A Project Report

on

Sentiment Analysis of YouTube Comments for Understanding User Engagement in Promotion Videos

Submitted in partial fulfillment of the
requirement for the award of the degree of

B.Tech in Computer Science and Engineering

GALGOTIAS UNIVERSITY
(Established under Galgotias University Uttar Pradesh Act No. 14 of 2011)

Under The Supervision of
Mr. Gautam Kumar
Assistant Professor

Submitted By

Abhishek Bhagat
19SCSE1010819
Prabin Kumar Baniya
19SCSE1010816

SCHOOL OF COMPUTING SCIENCE AND ENGINEERING
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
GALGOTIAS UNIVERSITY, GREATER NOIDA
INDIA
December, 2021

SCHOOL OF COMPUTING SCIENCE AND ENGINEERING
GALGOTIAS UNIVERSITY, GREATER NOIDA

CANDIDATE'S DECLARATION

We hereby certify that the work which is being presented in the project, entitled **"Sentiment Analysis of YouTube Comments for Understanding User Engagement in Promotion Videos"** in partial fulfillment of the requirements for the award of the BACHELORS OF COMPUTER SCIENCE AND ENGINNERING submitted in the School of Computing Science and Engineering of Galgotias University, Greater Noida, is an original work carried out during the period of September, 2021 to December, 2021, under the supervision of Mr. Gautam Kumar, Assistant Professor, Department of Computer Science and Engineering/Computer Application and Information and Science, of School of Computing Science and Engineering , Galgotias University, Greater Noida

The matter presented in the project has not been submitted by us for the award of any other degree of this or any other places.

Abhishek Bhagat, 19SCSE1010819
Prabin Kumar Baniya, 19SCSE1010816

This is to certify that the above statement made by the candidates is correct to the best of my knowledge.

Supervisor Name
Designation

CERTIFICATE

The Final Project Viva-Voce examination of Abhishek Bhagat (19SCSE1010819), Prabin Kumar Baniya (19SCSE1010816) has been held on _____ and their work is recommended for the award of B.Tech in Computer Science and Engineering.

Signature of Examiner(s) **Signature of Supervisor(s)**

Signature of Project Coordinator **Signature of Dean**

Date: December, 2021 Place: Greater Noida

Abstract

YouTube is one of the major players digital advertisements. Advertisers can advertise their content/product on the YouTube and can target wide range of audience. There are generally two major ways using which advertisers can advertise their product on the YouTube. First way is, using the Google AdSense. Second way, is directly reaching the content creators and paying them to introduce/advertise their product in their video. Since, the like and dislike ratio of any YouTube channel is a great way to understand how good the channel is and it helps to take decision of whether advertising the product in that YouTube channel is good or bad. But with the change of YouTube policies, now we can't see the number of dislikes in any video. This can be a great problem for the advertisers who want to advertise their product on YouTube and aren't aware of which content creator fits best for them.

To cope with this problem, we can use sentiment analysis to figure out how many videos are receiving positive and negative response on any video. This tool will help the advertisers to find out which YouTube Channel would be best for advertising their product.

To perform the sentiment analysis, we can use Naive Bayes Multi class classification to perform the sentiment analysis. A dataset containing the positive and negative comments can we trained and use to build a model using this algorithm. We can expose this model in the form of an API and can create a frontend application which consumes this API and show the result in graphical way.

As a result of this project, we shall expect a web application that ask for the YouTube comments URL and then starts scraping the comments of that particular video. Once the comments are scraped, we can use our model for finding sentiment for each comment. Based on that, a graphical result shall be shown which consists of how many positive, negative comments are found on that particular video.

YouTube is one of the great places to advertise a product from any creator. Whereas, the likes-dislikes ratio plays a vital role in finding the right creator and asking to advertise the product. The like-dislike ratio helps to understand how much user engagement will be received on any particular channel.

