tvshows-eda

August 1, 2024

0.1 Introduction:

We have a dataset of TV shows in CSV format, which we are going to analyze. First, we will read the data by importing the dataset using the "pd.read_csv" function. After importing the data, we will observe the dataset using the "shape" and "info" functions. Then, we will check for duplicate values, missing values, and unstructured data types, etc. After that, we will visualize the data based on the most number of shows produced in a year by platforms and will visualize shows by IMDb ratings.

[1]:
```
## Importing Library for EDA Process
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import numpy as np
```

[3]:
```
df=pd.read_csv("https://raw.githubusercontent.com/MainakRepositor/Datasets/
 ↪master/TV_Shows.csv") # Reading the data set
df.head()       # Using head function to show top 5 rows of the data
```

[3]:

	Unnamed: 0	Title	Year	Age	IMDb	Rotten Tomatoes	Netflix	\
0	0	Breaking Bad	2008	18+	9.5	96%	1	
1	1	Stranger Things	2016	16+	8.8	93%	1	
2	2	Money Heist	2017	18+	8.4	91%	1	
3	3	Sherlock	2010	16+	9.1	78%	1	
4	4	Better Call Saul	2015	18+	8.7	97%	1	

	Hulu	Prime Video	Disney+	type
0	0	0	0	1
1	0	0	0	1
2	0	0	0	1
3	0	0	0	1
4	0	0	0	1

[4]:
```
df.shape      # Using shape function to see rows and columns
```

[4]:
```
(5611, 11)
```

[5]:
```
df.info() # Using info() function to get full information regarding the data.
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5611 entries, 0 to 5610
Data columns (total 11 columns):
 #   Column           Non-Null Count  Dtype  
---  ------           --------------  -----  
 0   Unnamed: 0       5611 non-null   int64  
 1   Title            5611 non-null   object 
 2   Year             5611 non-null   int64  
 3   Age              3165 non-null   object 
 4   IMDb             4450 non-null   float64
 5   Rotten Tomatoes  1010 non-null   object 
 6   Netflix          5611 non-null   int64  
 7   Hulu             5611 non-null   int64  
 8   Prime Video      5611 non-null   int64  
 9   Disney+          5611 non-null   int64  
 10  type             5611 non-null   int64  
dtypes: float64(1), int64(7), object(3)
memory usage: 482.3+ KB
```

[6]:
```
df.isnull().sum()    ## Checking and counting Missing values
```

[6]:
```
Unnamed: 0           0
Title                0
Year                 0
Age               2446
IMDb              1161
Rotten Tomatoes   4601
Netflix              0
Hulu                 0
Prime Video          0
Disney+              0
type                 0
dtype: int64
```

[7]:
```
(df.isnull().sum()/df.shape[0])*100    ## checking missing values in percentage␣
 ↪by columns wise
```

[7]:
```
Unnamed: 0          0.000000
Title               0.000000
Year                0.000000
Age                43.592942
IMDb               20.691499
Rotten Tomatoes    81.999644
Netflix             0.000000
Hulu                0.000000
Prime Video         0.000000
Disney+             0.000000
```

```
type                0.000000
dtype: float64
```

[8]:
```
(df.isnull().sum().sum()/(df.shape[0]*df.shape[1]))*100  ## Total missing values in the data are 13%
```

[8]:
```
13.298553166669366
```

[9]:
```
df.drop(["Rotten Tomatoes"],axis=1,inplace=True)    ## dropping a columns
```

[10]:
```
df["Age"]=df["Age"].str.replace("+","")  ## replacing (+) with ("") in a particular column
```

[11]:
```
df.rename(columns={"Age":"Age+","IMDb":"IMDB"},inplace=True)   ## Renaming a column
```

[12]:
```
df["Age+"].mode()[0] ## using  mode function
```

[12]:
```
'16'
```

[13]:
```
df["Age+"]=df["Age+"].fillna(df["Age+"].mode()[0]) ## Handle missing values
```

[14]:
```
df.drop(["Unnamed: 0","type"],axis=1,inplace=True) ## Again dropping two columns
```

[15]:
```
df["IMDB"]=(df["IMDB"]*100/10)  # converting into float
```

[16]:
```
df["IMDB"].mean().round() # using  round function
```

[16]:
```
71.0
```

[17]:
```
df["IMDB"]=df["IMDB"].fillna(df["IMDB"].mean().round())   ### filling  mean value
```

