Analysis of Trending Youtube Videos

Daniel Leung
Department of Computer Science
Princeton University
danielleung@princeton.edu

Abstract

Making Youtube videos may just be a hobby to some, but to others it is their main source of income. Figuring out what titles, descriptions, or thumbnail images give the most views and likes could be a very lucrative business. In this work we apply a number of supervised methods (Linear Regression, AdaBoost, etc) to find the top weighted words for predicting certain metrics. We also apply unsupervised methods (LDA, PCA) to find which words go well with each other in order to have a trending video. Our goal is to be able to increase our chances of having a trending video or predict what videos will become trending.

1 Introduction

YouTube maintains a list of top trending videos on their platform. Top trending videos are not determined solely by its popularity, but are determined by a number of factors including number of views, shares, comments, and likes[1]. We have access to these metrics for a number of trending videos as well as their titles, descriptions, and thumbnail images from the Kaggle Youtube dataset. In this work, we develop and evaluate a number of unsupervised learning methods in order to find latent topics within our text data. We also attempt to use supervised methods to predict views, likes, etc through the text since we know there is a positive correlation between views, likes, etc for trending videos. We also hypothesize and test whether the date a video became trending depended on the text. We focus on the text (description, titles) data because they are some of the only factors that the content creators can manipulate if they desire to have a trending video.

2 Related Work

We wanted to figure out what factors affect a video's click through rate[2]. A video's title would obviously affect its click through rate and would be easy to model; however, we read that a video's click through rate could increase by a good amount depending on its thumbnail so we decided to utilize the thumbnail image data as well.

We also looked at GoogleTrends[3] and TrendoGate[4] to see how trending words change overtime. We then decided to test whether our text data has a correlation to the date it became trending. If we find this to be true, we can see if the video used words that were trending at the same time the video became trending.

We also looked at whether a Youtuber's subscriber count affects whether their videos are more likely to be trending. We found that there is a correlation[5]. Ideally we would want to figure out how to make any video trending despite the number of subscribers we currentlly have so we applied our methods for the dataset filtering out the top subscribed youtubers to see how the results change. We also attempted to predict a video's views and likes factoring in the creator's total subscribers and channel views to see how much past success influences the current success of a video.

2.1 Data set

Kaggle has prepared a dataset of 40950 trending videos from November 13, 2017 to May 17th, 2018. For each video, we are provided with the date it became trending, title and description of the video, channel name, category id, video publish date, tags, number of views/likes/dislikes, and the number of comments.

2.2 Text Processing

For the titles and descriptions data, we used a modified version of the provided preprocessSentences_v3.py in Assignment 1 to preprocess the data[12]. This script uses the Python NLTK library to tokenize, convert to lower case, remove stop words, lemmatize, stem each word using Porter stemming method, and filtering words that occurred fewer than n times in the corpus[11]. We used $n = 20, 30, 40$ for the titles data and $n = 100, 200, 300$ for the descriptions data since there are more text data for descriptions. We will refer to these datasets as titles20, titles 30, etc. The resulting vocabularies contained 2521, 1802, 1343 and 5295, 3032, 2170 words respectively.

2.3 Image Processing

We used a script to download all of the thumbnail image links from the dataset and used the SciPy[6] library to convert all of the 120x90 images into a flattened 10800x3 numpy array where each array of 3 are the RGB values for that corresponding pixel.

2.4 Dates

For each trending date of each video, we converted the date into a numeric value.

$$year * 365 + month * 30 + days$$

We then subtracted the minimum value from each date value so our values start at 0.

2.5 Filtered Dataset

Using the top 5000 subscribed Youtubers dataset from SocialBlade[7], we were able to get the top 100 subscribed youtubers and created a copy of the original dataset except we removed the top 100 youtubers. We also created a dataset which only contained the top 5000 subscribed channels and appended the subscriber count and total channel video view count (normalized to be between 0 and 1) to the bag of words model to test how past success affects the current success of a video.

