YouTube Spam Comments Classification

PREPARED BY

Manan Pareek (2018BtechCSE110)

FACULTY GUIDE

Dr. Amit Sinhal

Department of Computer Science & Engineering

Institute of Engineering and Technology (IET)

JK Lakshmipat University Jaipur

May 2022

YouTube Spam Comments Classification

Practice School-2

Submitted in partial fulfilment of the requirements

for the degree of

Bachelor of Technology in Computer Science Engineering

By:

Manan Paree (2018BtechCSE110)

Faculty Guide:

Amit Sinhal

Department of Computer Science & Engineering

Institute of Engineering and Technology (IET)

JK Lakshmipat University Jaipur

May 2022

CERTIFICATE

This is to certify that the Practice School-2 project work entitled "**Youtube Spam Comment Classification**" submitted by **Manan Pareek (2018BtechCSE110)**, towards the partial fulfillment of the requirements for the degree of **Bachelor of Technology in Computer Science Engineering** of JK Lakshmipat University Jaipur is the record of work carried out by them under my supervision and guidance. In my opinion, the submitted work has reached a level required for being accepted for Practice School-II examination.

Dr. Amit Sinhal
Professor,

Department of Computer Science Engineering
Institute of Engineering & Technology (IET)
JK Lakshmipat University, Jaipur

Dr. Gireesh Kumar
Assistant Professor,

Department of Computer Science Engineering
Institute of Engineering & Technology (IET)
JK Lakshmipat University, Jaipur

Date of Submission: 27th May 2022

Prof. Dr. – Sanjay Goel
Director
Institute of Engineering & Technology
JK Lakshmipat University, Jaipur

ACKNOWLEDGEMENTS

The internship opportunity I had with Futurense Technologies Pvt. Ltd. was a great chance for learning and professional development. Therefore, I consider myself fortunate individual as I was provided with an opportunity to be part of Futurense. I am also grateful for having a chance to meet so many wonderful people and professionals who led me through this internship period

I would like to express our heartfelt gratitude to **Prof Dr. Sanjay Goel**, Director, Institute of Engineering for providing this opportunity to me. I would also like to thank **Dr Amit Sinhal, Dr. Sonal Jain,** for their constant support and guidance.

I am highly indebted to Futurense Technologies Pvt Ltd. for their guidance and constant supervision as well as for providing necessary information regarding the project & also for their support in completing the project**.**

Sincerely yours,

Manan Pareek

(2018BtechCSE110)

ABSTRACT

YouTube is considered as one of the most popular video sharing websites that is growing very fast. Because of its popularity, it attracts spammers to distribute spam through comments on YouTube. This has become concern because spam can lead to phishing attack which the target can be user that click on any malicious link. YouTube is running its own spam blocking system but continues to fail to block them properly. Therefore, we examined related studies on YouTube spam comment screening and conducted classification experiments with three different machine learning techniques (Random Forest, SVM, Gradient Boosting). We created own dataset for training and testing purpose.

LIST OF FIGURES

CONTENTS

CHAPTER 1 : INTRODUCTION

1.1 About Futurense Technologies:

Futurense Technologies Pvt. Ltd. is a global IT consulting and software development firm; our core focus is towards making digital transformation the norm. We help our clients make the most of digital technologies to innovate for sustainable growth.

Started in 2020 with a vision to put India at the forefront of the technology revolution.

We aim to do this with a two-pronged approach:

- Build futuristic tech solutions
- Nurture future-ready careers

Figure 1.1: Futurense Technologies

1.2 About Training:

The training is divided in two phases

1st Phase:- 2 Months Online Training where they will train us on Python and SQL.

2nd Phase:- 3 Months Offline Training where they will be giving us training on Big Data Technologies like Hadoop, spark, scala, hive, flume, Clouds like AWS, Azure and Databases like Cassandra, Toad etc.

My training started from 25th November 2021,

From 25th Nov – 21st Dec 2021 they trained us on Python, from beginner to intermediate level, first they taught us basics of python, then taught us about strings, tuples, dictionaries and then they taught us some higher concepts like classes & objects, and OOPS concept. On these topics they gave us assignments on almost day to day basis.

For Python Coding tools used were: Spyder, Jupyter Notebook and PyCharm.

From 23rd Dec 2021 onwards they started teaching us RDBMS, in starting they taught us about the basics of RDBMS like Schema, Entity, attributes and relations.

Then they took a E-commerce database and start building it from scratch. For first 2 days we created around 20-23 tables for our database. After creating those tables we inserted the data in them.

After this they taught us about different operation which we can do on data to extract meaningful information. We got to know how triggers and procedures work in SQL and how views can help us in making a complex query simple.

They also taught us how to grant and revoke permission in database for different users.

1.3 Project at Futurense :

1.3.1 Problem Statement:

YouTube uses few key factors like measuring user interactions; number of views, shares, comments, and likes etc. to rank the videos as the top trending on the platform. The top trending list of videos mostly belongs to movies, music, celebrities, reality shows, TV shows, etc. This dataset includes a few months of daily trending YouTube videos.
The dataset includes a category_id field and its master is also given.

1.3.2 Objective:

Use SQL features for data analysis and showing the valuable insights.

- **Domain:** YouTube video analytics
- **Analysis to be done:** Exploratory analysis to determine actionable insights

About Data: Dataset consists of several months (and counting) of records on each day trending YouTube movies. data is covered for the us, GB, DE, CA, and FR areas (u.s.a., super Britain, Germany, Canada, and France, respectively), with as much as two hundred listed trending videos according to day.
We took the US region data for our Project.

Tools used for Accomplishing the Project: Oracle Toad
TOAD(Tool for Oracle Application Developers)
Toad is a database management toolset from Quest Software for managing relational and non-relational databases.
TOAD uses MySQL.