[18]:
```
df["Age+"]=pd.to_numeric(df["Age+"],errors="coerce")  # Using to numeric function to change  the data type
```

[19]:
```
df["Age+"]=df["Age+"].fillna(df["Age+"].mean().round())  # filling mean value in round figure
```

[20]:
```
df["Age+"]=df["Age+"].astype(int)   ## changing data type
```

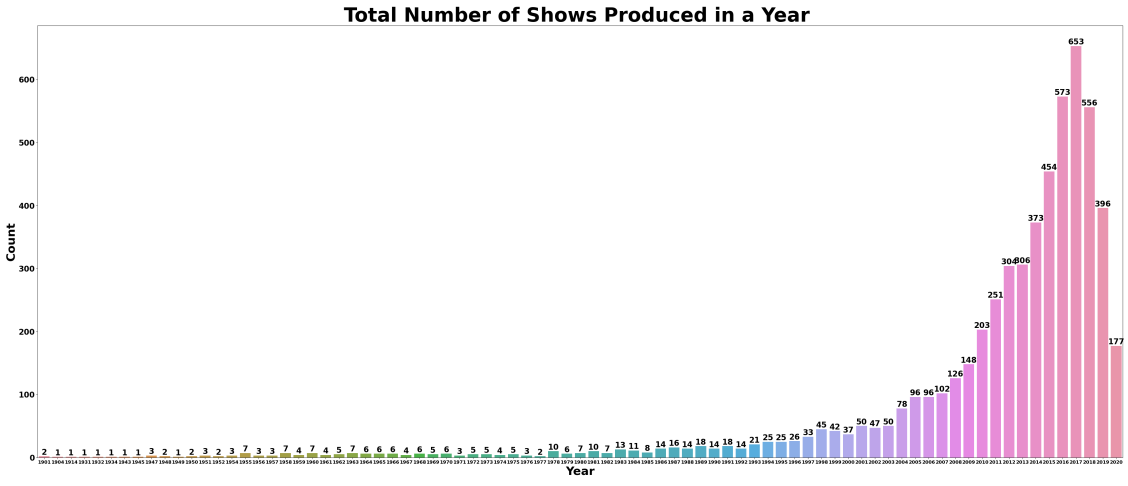
[21]:
```
## Counting ott_Platforms in a columns by using define function
def ott_platforms (x):
    if x["Disney+"]==1 and x["Netflix"]==0 and x["Hulu"]==0 and x["Prime Video"]==0:
        return "Disney+"
```

```
    if x["Disney+"]==0 and x["Netflix"]==0 and x["Hulu"]==0 and x["Prime␣
↪Video"]==1:
        return "Prime Video"
    if x["Disney+"]==0 and x["Netflix"]==1 and x["Hulu"]==0 and x["Prime␣
↪Video"]==0:
        return "Netflix"
    if x["Disney+"]==0 and x["Netflix"]==0 and x["Hulu"]==1 and x["Prime␣
↪Video"]==0:
        return "Hulu"
    return "Multiple"
```

[22]:
```
df["ott_platforms"]=df.apply(lambda row: ott_platforms(row),axis=1)
```

0.1.1 Shows produced in a year:

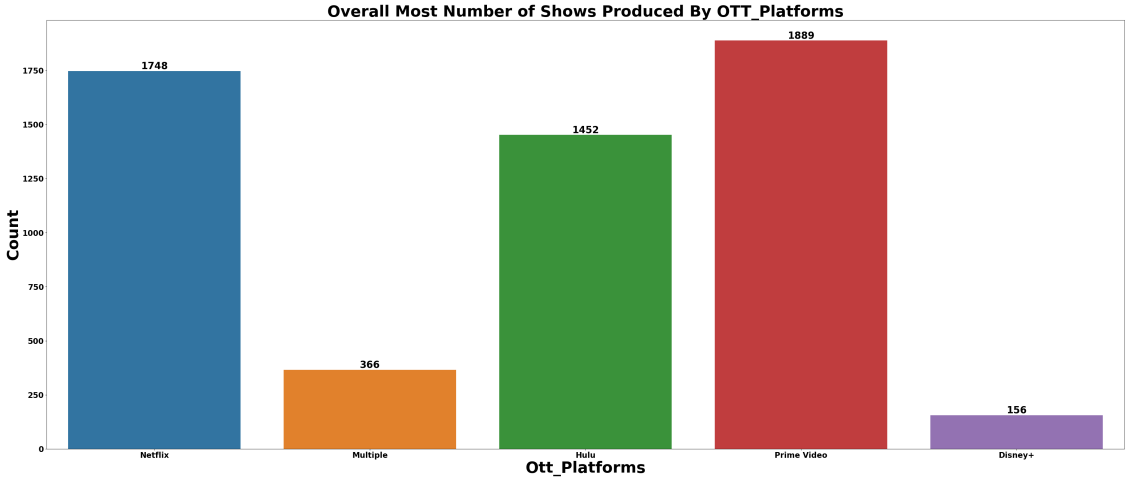
[23]:
```
## Plotting a bar graph Total year vs count
plt.figure(figsize=(50,20))
ax=sns.countplot(x=df["Year"],data=df)
for bars in ax.containers:
    ax.bar_label(bars,size=20,fontweight="bold")
plt.xlabel("Year",size=30,fontweight='bold')
plt.ylabel("Count",size=30,fontweight='bold')
plt.xticks(fontweight='bold',size=12)
plt.yticks(fontweight='bold',size=20)
plt.title("Total Number of Shows Produced in a Year ",size=50,fontweight='bold')
plt.show()
```