2.6 Evaluation metrics

For prediction, we used accuracy, recall, precision, and f1 for the classifcation metrics. For the continuous metrics, we used median absolute error.
To evaluate the LDA method, we checked the log-likelihood and tested LDA against other unsupervised learning methods such as PCA. We tested the prediction accurracies when training classifiers on data fitted by LDA and PCA to predict the views, likes, and categories of the videos.
For the category predictions, we checked the highest weighted words for each category to ensure that they make sense.
To evaluate feature selection via threshold tuning when constructing our bag of words model, we created a score for each metric. For classification metrics, we just used the accuracy. For regression metrics where we calculate median absolute error, we needed to assign higher scores for lower MAE. We divided each MAE by the max MAE of each dataset and then subtracted this value from 1 so that a lower MAE gives us a higher score. This formula is only supposed to give us an idea of how the datasets perform to each other relatively. We explain this more in the feature selection section.

3 Methods

3.1 Latent Variable Modeling

We used three different types of unsupervised learning methods from the SciKitLearn Python libraries[6]. We used LDA so we can see the latent topics, and PCA to compare with LDA. TSNE was mainly used to visualize and evaluate the performance of LDA. All parameters were default unless specified. Each model was tested with a varied number of components.

1. Latent Dirichlet Allocation (LDA): with 5 components and 10 components
2. Principal Component Analysis (PCA): with 5 components and 10 components
3. t-distributed Stochastic Neighbor Embedding (TSNE): with 2 components, 1 verbosity, 0.99 angle, PCA initialization

3.2 Predictive Modeling Methods

For each prediction task, we used three different methods from the SciKitLearn Python libraries[10]. We used a linear model (LR) and an ensemble model(ADA) to see if our data is linear. We used Ridge Regression to compare with LR. All parameterizations are default unless specified otherwise. I used the regressor class from the library when dealing with continuous variables and the classification class when dealing with binary variables.

1. Ridge Regression / Ridge Classification (Ridge): with the default values
2. Linear Regression / Logistic Regression (both referred to as LR): with L_2 penalty
3. Ada Boost (ADA): using 50 estimators

3.3 Generative Methods

We used generative adversarial (GAN) networks on the thumbnail data in order to generate a trending thumbnail. We used the corresponding hyperparameters: 100 noise size, 0.00004 learning rate for the discriminator, 0.0004 learning rate for the generator, and 64 batch size.

4 Results

4.1 Prediction

	Titles				Description			
	Date	Views	Likes	Category	Date	Views	Likes	Category
LR	**0.16**	2,928,258	0.070	**0.63**	0.13	12,262,134	0.38	0.16
ADA	0.050	**1,377,108**	0.19	0.35	0.077	1,853,795	0.13	0.26
Ridge	0.160	2,087,068	**0.052**	0.60	0.13	3,394,866	0.091	0.16

Table 1: Predicting accuracy of date and cateogory classification and median absolute errors of views and likes (likes/dislikes ratio) regression for titles20 data and descriptions100 data.

Because words become trending over a period of time rather than a single day, we used classification for predicting the dates rather than regression. We grouped every 90 days as a class and predicted these classes using the text data. We see that the accuracy was poor across titles and descriptions for all models which indicates that there is little correlation between a video's text and the date it becomes trending. In other words, using a word that is trending in April may not increase the chances of that video to become trending in April.
We also see that we get the best performance for each metric with the titles data. This means that titles have more influence on metrics that determine whether a video becomes trending. This makes sense since a lot of views for a video is generally due to "clickbait" titles.
We also see that our linear methods worked better in predicting the dates, likes, and categories while the ensemble method (ADA) worked better in predicting views.

