Laporan Akhir

Dipergunakan untuk memenuhi tugas individu

Mata Kuliah Statistika Sains Data

Dosen Pengampu:

Mika Alvionita S, S.Si., M.Si.

Riksa Meidy Karim, S.Kom., M.Si., M.Sc.

Disusun Oleh:

Dimas Wahyu Saputro

NIM. 120450081

Kelas : SSD RA

Program Studi Sains Data

Jurusan Sains

Institut Teknologi Sumatera

2021/2022

Analysis of YouTube Trending Videos - Tubes SSD - Dimas Wahyu Saputro

May 19, 2022

1 Introduction

YouTube adalah platform video paling populer dan paling banyak digunakan di dunia saat ini. Salah satu hal yang menarik dari YouTube, adalah banyak sekali data yang bisa didapatkan, seperti jumlah penonton, deskripsi, komentar, dan lain-lain. Saya akan menggunakan packages seperti Matplotlib, Pandas, dan packages lainnya untuk melakukan Data Wrangling, Data Visualization, Data Processing, dan Model Implementation.

Tujuan utama dari analisis ini adalah untuk menemukan fakta dan pola yang menarik dengan mengeksplorasi data dan dengan menggunakan visualisasi yang efektif. Data yang saya gunakan adalah data Trending Youtube yang digunakan saat Praktikum 2 Statistika Sains Data.

Hal yang bisa dipahami setelah membaca berkas ini: 1. 10 video dengan jumlah penonton terendah. 2. 10 video dengan jumlah penonton terbanyak 3. 10 video dengan jumlah likes terbanyak. 4. 10 video dengan jumlah dislikes terbanyak. 5. 10 video yang paling lama berada di Trending List. 6. Kategori yang paling banyak memiliki video trending. 7. Video dengan jumlah ulasan (likes/dislikes) paling banyak. 8. Channel yang paling banyak video trending. 9. Korelasi antar kolom (heat map)

2 Getting Ready

2.1 Modules Import

Pertama, saya mengimpor beberapa paket Python yang akan membantu untuk menganalisis data, terutama pandas untuk analisis data, matplotlib untuk visualisasi, dan sklearn untuk pemodelan Machine Learning.

[1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from matplotlib import cm
import seaborn as sns
sns.set_style('darkgrid')

from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.neighbors import KNeighborsClassifier
```

```
from sklearn.metrics import classification_report
from sklearn.model_selection import cross_val_score
```

Untuk menghilangkan banyak peringatan yang muncul, saya menggunakan perintah di bawah.

[2]:
```
import warnings
warnings.filterwarnings('ignore')
```

2.2 Data Import

Setelah mengimpor paket yang dibutuhkan, selanjutnya mengimpor data. Data yang saya gunakan adalah data USVideos.csv yang digunakan saat praktikum 2.

[3]:
```
import pandas as pd
url = 'https://drive.google.com/file/d/1j79kBhntz5PaizKBPDp03hzPuLgoT-CM/view'
url = 'https://drive.google.com/uc?id=' +url.split('/')[-2]
df = pd.read_csv(url, parse_dates=["publish_time"])
```

3 Overview

3.1 Heading of Data

Ketika sudah mengimpor data, untuk dapat memastikan bahwa data yang kita impor sudah benar, kita dapat menggunakan df.head() untuk melihat data sebanyak n pada kolom awal.

[ ]:
```
df.head()
```

3.2 Describe of Data

Untuk mengetahui jumlah data, kolom, dan tipe data, kita dapat menggunakan df.info()

[5]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 40949 entries, 0 to 40948
Data columns (total 16 columns):
 #   Column                  Non-Null Count  Dtype
---  ------                  --------------  -----
 0   video_id                40949 non-null  object
 1   trending_date           40949 non-null  object
 2   title                   40949 non-null  object
 3   channel_title           40949 non-null  object
 4   category_id             40949 non-null  int64
 5   publish_time            40949 non-null  datetime64[ns, UTC]
 6   tags                    40949 non-null  object
 7   views                   40949 non-null  int64
 8   likes                   40949 non-null  int64
 9   dislikes                40949 non-null  int64
```

```
 10  comment_count           40949 non-null  int64
 11  thumbnail_link          40949 non-null  object
 12  comments_disabled       40949 non-null  bool
 13  ratings_disabled        40949 non-null  bool
 14  video_error_or_removed  40949 non-null  bool
 15  description             40379 non-null  object
dtypes: bool(3), datetime64[ns, UTC](1), int64(5), object(7)
memory usage: 4.2+ MB
```

3.3 Show Row 1

Menggunakan perintah head(1) untuk menampilkan baris data pertama, kemudian gunakan transpose untuk mengubah kolom menjadi baris.

[6]:
```
df.head(1).transpose()
```

[6]:

	0
video_id	2kyS6SvSYSE
trending_date	17.14.11
title	WE WANT TO TALK ABOUT OUR MARRIAGE
channel_title	CaseyNeistat
category_id	22
publish_time	2017-11-13 17:13:01+00:00
tags	SHANtell martin
views	748374
likes	57527
dislikes	2966
comment_count	15954
thumbnail_link	https://i.ytimg.com/vi/2kyS6SvSYSE/default.jpg
comments_disabled	False
ratings_disabled	False
video_error_or_removed	False
description	SHANTELL'S CHANNEL - https://www.youtube.com/s…

3.4 Dataset Dimension

[7]:
```
df.shape
```

[7]:
```
(40949, 16)
```

Terlihat bahwa dataset yang kita gunakan memiliki dimensi (row, column) sebanyak (40949, 16).