Some Snapshots of Project:

Created the Schema for table

```
create table tbl_you_tube(video_id varchar2(200),trending_date varchar2(200),title varchar2(200),channel_title varchar2(200),category_id number,
publish_time varchar(100),  tags varchar(400), views number,likes number,dislikes number,comment_count number,comments_disabled varchar2(10),
ratings_disabled varchar2(10));
```

Then we inserted the data from CSV file to table we created, using TOAD. The table include 40500 rows.

- **Top 3 videos for which user interaction (views + likes + dislikes + comments) is the highest.**

```
select video_id, title, category_id, (likes+dislikes+comment_count+views) as User_Interaction
from tbl_you_tube order by user_interaction desc  ;
```

VIDEO_ID	TITLE	CATEGORY_ID	USER_INTERACTION
n1WpP7iowLc	Eminem - Walk On Water (Audio) ft. Beyoncé	10	18115252
5qpjK5DgCt4	Racist Superman | Rudy Mancuso, King Bach & Lele Pons	23	3350987
1ZAPwfrtAFY	The Trump Presidency: Last Week Tonight with John Oliver (HBO)	24	2534817

- **Bottom 3 videos for which user interaction (views + likes + dislikes + comments) is lowest.**

```
select video_id, title, category_id, (likes+dislikes+comment_count+views) as User_Interaction
from tbl_you_tube order by user_interaction asc  ;
```

VIDEO_ID	TITLE	CATEGORY_ID	USER_INTERACTION
NZFhMSgbKKM	Dennis Smith Jr. and LeBron James go back and forth	17	965
zNqCVTs38nU	Heidelberg's nifty hook-and-lateral to the left tackle	17	4641
c-3JxzN2u34	Lin-Manuel Miranda's next act: Helping rebuild Puerto Rico	25	6699

- **Top 3 videos of each category in each year**
  - By number of views
  - By number of comments
  - By number of likes
  - Highest user interaction

```
SELECT * FROM
(
    SELECT VIDEO_ID,title,  Category_ID, comment_count,
    ROW_NUMBER() OVER (PARTITION BY Category_ID Order by comment_count DESC) AS Sno#
    FROM tbl_you_tube
)RNK
WHERE Sno# <=3
```

VIDEO_ID	TITLE	CATEGORY_ID	COMMENT_COUNT	SNO#
Om_zGhJLZ5U	TL;DW - Every DCEU Movie Before Justice League	1	2111	1
jr9QtXwC9vc	The Greatest Showman | Official Trailer 2 [HD] | 20th Century FOX	1	340	2
n30k5CwLhS4	Nick Andopolis: Drummer	1	246	3
U0hAC8O7RoI	I TOOK THE $3,000,000 LAMBO TO CARMAX! They offered me......	2	486	1
n1WpP7iowLc	Eminem - Walk On Water (Audio) ft. Beyoncé	10	125882	1
5E4ZBSInqUU	Marshmello - Blocks (Official Music Video)	10	8371	2
0tO_l_Ed5Rs	Matthew Santoro - FACTS (Official Music Video) f. Ellevan & Humble the Poet	10	7484	3
TaTleo4cOs8	SHOPPING FOR NEW FISH!!!	15	2120	1
sbcbvuitiTc	Stephon Marbury and Jimmer Fredette fight in China	17	1447	1
9wRQljFNDW8	Dion Lewis' 103-Yd Kick Return TD vs. Denver! | Can't-Miss Play | NFL Wk 10 Highlights	17	177	2
zNqCVTs38nU	Heidelberg's nifty hook-and-lateral to the left tackle	17	19	3
p2hJxyF7mok	New Emirates First Class Suite | Boeing 777 | Emirates	19	236	1
2kyS6SvSYSE	WE WANT TO TALK ABOUT OUR MARRIAGE	22	15954	1
STI2fI7sKMo	AFFAIRS, EX BOYFRIENDS, $18MILLION NET WORTH - GOOGLE US | Shawn and And...	22	895	2
0mlNzVSJrT0	Me-O Cats Commercial	22	532	3
5qpjK5DgCt4	Racist Superman | Rudy Mancuso, King Bach & Lele Pons	23	8181	1
dQvIbulWCM4	Celebrities on Thanksgiving 2017!	23	3412	2
ZAQs-ctOqXQ	One Change That Would Make Pacific Rim a Classic	23	1256	3

```sql
SELECT
    video_id,title,category_id,(views + likes + dislikes + comment_count) as USER_INTERACTION
FROM    tbl_you_tube ;

SELECT * FROM
(
SELECT VIDEO_ID, EXTRACT (YEAR FROM to_date(trending_date , 'YY.DD.MM')) as YEAR, title, Category_ID,(views + likes + dislikes + comment_count),
    ROW_NUMBER() OVER (PARTITION BY Category_ID Order by (views + likes + dislikes + comment_count) DESC) AS Sno#
    FROM tbl_you_tube
)RNK WHERE Sno# <=3
```

