```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import sklearn
import warnings
warnings.filterwarnings('ignore')
```

```
from google.colab import files
uploaded = files.upload()
```

Choose Files No file chosen Upload widget is only available when the cell has been executed in the current browser session. Please rerun this cell to enable.

```
Saving archive (1).zip to archive (1).zip
```

```
full_data = pd.read_csv("/content/archive (1).zip", encoding='ISO-8859-1')
```

```
full_data
```

	Name	Year	Duration	Genre	Rating	Votes	Director	Actor 1	Actor 2	Actor 3
0		NaN	NaN	Drama	NaN	NaN	J.S. Randhawa	Manmauji	Birbal	Rajendra Bhatia
1	#Gadhvi (He thought he was Gandhi)	(2019)	109 min	Drama	7.0	8	Gaurav Bakshi	Rasika Dugal	Vivek Ghamande	Arvind Jangid
2	#Homecoming	(2021)	90 min	Drama, Musical	NaN	NaN	Soumyajit Majumdar	Sayani Gupta	Plabita Borthakur	Roy Angana
3	#Yaaram	(2019)	110 min	Comedy, Romance	4.4	35	Ovais Khan	Prateik	Ishita Raj	Siddhant Kapoor
4	...And Once Again	(2010)	105 min	Drama	NaN	NaN	Amol Palekar	Rajat Kapoor	Rituparna Sengupta	Antara Mali
...	...	...	...	...	...	...	...	...	...	...
15504	Zulm Ko Jala Doonga	(1988)	NaN	Action	4.6	11	Mahendra Shah	Naseeruddin Shah	Sumeet Saigal	Suparna Anand
15505	Zulmi	(1999)	129 min	Action, Drama	4.5	655	Kuku Kohli	Akshay Kumar	Twinkle Khanna	Aruna Irani

```
full_data.head(3)
```

	Name	Year	Duration	Genre	Rating	Votes	Director	Actor 1	Actor 2	Actor 3
0		NaN	NaN	Drama	NaN	NaN	J.S. Randhawa	Manmauji	Birbal	Rajendra Bhatia
1	#Gadhvi (He thought he was Gandhi)	(2019)	109 min	Drama	7.0	8	Gaurav Bakshi	Rasika Dugal	Vivek Ghamande	Arvind Jangid

```
full_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 15509 entries, 0 to 15508
Data columns (total 10 columns):
 #   Column    Non-Null Count  Dtype
---  ------    --------------  -----
 0   Name      15509 non-null  object
 1   Year      14981 non-null  object
 2   Duration  7240 non-null   object
 3   Genre     13632 non-null  object
 4   Rating    7919 non-null   float64
 5   Votes     7920 non-null   object
 6   Director  14984 non-null  object
 7   Actor 1   13892 non-null  object
 8   Actor 2   13125 non-null  object
 9   Actor 3   12365 non-null  object
dtypes: float64(1), object(9)
memory usage: 1.2+ MB
```

