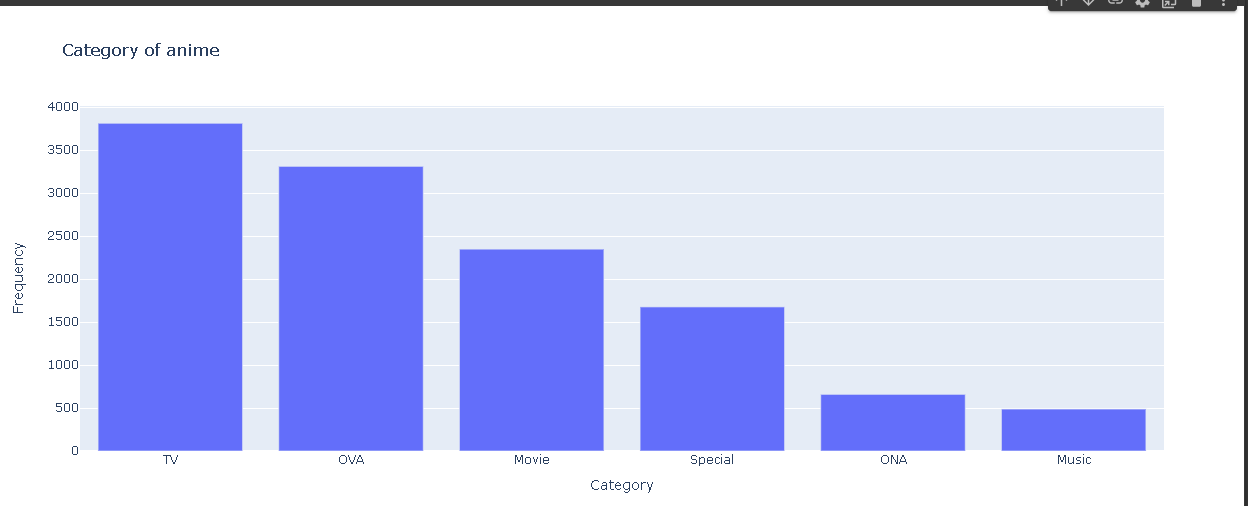
labels={'x': 'Category', 'y': 'Frequency'},

title='Category of anime'

)

fig2.show()

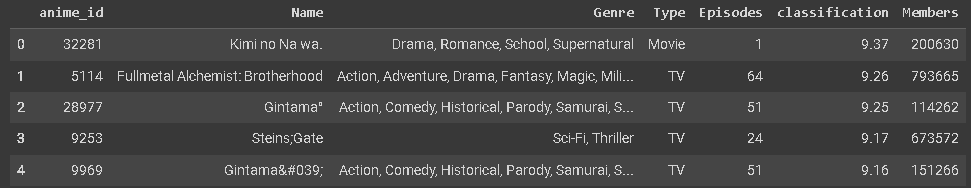
O/p-10:



I/p-16:

anime.head()

O/p-11:



Result:

Therefore, the most recommended anime is kimi no Na wa.