The above graph shows that the most number of TV shows were produced between 2010 and 2020, with 653 shows produced in 2017 alone.

[24]:
```
### Plotting a bar graph for ott_Platforms and it's count

plt.figure(figsize=(50,20))
az=sns.countplot(x=df["ott_platforms"],data=df)
for bars in az.containers:
    az.bar_label(bars,size=25,fontweight="bold")
plt.xlabel("Ott_Platforms",size=40,fontweight='bold')
plt.ylabel("Count",size=40,fontweight='bold')
plt.xticks(fontweight='bold',size=20)
plt.yticks(fontweight='bold',size=20)
plt.title("Overall Most Number of Shows Produced By␣
 ↪OTT_Platforms",size=40,fontweight='bold')
plt.show()
```



In the graph above shows that Prime Video Produced the maximum number of shows,followed by Netflix and Hullu.While, Disney+ produced fewer shows.

[25]:
```
df["ott_platforms"].value_counts()
```

[25]:
```
ott_platforms
Prime Video    1889
Netflix        1748
Hulu           1452
Multiple        366
Disney+         156
Name: count, dtype: int64
```

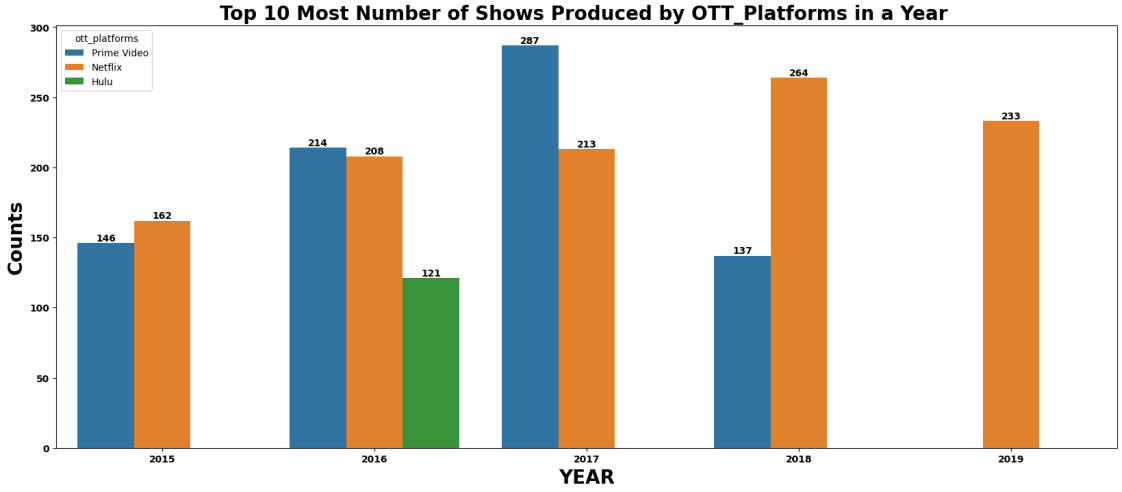
[26]:
```
ott_year=df.groupby(df["Year"])["ott_platforms"].value_counts().
 ↪reset_index(name="count").sort_values(by=["count"],ascending=False).head(10)
```

