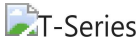
Comment Count Analysis for T-Series YouTube Dataset

Introduction to T-Series

Gulshan Kumar established the Indian music record label and motion picture production firm T-Series. T-Series also owns and runs the most popular YouTube channel, with over 248 million subscribers and 232 billion views overall as of September 7, 2022.



Comments on YouTube videos are important because:

They provide viewer feedback, which is valuable for satisfying your viewers and achieving success on YouTube.
They boost video engagement since they often lead back to other videos/channels that feature similar content.
They present an opportunity for positive community building and engagement.

The dataset used is the youtube t-series dataset with the following columns:

videoId: This column contains unique identifiers or codes for each video. These are alphanumeric strings that serve as identifiers for each video.
publishedTime: This column represents the date and time when each video was published. The date and time are in the format "DD-MM-YYYY HH:MM." durationSec: This column indicates the duration of each video in seconds. It shows how long each video lasts.
viewCount: This column displays the number of views each video has received. It represents the total count of how many times each video has been watched.
likeCount: This column shows the number of likes (or thumbs up) that each video has received. It represents the count of users who have liked the video.
dislikeCount: This column indicates the number of dislikes (or thumbs down) that each video has received. It represents the count of users who have disliked the video.

commentCount: This column displays the number of comments that have been posted on each video. It represents the count of user comments associated with each video.

MultiLinear Regression

Multiple linear regression (MLR) is a statistical technique that uses several explanatory variables to predict the outcome of a response varia ble. It is an extension of linear regression that uses just one explanatory variable. Multiple linear regression is used to estimate the relationship between two or more independent variables and one dependent variable.

In [1]:
```
#required libraries
import numpy as np
import pandas as pd
```

In [2]:
```
#load the dataset into pandas dataframe
data=pd.read_csv("C:\\Users\\soumy\\OneDrive\\Desktop\\T-Series.csv")
data
```

Out[2]:

	videoId	publishedTime	durationSec	viewCount	likeCount	dislikeCount	commentCount
0	O5X3bnTI1Go	17-05-2020 12:00	235	320360	13976.0	25966.0	3175
1	bTPt1voMr8g	17-05-2020 09:07	189	277845	33600.0	1645.0	3313
2	#NAME?	17-05-2020 10:39	376	62515	2334.0	373.0	291
3	g4Xa7WwOFw0	17-05-2020 05:30	140	117328	6043.0	604.0	685
4	tXIHXccXcCg	16-05-2020 08:31	20	996968	50636.0	2307.0	2712
...	...	...	...	...	...	...	...
831	G1D0n68lOS8	23-10-2019 08:30	243	135862	1874.0	128.0	146
832	tDHQPHCLd6c	23-10-2019 09:30	278	45847	525.0	98.0	83
833	B88D1SGN0Ig	23-10-2019 10:15	168	80220	898.0	136.0	100
834	frTgD-MC5pI	25-10-2019 05:30	153	2943938	39888.0	2534.0	1266
835	UHS1nCK3Smc	22-10-2019 10:36	23	581443	15215.0	1074.0	623

836 rows × 7 columns

In [3]:
```
data.head()
```

Out[3]:

	videoId	publishedTime	durationSec	viewCount	likeCount	dislikeCount	commentCount
0	O5X3bnTI1Go	17-05-2020 12:00	235	320360	13976.0	25966.0	3175
1	bTPt1voMr8g	17-05-2020 09:07	189	277845	33600.0	1645.0	3313
2	#NAME?	17-05-2020 10:39	376	62515	2334.0	373.0	291
3	g4Xa7WwOFw0	17-05-2020 05:30	140	117328	6043.0	604.0	685
4	tXIHXccXcCg	16-05-2020 08:31	20	996968	50636.0	2307.0	2712

