REPORT

YOUTUBE VIDEOS AND CHANNELS METADATA

Submitted By

Aarathy S P

CB.PS.P2ASD22009

IST M.SC ASDA

DEPT.MATHEMATICS

aarathysureshp@gmail.com

Submitted To

Dr Praveen I

Assistant Professor

DEPT.MATHEMATICS

YouTube Videos and Channel Metadata

OBJECTIVE:

The objective of this project is to build a deep understanding of how the YouTube videos are and to analyze the statistical relation between videos (Most popular data science YouTube channel and it's Statistical analysis over different channels based on total likes, dislikes, comment counts). This data offers an intriguing look into consumer behavior as we can explore what drives people to watch specific videos at certain times or appreciate certain channels more than others.

The data was scraped from ToolDatabase website and analysed using python. The different plots used to analyse the data are bar plot, line graph, pie chart and heatmap. The various python packages used are numpy, pandas, matplotlib.pyplot ,seaborn and tkinter.

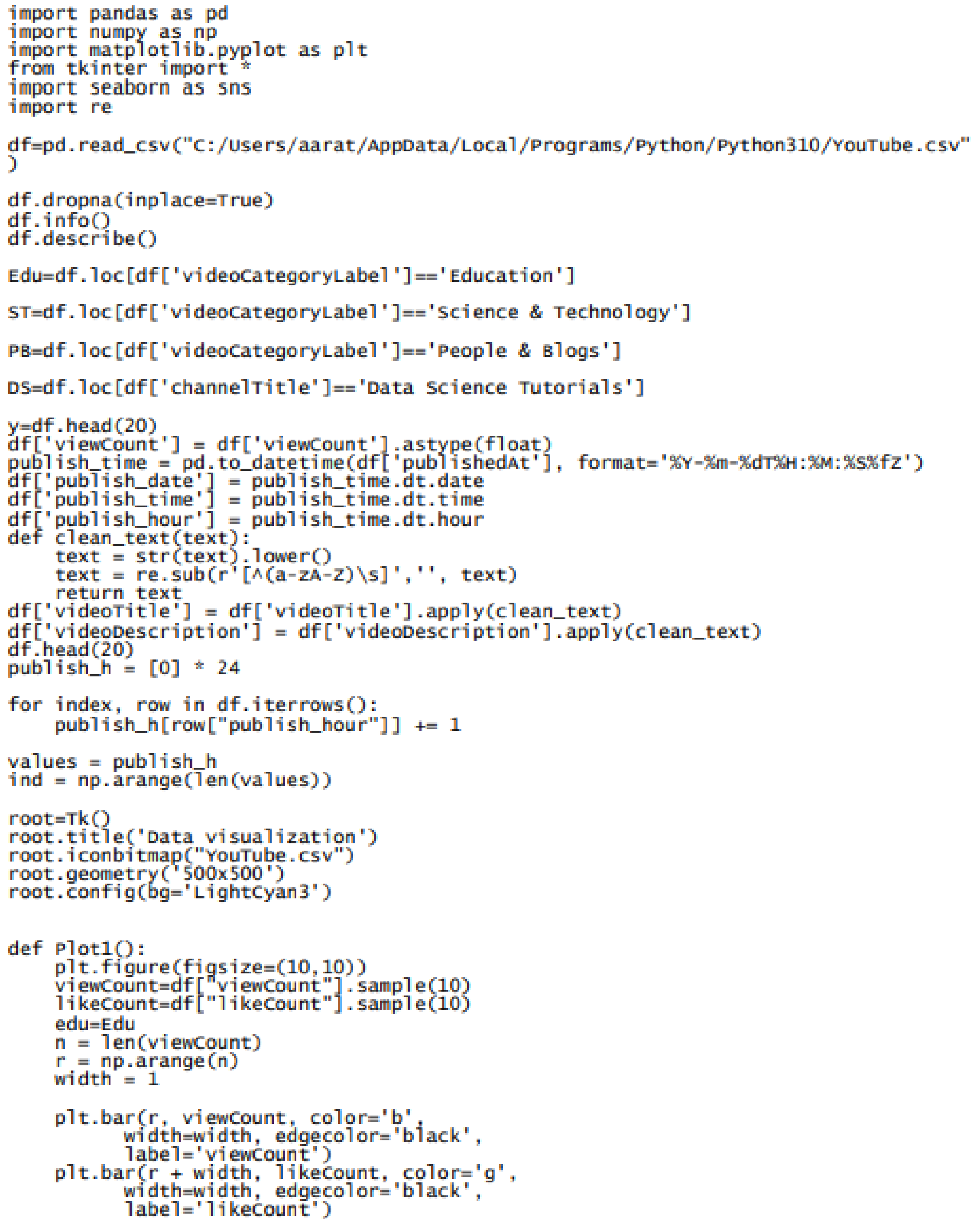
The no. of Variables in dataset 21 . They are:

1. **channelId** - youtube channel id.

2. **channelTitle -** youtube channel name.
3. **videoId** – Video ID
4. **publishedAt -** Published date. Yyyy – mm – dd Thh : mm : ss Z format.
5. **publishedAtSQL** – Published At
6. **videoTitle** – Video Title
7. **videoDescription** – Video Description
8. **videoCategoryId** – Video Category ID (28 ,44,…)
9. **videoCategoryLabel** – Video Category (Education, Science and technology,…)
10. **duration** – Video Duration
11. **durationSec** – Video Duration in Seconds
12. **dimension** – Video Dimension
13. **definition** – Video Definition (Quality)
14. **caption**
15. **thumbnail_maxres -**Video thumbnail url
16. **licensedContent** – Licensed Content
17. **viewCount** – Number of Views
18. **likeCount** – Number of Like
19. **dislikeCount** – Number of Dislike
20. **favoriteCount** – Favorite Count
21. **commentCount** – Comment Count

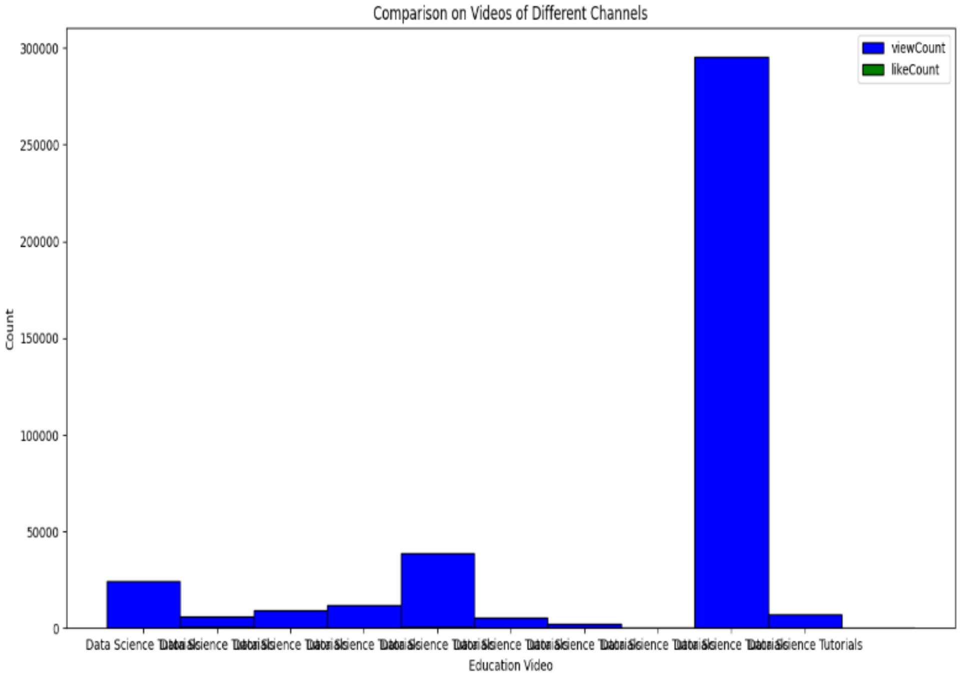
Variables such as channelId, videoDescription, etc are Categorical Variables and describe about the videos' unique characters.

