```
import pandas as pd
import matplotlib.pyplot as plt
from wordcloud import WordCloud
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.metrics.pairwise import linear_kernel
```

```
movies=pd.read_csv('/content/movies.csv')
ratings=pd.read_csv('/content/ratings.csv')
```

```
movies.info()

```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10329 entries, 0 to 10328
Data columns (total 3 columns):
 #   Column   Non-Null Count  Dtype 
---  ------   --------------  ----- 
 0   movieId  10329 non-null  int64 
 1   title    10329 non-null  object
 2   genres   10329 non-null  object
dtypes: int64(1), object(2)
memory usage: 242.2+ KB
```

```
ratings.info()

```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 105339 entries, 0 to 105338
Data columns (total 4 columns):
 #   Column     Non-Null Count   Dtype  
---  ------     --------------   -----  
 0   userId     105339 non-null  int64  
 1   movieId    105339 non-null  int64  
 2   rating     105339 non-null  float64
 3   timestamp  105339 non-null  int64  
dtypes: float64(1), int64(3)
memory usage: 3.2 MB
```

```
movies.shape

```

```
(10329, 3)
```

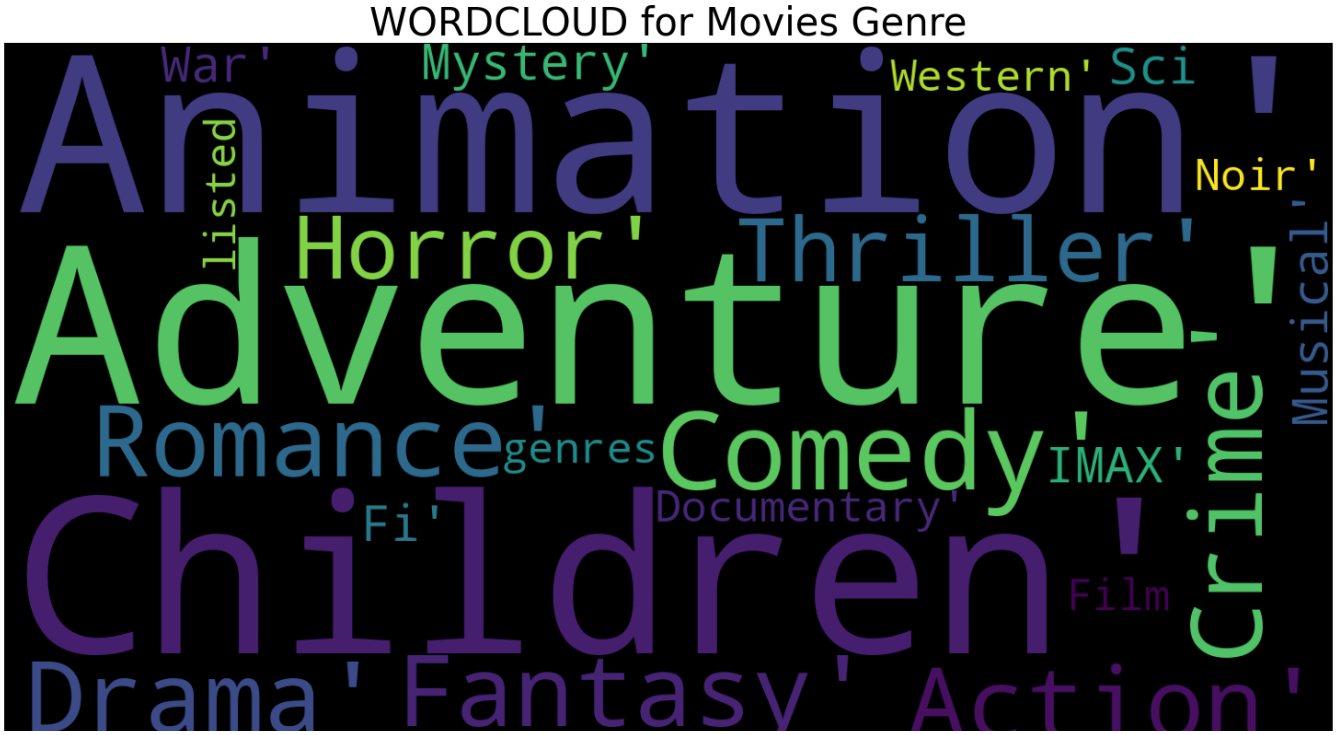
```
genres=[]
for genre in movies.genres:

    x=genre.split('|')
    for i in x:
```

```
        if i not in genres:
           genres.append(str(i))
genres=str(genres)
movie_title=[]
for title in movies.title:
    movie_title.append(title[0:-7])
movie_title=str(movie_title)
```

```
wordcloud_genre=WordCloud(width=1500,height=800,background_color='black',min_font_size=2
                    ,min_word_length=3).generate(genres)
wordcloud_title=WordCloud(width=1500,height=800,background_color='cyan',min_font_size=2
                    ,min_word_length=3).generate(movie_title)
```