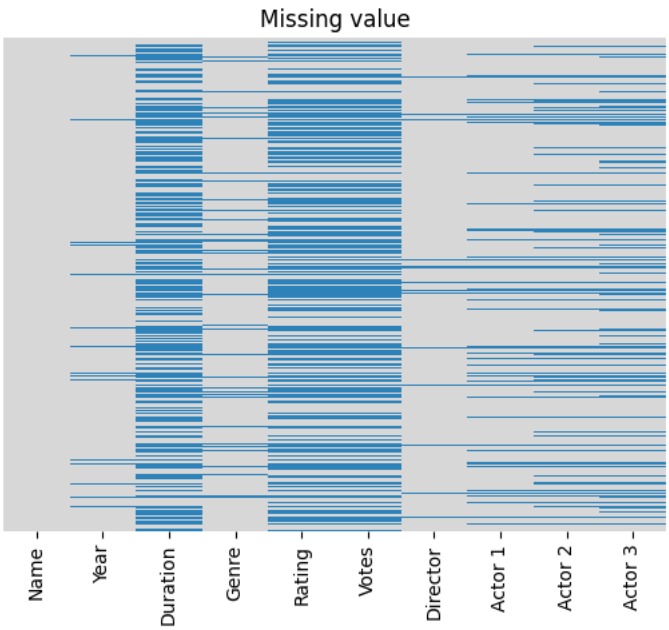
```
full_data.describe()
```

	Rating
count	7919.000000
mean	5.841621
std	1.381777
min	1.100000
25%	4.900000
50%	6.000000
75%	6.800000
max	10.000000

```
full_data.isnull().sum()
```

```
Name           0
Year         528
Duration    8269
Genre       1877
Rating      7590
Votes       7589
Director     525
Actor 1     1617
Actor 2     2384
Actor 3     3144
dtype: int64
```

```
sns.heatmap(full_data.isnull () , cmap = 'tab20c_r' , yticklabels = False , cbar = False)
plt.title("Missing value")
plt.show()
```



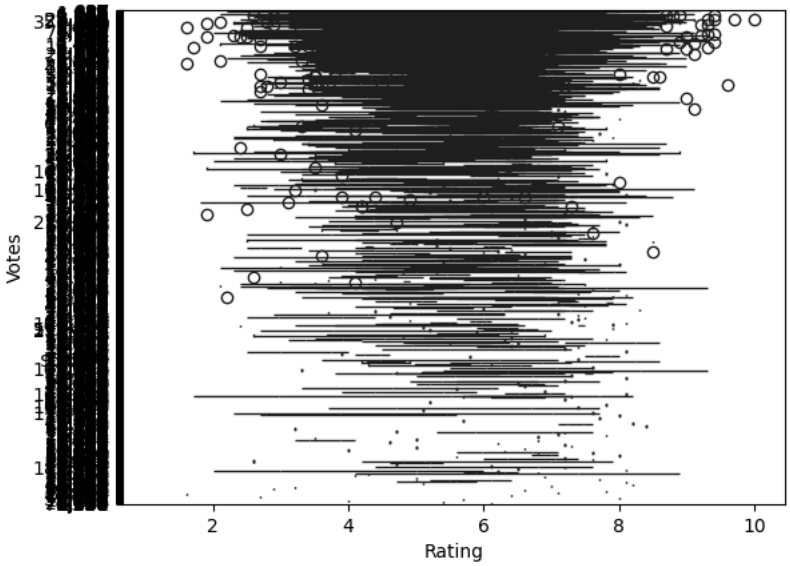

```
x= full_data.iloc [:, :-1].values
y= full_data.iloc [:, 4].values
print(x)
```

```
[[' ' nan nan ... 'J.S. Randhawa' 'Manmauji' 'Birbal']
 ['#Gadhvi (He thought he was Gandhi)' '(2019)' '109 min' ...
  'Gaurav Bakshi' 'Rasika Dugal' 'Vivek Ghamande']
 ['#Homecoming' '(2021)' '90 min' ... 'Soumyajit Majumdar' 'Sayani Gupta'
  'Plabita Borthakur']
 ...
 ['Zulmi Raj' '(2005)' nan ... 'Kiran Thej' 'Sangeeta Tiwari' nan]
 ['Zulmi Shikari' '(1988)' nan ... nan nan nan]
 ['Zulm-O-Sitam' '(1998)' '130 min' ... 'K.C. Bokadia' 'Dharmendra'
  'Jaya Prada']]
```

```
print(y)
```

```
[nan 7.  nan ... nan nan 6.2]
```

```
sns.boxplot(x = 'Rating' , y = 'Votes' , data = full_data , palette = "GnBu_d" )
plt.show()
```




```
full_data.isnull()
```

