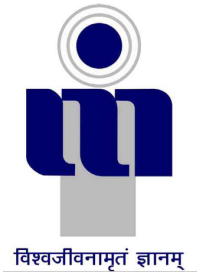
YouTube Video Classifier Browser Plugin using NLTK and Naive Bayes Classifier

A project report submitted in partial fulfillment of the requirements for the award of the degree of

B.Tech.

by

Faiz Malik (2012IPG-033)
Jayant Singh (2012IPG-045)
Kushagra Varshney (2012IPG-052)



ABV INDIAN INSTITUTE OF INFORMATION TECHNOLOGY AND MANAGEMENT
GWALIOR-474 015

2015-16

CANDIDATES DECLARATION

We hereby certify that the work, which is being presented in the thesis, entitled **YouTube video Classifier using nltk and Naive Bayes Classifier** , in partial fulfillment of the requirement for major project of **B.Tech** and submitted to the institution is an authentic record of our own work carried out during the period *May 2015* to *August 2015* under the supervision of **Dr. Ajay Kumar and Dr. Pradip Swarnkar**. We have also cited the reference about the text(s)/ figure(s)/ table(s) from where they have been taken.

Date: Signatures of the Candidates

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

Date: Signatures of the Research Supervisors

ABSTRACT

YouTube is a web place to share videos. It can be seen as worlds largest warehouse of videos. People having Google account can express their views about any video by commenting on them. For popular videos, large number of comments are present, so we can use these comments to classify the video. YouTube Video classifier aims to classify the videos into several categories like News, song, politics, comedy and educational. The classifier will be in the form of browser plugin which will display a pie chart depicting fraction of different content present in video. This will help the user to decide whether to watch it or not depending on his interest and mood. This project uses Nave Bayes Classifier and nltk library to distinguish between different classes of videos. Nave Bayes Classifier is trained using data sets. Data set contains words pertaining to some class and class name. Comments of the video in query are then fetched by the server which compares them against the trained data. Comparison against trained data using Nave Bayes Classifier gives probability of different classes that can define the video. Based on this probability the project forms a pie chart showing different classes. The project is found to be in tandem with the genre of video in query.

Keywords: Nave Bayes Classifier, nltk library, YouTube, Plugin, Data set .

ACKNOWLEDGEMENTS

We are highly indebted to **Dr. Ajay Kumar and Dr. Pradip Swarnkar**, and obliged for giving us the autonomy of functioning and experimenting with ideas. We would like to take this opportunity to express our profound gratitude to him not only for their academic guidance but also for their personal interest in our project and constant support coupled with confidence boosting and motivating sessions which proved very fruitful and were instrumental in infusing self-assurance and trust within us. The nurturing and blossoming of the present work is mainly due to his valuable guidance, suggestions, astute judgment, constructive criticism and an eye for perfection. Our mentor always answered myriad of doubts with smiling graciousness and prodigious patience, never letting us feel that We are novice by always lending an ear to our views, appreciating and improving them and by giving us a free hand in the project. It's only because of his overwhelming interest and helpful attitude, the present work has attained the stage it has.

We are also highly obliged to our parents who constantly helped us through their moral support. Without their motivation we couldn't have delivered the project.

Finally, We are grateful to our Institution and colleagues whose constant encouragement served to renew spirit, refocus attention and energy and helped us in carrying out this work.

(Faiz Malik) (Jayant Singh) (Kushagra Varshney)

TABLE OF CONTENTS

List of Tables and figures

ABBREVIATIONS

NLTK	Natural Language Toolkit
JSON	JavaScript Object Notation
HTML	HyperText Markup Language
JS	Java Script
RE	Regular Expressions
API	Application Programming Interface
SNS	Social Networking Service

CHAPTER 1

Introduction and Literature Review

1.1 Introduction

Currently SNS has been a popular area of research work. Researchers are developing different techniques to predict and analyze human sentiments and outcomes based on the huge data provided by these sites viz. Twitter, Facebook, YouTube. These sites have also provided different APIs [1] to work on huge data present on these sites. Some areas of SNS are are enumerated in the following:

- Influence monitoring and outlook Spotting influencers.
- Mathematical models implied by the growth of the network.
- Clustering of complex networks.
- Recommendation of interesting persons and resources.
- Terrorist identification.
- Privacy preservation.
- Understanding how networks change over time.
- Understanding how people form communities.

- Information diffusion among people in a network.
- Identifying powerful and influential participants.

Surfing videos for entertainment or informative purpose is a part of human's busy life now. Sometimes it is cumbersome to know the genre of video either by watching or reading reviews. To solve this problem the project uses Nave Bayes classifier [2] the project calculates the probability of a every class(genre) present in the video based on comments that is matched against trained data. The trained data has been accumulated from comments of videos of different genres. While making data set for training of classifier, care was taken so that videos of different domains of same genre are included.

1.1.1 Objective

Purpose of the project is to classify the videos in different genres/classes using Nave Bayes classifier so that it is easy to identify type of video without viewing it. It also shows a pie chart depicting fraction of classes present in video in query viz. Comedy, Music, Politics. This classifier will be in the form of a browser plugin, will form pie chart as soon as plugin is clicked with some YouTube video in query in browser.

1.1.2 Methodology

The methodology here involves collection of sentences and data referring to a particular genre/class. This data is then used to train Nave Bayes Classifier. The libraries used are textblob [3], nltk [4], requests, json [5]. The project uses The plugin fetches the video id of the video in query and sends it to server. Server then fetches the comments of the video using YouTube API. The fetched comments are then matched against trained data the results of which are sent to plugin using post method.

1.1.3 Expected Outcomes

The project will give users a way to identify the content in video without watching it. This will also help in sorting videos based on genres. Our API can be used by others to distinguish between videos based on a specific genre.