Contents

List of Table

List of Figures

Acronyms

B.Tech.	Bachelor of Technology
M.Tech.	Master of Technology
BCA	Bachelor of Computer Applications
MCA	Master of Computer Applications
B.Sc. (CS)	Bachelor of Science in Computer Science
M.Sc. (CS)	Master of Science in Computer Science
SCSE	School of Computing Science and Engineering

CHAPTER-1 Introduction

1.1 Introduction to YouTube Advertisement

Most businesses want to scale and want more customers throughout the world. Digital Advertisement plays a vital role in helping the business to grow equally. Among different digital advertisement platform, YouTube is the most essential tool. This is because YouTube is the best way to enhance the products and services of company through video advertising and video tutorial. YouTube advertising is an online advertising which can let firm post up a lot of videos to advertising their small company product and information because this online advertising system is very cheap with another advertisement. Those videos will show to all people in the world who have online in YouTube. The internet researchers can upload and watch their advertisement video and tutorials on the internet for free. Nowadays YouTube is a great new approach to promote business. Through the advertising of video in the YouTube, it can quickly present our new ideas and new services in front of people who able to use YouTube. Then the video tutorials are the fastest way to help people to understanding our business. Second, it can introduce our company to help building a bridge between us and client through the YouTube advertising. Furthermore, video advertising in the YouTube also can help to promoting new proposals and announcements of company's product and services to every client. But it also has some disadvantages. For big business, the owners need to pay a higher fee to promote their products. Then the advertising of the business will clearly to show out in YouTube. There are generally two major ways using which advertisers can advertise their product on the YouTube. First way is, using the Google AdSense. Second way, is directly reaching the content creators and paying them to introduce/advertise their product in their video. Since, the like and dislike ratio of any YouTube channel is a great way to understand how good the channel is and it helps to take decision of whether advertising the product in that YouTube channel is good or bad. But with the change of YouTube policies, now we can't see the number of dislikes in any video. This can be a great problem for the advertisers who want to advertise their product on YouTube and aren't aware of which content creator fits best for them.

1.2 Introduction to Sentiment Analysis

Sentiment analysis also referred to as opinion mining, is an approach to natural language processing (NLP) that identifies the emotional tone behind a body of text. This is a popular way for organizations to determine and categorize opinions about a product, service, or idea. It involves the use of data mining, machine learning (ML), and artificial intelligence (AI) to mine text for sentiment and subjective information.

Sentiment analysis systems help organizations gather insights from unorganized and unstructured text that comes from online sources such as emails, blog posts, support tickets, webchats, social media channels, forums, and comments. Algorithms replace manual data processing by implementing rule-based, automatic, or hybrid methods. Rule-based systems perform sentiment analysis based on predefined, lexicon-based rules while automatic systems learn from data with machine learning techniques. A hybrid sentiment analysis combines both approaches. In addition to identifying sentiment, opinion mining can extract the polarity (or the amount of positivity and negativity), subject, and opinion holder within the text. Furthermore, sentiment analysis can be applied to varying scopes such as document, paragraph, sentence, and sub-sentence levels. Challenges associated with sentiment analysis typically revolve around inaccuracies in training models. Objectivity, or comments with a neutral sentiment, tend to pose a problem for systems and are often misidentified. For example, if a customer received the wrong color item and submitted a comment "The product was blue," this would be identified as neutral when in fact it should be negative. Sentiment can also be challenging to identify when systems cannot understand the context or tone. Answers to polls or survey questions like "nothing" or "everything" are hard to categorize when the context is not given, as they could be labeled as positive or negative depending on the question. Similarly, irony and sarcasm often cannot be explicitly trained and lead to falsely labeled sentiments.

Computer programs also have trouble when encountering emojis and irrelevant information. Special attention needs to be given to training models with emojis and neutral data so as to not improperly flag texts. Finally, people can be contradictory in their statements. Most reviews will have both positive and negative comments, which is somewhat manageable by analyzing sentences one at a time. However, the more informal the medium (Twitter or blog posts, for example), the more likely people are to combine different opinions in the same sentence and the more difficult it will be for a computer to parse.

1.3 Formulation of Problem

Dislikes are a huge analytical instrument in the arsenal of YouTube content categorization. The system uses dislikes directly, to analyze the behavior of the user and offer only the most relevant content, in order to keep them engaged within the platform as long as possible. So, each dislike that a particular person hits, is helping YouTube to form the list of recommended videos and improve their experience on the network. From this perspective, creators can have less exposure, if thumbs down are hit by a vast group of the audience. More dislikes from different people would mean the inevitable loss of popularity, due to the fact that your video becomes interesting and relevant for a lesser target group.

