Data analysis on K.J YouTuber

Dan Hua Li

12/7/2022

This project attempt to help better understand the growth of YouTube community and find the best ways that as a data scientist how to improve the benefit for their Youtuber in data science area. Definitely, the best ways to be a successful YouTuber is to tell a wonderful story and work hard on the content. However, here I just ignore the content of Ken Jee's YouTube but focus on the title, video's length and I assume those are the important factors that impact the profit. Hopefully, through this project may help making profit on our channel at the beginning.

This project objective will try to answer the follow questions:

- what is the outlook of data science, is it popular?
- What types of video titles drive the most traffic?
- Does it exist a appropriate length of video that could help maximizing the profit?

This project will combine the tools of Excel, Python and Weka, to envision, execute, and summarize above issues based on a data-science-oriented study. Processing data includes:

- Data description
  - Data Background information
  - Data Dictionary
  - Missing values
- Data mining process
  - data cleansing,
  - attribute selection
  - transformation,
  - training and testing process (10/5-fold cross-validation).
  - Linear Regression model
- Final Results and Recommendation

Data Description

Data Background information: The data for this project is loading from Ken's Kaggle, a famous YouTuber in data science, who provided his personal YouTube data

for analysis. Notice: the study is only based on the data science YouTuber and the limitation of Ken Jee's Private YouTube Data source. The data set I selected includes two parts:

1) Aggregated Metrics By Video with Country and Subscriber Status
   - data includes dimensions for which country people are viewing from and if the viewers are subscribed to the channel or not. - Attributes:15 instances: 55292

2) Aggregated Metrics By Video
- includes all the topline metrics from the channel from its start (around 2015 to Jan 22 2022). There are 111857 original records and group it into 224 records.

Data Dictionary: - I only select the useful variables as bellowing.

	variable	*type*	*description*
1	Title	Nominal	The title of the videos
2	Video Length	Numeric	The Length of the video
	Country code	Nominal	The country code
3	Shares	Numeric	the number of the viewers share the video
4	Dis-likes	Numeric	The number of the viewers dislike the video
5	Likes	Numeric	The number of the viewers like the video
6	Subscribers lost	Numeric	The number of subscribers lost
7	Subscribers gained	Numeric	The number of subscribers gained
8	RPM (USD)	Numeric	Revenue Per Mille (RPM) is a metric that represents how much money you have earned per 1,000 video views
9	CPM (USD)	Numeric	The estimated gross revenue per thousand ad impressions
10	Average percentage viewed %	Numeric	The average percentage of a video watched during a video playback
11	Views	Numeric	The number of times the viewers watch the video
12	Watch time(hours)	Numeric	the number of hours the viewers watch the video
	Average Watch time	Numeric	The average hours of a video watched
13	Sub-scribers	Numeric	The numbers of subscribers (Subscribers gained) - (Subscribers lost)
14	Your estimated revenue(USD)	Numeric	The total estimated net revenue from all Google-sold advertising sources as well as from non-advertising sources for the selected date range and region.
15	Lm-pressions	Numeric	How many times thumbnails were shown to viewers on YouTube through registered impressions

Missing Values --> Data Cleansing

```
video_df=pd.read_csv('Aggregated_Metrics_By_Video.csv')
video_df.isnull().sum()
```

```
)5]: Video                                0
     Video title                          1
     Video publish time                   1
     Comments added                       0
     Shares                               0
     Dislikes                             0
     Likes                                0
     Subscribers lost                     0
     Subscribers gained                   0
     RPM (USD)                            0
     CPM (USD)                            2
     Average percentage viewed (%)        0
     Average view duration                0
     Views                                0
     Watch time (hours)                   0
     Subscribers                          0
     Your estimated revenue (USD)         0
     Impressions                          0
     Impressions click-through rate (%)   0
     dtype: int64
```

```
video2_df= video_df.dropna()
video2_df.isnull().sum()
```

```
0]: Video                                0
    Video title                          0
    Video publish time                   0
    Comments added                       0
    Shares                               0
    Dislikes                             0
    Likes                                0
    Subscribers lost                     0
    Subscribers gained                   0
    RPM (USD)                            0
    CPM (USD)                            0
    Average percentage viewed (%)        0
    Average view duration                0
    Views                                0
    Watch time (hours)                   0
    Subscribers                          0
    Your estimated revenue (USD)         0
    Impressions                          0
    Impressions click-through rate (%)   0
    dtype: int64
```

1. outlook of data science

To answer the question 1. what is the outlook of data science, is it popular? I select the #1 dataset. Then implement: Format the country code, visualize the result.