[27]:
```
ott_year
```

[27]:

	Year	ott_platforms	count
253	2017	Prime Video	287
258	2018	Netflix	264
263	2019	Netflix	233
248	2016	Prime Video	214
254	2017	Netflix	213
249	2016	Netflix	208
243	2015	Netflix	162
244	2015	Prime Video	146
259	2018	Prime Video	137
250	2016	Hulu	121

[28]:
```
## Plotting a bar graph
plt.figure(figsize=(20,8))
aq=sns.barplot(x="Year",y="count",data=ott_year,hue="ott_platforms")
for bars in aq.containers:
    aq.bar_label(bars,fontweight='bold')
plt.xlabel("YEAR",size=20,fontweight="bold")
plt.ylabel("Counts",size=20,fontweight="bold")
plt.xticks(fontweight="bold")
plt.yticks(fontweight="bold")
plt.title("Top 10 Most Number of Shows Produced by OTT_Platforms in a Year␣
 ↪",size=20,fontweight='bold')
plt.show()
```

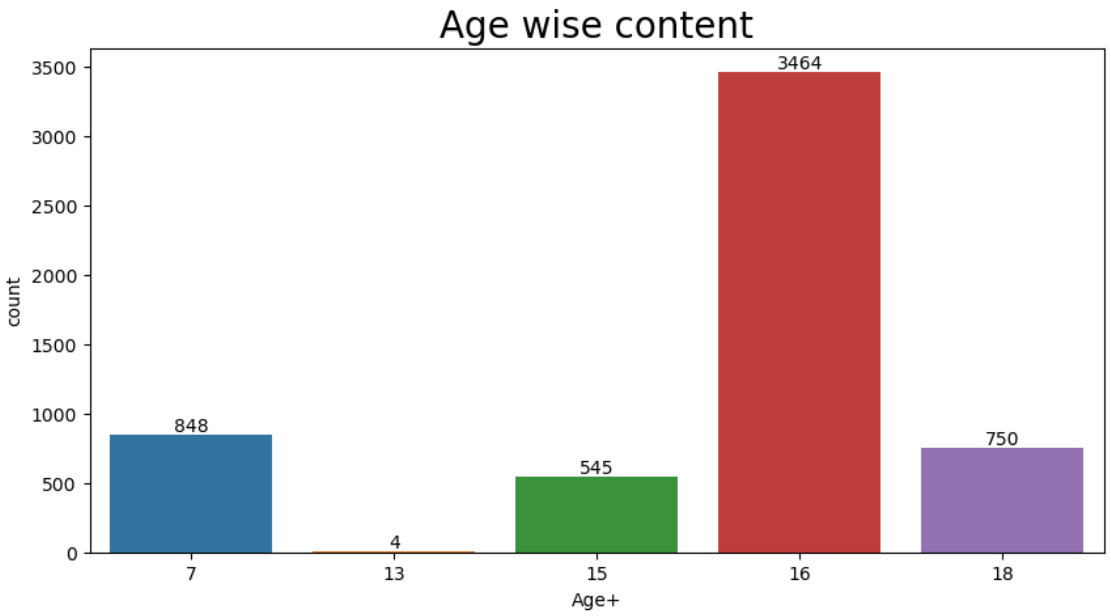


The above graph shows that the maximum number of TV Shows produced in 2017 by Prime Video,followed by Netflix in same year.

[29]:
```
df['Age+'].value_counts()
```

[29]:
```
Age+
16    3464
7      848
18     750
15     545
13       4
Name: count, dtype: int64
```

[30]:
```
#### Plotting a bar graph for Age and it's count
plt.figure(figsize=(10,5))
az=sns.countplot(x=df["Age+"],data=df)
for bars in az.containers:
    az.bar_label(bars)
plt.title("Age wise content",size=20)
plt.show()
```



In the above graph,the maximum number of TV shows were produced for audiences aged 16 and above.

