A Sentiment Analysis Case Study to Understand How A YouTuber Can Derive Decision Insights From Comments

Chandreyi Chowdhury
School of Advanced Sciences
Vellore Institute of Technology
Vellore, India
chandreyi.surat@gmail.com

Pavitra Mathur
School of Advanced Sciences
Vellore Institute of Technology
Vellore, India
pavitramathur@gmail.com

Sri Rama Vara Prasad Bhuvanagiri
School of Advanced Sciences
Vellore Institute of Technology
Vellore, India
srvprasad.bh@vit.ac.in

***Abstract*—YouTube is considered the biggest platform for content creators to share their content with the world. Usually, a YouTuber aims to give his/her viewers the best content possible by going through the comments of their past videos. On average, the comments can go up to 10 thousand; hence, it becomes practically impossible to go through every comment and get an idea of what the viewers want or expect.**

Our work provides a model based on Python that extracts the comments of a YouTube video which then becomes our dataset. A Machine Learning technique known as Sentiment Analysis (Classification Model) is applied to the dataset extracted to provide the YouTuber with a better understanding of the distribution of the sentiment of his/ her viewers, which in turn helps them get an idea of the thoughts of the viewers and also what the viewers expect from their future videos.

Keywords—YouTube, Sentiment Analysis, Classification, Decision Insights, Case Study

I. Introduction

Social media platforms are considered the easiest and quickest way to reach millions of people at once, be your information for the general public or targeted to a community, age etc. Social media platforms allow one to share their thoughts with a remote person miles away. Many web platforms are used to share non-textual content, such as videos, images, and animations that allow users to add comments for each item. YouTube is the most popular among these apps, which gives you a medium to share your content in the form of videos, with millions of videos uploaded by its users and billions of comments for all of these videos [4]. These videos contain information that can dramatically affect the reputation of a person or an organisation. A simple way to analyse the reputation of a video is by the number of likes and dislikes. If the number of likes is much higher than the number of dislikes, it is good content, but a high number of dislikes compared to likes usually means bad content [1].

However, there are a few ways to measure reputation quantitatively. This raises the importance of automatically extracting sentiments and opinions expressed in social media [5]. Therefore, sentiment analysis on the data extracted from text content, the comments section of a video, can be used to examine YouTube users' perceptions of video content. A text-mining approach becomes the best alternative for interpreting the meaning of each comment. The classification of positive and negative content becomes very important for YouTube users to evaluate how meaningful the published content is based on user opinion comments [7].

This research focuses on performing Sentiment Analysis through a Machine Learning approach on two case study datasets scrapped from YouTube using various classifiers available in Python's Scikit-learn library. The text mining part uses functions from Selenium and BeautifulSoup packages available in Python. Stemming, lemmatisation and text pre-processing on the scraped data is performed using various functions of the package Natural Language Toolkit.

II. Related Works

Several researches have been conducted on the application of Sentiment analysis using approaches like Machine learning, Naive Bias, Lexicon oriented techniques, mBERT etc.

Research by Abbi Nizar Muhammad, Saiful Bukhori, Priza Pandunata develops the sentiment analysis approach using Naïve Bayes - Support Vector Machine (NBSVM) method with a Binary Classification approach. Their approach has a precision of 91%.

Sudhanshu Ranjan, Dheeraj Mekala, Jingbo Shang in their research performs sentiment analysis on the data that switch the language in a conversation referred as code switching and therefore adopts the mBERT approach for the code-switching data. Their framework improves performance across multiple datasets.

Tanvi Mehta and Ganesh Deshmukh uses Techniques like Linear Regression, Support Vector Machine, Decision Tree, Random Forest, and Artificial Neural Network are used and based on their results comparative analysis is done. The decision tree algorithm obtained the minimum RMSE value of 260.193 and ANN acquired 287.919 RMSE value.

In their study Ritika Singh and Ayushka Tiwari created a system that implements six distinct machine learning techniques, including Naive-Bayes, Support Vector Machine, Logistic Regression, Decision Tree, K-Nearest Neighbour, and Random Forest. The system's correctness is then assessed using several assessment metrics, such as the F-score and Accuracy score.

Hanif Bhuiyan, Jinat Ara, Rajon Bardhan, Md. Rashedul Islam research entitled Retrieving YouTube video by sentiment analysis on user comment is an NLP based sentiment analysis approach that finds out relevant and popular video of YouTube according to the search, proved by a data driven experiment in terms of accuracy of finding relevant, popular and high quality video.

The research project by Martin Wöllmer, Felix Weninger, Tobias Knaup, Björn Schuller, and Congkai Sun focuses on automatically analysing speaker sentiment in online movie reviews videos. Their methodology takes into account speech-based emotional audio aspects and encoding beneficial valence content provided by the speaker.

Muhammad Zubair Asghar, Shakeel Ahmad, Afsana Marwat, Fazal Masud Kundi research entitled uses Lexicon based technique for the detection of sentiment polarity based on different lexicons such as wordnet and senti word net.

In their research paper Olga Uryupina, Barbara Plank, Aliaksei Severyn, Agata Rotondi, Alessandro Moschitti In present “SenTube” – a dataset of user-generated comments on YouTube videos annotated for information content and sentiment polarity containing annotations develop classifiers for several important NLP tasks like sentiment analysis, text categorization etc.