Already imported pandas, csv, os and installed plotly, pycountry by "pip".

```
Country_df = pd.read_csv('Data_Aggregated_Metrics_By_Country_And_Subscriber_Status.csv')
```

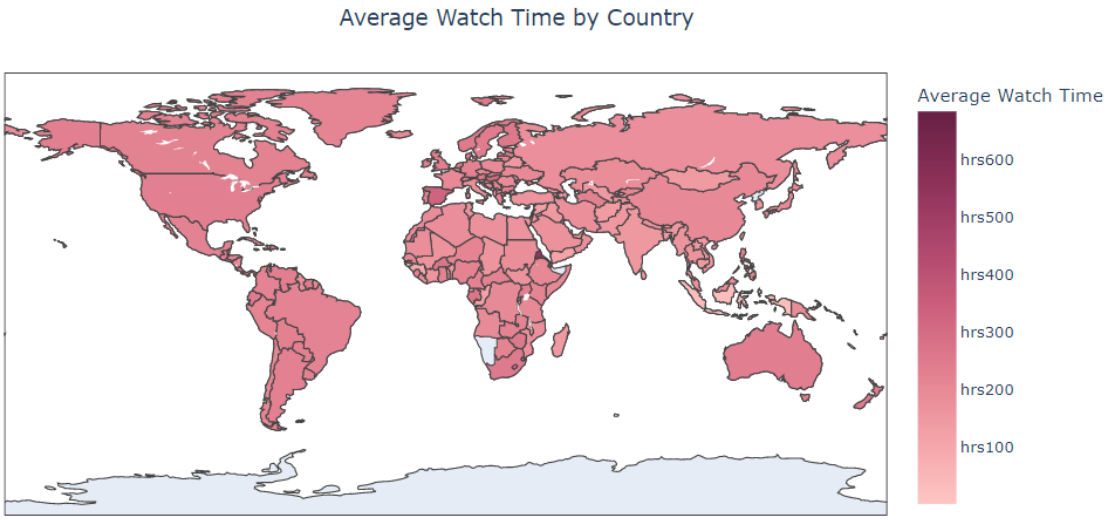
Format the country code.

```
import pycountry
def do_fuzzy_search(country):
    try:
        result = pycountry.countries.search_fuzzy(country)
    except Exception:
        return np.nan
    else:
        return result[0].alpha_3

iso_map = {country: do_fuzzy_search(country) for country in Country_df["Country Code"].unique()}

Country_df["Country_Code"] = Country_df["Country Code"].map(iso_map)

Country_df = Country_df.loc[~(Country_df['Country_Code'].isna()),]

GIS_plot_df = Country_df.groupby(by=['Country_Code', 'Country Code'], as_index=False, dropna=True).mean()
GIS_plot_df.head()
```

	Country_Code	Country Code	Video Length	Is Subscribed	Views	Video Likes Added	Video Dislikes Added	Video Likes Removed	User Subscriptions Added	User Subscriptions Removed	A Perc
0	ABW	AW	700.216867	0.361446	3.036145	0.048193	0.000000	0.000000	0.132530	0.000000	0.3
1	AFG	AF	746.267857	0.422619	5.160714	0.184524	0.011905	0.011905	0.202381	0.017857	0.2
2	AGO	AO	912.624204	0.401274	5.210191	0.184713	0.012739	0.000000	0.171975	0.006369	0.3
3	ALA	AX	490.285714	0.000000	1.857143	0.000000	0.000000	0.000000	0.000000	0.000000	0.5
4	ALB	AL	906.320423	0.461268	9.728873	0.274648	0.035211	0.014085	0.123239	0.010563	0.2
...	...	...	...	...	...	...	...	...	...	...	

To analyze videos in geographical way

```
import plotly.graph_objects as go
fig = go.Figure(data=go.Choropleth(locations = GIS_plot_df['Country_Code'],
                                   z = GIS_plot_df['Average Watch Time'],
                                   text = GIS_plot_df['Country Code'],
                                   colorscale = 'burg',
                                   colorbar_tickprefix = 'hrs',
                                   colorbar_title = 'Average Watch Time')
)
fig.update_layout(
    title={'text': 'Average Watch Time by Country',
           'y':0.9,
           'x':0.5,
           'xanchor': 'center',
           'yanchor': 'top'})

fig.show()
```



Recall Ken Jee's YouTube channel: there are 223k subscribers, and approximately 200 videos about data science. How many hours audiences from all over the world are watching his videos averagely? The deeper red color present the more hours people from different countries spent time watching his data science video. Through analyzing videos in geographical way, visualizing the outlook of data science, it is easy to know data science learning became popular now.

2. Analyze the title of video

To find the solution for question 2: What types of video titles drive the most traffic? First, we need to fight out what attributions related to title. The **Views** should be correlated to **Like**.