1.1.4 Benefit to the Society

The project will benefit in monitoring content present in any YouTube video without having human interference. It will help individuals to identify the content in the video without viewing it, this will help users in deciding whether to watch video or not depending upon his mood. The project API can be implemented by other users to sort YouTube videos/channels depending upon genres.

1.2 Literature Review

To deliver our project we reviewed work done on SNS, YouTube APIs and Nave Bayes Classifier. Here, we have recited few of them. The foundation of SNS is based on the huge data that has been accumulated by these sites from human's response. These sites provide APIs to use the huge data for the benefit of humanity.

We studied a paper on Contextual Feature Based One-Class Classifier Approach for Detecting Video Response Spam on YouTube [1] the paper discusses methods used to find malicious content on YouTube videos. The paper uses different classification techniques. We studied a paper on Semantically Enriched Machine Learning Approach to Filter YouTube Comments for Socially Augmented User Models [6]
Mining YouTube to Discover Extremist Videos, Users and Hidden Communities [7] was revived to get an insight into YouTube video classification. To get information about Naive Bayes we studied Naive Bayes classifiers by Kevin P. Murphy [8]
To study practical applications and to know how we can implement it in our project we

studied Thumbs up?: sentiment classification using machine learning techniques [9]
To get an insight into SNS application of nltk we studied Classifying twitter data with Nave Bayes Classifier [10]. Machine learning in automated text categorization [11] gave us an insight into machine learning techniques. We gathered the following information about machine learning from this paper

Machine learning is make a computer learn to optimize a performance criterion using test data or examples of the past decisions . It is performed by the executing an algorithm based program on a computer to optimize the parameters of a given model, which can be predictive to make decisions in future or descriptive to gain knowledge from data or both. In this learning process there are two things that are needed to be achieved. First, the training needs to be performed with an efficient algorithm to solve an optimized problem, and to store and process the data. Second, the trained model needs an algorithm and efficient representation for inference.

1.2.1 Related Work

The similar papers for comment classifier only filter the comments from spam or the comments which are irrelevant to the user [12]. The work is aimed to categorize the video in several categories like inappropriate comments,spam etc.
Classification of Tweets is also related to it in which the tweets are classified into different categories to improve information filtering [13].
To study the behaviour of spammers and promoters so that we can know who is the legitimate user and who are not . It also talks on binary classification to detect person who spams on YouTube.

1.2.2 Current Fallouts and Gaps

Currently the research work are trying to filter out the irrelevant comment so that youtube user can have a idea about the video by reading the relevant comments. Users have to go through and read many comments to get the idea of the video. And there is nothing which can classify the video according to genre whether the video belongs to comedy category, song or etc. User is bound to watch most part of it and in case the video is not up to the level of genre which the user want then this is the waste of time and as a youtube is a internet based video player it takes a lots of data to play. This plugin will solve the problem by reading comments it self and simultaneously show the genre on a pie chart showing the proportion of genre it consists. In present related work there is nothing like classifying the video and that too with its comment. Which gave us a great motivation to work in this area.

CHAPTER 2

Data Extraction, Processing and Analysis

In this chapter detailed methodology to extract data for training, process the queried video's comments against the trained data and analysis of the results using NLP tools to specify which genre the video lies in. Let us first discuss what the YouTube and YouTube comments are in brief. YouTube is a SNS to share videos. Users can comment on the video displaying his emotions through it.

2.1 Generating training Data

Our project consist of training data against which the genre of video is tested using nave bayes classifier. The data has been collected from the comments on videos of different genres. The data set contains keyword which points to some specific genre.

E.g.

- ('song', 'song'), ('singer , arjit singh ', 'song'), ('song', 'song'), ('song', 'song'), ("beats ", 'song') this data corresponds to class 'Song'. ("governance", 'politics'), ("Mahatma Gandhi", 'politics'), ("Jayaprakash Narayan", 'politics'),
- ("speech", 'politics'), ("democratic", 'politics'), ("Naredra modi", 'politics') corre-

sponds to 'Politics' class.

- ('hahaa hahahaha', 'comedy'), (' lolllll','comedy'), ('lol','comedy'), (' smile ,humour', 'comedy'), ('funny', 'comedy'), ('entertaining', 'comedy') pertains to Comedy class.

Then the project will train the Nave Bayes Classifier against this data set.
Formula used for algorithms [14] - :

$$\phi_{k|label=y} = P(x_j = k|label = y) \tag{2.1}$$

$$\phi_{k|label=y} = \frac{\sum_{i=1}^{m}\sum_{j=1}^{n_{(i)}}\{x_j^i = k\, and\, label_{i=1}\} + 1}{(\sum_{i=1}^{m} 1\{label_i = y\}n_i) + |V|} \tag{2.2}$$

$\phi_{k|label=y}$=probability that a particular word in document of label(neg/pos) = y will be the kth word in the dictionary.
n_i =Number of words in i_{th} document.
m = Total Number of documents.

$$P(label = y) = \frac{\Sigma_{i=1}^{m} 1\{label_i = y\}}{m} \tag{2.3}$$

To calculate the score of each genre such as comedy, politics, news etc. we use

$$Decision = \log P(x|label = comedy) + \log P(label = comedy) \tag{2.4}$$

Similarly calculate

$$Decision = \log P(x|label = Politics) + \log P(label = Politics) \tag{2.5}$$

And the probabilities are used to make a pie chart.