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

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 836 entries, 0 to 835
Data columns (total 7 columns):
 #   Column         Non-Null Count  Dtype  
---  ------         --------------  -----  
 0   videoId        836 non-null    object 
 1   publishedTime  836 non-null    object 
 2   durationSec    836 non-null    int64  
 3   viewCount      836 non-null    int64  
 4   likeCount      834 non-null    float64
 5   dislikeCount   834 non-null    float64
 6   commentCount   836 non-null    int64  
dtypes: float64(2), int64(3), object(2)
memory usage: 45.8+ KB
```

Data Preprocessing

In [5]:
```
data.isnull().any()
```

Out[5]:
```
videoId          False
publishedTime    False
durationSec      False
viewCount        False
likeCount         True
dislikeCount      True
commentCount     False
dtype: bool
```

In [6]:
```
data['likeCount'].fillna(data['likeCount'].mean(),inplace=True)
data['dislikeCount'].fillna(data['dislikeCount'].mean(),inplace=True)
```

In [7]:
```
data.isnull().any()
```

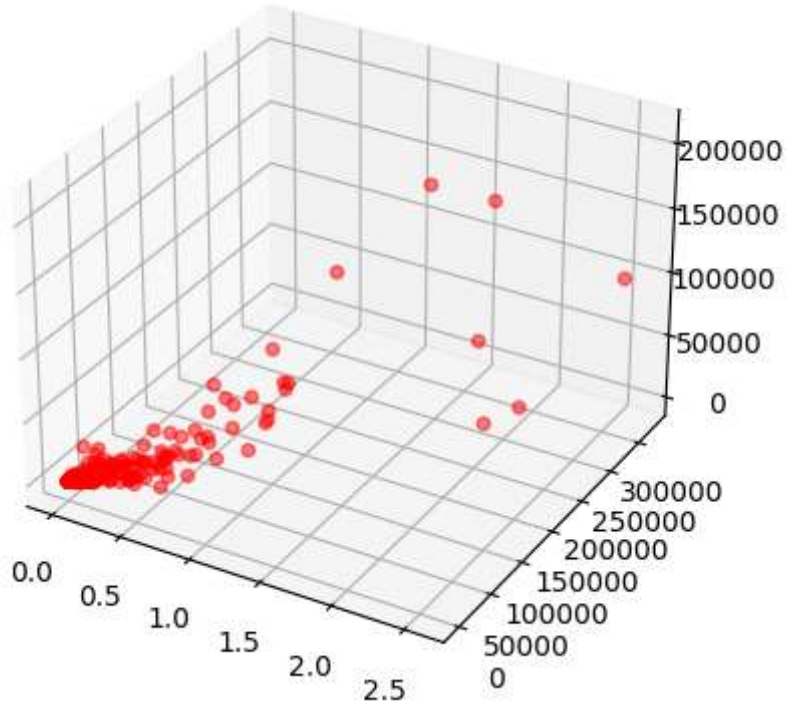
Out[7]:
```
videoId          False
publishedTime    False
durationSec      False
viewCount        False
likeCount        False
dislikeCount     False
commentCount     False
dtype: bool
```

Vizualizing the Data

In [8]:
```
import matplotlib.pyplot as plt
```

In [9]:
```
fig=plt.figure()
ax=fig.add_subplot(111,projection='3d')
ax.scatter(data['likeCount'],data['dislikeCount'],data['commentCount'],c='red',alpha=0.5)
```

Out[9]:
```
<mpl_toolkits.mplot3d.art3d.Path3DCollection at 0x1454f391e50>
```



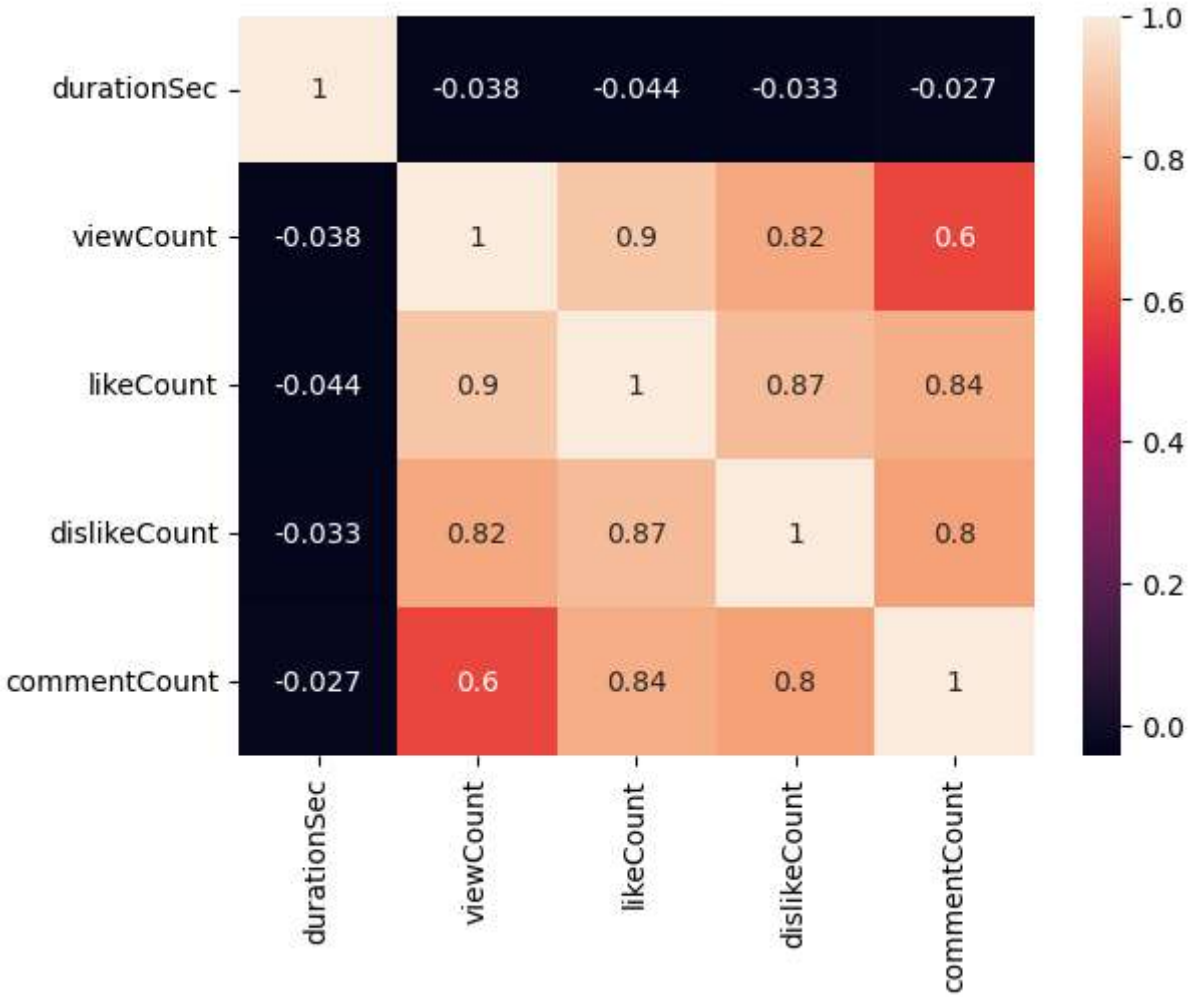
In [10]:
```
#correlation analysis
corr=data.iloc[:,2:7].corr()
corr
```

Out[10]:

	durationSec	viewCount	likeCount	dislikeCount	commentCount
durationSec	1.000000	-0.037876	-0.043883	-0.033498	-0.026743
viewCount	-0.037876	1.000000	0.900141	0.819763	0.598791
likeCount	-0.043883	0.900141	1.000000	0.872930	0.835181
dislikeCount	-0.033498	0.819763	0.872930	1.000000	0.802999
commentCount	-0.026743	0.598791	0.835181	0.802999	1.000000

In [11]:
```
import seaborn as sns
sns.heatmap(corr,annot=True)
```

Out[11]:
```
<Axes: >
```



Depenedent and Independent Variables

In [12]:
```
x=data.iloc[:,2:6].values
x
```

Out[12]:
```
array([[2.350000e+02, 3.203600e+05, 1.397600e+04, 2.596600e+04],
       [1.890000e+02, 2.778450e+05, 3.360000e+04, 1.645000e+03],
       [3.760000e+02, 6.251500e+04, 2.334000e+03, 3.730000e+02],
       ...,
       [1.680000e+02, 8.022000e+04, 8.980000e+02, 1.360000e+02],
       [1.530000e+02, 2.943938e+06, 3.988800e+04, 2.534000e+03],
       [2.300000e+01, 5.814430e+05, 1.521500e+04, 1.074000e+03]])
```

In [13]:
```
y=data.iloc[:,6:].values
y
```

Out[13]:
```
array([[  3175],
       [  3313],
       [   291],
       [   685],
       [  2712],
       [   736],
       [   780],
       [  2689],
       [  1523],
       [   876],
       [   241],
       [   894],
       [   842],
       [   822],
       [   332],
       [  1203],
       [  1303],
       [  1267],
       [ 16037],
       [  1062],
       [   930],
       [  2580],
       [   406],
       [  1698],
       [  1396],
       [  1022],
       [  1040],
       [  2089],
       [   467],
       [  1336],
       [   942],
       [  1194],
       [  1038],
       [  1346],
       [   884],
       [   333],
       [  3558],
       [ 24355],
       [   272],
       [   659],
       [  1199],
       [  1397],
```