3.5 Check Missing Data

Terlihat bahwa ada 40949 baris data. Sangat banyak sekali, oleh karena itu untuk mengetahui apakah ada data yang kosong, kita dapat menggunakan df.isnull().sum()

[8]:
```
# Missing values by column
df.isna().sum()
```

[8]:
```
video_id                    0
trending_date               0
title                       0
channel_title               0
category_id                 0
publish_time                0
tags                        0
views                       0
likes                       0
dislikes                    0
comment_count               0
thumbnail_link              0
comments_disabled           0
ratings_disabled            0
video_error_or_removed      0
description               570
dtype: int64
```

Setelah dilakukan perintah di atas, kita dapat melihat bahwa ada data yang kosong pada kolom description, sebanyak 570.

4 Data Wrangling

Data wrangling merupakan istilah yang digunakan bagi serangkaian proses dalam mengumpulkan dan mengolah, menganalisis, dan merapikan data mentah. Sehingga menjadi informasi lengkap dan sederhana yang dapat dibaca dengan mudah.

Pada dataset USVideos.csv, saya melakukan data wrangling untuk memperbaiki data yang kosong pada kolom description dengan mengisi dengan ' ', mengubah tipe data category_id menjadi string, menghapus beberapa kolom yang tidak diperlukan, dan memperbaiki format data di kolom date_trending.

4.1 Remove any unnecessary columns.

[9]:
```
# delete column tags
df.drop(columns=['tags'], inplace=True)

# delete column thumbnail_link, comments_disabled, ratings_disabled,␣
 ↪video_error_or_removed, and description
df.drop(columns=['thumbnail_link', 'comments_disabled', 'ratings_disabled',␣
 ↪'video_error_or_removed', 'description'], inplace=True)
```

4.2 Change Column category_id to str Type

[10]:
```
df.category_id = df.category_id.astype(str)
```

4.3 Fix Missing Values by ' '

[11]:
```
# change missing value to ' '
df.fillna(' ', inplace=True)
```

4.4 Correct Date Format in Column date_trending

[12]:
```
# If we look at the trending_date or publish_time columns, we see that they are␣
↪not yet in the correct format of datetime data.
df['trending_date'] = pd.to_datetime(df['trending_date'], format='%y.%d.%m')
df.head()
```

[12]:

	video_id	trending_date	title	channel_title	category_id	publish_time	views	likes	dislikes	comment_count
0	2kyS6SvSYSE	2017-11-14	WE WANT TO TALK ABOUT OUR MARRIAGE	CaseyNeistat	22	2017-11-13 17:13:01+00:00	748374	57527	2966	15954
1	1ZAPwfrtAFY	2017-11-14	The Trump Presidency: Last Week Tonight with J…	LastWeekTonight	24	2017-11-13 07:30:00+00:00	2418783	97185	6146	12703
2	5qpjK5DgCt4	2017-11-14	Racist Superman | Rudy Mancuso, King Bach & Le…	Rudy Mancuso	23	2017-11-12 19:05:24+00:00	3191434	146033	5339	8181
3	puqaWrEC7tY	2017-11-14	Nickelback Lyrics: Real or Fake?	Good Mythical Morning	24	2017-11-13 11:00:04+00:00	343168	10172	666	2146
4	d380meD0W0M	2017-11-14	I Dare You: GOING BALD!?	nigahiga	24	2017-11-12 18:01:41+00:00	2095731	132235	1989	17518

5 Data Visualization

Setelah melakukan data wrangling, kita dapat melakukan data visualisasi. Data visualisasi adalah mengubah data menjadi bentuk grafik yang mudah dibaca.

Beberapa visualisasi yang saya coba lakukan: 1. 10 video dengan jumlah penonton terendah. 2. 10 video dengan jumlah penonton terbanyak 3. 10 video dengan jumlah likes terbanyak. 4. 10 video dengan jumlah dislikes terbanyak. 5. 10 video yang paling lama berada di Trending List. 6. Kategori yang paling banyak memiliki video trending. 7. Video dengan jumlah ulasan (likes/dislikes) paling banyak. 8. Channel yang paling banyak video trending. 9. Korelasi antar kolom (heat map)

5.1 10 Videos

Salah satu hal yang menarik dari data YouTube adalah banyak sekali data. Disini saya mencoba untuk mendapatkan data "10 Video ...".

5.1.1 With Lowest Views

Hal yang saya lakukan pertama kali, yaitu saya menggunakan df.groupby() untuk mengumpulkan data dengan menggunakan kolom views. Lalu saya menggunakan df.sort_values() untuk mengurutkan data dengan menggunakan views. Fungsi aggregate() digunakan untuk mengambil judul video dan maksimal views.

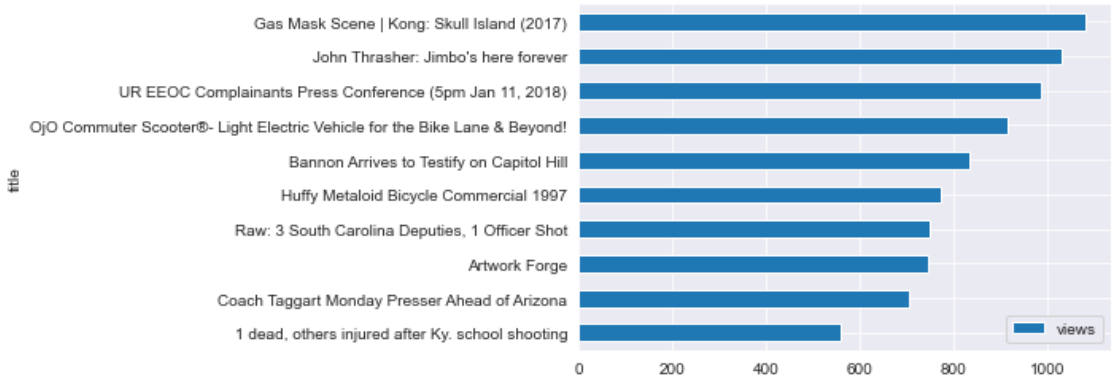
[13]:
```
low_video = df.groupby('video_id').agg({'title': 'first', 'views': 'max'}).
↪sort_values(by='views', ascending=True).head(10)
low_video
```