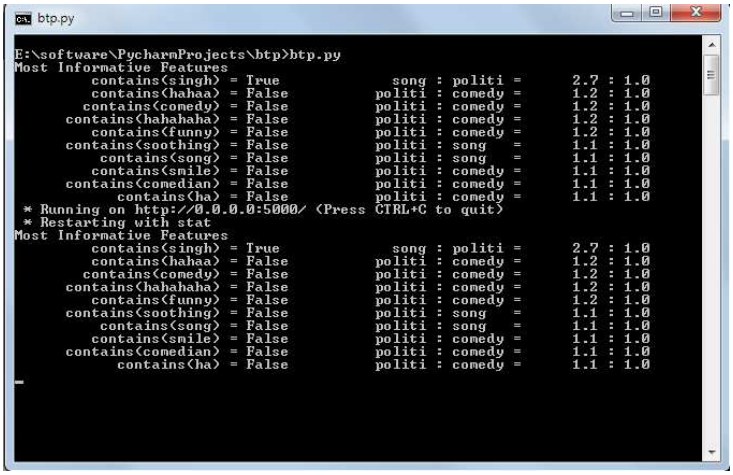


Figure 2.1: Most informative features used in Naive Bayes Classifier

Note that the first element of each tuple is now a dictionary, which is hashable. Now that your data is in place and the first element of each tuple is hashable, you can train the classifier like:

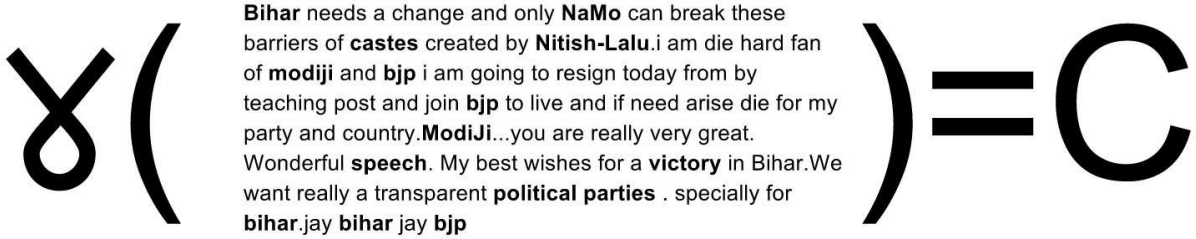


Figure 2.2: Bag of words

Figure 2.3: Bag of words represented using subset of words

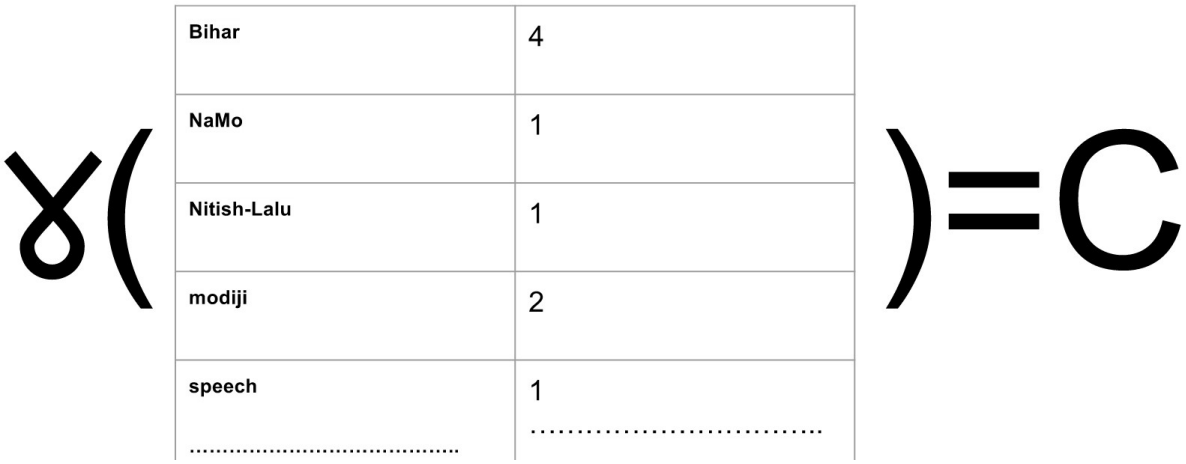


Figure 2.4: Vector of Words

2.2 Data Extraction

In data mining the extraction plays a vital role , to analyse the data and to make decision depending upon those data. We have extracted our Data from youtube which are the comments from the user who have seen the video .The extracted comments contains the sentiments which point towards a genre which we will try to know. To get the comments we have used youtube API version 3 which on providing video id gives all the comments of that video.comments that are obtained is in JSON format.

2.2.1 Web Server

Web server provides a linkage between chrome plugin and python script which is running at backend . By taking the video id from the browser extension for testing of that video comments and returning back the processed form of data inform of a json having probability of each genre.

2.3 Analysis of Data

Comments on the video posted by different users are extracted using textblob and YouTube API from the video in query are tested against trained data using Naive Bayes Classifier. The classifier then calculates the probability using classes as created using training data set. the probability is then sent to the browser in JSON format the browser plugin then converts the JSON data into user readable and displays pie chart.

2.4 Architecture

The server in which a python script is running is given a training data, the classifier is trained using this data. Whenever this chrome extension is clicked the video id of the YouTube video in query is sent to web server in which python script is running . The script receives the video id and uses the YouTube API to fetch comments of the video whose id has been fetched. Process of fetching comments:-

Figure 2.5: Id of the video in query fetched from the URL of the video.

https://www.googleapis.com/youtube/v3/commentThreads?key=AIzaSyDR_p3sWUI9R21TaewInZOBpKWrS19EfK4&textFormat=plainText&part=snippet&videoId=kffac

xfA7G4&maxResults=50

Python script attaches the video id received at the underlined place in the above URL and then collects the response of the link which is in json.

