Name: Sankalp Pandurang Waghe

Class: M.Sc.IT (Information Technology)

Subject: Big Data

1. Reading the Data & setting up configurations to improve visualization

*# ? Importing Modules*

import pandas as pd

import seaborn as sns

import numpy as np

import matplotlib.pyplot as plt

import warnings

from collections import Counter

import datetime

import json

df = pd.read\_csv(r"D:\Ruia\Sem2\BIgData\YoutubeVideoAnalysis\USvideos.csv") *# ? reading the filename*

PLOT\_COLORS = ["#268bd2", "#0052CC", "#FF5722", "#b58900", "#003f5c"]

pd.options.display.float\_format = '{:.2f}'.format

sns.set(style="ticks")

plt.rc('figure', figsize=(8, 5), dpi=100)

plt.rc('axes', labelpad=20, facecolor="#ffffff", linewidth=0.4, grid=True, labelsize=14)

plt.rc('patch', linewidth=0)

plt.rc('xtick.major', width=0.2)

plt.rc('ytick.major', width=0.2)

plt.rc('grid', color='#9E9E9E', linewidth=0.4)

plt.rc('font', family='Arial', weight='400', size=10)

plt.rc('text', color='#282828')

plt.rc('savefig', pad\_inches=0.3, dpi=300)

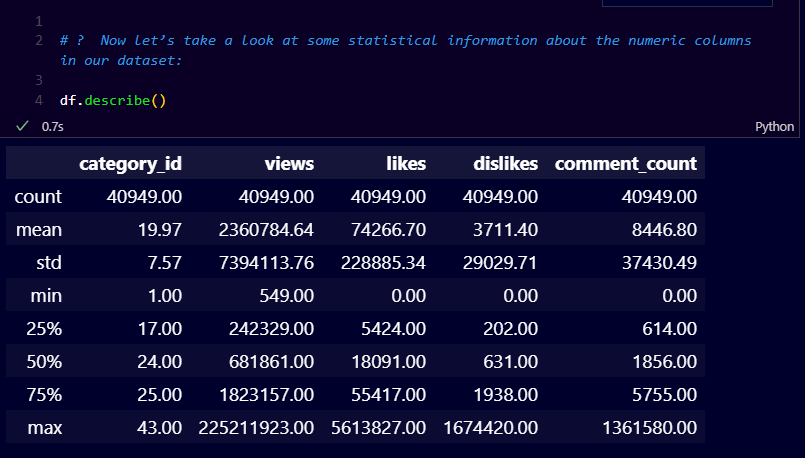
 Data Exploration

The description column has some null values.

These are some of the rows with null description values.

So, to do some sort of data cleaning and to get rid of those null values, I’m going to put an empty string in place of each null value in the Description column:

df["description"] = df["description"].fillna(value="")



Observations from the above table:

1. The average number of views of a trending video is 2,360,784.

2. The median value of the number of views is 681,861, which means that half of the trending videos have less than this number and the other half have views greater than this number.

3. The average number of likes for a trending video is 74,266, while the average number of likes is 3,711.

4. The average number of comments is 8,446 while the median is 1,856

                  Data Visualization

Now we want to see how many trending video titles have at least one uppercase word (eg WHAT).

To do this, we’ll add a new variable to the dataset whose value is True if the video title contains at least one uppercase word, and False otherwise:

def contains\_capitalized\_word(s):

   for w in s.split():

      if w.isupper():

         return True

   return False

df["contains\_capitalized"] = df["title"].apply(contains\_capitalized\_word)

value\_counts = df["contains\_capitalized"].value\_counts().to\_dict()