SYNTAX FOR ANALYSIS THE DATASET USING IDLE



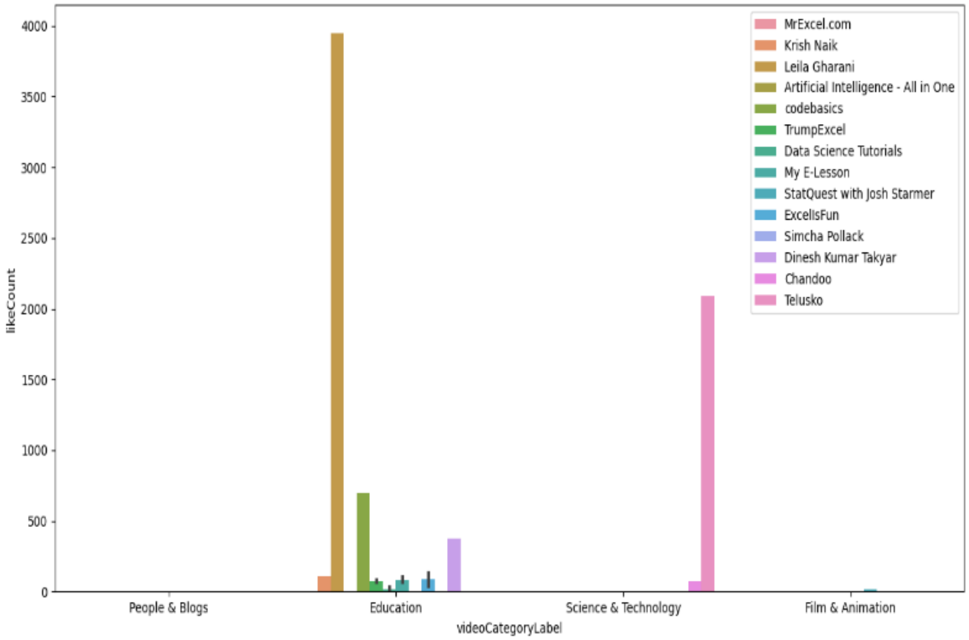
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from tkinter import *
import seaborn as sns
import re

df=pd.read_csv("C:/Users/aarat/AppData/Local/Programs/Python/Python310/YouTube.csv"
)

df.dropna(inplace=True)
df.info()
df.describe()

Edu=df.loc[df['videoCategoryLabel']=='Education']

ST=df.loc[df['videoCategoryLabel']=='Science & Technology']

PB=df.loc[df['videoCategoryLabel']=='People & Blogs']

DS=df.loc[df['channelTitle']=='Data Science Tutorials']

y=df.head(20)
df['viewCount'] = df['viewCount'].astype(float)
publish_time = pd.to_datetime(df['publishedAt'], format='%Y-%m-%dT%H:%M:%S%fZ')
df['publish_date'] = publish_time.dt.date
df['publish_time'] = publish_time.dt.time
df['publish_hour'] = publish_time.dt.hour
def clean_text(text):
    text = str(text).lower()
    text = re.sub(r'[^(a-zA-Z)\s]','', text)
    return text
df['videoTitle'] = df['videoTitle'].apply(clean_text)
df['videoDescription'] = df['videoDescription'].apply(clean_text)
df.head(20)
publish_h = [0] * 24

for index, row in df.iterrows():
    publish_h[row["publish_hour"]] += 1

values = publish_h
ind = np.arange(len(values))

root=Tk()
root.title('Data visualization')
root.iconbitmap("YouTube.csv")
root.geometry('500x500')
root.config(bg='LightCyan3')


def Plot1():
    plt.figure(figsize=(10,10))
    viewCount=df["viewCount"].sample(10)
    likeCount=df["likeCount"].sample(10)
    edu=Edu
    n = len(viewCount)
    r = np.arange(n)
    width = 1

    plt.bar(r, viewCount, color='b',
          width=width, edgecolor='black',
          label='viewCount')
    plt.bar(r + width, likeCount, color='g',
          width=width, edgecolor='black',
          label='likeCount')
    plt.xlabel("Education Video")
    plt.ylabel("Count")
    plt.title("Comparison on Videos of Different Channels")


    plt.xticks(r,df["channelTitle"].tail(10))
    plt.legend()

    plt.show()

l1= Label(root,text="Plot",font = ("Lucida Console", 14),bg='LightCyan3').grid(row=
0,column= 0)
b1 = Button(root,text="Education Videos",
command=Plot1,font=('Courier',12)).grid(row=0 ,column=1 )