Hitting dislike takes effort, and as we know it, this is an alert for creator that there is a possibility that he/she will lose exposure on the platform and thus the content will get less views, likes, and other important metrics. And when then video is less viewed it really loses the potential for monetization. And of course, no sponsors would be eager to support your creations if they see that any creator get a lot of negative feedback. This can be a great problem for the advertisers who want to advertise their product on YouTube and aren't aware of which content creator fits best for them. If they advertise their product on any creator channel which receives a lot of negative response from the audience, it might be a problematic situation. Because, neither the advertiser will get to know how many users genuinely liked and disliked product. So, there is no any transparency on user engagement to the advertiser's side.

1.3.1 Tools and Technology Used

In this project, we are using Naïve Bayes Multinomial Classification Algorithm in order to perform the sentiment analysis. We have a dataset containing user comments which are labeled as positive and negative. First, we perform the data cleaning using NLP library of python. Then, we train and test our model. After that, we expose our model with an API built using Django Framework. On the frontend side, we are using bootstrap to beautify our web page. On the frontend side, user will be asked to enter the URL of the YouTube video. On the backed side, the system will take the YouTube video URL and scrape the comments. Then sentiment analysis is performed and the result is shown in the browser.

CHAPTER-2 Literature Survey/ Product Design

2.1 Current Trend in YouTube Advertisement

YouTube is sometimes not even part of the social media conversation; it's just known as a video platform. But it is a social network. It's also the number-one video platform, second–most-visited website, and second-largest search engine in the world. Marketers should consider it simply because their target audience is definitely there. YouTube has over 2 billion monthly active users and their consumption is going up all the time on mobile, desktop, and even smart TVs. People are viewing content for longer, which is YouTube's goal: to increase time on the platform so they can sell advertising to build business. So, no doubt, YouTube is one of the emerging players in the market in the field of digital marketing and advertisement. There are generally two major ways using which advertisers can advertise their product on the YouTube. First way is, using the Google AdSense. Second way, is directly reaching the content creators and paying them to introduce/advertise their product in their video. Since, the like and dislike ratio of any YouTube channel is a great way to understand how good the channel is and it helps to take decision of whether advertising the product in that YouTube channel is good or bad.

2.2 Challenges in YouTube Advertisement

With the change in YouTube policies, now advertisers are going to struggle to find the right content creator for their product. If advertisers need to figure out whether a YouTube channel fits best for their content promotion, they need to see the views, likes-dislikes ratio. But since like-dislike ratio are no longer visible, it becomes problematic to trust a creator. Because, there won't be any proof to get to know the user engagement ratio in the particular YouTube channel. This will bring negative impact on advertisers choosing the right content creator for their product promotion.

2.3 Proposed Solution

To cope with this problem, we can use sentiment analysis to figure out how many videos are receiving positive and negative response on any video. This tool will help the advertisers to find out which YouTube Channel would be best for advertising their product. To perform the sentiment analysis, we can use Naive Bayes Multi class classification to perform the sentiment analysis. A dataset containing the positive and negative comments can we trained and use to build a model using this algorithm. We can expose this model in the form of an API and can create a frontend application which consumes this API and show the result in graphical way. As a result of this project, we shall expect a web application that ask for the YouTube comments URL and then starts scraping the comments of that particular video. Once the comments are scraped, we can use our model for finding sentiment for each comment. Based on that, a graphical result shall be shown which consists of how many positive, negative comments are found on that particular video.

Chapter 3 - Functionality/Working of Project

3.1 System Development Environment

The following table shows all the system configurations used in the project.

SN	Topic	Tools Used
1	Programming Language	Python, JavaScript
2	External API	YouTube Data API
3	Libraries	NLP, Numpy, Pandas, Scikit-learn
4	Algorithm	Multinomial Naïve Bayes
5	Frontend	Bootstrap
6	Backend	Django
7	Deployment	Heroku

Figure 1.1 – System Development Environment

3.2 Purposed System Architecture

In this project, we are using Naïve Bayes Multinomial Classification Algorithm in order to perform the sentiment analysis. We have a dataset containing user comments which are labeled as positive and negative. First, we perform the data cleaning using NLP library of python. Then, we train and test our model. After that, we expose our model with an API built using Django Framework. On the frontend side, we are using bootstrap to beautify our web page. On the frontend side, user will be asked to enter the URL of the YouTube video. On the backed side, the system will take the YouTube video URL and scrape the comments. Then sentiment analysis is performed and the result is shown in the browser.