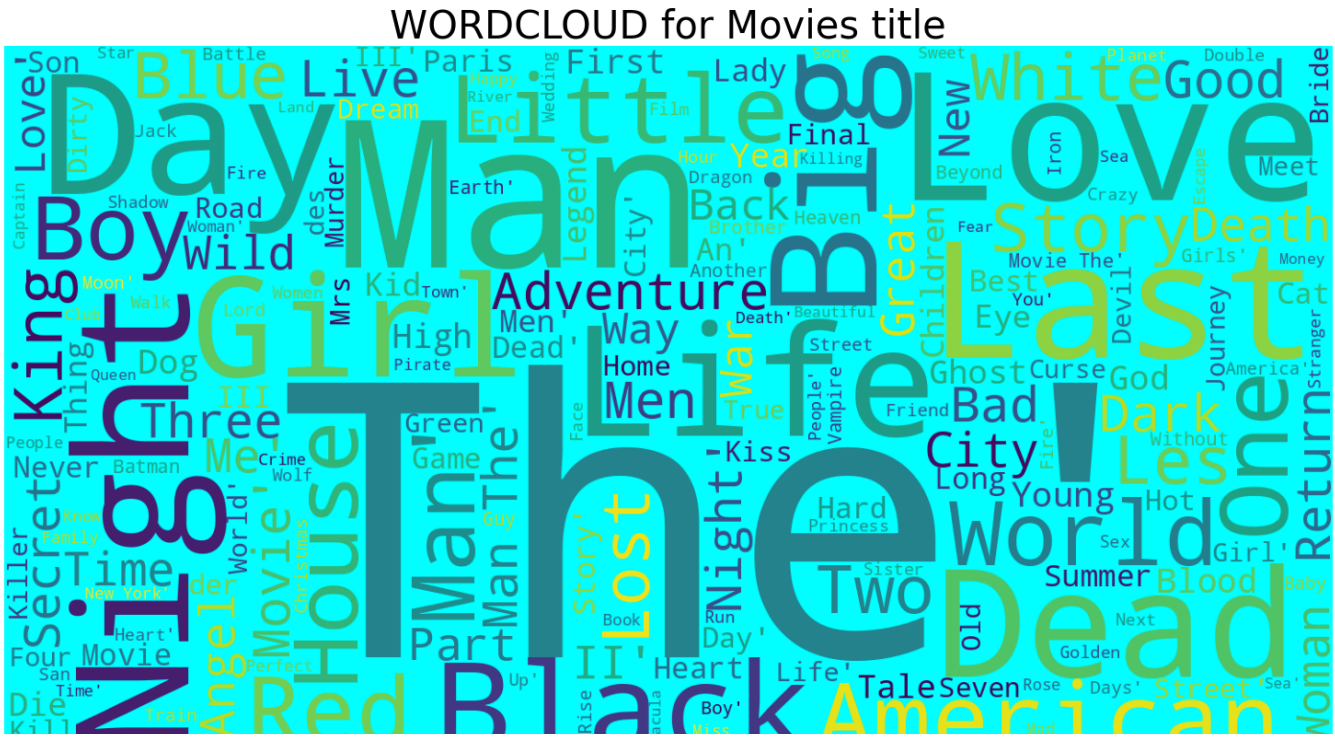
```
plt.figure(figsize=(30,10))
plt.axis('off')
plt.title('WORDCLOUD for Movies Genre',fontsize=30)
plt.imshow(wordcloud_genre)
```

```
<matplotlib.image.AxesImage at 0x7c32b075c160>
```




```
plt.figure(figsize=(30,10))
plt.axis('off')
plt.title('WORDCLOUD for Movies title',fontsize=30)
plt.imshow(wordcloud_title)
```

```
<matplotlib.image.AxesImage at 0x7c32af9aacb0>
```




```
df=pd.merge(ratings,movies, how='left',on='movieId')
df.head()
```