def Plot2():
    plt.figure(figsize=(15,15))

sns.barplot(x="videoCategoryLabel",y="likeCount",hue="channelTitle",data=df.sample(
n=20))
    plt.legend(loc="upper right")
    plt.show()
l2= Label(root,text="Videos By Channels",font = ("Lucida Console",
14),bg='LightCyan3').grid(row= 1,column=0 )
b2= Button(root,text="Type of Videos", command=Plot2,font=('Courier',12)).grid(row=
1,column= 1)

df['like_rate'] =  df['likeCount'] / df['viewCount'] * 100
df['dislike_rate'] =  df['dislikeCount'] / df['viewCount'] * 100
df['comment_rate'] =  df['commentCount'] / df['viewCount'] * 100

df = df.replace([np.inf, -np.inf], np.nan)
def plot3():
    plt.figure(figsize = (9,6))
    g1 = sns.distplot(df ['dislike_rate'], color='red',hist=False, label="Dislike")
    g1 = sns.distplot(df ['like_rate'], color='green',hist=False, label="Like")
    g1 = sns.distplot(df ['comment_rate'],hist=False,label="Comment")
    g1.set_title('CONVERT RATE DISTRIBUITION', fontsize=16)
    plt.xlabel('rate')
    plt.legend()
    plt.show()


l3= Label(root,text="Distribution",font = ("Lucida Console",
14),bg='LightCyan3').grid(row=2 ,column=0 )
b3= Button(root,text="Rate Distribution ",
command=plot3,font=('Courier',12)).grid(row=2 ,column=1 )
# Creating new plot
def plot4():
    fig = plt.figure(figsize=(20,10))
    ax = fig.add_subplot(111)
    ax.yaxis.grid()
    ax.xaxis.grid()
    bars = ax.bar(ind, values)

# Sampling of Colormap
    for i, b in enumerate(bars):
        b.set_color(plt.cm.viridis((values[i] - min(values))/(max(values)-
min(values))))

        plt.ylabel('Number of published videos', fontsize=20)
        plt.xlabel('Time of publishing', fontsize=20)
        plt.title('When most of the videos are published?', fontsize=35,
fontweight='bold')
        plt.xticks(np.arange(0, len(ind), len(ind)/6), [0, 4, 8, 12, 16, 20])
    plt.show()
l4= Label(root,text="BarPlot",font = ("Lucida Console",
14),bg='LightCyan3').grid(row= 3,column=0 )
b4= Button(root,text="Best time to publish Video ",
command=plot4,font=('Courier',12)).grid(row=3 ,column= 1)

def plot5():
    plt.figure(figsize = (12,6))

    plt.subplot(221)
    g1 = sns.distplot(df['viewCount'])
    g1.set_title("VIEWS DISTRIBUITION", fontsize=16)

    plt.subplot(224)
    g2 = sns.distplot(df['likeCount'],color='green')
    g2.set_title('LIKES DISTRIBUITION', fontsize=16)

    plt.subplot(223)
    g3 = sns.distplot(df['dislikeCount'], color='r')
    g3.set_title("DISLIKES DISTRIBUITION", fontsize=16)

    plt.subplot(222)
    g4 = sns.distplot(df['commentCount'])
    g4.set_title("COMMENTS DISTRIBUITION", fontsize=16)

    plt.subplots_adjust(wspace = 0.2, hspace = 0.4,top = 0.9)

    plt.show()


l5= Label(root,text="Distribution",font = ("Lucida Console",
14),bg='LightCyan3').grid(row= 4,column= 0)
b5= Button(root,text="Distribuition of Variables ",
command=plot5,font=('Courier',12)).grid(row= 4,column=1 )


def plot6():
    plt.figure(figsize = (10,8))