[31]:
```
top_shows=df.sort_values(by=["IMDB"],ascending=False).head(10)
```

[32]:
```
top_shows
```

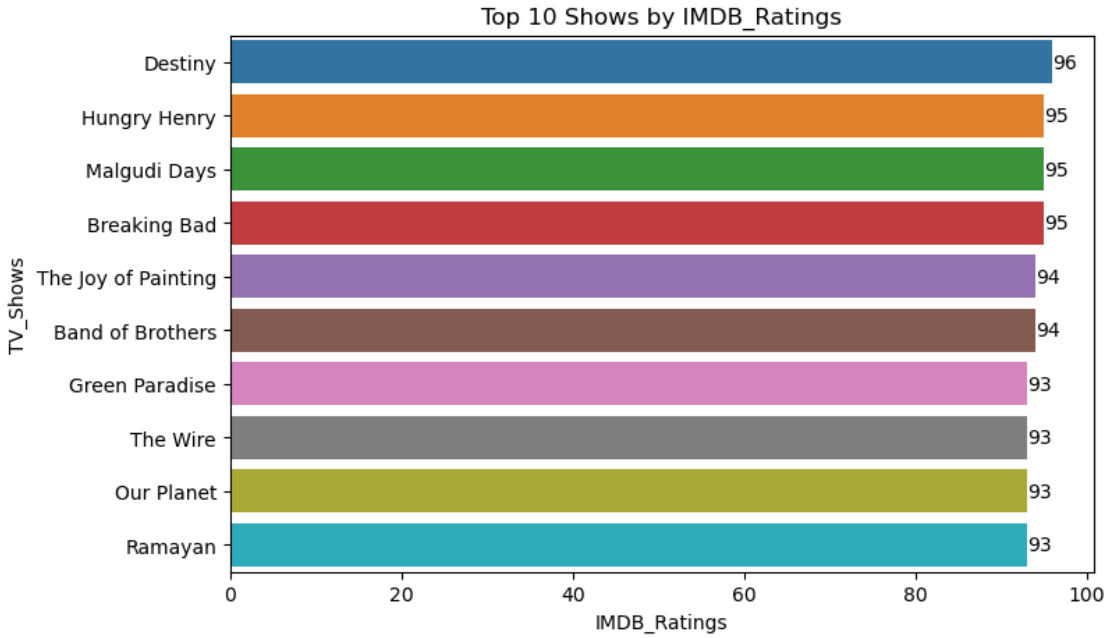
[32]:

	Title	Year	Age+	IMDB	Netflix	Hulu	Prime Video	\
3023	Destiny	2014	16	96.0	0	1	0	
3177	Hungry Henry	2014	16	95.0	0	1	0	
3747	Malgudi Days	1987	15	95.0	0	0	1	

```
0              Breaking Bad  2008    18  95.0        1     0            0
2365  The Joy of Painting  1983    15  94.0        0     1            1
3567     Band of Brothers  2001    18  94.0        0     0            1
4128       Green Paradise  2011    15  93.0        0     0            1
3566             The Wire  2002    18  93.0        0     0            1
91             Our Planet  2019     7  93.0        1     0            0
325               Ramayan  1987    15  93.0        1     0            0

      Disney+ ott_platforms
3023        0          Hulu
3177        0          Hulu
3747        0   Prime Video
0           0       Netflix
2365        0      Multiple
3567        0   Prime Video
4128        0   Prime Video
3566        0   Prime Video
91          0       Netflix
325         0       Netflix
```

[72]:
```
## Plotting a graph horizontally
plt.figure(figsize=(8,5))
zx=sns.barplot(x="IMDB",y="Title",data=top_shows)
for bars in zx.containers:
    zx.bar_label(bars)
plt.xlabel("IMDB_Ratings")
plt.ylabel("TV_Shows")
plt.title("Top 10 Shows by IMDB_Ratings")
plt.yticks(size=10)
plt.show()
```



In the graph above, the top TV show is Destiny by "IMDB" ratings.

[34]:
```
df
```

[34]:

	Title	Year	Age+	IMDB	Netflix	Hulu	\
0	Breaking Bad	2008	18	95.0	1	0	
1	Stranger Things	2016	16	88.0	1	0	
2	Money Heist	2017	18	84.0	1	0	
3	Sherlock	2010	16	91.0	1	0	
4	Better Call Saul	2015	18	87.0	1	0	
…	…	…	…	…	…	…	
5606	Tut's Treasures: Hidden Secrets	2018	16	71.0	0	0	
5607	Paradise Islands	2017	16	71.0	0	0	
5608	Wild Russia	2018	16	71.0	0	0	
5609	Love & Vets	2017	16	71.0	0	0	
5610	United States of Animals	2016	16	71.0	0	0	