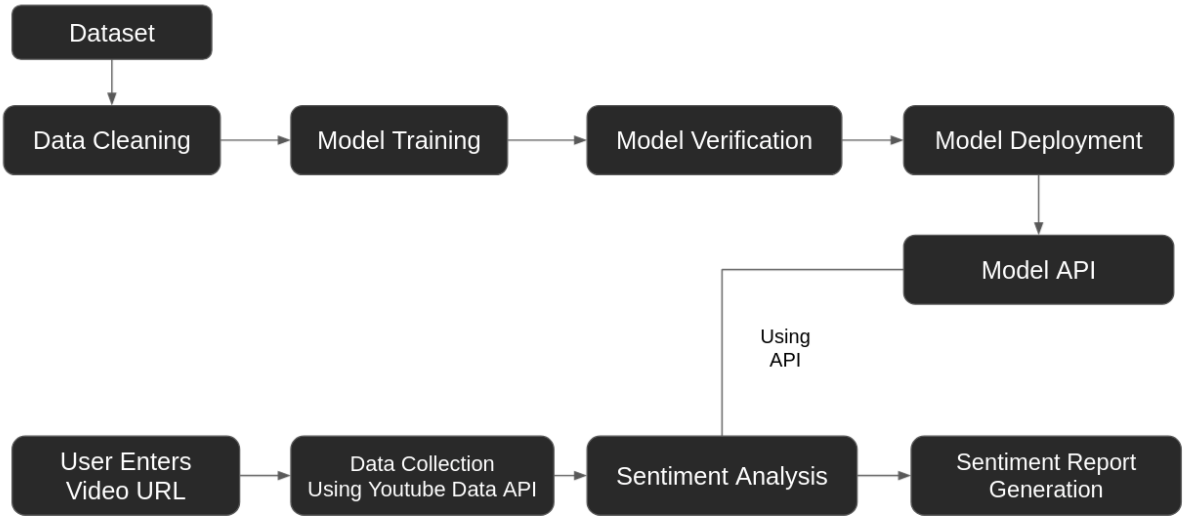

Figure: High Level System Architecture

3.2.1 Natural Language Processing

Natural Language Processing, in short NLP, is an automatic way of presenting or processing human language. It is the branch of AI that helps computer understand, interpret and manipulate human language. NLP draws from many disciplines including computer science & computer linguistics in its pursuit to fill the gap between human communication and computer understanding. It is broadly defined as the automatic manipulation of natural language, like speech & text, by software. We use NLP to extract meaningful data from textual data. Raw text data has to go through various stages just so that we can perform operations on the textual data set. Practical examples are Google voice search, sentiment analysis, etc. NLTK stands for Natural Language Toolkit, which is a platform used for building Python programs that work with human language data for applying in statistical natural language processing (NLP). It contains text processing libraries for tokenization, parsing, classification, stemming, tagging, and semantic reasoning. It is an open-source tool which is one of the most powerful libraries which contain packages to make machines understand human language and reply to them in an appropriate desired response. Sentiment analysis also referred to as opinion mining, is an approach to natural language processing (NLP) that identifies the emotional tone behind a body of text. This is a popular way for organizations to determine and categorize opinions about a product, service, or idea. It involves the use of data mining, machine learning (ML), and artificial intelligence (AI) to mine text for sentiment and subjective information.

3.2.2 Data Cleaning

In data cleaning, we convert the raw text into a list of words that are clean text. It is the process of detecting and correcting or removing corrupt or inaccurate records from a record set, table, or database and refers to identifying incomplete, incorrect, inaccurate, or irrelevant parts of the data and then replacing, modifying, or deleting the coarse data. This involves steps like tokenization, stop word removal, stemming, etc.
Data cleaning is a very crucial step in any machine learning model, but more so for NLP. Without the cleaning process, the dataset is often a cluster of words that the computer doesn't understand.