The authors S. Nawaz, M. Rizwan, and M. Rafiq offer a novel method for determining the recommendations of effectiveness of a YouTube video content by utilising quantitative sentiment analysis of its comments and replies through the Google API. Their study is titled "Recommendation of effectiveness of YouTube video contents by qualitative sentiment analysis of its comments and replies.".

In their research paper Modelling and Sentiment Analysis of YouTube Users' Comments," Philipp A. Toussaint, Maximilian Renner, Sebastian Lins, Scott Thiebes, and Ali Sunyaev apply Bing (binary), National Research Council Canada (NRC) emotion, and 9-level sentiment analysis to identify user attitudes toward DTC genetic testing-related videos as expressed in the comment section.

Despite these various publications dealing with text based sentiment analysis and multimodal emotion recognition, a method mainly for the YouTubers/ Influencers to be able to detect information and take future decisions based on the comments, which can help them to get the sentiment distribution of that particular video doesn’t exist, to the best of our knowledge.

Hence, this article can be seen as a first attempt to evaluate a procedure to automatically extract or scrape the comments from YouTube, apply various text processing techniques on those scrapped comments, and fit them into a machine learning model that gives out the sentiments categorization, and its total distribution, to provide a comprehensive study of how any content creator on YouTube can use this analysis to their benefit and understand the users’ overall view points on any content creators’ YouTube video.

III. METHODOLOGY

This section elaborates on the step–by–step procedure we have used in order to arrive at the final model. The proposed procedure works in the following four steps shown in Figure. 1. The first step is to select the video one wants to perform sentiment analysis on extraction code so that the comments can be downloaded and stored as a CSV (Comma Separated Values) file. Then using Python packages like Selenium and Beautiful Soup, one can extract the comments. Furthermore, one can perform Sentiment Analysis using our proposed model to get almost accurate classifications on those comments. The accuracy metric also has to be decided, and the validation score needs to be over ninety per cent to be called almost accurate. The steps are described more elaborately in the following sub-section.

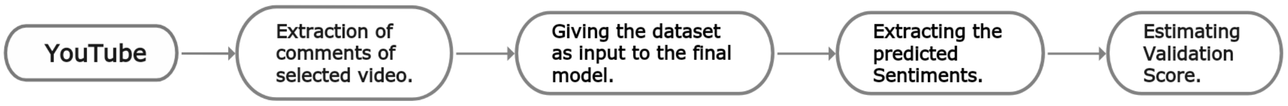
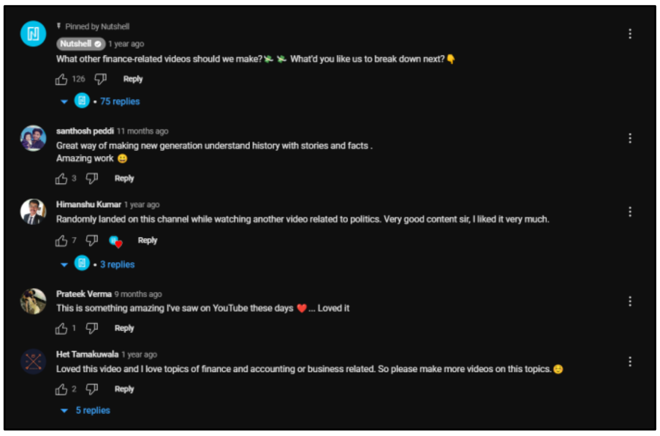

YouTube
Extraction of comments of selected video.
Giving the dataset as input to the final model.
Extracting the predicted Sentiments.
Estimating Validation Score.

Fig. 1 Sentiment Analysis Procedure

A. *Comments Extraction*

As discussed briefly above, comments extraction is the procedure through which the comments of the chosen YouTube video will be extracted into a CSV file. It is done in our model using the web-scraping libraries in Python called *Selenium* and *BeautifulSoup*. Python scripts that automate web browser interaction use the Selenium package. A library called Beautiful Soup makes it simple to gather data from websites. It sits atop an HTML or XML parser, providing Python-based idioms for iterating, searching, and modifying the parse tree. The whole code of the model can be found here. Figures 1, 2 and 3 show the extraction steps and some of the code used. Also, it is to be noted that in our model, we have yet to scrap all of the comments section, as it is against YouTube's policies. We have just scrapped some random amount of comments from each of the videos.



Pinned by Nutshell
Nutshell 1 year ago
What other finance-related videos should we make? What'd you like us to break down next?
126 Reply
75 replies

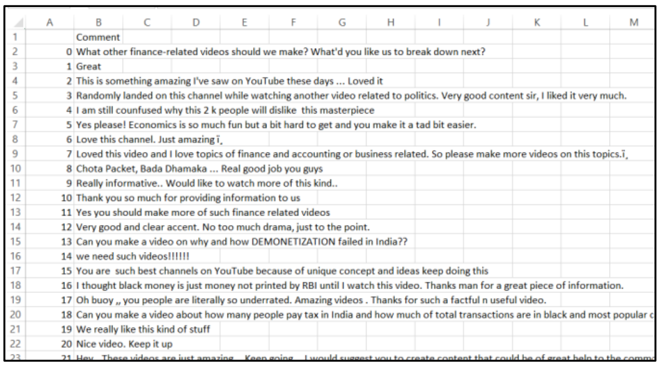
santhosh peddi 11 months ago
Great way of making new generation understand history with stories and facts .
Amazing work
3 Reply