```
[  1521],
[  1207],
[  4075],
[  1247],
[  2815],
[  8024],
[   462],
[   501],
[  1080],
[  1007],
[   119],
[   361],
[   624],
[   809],
[   723],
[  8834],
[   797],
[   459],
[   969],
[   551],
[   522],
[   752],
[   993],
[  1694],
[  1953],
[   999],
[  1810],
[   649],
[   676],
[   738],
[   537],
[   414],
[  1323],
[   792],
[   778],
[   756],
[   467],
[  1256],
[  9728],
[   882],
[  1045],
[  1120],
```

```
[  1120],
[   939],
[   568],
[   647],
[   618],
[100808],
[  1105],
[  5389],
[  1490],
[   620],
[  1305],
[ 11105],
[   457],
[   822],
[   932],
[   668],
[  1253],
[  1024],
[102947],
[  1085],
[  1075],
[   582],
[  1235],
[   205],
[   762],
[  1386],
[  1173],
[  1454],
[  7948],
[  5761],
[   885],
[  2108],
[   944],
[ 26970],
[   331],
[  2073],
[   936],
[  3470],
[  1093],
[  1032],
[   997],
[  1229],
```

```
[  1351],
[   266],
[   775],
[   553],
[   869],
[   938],
[   479],
[  1038],
[   758],
[   596],
[   998],
[ 52650],
[  1158],
[   647],
[  1267],
[   261],
[  1096],
[  7518],
[   791],
[  1638],
[   664],
[  3023],
[  2502],
[   977],
[  1078],
[  1152],
[   931],
[  1136],
[   801],
[  1649],
[  1366],
[  1041],
[   820],
[  2202],
[   721],
[   871],
[   952],
[  1121],
[  2363],
[  2020],
[   955],
[  1699],
```

```
[  2665],
[  1322],
[   779],
[   730],
[   329],
[   540],
[   502],
[   863],
[   736],
[   329],
[  1076],
[   249],
[   528],
[  4227],
[   512],
[130152],
[  1234],
[   772],
[   212],
[   701],
[   806],
[   395],
[   328],
[   195],
[   599],
[  3889],
[  2836],
[   506],
[  1395],
[  1311],
[   323],
[  3378],
[   752],
[   662],
[  1225],
[  1024],
[  9349],
[ 18267],
[  1735],
[   454],
[   465],
[  7443],
```

```
[210408],
[  1748],
[  1051],
[   443],
[   605],
[   446],
[   612],
[   269],
[   421],
[   229],
[   662],
[   209],
[ 20790],
[  2805],
[   490],
[   240],
[   828],
[ 32514],
[   294],
[   866],
[   538],
[  1878],
[  1752],
[   508],
[   275],
[   370],
[   468],
[   550],
[   557],
[   576],
[   200],
[  2180],
[   619],
[   996],
[   312],
[   129],
[   393],
[    69],
[  3414],
[  1087],
[  1289],
[   491],
```

```
[   544],
[  2176],
[   330],
[  6731],
[  1027],
[  2460],
[ 15764],
[   482],
[   818],
[   289],
[   630],
[   366],
[   741],
[   430],
[   638],
[ 70160],
[   592],
[  1689],
[  4638],
[  1342],
[  1262],
[   232],
[   648],
[   996],
[  8472],
[   935],
[   829],
[   704],
[   665],
[   603],
[   275],
[   567],
[   450],
[ 44301],
[   215],
[   216],
[   521],
[ 16248],
[  3429],
[  1025],
[   656],
[  5473],
```

```
[ 27652],
[   358],
[   641],
[   443],
[   435],
[   415],
[   151],
[   127],
[   164],
[   346],
[   368],
[   348],
[   480],
[   840],
[   288],
[  9089],
[   469],
[   125],
[  7298],
[  8842],
[   576],
[   244],
[   555],
[   493],
[   445],
[   367],
[   487],
[   139],
[   273],
[    92],
[  1255],
[   316],
[   482],
[   448],
[   132],
[   549],
[   328],
[  3424],
[  1437],
[   817],
[   287],
[   163],
```

```
[ 15189],
[  9802],
[  1558],
[   446],
[   289],
[   505],
[  8181],
[   587],
[   760],
[  1367],
[   318],
[  1609],
[   155],
[  9978],
[  5938],
[   788],
[  2675],
[   990],
[  1558],
[ 38982],
[  1946],
[   264],
[  2023],
[  1617],
[   878],
[ 13525],
[   646],
[  2032],
[   557],
[   176],
[  4766],
[ 10467],
[   650],
[  1872],
[   262],
[   519],
[ 11024],
[   237],
[   589],
[  8162],
[   475],
[   215],
```

