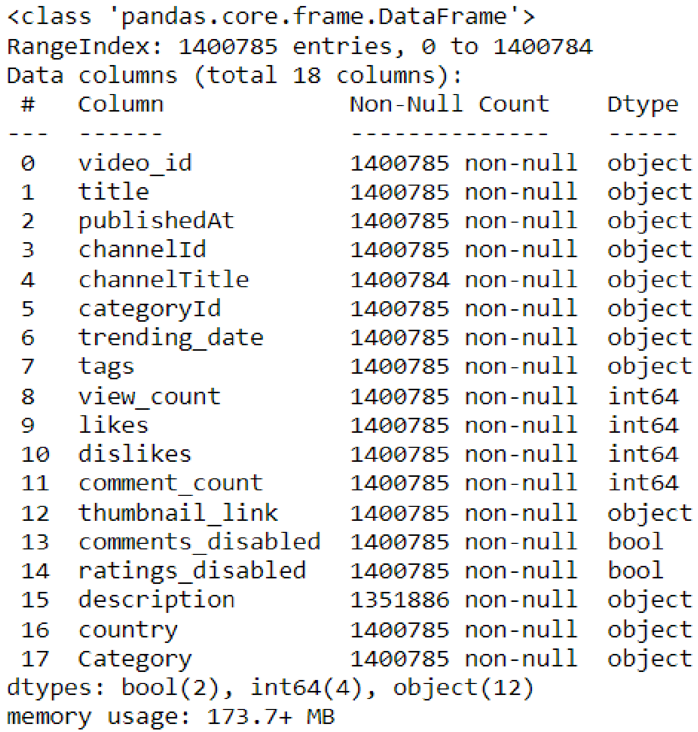
Exploring data from video sharing websites

Youmna Alsayed

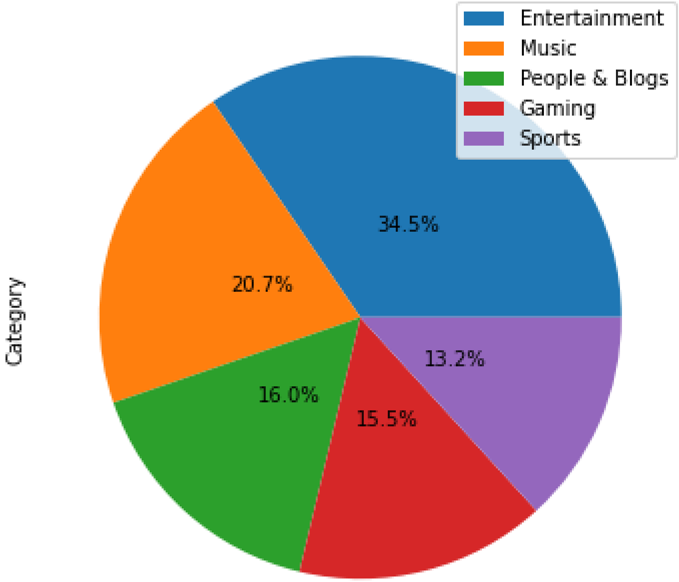
Dataset

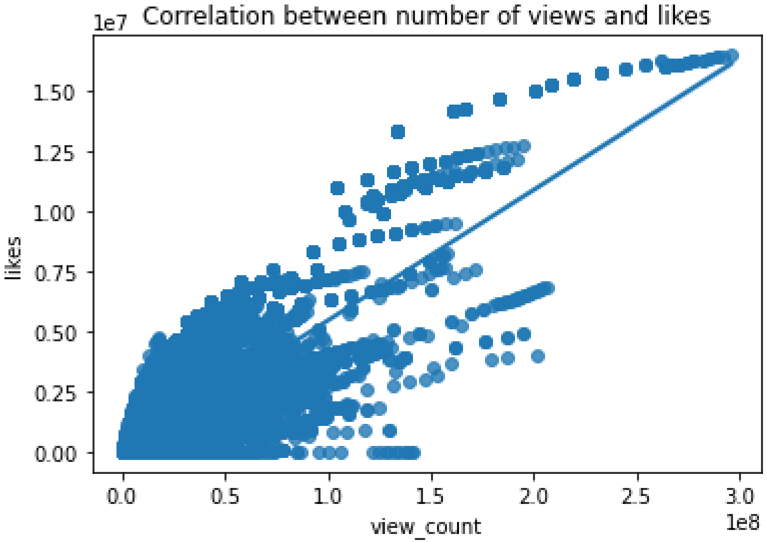
Dataset



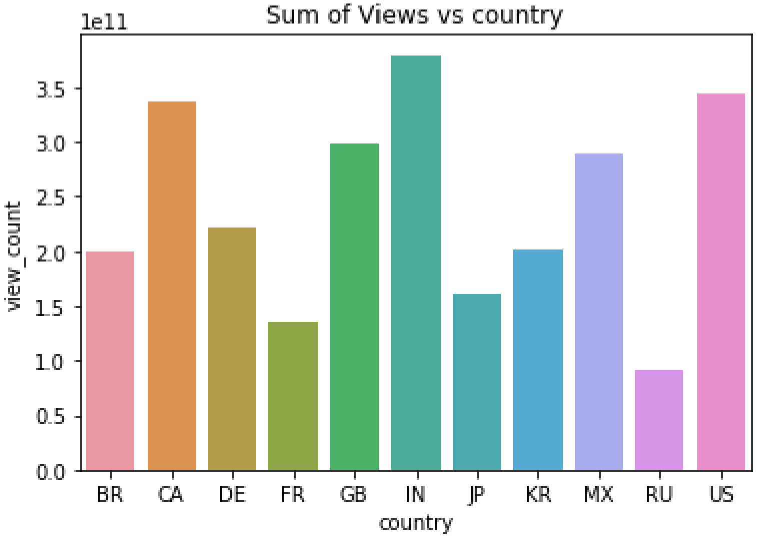
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1400785 entries, 0 to 1400784
Data columns (total 18 columns):
 #   Column             Non-Null Count    Dtype 
---  ------             --------------    ----- 
 0   video_id           1400785 non-null  object
 1   title              1400785 non-null  object
 2   publishedAt        1400785 non-null  object
 3   channelId          1400785 non-null  object
 4   channelTitle       1400784 non-null  object