Himanshu Kumar 1 year ago
Randomly landed on this channel while watching another video related to politics. Very good content sir, I liked it very much.
7 Reply
3 replies

Prateek Verma 9 months ago
This is something amazing I've saw on YouTube these days ❤️ ... Loved it
1 Reply

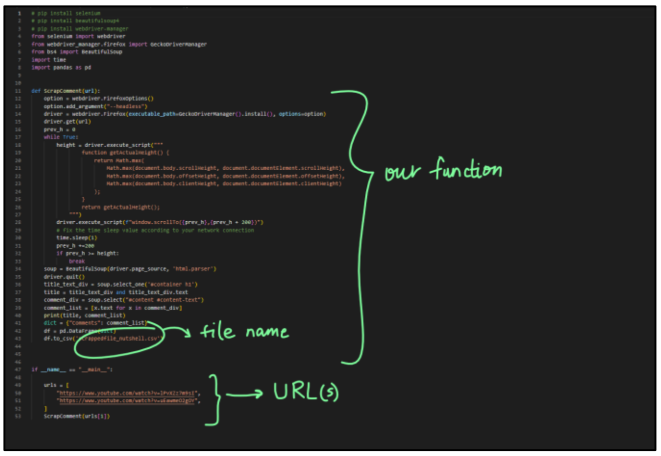
Het Tamakuwala 1 year ago
Loved this video and I love topics of finance and accounting or business related. So please make more videos on this topics.
2 Reply
5 replies

Fig. 2 User Comments on YouTube



	A	B
1		Comment
2	0	What other finance-related videos should we make? What'd you like us to break down next?
3	1	Great
4	2	This is something amazing I've saw on YouTube these days ... Loved it
5	3	Randomly landed on this channel while watching another video related to politics. Very good content sir, I liked it very much.
6	4	I am still counfused why this 2 k people will dislike this masterpiece
7	5	Yes please! Economics is so much fun but a bit hard to get and you make it a tad bit easier.
8	6	Love this channel. Just amazing ï¸
9	7	Loved this video and I love topics of finance and accounting or business related. So please make more videos on this topics.ï¸
10	8	Chota Packet, Bada Dhamaka ... Real good job you guys
11	9	Really informative.. Would like to watch more of this kind..
12	10	Thank you so much for providing information to us
13	11	Yes you should make more of such finance related videos
14	12	Very good and clear accent. No too much drama, just to the point.
15	13	Can you make a video on why and how DEMONETIZATION failed in India??
16	14	we need such videos!!!!!!
17	15	You are such best channels on YouTube because of unique concept and ideas keep doing this
18	16	I thought black money is just money not printed by RBI until I watch this video. Thanks man for a great piece of information.
19	17	Oh buoy ,, you people are literally so underrated. Amazing videos . Thanks for such a factful n useful video.
20	18	Can you make a video about how many people pay tax in India and how much of total transactions are in black and most popular c
21	19	We really like this kind of stuff
22	20	Nice video. Keep it up
23	21	Hey.. These videos are just amazing.. Keep going.. I would suggest you to create content that could be of great help to the comm

Fig. 3 Scrapped Comments



```python
# pip install selenium
# pip install beautifulsoup4
# pip install webdriver-manager
from selenium import webdriver
from webdriver_manager.firefox import GeckoDriverManager
from bs4 import BeautifulSoup
import time
import pandas as pd


def ScrapComment(url):
    option = webdriver.FirefoxOptions()
    option.add_argument("--headless")
    driver = webdriver.Firefox(executable_path=GeckoDriverManager().install(), options=option)
    driver.get(url)
    prev_h = 0
    while True:
        height = driver.execute_script("""
                function getActualHeight() {
                    return Math.max(
                        Math.max(document.body.scrollHeight, document.documentElement.scrollHeight),
                        Math.max(document.body.offsetHeight, document.documentElement.offsetHeight),
                        Math.max(document.body.clientHeight, document.documentElement.clientHeight)
                    );
                }
                return getActualHeight();
            """)
        driver.execute_script(f"window.scrollTo({prev_h},{prev_h + 200})")
        # fix the time sleep value according to your network connection
        time.sleep(1)
        prev_h +=200
        if prev_h >= height:
            break
    soup = BeautifulSoup(driver.page_source, 'html.parser')
    driver.quit()
    title_text_div = soup.select_one('#container h1')
    title = title_text_div and title_text_div.text
    comment_div = soup.select("#content #content-text")
    comment_list = [x.text for x in comment_div]
    print(title, comment_list)
    dict = {"comments": comment_list}
    df = pd.DataFrame(dict)
    df.to_csv("scrappedfile_nutshell.csv")


if __name__ == "__main__":

    urls = [
        "https://www.youtube.com/watch?v=lPvX2z7m9sI",
        "https://www.youtube.com/watch?v=a6wwmeO2gOY",
    ]
    ScrapComment(urls[1])
```