```
[   729],
[   378],
[   306],
[   245],
[   992],
[   513],
[   328],
[  6841],
[   699],
[  3406],
[   473],
[  3063],
[ 33525],
[   591],
[   622],
[   382],
[   258],
[  3332],
[   536],
[   917],
[   348],
[  2469],
[  3637],
[   154],
[  1501],
[  1662],
[ 13896],
[   444],
[ 33686],
[   594],
[   619],
[  1319],
[   312],
[    75],
[    61],
[   104],
[    86],
[   180],
[   163],
[   169],
[   482],
[   263],
```

```
[  2155],
[  1817],
[   365],
[   399],
[   880],
[   324],
[   830],
[   744],
[ 17514],
[  1310],
[   214],
[  1330],
[ 16955],
[   663],
[   723],
[   956],
[   678],
[ 20056],
[  1366],
[  1510],
[  1024],
[   857],
[   810],
[   681],
[   336],
[   176],
[   995],
[   260],
[   475],
[   514],
[   669],
[   310],
[ 35454],
[   362],
[   412],
[   361],
[   429],
[   714],
[   796],
[  4285],
[   212],
[  3189],
```

```
[ 19991],
[  1707],
[   297],
[  2905],
[  2436],
[  2169],
[  1210],
[   221],
[   497],
[   588],
[   197],
[   648],
[   717],
[   258],
[  2656],
[   331],
[  2250],
[   348],
[  2595],
[  1182],
[   923],
[   887],
[   734],
[   612],
[   382],
[   704],
[  5779],
[  1920],
[  2271],
[   609],
[   348],
[   442],
[   387],
[   407],
[   532],
[  1372],
[  1378],
[ 66172],
[ 11287],
[   130],
[   947],
[  1144],
```

```
[   369],
[   837],
[   687],
[ 30886],
[  1342],
[  1166],
[   447],
[  2694],
[   955],
[   975],
[   797],
[  1194],
[   284],
[   519],
[   531],
[   488],
[   260],
[  1719],
[  1223],
[ 59420],
[   669],
[  1569],
[   405],
[ 12756],
[  4888],
[   346],
[  1142],
[   385],
[ 35744],
[ 15785],
[  1300],
[   944],
[ 19692],
[   753],
[    85],
[   419],
[   383],
[  1433],
[   722],
[   164],
[   271],
[  1525],
```

```
[  8182],
[  2515],
[  2912],
[  2154],
[   575],
[  2707],
[   676],
[   773],
[   251],
[  2001],
[ 10256],
[   553],
[  2863],
[   167],
[   507],
[   107],
[   698],
[  3182],
[  1134],
[   137],
[   432],
[   821],
[   461],
[   223],
[   638],
[   316],
[   159],
[   190],
[   546],
[   399],
[   852],
[   321],
[   350],
[  9496],
[   168],
[   129],
[   536],
[  1259],
[  3194],
[   563],
[  1073],
[   196],
```

```
[   812],
[  2897],
[  1375],
[  1001],
[   178],
[  2052],
[  1681],
[  4658],
[ 13874],
[ 15343],
[  1096],
[   983],
[ 41063],
[   190],
[   213],
[   208],
[    94],
[    85],
[    69],
[    83],
[  2259],
[   252],
[   522],
[  1012],
[   264],
[   145],
[   433],
[   422],
[   677],
[   208],
[  4092],
[  4517],
[  2563],
[   132],
[   507],
[   895],
[   532],
[   470],
[   207],
[   561],
[  3801],
[   399],
```

```
[   228],
[   169],
[  1672],
[   474],
[   640],
[  2939],
[   763],
[ 21485],
[  6820],
[   779],
[   297],
[   968],
[  1108],
[  3164],
[  1851],
[   106],
[    81],
[   445],
[   184],
[  3893],
[ 43933],
[  8909],
[ 18231],
[   935],
[   590],
[   161],
[   829],
[   344],
[   600],
[   530],
[   329],
[  7971],
[ 15821],
[   411],
[   160],
[    99],
[ 20933],
[   510],
[   181],
[   213],
[   261],
[  1401],
```

```
[   861],
[ 25354],
[    94],
[   669],
[   488],
[   244],
[   103],
[   329],
[  3674],
[   676],
[   554],
[   391],
[   311],
[  3021],
[  1447],
[ 40064],
[   377],
[   447],
[   241],
[ 86762],
[   120],
[   142],
[   232],
[   431],
[  1091],
[  1235],
[  3913],
[104673],
[   831],
[  1917],
[   687],
[  4978],
[   388],
[   371],
[   656],
[  2292],
[   387],
[  1088],
[   674],
[  2421],
[  1928],
[   342],
```