VIDEO_ID	YEAR	TITLE	CATEGORY_ID	(VIEWS+LIKES+DISLIKES+COMMENT_COUNT)	SNO#
jr9QtXwC9vc	2017	The Greatest Showman | Official Trailer 2 [HD] | 20th Century FOX	1	830061	1
Om_zGhJLZ5U	2017	TL;DW - Every DCEU Movie Before Justice League	1	299340	2
n30k5CwLhS4	2017	Nick Andopolis: Drummer	1	52066	3
U0hAC8O7RoI	2017	I TOOK THE $3,000,000 LAMBO TO CARMAX! They offered me......	2	103394	1
n1WpP7iowLc	2017	Eminem - Walk On Water (Audio) ft. Beyoncé	10	18115252	1
5E4ZBSInqUU	2017	Marshmello - Blocks (Official Music Video)	10	811474	2
0tO_l_Ed5Rs	2017	Matthew Santoro - FACTS (Official Music Video) f. Ellevan & Humble the Poet	10	366448	3
TaTleo4cOs8	2017	SHOPPING FOR NEW FISH!!!	15	217371	1
sbcbvuitiTc	2017	Stephon Marbury and Jimmer Fredette fight in China	17	962058	1
9wRQljFNDW8	2017	Dion Lewis' 103-Yd Kick Return TD vs. Denver! | Can't-Miss Play | NFL Wk 10 Highlights	17	82234	2
zNqCVTs38nU	2017	Heidelberg's nifty hook-and-lateral to the left tackle	17	4641	3
p2hJxyF7mok	2017	New Emirates First Class Suite | Boeing 777 | Emirates	19	143115	1
2kyS6SvSYSE	2017	WE WANT TO TALK ABOUT OUR MARRIAGE	22	824821	1
STI2fI7sKMo	2017	AFFAIRS, EX BOYFRIENDS, $18MILLION NET WORTH - GOOGLE US | Shawn and And...	22	328171	2
0mlNzVSJrT0	2017	Me-O Cats Commercial	22	102168	3
5qpjK5DgCt4	2017	Racist Superman | Rudy Mancuso, King Bach & Lele Pons	23	3350987	1
dQvIbulWCM4	2017	Celebrities on Thanksgiving 2017!	23	649128	2

```sql
(
SELECT VIDEO_ID, title, Category_ID,LIKES,
    ROW_NUMBER() OVER (PARTITION BY Category_ID Order by LIKES DESC) AS Sno#
    FROM tbl_you_tube
)RNK WHERE Sno# <=3
```

VIDEO_ID	TITLE	CATEGORY_ID	LIKES	SNO#
Om_zGhJLZ5U	TL;DW - Every DCEU Movie Before Justice League	1	7515	1
jr9QtXwC9vc	The Greatest Showman | Official Trailer 2 [HD] | 20th Century FOX	1	3543	2
n30k5CwLhS4	Nick Andopolis: Drummer	1	715	3
U0hAC8O7RoI	I TOOK THE $3,000,000 LAMBO TO CARMAX! They offered me......	2	4035	1
n1WpP7iowLc	Eminem - Walk On Water (Audio) ft. Beyoncé	10	787419	1
5E4ZBSInqUU	Marshmello - Blocks (Official Music Video)	10	114188	2
0tO_I_Ed5Rs	Matthew Santoro - FACTS (Official Music Video) f. Ellevan & Humble the Poet	10	15186	3
TaTleo4cOs8	SHOPPING FOR NEW FISH!!!	15	7473	1
sbcbvuitiTc	Stephon Marbury and Jimmer Fredette fight in China	17	2017	1
9wRQljFNDW8	Dion Lewis' 103-Yd Kick Return TD vs. Denver! | Can't-Miss Play | NFL Wk 10 Highlights	17	655	2
zNqCVTs38nU	Heidelberg's nifty hook-and-lateral to the left tackle	17	35	3
p2hJxyF7mok	New Emirates First Class Suite | Boeing 777 | Emirates	19	1661	1
2kyS6SvSYSE	WE WANT TO TALK ABOUT OUR MARRIAGE	22	57527	1
STI2fI7sKMo	AFFAIRS, EX BOYFRIENDS, $18MILLION NET WORTH - GOOGLE US | Shawn and Andrew	22	4451	2
0mlNzVSJrT0	Me-O Cats Commercial	22	2486	3
5qpjK5DgCt4	Racist Superman | Rudy Mancuso, King Bach & Lele Pons	23	146033	1
dQvIbulWCM4	Celebrities on Thanksgiving 2017!	23	38397	2
ZAQs-ctOqXQ	One Change That Would Make Pacific Rim a Classic	23	8011	3

Channel-Wise total Views

```
SELECT  channel_title, sum(views) as total_views from tbl_you_tube group by channel_title order by total_views desc ;
```

CHANNEL_TITLE	TOTAL_VIEWS
EminemVEVO	17158531
Rudy Mancuso	3191434
LastWeekTonight	2418783
Saturday Night Live	2103417
nigahiga	2095731
FBE	2045386
Simply Nailogical	1842393
dope2111	1456472
NBA Highlights · ...	956169
20th Century Fox	826059
CrazyRussianHac...	817732
CaseyNeistat	748374
marshmello	687582
Niki and Gabi	605932
NowThis World	544770
Good Mythical M...	343168
MatthewSantoro	328330
Shawn Johnson ...	321053

CHAPTER 2 : YOUTUBE SPAM COMMENT CLASSIFICATION

YouTube is a veritably successful videotape participating company. It has further than one billion druggies and that nearly one third of the druggies of the whole internet. They watch a billion hours of YouTube vids and induce billions of views daily. At present, further than 400 hours of videotape are uploaded, and4.5 million vids are watched every nanosecond on YouTube. It's easy for druggies to watch and upload vids without any restrictions.

YouTube generators can monetize if they've further than subscribers and hours of watch time for the last 12 months. Consequently, spam commentary are being created to promote their channels or vids. Some generators closed the comment function due to aggression similar as political commentary, vituperative speech or depreciatory commentary not related to their vids. Spam causes numerous problems including wasting the stoner’s time, memory and use up network bandwidth. Organizations and druggies could face fiscal loss due to trouble of spam.

YouTube has its own spam filtering system, however there are still spam commentary that aren't being caught.