{

```
"kind": "youtube\#comment ThreadList Response",
"etag": etag,
"nextPageToken": string,
"pageInfo": \{
  "totalResults": integer,
  "resultsPerPage": integer
```

},

"items": [

comment Thread Resource

] }

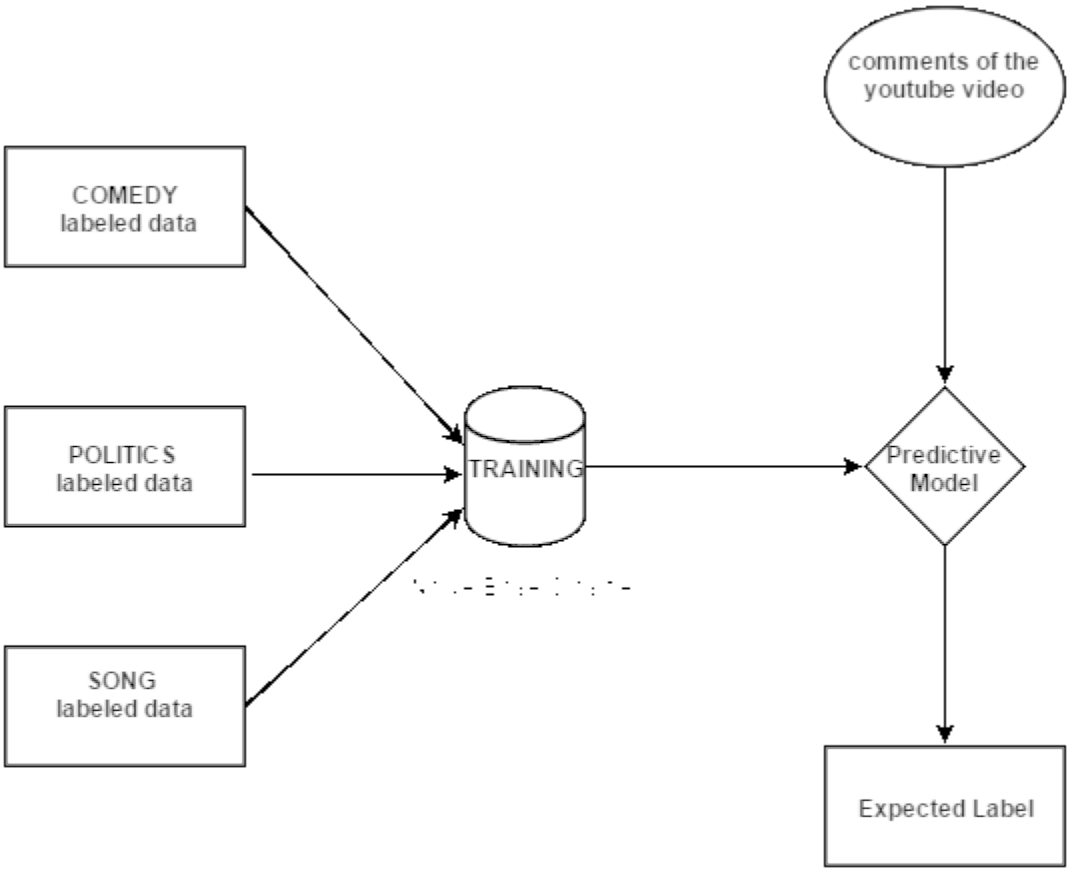


Figure 2.6: flowchart representing the training and fetching of comments data

The fetched comments are our test data which the nave bayes classifier uses to find out probability of each genre in the test data.

2.4.1 Nave Bayes Classifier

The Nave Bayes Classifier is formulated on the Bayesian theorem and it is helpful when the scope of input is high. Though Nave Bayes is simple it can beat expectations when compared to complex models [15]. To explain Naive Bayes Classifier concept , we have stated a example above . As shown, the points are classified as Comedy (GREEN), Songs

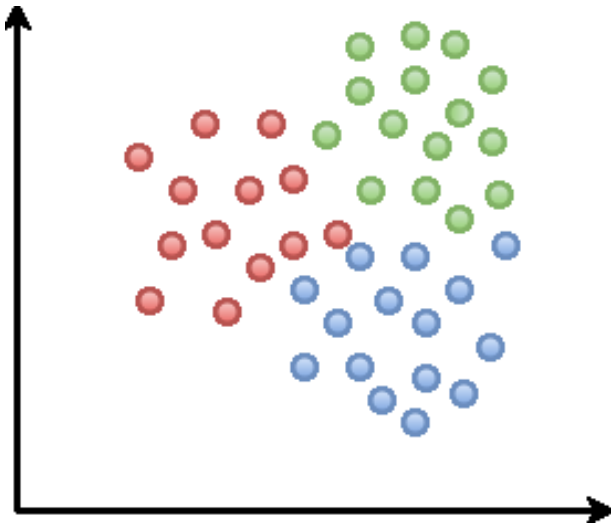


Figure 2.7: Graph representing data related to a particular genre present in video.

(RED) or Politics (BLUE). What we are doing is classifying the new cases which arrive , i.e., decide under which class level they come, based on the objects currently existing. We know in above figure it is shown that there are twice as many of GREEN points as BLUE and as RED, we can say that a new case (case whose class is to be found) is twice as probable to have congregation GREEN rather than Blue and Red. In this analysis, this is known as the prior probabilities. Prior probability as name suggests depends on past experience,here the percentage of Green points,Blue points and RED points, and they are used to predict the outcomes before they really happen.

