Machine Learning

on YouTube Development Strategy Analysis

Dan Hua Li

This project attempt to solve the business problems: How to start a new YouTube channel in a certain field, and how to design the strategies to target the market. Before that, we need to understand the growth of the YouTube community and what is key factors that gain revenue. From the corrent information, a YouTuber's revenue come from two indexs: Views and Impression. It means how many people watch our videos and how many good comments or like we got then those are important factors that determine our revenue.

Definitely, the best ways to be a successful YouTuber is to tell a wonderful story and work hard on the content. But how to define a good or bad video? Most of tme It depended on the audiences. For example. We invest a lot to make an advanced science video in the P.H.D field. But scientists are very few and rarely learn such science by watch YouTube. So determining our target market will be the most important. Meantime, without a wonderful title , even though a good video will never reach our audience by google or searching.

The objective of this project is to apply the machine learning techniques, help to find the target market through:

- Text Minning on impression content
- Text minning on titles

The key machine learning techniques we used are:

- LSA / LDA Word Embeddings in Comments Topic Modeling
- Grid Search / Gensim Search for best Topic
- Average Coherence Score Analysis:
  - LDA
  - U_MASS
  - C_UCI
  - C_NPMI
- Text Ming on Titles : Find the Frequent Words

Data Process including :

1. Data description

    - Background information

- Data Dictionary
- Missing values

2. Data Mining on Comments
   - LSA topic extraction
   - NMF topic extraction
   - LDA topic extraction
   - sklearn grid search: best number of topics
   - Average Coherency Score

3. Text Ming on Titles
   - Data cleansing,
   - Data Transforming
   - Visualization

4. ML weaknesses
5. Result and Recommendation

1. Data Description

Background information: The data set is loaded from Ken's Kaggle, a famous YouTuber in data science. The link : https://www.kaggle.com/datasets/kenjee/ken-jee-youtube-data . Data size: 7 MB, The dataset includes four parts: We selected two useful parts:

1) Aggregated Metrics By Video with Country and Subscriber Status
   - data includes dimensions for which country people are viewing - Attributes:15 instances: 55292
2) Aggregated Metrics By Video
   - includes all the topline metrics from the channel start from 2015 to Jan 22, 2022. There are 111857 original records grouped by titles into 224 records. Selected Comments as one demension dataset with 10217 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
16	Comments	Nominal	Comments on videos

outlook of data science

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

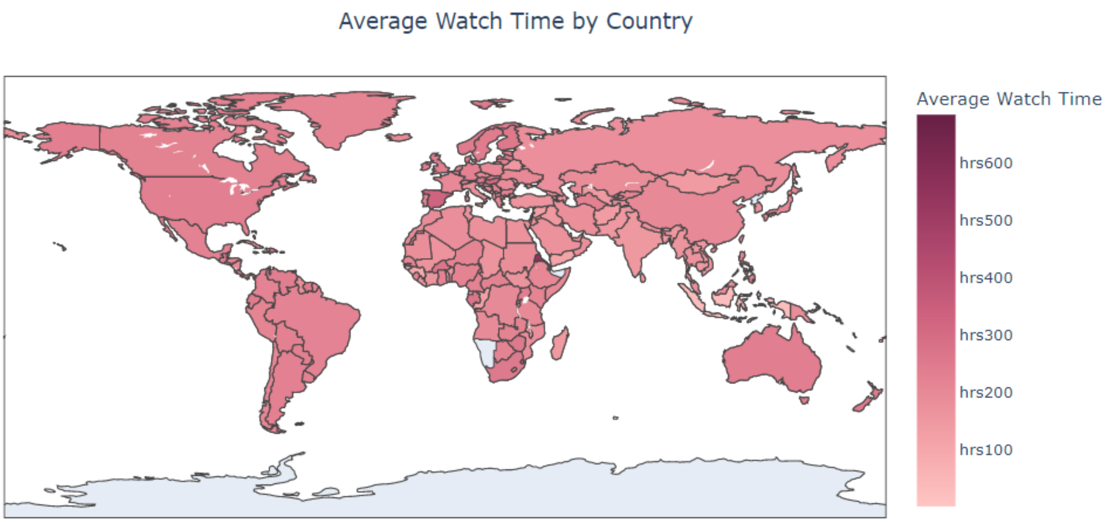
Already imported pandas, csv, os and installed plotly, pycountry.

Format the country code.

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

```
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.:
1	AFG	AF	746.267857	0.422619	5.160714	0.184524	0.011905	0.011905	0.202381	0.017857	0.:
2	AGO	AO	912.624204	0.401274	5.210191	0.184713	0.012739	0.000000	0.171975	0.006369	0.:
3	ALA	AX	490.285714	0.000000	1.857143	0.000000	0.000000	0.000000	0.000000	0.000000	0.!
4	ALB	AL	906.320423	0.461268	9.728873	0.274648	0.035211	0.014085	0.123239	0.010563	0.:
...	...	...	...	...	...	...	...	...	...	...	



Recall K. J YouTube channel: there are 223,000 subscribers, and over 200 videos about data science. How many hours audiences from all over the world are watching his videos on average? The deeper red color presents hours people from different countries spent time watching his video. Through analyzing videos in geographical way, visualizing the outlook of data science, know that data science became popular now.

Text Minning in Comments

1. ***Features Selection in Excel***
2. ***Data Cleaning in Excel (have to)***
   - https, youtube.com, github, .com (impact frequent words)
   - symbols from Emoticons (make insert data into python difficultly)
   - different language..
   - special symbols.

3. **Data Import** : pd.read_csv('Clean_comments-UTF8.csv',encoding='utf-8', sep=',')

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10217 entries, 0 to 10216
Data columns (total 1 columns):
 #   Column   Non-Null Count  Dtype 
---  ------   --------------  ----- 
 0   Reviews  10217 non-null  object
dtypes: object(1)
memory usage: 79.9+ KB
```

	Reviews
0	Thanks fucken.\n\nI decided to go into Tech in...
1	Hello ken jee!!! I'm doing a graduation on Com...
2	Thanks fuck for
3	Great video!!! I started learning Python 8 mon...
4	Been watching hours ofucknow that it is not an...

4. **Def a lemmatize corpus function**

- Remove punctuation
- Lowercase for all words
- Use nltk's English stopwords list
- add 'ha, hey, hi, woo, wa' ,to stopword list for removal

5. **convert sparse to dense matrix**:

10212	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0
10213	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0
10214	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0
10215	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0
10216	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0

10217 rows × 11884 columns

6. Use TfidfVectorizer gives score value to it

- Base on term-frequency' and 'inverse document frequency' statistics
- Convert a collection of raw documents to a matrix of TF-IDF features.
- Apply lemmatize corpus function (corpus, stopword, lowercase)

7. **LSA**: **linear dimensionality (not CPA)**

- *Lsa = TruncatedSVD (n_components=10)*
- *fit*

```
lsa_tf_topics = lsa.fit_transform(tf_sparse)
lsa_tf_topics.shape
```

```
(10217, 10)
```

```
lsa.components_.shape
```

```
(10, 11884)
```

After LSA Linear dimensionality, There are 10,217 rows and 10 components will be selected among 11,884 columns (words).

8. LSA Topics Result

We only try to find what kind of contents audiences like in their comments instead o f sensitive analyzing those tipics and indicating negative or possitive emotions. The result shows people like Data science in **Project, Master field**, **deep machine learning,** and **jobs**.

```
print_top_terms(lsa, tf_dictionary, 10)
```

```
LSA topics based on term-document matrix:
Topic #0: data science ken video fuck im learning great would project
Topic #1: data science scientist machine degree job computer analyst master field
Topic #2: fuck im fucke ofuck hi get project learning much know
Topic #3: video fuck im really learning like great get would ofuck
Topic #4: im project really get learning would one like good fucke
Topic #5: project thanks data scientist one great learning would question model
Topic #6: science learning thanks project machine course computer master learn deep
Topic #7: great learning would thank machine really one ken like content
Topic #8: learning video thank much get like learn machine time fucke
Topic #9: thanks learning great machine get lot really scientist fucke content
```

9. LDA Topics Result:

```
lda.fit_transform(tf_sparse)
print('LDA topics based on term-document matrix:')
print_top_terms(lda, tf_dictionary, 10)
```

```
LDA topics based on term-document matrix:
Topic #0: project use using like problem cool code deep idea model
Topic #1: topic bro recommendation list got 0 day life habit ,
Topic #2: data science im ken fuck learning learn get course would
Topic #3: resume excited , wait cant email necessary already sent 25
Topic #4: error getting tweet help line please first info element session
Topic #5: video ken great thanks fuck thank really much love content
Topic #6:  subscribed applying congrats „ 100k glad bring pas develop
Topic #7: nice youre lol ” seems check people na bit gon
Topic #8: comment company tell review like u 2 everyone ifuck see
Topic #9: link page discord name api episode column conversation drop indian
```

```
lda.fit_transform(tfidf_sparse)
print('LDA topics based on tfidf matrix:')
print_top_terms(lda, tfidf_dictionary, 10)
```

```
LDA topics based on tfidf matrix:
Topic #0: keyboard shirt eagerly john mouse kernel portfucken  snow de
Topic #1: pizza pc papaya monitor funny commenting dope hot pick hardware
Topic #2: honest discount annual lighting 365 titan super keen fuckenjee neural
Topic #3: helpfucks datacamp x excited period ì haircut soo mic voice
Topic #4: , brother error thanx csv import wall sa object 115
Topic #5: ken video data thanks fuck great science thank really im
Topic #6:  10k plant biomedical keyword silver pycharm greate vomit woo
Topic #7: thumbnail lmao gem absolute wink beautiful exact epic stormbreaker musk
Topic #8: subscribed congrats road extra lol “ 100k sub remotely fast
Topic #9: name green stufucken discord tea brilliant link secret congratulation algo
```

LDA topics based on term-document matrix and tfidf matrix are full of emotional contents. But Project and Resume content show up.

10. Search for best number of topics

- Sklearn grid search: Best number of topics, = 3, → not good.
- Gensim search: 5 → emotional words

11. Average coherence score →Maximize

measures how similar these words are to each other.

Average coherence score (LDA): -2.365187803749801 (short-text documents)

```
Average coherence score (c_v): 0.45430698132842046
Average coherence score (u_mass): -2.9950420939880367
Average coherence score (c_uci): -0.10625597414531388
Average coherence score (c_npmi): 0.019166505933121046
Model perplexity: -7.763722252439627
```

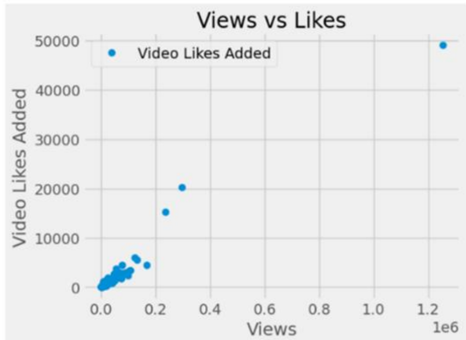
Text Minning on Titles

Why titles ?

- Views correlated to Like
- Views Definitely related to Titles
- Target: What titles drive the most traffic?
- Through most Views find the fancy Titles
- Visualize the most top_15 and freeze_15 Titles
- Analyze the frequent words of top_50 and freeze_50 titles

```
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()
```



We find out what attributions related to title. Through visualize the correlation of Views and Like. But it does mean audience like the titles will add Like to the video. People view the video may not like it But they will find the video by searching the title. That could increase the probability of giving Like to the videos. So to target the fancy titles, We should start from catching videos with the largest amount of Views. 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.

1. Import Data Feature selection : Group by titles (in 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

2. Data process

- Missing value → dropna

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

```
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()
```

```
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
```

3. Turn the dataset into Dictionary (for visualization)

```
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)
```

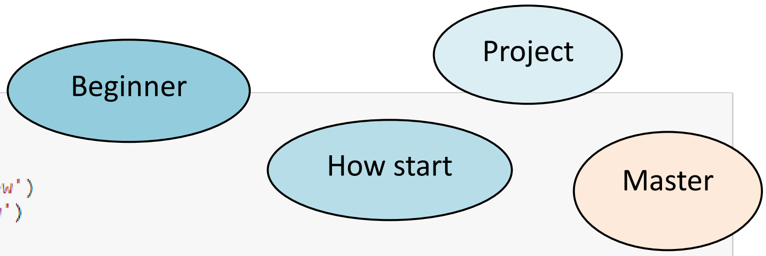
```
{'How I Would Learn Data Science (If I Had to Start Over)': '1252970', 'The Best Free Data Science
Courses Nobody is Talking About': '297050', '3 Proven Data Science Projects for Beginners (Kaggl
e)': '237192', 'Beginner Kaggle Data Science Project Walk-Through (Titanic)': '167881', 'The Projec
ts 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
```

4. Turn the dictionary into list through its keys

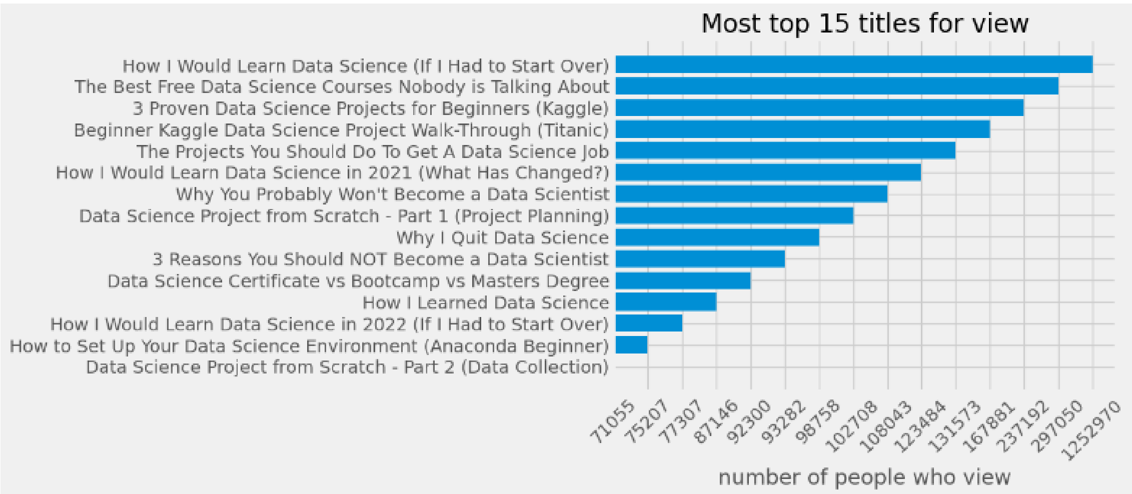
```python
#turn the dic into two list: one for keys another for value. later use them for plot
title_list=list(data_title_views.keys())[0:15]
views_list=list(data_title_views.values())[0:15]
print(title_list)
print(views_list)
```

```
['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']
```

5. Visualization




```python
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()
```



Visualize the top_15 hot and freeze_15 title of videos in Python, analyze the character of words. what is the important factors that catch the users' attention? The title of the video might play a main role. The top titles gained 1,252,970 Views.

The characteristics of words in the title also provide some important information for targeting the users. Such as "Beginner, how to start, Project, Master".

Implement: Group the data by the Title of Video in Excel, sort the data into ascend and descend order by view in Excel.

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

```
# 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:]
```

```
print(data_lemmatized)
```

```
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()
```

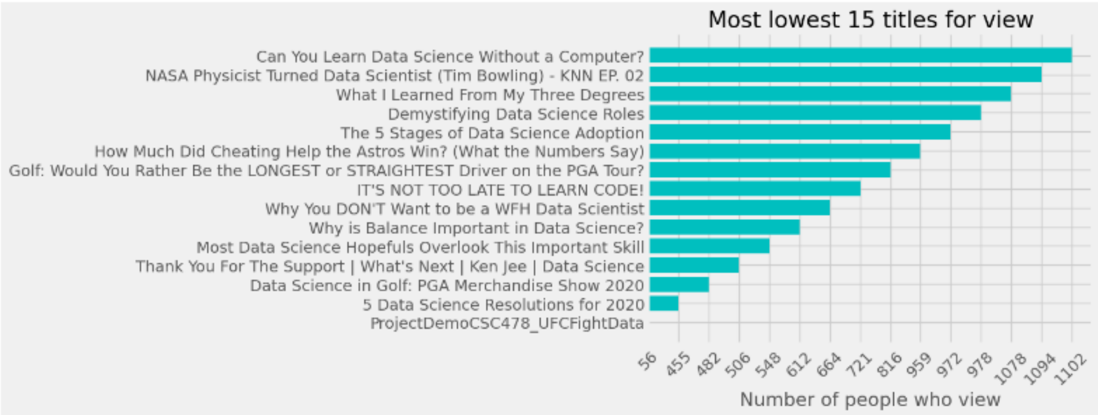
Computer

Resolution

Skills

Support

PGA



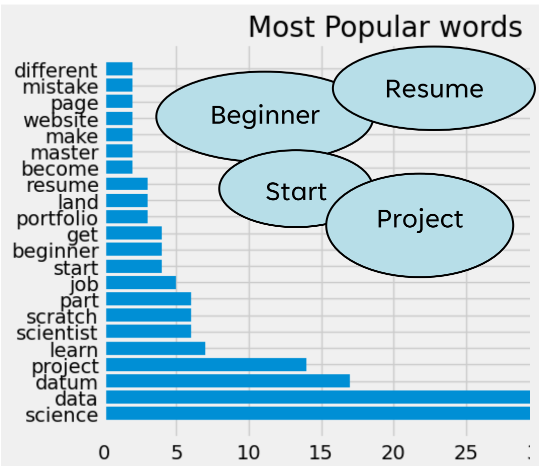
The lowest View of title only gained 56 Views. It seems those titles are not focused on popular questions in the data science area.

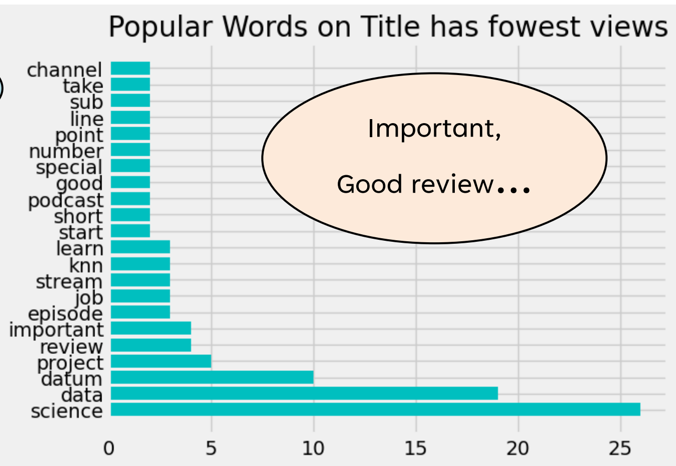
6. Text Mining on top 50 titles and freeze 50 titles .

convert to list, take out '[ ]' , Lemmatization

```
# 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,
```





Weakness:

1. Word Embeddings / Topics Modeling

- Similar subject
- Only LSA performance Good
- Other methods extract emotional content

2. Average Coherence Score:

Short Comments

Negative Score

Due to Topic Modeling using comments on same subject of data science, most of comments have similar content. ML techniques performan unexpectedly. Only LSA catch the content of comments well but it don't distinguish obversely among those tipics. LDA and NMP distinguish the topics by emotion and close to human nature.

When the comments are too short, Average coherence score appears negative value. Which lead to measuring the words similarity difficultly.

Result

1. Comments Topic Modeling :Extraction→ Impressive content

Target audiences: Who want to

- Do Data science (DS) Project
- Learn Master field in DS
- Learn deep Machine learning
- Find a job in DS

2. Title extraction :

Target audiences: Who are

- DS beginner
- Willing start DS learning
- Checking DS job
- Preparing Resume

Recommendation

When we start a YouTube channel, Our Target Market is very important. We determine our Target audience, Through ML technques start with

- Find a few famous YouTuber with similar subject
- Impressive content extraction from comments
- Analysis our Target audience and improve your video’s content

We need to design the video title according to precisely target audiences

- Through ML extract the frequent words from titles of top Views
- Analysis what kind of titles drive the most traffic
- Analysis the target audience
- Design our titles to make it easy to reach audiences.

- End
4/24/2023