Types of Spam on YouTube

• Link Based Spam- It's a veritably common form of spam frequently seen on YouTube. Commentary contain Hypertext(HTTP) links to other websites, generally other vids on YouTube itself. Numerous links deflect the stoner to potentially vicious webpages frequently without the knowledge of the stoner.

• Channel Promotional Spam- It's the most current form of spam on YouTube. These types of commentary generally correspond of druggies who essay to promote their own channel by requesting for subscribers, posting links to their vidsetc.

Downsides of Current System-

• YouTube has tried to combat link- grounded spam by blocking all commentary containing Hypertext links(HTTP). Although effective, this form of filtering has had it’s share of problems as it has led to spammers resorting to further creative ways similar as fitting whitespace characters between links to avoid discovery.

• For Illustration- **This song is really amazing and i go JB still busy using authentic views fleck c0m and reaching millions of views.**

Similar types of commentary are innately designed to bypass You tube's link discovery sludge. Utmost link pollutants and blacklists prove ineffective against similar blurred commentary.

Lately as of November 2017, YouTube has faced adding review about its incapability to moderate uploaded content(1). A large stoner- base of YouTube consists of children who are frequently exposed to vicious and dangerous material in the form of commentarySuch types of comments are inherently designed to bypass You tube's link detection filter. Most link filters and blacklists prove ineffective against such obfuscated comments.

Recently as of November 2017, YouTube has faced increasing criticism about its inability to moderate uploaded content [1]. A large user-base of YouTube consists of children who are often exposed to malicious and harmful material in the form of comments.

2.1 Related Work

- Much research has been done in the area of spam comments detection. Alberto et al. (2015) showed that many classification techniques can be used to find the spam comments in YouTube. These techniques include decision trees, logistic regression, random forests, linear and Gaussian support vector machines. They used a dataset collected from YouTube which has 1956 real user comments that are related to five most viewed YouTube Videos.
- Alex Kantchelian et al. [2] developed a spam detection technique which can quantify fruitless and superfluous features in blogs, making meaningful stories more accessible to the perpetual stakeholders. They suggested extension of their work to broaden the definition of spam such as URLs, short message removal, etc.
- Enhua Tan et al. [3] designed a runtime spam detection scheme called as BARS: Blacklist-Assisted Runtime Spam Detection which constructed a database of Spam URLs against which URL of every new post was analyzed to determine if the post was spam or not.

- M. McCord et al. [4] harnessed machine learning algorithms which were trained with content and user-centered facets to identify spammers. They tested their algorithms with twitter data and discovered that the Random Forest Classifier offered the best results.

- Igor Santos et al. [5] applied the concept of anomaly detection wherein the deviation from authentic emails was used as a metric to classify emails as spam or ham. Better accuracy was achieved owing to the limited training sets as seen in labelling-based systems.

- The authors, Qingxi Peng et al. [6] employed the concepts of sentiment analysis to detect spam reviews. They concluded by proposing some improvements in the calculation of sentiment score which was the base of their paper.

- Ammari et al. (2011) worked on creating user models or profiles that are enriched by the characteristics of the users which are obtained from social websites.

2.2 Research Objectives

OBJECTIVE 1: To extract comments from different YouTube Channels using Youtube APIs.
OBJECTIVE 2: To classify extracted data into Spam or Ham.
OBJECTIVE 3: To calculate the accuracy of extracted data and compare it with other's accuracy.