    #Let's verify the correlation of each value
    sns.heatmap(df.corr(), annot=True)
    plt.show()

l6= Label(root,text="Correlation Graph",font = ("Lucida Console",
14),bg='LightCyan3').grid(row= 5,column=0 )
b6= Button(root,text="Correlation ", command=plot6,font=('Courier',12)).grid(row=
5,column=1 )
def plot7():
    plt.figure(figsize = (10,8))
    z=df['videoCategoryLabel'].unique()
    y=df['videoCategoryId'].unique()
    plt.pie(y,labels=z)
    plt.show
l7= Label(root,text="Percentage of Video Categories",font = ("Lucida Console",
14),bg='LightCyan3').grid(row= 6,column=0 )
b7= Button(root,text="Pie Chart ", command=plot7,font=('Courier',12)).grid(row=6
,column= 1)
root.mainloop()
```

PLOTS



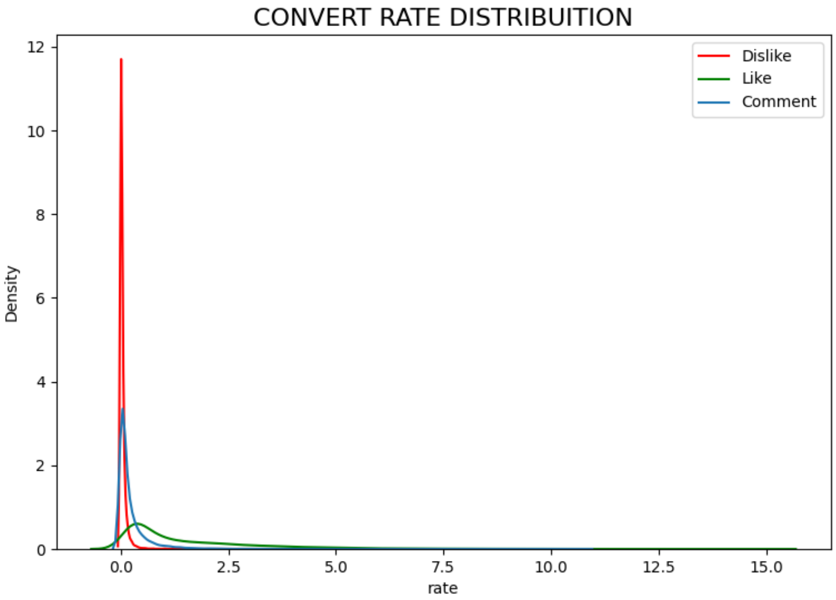
Fig(1)

The Fig (1) plot the likes and view count of videos of education category from a Specified channel(since the sample is taken, channel name changes for every time we run the syntax).Graph Shows that view counts are high but like counts are low .We can conclude that the videos are not very useful.



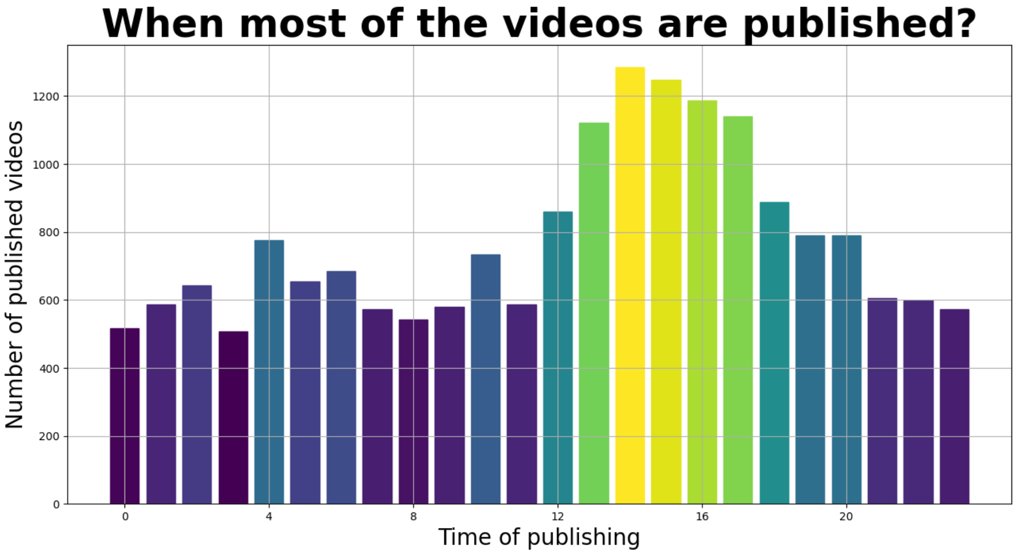
Fig(2)