	Titles			Descriptions		
	Precision	Recall	F1	Precision	Recall	F1
Music	0.84	0.79	0.81	0.20	0.19	0.19
Sports	0.79	0.84	0.81	0.043	0.024	0.031
Entertainment	0.60	0.64	0.62	0.26	0.37	0.31
Science & Tech	0.60	0.63	0.61	0.045	0.020	0.028

Table 2: Recall, precision, and f1 scores for select categories for category prediction for titles20 data and descriptions100 data for the LR model

Our predictive models for the categories of each video consisted of 16 classes (categories). In table 2, we show the recall, precision, and f1 scores of selected categories. Again we see that the titles data gives us much better results.

	titles20				
Music	mv	billboard	bjork	sampl	chainsmok
Sports	espn	gopro	wwe	nba	candid
Entertainment	wwhl	ellen	choreographi	versu	bachelor
Science & Tech	mission	numberphil	tech	smartphon	smarter
	titles20 (filtered)				
Music	mv	billboard	bjork	chainsmok	audio
Sports	espn	gopro	wwe	candid	nba
Entertainment	wwhl	babish	choreographi	versu	snl
Science & Tech	mission	numberphil	tech	smartphon	smarter
	descriptions300				
Music	coconut	festiv	station	hulkbust	luci
Sports	keep	asmr	health	easi	foot
Entertainment	half	kimmel	blind	fluffi	jona
Science & Tech	roy	celeb	jenner	comput	hope

Table 3: Top 5 predictive words for each selected category for titles20, titles20 with top 100 youtubers filtered out, and descriptions300 datasets

In table 3, we can see the top 5 predictive words for predicting each category. We see that these words make sense for each corresponding category: ”espn” and ”nba” are highly weighted for predicting whether a video is sports related. This means that a lot of trending sports videos use the words ”espn” and a lot of trending music videos use ”billboard”. So if we are looking to create trending videos in these categories, we could use these words to increase our chances of becoming trending.

We also tried filtering out videos created by top 100 subscribed Youtubers, titles20 (filtered), to see what words are used in each category by less known Youtubers. We see that the words are very similar but slightly different. we could filter out more top Youtubers to see how the words gradually change.

Views	rewind	offici	infin	lovato	maluma	delic	shape	trap	la	twice
Likes	sidemen	closer	app	glynn	hustl	speechless	keynot	stirl	span	minnesota
Dislikes	domest	utah	c	nra	bbq	zombi	fergi	access	net	jess

Table 4: Top 10 predictive words for predicting views, likes, and dislikes for the titles20 dataset.

Some examples of the best predicted videos for views (lowest MAE) were: ”TRYING ON CHEAP PROM DRESSES FROM EBAY/AMAZON”, ”Made Defiant: The Mixtape ft. Neymar Jr., Kane, zil and Mendy — Beats by Dre”, and ”GIANT Bowl of Lucky Charms CHALLENGE (5,000+ Calories)”. Similarly for likes: ”BLIND GIRL DOES MY MAKEUP ft. MOLLY BURKE”, ”Which is Worse For You: Sugar or Fat?”, and ”Catching a SHARK by HAND!”

The median of the number of views of the test set is 1,509,880, so our median absolute error for the best model does a decent job. We also get good results when predicting the likes to dislikes ratio. We use the ratio because a video with a lot of dislikes may not necessarily be a disliked video if there

are many more likes. The more popular a video is, the more dislikes it will inevitably get. In table 4, we can see the top predictive words for predicting the likes, dislikes, and views. We can see that the predictive words for views makes sense: ”rewind” could refer to Youtube’s yearly ”rewind” videos which are one of the most popular videos on Youtbe. The word ”official”, indicated by ”offici” is also often used in popular music videos or trailers for movies

4.2 Feature Selection

We performed feature selection by building a bag-of-words representation for the titles and descriptions datasets using different word count thresholds. In Figure 1, we show the difference in performance. Because our variables are measured differently (views are measured with MAE and lower is better while categories is between 0 and 1 and higher is better), we normalized all of our values to be between 0 and 1. For views and likes, we subtracted that value from 1 so that a lower MAE would give a higher score. This chart is to show the relative differences between each dataset, the meaning of the score is different for each variable.