2.3 Methodology Adopted

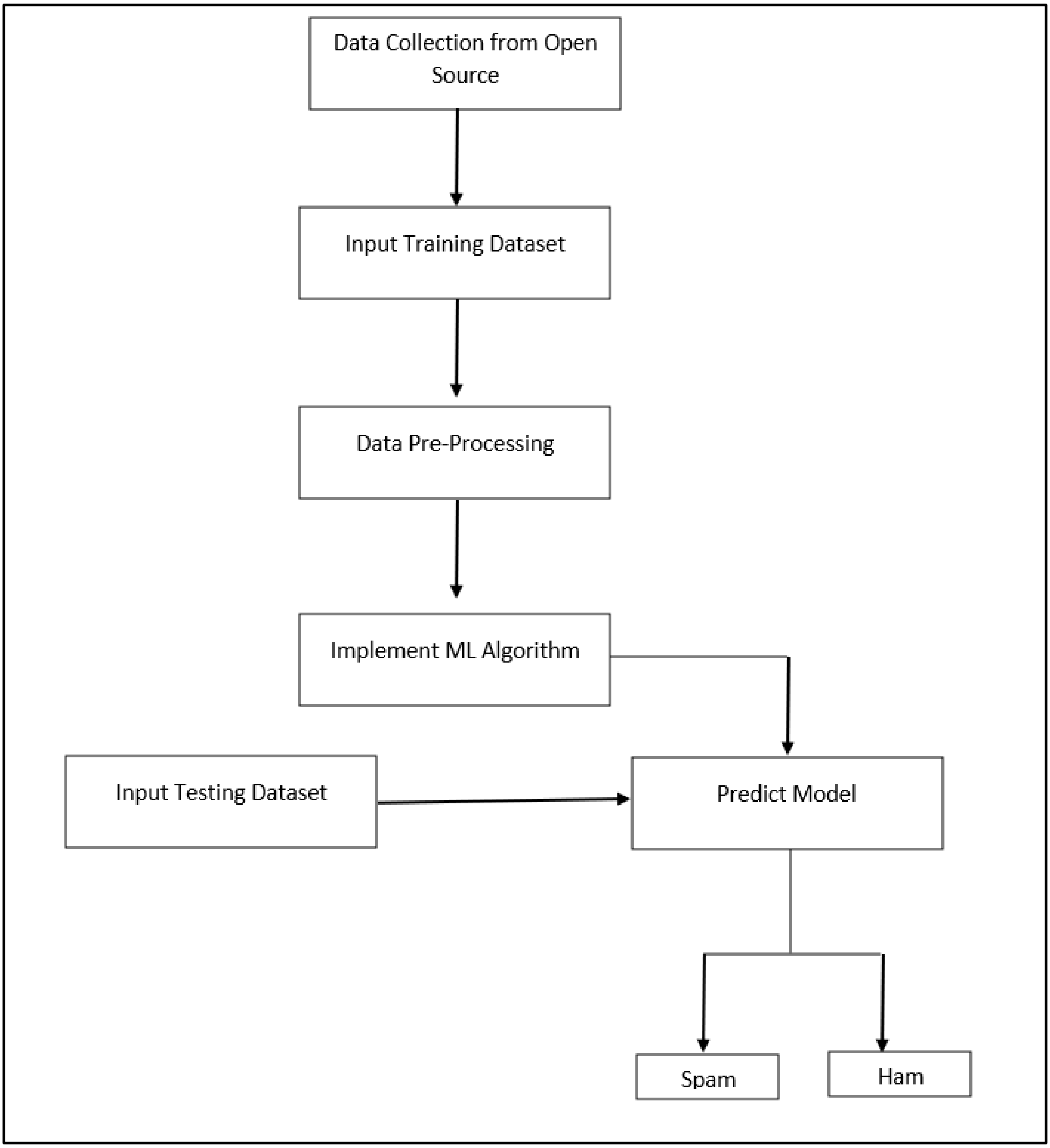

Figure 2.1: Methodology

2.4 About Dataset:

The used datasets are extracted from YouTube using YouTube Data API . The extracted datasets are based on two main characteristics: the popularity of the channel and the availability of up-to-date comments. There was no other consideration apart from these two characteristics. Therefore, the datasets were randomly selected (not based on celebrities or whatsoever). The total number of used YouTube channels is 4-5 and the overall samples are 14,000. Nevertheless, not all channels have presented the exact same sample share.

All datasets are combined into one file which composites the followings:

- video_id
- video_title
- author
- comment_text
- class

The “Classification” field is presenting the essence of the comment “spam” or “ham”

2.5 Algorithms Used:

- **Random Forest Classification:**

Random forest, like its name implies, consists of a large number of individual decision trees that operate as an ensemble. Each individual tree in the random forest spits out a class prediction and the class with the most votes becomes our model’s prediction. The low correlation between models is the key. Just like how investments with low correlations (like stocks and bonds) come together to form a portfolio that is greater than the sum of its parts, uncorrelated models can produce ensemble predictions that are more accurate than any of the individual predictions. The reason for this wonderful effect is that the trees protect each other from their individual errors (as long as they don’t constantly all err in the same direction). While some trees may be wrong, many other trees will be right, so as a group the trees are able to move in the correct direction.

- **Gradient Boosting Algorithm:**

The main idea behind this algorithm is to build models sequentially and these subsequent models try to reduce the errors of the previous model. When the target column is continuous, we use Gradient Boosting Regressor whereas when it is a classification problem, we use Gradient Boosting Classifier. The only difference between the two is the *"Loss function"*. The objective here is to minimize this loss function by adding weak learners using gradient descent. Since it is based on loss function hence for regression problems, we'll have different loss functions like Mean squared error (MSE) and for classification, we will have different for e.g log-likelihood.

2.6: Features Selection and Extraction

Features Selection and Extraction is the third phase for this detection framework. The process in this phase are shown as below in Figure 2.

There are Data that has been cleaned, identify the suitable features based on the YouTube comment, split the features into three set in order to identify the best features set and finally is test the features by using classification techniques such as Random Forest Classification and Gradient Boosting Classification.

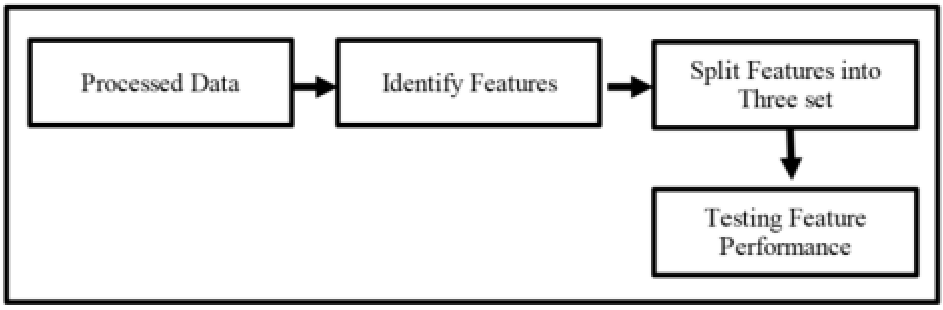

Figure 2.2 Feature Selection and Extraction

For this project, three experiments had been conducted to identify most suitable features to be used in this research. Therefore, the selected features are as follows:

- **Presence of Links:** The presence of links or URL is commonly detected as a spam message or comments. This feature we noted in boolean expression where a value of 1 as presence and value of 0 as an absence.
- **Length of Comments:** The length of comments on this research is calculated after pre-processing. The value of this feature is numerical.
- **Spam Keyword:** This feature also denoted as Boolean expression where if there are the spam keyword in the comment, the value 1 will be denoted while value 0 if there are no spam keyword.

CHAPTER 3 : OBSERVATIONS & COMPARITVE STUDY

Below graphs shows the comparison between the accuracy of different algorithms implemented by different authors.

As we compare our algoirthm accuracy with others dataset we find out the we have received **95.1% Accuracy with Random Forest Classification** and **92.1% with Gradient Boosting Classification.**

We have checked this accuracy on our own dataset which we have fetched from YouTube using YouTube API.