1. Text Normalization

Text normalization is the process of standardizing text data so that our model can better understand human input with the goal to being able to perform sentiment analysis more effectively. Here in this process, we convert all the uppercase words into lowercase words because computer can get confused between the same word. Here we remove capitalization that would confuse a computer model:

- 'Hey' becomes 'hey'.
- 'Amazon' becomes 'amazon'.
- 'PLEASE FIX' becomes 'please fix'.
- '@AmazonHelp' becomes '@amazonhelp'.

2. Punctuation Removal

Punctuation removal is necessary so that it won't bring ambiguity while training the model. Punctuations, emojis, @ and URL's can confuse AI model they end up being translated unhelpfully. Punctuation also creates noise and impedes NLP understanding because it relates to the tone of the specific sentence, not necessarily the word it is attached to.

3. Tokenization

Tokenization is actually breaking a phrase, sentence, paragraph, or whole text document into smaller units, such as individual words or terminology. Each of these subdivisions is called tokens. It is one of the most important steps in NLP. A tokenizer breaks down random data and native language text into pieces of information that can be considered as discrete elements. Token occurrences in a document can be used directly as a vector representing that document.

Example: For a sentence: "I love singing" it can be tokenized into ["I", "love", "singing"]

4. Stop Word Removal

One of the ways to process pre-processing is to filter out useless data. In natural language processing, useless words, are called stop words. In general, stop word removal means to filter out those words from a text data whose presence makes no any difference during the analysis of our text data. We remove stop words so that our model doesn't get more complicated.

5. Lemmatize/ Stem

Both lemmatize and stemming are text normalizing procedures, used in NLP which is responsible for text pre-processing analysis. Stemming is a rule-based approach where it basically removes prefix or suffix from the inflected word as per the need of the word. The result may not be a actual word. Lemmatization is a process where the inflected word is reduced to its root form. So, lemmatization is preferred over stemming.

Stemming Example:
stem('believe') ⇒ believ
stem('believing') ⇒ believ
stem('believed') ⇒ believ
stem('believes') ⇒ believ

Lemmatization Example:
lemmatize('believe') ⇒ believe
lemmatize('believing') ⇒ believing
lemmatize('believed') ⇒ believed
lemmatize('believes') ⇒ belief.

6. Other steps

Other cleaning steps can be performed based on the data. I have listed a few of them below,

- Remove URLs
- Remove HTML tags
- Remove emoji
- Remove numbers

3.2.3 Classification Algorithms

Classification algorithm is a Supervised Learning method used to identify the category of new observations on the basis of training data. In Classification, the program learns from a given or marked database and separates the new recognition by the number of classes or groups. Such as, Yes or No, 0 or 1, Spam or No Spam, cat or dog, etc. Classes can be called targeted / labels or categories.

Unlike retreat, the variance of the Output is a category, not a value, such as "Green or Blue", "fruit or animal", etc. Since the Separation algorithm is a supervised learning method, this is why it captures input data labelled, which means it contains the corresponding input and output.

Its types are:

- Logistic Regression
- Naïve Bayes
- Decreased Stochastic Gradient
- K-Nearby Neighbors
- Decision Tree
- Random Forest
- Support Vector Machine

3.2.4 Naïve Bayes Algorithm

Naive Bayes Classifier Algorithm is a family of probabilistic algorithms based on applying Bayes' theorem with the "naive" assumption of conditional independence between every pair of a feature.

Bayes theorem calculates probability P(c|x) where c is the class of the possible outcomes and x is the given instance which has to be classified, representing some certain features.
Formula for Bayes Theorem
$P(A|B) = P(A \cap B) / P(B) = [P(B|A) * P(A)] / P(B)$

It is a math theory based on conditional probability to calculate the probability of an event. It calculates the probability of an event based on prior probability of an event. In other words, it shows how much the prior probability of an event affect to the final probability.

- Conditional Probability:

| represents the conditional probability.
P(A|B) = P(A AND B) / P(B)

- Posterior probability

A posterior probability, in Bayesian statistics, is the revised or updated probability of an event occurring after taking into consideration new information.