Fig(2) plots about different type of videos uploaded by different channels and like counts they got. Here data is taken as a sample and most of the sample , bar is usually high for Education Videos and science and Technology Videos uploaded by Artificial Intelligence has peak value.



Fig(3)

Fig(3) analyse the percent of likes, dislikes, comment by each category to discover what category have the highest engagement in the data.



Fig(4)

Fig (4) gives us best time when most of the videos are published, which is around 14hrs. (which implies that viewers watch videos during their leisure time)

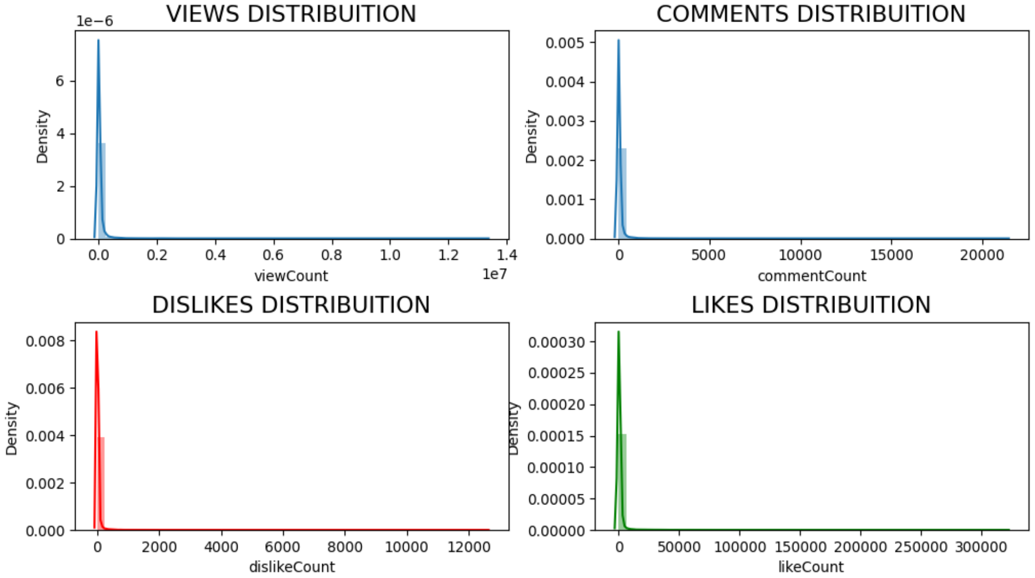
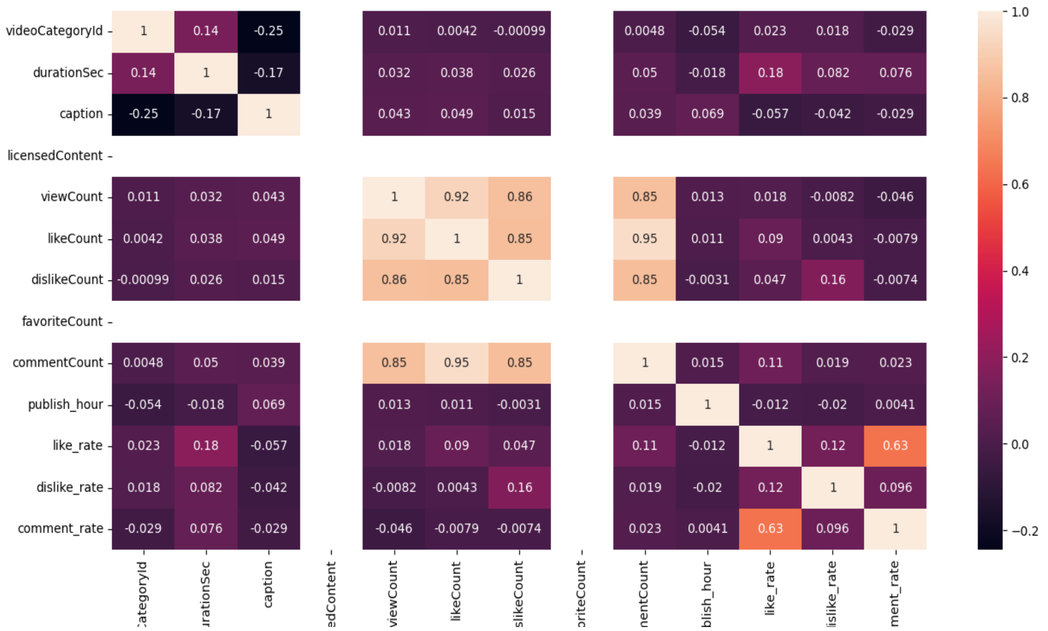


Fig (5)

Fig (5) shows the distribution of the main parameters (Views, likes, comments, dislikes) in the given dataset of Data Science Videos. *A Distplot or distribution plot, **depicts the variation in the data distribution**. The distribution plot is suitable for comparing range and distribution for groups of numerical data. The distribution plot is not relevant for detailed analysis of the data as it deals with a summary of the data distribution Seaborn Distplot represents the overall distribution of continuous data variables.*



Fig(6)