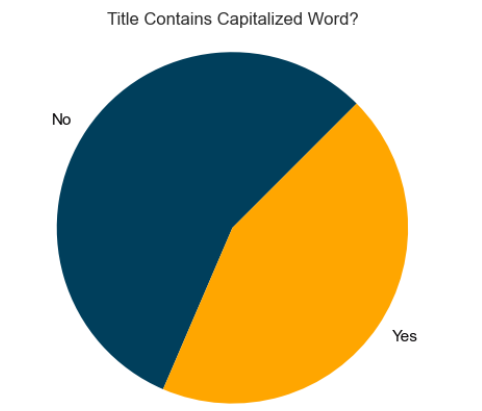
fig, ax = plt.subplots()

\_ = ax.pie([value\_counts[False], value\_counts[True]], labels=['No', 'Yes'],

           colors=['#003f5c', '#ffa600'], textprops={'color': '#040204'}, startangle=45)

\_ = ax.axis('equal')

\_ = ax.set\_title('Title Contains Capitalized Word?')

 **Here We can see that 44% of trending video titles contain at least one word in all caps. We will use our added variable later to analyze the correlation between the variables.**

Let’s add another column to our dataset to analyze the length of titles of videos, then plot the title length histogram to get an idea of the length of trending video titles:

df["title\_length"] = df["title"].apply(lambda x: len(x))

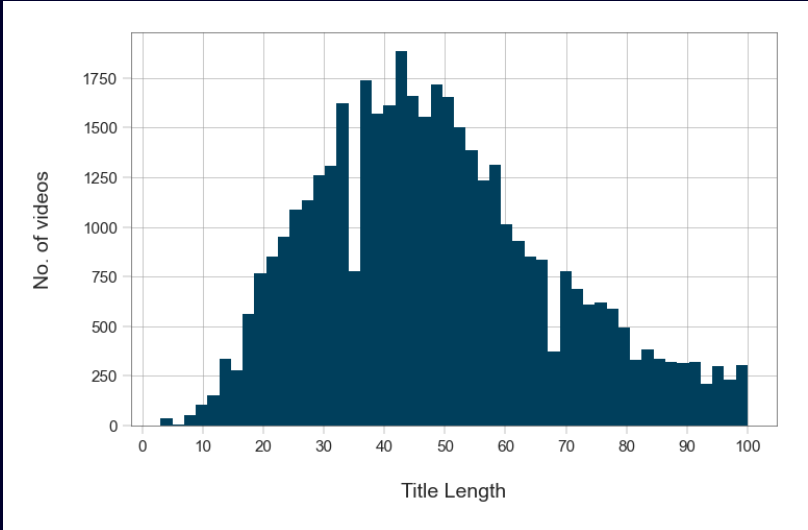
fig, ax = plt.subplots()

\_ = sns.distplot(df["title\_length"], kde=False, rug=False,

                 color=PLOT\_COLORS[4], hist\_kws={'alpha': 1}, ax=ax)

\_ = ax.set(xlabel="Title Length", ylabel="No. of videos",

           xticks=range(0, 110, 10))

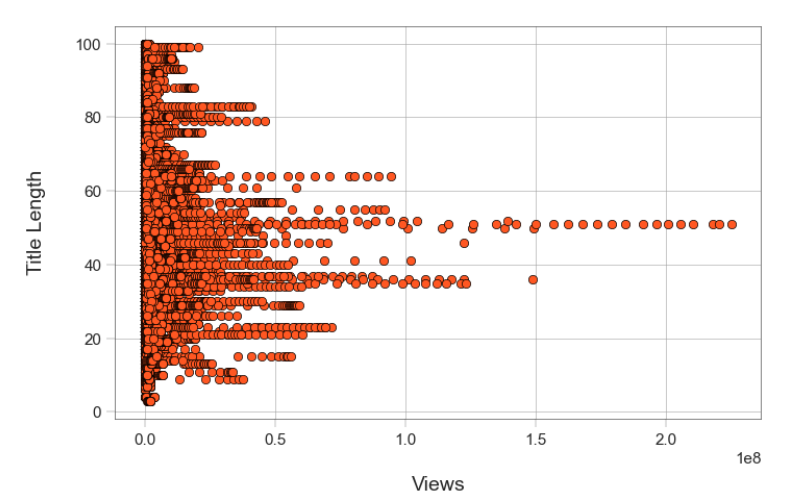
 We can see that the videos title length distribution looks like a normal distribution, where most videos have a title length of around 30-60 characters.