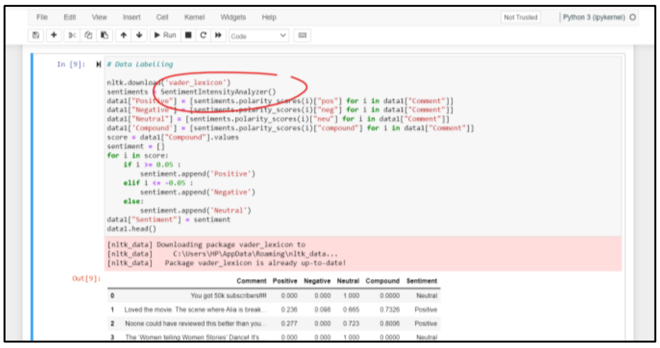
Fig. 4 Extraction Code Snip

B. *Text Preprocessing and Model Building*

After the extraction of the comments is done. The CSV file can be imported to Python, and required pre-processing can be done on it. In our case, we deleted the comments in any language other than English and the symbolic comments. Although the suggestive comments (or emoji-based comments) could also prove to be extremely helpful in describing the user's viewpoint, in our model's reach and context, we could not include those. The target sentiments were decided to be Positive, Negative and Neutral, and the dataset was transformed.

Text sentiment analysis in our model is carried out using the VADER (Valence Aware Dictionary for Sentiment Reasoning) model, which is sensitive to both the polarity (positive/negative) and intensity (strong) of emotion. It may be used immediately on unlabelled text data and is included in the NLTK package. The VADER sentimental analysis uses a dictionary that converts lexical data into sentiment scores, which measure the intensity of emotion. By adding the intensity of each word in a text, one can determine the sentiment score of that text. For instance, the words "love," "enjoy," "glad," and "like" all express a good feeling. Additionally, VADER is wise enough to comprehend the underlying meaning of these terms, such as the negative connotation of the phrase "did not love." Additionally, it recognises how to emphasise words using capitalisation and punctuation, such as "ENJOY."

The SentimentIntensityAnalyzer() function of VADER accepts a string and produces a dictionary of scores for each category of negative, neutral, positive, and compound (computed by normalising the scores above). Figure 4 shows the usage of the same.

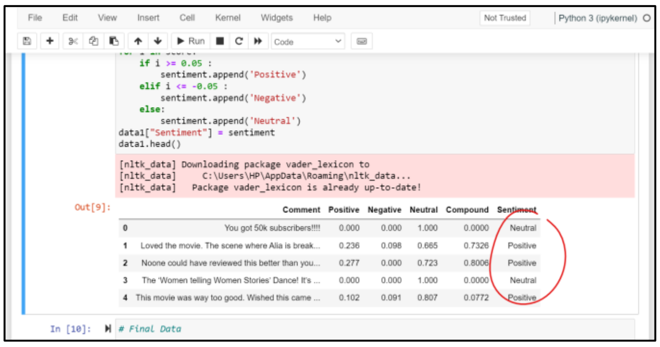


```
In [9]:  # Data Labelling

         nltk.download('vader_lexicon')
         sentiments = SentimentIntensityAnalyzer()
         data1["Positive"] = [sentiments.polarity_scores(i)["pos"] for i in data1["Comment"]]
         data1["Negative"] = [sentiments.polarity_scores(i)["neg"] for i in data1["Comment"]]
         data1["Neutral"] = [sentiments.polarity_scores(i)["neu"] for i in data1["Comment"]]
         data1['Compound'] = [sentiments.polarity_scores(i)["compound"] for i in data1["Comment"]]
         score = data1["Compound"].values
         sentiment = []
         for i in score:
             if i >= 0.05 :
                 sentiment.append('Positive')
             elif i <= -0.05 :
                 sentiment.append('Negative')
             else:
                 sentiment.append('Neutral')
         data1["Sentiment"] = sentiment
         data1.head()

         [nltk_data] Downloading package vader_lexicon to
         [nltk_data]     C:\Users\HP\AppData\Roaming\nltk_data...
         [nltk_data]   Package vader_lexicon is already up-to-date!

Out[9]:
```

	Comment	Positive	Negative	Neutral	Compound	Sentiment
0	You got 50k subscribers!!!!	0.000	0.000	1.000	0.0000	Neutral
1	Loved the movie. The scene where Alia is break...	0.236	0.098	0.665	0.7326	Positive
2	Noone could have reviewed this better than you...	0.277	0.000	0.723	0.8006	Positive
3	The 'Women telling Women Stories' Dance! It's ...	0.000	0.000	1.000	0.0000	Neutral

Fig. 4 SentimentIntensityAnalyser() Function Usage

Finally, we get a complete analysis of every review as positive, negative, or neutral. Figure 5 shows the result quite evidently.



```
    if i >= 0.05 :
        sentiment.append('Positive')
    elif i <= -0.05 :
        sentiment.append('Negative')
    else:
        sentiment.append('Neutral')
data1["Sentiment"] = sentiment
data1.head()

[nltk_data] Downloading package vader_lexicon to
[nltk_data]     C:\Users\HP\AppData\Roaming\nltk_data...
[nltk_data]   Package vader_lexicon is already up-to-date!

Out[9]:
```