Thus, equations:

$$Prior\,probability\,for\,Green \propto \frac{No.\,of\,Green\,objects}{Total\,no.\,of\,objects} \tag{2.6}$$

$$Prior\,probability\,for\,Red \propto \frac{No.\,of\,Red\,objects}{Total\,no.\,of\,objects} \tag{2.7}$$

$$Prior\,probability\,for\,Blue \propto \frac{No.\,of\,Blue\,objects}{Total\,no.\,of\,objects} \tag{2.8}$$

we know there are total of 80 points, 40 of them are Green ,and 20 of them are BLUE and 20 of them are Red, Our prior probabilities for a particular class congregation are:

$$Prior\ probability\ for\ Green\ \propto\ \frac{40}{80} \tag{2.9}$$

$$Prior\ probability\ for\ Red\ \propto\ \frac{20}{80} \tag{2.10}$$

$$Prior\ probability\ for\ Blue\ \propto\ \frac{20}{80} \tag{2.11}$$

Having stated our own prior probabilities , we will now try to classify a new point which

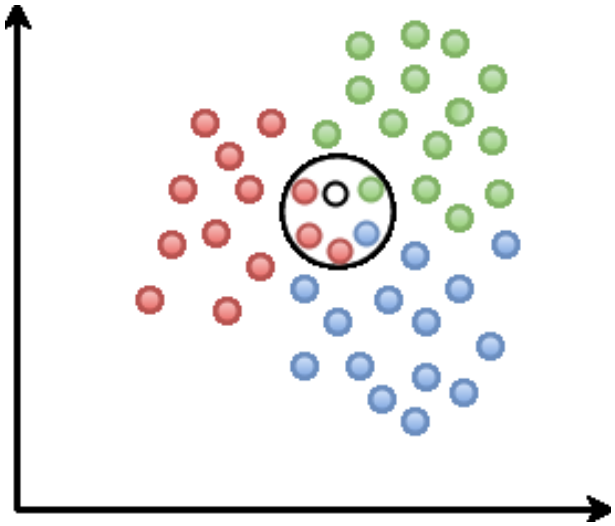


Figure 2.8: Classifying a new object

is a circle of WHITE colour . Since the members are well clustered together,we can have assumption that the more RED (or BLUE or WHITE) points in the area of X, then it is much more likely that the new arriving case which are here white belong to particular colour. To calculate this likelihood, Make a circle around the X which encircles a number of objects not depending upon of their labels of class. After this we calculate number of objects which are encircled belonging to each of the given class label. we can calculate the likelihood of each colour:

$$Likelihood\ of\ X\ given\ Green\ \propto\ \frac{No.\ of\ Green\ in\ the\ vicinity\ of\ X}{Total\ no.\ of\ Green\ cases} \tag{2.12}$$

$$Likelihood\ of\ X\ given\ Red\ \propto\ \frac{No.\ of\ Red\ in\ the\ vicinity\ of\ X}{Total\ no.\ of\ Red\ cases} \tag{2.13}$$

$$Likelihood\,of\,X\,given\,Blue\ \propto\ \frac{No.\,of\,Blue\,in\,the\,vicinity\,of\,X}{Total\,no.\,of\,Blue\,cases} \tag{2.14}$$

From the equations above, it is clear that the Likelihood of X given BLUE and GREEN is smaller than Likelihood of X given RED, since the circle encircles 1 GREEN, 1 BLUE point and 3 RED ones. Thus:

$$Probability\,of\,X\,given\,Green\ \propto\ \frac{1}{40} \tag{2.15}$$

$$Probability\,of\,X\,given\,Red\ \propto\ \frac{3}{20} \tag{2.16}$$

$$Probability\,of\,X\,given\,Blue\ \propto\ \frac{1}{20} \tag{2.17}$$

the prior probabilities shows that X may belong to GREEN one (it is given that there are double as many GREEN compared to BLUE and RED) the likelihood shows otherwise(opposite), that the class congregation (membership) of X is RED (it is given that there are more RED objects in the vicinity of X than BLUE and GREEN). In this Bayesian analysis, the resultant classifications is generated by combining both sources of information, i.e.,the likelihood and the prior, to form a posterior probability using the Bayes' rule.

Posterior probability of X being Green $\propto$ Prior probability of Green $\times$ Likelihood of X given Green=$\frac{4}{8} \times \frac{1}{40} = \frac{1}{80}$
Posterior probability of X being Red $\propto$ Prior probability of Red $\times$ Likelihood of X given Red=$\frac{2}{8} \times \frac{3}{20} = \frac{3}{80}$
Posterior probability of X being Blue $\propto$ Prior probability of Blue $\times$ Likelihood of X given Blue=$\frac{2}{8} \times \frac{1}{20} = \frac{1}{80}$

Finally, we X is classified as RED because its class congregation have the largest posterior probability.

CHAPTER 3

Results and Discussions

3.1 Results

Figures showing results.

Table 3.1: Table representing probability values of different videos

Video Name	Comedy	Politics	Songs	Genre
PM Modi's speech in Bhagalpur.	0.04	0.9265	0.0335	Politics
Comedy Movie from Bollywood	0.9904	0.0023	0.0073	Comedy
Comedy Nights With Kapil	0.9996	0.0002	0.0002	Comedy
'Tu Jo Mila' VIDEO Song - K.K.	0.0264	0.0788	0.8948	Song
Russell Peters Outsourced	0.9999	0.0	0.0001	Comedy
Katy Perry - Dark Horse	0.0079	0.0261	0.966	Song
Ultimate Rajesh Khanna Jukebox	0.0433	0.0236	0.9331	Songs
Katy Perry ft. Juicy PARODY	0.7611	0.0015	0.2374	Comedy/song
Rahul Gandhi addresses party	0.1659	0.7372	0.0969	Politics

