

Importing libraries and Anime Data set

```
In [1]: import pandas as pd
import numpy as np
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.metrics.pairwise import cosine_similarity

df = pd.read_csv(r"d:\Users\USER\Desktop\New folder\Anime database\anime.csv")

In [2]: df.head()
```

	anime_id	name	genre	type	episodes	rating	members
0	32281	Kimi no Na wa.	Drama, Romance, School, Supernatural	Movie	1	9.37	200630
1	5114	Fullmetal Alchemist: Brotherhood	Action, Adventure, Drama, Fantasy, Magic, Mil...	TV	64	9.26	793665
2	28977	Gintama°	Action, Comedy, Historical, Parody, Samurai, S...	TV	51	9.25	114262
3	9253	Steins;Gate	Sci-Fi, Thriller	TV	24	9.17	673572
4	9969	Gintama&#039;	Action, Comedy, Historical, Parody, Samurai, S...	TV	51	9.16	151266

```
In [3]: df.shape

Out[3]: (12294, 7)
```

```
In [4]: df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 12294 entries, 0 to 12293
Data columns (total 7 columns):
 #   Column      Non-Null Count  Dtype
---  -
 0   anime_id    12294 non-null  int64
 1   name        12294 non-null  object
 2   genre       12232 non-null  object
 3   type        12269 non-null  object
 4   episodes    12294 non-null  object
 5   rating      12064 non-null  float64
 6   members     12294 non-null  int64
dtypes: float64(1), int64(2), object(4)
memory usage: 672.5+ KB

==== Check for the Missing Values ====
```

```
In [5]: ((df.isnull().sum().sort_values(ascending = False))/(len(df)))*100

Out[5]: rating      1.870831
genre         0.504311
type          0.203351
anime_id      0.000000
name          0.000000
episodes      0.000000
members       0.000000
dtype: float64

It seems 3 Cloumns have null Values

Instead of Deleting an anime for having Null Value lets fill

- 'rating' with 6.3 rating saying it is neither Good nor Bad
- 'type' and 'Genre' with the Mode Value
```

```
In [6]: df['rating'].fillna(6.3,inplace=True)
df['type'].fillna(df['type'].mode().values[0],inplace=True)
df['genre'].fillna(df['genre'].mode().values[0],inplace=True)

((df.isnull().sum().sort_values(ascending = False))/(len(df)))*100

Out[6]: anime_id      0.0
name          0.0
genre         0.0
type          0.0
episodes      0.0
rating        0.0
members       0.0
dtype: float64

In [7]: print(df['type'].mode())
print(df['genre'].mode())

0      TV
Name: type, dtype: object
0      Hentai
Name: genre, dtype: object

OOO Wierd!!! Lets Make This Dataset Kids Friendly
```

```
In [8]: df=df[df['genre'] != 'Hentai']
print(df['genre'].mode())
print(len(df))

0      Comedy
Name: genre, dtype: object
11409

From 12294 to 11409 Rows
```

```
In [9]: #combine The columns 'name', 'genre'and 'type'

df['combine']=df['name']+" "+df['genre']+" "+df['type']
df['combine'].str.replace(' ', "")
df.head()

Out[9]: anime_id      name      genre      type      episodes      rating      members      combine

0      32281      Kimi no Na wa.      Drama, Romance, School, Supernatural      Movie      1      9.37      200630      Kimi no Na wa. Drama, Romance, School, Superna...
1      5114      Fullmetal Alchemist: Brotherhood      Action, Adventure, Drama, Fantasy, Magic, Mil...      TV      64      9.26      793665      Fullmetal Alchemist: Brotherhood Action, Adven...
2      28977      Gintama°      Action, Comedy, Historical, Parody, Samurai, S...      TV      51      9.25      114262      Gintama° Action, Comedy, Historical, Parody, S...
3      9253      Steins;Gate      Sci-Fi, Thriller      TV      24      9.17      673572      Steins;Gate Sci-Fi, Thriller TV
4      9969      Gintama&#039;      Action, Comedy, Historical, Parody, Samurai, S...      TV      51      9.16      151266      Gintama&#039; Action, Comedy, Historical, Paro...
```

We make new DataFrame with Anime\_id and combine columns

```
In [10]: df=df[['anime_id', 'name', 'combine']]
df.head()

Out[10]: anime_id      name      combine

0      32281      Kimi no Na wa.      Kimi no Na wa. Drama, Romance, School, Superna...
1      5114      Fullmetal Alchemist: Brotherhood      Fullmetal Alchemist: Brotherhood Action, Adven...
2      28977      Gintama°      Gintama° Action, Comedy, Historical, Parody, S...
3      9253      Steins;Gate      Steins;Gate Sci-Fi, Thriller TV
4      9969      Gintama&#039;      Gintama&#039; Action, Comedy, Historical, Paro...
```