	Prime Video	Disney+	ott_platforms
0	0	0	Netflix
1	0	0	Netflix
2	0	0	Netflix
3	0	0	Netflix
4	0	0	Netflix
…	…	…	…
5606	0	1	Disney+
5607	0	1	Disney+

```
5608            0        1       Disney+
5609            0        1       Disney+
5610            0        1       Disney+

[5611 rows x 9 columns]
```

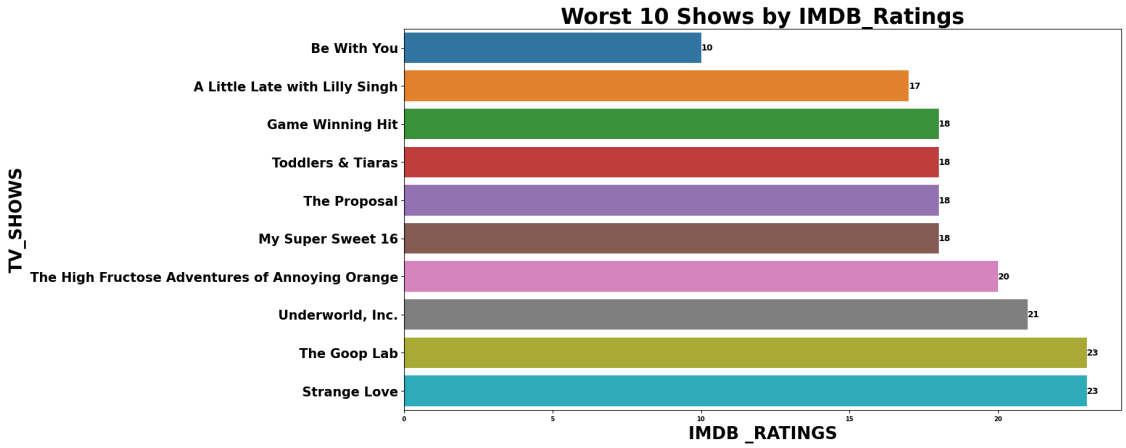
[35]:
```
worst_shows=df.sort_values(by=["IMDB"],ascending=True).head(10)
```

[36]:
```
worst_shows
```

[36]:
```
                                          Title  Year  Age+  IMDB  \
1807                                Be With You  2015    16  10.0
2999                A Little Late with Lilly Singh  2019    16  17.0
1818                             Game Winning Hit  2009    16  18.0
3104                           Toddlers & Tiaras  2009     7  18.0
3145                                The Proposal  2018    16  18.0
3144                           My Super Sweet 16  2005     7  18.0
3060  The High Fructose Adventures of Annoying Orange  2012     7  20.0
3292                             Underworld, Inc.  2015     7  21.0
1498                                The Goop Lab  2020    18  23.0
4551                                Strange Love  2005    16  23.0

      Netflix  Hulu  Prime Video  Disney+ ott_platforms
1807        1     0            0        0       Netflix
2999        0     1            0        0          Hulu
1818        1     0            0        0       Netflix
3104        0     1            1        0      Multiple
3145        0     1            0        0          Hulu
3144        0     1            0        0          Hulu
3060        0     1            1        0      Multiple
3292        0     1            0        0          Hulu
1498        1     0            0        0       Netflix
4551        0     0            1        0   Prime Video
```

[69]:
```
### Plotting a graph horizontally
plt.figure(figsize=(15,8))
zx=sns.barplot(x="IMDB",y="Title",data=worst_shows)
for bars in zx.containers:
    zx.bar_label(bars,fontweight='bold')
plt.xlabel("IMDB _RATINGS",size=20,fontweight='bold')
plt.ylabel("TV_SHOWS",size=20,fontweight='bold')
plt.title("Worst 10 Shows by IMDB_Ratings",size=25,fontweight='bold')
plt.xticks(size=7,fontweight='bold')
plt.yticks(size=15,fontweight='bold')
plt.show()
```



The above graph shows that "Be with you" is the worst Tv Show by "IMDB" ratings

0.1.2 Conclusion:

1- Maximum number of TV shows produced between 2010 and 2020. 2- 653 Tv shows produced in 2017 alone. 3- Prime video produced maximum number of TV shows around 1889 while Disney+ produced fewer which is around 156 . 4- In 2017 alone,Prime video produced maximum number of TV shows that is around 287,followed by Netflix which is around 213. 5- Maximum number of TV shows produced for the people aged 16 and above. 6- TV shows "Destiny" got highest IMDB ratings while "Be with you" Tv shows got worst ratings. 7- Target Audience are above 16 years.