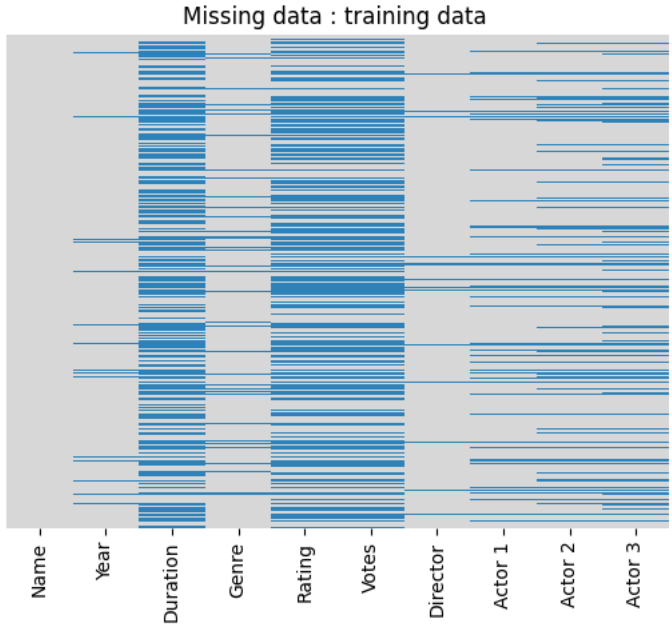
	Name	Year	Duration	Genre	Rating	Votes	Director	Actor 1	Actor 2	Actor 3
0	False	True	True	False	True	True	False	False	False	False
1	False	False	False	False	False	False	False	False	False	False
2	False	False	False	False	True	True	False	False	False	False
3	False	False	False	False	False	False	False	False	False	False
4	False	False	False	False	True	True	False	False	False	False
...	...	...	...	...	...	...	...	...	...	...
15504	False	False	True	False	False	False	False	False	False	False
15505	False	False	False	False	False	False	False	False	False	False
15506	False	False	True	False	True	True	False	False	True	True
15507	False	False	True	False	True	True	True	True	True	True
15508	False	False	False	False	False	False	False	False	False	False

15509 rows × 10 columns

```
full_data.head(4)
```

	Name	Year	Duration	Genre	Rating	Votes	Director	Actor 1	Actor 2	Actor 3
0		NaN	NaN	Drama	NaN	NaN	J.S. Randhawa	Manmauji	Birbal	Rajendra Bhatia
1	#Gadhvi (He thought he was Gandhi)	(2019)	109 min	Drama	7.0	8	Gaurav Bakshi	Rasika Dugal	Vivek Ghamande	Arvind Jangid
2	#Homecoming	(2021)	90 min	Drama, Musical	NaN	NaN	Soumyajit Majumdar	Sayani Gupta	Plabita Borthakur	Roy Angana

```
sns.heatmap(full_data.isnull () , yticklabels = False , cbar = False , cmap = "tab20c_r")
plt.title("Missing data : training data ")
plt.show()
```




```
full_data.drop('Name' , axis = 1 , inplace = True)
```

```
full_data.head(3)
```

	Year	Duration	Genre	Rating	Votes	Director	Actor 1	Actor 2	Actor 3
0	NaN	NaN	Drama	NaN	NaN	J.S. Randhawa	Manmauji	Birbal	Rajendra Bhatia
1	(2019)	109 min	Drama	7.0	8	Gaurav Bakshi	Rasika Dugal	Vivek Ghamande	Arvind Jangid
2	(2021)	90 min	Drama, Musical	NaN	NaN	Soumyajit Majumdar	Sayani Gupta	Plabita Borthakur	Roy Angana

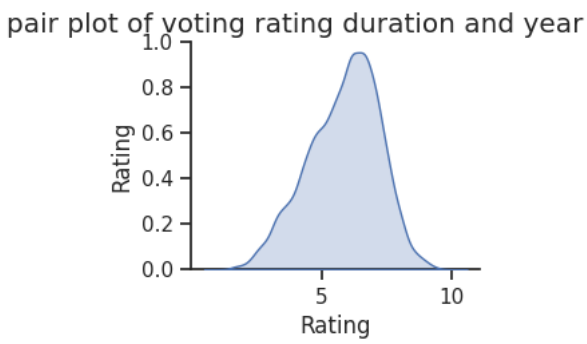
```
import pandas as pd

#Assuming you have a data frame called 'data' with your dataset
data = pd.get_dummies(full_data, columns=["Genre", "Director", "Actor 1", "Actor 2", "Actor 3"])


from sklearn.preprocessing import LabelEncoder
le = LabelEncoder()

#Assuming 'full_data' is your dataframe
full_data['Director'] = le.fit_transform(full_data['Director'])
full_data['Genre'] = le.fit_transform(full_data['Genre'])
full_data['Actor 1'] = le.fit_transform(full_data['Actor 1'])
full_data['Actor 2'] = le.fit_transform(full_data['Actor 2'])
full_data['Actor 3'] = le.fit_transform(full_data['Actor 3'])


column_of_interest = ['Votes','Rating','Duration','Year']
sns.set(style="ticks")
sns.pairplot(full_data[column_of_interest],diag_kind='kde',markers='o',palette='viridis',height=2.5,aspect=1.2)
plt.suptitle('pair plot of voting rating duration and year',y=1.02)
plt.show()
```



```
from sklearn.preprocessing import LabelEncoder
le = LabelEncoder()

#Assuming 'full_data' is your dataframe
full_data['Director'] = le.fit_transform(full_data['Director'])
full_data['Genre'] = le.fit_transform(full_data['Genre'])
full_data['Actor 1'] = le.fit_transform(full_data['Actor 1'])
full_data['Actor 2'] = le.fit_transform(full_data['Actor 2'])
full_data['Actor 3'] = le.fit_transform(full_data['Actor 3'])
```

```
full_data
```