[13]:

video_id	title	views
y6KYFcta4SE	1 dead, others injured after Ky. school shooting	559
-JVITToppE0	Coach Taggart Monday Presser Ahead of Arizona	704
dQMZLXaa1L8	Artwork Forge	745
zeQaJGkFyqQ	Raw: 3 South Carolina Deputies, 1 Officer Shot	748
qg0GdM60syI	Huffy Metaloid Bicycle Commercial 1997	773
tKX8nUCSBjM	Bannon Arrives to Testify on Capitol Hill	835
JNv4w6DFoYs	OjO Commuter Scooter®- Light Electric Vehicle …	917
TKMXw1YI5S4	UR EEOC Complainants Press Conference (5pm Jan…	988
g3VgrgV3kFk	John Thrasher: Jimbo's here forever	1032
s_FAjI51LPU	Gas Mask Scene | Kong: Skull Island (2017)	1082

[14]:
```
low_video.plot(kind='barh', x='title', y='views')
```

[14]:
```
<AxesSubplot:ylabel='title'>
```



hal yang menarik dari visualisasi ini adalah, video dengan jumlah penonton tidak sampai 500, memungkinkan untuk masuk ke Trending List. Video dengan judul "1 dead, others injured after .." yang memiliki views sebanyak 500 merupakan bukti nyata.

5.1.2 With Top Views

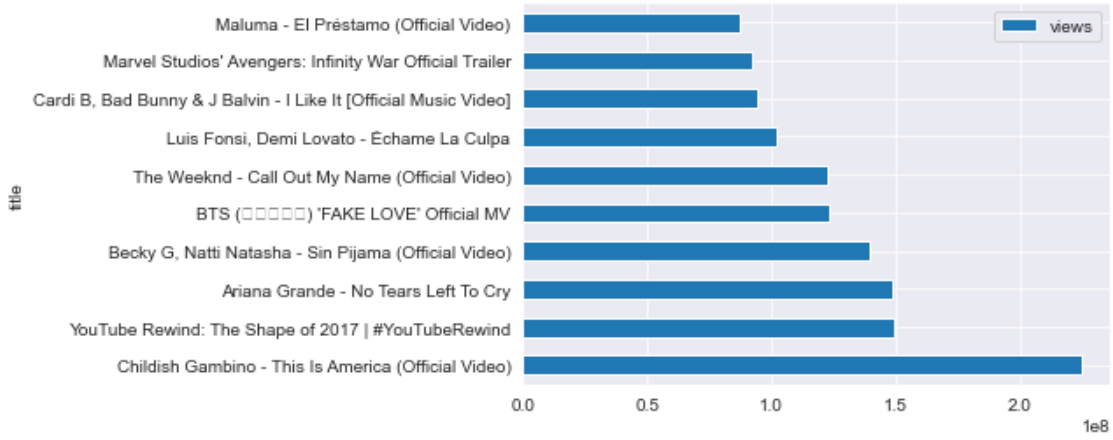
[15]:
```
best_video = df.groupby('video_id').agg({'title': 'first', 'views': 'max'}).
↪sort_values(by='views', ascending=False).head(10)
best_video
```

[15]:

video_id	title	views
VYOjWnS4cMY	Childish Gambino - This Is America (Official V…	225211923
FlsCjmMhFmw	YouTube Rewind: The Shape of 2017 | #YouTubeRe…	149376127
ffxKSjUwKdU	Ariana Grande - No Tears Left To Cry	148689896
zEf423kYfqk	Becky G, Natti Natasha - Sin Pijama (Official …	139334502
7C2z4GqqS5E	BTS ( ) 'FAKE LOVE' Official MV	123010920
M4ZoCHID9GI	The Weeknd - Call Out My Name (Official Video)	122544931
TyHvyGVs42U	Luis Fonsi, Demi Lovato - Échame La Culpa	102012605
xTlNMmZKwpA	Cardi B, Bad Bunny & J Balvin - I Like It [Off…	94254507
6ZfuNTqbHE8	Marvel Studios' Avengers: Infinity War Officia…	91933007
-BQJo3vK8O8	Maluma - El Préstamo (Official Video)	87264467

[16]:
```
best_video.plot(kind='barh', x='title', y='views')
```

[16]:
```
<AxesSubplot:ylabel='title'>
```



Hal yang menarik dari visualisasi ini adalah, video dengan jumlah penonton terbanyak didominasi oleh video dengan jenis musik.