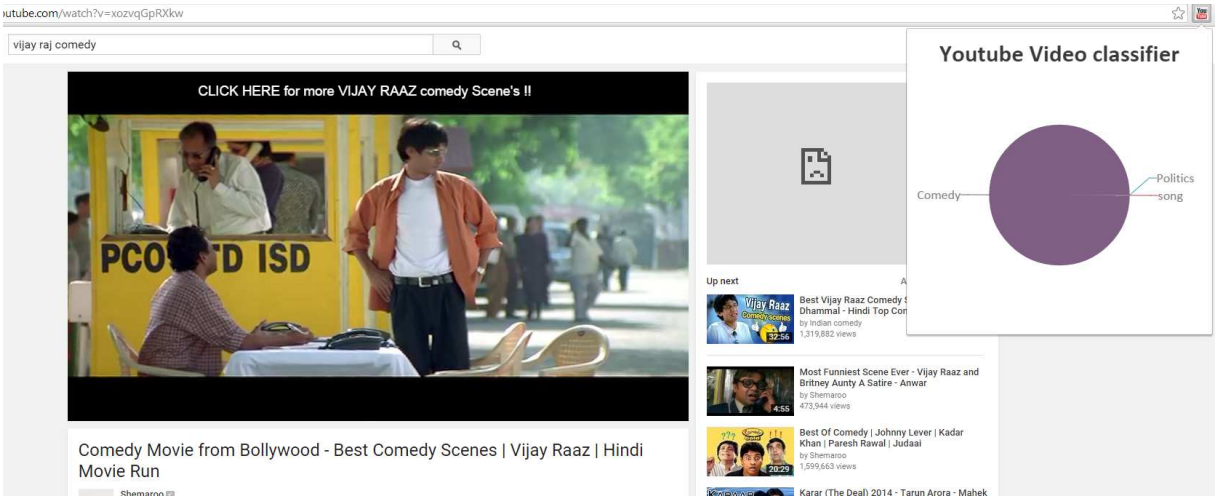


Figure 3.2: Results of Comedy Video

Values of probability of different classes(genres) for Comedy video:

`{politics:0.0022,comedy:0.9903,song:0.0075}`

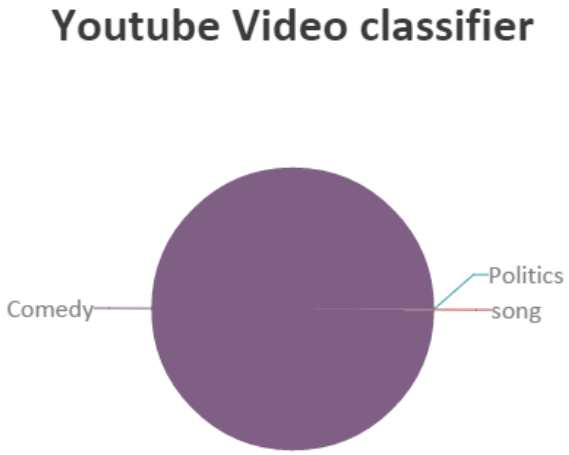


Figure 3.3: Pie chart representing probability values of Comedy video

Figure 3.4: Results of Song Video

Values of probability of different classes(genres) for Musical video:

`{politics:0.002,comedy:0.0264,song:0.9734}`

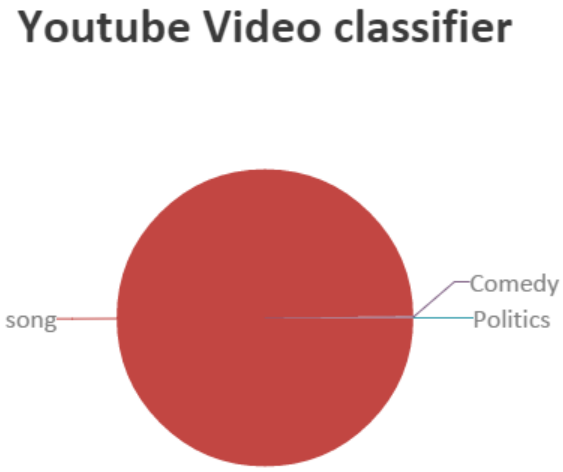


Figure 3.5: Pie chart representing probability values of Musical video

Figure 3.6: Results of Political Video

Values of probability of different classes(genres) for Political video:

`{politics:0.9265,comedy:0.04,song:0.335}`

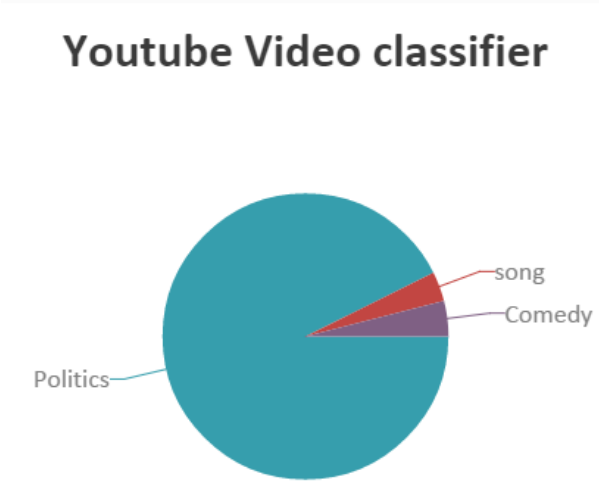


Figure 3.7: Pie chart representing probability values of Political video

3.2 Discussions

The probabilistic values of the videos based on comments on them are found to be accurate. The project also satisfy the case where multiple classes are present in same video. For e.g. in table 3.1 result of the *Katy Perry ft. Juicy PARODY* video is both comedy and songs since the video is a parody hence it contains both musical and comedy content. The project can further be extended by increasing the number of classes. The project can further be accelerated to gather usefulness of a video based on number of likes and sentiments of comments.

CHAPTER 4

Conclusions and Future Scope

We were able to show the genre of the video on a single click of a Browser extension in form of a user interactive pie chart, which is always present so whenever user wants to use it he or she can use it and get the related information on a pie chart without wasting his or her time (buffering) and internet uses.