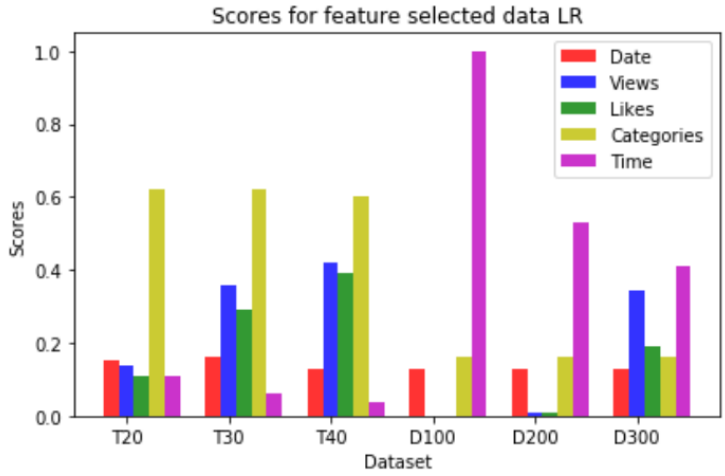

Figure 1: Performance scores for each dataset, T20 (titles 20), etc for 5 variables using Linear Regression. Corresponds to table 5.

	Date	Views	Likes	Categories
T20	0.16	2928258	0.070	0.62
T30	0.15	2168803	0.056	0.62
T40	0.16	1979328	0.050	0.60
D100	0.13	12262134	0.38	0.16
D200	0.13	3390527	0.078	0.16
D300	0.13	2220688	0.063	0.16

Table 5: Predictions for each feature selected dataset using Linear Regression. Corresponds to figure 1

We can see that we get lower training times when increasing our thresholds and the descriptions datasets take a lot longer to train since there is more data. We also see a large improvement in the views and likes prediction when using higher thresholds for titles and descriptions while the categories performance increases slightly at T30, but then decreases at T40 for titles. This means the optimal threshold for categories prediction is somewhere between, while for views and likes, we may be able to increase our performance if we used an even higher threshold.

4.3 Unsupervised Methods / Latent Structure

We applied LDA with 5 components on the titles20 data and the descriptions300 data to find latent topics in trending videos.

4.3.1 Findings

Latent Topic 1	live	tri	cake	one	’	make	”	v	”	ft
Latent Topic 2	offici	video	trailer	hd	audio	music	lyric	ft	2	first
Latent Topic 3	2018	makeup	super	’	bowl	v	life	full	challeng	commerci
Latent Topic 4	2017	2018	new	star	last	’	best	award	show	war
Latent Topic 5	make	day	$	test	food	new	5	break	time	2018

Table 6: Latent topics for titles20 with 5 topics

Latent Topic 1	fifti	jam	basic	hiddleston	pleas	killer	keyboard	stop	harri	racist
Latent Topic 2	jam	stop	cancel	stori	store	fifti	korean	bloodpop	koshi	starter
Latent Topic 3	fifti	cancel	jam	starter	steve	korean	self	former	hiddleston	deliv
Latent Topic 4	jam	fifti	cancel	hiddleston	basic	starter	sheeran	korean	stir	alesso
Latent Topic 5	fifti	jam	cancel	korean	starter	enriqu	elect	prep	shut	hard

Table 7: Latent topics for descriptions300 with 5 topics

Looking at table 6, we can make sense of latent topic 2. We see words such as ”video”, ”trailer”, ”music”, and ”hd”, which indicate that videos which contain a combination of these words are common in trending videos. This makes sense since a lot of movie trailers or music videos are usually very popular on Youtube. From above, we see that our predictive models do not perform very well on the descriptions data. This could explain why our latent topics for the descriptions data does not make much sense.