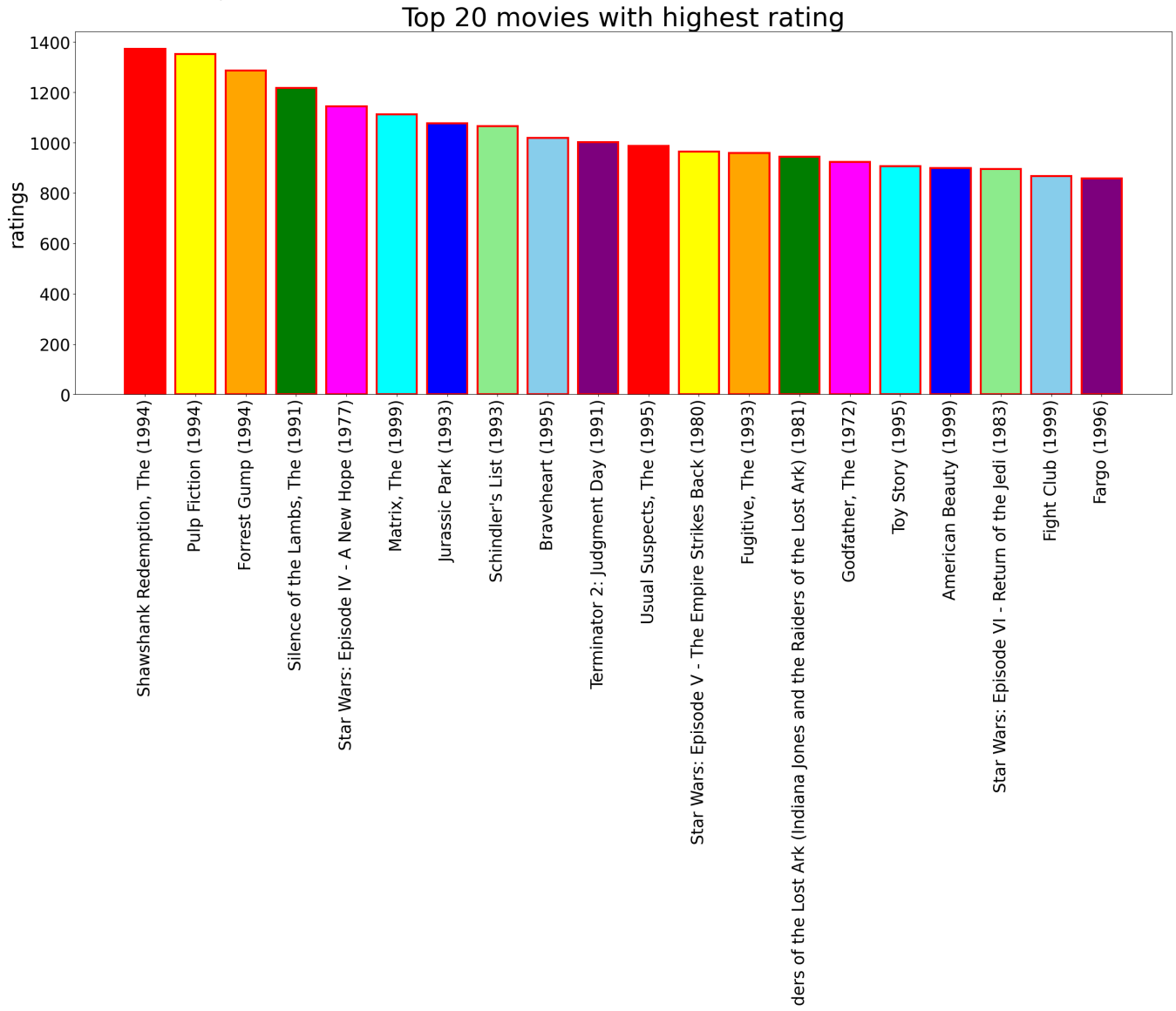
	userId	movieId	rating	timestamp	title	genres
0	1	16	4.0	1217897793	Casino (1995)	Crime|Drama
1	1	24	1.5	1217895807	Powder (1995)	Drama|Sci-Fi
2	1	32	4.0	1217896246	Twelve Monkeys (a.k.a. 12 Monkeys) (1995)	Mystery|Sci-Fi|Thriller
3	1	47	4.0	1217896556	Seven (a.k.a. Se7en) (1995)	Mystery|Thriller
4	1	50	4.0	1217896523	Usual Suspects, The (1995)	Crime|Mystery|Thriller

```
df1=df.groupby(['title'])[['rating']].sum()
high_rated=df1.nlargest(20,'rating')
high_rated.head()
```

title	rating
Shawshank Redemption, The (1994)	1372.0
Pulp Fiction (1994)	1352.0
Forrest Gump (1994)	1287.0
Silence of the Lambs, The (1991)	1216.5
Star Wars: Episode IV - A New Hope (1977)	1143.5

```
plt.figure(figsize=(30,10))
plt.title('Top 20 movies with highest rating',fontsize=40)
colors=['red','yellow','orange','green','magenta','cyan','blue','lightgreen','skyblue','p
plt.ylabel('ratings',fontsize=30)
plt.xticks(fontsize=25,rotation=90)
plt.xlabel('movies title',fontsize=30)
plt.yticks(fontsize=25)
plt.bar(high_rated.index,high_rated['rating'],linewidth=3,edgecolor='red',color=colors)
```

```
<BarContainer object of 20 artists>
```




```
1 df2=df.groupby('title')[['rating']].count()
2 rating_count_20=df2.nlargest(20,'rating')
3 rating_count_20.head()
```

title	rating
Pulp Fiction (1994)	325
Forrest Gump (1994)	311
Shawshank Redemption, The (1994)	308
Jurassic Park (1993)	294
Silence of the Lambs, The (1991)	290