	Year	Duration	Genre	Rating	Votes	Director	Actor 1	Actor 2	Actor 3
1	(2019)	109 min	299	7.0	8	1548	3280	4790	527
2	(2021)	90 min	351	7.0	8	5123	3713	2866	3450
3	(2019)	110 min	228	4.4	35	3319	2917	1504	4020
4	(2010)	105 min	299	4.4	35	385	3112	3462	405
5	(1997)	147 min	197	4.7	827	3800	895	123	3829
...	...	...	...	...	...	...	...	...	...
15504	(1988)	125 min	0	4.6	11	2690	2586	4299	4262
15505	(1999)	129 min	40	4.5	655	2499	227	4532	519
15506	(2005)	129 min	0	4.5	655	2424	3609	4891	4820
15507	(1988)	129 min	0	4.5	655	5938	4718	4891	4820
15508	(1998)	130 min	40	6.2	20	2195	1139	1589	490

15508 rows × 9 columns

```
data.fillna(0, inplace=True)
full_data
```

	Year	Duration	Genre	Rating	Votes	Director	Actor 1	Actor 2	Actor 3
1	(2019)	109 min	299	7.0	8	1548	3280	4790	527
2	(2021)	90 min	351	7.0	8	5123	3713	2866	3450
3	(2019)	110 min	228	4.4	35	3319	2917	1504	4020
4	(2010)	105 min	299	4.4	35	385	3112	3462	405
5	(1997)	147 min	197	4.7	827	3800	895	123	3829
...	...	...	...	...	...	...	...	...	...
15504	(1988)	125 min	0	4.6	11	2690	2586	4299	4262
15505	(1999)	129 min	40	4.5	655	2499	227	4532	519
15506	(2005)	129 min	0	4.5	655	2424	3609	4891	4820
15507	(1988)	129 min	0	4.5	655	5938	4718	4891	4820
15508	(1998)	130 min	40	6.2	20	2195	1139	1589	490

15508 rows × 9 columns

```
full_data.fillna(method='ffill', inplace=True)
data.interpolate(method='linear', inplace=True)
full_data['Genre'].fillna('unknown', inplace=True)
null_counts = full_data.isnull().sum()
print(null_counts)
```

```
Year        0
Duration    0
Genre       0
Rating      0
Votes       0
Director    0
Actor 1     0
Actor 2     0
Actor 3     0
dtype: int64
```

```
columns_to_check = ["Year", "Duration", "Rating", "Votes"]
full_data.dropna(subset=columns_to_check , inplace = True)
```

```
full_data.isnull()
```

	Year	Duration	Genre	Rating	Votes	Director	Actor 1	Actor 2	Actor 3
1	False	False	False	False	False	False	False	False	False
2	False	False	False	False	False	False	False	False	False
3	False	False	False	False	False	False	False	False	False
4	False	False	False	False	False	False	False	False	False
5	False	False	False	False	False	False	False	False	False
...	...	...	...	...	...	...	...	...	...
15504	False	False	False	False	False	False	False	False	False
15505	False	False	False	False	False	False	False	False	False
15506	False	False	False	False	False	False	False	False	False
15507	False	False	False	False	False	False	False	False	False
15508	False	False	False	False	False	False	False	False	False

15508 rows × 9 columns

```
full_data.isnull().sum()
```

```
Year        0
Duration    0
Genre       0
Rating      0
Votes       0
Director    0
Actor 1     0
Actor 2     0
Actor 3     0
dtype: int64
```

```
x = full_data.select_dtypes(include=['object'])
print(x.columns)
```

```
Index(['Year', 'Duration', 'Votes'], dtype='object')
```

```
full_data
```