4.3.2 Evaluation

The log likelihood for our LDA model applied with 5 and 10 components is -1103911 and -1082943 respectively. We see that we get a better score with more components. The LDA model is also evaluated through comparison between other methods.

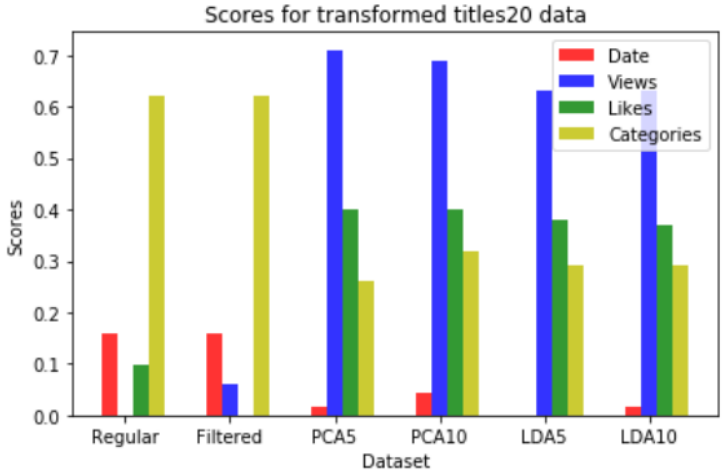

Figure 2: Data corresponding to table 7, metrics predicted for titles20 dataset with PCA, LDA, and top 100 youtuber filtering applied

	Date	Categories	Views	Likes
Regular	0.16	0.62	2928258	0.070
Filtered	0.16	0.62	2750379	0.077
PCA (5)	0.016	0.26	857229	0.046
PCA (10)	0.042	0.32	919008	0.046
LDA (5)	0	0.29	1093666	0.048
LDA (10)	0.017	0.29	1085068	0.049

Table 8: Data corresponding to figure 2, metrics predicted for titles20 dataset with PCA, LDA, and top 100 youtuber filtering applied

For figure 2, we used the same scoring idea as figure 1. We see that applying PCA and LDA improves our prediction for views and likes by a a significant amount compared to the regular titles20 dataset. We do get worse performance for categories and date prediction however. We see that PCA provides better results for similar component sizes compared to LDA, and more components provide very similar results for PCA and LDA. Because PCA improves our results this could mean that the data is highly linear.

Next we compare the results of LDA with t-distributed stochastic neighbor embedding (t-SNE). We transformed the data with t-SNE to 2 components so they can be visualized in a plot. For each video datapoint, we defined a color for each of the 5 latent topics from LDA and colored each datapoint with the corresponding color for the latent topic they had the highest proportion for. From figure 3, we can see that each color is clustered which means LDA did a decent job. However, we see from the picture that there are lot more smaller clusters rather than a few large distinct clusters, which could mean that increasing the number of topics could improve results.

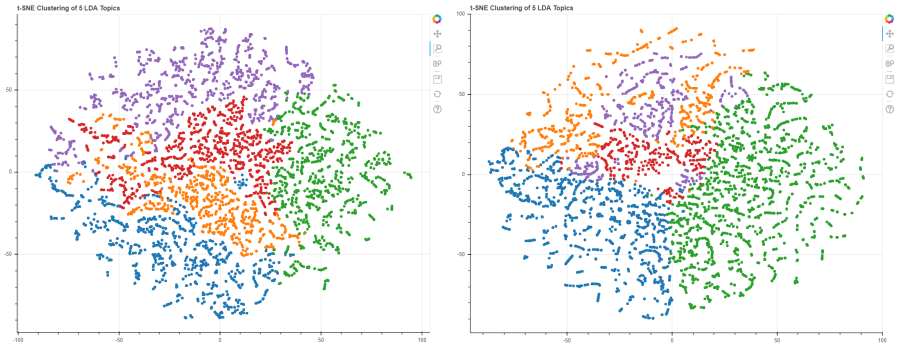

Figure 3: Data transformed to 2 components via t-SNE. Left is titles20 data and right is descriptions300 data. Color corresponds to the latent topic each video identifies most with[8]