5.1.3 With Top Likes

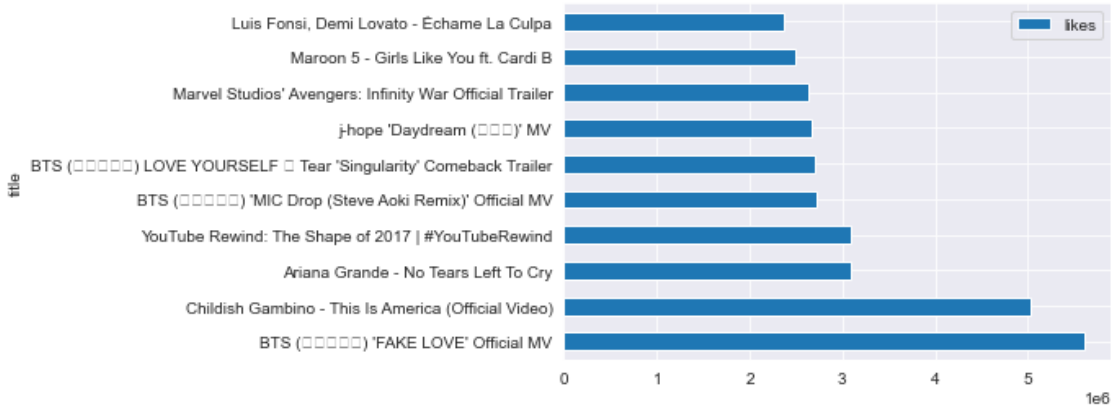
[17]:
```
top_likes = df.groupby('video_id').agg({'title': 'first', 'likes': 'max'}).
↪sort_values(by='likes', ascending=False).head(10)
top_likes
```

[17]:

video_id	title	likes
7C2z4GqqS5E	BTS ( ) 'FAKE LOVE' Official MV	5613827
VYOjWnS4cMY	Childish Gambino - This Is America (Official V…	5023450
ffxKSjUwKdU	Ariana Grande - No Tears Left To Cry	3094021
FlsCjmMhFmw	YouTube Rewind: The Shape of 2017 | #YouTubeRe…	3093544
kTlv5_Bs8aw	BTS ( ) 'MIC Drop (Steve Aoki Remix)' Offi…	2729292
p8npDG2ulKQ	BTS ( ) LOVE YOURSELF Tear 'Singularity'…	2700800
OK3GJOWIQ8s	j-hope 'Daydream ( )' MV	2672431
6ZfuNTqbHE8	Marvel Studios' Avengers: Infinity War Officia…	2625661
aJOTlE1K90k	Maroon 5 - Girls Like You ft. Cardi B	2488565
TyHvyGVs42U	Luis Fonsi, Demi Lovato - Échame La Culpa	2376636

[19]:
```
top_likes.plot(kind='barh', x='title', y='likes')
```

[19]:
```
<AxesSubplot:ylabel='title'>
```



5.1.4 With Top Dislikes

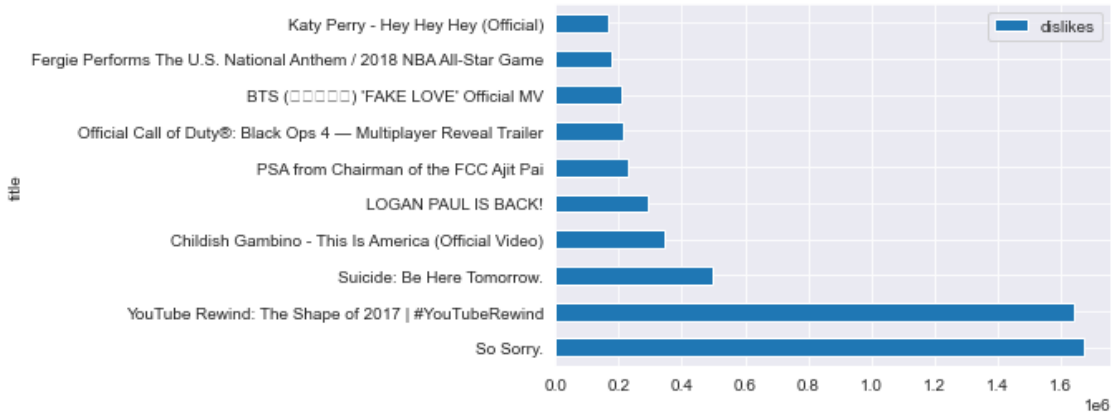
[20]:
```
top_dislikes = df.groupby('video_id').agg({'title': 'first', 'dislikes':␣
↪'max'}).sort_values(by='dislikes', ascending=False).head(10)
top_dislikes
```

[20]:

video_id	title	dislikes
QwZT7T-TXT0	So Sorry.	1674420
FlsCjmMhFmw	YouTube Rewind: The Shape of 2017 | #YouTubeRe…	1643059
oWjxSkJpxFU	Suicide: Be Here Tomorrow.	497847
VYOjWnS4cMY	Childish Gambino - This Is America (Official V…	343541
_5d-sQ7Fh5M	LOGAN PAUL IS BACK!	291900
LFhT6H6pRWg	PSA from Chairman of the FCC Ajit Pai	228426
ooyjaVdt-jA	Official Call of Duty®: Black Ops 4 - Multipla…	212976
7C2z4GqqS5E	BTS ( ) 'FAKE LOVE' Official MV	206892
V5cOvyDpWfM	Fergie Performs The U.S. National Anthem / 201…	176903
WS7f5xpGYn8	Katy Perry - Hey Hey Hey (Official)	165109

[21]:
```
top_dislikes.plot(kind='barh', x='title', y='dislikes')
```

[21]:
```
<AxesSubplot:ylabel='title'>
```



Jika kita perhatikan dengan saksama, ada beberapa video yang selalu muncul, baik di visualisasi jumlah penonton terbanyak, jumlah likes terbanyak, dan jumlah dislikes terbanyak. Contohnya seperti video dengan judul "Childish Gambino - This is America (Official Video)".