People view the video may not like it But they will search and view the fancy title, such actions could increase the Like. So to target the fancy titles, We should start from catching videos with the largest amount of **Views**. The Video Length gives a picture of cost for producing video (use it later).

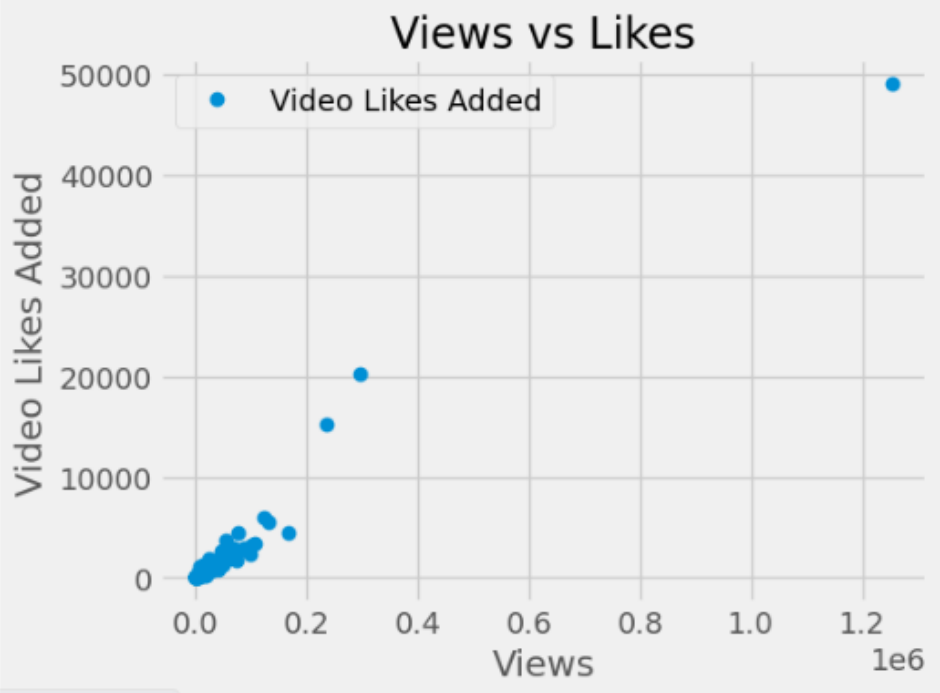
Let check the relationship between ***View*** and ***Like added*** by using the previous data set and select the variables sort it into *video view_like.csv* by using Excel.

```
import pandas as pd
import statsmodels.api as sm
dataset = pd.read_csv('video view_like.csv')
dataset.head()
```

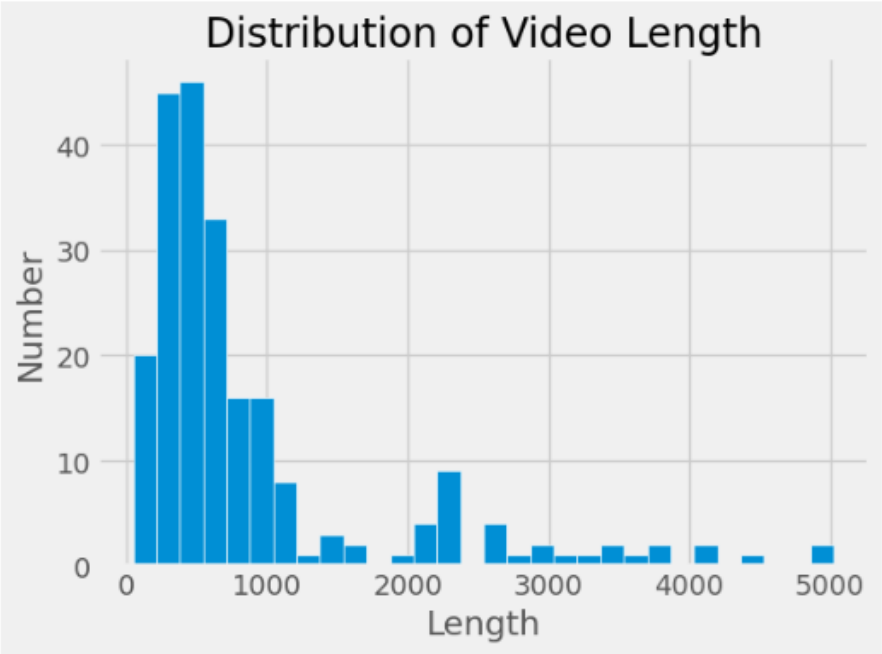
	Video Title	Video Length	Views	Video Likes Added
0	Hot Topics in Tech: Data Science Explained #SH...	59	8003	409
1	git for Data Science Made Simple... (Hopefully)	392	12629	667
2	Work From Home Data Scientist: Day in the Life	331	26582	754
3	Why is Balance Important in Data Science?	238	612	33
4	Why are APIs Important for Data Science?	322	6537	363

```
import matplotlib.pyplot as plt
%matplotlib inline
```

```
plt.style.use('fivethirtyeight')
dataset.plot(x='Views', y='Video Likes Added', style='o')
plt.title('Views vs Likes')
plt.xlabel('Views')
plt.ylabel('Video Likes Added')
plt.tight_layout()
plt.show()
```



```
plt.hist(dataset["Video Length"],bins=30,edgecolor='white')
plt.title('Distribution of Video Length')
plt.xlabel('Length')
plt.ylabel('Number')
plt.tight_layout()
plt.show()
```