4.4 Thumbnails

We attempted to use a generative adversarial network (GAN) to learn on the thumbnail images of the videos[9]. We created the images dataset out of videos with the "entertainment" category to try to get a more uniform set of pictures. From figure 4, we see that the discriminator loss is very low so that means the discriminator can easily determine whether an image was generated or not, and the generator loss is high so our generator was not able to generate images that resemble the original input too well.

```
D Loss: 0.03131
G Loss: 12.17627
```

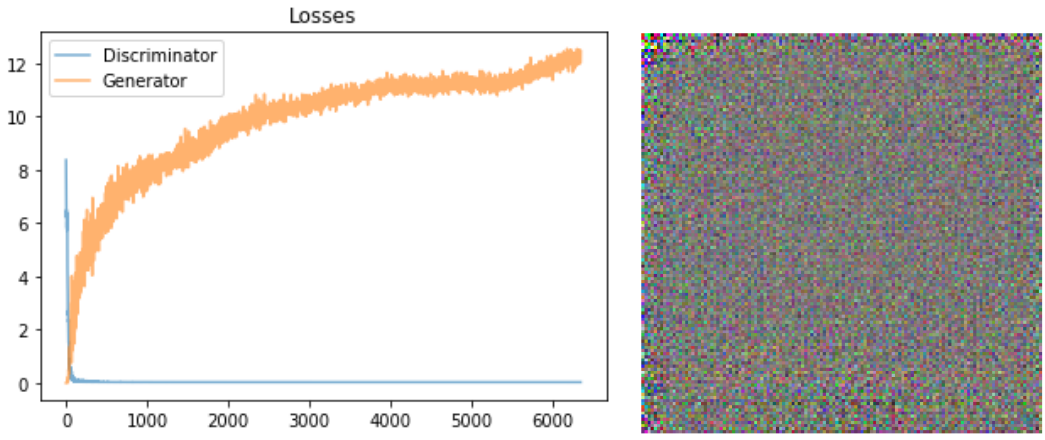

Figure 4: Discriminator loss and Generator loss. Example of a generated output.

We also see that the generated output does not look too promising. This means that the dataset of images is still much too diverse for the model to learn to generate an image that resembles thumbnails of "trending videos". If we could pick out only thumbnails that looked similar to each other, perhaps thumbnails where most of the image is taken up by a face, we could get better results.
We also tried predicting the number of views for a video based on its thumbnail using Linear Regression, however we did not get good results either. This again, indicates that thumbnail images are much too diverse such that a single type of thumbnail image cannot guarantee an increase in views.

4.5 Previous Channel Success

Using the top 5000 subscribers dataset, we are able to get the subscriber count of a channel and the total video views count of a channel. We added 2 new features to the bag of words model of each video: the channel's subscriber and view count. We want to see if we can improve the views and likes prediction of a video if we factor in a channel's previous success.

	Views	Likes
LR	4023199	0.062
ADA	18826038	0.19
Ridge	2390284	0.039

Table 9: Median absolute error for views and likes prediction on titles40 data with only top 5000 youtubers

Comparing table 9 and table 5, we can see that we do slightly worse with the median absolute error for views for our best classifier. This may be because we needed to drop datapoints where the channel was not among the top 5000 subscribed Youtube channels since we did not have subscriber and view counts for that channel. This reduced our dataset by almost a two thirds so we had less data to train on. Also, the median view count of the dataset increased from 1,509,880 to 1,901,655 after filtering to only consider top 5000 Youtubers so overall, we are predicting higher numbers which leads to higher errors. Looking at the likes ratio, we see we get a good MAE of 0.039. Although table 5 shows the results for LR, we tested and found that applying Ridge on the regular titles40 dataset gives us 0.044 so adding the subscriber and view count does increase our accuracy for likes prediction on our best model.

5 Discussion and Conclusion

We trained a number of prediction methods on video titles and descriptions in order to predict the date, views, likes, and categories of Youtube videos. We found that there was little correlation between the text data and the date the video became trending. We also found that we had much better results training on the title data. We also applied feature selection on the datasets by increasing our word count thresholds when creating the bag-of-words model. We saw a lot of improvement in prediction for views and likes when increasing the thresholds while categories and dates had similar results. We also applied LDA to the dataset and found latent topics within our data. To test the findings, we compared the performance of LDA between similar unsupervised methods such as PCA and t-SNE. We found that predicting views and likes on the feature selected data had much better performance comparing to the regular data, but the prediction for categories and dates dropped. We attempted to apply a generative adversarial network on the thumbnail data but we could not get any interesting results because the dataset of images were too diverse. Using the images to predict views also had poor results. We also tried factoring in the previous success of the channel when predicting a video and found that this improved our prediction for likes.

There are some possible extensions to improve our current results. We could increase our thresholds for our bag of words models even more since we saw increasing performance as we increased the thresholds. We could also filter out more than just the top 100 Youtubers since our data did not change too much after filtering out the top 100. For the thumbnails data, we could filter them such that the dataset is less diverse (for example, only using thumbnails with faces) either by hand or by using existing models. Also Instead of applying the GAN on the images directly, we could build on existing image classifiers and models. For example, there is an existing NSFW classifier[13] for images built by Yahoo so we could check if there is a correlation between NSFW thumbnails and view count. We could also use exisiting "clickbait" classifiers[14] on the titles to see if there is a correlation between "clickbait" titles and views.

References

[1] J, Mitchell. Trending YouTube Video Statistics. Kaggle, 20 Nov. 2018, www.kaggle.com/datasnaek/youtube-new.

[2] How Vidyard Increased the CTR on Its Homepage Video by 15

[3] Google Trends, Google, trends.google.com/trends/?geo=US.

[4] TrendoGate.com — Twitter Trends Archive Trends Everywhere Anytime. TrendoGate.com — Twitter Trends Archive Trends Everywhere Anytime, trendogate.com/.

[5] Dean, Brian. We Analyzed 1.3 Million YouTube Videos. Here's What We Learned About YouTube SEO. Backlinko, 28 Feb. 2017, backlinko.com/youtube-ranking-factors.

[6] SciPy: Python-based ecosystem of open-source software for mathematics, science, and engineering. Retrieved from https://www.scipy.org/

[7] Urgo. Top 5000 YouTubers Sorted by SB Score - Socialblade YouTube Stats — YouTube Statistics. Top 5000 YouTubers Sorted by SB Score - Socialblade YouTube Stats — YouTube Statistics, socialblade.com/youtube/top/5000.

[8] Topic Modeling Visualization - How to Present Results of LDA Model? — ML . Machine Learning Plus, 4 Dec. 2018, www.machinelearningplus.com/nlp/topic-modeling-visualization-how-to-present-results-lda-models/.

[9] Surma, Greg, and Greg Surma. Image Generator - Drawing Cartoons with Generative Adversarial Networks. Towards Data Science, Towards Data Science, 11 Feb. 2019, towardsdatascience.com/image-generator-drawing-cartoons-with-generative-adversarial-networks-45e814ca9b6b.

[10] scikit-learn: machine learning in Python. (2019). Retrieved from https://scikit-learn.org/stable/

[11] Steven Bird, Ewan Klein, and Edward Loper.Natural Language Processing with Python.OReilly Media, Inc., 1st edition, 2009.

[12] B. Engelhardt. Fast classification of newsgroup posts. Example homework provided on Piazza.

[13] Yahoo. Yahoo/open_nsfw. GitHub, 6 Nov. 2018, github.com/yahoo/open_nsfw.

[14] Peterldowns. Peterldowns/Clickbait-Classifier. GitHub, 16 June 2016, github.com/peterldowns/clickbait-classifier.