	Comment	Positive	Negative	Neutral	Compound	Sentiment
0	You got 50k subscribers!!!!	0.000	0.000	1.000	0.0000	Neutral
1	Loved the movie. The scene where Alia is break...	0.236	0.098	0.665	0.7326	Positive
2	Noone could have reviewed this better than you...	0.277	0.000	0.723	0.8006	Positive
3	The 'Women telling Women Stories' Dance! It's ...	0.000	0.000	1.000	0.0000	Neutral
4	This movie was way too good. Wished this came ...	0.102	0.091	0.807	0.0772	Positive

```
In [10]:  # Final Data
```

Fig. 5 Sentiments Classification

Then we apply text transformation techniques of stemming and Lemmatization. Natural language processing specialists utilise text normalisation techniques like stemming and lemmatisation to get text, words, and documents ready for processing. The stemming process entails creating morphological variations of a root or base word. Stemming algorithms or stemmers are other names for stemming programmes. It frequently helps if the search returns different keyword spellings while looking up information in the text. For instance, "boat" and "boating" could also come up while looking for "boat". The stem for [boat, boater, boating, boats] in this instance would be "boat." Some of the standard stemming methods include PorterStemmer(), LancasterStemmer(), and SnowballStemmer(). We use these three for our model. Lemmatization, as opposed to stemming, applies a morphological analysis to words by considering a language's entire Lexicon and word reduction. Lemma for "was" is "be," while lemma for "mice" is "mouse." Lemmatization does not classify sentences; instead, it analyses the context of a word to determine its part of speech. In our model, we use the standard WordNetLemmtizer() method.

After all the transformation was done, we took the help of several models like Gaussian Naïve Bayes, Decision Tree Classifier, Random Forest Classifier, AdaBoost Classifier, etc., to implement sentiment analysis on our dataset. Some of the implementations of the models are discussed below:

- The first model we used is the *Gaussian Naïve Bayes Classifier*. One of the most simple and effective classification algorithms is the Naive Bayes Classifier. It aids in developing quick machine-learning models capable of making accurate predictions.

- The second model we used in our case studies is the *Logistic Regression* model. Another one of the most often used Machine Learning algorithms within the category of Supervised Learning is logistic regression. A predetermined set of independent factors predicts the categorical dependent variable.
- The third model we have used is the *AdaBoost Classifier*. The Boosting technique known as the AdaBoost algorithm, sometimes called Adaptive Boosting, is used as an Ensemble Method in machine learning. The weights are redistributed to each instance, with higher weights given to mistakenly categorised instances, hence the name "adaptive boosting." For supervised learning, boosting is used to lower bias and variation. It operates under the premise that students advance in stages. Each student after the first is developed from a prior learner except for the first. Simply said, weak students are transformed into strong ones.
- *Random Forest Classifier* is a classifier that uses many decision trees on different subsets of the input dataset and averages the results to increase the dataset's predicted accuracy. Instead of depending on a single decision tree, the random forest uses each tree's forecasts and predicts the result based on the votes of the majority of predictions. The more significant number of trees in the forest prevents higher accuracy and over-fitting. This is our fourth model.
- The K-NN algorithm or the *K Neighbours Classifier* assumes that the new and existing cases are comparable, placing the new instance in the category most like the existing ones. Although the K-NN approach is most frequently employed for classification problems, it can also be utilised for regression. This is our fifth model.
- Furthermore, our final model is the *Decision Tree Classifier*. A supervised learning method called a decision tree can be used to solve classification and regression problems, but it is typically favoured for doing so. It is a tree-structured classifier, where internal nodes stand in for a dataset's features, branches for decision-making, and each leaf node for the classification result.

All these models will be explicitly discussed regarding each dataset in the case studies section (Section IV).

C. Validation Steps

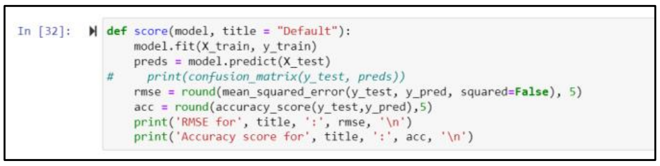
The process of ensuring that a model truly serves its intended function is known as model validation. This usually entails verifying that the model is accurate in the circumstances of its intended application. Validation of a model is done using an accuracy metric. Several accuracy metrics include accuracy score, logarithmic loss, confusion matrix, root mean squared error, the area under the curve, etc. Our model used accuracy score, and root mean squared error as our accuracy metrics. The basic intuitive notion of the formula for accuracy score is:

$$Accuracy = \frac{number\ of\ correct\ predictions}{Total\ number\ of\ predictions} \tag{1}$$

The method that gives the accuracy score in Python is accuracy_score(), found in the package sklearn. Moreover, for the root mean squared error, the intuitive formula is:

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N}(Predicted\ value - Actual\ Value)^2}{N}} \tag{2}$$

The method that gives the root mean squared error in Python is mean_squared_error(), found in the package sklearn. We used a scoring function to give out the accuracy score, and the root mean squared error value for any model inserted. Figure 6 shows the body of the function used.



```
In [32]:  def score(model, title = "Default"):
              model.fit(X_train, y_train)
              preds = model.predict(X_test)
          #     print(confusion_matrix(y_test, preds))
              rmse = round(mean_squared_error(y_test, y_pred, squared=False), 5)
              acc = round(accuracy_score(y_test,y_pred),5)
              print('RMSE for', title, ':', rmse, '\n')
              print('Accuracy score for', title, ':', acc, '\n')
```

Fig. 6 Scoring Function Used In Our Model

D. Final Model

The final model is selected based on the accuracy metric's scores. Whichever model gives the highest score is chosen as the final model. The selection and the scores will vary for every dataset and hence cannot be predicted beforehand. So, the final model selection will be discussed further in the case studies section (Section IV).