Fig (6) shows the heatmap for every variable in the dataset to describe the relation between variables. *A heatmap is a two-dimensional graphical representation of data where the individual values that are contained in a matrix are represented as colors. In this plot the color varies from shade of white for strongly correlated to deep magenta or approximately black for weakly correlated variables. The values in each box represent the correlation coefficient. Correlation coefficient ranges from -1 to +1. If the coefficient value lies between 0.50 and 1, then it is said to be a positively correlated and if it is between -0.05 and -1, then it is said to be negatively correlated.*

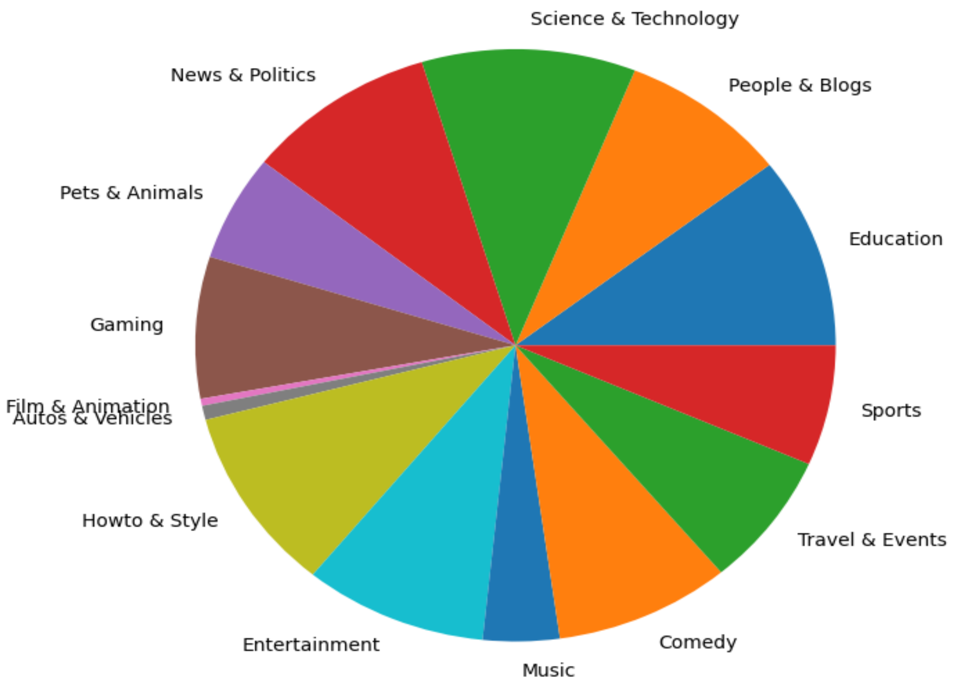


Fig (7)

Fig (7) depicts the percentage of different types of videos in the given dataset. *The "pie chart"is also known as a "circle chart", **dividing** the **circular statistical graphic into** sectors or sections to **illustrate** the **numerical problems.** Each sector denotes a proportionate part of the whole.*

CONCLUSION

The Data analysis of YouTube Videos and Channels Metadata help as to understand the category of videos that are most popular , the time during which most users watch videos were also understandable, category of videos that are least popular among the Data Science YouTube Channels.