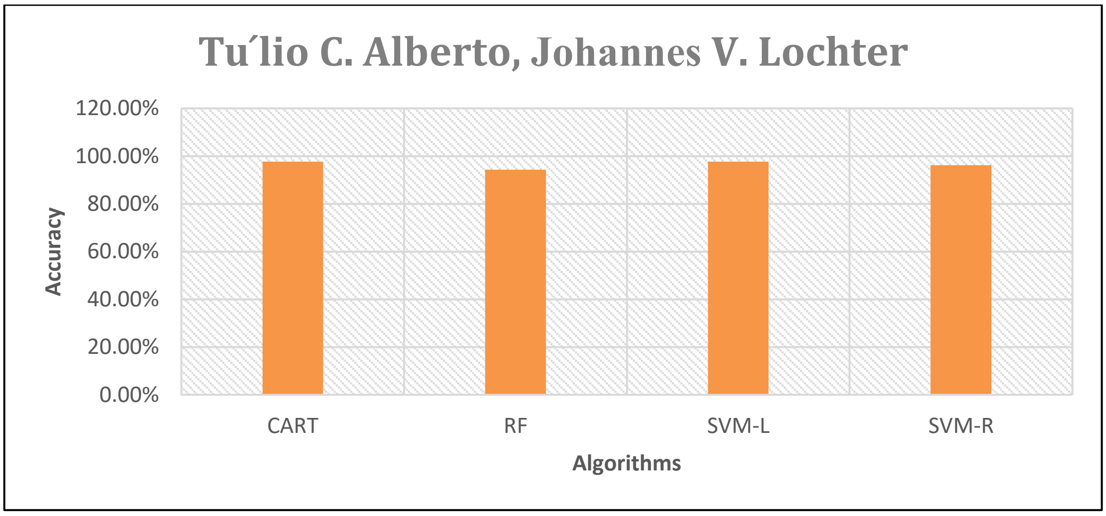

Figure 3.1 Alberto TubeSpam Classification

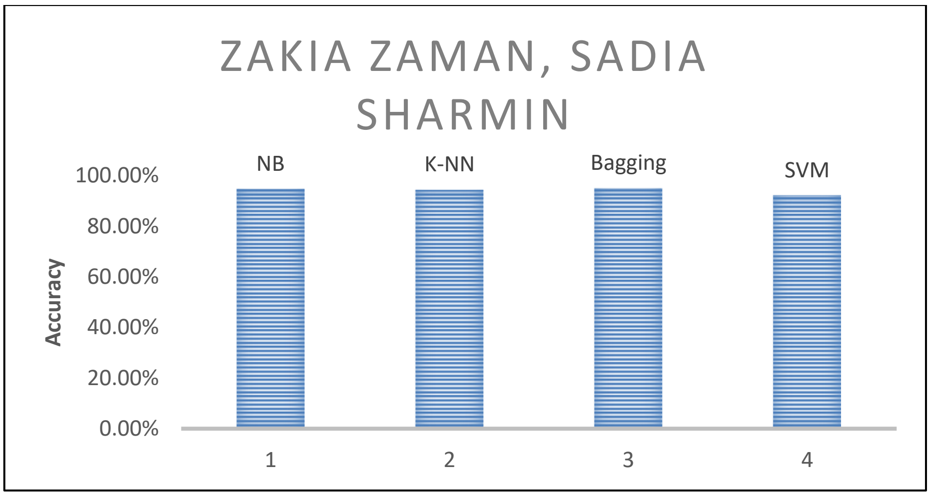

Figure 3.2 Zakai and Sadia's Algorithms Accuaracy

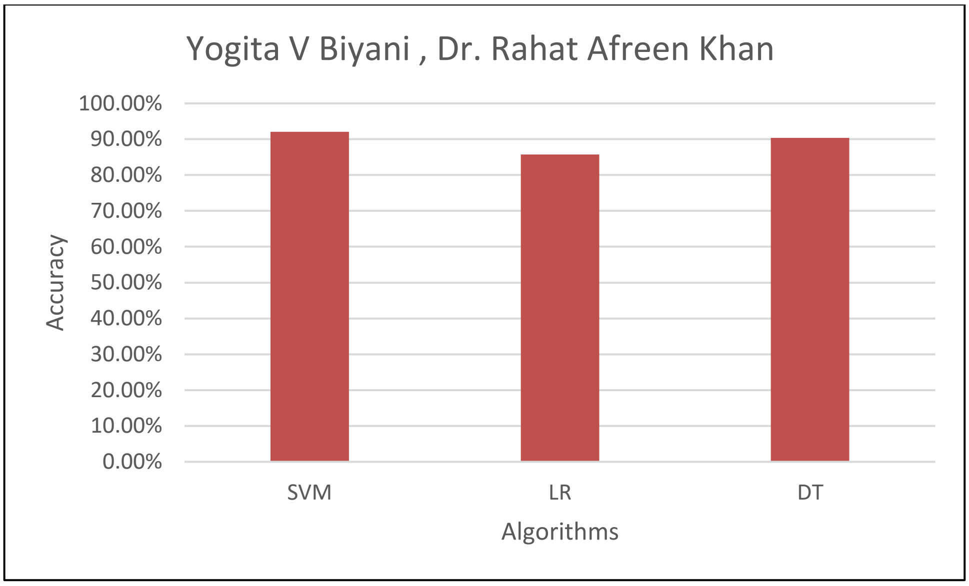

Figure 3.3 Yogita and Dr. Rahat's Algorithm's Accuracy

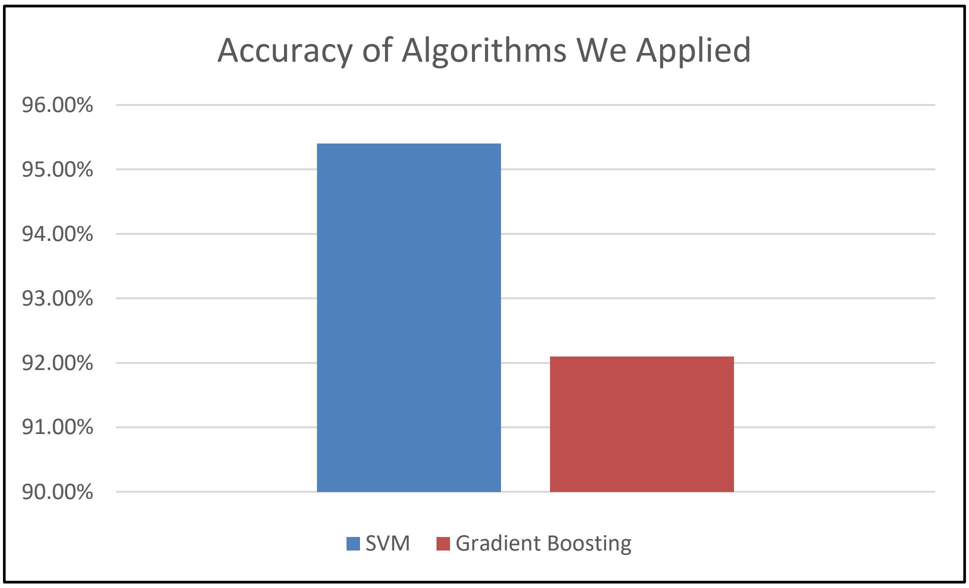

Figure 3.4 Accuracy we got by Applying these Algoritms on our own dataset

CHAPTER 4 : RESULTS

The major goal that we wanted to achieve via this project is being able to detect spam comments of any kind that may harm the users. The experimental purpose of the machine-learning model include training in such a way that it can successfully identify whether a comment is a spam or not. It can detect link-based spam, channel promotional spam and also covers whitespaces characters between links problem which is a major drawback of the existing solution.

The approach used in the solution is to first collect data and then defining training dataset and at same time pre-processing the data and then implementing the Machine learning Algorithms and using testing dataset to predict the model. In this project the machine learning techniques used are Random Forest, Support Vector Machine (SVM) and Gradient Boosting.

Initially, the research question was: “How we can classify between a spam comment and a normal comment?”.