IV. CASE STUDIES

A case study of two YouTube videos is presented to understand how a YouTuber can derive decision insights from the comments on his/her videos to improve the content or to know what his/her viewers want. Below are the two enlisted analyses and inferences of the sentiment distributions of the respective videos.

A. Movie Reviews by Sucharita -> Darlings (the 5th of August, 2022)

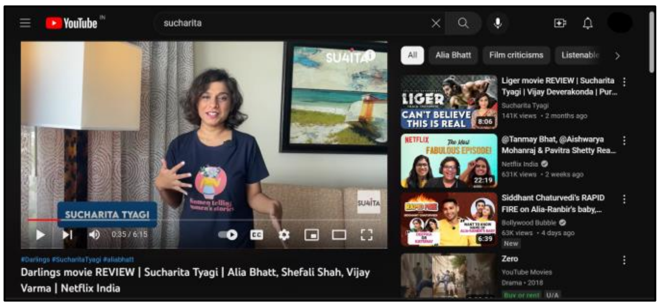

YouTube
sucharita
All
Alia Bhatt
Film criticisms
SUCHARITA TYAGI
Darlings movie REVIEW | Sucharita Tyagi | Alia Bhatt, Shefali Shah, Vijay Varma | Netflix India
Liger movie REVIEW | Sucharita Tyagi | Vijay Deverakonda | Pur...
@Tanmay Bhat, @Aishwarya Mohanraj & Pavitra Shetty Rea...
Siddhant Chaturvedi's RAPID FIRE on Alia-Ranbir's baby,...
Zero

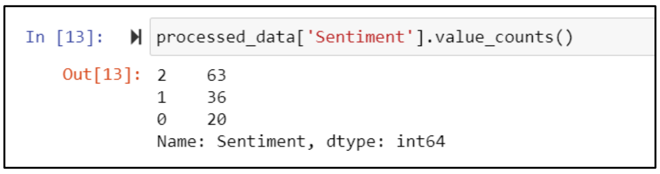
Fig. 8 Video 1 Snip

We chose our first case study for the account "Sucharita" on YouTube, shown in Figure 8. She is a full-time social media influencer whose content mainly concerns reviewing recent movies, series and online stream-able entertainment. We chose her because, being an influencer, she has to create content now and then and keep her audience engaged. It is highly beneficial for her to analyse and see the distribution of her viewers' opinions. What they want more of, what they detest, and so forth. For her account, we proceeded to analyse the video for the movie review: "Darlings", which is among the most viewed videos on her YouTube channel. The video was published on YouTube through the internet on the 5th of August, 2022. There has been a collection of 193 comments on this video over the three months, and we have scrapped 119 comments to work. The exact caption for the video is: “Darlings movie REVIEW | Sucharita Tyagi | Alia Bhatt, Shefali Shah, Vijay Varma | Netflix India”.

TABLE I. FIRST CASE STUDY VIDEO DETAILS

Account	Sucharita Tyagi
Subscribers	62.5K
Video Caption	Darlings movie REVIEW | Sucharita Tyagi | Alia Bhatt, Shefali Shah, Vijay Varma | Netflix India
Published Date	The 5th of August, 2022
Video URL	https://www.youtube.com/watch?v=lPvXZz7m9sI&t=2s
Total Comments	193 (as of the 2nd of November 2022)
Scrapped Comments	119

Proceeding towards the model description and analysis of this dataset, we found that the processed data had 20 negative classes, 63 positive classes and 36 neutral classes. Figure 7 shows the same distribution result we got in the code output.



```
In [13]:   processed_data['Sentiment'].value_counts()

Out[13]:  2    63
          1    36
          0    20
          Name: Sentiment, dtype: int64
```

Fig. 7 Processed Data Distribution For Video 1

Note that 0 denotes the 'negative' class, 1 denotes the 'neutral' class, and 2 denotes the 'positive' class in the Figure. After this step, we unsample the minority classes, the 'neutral' and 'negative' classes. Upsampling involves injecting artificially produced data points (corresponding to the minority class) into the dataset. The counts for each label are nearly equal after this procedure. The model does not tend to favour the class with most members, thanks to this equalisation process and concatenates the new dataframes to the majority class. So finally, our data at this stage contains 205 negative classes, 205 neutral classes and 63 positive classes.

Applying several machine learning classifiers on the data by training the models with 70 per cent of the data and saving the rest 30 per cent for testing, we get the best accuracy score of 0.9577 from the Gaussian Naïve Bayes Classifier. Thus, we conclude that the distribution we initially got is almost correct. It means that approximately 53 per cent of her viewers have a favourable opinion, 30 per cent have a neutral opinion, and the rest 17 per cent have a negative opinion about her video. Figure 9 shows the same diagrammatically.