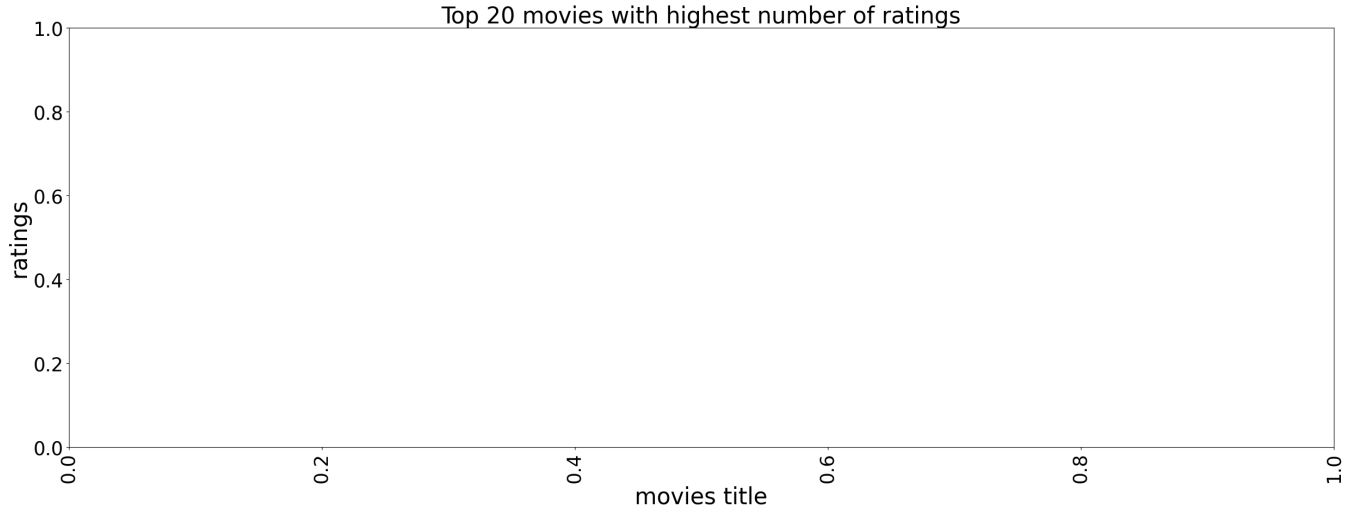
```
1 plt.figure(figsize=(30,10))
2 plt.title('Top 20 movies with highest number of ratings',fontsize=30)
```

```
plt.title('Top 20 movies with highest number of ratings',fontsize=30)
plt.xticks(fontsize=25,rotation=90)
plt.yticks(fontsize=25)
plt.xlabel('movies title',fontsize=30)
plt.ylabel('ratings',fontsize=30)

plt.bar
```

matplotlib.pyplot.bar
def bar(x, height, width=0.8, bottom=None, *, align='center', data=None, **kwargs)

Make a bar plot.

The bars are positioned at *x* with the given *align*\ment. Their dimensions are given by *height* and *width*. The vertical baseline is *bottom* (default 0).




```
cv=TfidfVectorizer()
tfidf_matrix=cv.fit_transform(movies['genres'])
```

```
movie_user = df.pivot_table(index='userId',columns='title',values='rating')
movie_user.head()
```

title	'71 (2014)	'Hellboy': The Seeds of Creation (2004)	'Round Midnight (1986)	'Til There Was You (1997)	'burbs, The (1989)	'night Mother (1986)	(500) Days of Summer (2009)	*batteries not included (1987)	...A Justi for A (197

```
cosine_sim = linear_kernel(tfidf_matrix, tfidf_matrix)

```

```
indices=pd.Series(movies.index,index=movies['title'])
titles=movies['title']
def recommendations(title):
    idx = indices[title]
    sim_scores = list(enumerate(cosine_sim[idx]))
    sim_scores = sorted(sim_scores, key=lambda x: x[1], reverse=True)
    sim_scores = sim_scores[1:21]
    movie_indices = [i[0] for i in sim_scores]
    return titles.iloc[movie_indices]

```

```
recommendations('Toy Story (1995)')

```

	title
1815	Antz (1998)
2496	Toy Story 2 (1999)
2967	Adventures of Rocky and Bullwinkle, The (2000)
3166	Emperor's New Groove, The (2000)
3811	Monsters, Inc. (2001)
6617	DuckTales: The Movie - Treasure of the Lost La...
6997	Wild, The (2006)