4.1 Future Scope

A web portal will be designed which will store the data of all videos queried by user and save it on a database. The database can then be used to sort the videos according to genre.

The portal would work as:- It will contains categories viz. Music, Politics, Comedy, Educational, Spiritual, News, Abusive, Adult. So that if user is interested in particular type of genre he will get a list showing the videos of that particular genre.

When the user uses our plugin to categorize the video ,the portal will save the results in itś database. The click able link with Description would be displayed in that particular category.

If the percentage of two or more attribute is greater then the threshold set by us ,then the video will be stored in all the categories whose percentage is greater then the threshold ,For e.g. if we set threshold to 35% and if the percentage of attribute for a particular video

is 40% comedy ,36% song ,24% politics then the link of video will be saved in comedy and song section of the website.
We would also include more categories like Adult, Abusive, Educational, Spiritual, News.

4.2 Limitations of the proposed strategy

The project might deviate from accurate results if there are irrelevant comments on a video. There are videos on YouTube on which there are no comments or they are very less in either case the project might not give satisfactory result. One way to overcome this is to analyze the Description of video. YouTube has implemented different languages, so if the comment are in any language other than English the project will not give satisfactory result. One way to overcome this is to translate the comments into one standard language(English in our case) and then classifying them though the process will take time as translation is not so quick and translating large amount of data would be cumbersome machine learning on such translation may be employed for selective translation.

REFERENCES

[1] V. Chaudhary and A. Sureka, "Contextual feature based one-class classifier approach for detecting video response spam on youtube," in *Privacy, Security and Trust (PST), 2013 Eleventh Annual International Conference on.* IEEE, 2013, pp. 195–204.

[2] I. Rish, "An empirical study of the naive bayes classifier," in *IJCAI 2001 workshop on empirical methods in artificial intelligence*, vol. 3, no. 22. IBM New York, 2001, pp. 41–46.

[3] A. Schuth, M. Marx, and M. de Rijke, "Extracting the discussion structure in comments on news-articles," in *Proceedings of the 9th annual ACM international workshop on Web information and data management.* ACM, 2007, pp. 97–104.

[4] S. Bird, "Nltk: the natural language toolkit," in *Proceedings of the COLING/ACL on Interactive presentation sessions.* Association for Computational Linguistics, 2006, pp. 69–72.

[5] D. Crockford, "The application/json media type for javascript object notation (json)," 2006.

[6] A. Ammari, V. Dimitrova, and D. Despotakis, "Semantically enriched machine learning approach to filter youtube comments for socially augmented user models," *UMAP*, pp. 71–85, 2011.

[7] A. Sureka, P. Kumaraguru, A. Goyal, and S. Chhabra, "Mining youtube to discover extremist videos, users and hidden communities," in *Information retrieval technology.* Springer, 2010, pp. 13–24.

[8] K. P. Murphy, "Naive bayes classifiers," *University of British Columbia*, 2006.

[9] B. Pang, L. Lee, and S. Vaithyanathan, "Thumbs up?: sentiment classification using machine learning techniques," in *Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10*. Association for Computational Linguistics, 2002, pp. 79–86.

[10] C.-C. Tseng, N. Pateli, H. Paranjape, T. Lin, and S. Teoh, "Classifying twitter data with naive bayes classifier," in *Granular Computing (GrC), 2012 IEEE International Conference on*. IEEE, 2012, pp. 294–299.

[11] F. Sebastiani, "Machine learning in automated text categorization," *ACM computing surveys (CSUR)*, vol. 34, no. 1, pp. 1–47, 2002.

[12] A. Rajadesingan and A. Mahendran, "Comment spam classification in blogs through comment analysis and comment-blog post relationships," in *Computational Linguistics and Intelligent Text Processing*, ser. Lecture Notes in Computer Science, A. Gelbukh, Ed. Springer Berlin Heidelberg, 2012, vol. 7182, pp. 490–501. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-28601-8_41

[13] A. Go, R. Bhayani, and L. Huang, "Twitter sentiment classification using distant supervision," *CS224N Project Report, Stanford*, vol. 1, p. 12, 2009.

[14] D. Lewis, "Naive (bayes) at forty: The independence assumption in information retrieval," in *Machine Learning: ECML-98*, ser. Lecture Notes in Computer Science, C. Ndellec and C. Rouveirol, Eds. Springer Berlin Heidelberg, 1998, vol. 1398, pp. 4–15. [Online]. Available: http://dx.doi.org/10.1007/BFb0026666

[15] Y. Lin, "Support vector machines and the bayes rule in classification," *Data Mining and Knowledge Discovery*, vol. 6, no. 3, pp. 259–275, 2002. [Online]. Available: http://dx.doi.org/10.1023/A%3A1015469627679

APPENDIX

NLTK library is a tool kit which provides a function "Naive baise classifier" in which data sets are fed.
JSON is a light wieghted array of data information , a machine can easily parse it and get the information from the tags assigned to each value. It is a subset of a java script language