5.2 Videos that appeared on the trending list the most

Karena ada kolom trending_date dam publish_date, kita dapat mengetahui berapa lama video berada di Trending List.

[22]:
```
# Videos that appeared on the trending list the most days
vid_id_trend_30 = df.groupby('video_id').size().sort_values(ascending=False).
 ↪head(11)
index = vid_id_trend_30.index
trend_30 = df[df['video_id'].isin(index)].sort_values(by='trending_date',␣
 ↪ascending=True).drop_duplicates(subset=['video_id'])
trend_30.sort_values(by='trending_date', ascending=False)

# show trend_30 with column video_id, title, channel_title, and category_id
trend_30[['video_id', 'title', 'channel_title', 'category_id']].head(10)
```

[22]:

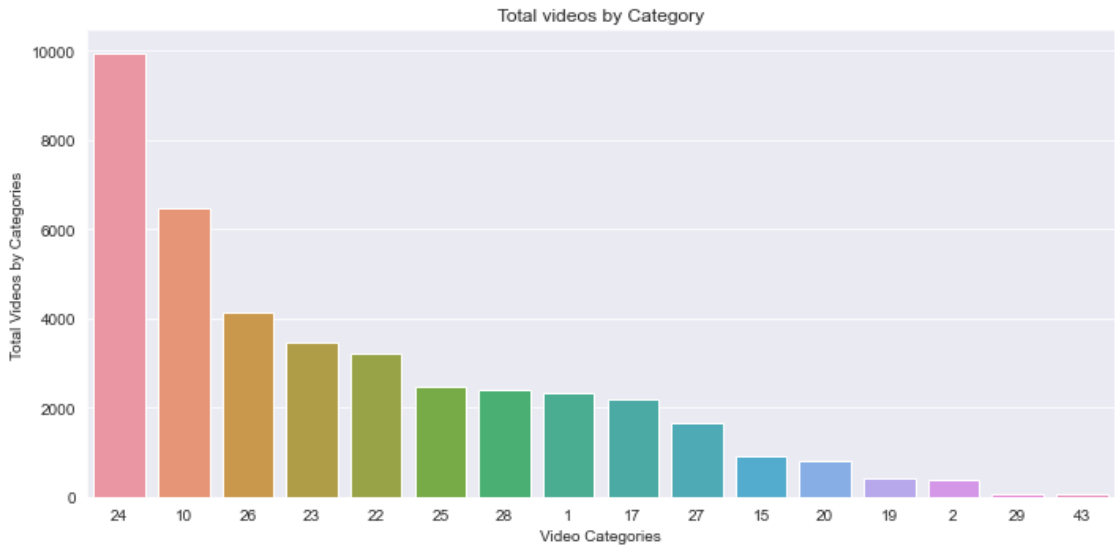
	video_id	title	channel_title	category_id
33750	8h--kFui1JA	Sam Smith - Pray (Official Video) ft. Logic		
33951	WIV3xNz8NoM	Cobra Kai Season 2		
34157	iILJvqrAQ_w	Charlie Puth - BOY [Official Audio]		
34350	UfKmSfgFxi8	FORTNITE The Movie (Official Fake Trailer)		
34359	mdWcaWBxxcY	Rita Ora - Girls ft. Cardi B, Bebe Rexha & Cha…		
34550	j4KvrAUjn6c	WE MADE OUR MOM CRY…HER DREAM CAME TRUE!		
34899	QBL8IRJ5yHU	Why I'm So Scared (being myself and crying too…		
34909	NBSAQenU2Bk	Rooster Teeth Animated Adventures - Millie So …		
34908	r-3iathMo7o	The ULTIMATE $30,000 Gaming PC Setup		
34902	MAjY8mCTXWk	Jay Chou	If You Don't Love Me, It's…	

```
33750  SamSmithWorldVEVO          10
33951           Cobra Kai         24
34157        Charlie Puth         10
34350             nigahiga       24
34359            Rita Ora        24
34550    Lucas and Marcus         24
34899       grav3yardgirl         26
34909       Rooster Teeth          1
34908       Unbox Therapy         28
34902         JVR Music          10
```

5.3 What Category have most trending videos?

[23]:
```
plt.figure(figsize=(10,5))
sns.countplot(x='category_id',data=df, order=df['category_id'].value_counts().
 ↪index)
plt.xlabel('Video Categories')
plt.ylabel('Total Videos by Categories')
plt.title('Total videos by Category')

plt.tight_layout()
plt.show()
```



Karena saya tidak mengetahui label dari setiap kategori, maka visualisasi terlihat seperti pada gambar di atas.