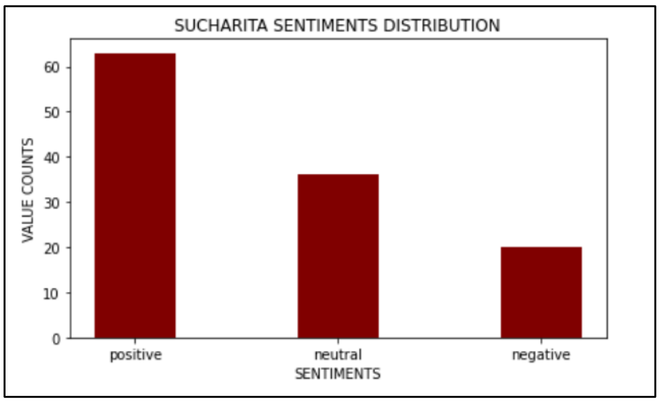


Fig. 9 Video 1 Sentiments Distribution

B. Nutshell -> Black Money (the 21st of May, 2021)

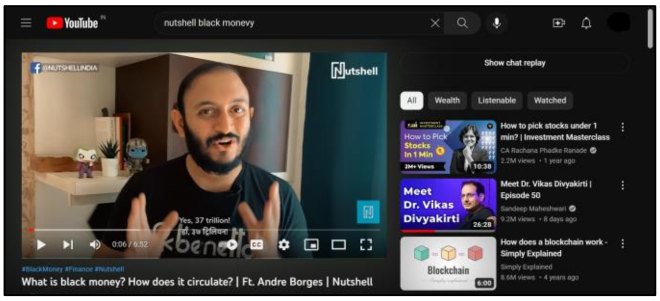

YouTube
nutshell black monevy
Nutshell
Show chat replay
All
Wealth
Listenable
Watched
How to pick stocks under 1 min? | Investment Masterclass
Meet Dr. Vikas Divyakirti | Episode 50
How does a blockchain work - Simply Explained
What is black money? How does it circulate? | Ft. Andre Borges | Nutshell

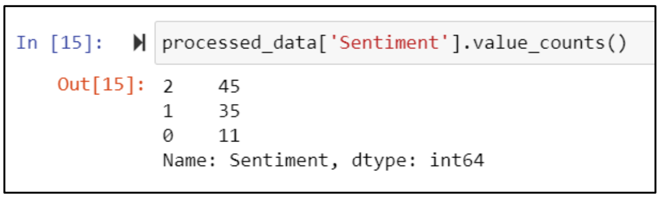
Fig. 10 Video 2 Snip

Secondly, we proceeded to analyse and draw inferences from one of the channel's " Nutshell " videos. It is shown in Figure 10. This channel's main aim is to bring our long lost and un-digged history stories or awareness videos about current affairs in a relatable way, revealing new fact survey videos etc. It can be called an encyclopaedia of knowledge of current and past news. We again chose a top-viewed video on their channel, which was about "Black Money". The exact caption of the video is "What is black money? How does it circulate? | Ft. Andre Borges | Nutshell”. The video was published on YouTube through the internet on the 21st of May, 2021. There are 228 comments on the video for the seventeen months, and our algorithm has scrapped 93 comments to work upon.

TABLE II. SECOND CASE STUDY VIDEO DETAILS

Account	Nutshell
Subscribers	243K
Video Caption	What is black money? How does it circulate? | Ft. Andre Borges | Nutshell
Published Date	The 21st of May, 2021
Video URL	https://www.youtube.com/watch?v=uEawmeO2gOY&t=7s
Total Comments	228 (as of the 2nd of November 2022)
Scrapped Comments	93

For this video, proceeding towards the model description and analysis, we found that the processed data had 11 negative classes, 45 positive classes and 35 neutral classes. Figure 11 shows the same distribution result we got in the code output.



```
In [15]:  processed_data['Sentiment'].value_counts()

Out[15]: 2    45
         1    35
         0    11
         Name: Sentiment, dtype: int64
```

Fig. 11 Processed Data Distribution for Video 2

Again, note that 0 denotes the 'negative' class, 1 denotes the 'neutral' class, and 2 denotes the 'positive' class in the Figure. Unsampling the minority classes, which are the classes for ‘neutral’ and ‘negative’ here (again), we get a dataset containing 205 negative classes, 205 neutral classes and 45 positive classes.

Finally, on this dataset, applying several machine learning classifiers by similarly training the models with 70 per cent of the data and saving the rest 30 per cent for testing, we get the best accuracy score of 0.9781 from the Gaussian Naïve Bayes Classifier. Hence, we can also conclude that the distribution we initially got is almost correct. So, it means that approximately 48 per cent of its viewers have a favourable opinion, 37 per cent have a neutral opinion and the rest 15 per cent have a negative opinion about the video. Figure 12 shows the same diagrammatically.