Now We use propeties of CountVectorizer.fit\_transform and toarray functions to convert data in 'combine' Column into as array of Tokens

```
In [11]: cv=CountVectorizer()
simmatrix = cv.fit_transform(df["combine"])
simmatrix=simmatrix.toarray()

In [12]: #Using Cosine Similarity Model

cosinesim=cosine_similarity(simmatrix)

In [13]: # df of anime similarities
anisimdf = pd.DataFrame(cosinesim,index=df.name,columns=df.name)
anisimdf.head()
```

	name	Kimi no Na wa.	Fullmetal Alchemist: Brotherhood	Gintama°	Steins;Gate	Gintama&#039;	Haikyuu!!: Karasuno Koukou VS Shiratorizawa Gakuen Koukou	Hunter x Hunter (2011)	Ginga Eiyuu Densetsu	Gintama Movie: Kanketsu-hen - Yorozuya yo Eien Nare	Gintama&#039;; Enchousen	...	Hi Gekiga Ukiyoe Senya Ichiya	Hitorijime My Hero	Ikenai Boy	Ikenai Boy: Iksu Maruhi Hand Power	Manga Edo Erobanashi	Nudl Nude	Nudl Nude 2	Orig Vi Roma Anima
name																				
Kimi no Na wa.	1.000000	0.100504	0.000000	0.000000	0.000000	0.000000	0.172133	0.000000	0.111111	0.152944	0.000000	...	0.117851	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.125
Fullmetal Alchemist: Brotherhood	0.100504	1.000000	0.286039	0.123091	0.272727	0.233550	0.363636	0.201008	0.138343	0.261116	...	0.106600	0.134840	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000
Gintama°	0.000000	0.286039	1.000000	0.387298	0.953463	0.244949	0.286039	0.210819	0.652929	0.912871	...	0.111803	0.141421	0.129099	0.105409	0.239046	0.258199	0.258199	0.115	0.115
Steins;Gate	0.000000	0.123091	0.387298	1.000000	0.369274	0.105409	0.123091	0.272166	0.187317	0.353553	...	0.000000	0.182574	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000
Gintama&#039;;	0.000000	0.272727	0.953463	0.369274	1.000000	0.233550	0.272727	0.201008	0.622543	0.957427	...	0.106600	0.134840	0.123091	0.100504	0.227921	0.246183	0.246183	0.115	0.115

5 rows x 11409 columns

The Diagnol Values are 1 since it is Exact Match To Itself

```
In [16]: x=input("Enter Anime name : ")

Enter Anime name : Haikyuu!!

In [15]: y=anisimdf[anisimdf.index==x]
y=y.transpose()
print("Top 10 similar anime recommendations are:")
y=y.sort_values(by=[x],ascending=False)

for i in range(len(y))[1:11]:
    print(y.index[i])

Top 10 similar anime recommendations are:
Haikyuu!! Second Season
Slam Dunk
Teekyuu 2
Teekyuu
Teekyuu 3
Major S4
Cheonbangjichuk Hani
Yowamushi Pedal
Major S1
Major S2

Haikyuu!! Being A sports Anime We got some recommendations of Sports Anime

Entire Code
```

```
In [96]: import pandas as pd
import numpy as np
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.metrics.pairwise import cosine_similarity

df = pd.read_csv(r"d:\Users\USER\Desktop\New folder\Anime database\anime.csv")

df['rating'].fillna(6.3,inplace=True)
df['type'].fillna(df['type'].mode().values[0],inplace=True)
df['genre'].fillna(df['genre'].mode().values[0],inplace=True)

df=df[df['genre'] != 'Hentai']

df['combine']=df['name']+" "+df['genre']+" "+df['type']
df['combine'].str.replace(' ', "")

df=df[['anime_id', 'name', 'combine']]

cv=CountVectorizer()
simmatrix = cv.fit_transform(df["combine"])
simmatrix=simmatrix.toarray()

cosinesim=cosine_similarity(simmatrix)

anisimdf = pd.DataFrame(cosinesim,index=df.name,columns=df.name)

x=input("Enter Anime name : ")

y=anisimdf[anisimdf.index==x]
y=y.transpose()
print("Top 10 similar anime recommendations are:")
y=y.sort_values(by=[x],ascending=False)

for i in range(len(y))[1:11]:
    print(y.index[i])

Top 10 similar anime recommendations are:
Boku no Hero Academia 2nd Season
Boku no Hero Academia: Jump Festa 2016 Special
Code:Breaker
Bleach
Tenjou Tenge
Medaka Box
Tokyo Ravens
Katekyo Hitman Reborn!
Naruto
Yozakura Quartet: Hana no Uta
```