Now, let’s draw a scatter plot to analyze the relationship between the title length and the number of views:

fig, ax = plt.subplots()

\_ = ax.scatter(x=df['views'], y=df['title\_length'], color=PLOT\_COLORS[2], edgecolors="#000000", linewidths=0.5)

\_ = ax.set(xlabel="Views", ylabel="Title Length")

 Looking at the scatter plot, we can tell that there is no relationship between the length of the title and the number of views. However, we do notice an interesting thing that the Videos having 100,000,000 and more views have a title length of between 33 and 55 characters or so    Youtube Trending Videos Analysis : Correlation

Now let’s see how the variables in the dataset correlate to each other: for example, we would like to see how views and likes correlate, meaning that views and likes increase and decrease together:

h\_labels = [

   x.replace('\_', ' ').title() for x in list(

      df.select\_dtypes(

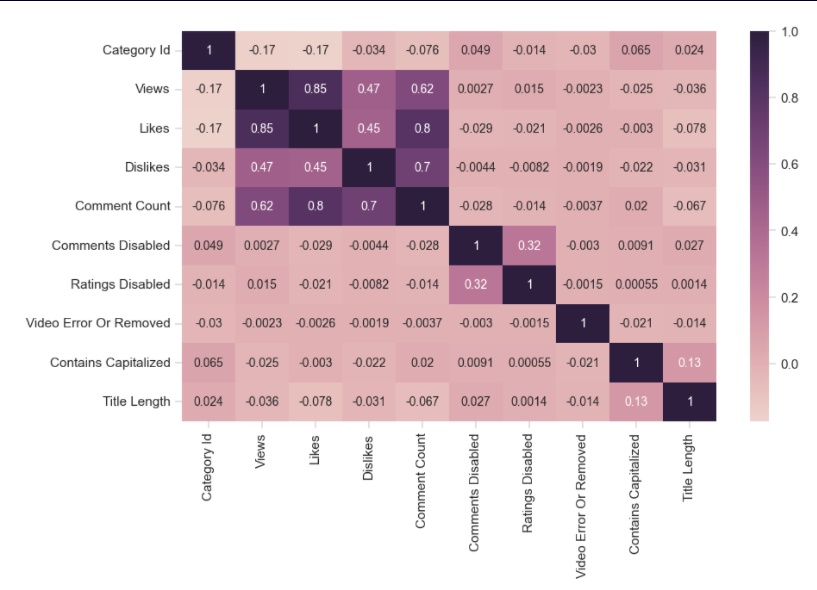
         include=['number', 'bool']).columns.values

      )

   ]

fig, ax = plt.subplots(figsize=(10,6))

\_ = sns.heatmap(df.corr(), annot=True, xticklabels=h\_labels, yticklabels=h\_labels, cmap=sns.cubehelix\_palette(as\_cmap=True), ax=ax)

 The correlation map and correlation table above indicate that views and likes are strongly positively correlated.

Let’s see if some words are used meaningfully in trending video titles. Let’s draw a word cloud for the titles of our trending videos, which is a way to visualize the most common words in the titles; the more common the word, the larger its font-size:

import wordcloud

title\_words = list(df["title"].apply(lambda x: x.split()))

title\_words = [x for y in title\_words for x in y]

wc = wordcloud.WordCloud(width=1200, height=500,

                         collocations=False, background\_color="white",

                         colormap="tab20b").generate(" ".join(title\_words))

plt.figure(figsize=(15,10))

plt.imshow(wc, interpolation='bilinear')

\_ = plt.axis("off")