After using training & testing datasets and applying ML Algorithm the prediction model will be able to predict Spam which means comments considered harmful for the YouTube users and Ham are the comments which is unharmful for the users.

With Random Forest classification we get **95.4%** Accuracy and by using Gradient Boosting Classification we got **92.1%** Accuracy.

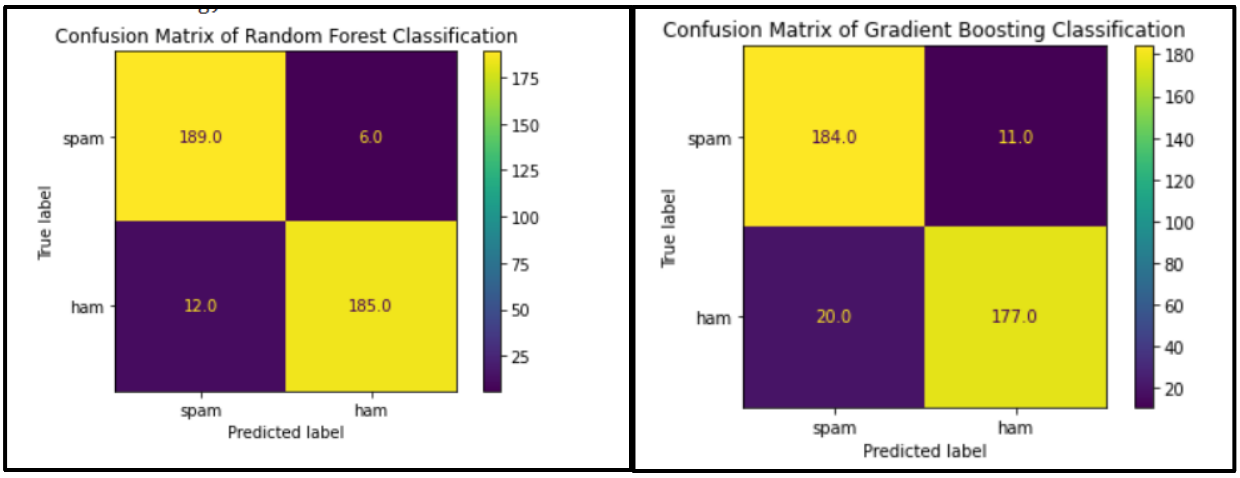

Figure 4.1 Confusion Matrix for Random Forest Classification & Gradient Boosting Classification

CHAPTER 5 : CONCLUSION

The research topic was to analyze and declare whether a comment on a YouTube video is a spam or not. With the increase in phishing attacks using YouTube comment, users are unsafe since they might end up losing important data that really matters or might get scammed. This research topic was important because there is still a potential in improvement in the filtering of comments that are harmful for the users.

In conclusion the main motive of this research project is to make YouTube comments a healthy place for discussions, for users to carry out conversation which leads to a better interacting place which might end up helping someone in need. Also, this project focusses on stopping majority of those phishing attacks.