- P(A|B) probability of occurrence of event A when event B has already occurred.
- Likelihood

P(B|A) probability of occurrence of event B when event A has already occurred.

- Prior probability

Prior probability, in Bayesian statistical inference, is the probability of an event before new data is collected. Prior probability shows the likelihood of an outcome in a given dataset

- Prior probability P(A)

Probability of event A that has already occurred

- Marginal likelihood

P(B), can be ignored

Derivation of Conditional Probability equation
$P(A \cap B) = P(A|B)*P(B)$----(i)

$P(B \cap A) = P(B|A)*P(A)$----(ii)
Equating equations (i) and (ii), we get
$P(A|B)*P(B) = P(B|A)*P(A)$
$P(A|B) = [P(B|A) * P(A)] / P(B)$

Let us understand better by taking an example for finding whether the email is spam or not
x= bunch of emails(the test data)
y= the classes,0 represents not spam; 1 represents spam
Let us put these into Bayes theorem and analyse further
Probability that the email is spam, using Bayes theorem, we get
$P(y=1|x) = P(x|y=1) * P(y=1)/P(x)$
Probability that the email is not spam, using Bayes theorem, we get
$P(y=0|x) = P(x|y=0) * P(y=0)/P(x)$

Consider a school with a total population of 100 persons. These 100 persons can be seen either as 'Students' and 'Teachers' or as a population of 'Males' and 'Females'.
With below tabulation of the 100 people, what is the conditional probability that a certain member of the school is a 'Teacher' given that he is a 'Man'?

So the required conditional probability P(Teacher | Male) = 12 / 60 = 0.2.
$P(Teacher|Male) = P(Teacher \cap Male)/ P(Male)\backslash$
=12/60
=0.2
The Bayes Rule is a way of going from P(X|Y), known from the training dataset, to find P(Y|X).
To do this, we replace A and B in the above formula, with the feature X and response Y.
For observations in test or scoring data, the X would be known while Y is unknown. And for each row of the test dataset, you want to compute the probability of Y given the X has already happened.

3.2.5 Use of Multinomial Naïve Bayes in Sentiment Analysis

Multinomial Naive Bayes classification algorithm tends to be a baseline solution for sentiment analysis task. The basic idea of Naive Bayes technique is to find the probabilities of classes assigned to texts by using the joint probabilities of words and classes. MultinomialNB implements the naive Bayes algorithm for multinomially distributed data, and is one of the two classic naive Bayes variants used in text classification (where the data are typically represented as word vector counts, although tf-idf vectors are also known to work well in practice). Multinomial Naive Bayes classification algorithm tends to be a baseline solution for sentiment analysis task. The basic idea of Naive Bayes technique is to find the probabilities of classes assigned to texts by using the joint probabilities of words and classes. Let's have a brief look at maths.

Given the dependent feature vector (x_1, …, xn) and the class Ck. Bayes' theorem is stated mathematically as the following relationship:

$$P(C_k \mid x_1, \ldots, x_n) = \frac{P(C_k)P(x_1, \ldots, x_n \mid C_k)}{P(x_1, \ldots, x_n)}$$

According to the "naive" conditional independence assumptions, for the given class Ck each feature of vector xi is conditionally independent of every other feature xj for i≠j..

$$P(x_i \mid C_k, x_1, \ldots, x_n) = P(x_i \mid C_k)$$

Thus, the relation can be simplified to

$$P(C_k \mid x_1, \ldots, x_n) = \frac{P(C_k)\prod_{i=1}^{n} P(x_i \mid C_k)}{P(x_1, \ldots, x_n)}$$

Since P(x_1, …, xn) is constant, if the values of the feature variables are known, the following classification rule can be used:

$$P(C_k \mid x_1, \ldots, x_n) \propto P(C_k)\prod_{i=1}^{n} P(x_i \mid C_k)$$

$$\Downarrow$$

$$\hat{y} = \arg\max_k P(C_k)\prod_{i=1}^{n} P(x_i \mid C_k)$$

To avoid underflow, log probabilities can be used.

$$\hat{y} = \arg\max_k (\ln P(C_k) + \sum_{i=1}^{n} \ln P(x_i \mid C_k))$$

The variety of naive Bayes classifiers primarily differs between each other by the assumptions they make regarding the distribution of P(xi|Ck), while P(Ck) is usually defined as the relative frequency of class Ck in the training dataset.