5.4 Most Reviewed Videos (Likes/Dislikes)

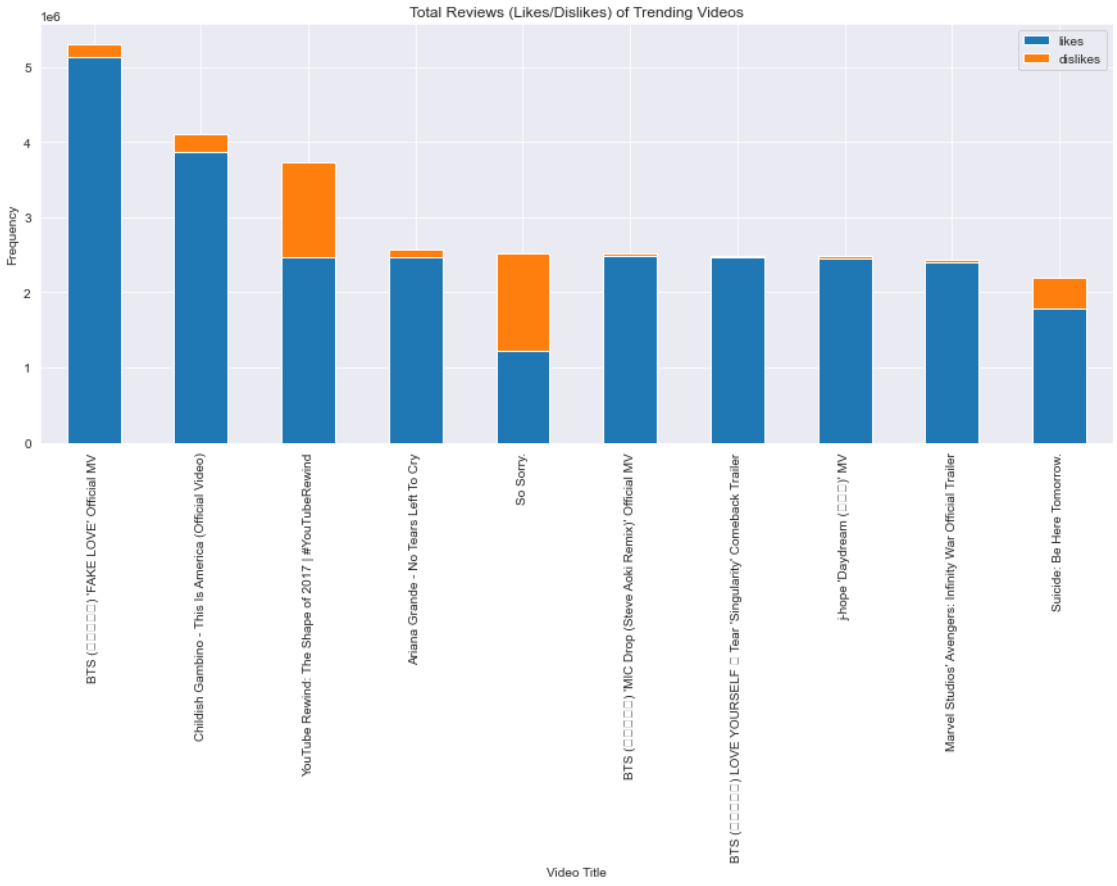
[24]:
```
unique_df_title = df.reset_index().groupby('title')['likes','dislikes'].mean()
unique_df_title['total_reviews'] = round(unique_df_title['likes'] +␣
↪unique_df_title['dislikes'], 2)
unique_df_title = unique_df_title.sort_values(by='total_reviews',␣
↪ascending=False).head(10)
unique_df_title
```

[24]:

title	likes	dislikes	total_reviews
BTS ( ) 'FAKE LOVE' Official MV	5.131075e+06	1.706983e+05	5301773.29
Childish Gambino - This Is America (Official Vi…	3.868033e+06	2.421774e+05	4110210.08
YouTube Rewind: The Shape of 2017 | #YouTubeRewind	2.472672e+06	1.263894e+06	3736565.62
Ariana Grande - No Tears Left To Cry	2.472568e+06	9.389940e+04	2566467.05
So Sorry.	1.213676e+06	1.313220e+06	2526896.00
BTS ( ) 'MIC Drop (Steve Aoki Remix)' Offic…	2.484110e+06	3.903214e+04	2523142.14
BTS ( ) LOVE YOURSELF Tear 'Singularity' …	2.465552e+06	2.369844e+04	2489250.00
j-hope 'Daydream ( )' MV	2.454092e+06	2.362020e+04	2477712.20
Marvel Studios' Avengers: Infinity War Official…	2.395048e+06	4.471600e+04	2439763.78
Suicide: Be Here Tomorrow.	1.783878e+06	4.049249e+05	2188803.12

[25]:
```
unique_df_title[['likes','dislikes']].plot.bar(stacked=True, figsize=(15,6))
plt.xlabel('Video Title')
plt.ylabel('Frequency')
plt.title('Total Reviews (Likes/Dislikes) of Trending Videos')

plt.show()
```

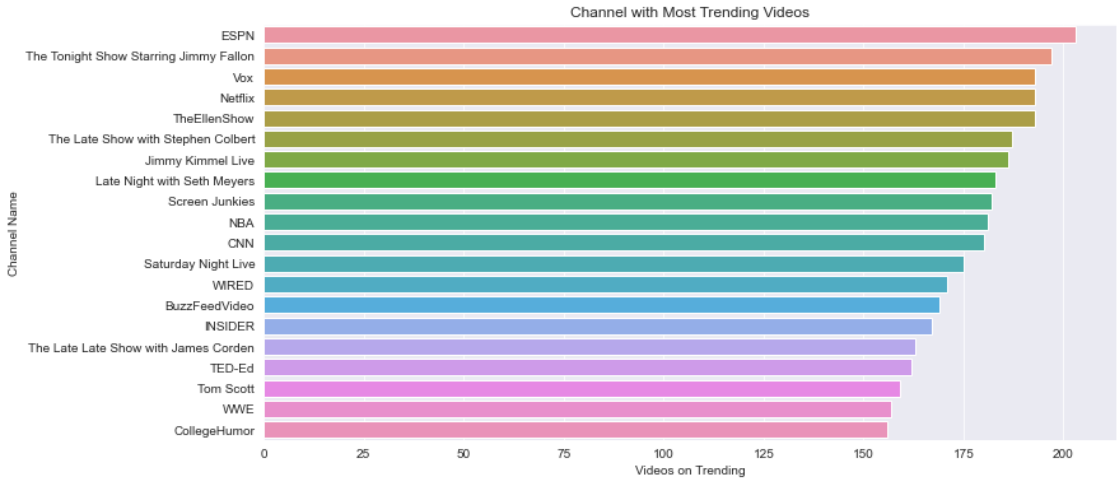


5.5 Channel most trending video

[26]:
```
top10_channels = df.groupby('channel_title')['title'].count().
↪sort_values(ascending=False).head(20)

f = plt.figure(figsize=(12,6))
ax = f.add_subplot(111)
sns.barplot(y=top10_channels.index, x=top10_channels)
plt.xlabel('Videos on Trending')
plt.ylabel('Channel Name')
plt.title('Channel with Most Trending Videos')
```

[26]:
```
Text(0.5, 1.0, 'Channel with Most Trending Videos')
```



5.6 Heat Map

Mengambil relasi input dengan kolom numerik, dan menghitung Koefisien Korelasi Pearson antara setiap pasangan kolom inputnya.

[27]:
```
keep_columns = ['views', 'likes', 'dislikes', 'comment_count'] # only looking␣
↪at correlations between these variables
corr_matrix = df[keep_columns].corr()
corr_matrix
```

[27]:

	views	likes	dislikes	comment_count
views	1.000000	0.849177	0.472213	0.617621
likes	0.849177	1.000000	0.447186	0.803057
dislikes	0.472213	0.447186	1.000000	0.700184
comment_count	0.617621	0.803057	0.700184	1.000000

Semakin nilainya menuju 1, maka dapat disimpulkan bahwa korelasi antara kolom tersebut sangat besar. Conthonya seperti likes dan views.

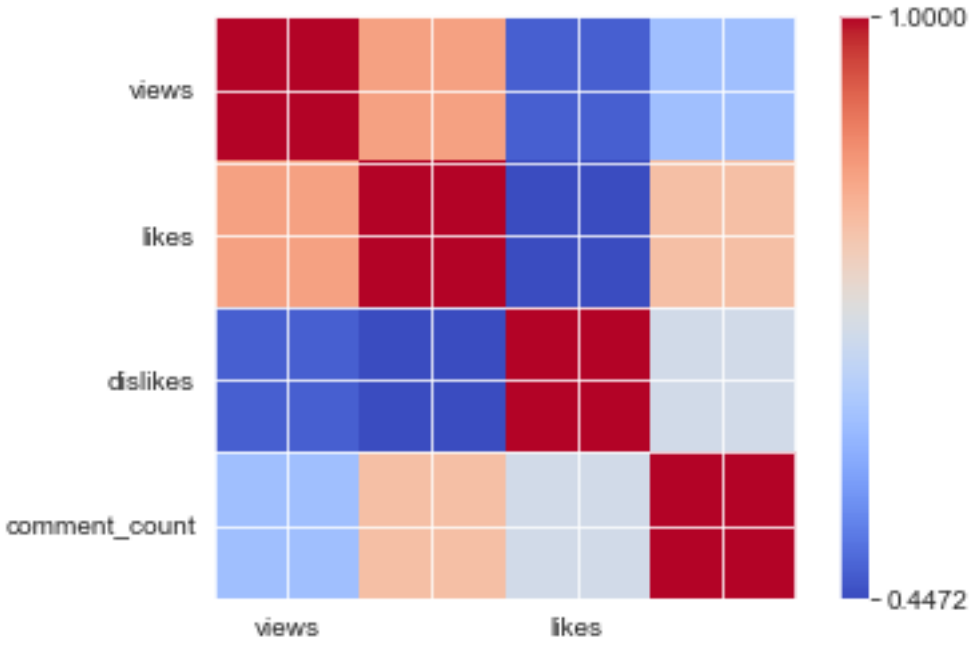
[28]:
```
fig, ax = plt.subplots()
heatmap = ax.imshow(corr_matrix, interpolation='nearest', cmap=cm.coolwarm)

# making the colorbar on the side
cbar_min = corr_matrix.min().min()
cbar_max = corr_matrix.max().max()
cbar = fig.colorbar(heatmap, ticks=[cbar_min, cbar_max])

# making the labels
labels = ['']
```

```
for column in keep_columns:
    labels.append(column)
    labels.append('')
ax.set_yticklabels(labels, minor=False)
ax.set_xticklabels(labels, minor=False)

plt.show()
```



6 Data Processing

Pada tahap ini, saya akan melakukan data processing. Data processing adalah proses yang dilakukan untuk memperbaiki data yang kita gunakan. Supaya dataset USVideos.csv dapat digunakan untuk pemodelan menggunakan KNN, saya akan membagi data menjadi data train dan data test.

6.1 Train Test Split

[29]:
```
# get X: views + likes + dislikes + comment_count
X = df[['views', 'likes', 'dislikes', 'comment_count']]
# get y: category_id
y = df['category_id']
```

[30]:
```
X
```

[30]:
```
          views   likes  dislikes  comment_count
0        748374   57527      2966          15954
1       2418783   97185      6146          12703
2       3191434  146033      5339           8181
3        343168   10172       666           2146
4       2095731  132235      1989          17518
…           …       …         …              …
40944   1685609   38160      1385           2657
40945   1064798   60008       382           3936
40946   1066451   48068      1032           3992
40947   5660813  192957      2846          13088
40948  10306119  357079    212976         144795

[40949 rows x 4 columns]
```

[31]:
```
y
```

[31]:
```
0        22
1        24
2        23
3        24
4        24
        ..
40944    15
40945    22
40946    24
40947     1
40948    20
Name: category_id, Length: 40949, dtype: object
```

[32]:
```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,␣
 ↪random_state=10)
```

6.2 Feature Scaling

Feature Scaling adalah suatu cara untuk membuat numerical data pada dataset memiliki rentang nilai (scale) yang sama. Tidak ada lagi satu variabel data yang mendominasi variabel data lainnya

[33]:
```
from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()
scaler.fit(X_train)

X_train = scaler.transform(X_train)
X_test = scaler.transform(X_test)
```

7 Model Implementation

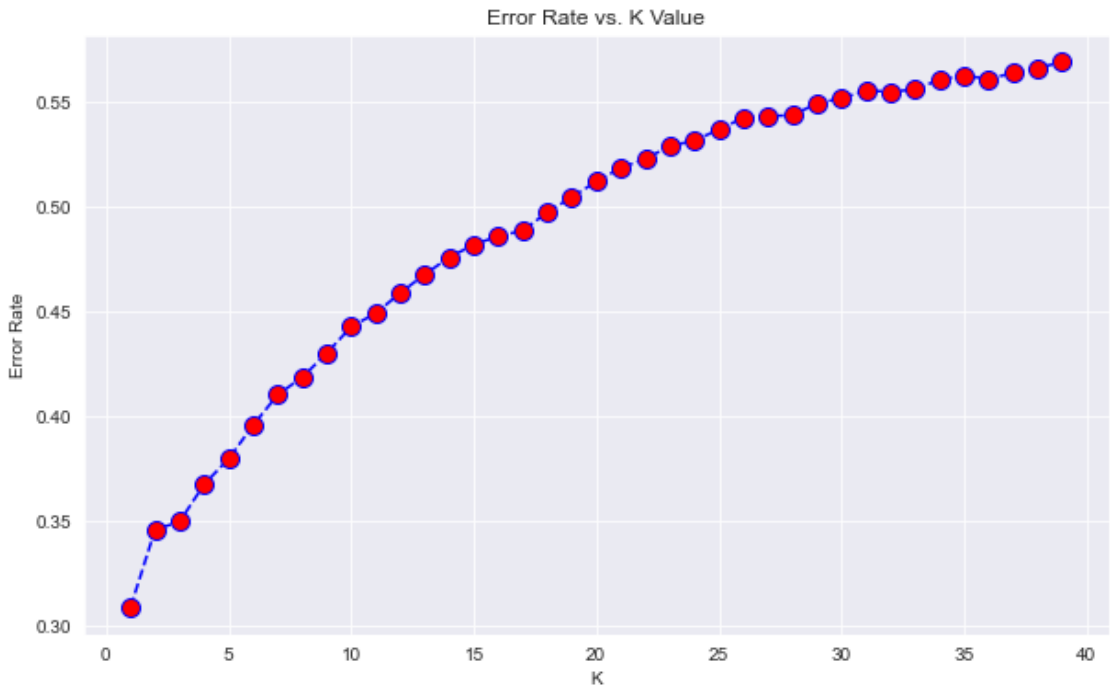
7.1 Create KNN

7.1.1 Check the Best K

[34]:
```
error_rate = []
for i in range(1,40):
    knn = KNeighborsClassifier(n_neighbors=i)
    knn.fit(X_train,y_train)
    pred_i = knn.predict(X_test)
    error_rate.append(np.mean(pred_i != y_test))

plt.figure(figsize=(10,6))
plt.plot(range(1,40),error_rate,color='blue', linestyle='dashed',
         marker='o',markerfacecolor='red', markersize=10)
plt.title('Error Rate vs. K Value')
plt.xlabel('K')
plt.ylabel('Error Rate')
print("Minimum error:-",min(error_rate),"at K =",error_rate.
 ↪index(min(error_rate)))
```

```
Minimum error:- 0.3086691086691087 at K = 0
```



[35]:
```
from sklearn.neighbors import KNeighborsClassifier
knn = KNeighborsClassifier(n_neighbors=2)
```

[36]:
```
knn.fit(X_train, y_train)
```

[36]:
```
KNeighborsClassifier(n_neighbors=2)
```

[37]:
```
knn.score(X_test, y_test)
```

[37]:
```
0.6543345543345543
```

[38]:
```
knn.predict(X_test)
```

[38]:
```
array(['10', '10', '10', …, '20', '10', '23'], dtype=object)
```

7.2 Classification Report

[39]:
```
from sklearn.metrics import classification_report

y_pred = knn.predict(X_test)
print(classification_report(y_test, y_pred))
```

```
              precision    recall  f1-score   support

           1       0.53      0.74      0.62       467
          10       0.62      0.71      0.66      1308
          15       0.46      0.50      0.48       195
          17       0.59      0.73      0.65       417
          19       0.45      0.54      0.49        93
           2       0.54      0.65      0.59        94
          20       0.72      0.80      0.76       169
          22       0.59      0.67      0.63       620
          23       0.64      0.66      0.65       713
          24       0.71      0.66      0.68      1966
          25       0.82      0.67      0.74       457
          26       0.73      0.59      0.65       834
          27       0.74      0.54      0.63       342
          28       0.81      0.55      0.65       495
          29       1.00      0.27      0.43        11
          43       0.89      0.89      0.89         9

    accuracy                           0.65      8190
   macro avg       0.68      0.64      0.64      8190
weighted avg       0.67      0.65      0.66      8190
```

Terlihat bahwa akurasi dari model KNN adalah 0.65