Python code is written in back-end to fetch the large comments of a particular YouTube video can be be easily fetched and analysis can be performed on it.

```
import re
import oauth2
from distutils.command.build import build
import time
import urllib2
import json
import os
import sys
from proxy import proxy
import requests
import json, ast
import nltk
import nltk.data
tokenizer = nltk.data.load('nltk:tokenizers/punkt/english.pickle')
import sys
from textblob import TextBlob
from textblob.compat import PY2
from nltk.tokenize import word_tokenize
from nltk import word_tokenize,sent_tokenize
from textblob.classifiers import NaiveBayesClassifier
def func(xyz):
```

```
url= "https://www.googleapis.com/youtube/v3/commentThreads"
params=dict()
api_key='AIzaSyDR_p3sWUI9R21TaewInZOBpKWrS19EfK4'
videoId=xyz#'8Deidy3ONqg'
{params["part"] = "snippet"      #mandatory
{params["maxResults"] = "80"        #optional
{params["textFormat"] = "plainText"  #or html
params["videoId"] = xyz
params["key"] = api_key
url=url+'?'
i=0
for key,value in params.iteritems():
    if i==0:
        url=url+key+'='+value
        i=i+1
        else:
        url=url+'&'+key+'='+value
print (url)
proxies =
"http": "http://ipg_2012045:"+proxy()+"@192.168.1.107:3128",
"https": "https://ipg_2012045:"+proxy()+"@192.168.1.107:3128",
#'https://www.googleapis.com/youtube/v3/commentThreads?key=AIzaSyDR
_p3sWUI9R21TaewInZ
OBpKWrS19EfK4&textFormat=plainText&part=snippet&videoId=kffacxfA7G4
&maxResults=50'
r = requests.get(url, proxies=proxies)
commit_data = (r.json())
dot=". "
stringcocat=""
commit_item=commit_data[ u'items']
```

```
for i in commit_data[u'items']:
     commit_tag=json.dumps(i)
     jdata = json.loads(commit_tag)
     hello=jdata[ u'snippet']
     comma=hello[u'topLevelComment']
     commentdata =json.dumps(comma)
     moondata = json.loads(commentdata)
     textDisplay=moondata[u'snippet']
     commentdfdata =json.dumps(textDisplay)
     moondafaata = json.loads(commentdfdata)
     string=json.dumps(moondafaata[u'textDisplay'])
     string = string[1:-1]
     stringcocat=stringcocat+string.replace(".", "")+dot
withooutslash=stringcocat.replace("\\ud83d","")
     prob_dist0= cl.prob_classify(withooutslash)
     prob_dist.max()
     original_list=[round(prob_dist.prob("politics"),4),round
     (prob_dist.prob("song"),4),round
     (prob_dist.prob("comedy"), 4)]
     list_dump = json.dumps(original_list)
     def json_list(list_dump):
  lst = []
  d = {}
  lst = []
  d['politics']=original_list[0]
  d['song']=original_list[1]
  d['comedy']=original_list[2]
  #lst.append(d)
  loo=json.dumps(d)
  print json.dumps(d)
```

```
  return json.loads(loo)
    from flask import Flask, request, jsonify
    app = Flask(__name__)
    from flask.ext.restful import Api
    from flask.ext.restful import Resource
    api=Api(app)
    train = [
('song', 'song'),
('singer','song'),
('song', 'song'),
("beats ", 'song'),
("soulful ", 'song'),
("singing", 'song'),
("vocal", 'song'),
("lyrics", 'song'),
("lyric", 'song'),
("music", 'song'),
("music director", 'song'),
("chords", 'song'),
("guitar", 'song'),
("soothing ", 'song'),
("nostalgic ", 'song'),
("soothing ", 'song'),
("melodious ", 'song'),
("melody ", 'song'),
("flute ", 'song'),
("rap", 'song'),
("raps", 'song'),
("harmonica ", 'song'),
("harmonium ", 'song'),
```

```
("piano ", 'song'),
("band ", 'song'),
("concert ", 'song'),
("singing", 'song'),
("taylor swift", 'song'),
("arjit", 'song'),
("gana", 'song'),
('democracy politics', 'politics'),
('manmohan', 'politics'),
("Parliament", 'politics'),
("Caste", 'politics'),
("Nehru", 'politics'),
("Politician", 'politics'),
("politics", 'politics'),
("governance", 'politics'),
("Mahatma Gandhi", 'politics'),
("speech", 'politics'),
("democratic", 'politics'),
("Naredra modi", 'politics'),
("APJ Abdul kalam", 'politics'),
("Pratibha Patil", 'politics'),
("Rajendra Prasad", 'politics'),
("Chief Justice", 'politics'),
("rti , RTI", 'politics'),
("sonia gandhi", 'politics'),
("rahul gandhi", 'politics'),
("PM", 'politics'),
("Minister", 'politics'),
("lok sabha", 'politics'),
("rajya sabha", 'politics'),
```

```
("rashtriyapati", 'politics'),
("Pt. Jawaharlal Nehru", 'politics'),
("B. R. Ambedkar", 'politics'),
("Atal Behari Vajpayee", 'politics'),
("Bahadur Shastri", 'politics'),
("Vallabhbhai", 'politics'),
("Chandra Bose", 'politics'),
("namo", 'politics'),
("obama", 'politics'),
("Dadabhai Naoroji", 'politics'),
("criminal", 'politics'),
("Shashi Tharoor", 'politics'),
("speaker public", 'politics'),
("Tharoor", 'politics'),
("gandhi", 'politics'),
("President ", 'politics'),
("conference ", 'politics'),
("AAP", 'politics'),
("aam admi ", 'politics'),
("congress", 'politics'),
("congres", 'politics'),
("rajniti", 'politics'),
("rajneeti", 'politics'),
('haha ', 'comedy'),
(' hahahahah', 'comedy'),
('hahaa hahahaha', 'comedy'),
('hahaa hahahaha', 'comedy'),
(' lollllll','comedy'),
(' loollll','comedy'),
('lol','comedy'),
```

```
('humour', 'comedy'),
('funny ', 'comedy'),
('entertaining', 'comedy'),
('Comedy  ', 'comedy'),
('comedy', 'comedy')
]
     cl = NaiveBayesClassifier(train)
     cl.show_informative_features()
     class hello(Resource):
     def post(self):
  v_id=request.get_data()
  return func(v_id)
     api.add_resource(hello,'/',methods=['POST'])
     if __name__ == "__main__":
  app.debug=True
  app.run(host="0.0.0.0",port=5000)
```