The multinomial distribution is parametrized by vector θk=(θk1,…,θkn) for each class Ck, where n is the number of features (i.e. the size of the vocabulary) and θki is the probability P(xi|Ck) of feature i appearing in a sample that belongs to the class Ck.

The parameters θk is estimated by a smoothed version of maximum likelihood, i.e. relative frequency counting:

$$\hat{\theta}_{ki} = \frac{N_{ki} + \alpha}{N_k + \alpha n}$$

where Nki is the number of times feature i appears in a sample of class k in the training set T, and Ny is the total count of all features for class Ck. The smoothing priors α≥0 accounts for features not present in the learning samples and prevents zero probabilities in further computations. Setting α=1 is called Laplace smoothing, while α<1 is called Lidstone smoothing.

Thus, the final decision rule is defined as follows:

$$\hat{y} = \arg\max_k (\ln P(C_k) + \sum_{i=1}^{n} \ln \frac{N_{ki} + \alpha}{N_k + \alpha n})$$

1. **Data Cleaning**

```
nltk.download("stopwords")
tokenizer = RegexpTokenizer(r'\w+')
en_stopwords = set(stopwords.words("english"))
ps = PorterStemmer()

def getCleanData(text):
    # Converting sting into lower case
    text = str.lower(text)
    # Removing urls from the text
    text = re.sub(r'^https?:\/\/.*[\r\n]*', ' ', text)
    # Generating Patterns for Emoji
    emoji_pattern = re.compile("["
                               u"\U0001F600-\U0001F64F"  # emoticons
                               u"\U0001F300-\U0001F5FF"  # symbols & pictographs
                               u"\U0001F680-\U0001F6FF"  # transport & map
symbols
                               u"\U0001F1E0-\U0001F1FF"  # flags (iOS)
                               u"\U00002702-\U000027B0"
                               u"\U000024C2-\U0001F251"
                               "]+", flags=re.UNICODE)  # removing emoticons
    # Removing Emoji from the string
    text = emoji_pattern.sub(r'', text)
    # removes the digits from the string
    text = re.sub(" \d+", " ", text)
    # Removing the white spaces from the beginning and end of string
    text = text.strip()
    # Tokenizing the text
    tokens = tokenizer.tokenize(text)
    # Removing the stop words
    new_tokens = [token for token in tokens if token not in en_stopwords]
    # Stemming
    stemmed_tokens = [ps.stem(token) for token in new_tokens]
    clean_text = " ".join(stemmed_tokens)
    return clean_text
```

2. **Dataset Loading/ Manipulation**

```
comments_csv_path = os.path.join(BASE_DIR,
'ytcomments/modules/comments.csv')
comments = pd.read_csv(comments_csv_path, sep='\t', names=['comment'])
# Loading data from the csv to dataframe and then cleaning the data
df = comments['comment'].apply(getCleanData)
df.replace('\n', 'NaN')
df.replace(' ', 'NaN')



df = df.to_numpy()

i=0
# Creating a new csv file with cleaned data
cleandedComments_csv_path = os.path.join(BASE_DIR,
'ytcomments/modules/cleanedComments.csv')
with open(cleandedComments_csv_path, "w") as csv_file:
    writer = csv.writer(csv_file, delimiter=',')
    for i in range(0, 450):
        str1 = ''.join(str(e) for e in df[i])
        if (str1 != '\n' and str1 != ''):
            csv_file.write(str1 + "\n")
        else:
            continue



# Getting the training and testing data
dataset_csv_path = os.path.join(BASE_DIR,
'ytcomments/modules/DataSet.csv')
cleandedComments_csv_path = os.path.join(BASE_DIR,
'ytcomments/modules/cleanedComments.csv')
```

