

THE PYTHON PROGRAMMING LANGUAGE

*MSC INFORMATION SYSTEMS AND SERVICES
SPECIALIZATION: BIG DATA AND ANALYTICS*

PROJECT TITLE: DIGITAL CONTENTS DATA ANALYSIS

PANAGIOTAKOPOULOS GEORGIOS(ME2030)

SUPERVISOR

SGOUROS NIKOLAOS

DEADLINE: 24/1/2021

Table of Contents

Introduction	2
1. Description of the problem and available data	3
1.1. Data	3
2. Description of the system.....	6
2.1. Techniques	6
2.2. Application Structure	7
3. Results.....	8
3.1. Evolution charts of views, likes, dislikes per run per video	10
3.2. Index Correlation diagram.....	11
3.3. Tests with weight change in the composite index.....	13
4. Conclusions.....	15
5. Bibliography	16
APPENDIX A	17

Introduction

Today's era is flooded with digital content of any kind. The popularity of social media is growing rapidly with the constant addition of new users. Their ease of use and sharing of images, videos and music has led to an increase in the number of views of these media. There are many online platforms used to view and transmit non-textual content such as videos, images, animations that allow users to rate and comment on these media [1]. The YouTube platform is considered the most popular. More than two billion users connect monthly for entertainment and information, and more than 500 million hours of digital content 'upload' per minute. According to the study by Xu Cheng, Cameron Dale, Jiangchuan Liu [3], the HTTP traffic of the YouTube website is 20% of the total, while of the general Internet traffic, 10% of the total search. The popularity of this website of course also attracts the interest of companies that aim to advertise products and services to users. The question therefore arises as to whether there are correlations regarding digital content that could lead to the optimal costing of business ventures. For example, the analysis of a user's comments [4] or ratings (like / dislike) is considered important, as it is a source through which useful data can be extracted. Through this data, similar content is suggested and an advertising product is promoted that is likely to attract the user. Another evaluation criterion could be the emotional analysis of text, whether it is analysis of user comments or analysis of subtitles. The ultimate goal of emotional analysis is to assess whether the number of views or the number of Likes / dislikes is related to the emotional content of the analyzed text. Digital content is constantly growing and with it the efforts and studies of the data that come from it. Pattern recognition is a key pillar of machine learning and data science.

1. Description of the problem and available data

The purpose of this paper is to study the relationship between the emotional content of YouTube animated videos, and the number of views. The analysis is done from a commercial perspective as the results of this analysis are done with the aim of drawing up a proper digital marketing strategy by a company operating in the field. For example, a digital marketing strategic policy is to search for videos with a high emotional composite index in order to display ads of a specific type. Objective difficulties encountered in the mining and analysis of the above data are:

1. The restriction on the limit of ‘scrapping’ the videos from the specific website.
2. The possible existence of low quality subtitles such as the frequent existence of audio phrases or the limited dictionaries of emotions.
3. Difficulty in choosing the right polarity criteria.

It will be sought to study whether the videos are correlated with a high positive or negative emotion composite index, with the number of their views.

1.1. Data

The data to be analyzed comes from the YouTube platform [2]. In particular, data is retrieved and analyzed from online videos through natural language processing techniques (English subtitles are analyzed) in order to create a classification system based on the emotional analysis of the specific videos. The results type of a YouTube search are two:

1. Individual videos with a specific duration.
2. Playlists, which in turn contain individual videos.

As there is a possibility that there are video collections that have already been found individually, filtering that rejects duplicate recordings is also necessary. These data are the primary information for analysis. For each video the application extracts the information of Table 1.

Table 1. Data stored for each video (database table of videos)

Database table column name	Data description
Id	The unique id of the video
Type	Type of video (whether it is a single video that resulted from the search or a video that resulted from a playlist).
Title	The title of the video
Link	The URL of the page of this video
Duration	Duration of each video in seconds.
Polarity_Pattern	Result of emotional analysis of the PatternAnalyzer of the Textblob library
Polarity_Pattern_Norm	Normalized Emotional Analysis Result of PatternAnalyzer of Textblob Library
Polarity_Naive	Result of NaiveBayesAnalyzer (NLTK) emotional analysis of the Textblob library.
Polarity_Naive_Norm	Normalized result of emotional analysis NaiveBayesAnalyzer (NLTK) of the Textblob library
Polarity_composite	Composite index - ci emotional intensity of the text, which is calculated as the average of the above two indicators.

Essentially from the above information of table 1 what we took as data from the mining, are Id, type, title, link while we calculated during the natural language processing analysis, the other counters. For each playlist having set a maximum of 3 playlists, the application finds the individual videos that make it up. The maximum number of videos used by each playlist is 100. There are also 100 individual videos used in the search results. These videos are stored together with the emotional analysis information in the videos panel. The system also maintains an additional runs, in which the information in Table 2 is recorded for each run.

Table 2. Data kept in the “run” table for each execution of the application

Database table column name	Data description
Run_id	The run number in serial sequence.
Video_id	The unique id of the video as it is posted on YouTube.
View_count	Number of views measured during the specific execution for the respective video.
Likes	Number of likes that were counted during the specific execution for the respective video.
Dislikes	Number of negative likes that were measured in this particular execution.
timestamp	The time (date and time at which this information was recorded).

2. Description of the system

2.1. Techniques

The Python language was used to implement the application, while the Pandas library was extensively used to manage the data generated by YouTube. The following Python libraries have been used specifically to handle YouTube video management features:

Table 3. YouTube video manipulation libraries

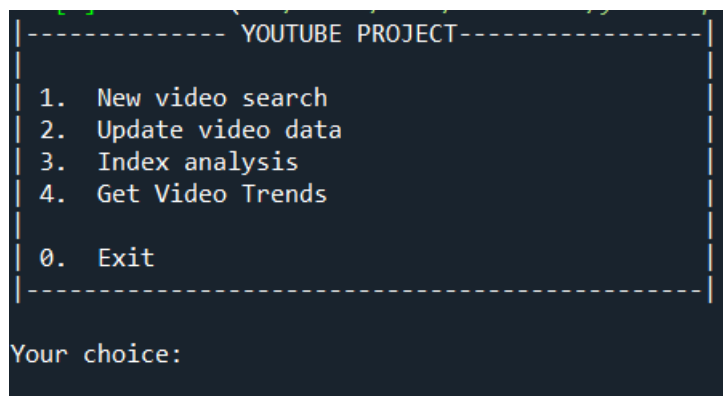
Library	Use
youtubearchpython	<p>Search, ResultMode, PlaylistsSearch methods were used to search YouTube using user search terms, extract results and search videos on playlists.</p> <p>This library was selected because it was not possible to export data from YouTube pages with HTML scrapping method.</p> <p>YouTube returns data not in HTML format but in JSON format, which is formatted by appropriate JavaScript functions. This data is particularly difficult to decode to obtain the desired data for each video.</p>
pyyoutube	<p>This library makes it easy to retrieve information about a video based on its unique YouTube ID.</p>
youtube_transcript_api	<p>Allows you to export the subtitles of a YouTube video in the desired language. The language of choice here is English. Videos that do not have English subtitles are rejected from editing.</p>
textblob	<p>Emotion analysis algorithms focus primarily on defining views, attitudes, and even emoticon faces in a set of texts. The range of established emotions varies considerably from one method to another. While a typical analyst identifies up to three basic polar emotions (positive, negative, neutral), the limit of the more advanced models is wider.</p> <p>The TextBlob approach to emotion analysis differs in that it is rule-based and therefore requires a predefined set of categorized words. These words can, for example, be downloaded from the NLTK database.</p> <p>In addition, emotions are defined based on the semantic relationships and frequency of each word in an introductory sentence that allows a more accurate output to be obtained as a result.</p>

2.2. Application Structure

In essence the application consists of three (3) sections:

1. The video search section (individual and videos from playlists), extracting their key features, extracting subtitles and extracting an emotional grade for each video. Creates the table which includes, among other things, the id of the videos that the application will edit next.
2. The section where we integrate the data into MySQL database.
3. The section of analysis of stored data and their presentation in a diagrammatic way.
4. The section that repeatedly runs section two to complete 48 runs in a 48-hour period (1 run per hour). There was a run every half hour in 24 hours but due to the comparatively negligible change in views, likes, dislikes, the further analysis was chosen to be done with the 48 runs per hour.

For the convenience of the user, an option menu was created so that the user can choose which section the code wishes to run. More detail, in Figure 1:



```
----- YOUTUBE PROJECT-----
1. New video search
2. Update video data
3. Index analysis
4. Get Video Trends
0. Exit
-----
Your choice:
```

Figure 1. Menu of sections

3. Results

The videos that result from the processing of the results of the user query are stored in the videos panel of the system database. This table records the unchanged data for each video and in detail:

Table 4. Columns of the video table

Column	Data
Id	Records the execution number of the module. Every 60 minutes, while the application is executed, run_id increases by 1. For each video, run_id has the same value on each execution.
Type	It gets the price youtube # video. The youtube API also returns the playlist value to indicate if a result is a playlist. The application excludes all other types of results and saves only individual videos.
Title	The title of the video, as it appears on the YouTube search results page.
Link	The URL to the corresponding YouTube video page.
Duration	Video duration, as recorded in the first video processing.
View_count	The number of views, as recorded during the first preprocessing of each video and before the start of the consecutive updates (runs).
Likes	The number of likes of the video, as recorded during the first processing of each video and before the start of recurring updates (runs).
Dislikes	The number of video dislikes, as recorded during the first processing of each video and before the start of the recurring updates (runs).
Polarity_Pattern	The size of the emotional analysis of the video, based on the pattern analyzer of the textblob library
Polarity_Pattern_Norm	The normalized size of the emotional volume of the video, based on the pattern analyzer of the textblob library
Polarity_Naive	The size of the emotional intensity of the video, based on it NaiveBayesAnalyzer of the textblob library
Polarity_Naive_Norm	Normalized video size, based on textblob's NaiveBayesAnalyzer
Polarity_composite	The ci index of the video. Calculated as the average of normalized indicators of pattern analyzer and the size of the

	emotional intensity of the video, based on NaiveBayesAnalyzer.
Video_index_p	The ratio of the number of average values of likes and dislikes for the time of observation.
Video_index_r	The weighted annual number of views to the total number of views at the end of the observation period.
Video_index_lpv	The number of likes in the specific time period weighted in terms of the number of views in the specific time period
Video_index_dpv	The number of dislikes in the specific time period weighted in terms of the number of views in the specific time period
Vide_index_vpd	The number of daily views (for the duration of the observation)

The section 1 as it performs the natural language processing analysis takes a long time with the results shown in Figure 2 below:

```
run_id  run_duration
0      2391.171009
```

Figure 2 : Time of implementation of the first section

The MySQL benchmarks table (Figure 3) shows the exported videos, with the time of their export, the number of views, the Likes and dislikes:

run_id	video_id	view_count	likes	dislikes	timestamp
1	W18nAXue7hM	15977204	97850	20662	2021-01-21 00:04:58
1	KTJQdbvCMag	19912292	81811	23219	2021-01-21 00:04:58
1	83sdwFOL1r8	33825150	497253	26346	2021-01-21 00:04:59
1	AN3Lu8EveHY	27959723	111965	23996	2021-01-21 00:04:59
1	kssoXnSwMSQ	8753010	42993	5704	2021-01-21 00:04:59
1	x6AcuhX4ZUU	6306773	36915	3861	2021-01-21 00:04:59
1	HdJTQqyZsNg	12666164	55391	6816	2021-01-21 00:04:59
1	9G2DyRGJYJk	4949408	24069	2113	2021-01-21 00:04:59
1	mxCqLbWc5ZU	18924730	86921	16259	2021-01-21 00:05:00
1	Lpi4WuSoc94	5639417	26575	1786	2021-01-21 00:05:00
1	HB13XUQwWFW	24769969	91158	19303	2021-01-21 00:05:00
1	7QaHPCwjKk0	14118193	38863	6253	2021-01-21 00:05:00
1	NWtGFCaFuHU	18986156	57223	10958	2021-01-21 00:05:00
1	W7uzZhZ_m68	10328610	34499	4235	2021-01-21 00:05:00
1	yJErPi2Tq0U	16350429	56437	8921	2021-01-21 00:05:01
1	GtDYEkXwzVk	21679508	47895	12096	2021-01-21 00:05:01
1	bGaiE5sbC2k	7415629	35660	2622	2021-01-21 00:05:01
1	sdHNjMc9FQo	57693811	162474	52413	2021-01-21 00:05:01
1	gSL6yoAmG90	17928617	60350	11102	2021-01-21 00:05:01
1	6zG9xYUDT6s	109491382	232796	100499	2021-01-21 00:05:01
1	jMVwRZGH58I	23370589	62516	14049	2021-01-21 00:05:02
1	MTHrcwrq-kM	10577762	38967	6388	2021-01-21 00:05:02
1	G-pqP_9r55E	38969907	104053	20122	2021-01-21 00:05:02
1	PZTeilj-kN4	41094430	189993	28843	2021-01-21 00:05:02
1	1vt2p5ZVZpM	28482100	98853	23514	2021-01-21 00:05:02

Figure 3. Benchmark Table

Due to the filtering performed in section one none was found Playlist with 100 videos (Figure 4):



Video ID	Count
PLC6qlbU1olyXQe1WOKt8UJ4hErX3D7qt8	78
PLtfMA3Fvmjaq_b15mC6IzktHIYtdNXi43	20
PLb9iwmkm5jysRvy8zTnhDB3bFncYFk87	15

Figure 4. Booked videos from Playlist:

3.1. Evolution charts of views, likes, dislikes per run per video

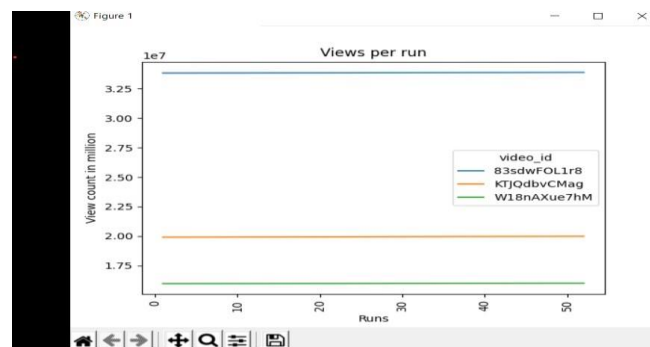


Figure 5. Evolution of the views of the three videos with the best ci (composite index)

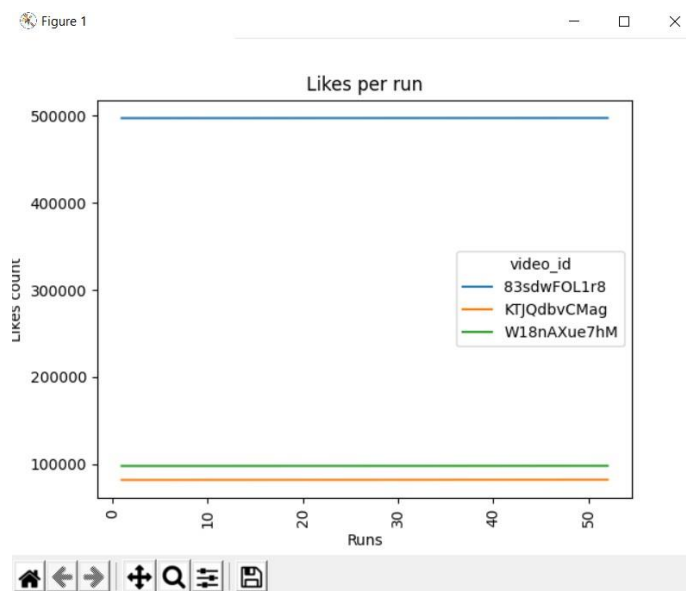


Figure 6. Evolution of the likes of the three videos with the best ci (composite index)

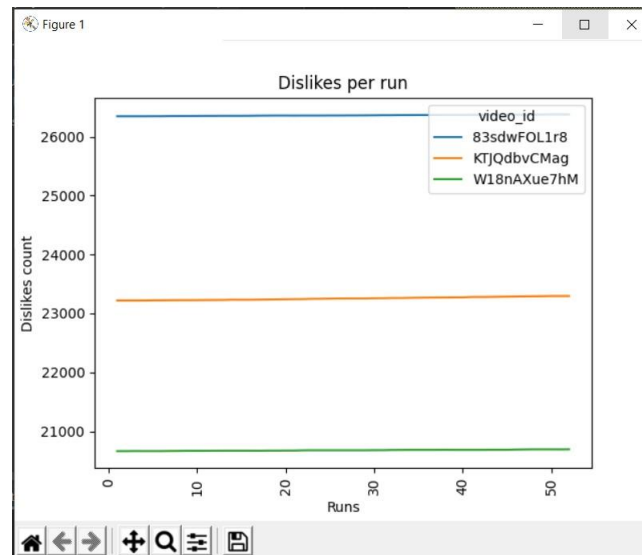


Figure 7. Evolution of the dislikes of the three videos with the best ci (composite index)

Useful observations are that given the scale (small change of likes, dislikes, views in the 48 hours) that we analyze the variability is not obvious. A ratio between likes and dislikes seems to be maintained. That is, the video with the most likes has the most dislikes.

3.2. Index Correlation diagram

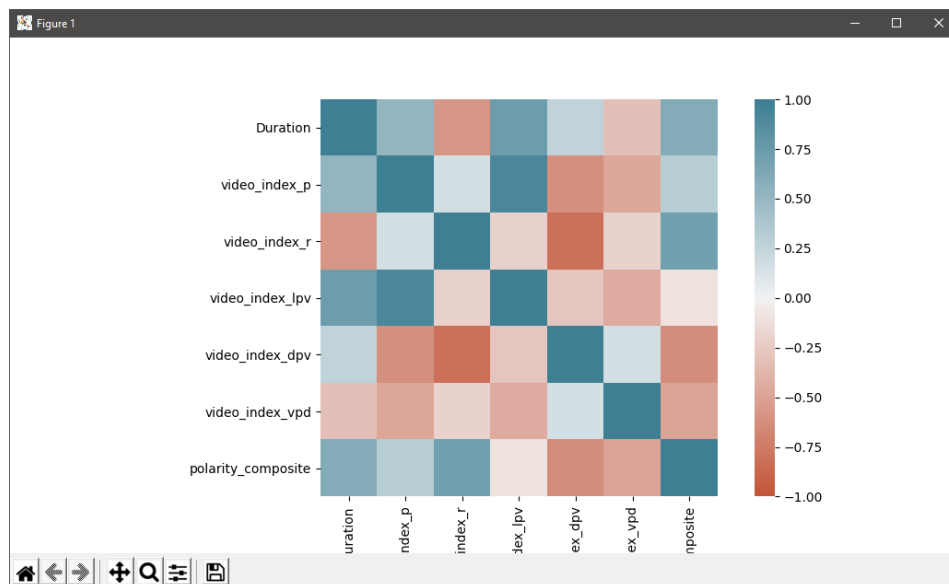


Figure 8. Representation of the correlation of observation sizes in heatmap format

Using the color scale of the heatmap, we can draw conclusions about how positive or negatively correlated the sizes are in pairs. A positive correlation means that when one size increases, the other increases, while a negative correlation means that when one increases, the other decreases.

Observing the heatmap of Figure 1, we come to some conclusions:

Table 5. Indicative analysis of observation size correlations

Duration / index_p	The correlation between the two is positive (blue tint) and on the scale of about 0.50, which means that the correlation is not enough powerful. This means that when the length of the video increases, the video gets relatively more likes than dislikes.
Duration / LPV	The correlation of the two quantities is positive and we observe that belongs to a high area of the scale, so it is quite strong. This means that the longer a video lasts, the more likes it has in relation to its total number of views.
Duration / DPV	The correlation of the two quantities is positive and we observe that it belongs to a low area of the scale, so the correlation is not strong. This means that as the duration of a video increases, may increase the number of dislikes in relation to the total number of views (views).
p / DPV	The correlation of the two quantities is negative and we observe that it belongs to a negative area of the scale (approximately 0.62), so the correlation is negatively strong. This means that as the duration of a video increases, the ratio of dislikes to the total number of views (views) decreases.
Duration / VPD	The correlation of the two quantities is negative and we observe that it belongs to a low negative area of the scale (approximately 0.25), so rather weak. This means that as the duration of a video increases, the ratio of daily views during the viewing period may decrease.

Table 5. Evolution of the number of likes of the three videos with the best ci and the three with the worst ci

The observation shows as expected that the video with the most views, gets more likes, compared to the rest of the videos. Again, the differences can not be plotted on the same chart, due to the difference in the order of magnitude of the likes.

3.3. Tests with weight change in the composite index

The first correlation analysis was performed by dividing the weights of the pattern and the sentiment (polarity patter & polarity naive) in half.

In the second phase, tests were performed with changes in the weights of the 2 factors giving three new ci. Specifically 135 pattern / 65 naive, 150 pattern / 50 naive, 50 pattern / 150 naive. It was observed that there is a negative correlation between likes and composite index influenced more by the pattern.

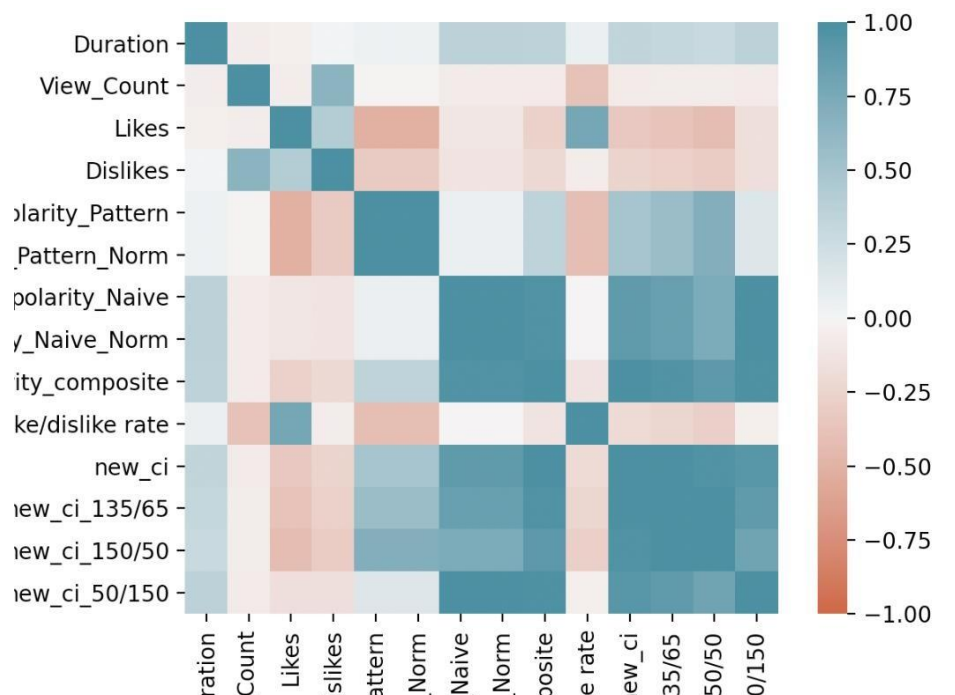


Figure 9: Heatmap with modified CI

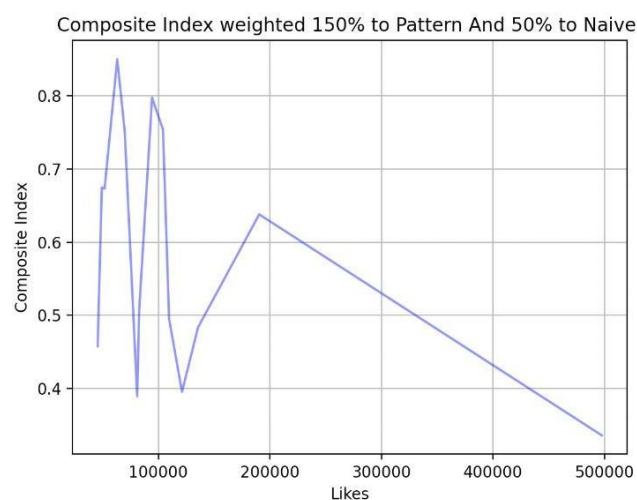


Figure 10. Like and composite index 150/50 correlation

At the same time, it is observed that there is no absolutely linear relationship between like and views.

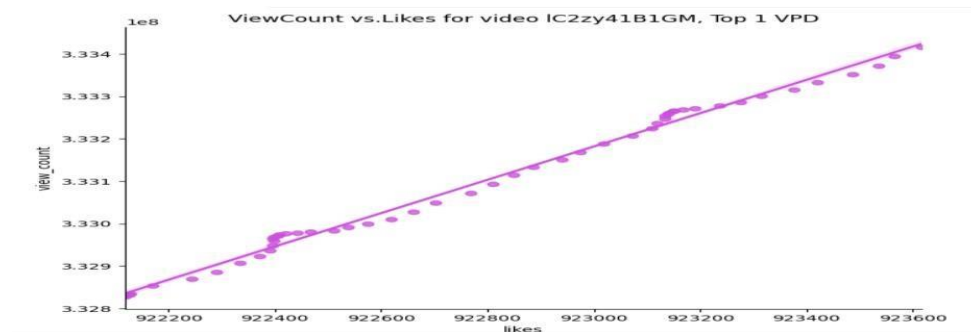


Figure 11. Correlation of Likes with views.

Likewise for dislikes:

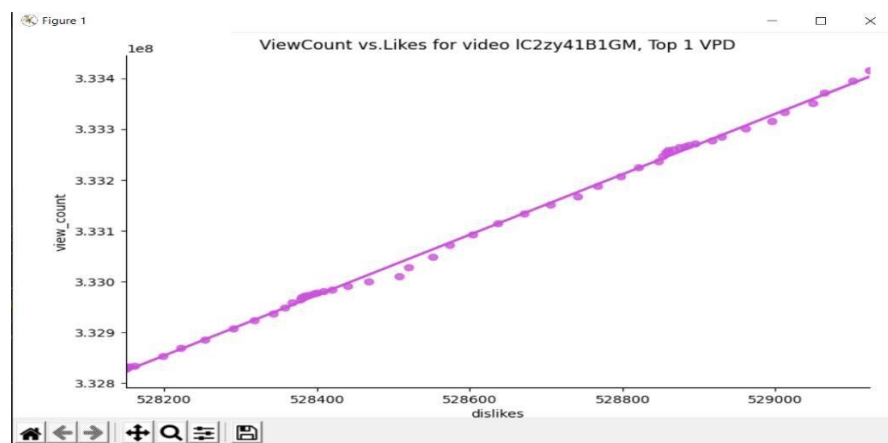


Figure 12. Correlation of dislikes with views.

4. Conclusions

- It was initially found that data retrieval from YouTube videos could not be done using HTML scrapping methods, because YouTube returns search results in a complex JSON format, which is then formatted using JavaScript.
- There is no linear correlation between like / dislike and views.
- For longer videos there is an increasing trend in Likes.
- There is a negative correlation between likes and composite index influenced more by the pattern.

5. Bibliography

- [1] S. Choudhury και J. Breslin, «User sentiment detection: a YouTube use case,» Galway, Ireland, 2010
- [2] Youtube site: <https://blog.youtube/press>
- [3] X. Cheng, C. Dale και J. Liu, «Understanding the characteristics of internet short video sharing: YouTube as a case study,» *arXiv*, 2007
- [4] S. Siersdorfer, S. Chelaru, W. Nejdl και J. San Pedro, «How useful are your comments?: analyzing and predicting youtube comments and comment ratings,» 2010

APPENDIX A

FLOW CHARTS

The workflow of the individual sections of the application is shown in Figures 2, 3 and 4.

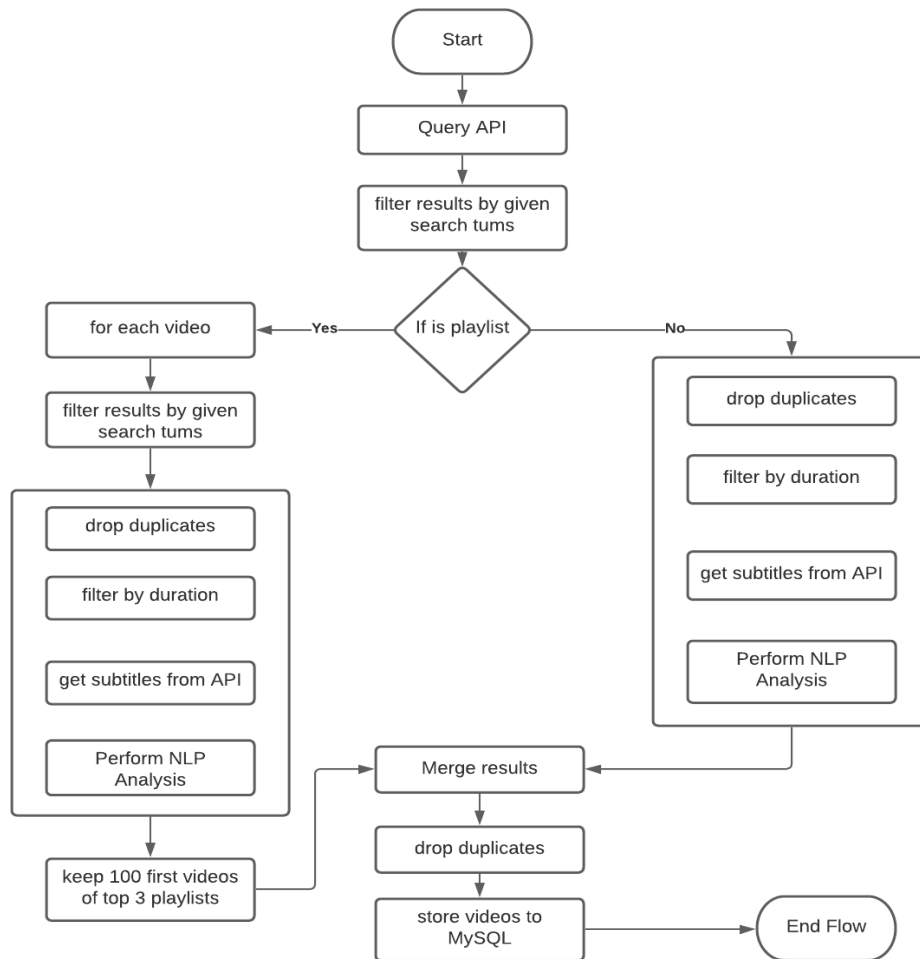


Figure 2. Search module workflow

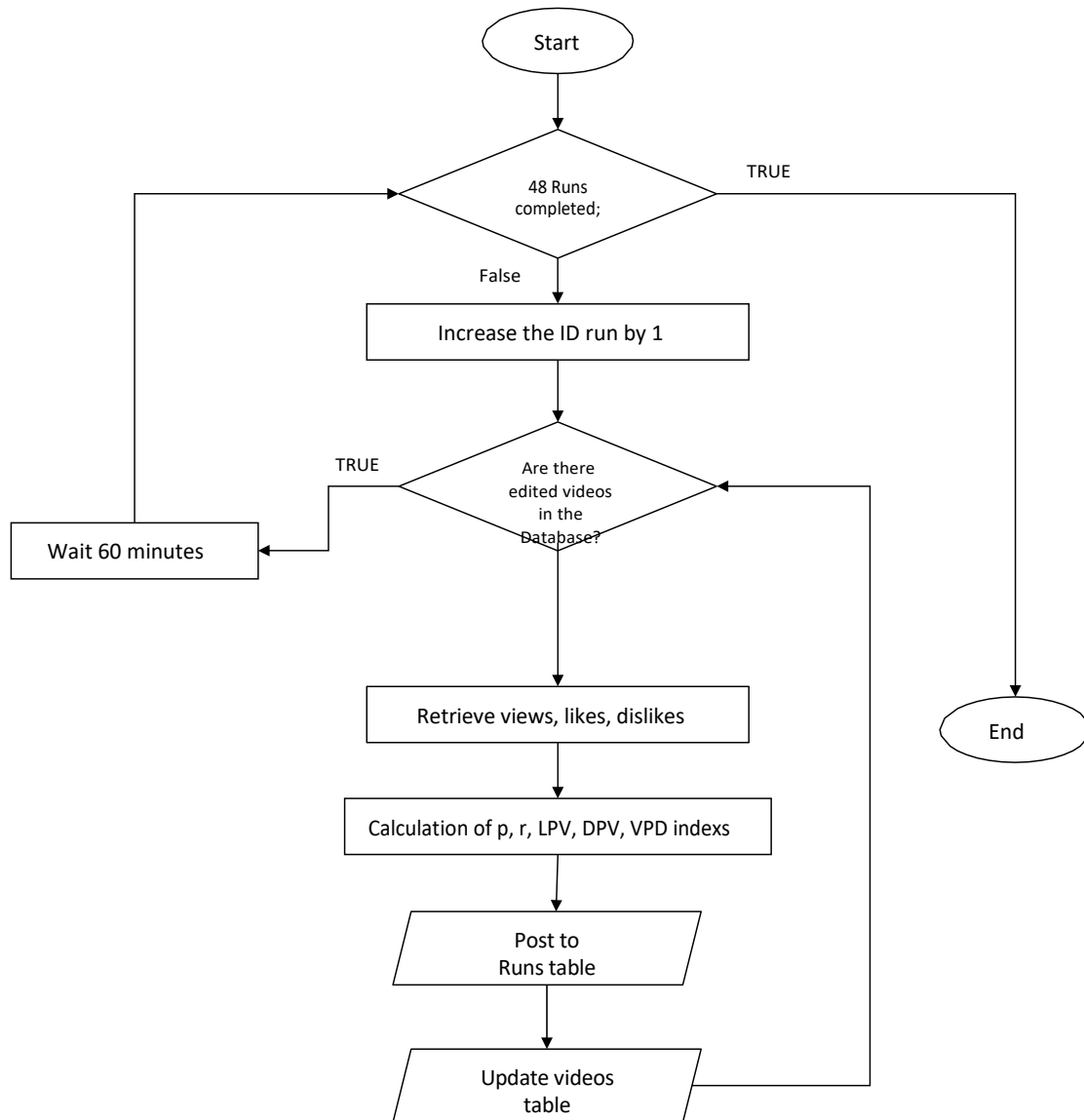


Figure 3. Workflow module for extracting evolutionary data from videos

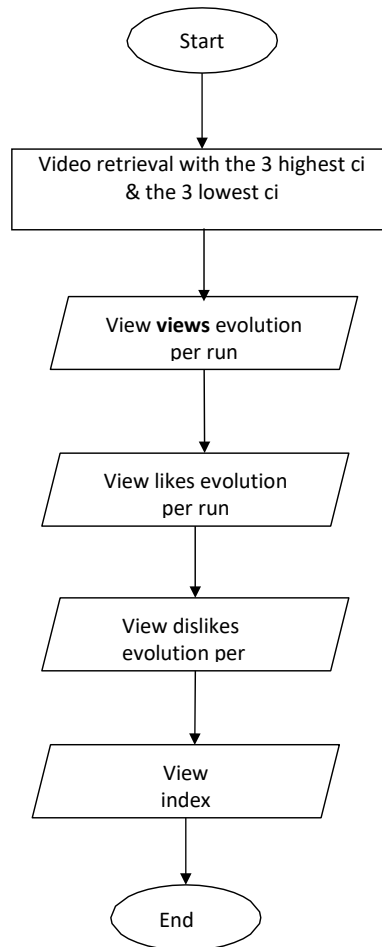


Figure 4. Flowchart of a diagrammatic data presentation module