```
[   255],
[  1264],
[  1916],
[   122],
[   290],
[ 20060],
[  1365],
[   299],
[  5203],
[  1070],
[   306],
[ 17690],
[  4578],
[    89],
[   148],
[    90],
[   366],
[ 13571],
[ 15238],
[  1812],
[  1613],
[  7057],
[ 12903],
[  3877],
[   672],
[   698],
[   502],
[   309],
[   218],
[   538],
[   374],
[   247],
[  4113],
[  9210],
[   434],
[   202],
[  4534],
[   207],
[   161],
[    81],
[   228],
[   213],
```

```
[   133],
[ 11956],
[  1227],
[  1490],
[  1715],
[   396],
[   418],
[  1317],
[   798],
[   676],
[   373],
[  1351],
[ 27771],
[  2398],
[   637],
[   322],
[   449],
[  1880],
[  1101],
[  1259],
[  2398],
[ 23707],
[   531],
[   331],
[   208],
[   105],
[   138],
[   247],
[   568],
[   454],
[   286],
[   370],
[   132],
[    83],
[  5896],
[   373],
[  3049],
[   588],
[   110],
[   566],
[   257],
[ 15498],
```

```
[    76],
[   256],
[    66],
[   133],
[  1381],
[  7310],
[  2007],
[   337],
[   172],
[   617],
[  1227],
[  9805],
[   573],
[ 16552],
[   596],
[   112],
[   216],
[  1074],
[   642],
[   731],
[ 16940],
[   904],
[   742],
[   220],
[   610],
[   366],
[   298],
[   115],
[   213],
[   202],
[   239],
[   136],
[   193],
[   146],
[    83],
[   100],
[  1266],
[   623]], dtype=int64)
```

In [14]:
```
x.shape
```

Out[14]: (836, 4)

In [15]:
```
y.shape
```

Out[15]: (836, 1)

Split this data into training and testing data set

In [16]:
```
from sklearn.model_selection import train_test_split
```

In [17]:
```
x_train, x_test, y_train, y_test = train_test_split(x,y,test_size = 0.2,random_state = 0)
```

In [18]:
```
x_train.shape
```

Out[18]: (668, 4)

In [19]:
```
x_test.shape
```

Out[19]: (168, 4)

In [20]:
```
y_train.shape
```

Out[20]: (668, 1)

In [21]:
```
y_test.shape
```

Out[21]: (168, 1)

Build the model

In [22]:
```
from sklearn.linear_model import LinearRegression
```

In [23]:
```
mlr = LinearRegression()
mlr.fit(x_train, y_train)
```

Out[23]:

```
LinearRegression()
```

Test the model

In [24]:

```
y_pred = mlr.predict(x_test)
y_pred
```

Out[24]:
```
array([[   238.91251038],
       [  1422.0023255 ],
       [   299.81434734],
       [   836.9545628 ],
       [  4670.69496035],
       [  1174.52247327],
       [   246.74220261],
       [  1027.65987267],
       [  1039.84569115],
       [ 21263.51246526],
       [  1261.76179968],
       [   915.94504225],
       [   495.06629344],
       [  1814.03707863],
       [   239.00785889],
       [   720.14415727],
       [   566.84111325],
       [   651.64516928],
       [   342.86220443],
       [   273.45036346],
       [   447.15673851],
       [   663.01102196],
       [  1084.97894865],
       [  7178.84363682],
       [  3763.43105188],
       [   223.96240972],
       [   373.17996413],
       [  1051.50538957],
       [   491.81924464],
       [  1212.07507672],
       [  1085.96432679],
       [   412.52545909],
       [119222.23451575],
       [   561.75709584],
       [  1027.40339944],
       [   872.25232969],
       [  1447.95187503],
       [  1166.59157904],
       [  2838.27581629],
       [ 37224.76709023],
       [   229.28360577],
       [   345.89377095],
```

```
[   437.17313601],
[ -8124.06691363],
[  1107.63019377],
[   382.48616899],
[   522.72709265],
[ -4060.4717921 ],
[   367.14160596],
[   646.84696407],
[   493.05695076],
[  4602.14718531],
[   286.74090216],
[   716.52311004],
[  4621.17340918],
[  1518.84464686],
[   657.86354699],
[  1616.71577801],
[   620.89592389],
[ -9165.26677838],
[  1727.85773572],
[ 55721.09895321],
[   213.88613829],
[   656.0025609 ],
[   494.70003154],
[   809.55616222],
[   264.47284311],
[   417.61376737],
[   644.92762263],
[  1840.76624596],
[   260.94226466],
[   467.35860432],
[   835.81213625],
[   310.48445603],
[   703.46978852],
[  2182.26339994],
[   474.45352519],
[   727.3245829 ],
[  1854.90563389],
[   325.61707946],
[  1024.53420623],
[ 43519.59819123],
[ 54297.39432816],
[   390.79608146],
```

