**LAB 2**

**SOCIAL MEDIA ANALYSIS**

[**https://www.kaggle.com/datasnaek/youtube-new/**](https://www.kaggle.com/datasnaek/youtube-new/download)

**Importing Libraries**

import pandas as pd

import numpy as np

import matplotlib as mpl

from matplotlib import pyplot as plt

import seaborn as sns

import warnings

from collections import Counter

import datetime

import wordcloud

import json

**Uploading Data**

from google.colab import files

uploaded = files.upload()

**Reading Data**

df = pd.read\_csv("USvideos.csv")

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**

df.describe()

**Visualizing Data**

**PieChart**

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?')

**Histogram**

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))

**Scatter Plot**

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")

**Youtube Trending Video Analysis(Correlation)**

**Heat Map**

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)

**Word Cloud**

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")