	Year	Duration	Genre	Rating	Votes	Director	Actor 1	Actor 2	Actor 3
1	(2019)	109 min	299	7.0	8	1548	3280	4790	527
2	(2021)	90 min	351	7.0	8	5123	3713	2866	3450
3	(2019)	110 min	228	4.4	35	3319	2917	1504	4020
4	(2010)	105 min	299	4.4	35	385	3112	3462	405
5	(1997)	147 min	197	4.7	827	3800	895	123	3829
...	...	...	...	...	...	...	...	...	...
15504	(1988)	125 min	0	4.6	11	2690	2586	4299	4262
15505	(1999)	129 min	40	4.5	655	2499	227	4532	519
15506	(2005)	129 min	0	4.5	655	2424	3609	4891	4820
15507	(1988)	129 min	0	4.5	655	5938	4718	4891	4820
15508	(1998)	130 min	40	6.2	20	2195	1139	1589	490

15508 rows × 9 columns

```
sns.heatmap(full_data.isnull () , yticklabels = False , cbar = False , cmap = "tab20c_r")
plt.title("Missing data : training data ")
plt.show()
```

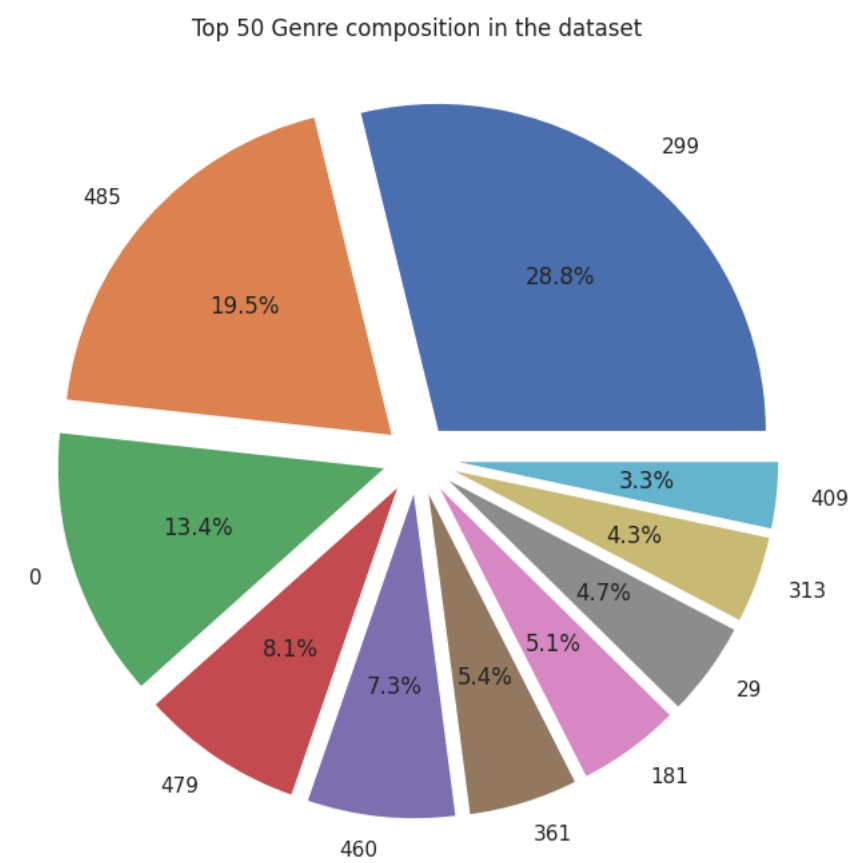

```
#Assume 'data' is your dataframe with 'Genre' column
top_50_genres = full_data['Genre'].value_counts().head(10)

plt.figure(figsize=(8,8))

#Create an 'explode' list with values to separate individual slices using a loop
explode = [0.1] * len(top_50_genres)

top_50_genres.plot(kind='pie', autopct='%1.1f%%', explode=explode)

plt.title('Top 50 Genre composition in the dataset')
plt.ylabel('')
plt.show()
```




```
full_data.shape
```

```
(15508, 9)
```

```
x = full_data.drop('Rating' , axis = 1)
y = full_data['Rating']
```