```
[ 14316.88741531],
[  7956.28446687],
[  1915.5590053 ],
[  2978.36237947],
[   805.93102501],
[   405.74428169],
[  1353.07250856],
[  1248.5121617 ],
[   413.75681709],
[   458.49497936],
[  1013.85662428],
[  1310.67240828],
[   663.45059694],
[   491.67049413],
[   625.76008557],
[  1610.18830419],
[   530.4145344 ],
[  1093.31674392],
[   795.36785776],
[   626.41054977],
[  5018.51441311],
[   343.51535509],
[  1114.85706594],
[  2192.25762389],
[  1642.63893502],
[  9398.21376307],
[   276.89185132],
[  1011.55600017],
[   470.5877086 ],
[   644.40424352],
[ 67227.66536009],
[   806.50312702],
[   627.75837186],
[   446.94714669],
[  1896.31166772],
[   581.41909822],
[  1113.87698649],
[  2306.4977046 ],
[ -1074.40357863],
[   578.32005857],
[   735.07814967],
[  4017.42812694],
```

```
[  6249.93148898],
[   409.90949097],
[   525.54632765],
[  1419.28966831],
[   449.11070019],
[  4160.49521017],
[  1532.10574814],
[  3601.22628032],
[   507.07538859],
[  3909.15828766],
[  2646.72266218],
[   392.10147044],
[ 29894.97627712],
[   950.65342102],
[  3274.19706784],
[ 64939.56682082],
[  1254.3038573 ],
[   323.64304993],
[   597.81647987],
[ 21984.68015511],
[   426.01379091],
[   507.4230885 ],
[  4320.71996868],
[   976.23752639],
[  1865.73648169],
[  5270.21031079],
[   715.02630731],
[   394.46412053],
[  1503.73313289],
[   343.68037769],
[   400.43356634],
[  1608.17250246],
[  3487.26852388],
[   384.83418832],
[   591.01064039],
[  6102.40204943],
[   361.86120999],
[   261.64461771],
[   505.99388221],
[   441.9620236 ],
[   480.01226012],
[   325.76676289]])
```

In [25]:
```
y_test
```

Out[25]:
```
array([[   104],
       [   880],
       [   161],
       [  1267],
       [  2250],
       [   820],
       [   125],
       [   729],
       [   600],
       [ 11024],
       [  1523],
       [  1096],
       [   405],
       [  1266],
       [   130],
       [  1024],
       [   373],
       [   538],
       [   228],
       [   127],
       [    99],
       [   612],
       [  1501],
       [  8909],
       [  6731],
       [    66],
       [   261],
       [   791],
       [   640],
       [   672],
       [  1194],
       [   290],
       [210408],
       [   329],
       [   656],
       [   553],
       [  1087],
       [   704],
       [  2089],
       [ 27771],
       [   107],
       [   220],
```

```
[   480],
[  3889],
[  1510],
[   332],
[   329],
[  4517],
[   252],
[   809],
[   468],
[  2656],
[   200],
[  1085],
[  2176],
[  1880],
[   391],
[  1091],
[   605],
[ 15821],
[   756],
[ 86762],
[    69],
[   590],
[   488],
[   752],
[    83],
[   454],
[   469],
[  1262],
[    85],
[   365],
[   676],
[   324],
[   369],
[  1024],
[   446],
[   566],
[  1342],
[   331],
[   596],
[ 40064],
[ 44301],
[   223],
```

```
[  6820],
[ 13874],
[  2169],
[  3424],
[  1199],
[   366],
[  1108],
[  5761],
[   213],
[   348],
[   952],
[  1638],
[   642],
[   442],
[   544],
[  1365],
[   522],
[  1256],
[   744],
[   863],
[  2694],
[   202],
[   589],
[  1319],
[  1259],
[ 15343],
[    94],
[   857],
[   493],
[   939],
[ 66172],
[   792],
[   449],
[   321],
[  1225],
[   367],
[  1142],
[  3313],
[   136],
[   736],
[   638],
[  5203],
```