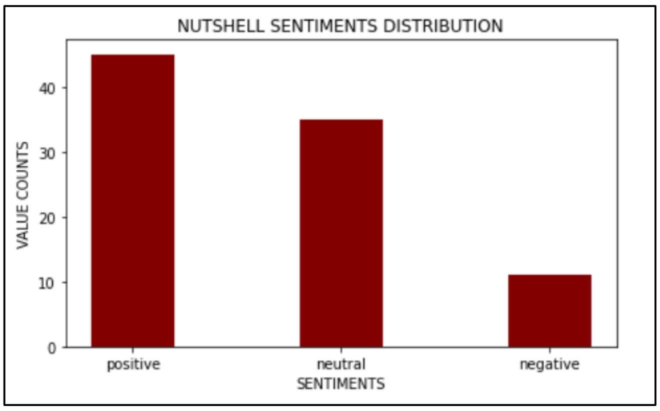


Fig. 12 Video 2 Sentiments Distribution

CONCLUSION

After performing these two case studies, we can say that most of the comments were on the positive side of both videos. Moreover, the respective viewers liked the contents of the two videos and want more of the same kind in future. The ratio of negative comments was meagre, i.e., less than 20 per cent for both videos, which is an ideal scenario for the content creators. These creators or YouTubers want the least number of negative comments, ideally. Hence, through this project, we have concluded that by performing sentiment analysis on the comments of one's videos, they can understand their viewers' opinions as a gist. And then further, if they want, they can go through the positive, negative and the neutral opinions separately according to their purpose. When they would like to thank the viewers for their appreciation, they can reply the positive comments. They can go through the negative comments when they want to improve their content. Furthermore, when they want suggestions for other content creation, they can go through the neutral comments, which generally contain suggestions from the viewers.

REFERENCES

[1] Alexandre Ashade Lassance Cunha(B), Melissa Carvalho Costa, and Marco Aur'elio C. Pacheco "Sentiment Analysis of YouTube Video Comments Using Deep Neural Networks" the 24th of May 2019.

[2] Hanif Bhuiyan; Jinat Ara, Rajon Bardhan, Md. Rashedul Islam "Retrieving YouTube video by sentiment analysis on user comment" the 14th of September 2017.

[3] Martin Wöllmer, Felix Weninger, Tobias Knaup, Björn Schuller, Congkai Sun “YouTube Movie Reviews: Sentiment Analysis in an Audio-Visual Context" the 27th of March 2013.

[4] Muhammad Zubair Asghar, Shakeel Ahmad, Afsana Marwat, Fazal Masud Kundi “Sentiment Analysis on YouTube: A Brief Survey” the 30th of November 2015.

[5] Olga Uryupina, Barbara Plank, Aliaksei Severyn, Agata Rotondi, Alessandro Moschitti “SenTube: A Corpus for Sentiment Analysis on YouTube Social Media” May 2014.

[6] Rawan Fahad Alhujaili, "Sentiment Analysis for Youtube Videos with user comments" the 12th of April 2021.

[7] Abbi Nizar Muhammad, Saiful Bukhori, Priza Pandunata "Sentiment Analysis of Positive and Negative of YouTube Comments Using Naïve Bayes" the 5th of December 2019.

[8] Annamaria Porreca, Francesca Scozzari & Marta Di Nicola "Using text mining and sentiment analysis to analyse YouTube Italian videos concerning vaccination" the 19th of February 2020.

[9] S. Nawaz, M. Rizwan and M. Rafiq “Recommendation Of Effectiveness Of YouTube Video Contents By Qualitative Sentiment Analysis Of Its Comments And Replies” December 2019.

[10] Shanta Rangaswamy Shubham Ghosh, Srishti Jha, Soodamani Ramalingam "Metadata extraction and classification of YouTube videos using sentiment analysis" the 16th of January 2017.

[11] Lakshmish Kaushik, Abhijeet Sangwan, John H. L. Hansen “Metadata extraction and classification of YouTube videos using sentiment analysis” the 9th of January 2014.

[12] Amar Krishna, Joseph Zambreno, Sandeep Krishnan "Polarity Trend Analysis of Public Sentiment on YouTube" the 1st of January 2014.

[13] Philipp A Toussaint Maximilian Renner, Sebas, Scott Thiebes Ali Sunyaev “Direct-to-Consumer Genetic Testing on Social Media: Topic Modeling and Sentiment Analysis of YouTube Users' Comments” the 15th of September 2022.

[14] Rahul Pradhan "Extracting Sentiments from YouTube Comments" the 10th of February 2022.

[15] F Peng, A McCallum “Accurate Information Extraction from Research Papers using Conditional Random Fields” January 2006.

[16] Mike Thelwali “Social media analytics for YouTube comments: potential and limitations” September 2017.

[17] Rhitabrat Pokharel Dixit Bhatta "Classifying YouTube Comments Based on Sentiment and Type of Sentence" the 31st of October 2021.

[18] M. Viny Christanti, Walda1, Tri Sutrisno, “Comments Scraping Application For Review Youtube Content” 2019.

[19] Ritika Singh “Youtube comment analysis” May 2021.

[20] Sudhansu Ranjan, Dheeraj Mekala, Jingbo, Shang “Progressive Sentiment Analysis for Code-Switched Text Data” the 25th of October 2022.

[21] Arpit Khare, Amisha Gangwar, Sudhakar Singh, Shiv Prakash "Sentiment Analysis and Sarcasm Detection of Indian General Election Tweets" the 3rd of January 2022.

[22] Rhitabrat Pokharel Dixit Bhatta "Classifying YouTube Comments Based on Sentiment and Type of Sentence" the 31st of October 2021.

[23] Tanvi Mehta, Ganesh Deshmukh “YouTube Ad View Sentiment Analysis using Deep Learning and Machine Learning” the 11th of May 2022.