Above charts illustrate that Using the value of ***View*** to find the fancy title is more than enough instead of combining both *View* and *Like Added* because they are correlated. The second chart appears most of video length are short and the median is around 548' (9.1 mins). It is not necessarily to make a long video to attract users' attention. If you dig deeply into the data, you will find out around 25% videos have only 1 minutes length.

Then what is the import factor that catch the users' attention? The title of video might play a main role. The characteristics of words in the title also provide some important information for targeting the users. Implement: Group the data by the Title of Video in Excel, sort the data into ascend and descend order by view in Excel, Check the missing Data and Data Cleansing and in Python, visualize the top_15 hot and freeze_15 title of videos in Python, analyze the character of words.

Next, I select only ***Title*** and ***View*** variables, group it by Title, and sort the order by the value of **View** in Excel. I use DictReader, turn the data set into a dictionary for easily getting top_15 hot and freeze title by its keys.

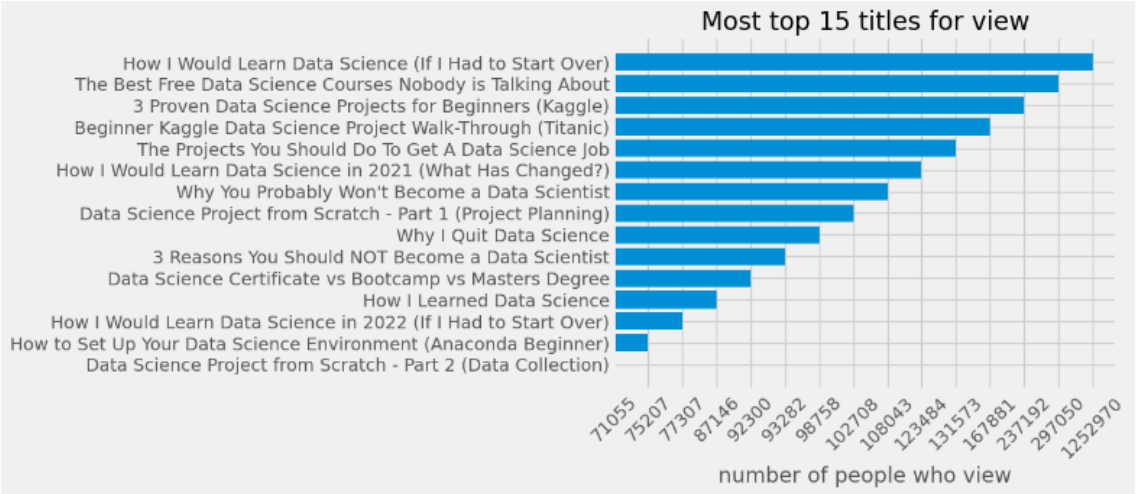
```
import csv
csvfile=open('data_title_views.csv','r')
data_title_views={}
for row in csv.DictReader(csvfile):
    #print(row)
    data_title_views[row['Title']]=row['Views']
    #print(data_title_views.keys())
print(data_title_views)
```

```
{'How I Would Learn Data Science (If I Had to Start Over)': '1252970', 'The Best Free Data Science
Courses Nobody is Talking About': '297050', '3 Proven Data Science Projects for Beginners (Kaggl
e)': '237192', 'Beginner Kaggle Data Science Project Walk-Through (Titanic)': '167881', 'The Projec
ts You Should Do To Get A Data Science Job': '131573', 'How I Would Learn Data Science in 2021 (Wha
t Has Changed?)': '123484', "Why You Probably Won't Become a Data Scientist": '108043', 'Data Scien
ce Project from Scratch - Part 1 (Project Planning)': '102708', 'Why I Quit Data Science': '98758',
'3 Reasons You Should NOT Become a Data Scientist': '93282', 'Data Science Certificate vs Bootcamp
vs Masters Degree': '92300', 'How I Learned Data Science': '87146', 'How I Would Learn Data Science
in 2022 (If I Had to Start Over)': '77307', 'How to Set Up Your Data Science Environment (Anaconda
Beginner)': '75207', 'Data Science Project from Scratch - Part 2 (Data Collection)': '71055', 'Is D
```

```
#turn the dic into two list: one for keys another for value. later use them for plot
title_list=list(data_title_views.keys())[0:15]
views_list=list(data_title_views.values())[0:15]
print(title_list)
print(views_list)
```