3. Model Building

```
data_train = pd.read_csv(dataset_csv_path, encoding="latin-1")

data_testing = pd.read_csv(cleandedComments_csv_path, encoding="latin-1",
names=["Comment"])

labels = data_train.Sentiment

X = data_train.SentimentText.apply(getCleanData)

y = data_train.Sentiment

y

stopset = set(stopwords.words("english"))

vectorizer = TfidfVectorizer(use_idf=True, lowercase=True,
strip_accents="ascii", stop_words=stopset)

X = vectorizer.fit_transform(X)

print(y.shape)

print(X.shape)

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=.60,
random_state=20)

clf = MultinomialNB()

clf.fit(X_train, y_train)
```

Chapter 4 - Results and Discussion

4.1 Expected Outcome

As a result of this project, we shall expect a web application that ask for the YouTube comments URL and then starts scraping the comments of that particular video. Once the comments are scraped, we can use our model for finding sentiment for each comment. Based on that, a graphical result shall be shown which consists of how many positive, negative comments are found on that particular video.

Let's take an example.

We have chosen a YouTube video on the topic of review of Samsung Z Fold 3 phone. On performing the sentiment analysis of the comments made on this video, we have got the following pie chart as a result. A more detailed description of each comment is shown in the deployed project.

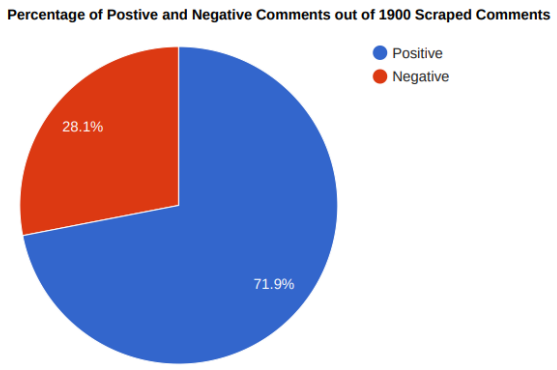

Fig 1.3: Sample Pie Chart Result of Sentiment Analysis

Such type of results helps to understand the reactions of the user and the negative comments which are labeled will help the advertiser to understand the downside of product. Whereas, user can also see and find how well the product is performing the market. If the video is of different category lets assume as Vlogs then performing sentiment analysis of those videos will helps us understand how well the vlog video creator is engaging with the audience.

4.2 Further Study

In this project, there can be further study on how YouTube comments can be well used in order to provide advertisers a data related to creator YouTube channel. Those data will help the advertisers in order to figure out whether the channel will be the best choice for promoting products or not. Whereas, further study of how the model accuracy can be increased, can be carried out.

Chapter 5 - Conclusion and Future Scope

YouTube comments are great way to figure out user engagement on the channel. It helps to understand how well the channel is going apart from the views. YouTube comments can be easily used to perform the sentiment analysis and to find out negative and positive comments. YouTube is one of the great places to advertise a product from any creator. Whereas, the likes-dislikes ratio plays a vital role in finding the right creator and asking to advertise the product. The like-dislike ratio helps to understand how much user engagement will be received on any particular channel. The sentiment analysis of YouTube comments will help the advertisers to get to know how well the creator is doing and the reviews received by the product and as an audience it will help us to determine whether a product is good or bad.

Reference

1. Akhtar, Mohd. (2019). Sentiment Analysis on Youtube Comments: A brief study.
2. Bhuiyan, Hanif & Ara, Jinat & Bardhan, Rajon & Islam, Dr. MD Rashedul. (2018). Retrieving YouTube Video by Sentiment Analysis on User Comment. 10.1109/ICSIPA.2017.8120658.
3. Duffett, Rodney & Edu, Tudor & Negricea, Costel & Milena, Zaharia. (2020). Effect of YouTube Advertising on Intention-To-Purchase and Purchase among Generation Y in a Developing African and European Country. Transformations in Business and Economics. 19. 112-132.
4. Abbas, Muhammad & Ali, Kamran & Memon, Saleem & Jamali, Abdul & Memon, Saleemullah & Ahmed, Anees. (2019). Multinomial Naive Bayes Classification Model for Sentiment Analysis. 10.13140/RG.2.2.30021.40169.
5. Youtube The Best Way For Advertisements Marketing Essay (ukessays.com)
6. How Do Dislikes Affect YouTube Videos: Everything You Wanted to Know (wales247.co.uk)
7. Online advertising - Wikipedia