 5   categoryId         1400785 non-null  object
 6   trending_date      1400785 non-null  object
 7   tags               1400785 non-null  object
 8   view_count         1400785 non-null  int64 
 9   likes              1400785 non-null  int64 
 10  dislikes           1400785 non-null  int64 
 11  comment_count      1400785 non-null  int64 
 12  thumbnail_link     1400785 non-null  object
 13  comments_disabled  1400785 non-null  bool  
 14  ratings_disabled   1400785 non-null  bool  
 15  description        1351886 non-null  object
 16  country            1400785 non-null  object
 17  Category           1400785 non-null  object
dtypes: bool(2), int64(4), object(12)
memory usage: 173.7+ MB
```

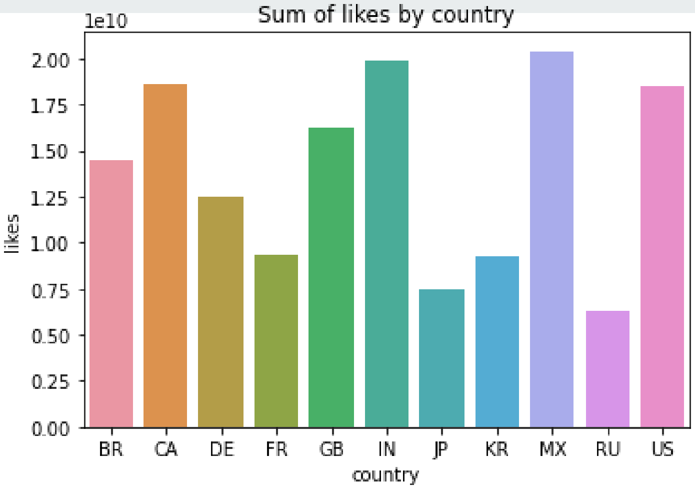
BASIC DATA EXPLORATION

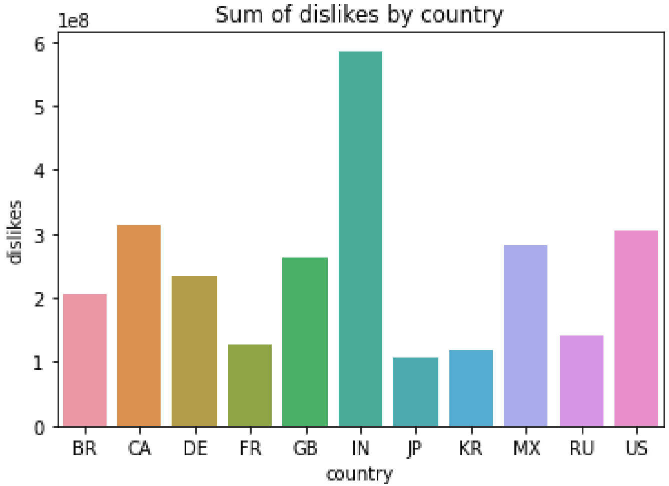




BASIC DATA EXPLORATION

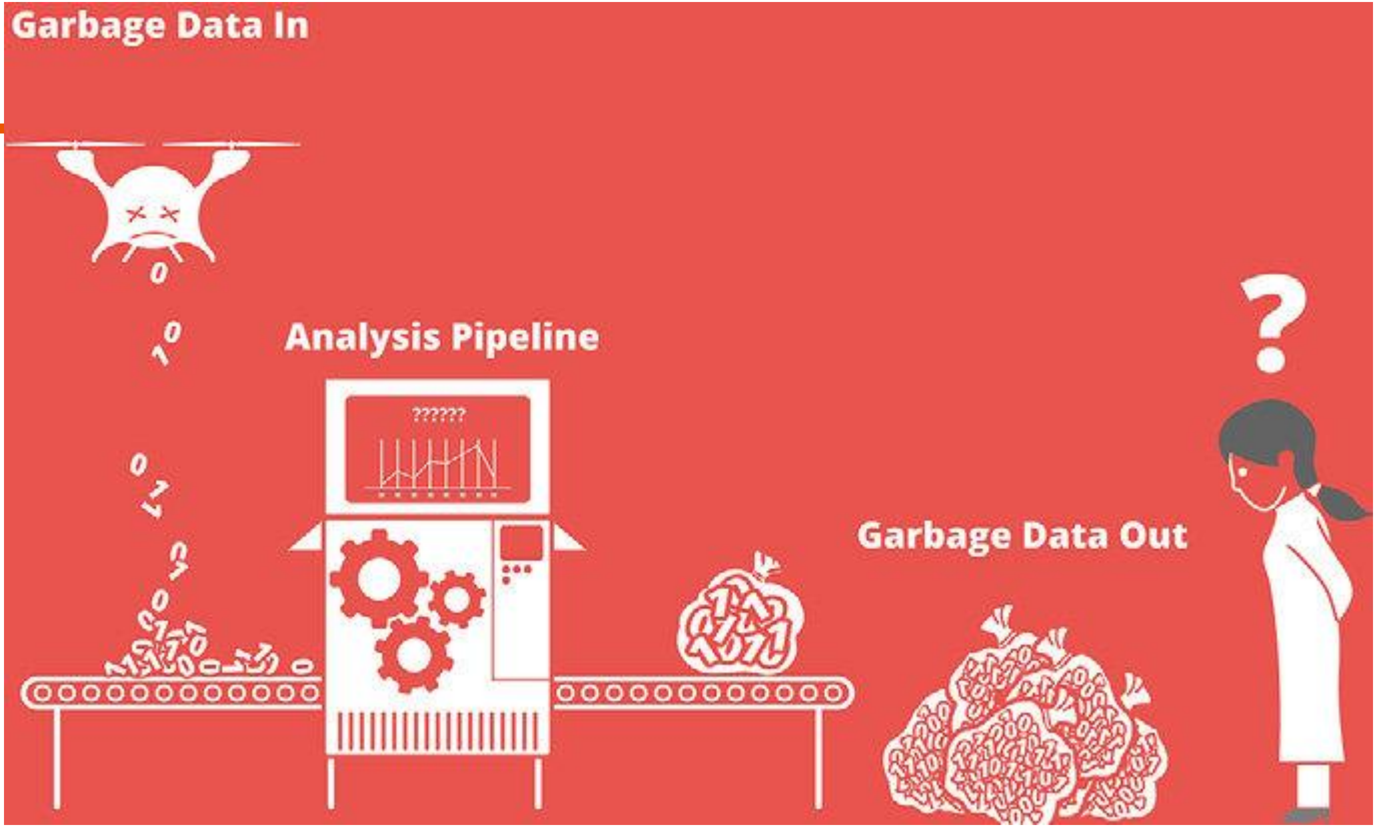




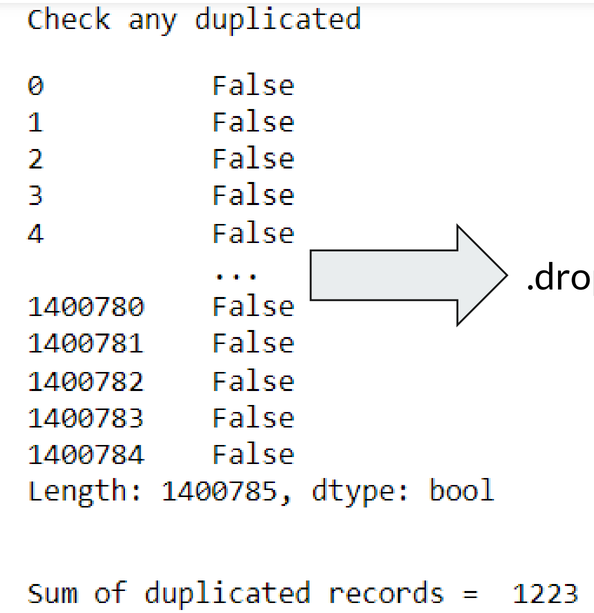


Motivation

- Categories’ importance → effect on the profit of the channels, and their position
- Views count → effect on the profit of the channels
- Does the coronavirus affect of the views and increased it in 2020 comparing to the other years (2021, 2022)?


Garbage Data In
Analysis Pipeline
??????
Garbage Data Out
?

Analyzing the quality of the data

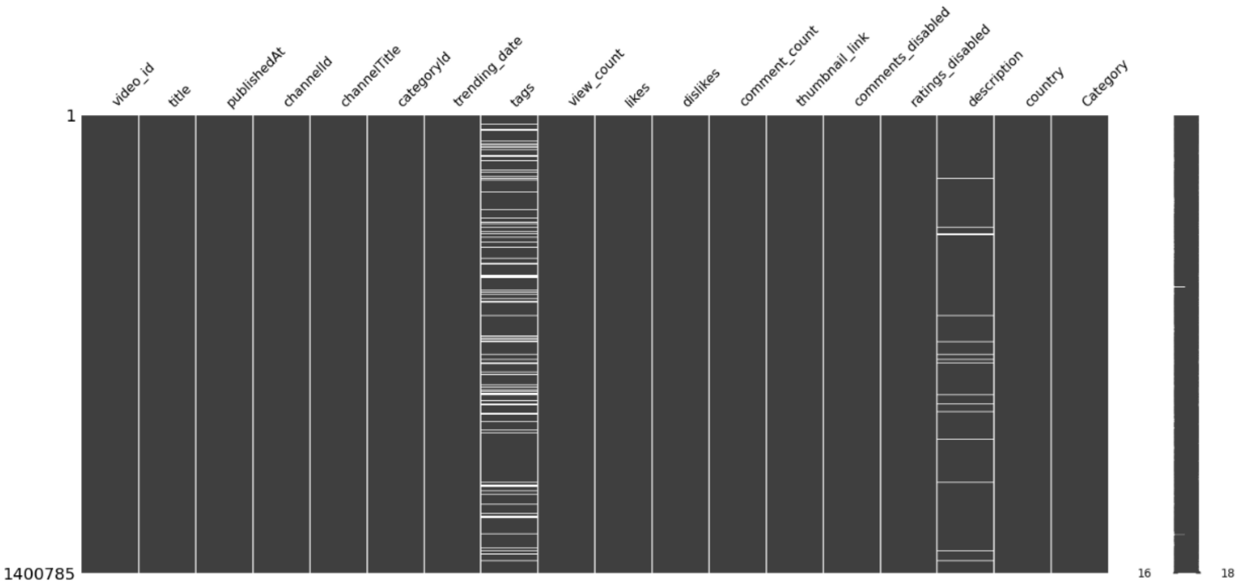


```
Check any duplicated

0          False
1          False
2          False
3          False
4          False
           ...
1400780    False
1400781    False
1400782    False
1400783    False
1400784    False
Length: 1400785, dtype: bool


Sum of duplicated records =  1223
```

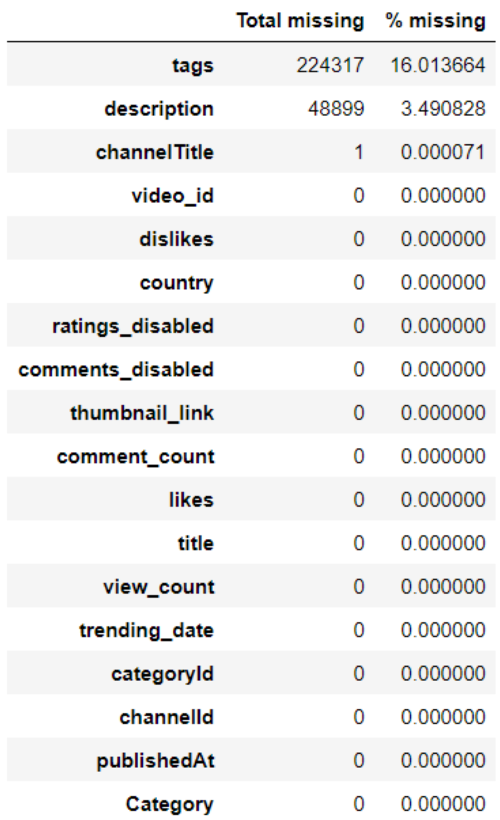
.drop_duplicates()

Sum of duplicated records after dropping= 0

Analyzing the quality of the data



Out[12]:



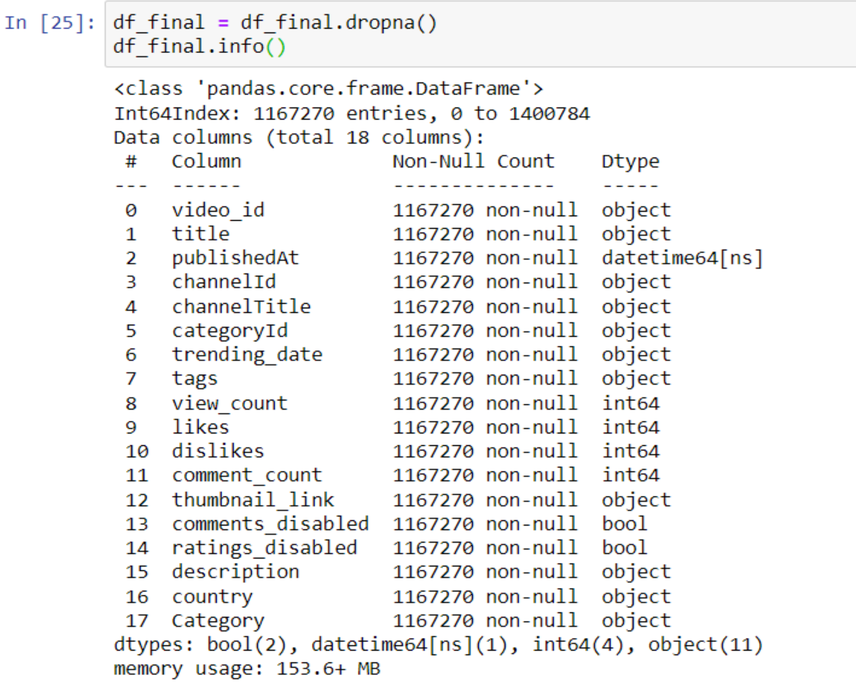
	Total missing	% missing
tags	224317	16.013664
description	48899	3.490828
channelTitle	1	0.000071
video_id	0	0.000000
dislikes	0	0.000000
country	0	0.000000
ratings_disabled	0	0.000000
comments_disabled	0	0.000000
thumbnail_link	0	0.000000
comment_count	0	0.000000
likes	0	0.000000
title	0	0.000000
view_count	0	0.000000
trending_date	0	0.000000
categoryId	0	0.000000
channelId	0	0.000000
publishedAt	0	0.000000
Category	0	0.000000

Feature Engineering

- Handling missing data
- New features extraction
- Handling skewed data
- Handling the duplicated videos (not duplicated in all features)
- Converting categorical data into numerical data

Feature Engineering (Handling missing data)

From our observations the missing values are Missing Not At Random (MNAR), as we think they depend on unobserved data, and we can not explain the pattern in the missing data. So, we are going to drop them.



```
In [25]: df_final = df_final.dropna()
         df_final.info()
```

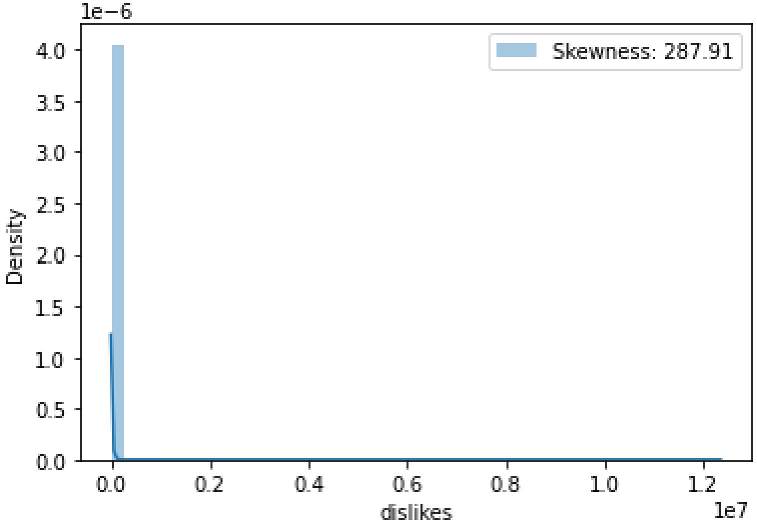
```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1167270 entries, 0 to 1400784
Data columns (total 18 columns):
 #   Column             Non-Null Count    Dtype
---  ------             --------------    -----
 0   video_id           1167270 non-null  object
 1   title              1167270 non-null  object
 2   publishedAt        1167270 non-null  datetime64[ns]
 3   channelId          1167270 non-null  object
 4   channelTitle       1167270 non-null  object
 5   categoryId         1167270 non-null  object
 6   trending_date      1167270 non-null  object
 7   tags               1167270 non-null  object
 8   view_count         1167270 non-null  int64
 9   likes              1167270 non-null  int64
 10  dislikes           1167270 non-null  int64
 11  comment_count      1167270 non-null  int64
 12  thumbnail_link     1167270 non-null  object
 13  comments_disabled  1167270 non-null  bool
 14  ratings_disabled   1167270 non-null  bool
 15  description        1167270 non-null  object
 16  country            1167270 non-null  object
 17  Category           1167270 non-null  object
dtypes: bool(2), datetime64[ns](1), int64(4), object(11)
memory usage: 153.6+ MB
```

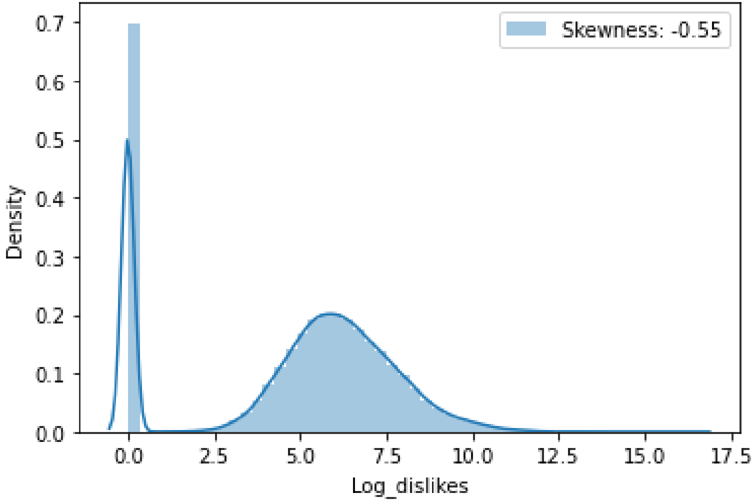
Feature Engineering (New features extraction)

<table>
<tr><th>tags</th><th>trending_date</th><th>publishedAt</th></tr>
<tr><td>Amber|amber vtuber|genshi|genshi game|genshi impact|genshi video|genshin|genshin game|genshin impact|genshin impact 2020|genshin impact game|genshin impact good|genshin impact graphics|genshin impact introduction||MMO PlayStation</td><td>2020-08-12</td><td>2020-08-11 22:21:49</td></tr>
</table>

Feature Engineering (Handling skewed data)

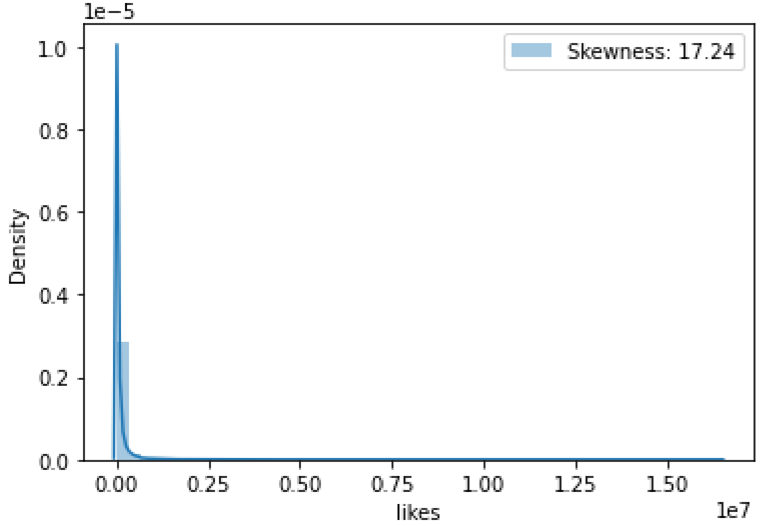
Dislikes

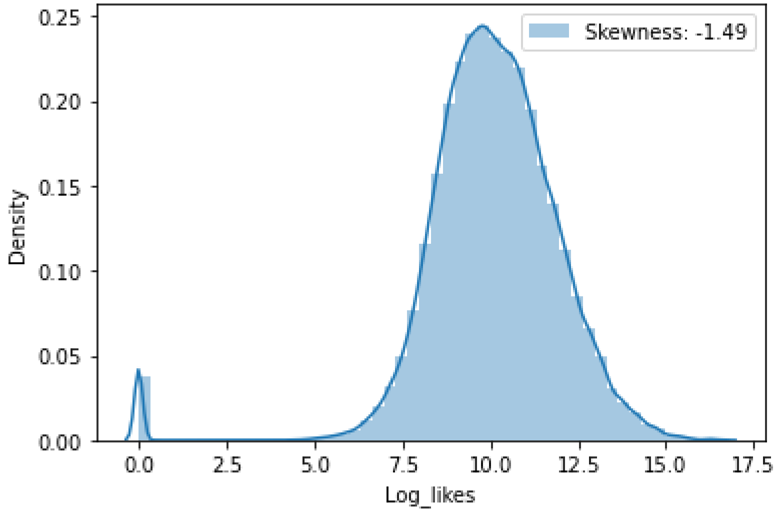




Feature Engineering (Handling skewed data)

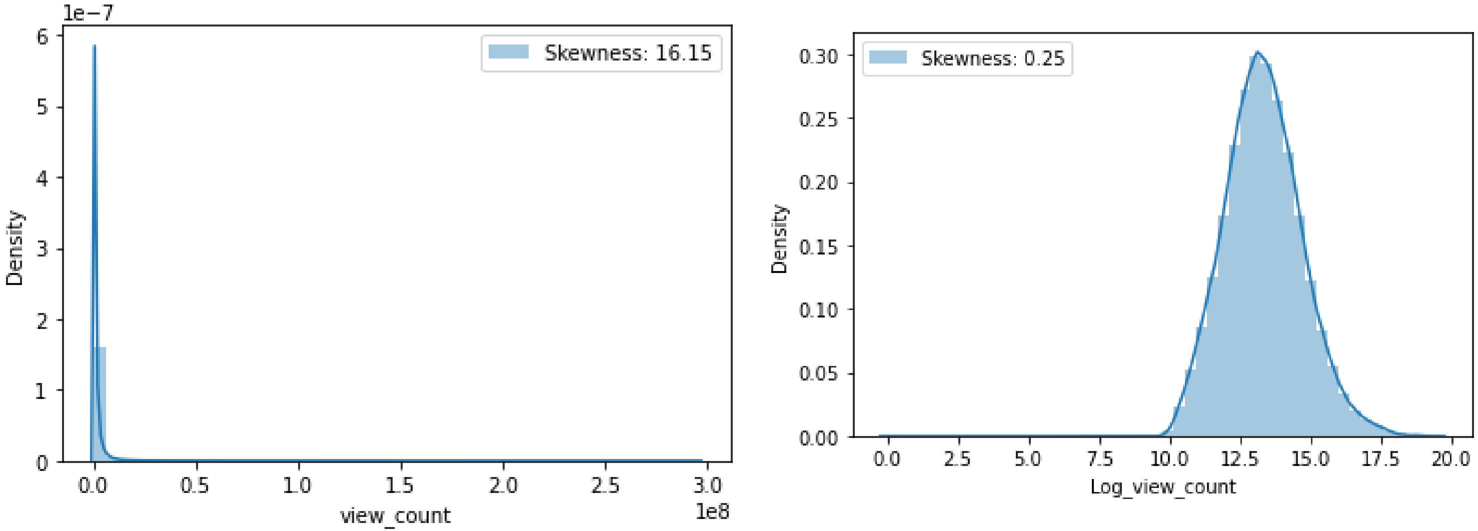
likes





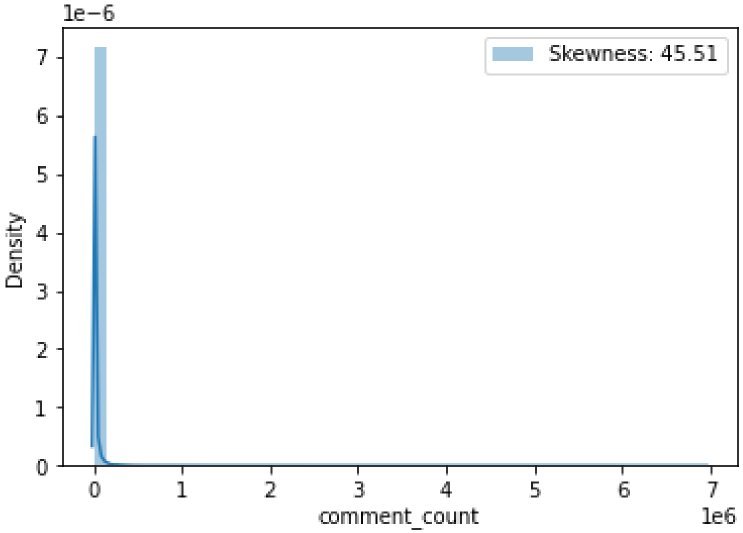
Feature Engineering (Handling skewed data)

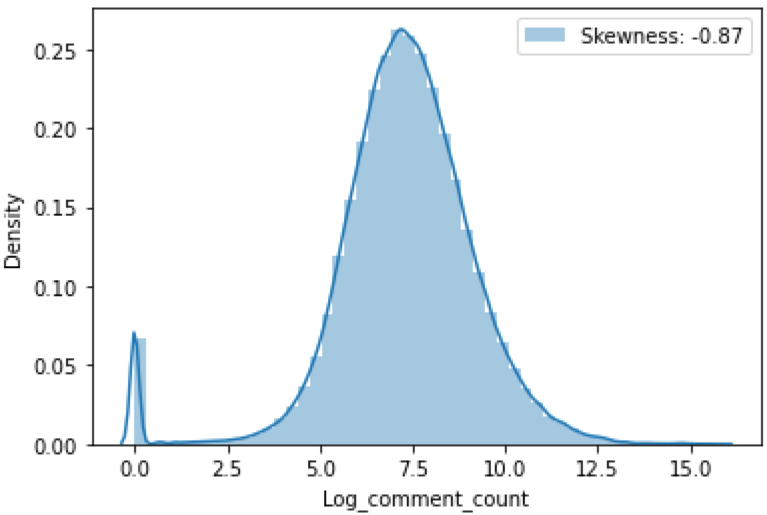
Count views



Feature Engineering (Handling skewed data)

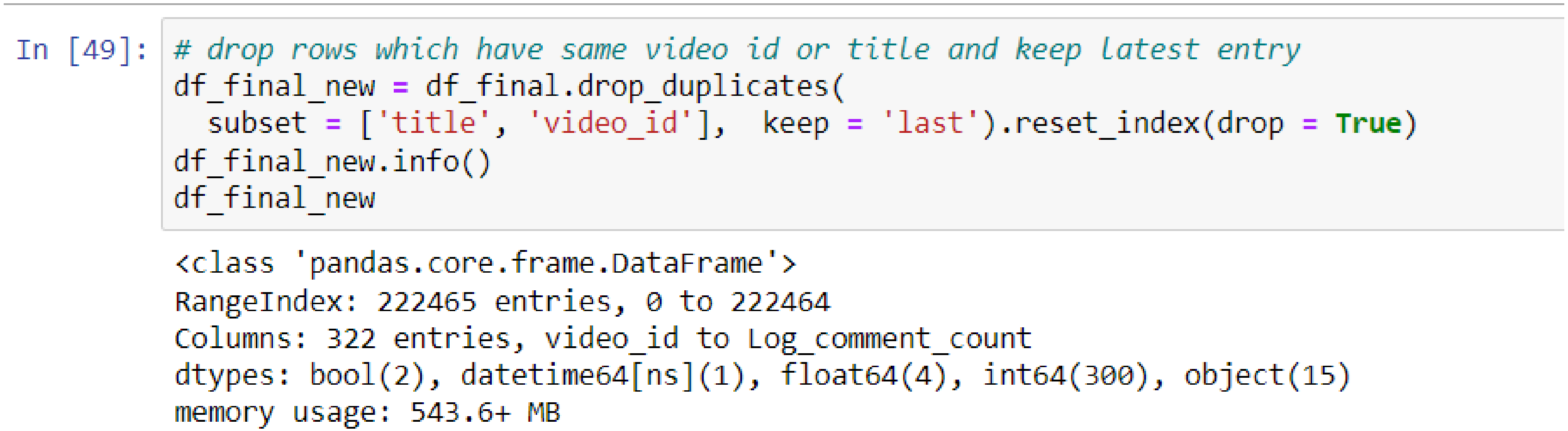
Comments count





Feature Engineering (Handling the duplicated videos)

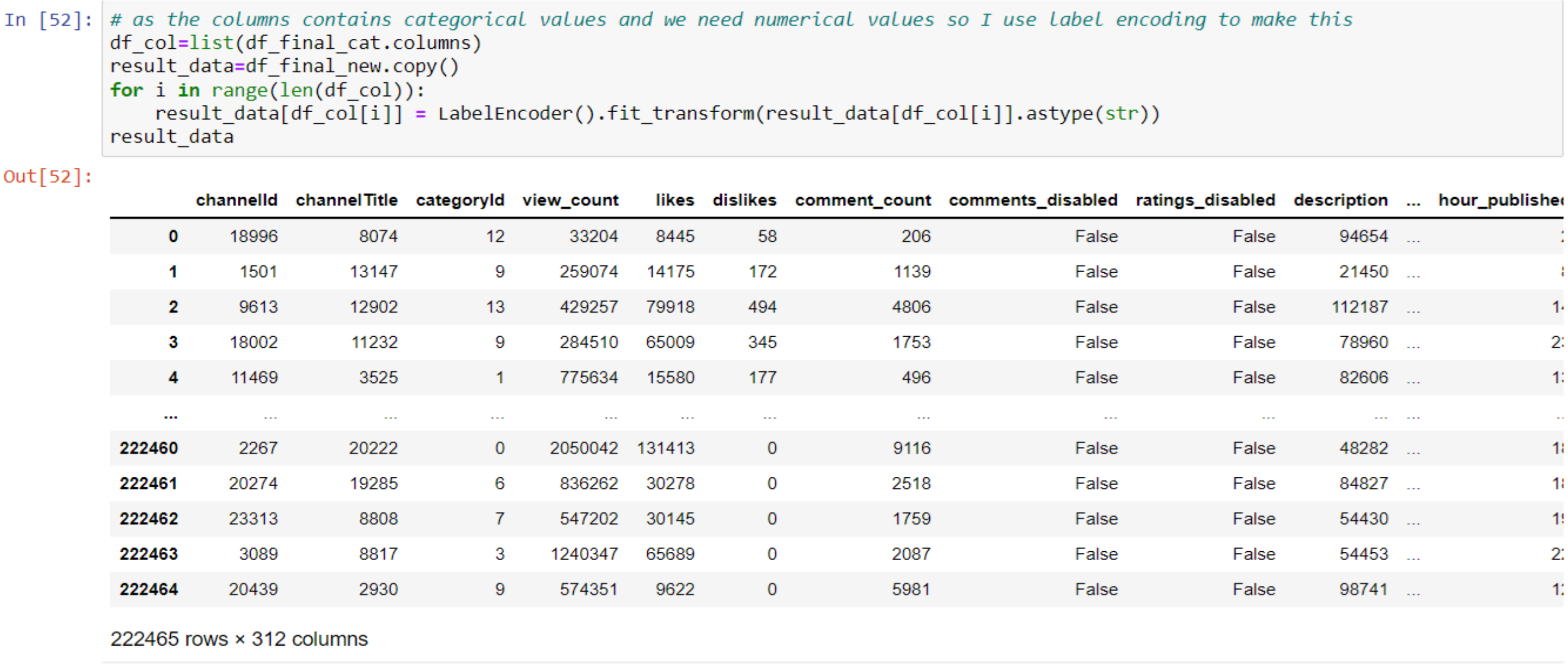
Drop the duplicated rows that have the same video id and same title with keeping the latest entry for them.



```
In [49]: # drop rows which have same video id or title and keep latest entry
         df_final_new = df_final.drop_duplicates(
           subset = ['title', 'video_id'],  keep = 'last').reset_index(drop = True)
         df_final_new.info()
         df_final_new
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 222465 entries, 0 to 222464
Columns: 322 entries, video_id to Log_comment_count
dtypes: bool(2), datetime64[ns](1), float64(4), int64(300), object(15)
memory usage: 543.6+ MB
```

Feature Engineering (Categorical → numerical data)



In [52]:

```
# as the columns contains categorical values and we need numerical values so I use label encoding to make this
df_col=list(df_final_cat.columns)
result_data=df_final_new.copy()
for i in range(len(df_col)):
    result_data[df_col[i]] = LabelEncoder().fit_transform(result_data[df_col[i]].astype(str))
result_data
```

Out[52]:

	channelId	channelTitle	categoryId	view_count	likes	dislikes	comment_count	comments_disabled	ratings_disabled	description	...	hour_publishe
0	18996	8074	12	33204	8445	58	206	False	False	94654	...	2
1	1501	13147	9	259074	14175	172	1139	False	False	21450	...	8
2	9613	12902	13	429257	79918	494	4806	False	False	112187	...	14
3	18002	11232	9	284510	65009	345	1753	False	False	78960	...	2
4	11469	3525	1	775634	15580	177	496	False	False	82606	...	1
...	...	...	...	...	...	...	...	...	...	...	...	..
222460	2267	20222	0	2050042	131413	0	9116	False	False	48282	...	18
222461	20274	19285	6	836262	30278	0	2518	False	False	84827	...	18
222462	23313	8808	7	547202	30145	0	1759	False	False	54430	...	19
222463	3089	8817	3	1240347	65689	0	2087	False	False	54453	...	2
222464	20439	2930	9	574351	9622	0	5981	False	False	98741	...	1

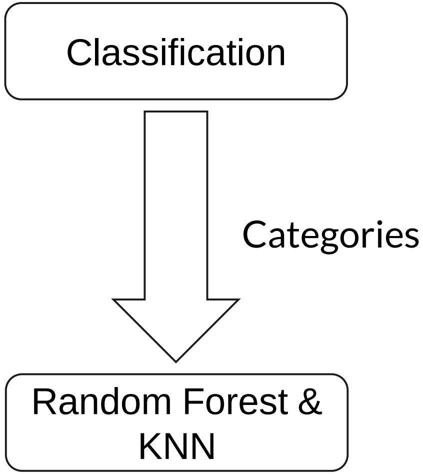
222465 rows × 312 columns

Feature Selection

✗ Working on all features → high computational & executed for the models

✓ Select features for each model based on embedded method by RandomForestRegressor by (n_estimators = 50)

Classification model:



Classification

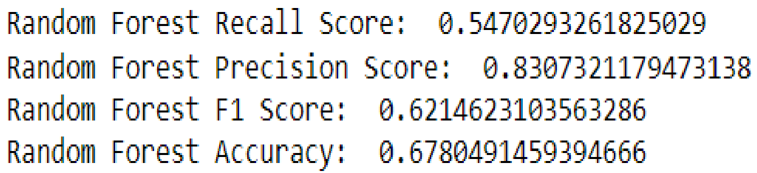
Categories

Random Forest & KNN

Results

Classification by Random Forest

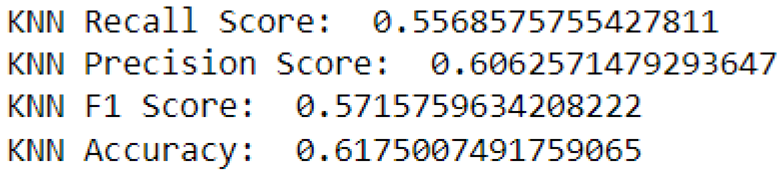
Hyperparameter tuning:
{'n_estimators': 150}



```
Random Forest Recall Score:  0.5470293261825029
Random Forest Precision Score:  0.8307321179473138
Random Forest F1 Score:  0.6214623103563286
Random Forest Accuracy:  0.6780491459394666
```

Classification by KNN

With k = 3



```
KNN Recall Score:  0.5568575755427811
KNN Precision Score:  0.6062571479293647
KNN F1 Score:  0.5715759634208222
KNN Accuracy:  0.6175007491759065
```

Limitations

- Very huge dataset, so our hardwares couldn’t deal with (Memory crashing)
- Much time in each algorithm
- Data is updated daily

Conclusion

- Data exploration or EDA is a good step to understand the data more.
- Data cleaning, and feature extraction are an important steps, and have high effect on the results.
- Choosing the hyperparameters effect on the model performance.

Thank You