```
[  2863],
[   184],
[   555],
[  1366],
[  2675],
[  1351],
[  1395],
[  2398],
[   387],
[  3063],
[  1078],
[  2469],
[ 33525],
[   646],
[  2436],
[ 59420],
[   788],
[   154],
[   659],
[ 16037],
[   258],
[   445],
[  2259],
[  1080],
[  1381],
[  3674],
[  1093],
[   190],
[  1672],
[   241],
[   348],
[  1810],
[  2052],
[   218],
[   411],
[  9349],
[   132],
[    94],
[   491],
[   291],
[   239],
[   160]], dtype=int64)
```

Checking r2_score (the evaluation metric)

In [26]:
```
from sklearn.metrics import r2_score
```

In [27]:
```
acc=r2_score(y_test,y_pred)
acc
```

Out[27]:
```
0.8286560931566199
```

Predicting the comment count by giving some new value as input to the model

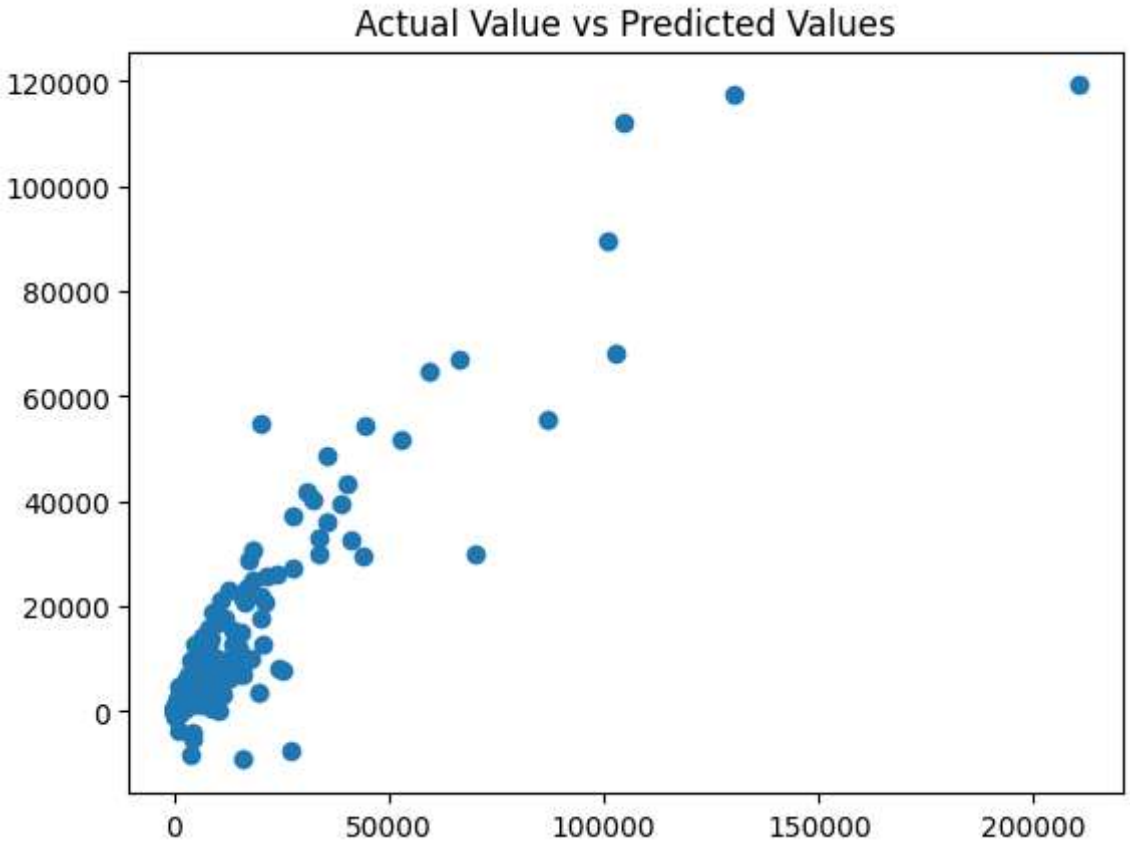
In [28]:
```
mlr.predict([[2,300,23,2]])
```

Out[28]:
```
array([[171.64014801]])
```

Visualizing Actual Y(commentCount) with Predicted Y(commentCount)

In [29]:
```
plt.scatter(y,mlr.predict(x))
plt.title("Actual Value vs Predicted Values")
```

Out[29]:
```
Text(0.5, 1.0, 'Actual Value vs Predicted Values')
```



In conclusion, an 82% R2 score indicates a reasonably good fit of MLR model to the data, that is significant portion of the variation in comment counts can be explained by the selected independent variables. Additionally, the accuracy of model can be assessed using cross validation, because there may be other unmeasured factors that influence comment counts on YouTube videos.

In [ ]: