Vallabhya: A Quantitative approach for prediction of YouTube Video's Polpularity

Video Content Analysis [Research project]

Devang Uniyal[1], Jitesh Hemji[2] y Veer Bhadra Yadav[3]

[1] *Indian Institute of Technology, Kanpur, India, devangu@iitk.ac.in*
[2] *Indian Institute of Technology, Kanpur, India, jithemji@iitk.ac.in*
[3] *Indian Institute of Technology, Kanpur, India, veerby@iitk.ac.in*

Abstract—This study of both qualitative and quantitative aspect of video provides a starting point to seek information about the video's universality. The primary goal of the proposed research project is to analyse and evaluate the content of the visuals. We know that video is comprised of two parts - image and audio. To initiate, we divided the problem into two major components - image analysis and audio analysis. We explored variety of features in both of these and extracted those features from each of these parts separately using different libraries like Librosa (for audio analysis) and for the video part we used image processing taking out the frames from the videos and process it using various libraries like OpenCV, beautifulsoup, youtubedl module etc. After preprocessing the dataset to get the prediction of like or view counts applies various classification models like random forest classifier, XGBoost based on certain X parameters like for the video part we have keypoints values, split in a video, blurness, brightness and similarly for the audio part we have - chroma vector, spectral features, mel frequency cepstral coefficient. Finally implemented SHAP plot to get feature importance. Since the videos are from different genres may have a variety of different sets attributes (that are mutually disjoints), deciding their popularity. To get a controllable domain, we are restricting our model analysis to the Data Analytics videos available on YouTube.

Keywords—Librosa, Recursive Feature Elimination(RFE), Principal Component Analysis(PCA), XGBoost, SHAP plot

INTRODUCTION

Social media platforms have become the pivot of almost every activity that involves mass sharing of information in any form. This is because of the very fact that there is no other means which is as economical and fast as social media. Despite being almost free for the users, the companies running these platforms earns huge profits via different channels, enough to gratify its content creators. But they expect more viral content from their creators as this is one of their main selling points. So, there is no denying of importance of popularity for a content creator. Also, one of the largest online public platforms that broadcast videos of content creators in exchange for the immense user engagement on their interface is YouTube. So, for research purposes, YouTube video collection can be chosen as the perfect sample space for predicting the success of any new, to-be-uploaded video.

The inspiration behind this study was a simple question that how a new content creator can start making good video content for YouTube. We all get recommendations from our friends, relatives, parents, etc. to watch different types of 'good' videos (as what they call it) on YouTube, Prime, etc. But what is this 'good' referring to? Is it content? Is it the video quality? Is it the background music? Or maybe it is the combination of various factors depending upon the demography of where the video is released?

As a creative and curious group of people, we want to explore the different dimensions of what makes a visual (in our case: a video) 'good' and try to predict whether any given video is 'good' or not. We will try to quantify our whole analysis as 'Good' is a relative and rather a quite subjective term. We have defined it in a more concrete way: 'GOOD' means more likes to views ratio [1,2]. Since the videos of different genres may have a variety of different sets of deciding attributes, that are mutually disjoints, deciding their popularity/reach, to get a controllable domain, we are restricting our model analysis to the Data Analytics Videos available on YouTube. Later, we can extend our study to various other genres (e.g., Entertainment videos, News videos, etc.). The main models used for prediction are the Random Forest Classifier model and XGBoost model, which includes accuracy parameters, features selection, etc. Both image and audio parts are analysed for features selection separately and integrated in the model used. Recursive Feature Elimination technique is used for feature selection. Different metric scores like Confusion matrix, Grid search with cross validation score, accuracy score etc., are used to quantify the performance of the models used.

The remainder of the paper is in the following format. Section 2 provides a literature review along with a comprehensive analysis of existing state-of-the-art work in the domain of this paper. Section 3 and 4 describes the research method following the dataset. Section 5 and 6 presents the experimental setup, ML algorithms used and results. Finally, the study is concluded in Section 7 following with the link to gitrepo.

LITERATURE REVIEW

Cha et al. [1], in his research, found the existence of a correlation between the number of views on early days and later days for a video on Youtube. His results showed that correlation coefficient between the second and ninetieth day is 0.84 and was highly linear. Lerman and Hogg [2] made predictions for Digg* stories using the model that assumes user behavior as stochastic process. Their main points of considerations were user interface of the system and social dynamics arising from friends feature. They found that major roles were played by the factors: the inherent quality and social influence, in the any story's popularity. But their model is not suitable for a platform like YouTube where the governing features of any video's popularity are much more complex qualitatively and quantitatively.

Haitao Li and team [3] built a novel propagation-based prediction tool, called Social network assisted Video Prediction (SoVP). This tool considers both the influence from the underlying propagation structure and the intrinsic attractiveness of a video.

Pinto, Jussara and Marcos [4] studied the early view patterns of YouTube videos to predict their popularity. For the first time, to predict the popularity evolution pattern of the video at prediction time, these researchers developed MRBF model. They also explored an alternative approach that based the popularity of an online video on the collected view trends upto a certain reference date.

Giulia, Marco and Alberto Del Bimbo [5] proposed an approach that utilised visual sentiment and content feature of web videos to predict their popularity. In their proposed method, specific prediction models for different popularity trend models are learned and visual content in the popularity prediction model is included for the first time. Their proposed method outperformed the other state-of-the-art ML and MRBF methods at that time.

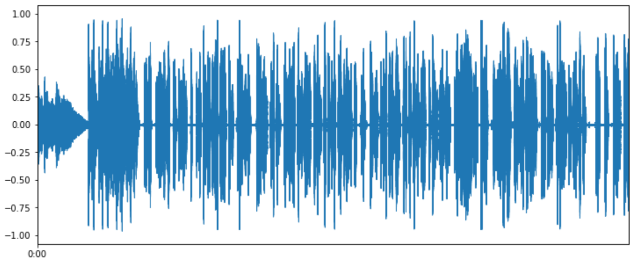
Meher UN Nisa and team [6] studied on 'Optimization of YouTube Video Popularity Using XGBoost' in which they analysed the importance of variables like Video quality, duration, etc. for the popularity (here, number of views) of a video on YouTube. The workflow that was used had Dataset being pre-processed for feature selection algorithm which then is trained on 3 different models (namely Decision tree, XGBoost and tuned XGBoost) for comparing accuracy, precision, F1 score and recall. Tuned XGBoost gave the maximum accuracy and precision of 88 % and 86 % respectively.

RESEARCH METHOD

A large set of images and an audio together makes a video. Analysing a video as a whole is impossible rather than breaking it into its component i.e., audio and image and analysing them separately.

Audio Analysis

An audio signal is expressed as a function of amplitude and time. We can not use audio directly for our analysis. We need to extract some relevant acoustic features to implement ML algorithms on it. Python has a very useful library named **Librosa** for feature extraction. Librosa uses audio files in '.wav' format. For audio file extraction from a given video URL, a library named **youtube_dl** is being used. Using this library we can also get some general features directly from URLs which are not related to content e.g., upload date, view count, like count, and many more other features which might not be of our use.



waveplot for an audio signal plotted using *Librosa*

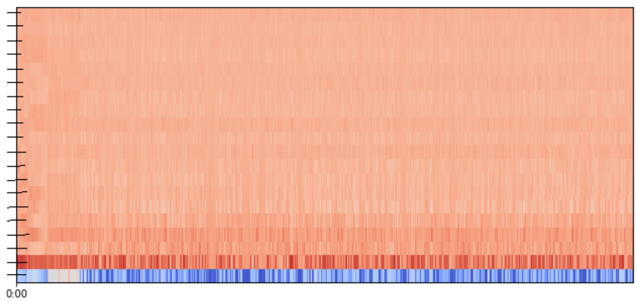
While watching a video, the two basic audio features which we can think of are the quality of voice(whether male or female, young or aged, language used) and background music. To account the first feature i.e., the quality of voice, we define Mel-Frequency Cepstral Coefficients (MFCCs). Mel-Frequency cepstrum(MFC) is collection of 10-20 of these coefficients which mathematically given by linear cosine transformation of log power spectrum on a nonlinear Mel scale of frequency.[7]**Mel scale**(Mel comes from melody) is an intuitive scale of frequencies judged by listeners placed at equal distance from one another. There have been many formulas proposed for Mel scale. One such formula from O'Shaughnessy's book

$$m = 2595 \log_{10}\left(1 + \frac{f}{700}\right)$$

It's inverse is given by

$$f = 700\left(10^{\frac{m}{2595}} - 1\right)$$

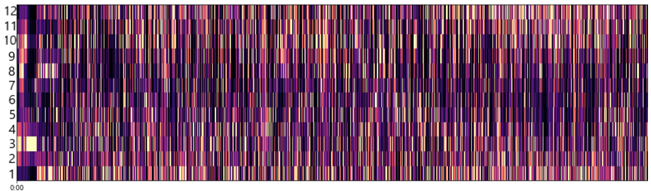
In the above formula, by taking equally spaced values of m_i, we get values of f_i. If we take values only at $f_i's$ then we might loose information at other frequencies. To prevent this we take weighted sum of frequencies near $f_i's$.



Colorbar for MFCCs for each sample

[8]In western music theory, there are 12 pitch classes. Chroma feature is a feature vector containing 12 values representing energy of each pitch class present in a particular audio signal. Chroma feature account for harmonic and melodic characteristics of music. Implementation of these features in python is discussed later.

Apart from these two basic audio features, there are some other ways to characterize an audio signal e.g., all the



Colorbar for Chroma Feature Vector for each sample

spectrum related parameters(frequency based features) obtained by converting the audio signal from time domain to frequency domain using Fourier transform. Although there are many but enlisting few of them like spectral bandwidth, spectral roll-off, spectral centroid, zero crossing rate etc.

Spectral bandwidth is the width of band at half the height of peak maximum. The frequency below which a certain percentage (generally 80-90 %) of the magnitude distribution of spectrum is concentrated is called Spectral roll-off. Spectral centroid is that frequency of spectrum at which its energy is centred. That's why spectral centroid is also known as centre of gravity of spectrum. It is similar to weighted mean represented as:

$$f_c = \frac{\Sigma_i S(i) F(i)}{\Sigma_i S(i)}$$

Here S(i) and F(i) represents the spectrum magnitude and frequency of i^{th} bin.

[9]The zero crossing rate is the measure of the smoothness of the signal. To calculate the rate we check the no. of time it crosses zero for each segment. Before calculating the value of zcr the following things should be kept in mind:

- Each segment of signal must have zero mean. Therefore, mean should be subtracted from each segment before proceeding further.
- Segment should be long enough that signal crosses zero line at least few times

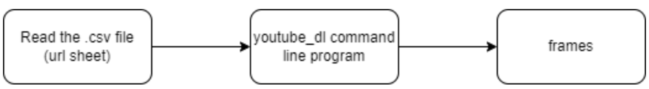
Mathematically it is represented as:

$$zcr = \frac{1}{T-1} \sum_{i=1}^{T-1} I\{s_i s_{i-1} < 0\}$$

here, s_i is the signal of length i and I{x} is an indicator function (=0 if x false else =1).

Image Analysis

As the video part analysis is concerned the full palate of image content analysis technique can be applied, OpenCV is a Open Source very powerful software library of Machine Learning which can be applied on images to extract various information about the image. So the first task involved here is to get the frames from the video as we know videos are a collection of Key-frames that run in a sequence to create a running video.



Initially we took fixed numbers of frames for the entire video i.e. 50 frames but to remove the inconsistency with longer video and optimize our model we increased it to tooking a frame each 2 seconds that is irrespective of video length. Now ,we are ready to obtain different features for video using image analysis-

KeyPoints

Keypoints is an important feature that plays a great role as a part of image properties, in general to understand Keypoints is just like how human brain recognizes different object so quickly, the shapes , the patterns within fractions of seconds. As we are triggered to the most interesting points in an image, so these interesting or interest points are called Keypoints in Machine learning applications. Interest points or the keypoints are kind of the same thing. They are spatial positions in the picture, or points in the image that determines what is fascinating and stand out. Just like Blob detection, which seeks to locate interesting points and spatial regions in the picture, is subset of interest point detection. On e the major benefit of the Keypoints estimation is like it gives the same result regardless how the image orients or changes in positions like rotates, expands or shrinks. The greater the value of keypoint the more appealing/interesting the image and vice-versa.
Basically, to get the mathematical understanding of keypoints we first know where we can find keypoints, Keypoints are defined well in those regions which has a sudden change in the pixel value or we can say immediate change in intensity. Process to calculate Keypoints involved two steps 1 ->to detect the corners and 2 ->threshold for suitable value to get the corners and dilate it to emphasize the feature. For detection of corner a very famous method is used k/as Harris Corner Detector[10]. In this, it basically finds the difference of intensity between two points let say *'u'* & *'v'* in all the directions-

$$E(u,v) = \sum_{x,y} w(x,y) \quad [I(x+u, y+u) - I(x,y)]^2 \qquad (1)$$

where $w(x,y)$ is window function, $I(x+u, y+u)$ is shifted intensity and $I(x,y)$ is intensity.

Window function is basically a small window around a pixel in an image. Transforming the above equation using Taylor series expansion we got-

$$E(u,v) \approx [u \;\; v] \; M \begin{bmatrix} u \\ v \end{bmatrix} \qquad (2)$$

where M = ∑x,y w(x,y) $\begin{bmatrix} I_x I_x & I_x I_y \\ I_x I_y & I_y I_y \end{bmatrix}$

Now as we calculate the values we need to define the edge, flat and where the region is corner for that we define a equation which define if the region is corner or not-

$$R = det(M) - k \cdot trace((M))^2 \qquad (3)$$

where

- $det(M) = \lambda_1 \cdot \lambda_2$
- $trace(M) = \lambda_1 + \lambda_2$
- λ_1 and λ_2 are eigenvalues of M

So, the value of R that depends on λ_1 and λ_2 define the region is corner or not, the threshold condition are-

- If $|R|$ is small : λ_1 and λ_2 are small, the region is flat.
- $R < 0$: $\lambda_1 \gg \lambda_2$ or vice versa, region is flat.
- R is large : λ_1 and λ_2 are large, the region is corner.

Now we got the value of R, threshold it and sum it up to get the values of number of key-points in the image, next to show in the image we dilate it to some value to make the feature emphasized on the image. The below picture the Key-points on two images with larger value of key-point and low value of key-points-

(a) High Keypoints

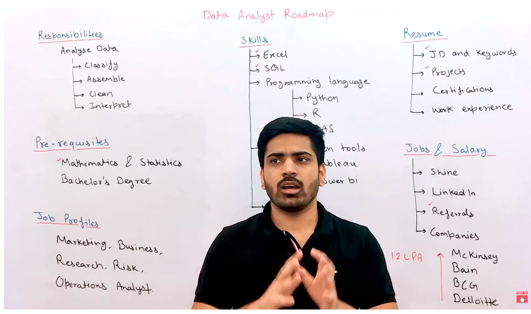
(b) Low Keypoints

Blurriness

Camera wobble and motion blur are both caused by the camera moving in relation to the subject. To calculate the blurriness involved in a video using Open CV and variance of Laplacian. For that we first need to calculate FFT(Fast Fourier Transform)[12] of an image to decide the distribution of high and low frequencies. At this point if we have low amount of high frequencies then the image is considered to be blurry. To get these two parameters the high and low frequency we need to take a single channel of gray-scale image and convolve it with [3 x 3] kernel.

And taking the variance if fall below some threshold value considered to be blur and if above than non blurry. this method works on a basic principle of taking second derivative of image. As the rapid changes to the intensity Laplacian highlights the region. It is also used for edge detection just like Canny edge detection. Here if the image contains the high variance than there is wide spread of responses, and if there is low variance, than there is a tiny spread of response. Tiny spread implies very little edges in turns image is blurred. The below images clearly show the difference of low value of *variance_of_Laplace* and high value of *variance_of_Laplace* to define the image as blurry and sharp-

(a) variance Value : 117.148, Blurry



(b) variance Value : 349.48, Non Blurry

Brightness

In image processing, brightness is described as measured intensity of all the pixels that make up an ensemble that makes up the image once it has been taken, processed, and presented. Pixel brightness is an essential component in images since it is the sole variable that processing algorithms may use to quantitatively modify the image (apart from colour). To calculate the brightness value using HSV method, first to get the RGB[14] values that stands for Red, Green and Blue, determining the value of brightness as-

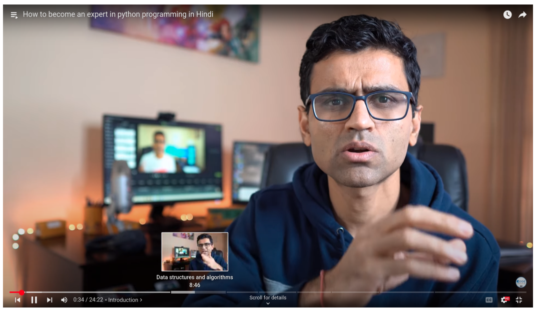
$$B = \sqrt{0{,}068 \cdot (b^2) + 0{,}691 \cdot (g^2) + 0{,}241 \cdot (r^2)} \qquad (4)$$

Here, the three constants represents the different degrees to which each of the primary RGB affects the overall brightness of a color.

Split

The next feature, that is not a kind of part of image processing but plays a greater role in video analysis part and that is splitted videos. As now a days most of the video weather its a NPTEL, MIT or simply YouTube educational video, the video is splitted into sub-parts on the player bar as suppose the video is of duration of 60 mins. The video splits to many part the first part is the intro section, the next one is uses, next basic functions and applications, next simple project and last is the conclusion, which gives a perfect model for viewer to move to the section in which he/she interested in.A simple version of this interface shown below-

To get this splitted sections we used *BeautifulSoup* that is very useful tool in webs-craping. We just scrap/inspects some of the webpages to find out that the class:*'ytp-chapter-hover-container ytp-exp-chapter-hover-container'* implies that it has a split and we simply inspects the whole source



(a) Splitted Video

code to get the sum up means the count value of this class to get the total number of splits in the video.

DATASET

As we know the primary goal of the proposed research project is to analyse and evaluate the content of the visuals. Also we know the videos are from different genres may have a variety of different sets attributes(that are mutually disjoints), deciding their popularity. To get a controllable domain, we are inhibited our model analysis to the Data Analytics videos. On first we started with 40 real time videos from Data Science and analytics, as there are alot of resources available on internet to scrap the videos data on multiple websites, as for our first draft we used youtube videos in order to understand and identify the importance of X-parameters. Here, X-parameters are various features that we later on achieve in two subparts of our model i.e. Video and Audio analysis. In order to collect all these data we first start to get the youtube links from various playlist which involves videos from some youtube creators, some educators and also some of NPTEL playlist that are avialbale on youtube. As we mentioned for the first draft we used at around 40 videos to start with, the n we increased it to around ~ 900 video dataset collected via *webscrapping*, To do this started python library *BeautifulSoup*, First we inspect the website to get identify the *class* to locate the url link. Once found out using method *soup.find_all* and *selenium webdriver*, It is used to open the ChromeDriver to able to open the playlist link in browser and do the above process on the youtube playlist link. By this we able to get the url of the videos to recreate and extend our dataset to around 900+ videos. In this increase Dataset we expand our domain to more genre of videos.

Feature Engineering

Audio

We have already discussed features in detail in approach section. At all we had 37 audio features (20 MFCCs, 12 chroma values, 3 spectral feature, 1 zero crossing rate, 1 zero crossings) but out of these 37, 36 features are themselves an array of values i.e., for each of my sample audio my feature vector is 2 dimensional. To make it one dimensional array of values, including the 5 (mean, median, standard deviation, max, min) values for each feature making an 1x181 array. As value for all these 181 features is extracted by us there is no NaN value. So our feature dataframe with 181 audio features for more than 200 video samples is ready for training and testing.

Image

For the image analysis part, initially we have several parameters like - keypoints, blurness, split, brightness. Each of these parameters depends on the frames taken from the videos so we took out the frames at every 2-3 seconds which means for a video of around one hour we have approx $\sim$ 1000. Also to fill some of the NaN values we used the median value of that particular channel videos and converting it to integers from float. Now for each of the image parameters the mean, median and mode for the 'n' frames of each video has been taken.

Common Features and y variables

Apart from audio and image features, few common features like 'upload date','duration','no of subscribers',etc. have also been taken into account.
As we are checking whether a person likes a video or not after seeing it i.e., probability of liking a video given it is seen, ratio of number of likes to number of views is calculated and scaled between 0 and 1 by dividing by maximum value of all ratios. In this study, y variable is divided in two classes depending on whether a fixed percent of viewers liked the video or not and that fixed number in our case is 20 % selected keeping in mind the class balance for our available dataset.

ALGORITHMS

In this section, the feature selection, ML algorithms used and evaluation metrics used are discussed.

Feature Selection

Recursive feature elimination(RFE)[15] is used for feature selection.In RFE,it starts with all the features and build a model using all those features. Based on the feature importance given by this model, the feature with least importance is discarded and the model is build again with rest of the feature and so on. this process is repeated until only the prescribed number of features are left. The parameter for number of features to be selected is tuned to 50. After getting those number of features, relevant ML algorithms applied in two ways: firstly using all features and later using only these features which were selected here.

Model Implementation

Random Forest Classifier (RFC)

Building a decision tree may tend to over-fit the training data. So in order to avoid this, Random Forest (collection of many decision trees differs slightly from each other) is used. Here, 60 decision trees are used with maximum features equal to 20. Optimum value for max feature in case of classification is generally close to square root of total features which is 13 in our case and we are taking value 20 which is much close to this optimum value 13 as compared to total features 195.

Gradient Boosting Classifier (GBC)

In random forest all trees computes in running parallel to each other. One more way is Gradient Boosting, building trees in serial manner where each tree tries to correct the mistake of previous one. Here, number of decision tree used is generally lesser to avoid over-fitting and also there is no need for randomization i.e., no need to choose maximum feature parameter as in case of random forest but a new parameter known as learning rate is required. In our case, 30 decision trees are used with learning rate of 0.2.

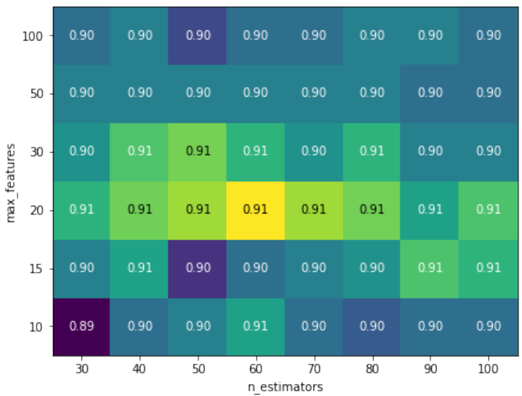
XGBoost Classifier (XGB)

Apart from Gradient Boosting, one more powerful model for implementation is XGBoost (*Extreme Gradient Boosting*). It is more regularized form of gradient boosting as it uses advanced regularization, which improves model generalization capabilities. In our case, 70 decision trees are used with any value for learning rate between 0.05 to 0.3.

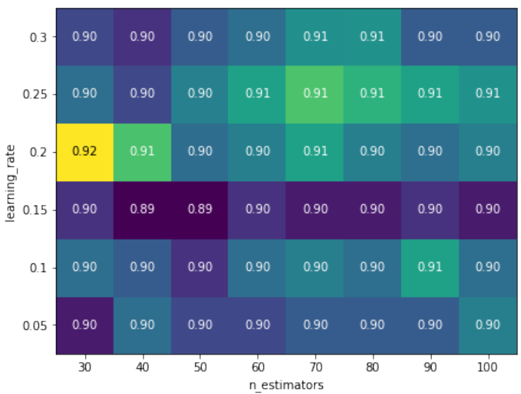
Evaluation Metrics

Grid Search with Cross Validation Score

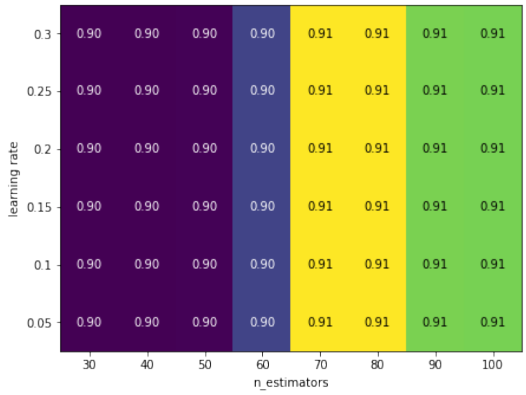
As we know train test split performs random split of data set where we might sometimes got luckier. In order to check how sensitive our model is to the selection of the training data set we use cross validation score. In this study we have used stratified k-fold cross validation because it results in more reliable estimates of performances.



Random Forest Classifier



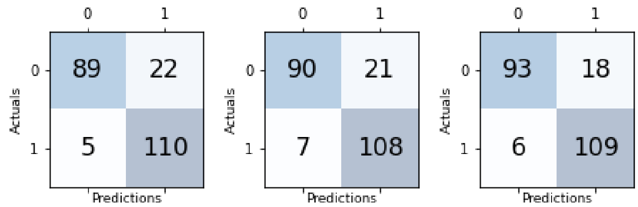
Gradient Boosting Classifier



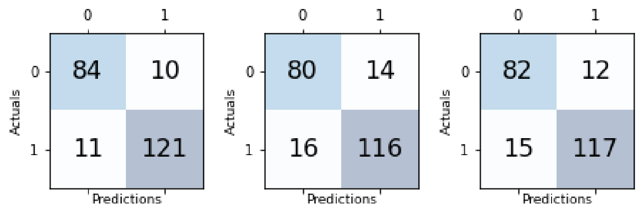
XGBoost Classifier

Confusion Matrix

One of the most comprehensive ways to represent the result of evaluating binary classification.



Before Feature Selection



After Feature Selection

Fig. 6: (a) RFC (b) GBC (c) XGB

RESULTS

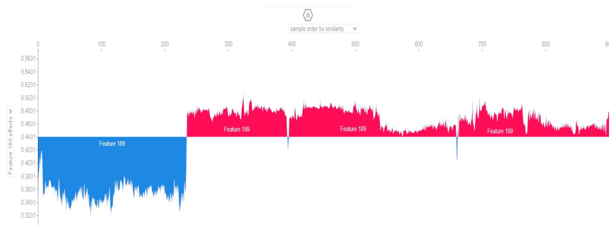
After implementing different models, we got highest accuracy score for XGBoost classifier before feature selection viz. 89.38 %. But after using only features selected using RFE, overall accuracy score improved to 90.70 % viz. for Random Forest Classifier model.

Table 1: Accuracy score for different models

ML MODEL	Feature Selection	
	Before	After
RFC	88.05 %	90.70 %
GBC	87.61 %	86.72 %
XGB	89.38 %	88.05 %

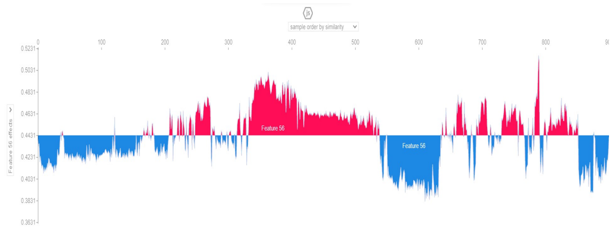
In case of GBC and XGB models, it is clearly visible that accuracy score didn't improve after using selected features only. The reason behind this is discussed in next section.

SHAP plot



(a) shap plot for feature 189(blurriness median)

(a) shap plot for feature 185(brightness mean)



(a) shap plot for feature 56(mfcc2_std)

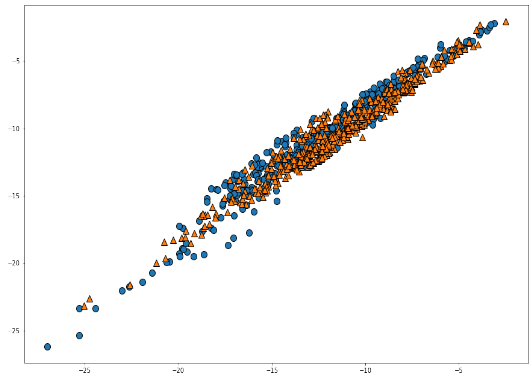
Here, shap plot is made for Random Forest Classifier model. These 3 plots represent variation of top 3 features (which are blurriness median, brightness mean, mfcc2 standard deviation) for complete dataset.

Discussion

Based on the dataset, we can comment on following arguments-

mfcc vs Popularity

As discussed earlier mfcc is a 20 value-vector and for each sample video *mfcc mean* and *mfcc median* are 20 value-vector. Principal component analysis is used to convert these 20 value-vector to a single value which are compared here for each sample video from data-set. This parameter lambda (λ) is intercept of the best fit line in the plot of mfcc mean vs mfcc median.



(a) mfcc mean and median plot

The results obtained is formulated in the given table-

Table 2: mfcc mean and median value relation

mfcc	**y variable**	
	Class: 0	**Class: 1**
mean+$\lambda > median$	125	354
mean+$\lambda <= median$	281	141

As it is distinctly visible that for videos with mean+λ greater than median are mostly belong to class 1 and vice versa.

Blurriness vs Popularity

Blurriness feature consist of three values vector that is converted to a single value using Principal component analysis and tabulated the obtained result in the following table.

Table 3: variation of blurriness with y variable

blurriness	**y variable**	
	Class: 0	**Class: 1**
<-0.4	7	158
-0.4 <= value <0.4	313	337
>= 0.4	86	0

Higher the blurriness value lower the chances of viewer to like the video and vice versa. Also it is observed for a blurriness value in between -0.4 and 0.4 we have almost same results for both classes.

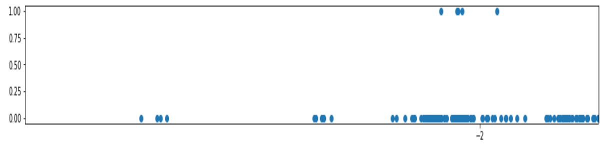
Brightness vs Popularity

As similar to blurriness feature, Brightness feature also consist of three values vector that is converted to a single value using Principal component analysis. From the following plot it is clearly observed that for a brightness value of less than -1.5, videos belongs to class 0.

Similarly for different intervals of brightness values is formulated in the table given below-

Table 4: variation of Brightness with y variable

brightness	**y variable**	
	Class: 0	**Class: 1**
<-1.5	169	12
-1.5<=value<1.5	229	284
>=1.5	8	199



(a) Brightness Plot for Value <-1.5

Duration vs Popularity

Shorter the duration of the video higher the chances of a viewer to like the video

Table 5: variation of duration with y variable

duration	**y variable**	
	Class: 0	**Class: 1**
<15 min	190	360
>15 min	216	135

Upload Date vs Popularity

As such there is no useful relationship between upload date (or in other words, number of days elapsed since upload) and popularity. It can be understood as by the time content of the video is not going to change so it should not affect the viewer for liking or disliking the video after viewing it.

Conclusion

In this paper, we researched data engineering in *Youtube video popularity*. We started with *Content Analysis of videos*. Our motivation for doing so is based on past research work done in this field, we figured out most of research work is done using only information gathered from youtube channel and video properties. We divided the analysis in two parts: images and audio. We tried to answer whether content of a video affects a viewer tendency to like or dislike a video. We discussed the impact of various features of video content on like count is to view count ratio in the discussion section. We found that the features like *brightness*, *blurriness*, *mfcc(speech properties)*, *background music* have significant impact on the viewer's mindset. The results from the study suggest that a same video when uploaded from different youtube channels may have different view count and like count but the like is to view ratio going to remain almost similar in both the cases.

Limitations and Future Scope

In this model, we have considered the 'Like-prediction ratio' as a static number (based on the observations from our sample data set). But this ratio for a more general and accurate model should be dynamic and as per the observed trend, it should decrease with the increase in total view count. The future scope of this work includes the aspect of increasing sensitivity of our boundary-line ratio for popularity definition. Also, the domain of our model is currently restricted to the Data-science study videos, but this can be extended across a much wider field of genres by dynamic prediction ratio. We plan to extend our model to the multiple genre domain set with variety of more content factors included.

Code

Link to git repo

Acknowledgement

We express our heartfelt gratitude to Dr. Faiz Hamid (Assistant Professor, Department of Industrial and Management Engineering, IIT Kanpur) who provided continuous support throughout the project. We also acknowledge the guidance rendered by him.

References

[1] M. Cha, H. Kwak, P. Rodriguez, Y. Ahn, and S. B.Moon. I tube, you tube, everybody tubes: Analyzingthe world's largest user generated content videosystem. In Proc. of IMC, 2007.

[2] K. Lerman and T. Hogg. Using a model of socialdynamics to predict popularity of news. In Proc.World Wide Web Conference. ACM, 2010

[3] 2013. Proceedings of the 22nd ACM international conference on Information Knowledge Management. Association for Computing Machinery, New York, NY, USA.

[4] 2013. Proceedings of the sixth ACM international conference on Web search and data mining. Association for Computing Machinery, New York, NY, USA.

[5] 2016. Proceedings of the 2016 ACM on International Conference on Multimedia Retrieval. Association for Computing Machinery, New York, NY, USA.

[6] Nisa, M.U.; Mahmood, D.; Ahmed, G.; Khan, S.; Mohammed, M.A.; Damaševičius, R. Optimizing Prediction of YouTube Video Popularity Using XGBoost. Electronics 2021, 10,2962.

[7] https://wiki.aalto.fi/display/ITSP/Cepstrum+and+MFCC.

[8] https://en.wikipedia.org/wiki/Chroma_feature.

[9] https://wiki.aalto.fi/display/ITSP/Zero-crossing+rate

[10] https://ai.stanford.edu/ syyeung/cvweb/tutorial2.html

[11] https://medium.com/analytics-vidhya/understanding-keypoints-and-orb-algorithm-f260e6fb640c

[12] J.L. Pech-Pacheco, G. Cristobal, J. Chamorro-Martinez : Diatom autofocusing in brightfield microscopy: a comparative study, 2002

[13] https://tinyurl.com/2esm9vpm

[14] https://www.nbdtech.com/Blog/archive/2008/04/27/Calculating-the-Perceived-Brightness-of-a-Color.aspx

[15] Müller, A., Guido, S. (2016). Introduction to Machine Learning with Python (1st ed., pp. 83-91). United States of America: O'Reilly Media, Inc.

[16] https://doi.org/10.3390/electronics10232962.

[17] https://en.wikipedia.org/wiki/FastFouriertransform.

[18] https://datahacker.rs/004-opencv-projects-how-to-extract-features-from-the-image-in-python/.

[19] https://datahacker.rs/opencv-harris-corner-detector-part1/.

[20] Multivariate Statistical Data Analysis- Principal Component Analysis (PCA)

[21] Aileen Bahl, Bryan Hellack, Mihaela Balas: Recursive feature elimination in random forest classification, 2019

[22] Christiano Couto Gava, Jean-Marc Hengen, Bertram Taetz Didier Stricker : Keypoint Detection and Matching on High Resolution Spherical Images, 2013