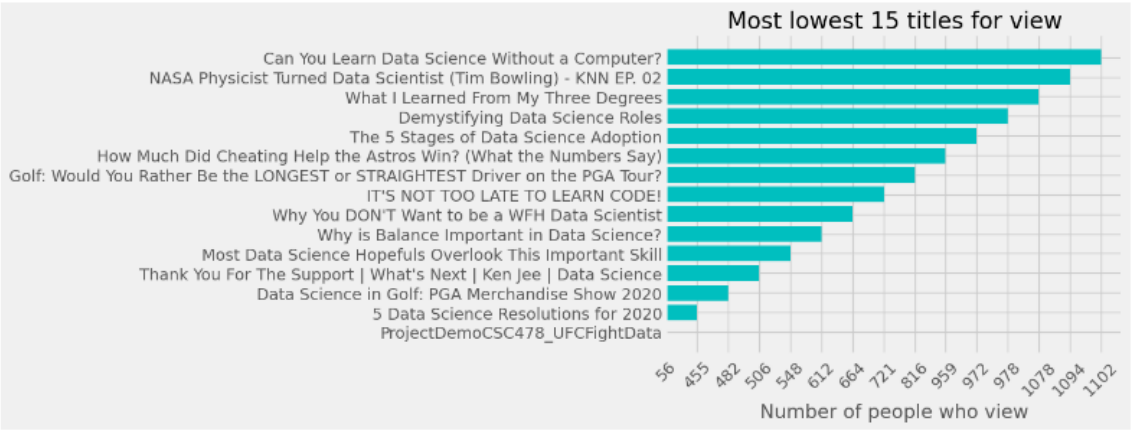
```
['How I Would Learn Data Science (If I Had to Start Over)', 'The Best Free Data Science Courses Nobod
y is Talking About', '3 Proven Data Science Projects for Beginners (Kaggle)', 'Beginner Kaggle Data S
cience Project Walk-Through (Titanic)', 'The Projects You Should Do To Get A Data Science Job', 'How
I Would Learn Data Science in 2021 (What Has Changed?)', "Why You Probably Won't Become a Data Scient
ist", 'Data Science Project from Scratch - Part 1 (Project Planning)', 'Why I Quit Data Science', '3
Reasons You Should NOT Become a Data Scientist', 'Data Science Certificate vs Bootcamp vs Masters Deg
ree', 'How I Learned Data Science', 'How I Would Learn Data Science in 2022 (If I Had to Start Ove
r)', 'How to Set Up Your Data Science Environment (Anaconda Beginner)', 'Data Science Project from Sc
ratch - Part 2 (Data Collection)']
['1252970', '297050', '237192', '167881', '131573', '123484', '108043', '102708', '98758', '93282',
'92300', '87146', '77307', '75207', '71055']
```

```
title_list.reverse()
views_list.reverse()
plt.barh(title_list,views_list)
plt.title('Most top 15 titles for view')
plt.xlabel('number of people who view')
plt.xticks(rotation=45)
plt.show()
```




```
# get the freeze-15 titles people don't like to view
title_list2=list(data_title_views.keys())[-15:]
views_list2=list(data_title_views.values())[-15:]
```

```
title_list2.reverse()
views_list2.reverse()
plt.barh(title_list2,width=views_list2,color='c')
plt.title('Most lowest 15 titles for view')
plt.xlabel('Number of people who view')
plt.xticks(rotation=45)
plt.show()
```



The difference in titles would rise 10,000 times variability on ***views,*** even though the videos are all about the data science. Next, I will check the frequency of words appear in top-50 and freeze-50 titles and analyze the result and what information they provide.

```
from sklearn.decomposition import LatentDirichletAllocation, TruncatedSVD
from sklearn.feature_extraction.text import CountVectorizer, TfidfVectorizer
from sklearn.model_selection import GridSearchCV
from pprint import pprint

from spacy.cli import download
download("en_core_web_sm")
import re, nltk, spacy, gensim
import matplotlib.pyplot as plt
%matplotlib inline
```

```
df = pd.read_csv('data_title_views.csv')
def sent_to_words(sentences):
    for sent in sentences:
        sent = re.sub("\'", "", sent)  # remove single quotes
        sent = gensim.utils.simple_preprocess(str(sent), deacc=True)
        yield(sent)

# Convert to list
data_words = list(sent_to_words(list(data_title_views.keys())[0:50]))
print(data_words)
```

```
[['how', 'would', 'learn', 'data', 'science', 'if', 'had', 'to', 'start', 'over'], ['the', 'best', 'f
ree', 'data', 'science', 'courses', 'nobody', 'is', 'talking', 'about'], ['proven', 'data', 'scienc
e', 'projects', 'for', 'beginners', 'kaggle'], ['beginner', 'kaggle', 'data', 'science', 'project',
'walk', 'through', 'titanic'], ['the', 'projects', 'you', 'should', 'do', 'to', 'get', 'data', 'scien
ce', 'job'], ['how', 'would', 'learn', 'data', 'science', 'in', 'what', 'has', 'changed'], ['why', 'y
ou', 'probably', 'wont', 'become', 'data', 'scientist'], ['data', 'science', 'project', 'from', 'scra
tch', 'part', 'project', 'planning'], ['why', 'quit', 'data', 'science'], ['reasons', 'you', 'shoul
d', 'not', 'become', 'data', 'scientist'], ['data', 'science', 'certificate', 'vs', 'bootcamp', 'vs',
'masters', 'degree'], ['how', 'learned', 'data', 'science'], ['how', 'would', 'learn', 'data', 'scien
ce', 'in', 'if', 'had', 'to', 'start', 'over'], ['how', 'to', 'set', 'up', 'your', 'data', 'science',
```

```
def lemmatization(texts, allowed_postags=['NOUN', 'ADJ', 'VERB', 'ADV','WH','FW']): #'NOUN', 'ADJ', 'VE
    texts_out = []
    for sent in texts:
        doc = nlp(" ".join(sent))
        texts_out.append(" ".join([token.lemma_ if token.lemma_ not in ['-PRON-'] else '' for token in
    return texts_out
```

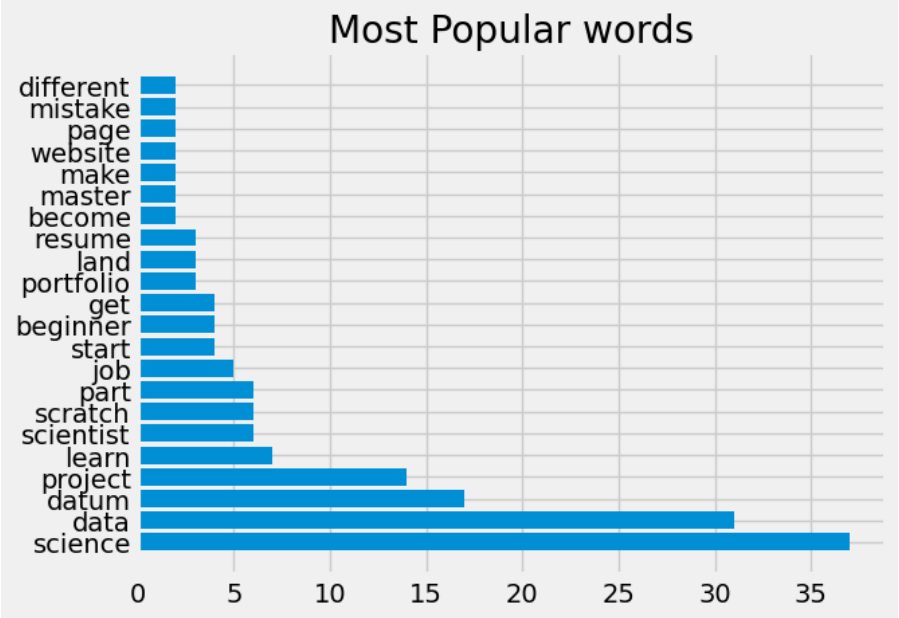
```
nlp = spacy.load('en_core_web_sm', disable=['parser', 'ner'])

# Do lemmatization keeping only Noun, Adj, Verb, Adverb, WH, FW
data_lemmatized = lemmatization(data_words, allowed_postags=['NOUN', 'ADJ', 'VERB', 'ADV','WH','FW'])

print(data_lemmatized)
```

```
['learn data science start', 'well free datum science course talk', 'prove data science project begin
ner kaggle', 'beginner science project walk titanic', 'project get datum science job', 'learn data sc
ience change', 'probably become data scientist', 'data science project scratch part project plannin
g', 'quit datum science', 'reason become data scientist', 'data science certificate vs bootcamp maste
r degree', 'learn data science', 'learn data science start', 'set data science environment', 'data sc
ience project scratch part datum collection', 'data science die', 'make data science portfolio websit
e page', 'datum science advice college student', 'ultralearn data science', 'daysofdata', 'essential
datum science project portfolio', 'land sport analytic job', 'data science project scratch part datum
```

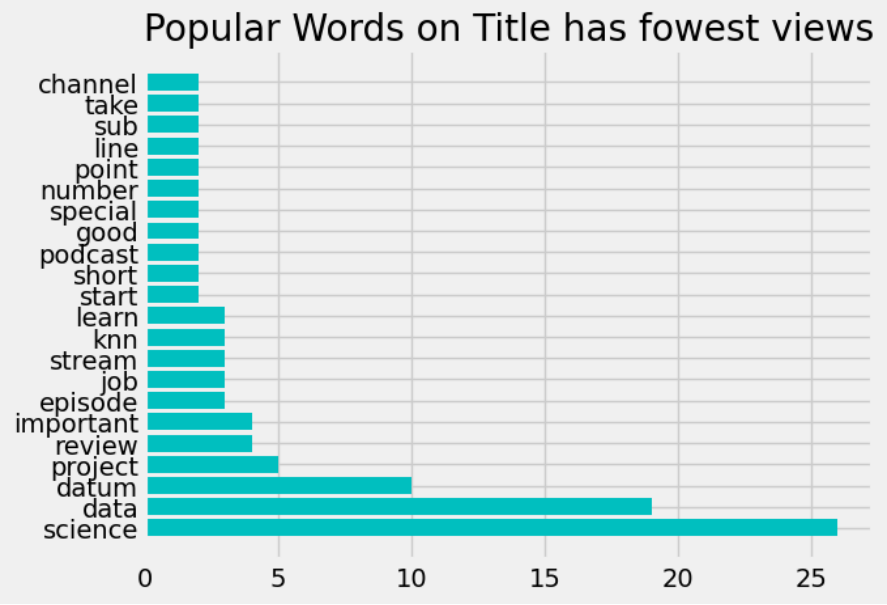
```
# count the frequency of words
from collections import Counter
words_counter=Counter()
for row in data_lemmatized:
    words_counter.update(row.split(' '))
print(words_counter)
```

```
Counter({'science': 37, 'data': 31, 'datum': 17, 'project': 14, 'learn': 7, 'scientist': 6, 'scratc
h': 6, 'part': 6, 'job': 5, 'start': 4, 'beginner': 4, 'get': 4, 'portfolio': 3, 'land': 3, 'resume':
3, 'become': 2, 'master': 2, 'make': 2, 'website': 2, 'page': 2, 'mistake': 2, 'different': 2, 'buil
d': 2, 'analyst': 2, 'review': 2, 'episode': 2, 'first': 2, 'well': 1, 'free': 1, 'course': 1, 'tal
k': 1, 'prove': 1, 'kaggle': 1, 'walk': 1, 'titanic': 1, 'change': 1, 'probably': 1, 'planning': 1,
```

```
words=[]
popularity=[]
for item in words_counter.most_common(22):
    words.append(item[0])
    popularity.append(item[1])
plt.style.use('fivethirtyeight')
plt.barh(words, popularity)
plt.title('Most Popular words')
#plt.xticks(rotation=45)
plt.show()
```



With the same procedure and similar codes, we get the frequency of words in the most freeze_50 titles.



Notice: the word “datum” has been incorrectly identified and it should be “data”.

'data', 'science' are using frequently in both group, But 'beginner', 'resume' appear in hot title frequently. Above 2 charts provide a very important information that such YouTube channels should position the marketing target toward the beginner of data science learning. '

Find the model for revenue in Weka

We already analyzed the title that impact the Views. Here, we will build the model for Revenue. So we have to combine two dataset: one for the relationship between **View** and **Length**. Because the Video Length determines the cost of production and it will directly impact the Profit (Profit=revenue- cost).

By Selecting the variables from previous CSV #2 - (Attributes:14 instances: 111857) find the relationship and weight between the Views, Length…

```
Average View Percentage =

     -0.0002 * Video Length +
     -0.0001 * Views +
      0.0009 * Video Likes Added +
      0.0024 * Video Dislikes Added +
     -0.0015 * Video Likes Removed +
      0.0002 * User Subscriptions Added +
      0.0008 * Average Watch Time +
      0.3465

Time taken to build model: 1.22 seconds

=== Cross-validation ===
=== Summary ===

Correlation coefficient                  0.7248
Mean absolute error                      0.1152
Root mean squared error                  0.1614
Relative absolute error                 63.9947 %
Root relative squared error             68.8905 %
Total Number of Instances           111857
```

Method: Linear Regression 5-flod VC: The variable Video Length has a ***negative*** weight to the ***Average View Percentage***, which is quite important variable to Revenue. (show the Revenue model next)

```
Video Likes Added =

     -0.0002 * Video Length +
      0.0382 * Views +
     -1.8252 * Video Dislikes Added +
      1.3524 * Video Likes Removed +
      0.0793 * User Subscriptions Added +
      5.8603 * User Subscriptions Removed +
      0.0692

Time taken to build model: 0.11 seconds

=== Cross-validation ===
=== Summary ===

Correlation coefficient                  0.948
Mean absolute error                      1.0138
Root mean squared error                  4.5416
Relative absolute error                 33.8242 %
Root relative squared error             31.84   %
Total Number of Instances          111857
```

```
User Subscriptions Added =

     -0.0001 * Video Length +
      0.0383 * Views +
      0.0882 * Video Likes Added +
     -0.9588 * Video Dislikes Added +
     -0.2585 * Video Likes Removed +
     -7.6047 * User Subscriptions Removed +
     -0.5641

Time taken to build model: 0.11 seconds

=== Cross-validation ===
=== Summary ===

Correlation coefficient                  0.9162
Mean absolute error                      1.0904
Root mean squared error                  4.8521
Relative absolute error                 59.0371 %
Root relative squared error             40.0696 %
Total Number of Instances          111857
```

Conclusion: Video-Like-Added and Subscriptions Added are not **positively** affected by the video length. Oppositely, the more Length of video, the worse Like and Subscription Added and they are only impacted by the content of the videos and the amount of Views. Further more, the longer video length, the higher cost of production.

Known that **Median** for video length : 548’ (**9.1** minutes) from Excel. We will check the important variables weather will affect the revenue.

```
Test mode:    5-fold cross-validation

=== Classifier model (full training set) ===


Linear Regression Model

Your es-tim-ated revenue (USD) =

      0.0663 * Com-ments ad-ded +
     -0.0073 * Shares +
     -0.0517 * Likes +
     -0.1123 * Sub-scribers lost +
      0.0222 * Sub-scribers gained +
     11.8513 * RPM (USD) +
      0.7142 * Av-er-age percent-age viewed (%) +
      0.0037 * Views +
      0.0198 * Watch time (hours) +
      0.0405 * Sub-scribers +
      0.0001 * Im-pressions +
    -78.6919

Time taken to build model: 0 seconds

=== Cross-validation ===
=== Summary ===

Correlation coefficient                  0.9836
Mean absolute error                    207.9129
Root mean squared error               2551.6025
Relative absolute error                 53.8232 %
Root relative squared error            126.5491 %
Total Number of Instances              224
```

The model select RPM as high weight variable(11.9). But RPM is the policy of paying to videos and this policy won't be revealed in detail. Let's try to find the rule from this data set. The result the model bellow makes no sense to get to know it by our limited data set.

```
RPM (USD) =

     -0.0002 * Shares +
      0.0071 * Dis-likes +
     -0      * Likes +
     -0.0003 * Sub-scribers lost +
      0.2638 * CPM (USD) +
     -0.0208 * Av-er-age percent-age viewed (%) +
      0      * Watch time (hours) +
      2.0667

Time taken to build model: 0 seconds

=== Cross-validation ===
=== Summary ===

Correlation coefficient                  0.272
Mean absolute error                      1.2284
Root mean squared error                  2.114
Relative absolute error                 89.5276 %
Root relative squared error            118.2189 %
Total Number of Instances              224
```

Other than RPM variables, the second important variable is **Average Percentage View**, and as the first model shows that the **Length** of video did not help to improve it but increase the cost of production logically. So it means that with a perfect content, a shorter video possibly take the advantage on revenue.

Except the variables related to the content of video, the **Views** is the important factor to impact the Revenue. Even thought the View has only 0.0037 weight but it has huge amount of value(shows previously: it is more than 100,000 times than other variables' value, such as Subscription, Likes…). But don't forget that **Views** also has a negative weight of 0.0001 to **Average Percentage View**.

Finally, due to the unknow and vary attribution RPM, further discussing the accuracy of this model become meaningless. But the model provide us some important information and I recommend:

- Design the length of video depended on the cost you can pay. Understand a long video is not a necessary condition to gain profit.
- It is a wise way to divide a long video into few short videos with integrity of

knowledge point in each video. a long video might possibly decrease the **Average Percentage View.** (coef= - 0.002 (Length to Average view %))

- Design the title of video according to Precisely target audiences.
- Making a video with good content should not be ignored. You will get 'penalty' on **Subscription Lost**. (coef= -0.1 (Lost subscription to revenue))

The link of K.J YouTube Data:
https://www.kaggle.com/datasets/kenjee/ken-jee-youtube-data?resource=download