REFERENCES

1. Youtube, Youtube Press Statistics, https://www.youtube.com/yt/about/press/
2. A. Kantchelian, J. Ma, L. Huang, S. Afroz, A. Joseph, J. D. Tygar, Robust detection of comment spam using entropy rate, in: Proceedings of the 5th ACM Workshop on Security and Artificial Intelligence, AISec ’12, ACM, New York, NY, USA, 2012, pp. 59–70. doi:10.1145/2381896.2381907
3. E. Tan, L. Guo, S. Chen, X. Zhang, Y. Zhao, Spammer behavior analysis and detection in user generated content on social networks, in: 2012 IEEE 32nd International Conference on Distributed Computing Systems, 2012, pp. 305–314. doi:10.1109/ICDCS.2012.40
4. M. McCord, M. Chuah, Spam Detection on Twitter Using Traditional Classifiers, Springer Berlin Heidelberg, Berlin, Heidelberg, 2011, pp. 175–186. doi:10.1007/978-3-642-23496-5 13.
5. I. Santos, C. Laorden, X. Ugarte-Pedrero, B. Sanz, P. G. Bringas, Spam Filtering through Anomaly Detection, Springer Berlin Heidelberg, Berlin, Heidelberg, 2012, pp. 203–216. doi:10.1007/978-3-642-35755-8 15.
6. Q. Peng, M. Zhong, Detecting spam review through sentiment analysis 9
7. G. Mishne and N. Glance, “Leave a reply: An analysis of weblog comments,” in Proc. of 3rd WWE, Edinburgh, UK, 2006
8. A. Sureka, “Mining user comment activity for detecting forum spammers in YouTube,” arXiv preprint arXiv:1103.5044, 2011
9. C. Smitashree and J. G. Breslin, “User Sentiment Detection: a YouTube Use Case,” The 21st National Conference on Artificial Intelligence and Cognitive Science, 2010.
10. W. M. Westenberg, “The influence of Youtubers on teenagers: a descriptive research about the role Youtubers play in the life of their teenage viewers,” Master Thesis. University of Twente, 2016.
11. Jain and D. M. Patel. (2019). Analyzing User Comments of Learning Videos From Youtube Using Machine Learning. [Online].
12. A. Kantchelian, J. Ma, and A. D. Joseph, “Robust Detection of Comment Spam Using Entropy Rate Categories and Subject Descriptors,” no. AISec, pp. 59–69, 2012.
13. S. Aiyar and N. P. Shetty, “N-Gram Assisted Youtube Spam Comment Detection,” Procedia Comput. Sci., vol. 132, no. Iccids, pp. 174–182, 2018, doi: 10.1016/j.procs.2018.05.181

APPENDIX

Data Extraction

```python
import json
import logging
import googleapiclient.discovery

logger = logging.getLogger(__name__)
FORMAT = "[%(filename)s:%(lineno)s:%(funcName)s()] %(message)s"
logging.basicConfig(format=FORMAT)
logger.setLevel(logging.DEBUG)


def build_youtube(API_KEY):
    api_service_name = "youtube"
    api_version = "v3"
    youtube = googleapiclient.discovery.build(
        api_service_name, api_version, developerKey=API_KEY)
    return youtube


def search_videos(youtube, channel_id, page_token):
    request = youtube.search().list(
        part="snippet",
        channelId=channel_id,
        type="video",
        pageToken=page_token
    )
    response = request.execute()
    logger.debug("Received Video Search Response")
    return response


def search_comments(youtube, video_id, page_token):
    request = youtube.commentThreads().list(
        part="snippet",
        videoId=video_id,
        pageToken=page_token
    )
    logger.debug(f"Received comment Search Response for {video_id} ")

    response = request.execute()
    return response


def process_search_response(response):
    next_page_token = response.get('nextPageToken')
    result = []
    for i, item in enumerate(response["items"]):
        video_id = item["id"]["videoId"]
        video_title = item['snippet']['title']
        result.append({
            'video_id': video_id,
            'video_title': video_title
        })
        logger.debug(f"Received comment Search Response for {video_id} ")

    return next_page_token, result
```

```python
def process_comments_response(response, video):
    next_page_token = response.get('nextPageToken')
    result = []
    for i, item in enumerate(response["items"]):
        comment = item["snippet"]["topLevelComment"]
        author = comment["snippet"]["authorDisplayName"]  # Use Later
        comment_text = comment["snippet"]["textDisplay"]
        video_id = video['video_id']
        video_title = video['video_title']
        result.append(
            {
                'video_id': video_id,
                'video_title': video_title,
                'author': author,
                'comment_text': comment_text
            }
        )
        logger.debug(f"Comment: {comment_text[:50]}... for {video_title}")

    return next_page_token, result

def main():
    api_key = "AIzaSyDtpQgR38OZke5KGp788mGNff8NvhPD_8A"
    channel_id = "UCZSNzBgFub_WWil6TOTYwAg"

    youtube = build_youtube(api_key)

    videos = []
    comments = []
    try:
        next_page = None
        while True:
            response = search_videos(youtube, channel_id, next_page)
            next_page, result = process_search_response(response)
            videos += result
            if not next_page:
                break

        for video in videos:
            next_page = None
            while True:
                response = search_comments(youtube, video['video_id'], next_page)
                next_page, result = process_comments_response(response, video)
                comments += result
                if not next_page:
                    break
    except Exception as e:
        logger.error(f"Error:\n{str(e)}")
    print(f"Total comments: {len(comments)}")
    print(f"Total videos: {len(videos)}")

    with open('all_comments.json', 'w', encoding='utf-8') as f:
        json.dump(comments, f, indent=4)

if __name__ == "__main__":
    main()
```

```python
import json
import csv

with open('all_comments.json') as json_file:
    jsondata = json.load(json_file)

data_file = open('new.csv', 'w', newline='')
csv_writer = csv.writer(data_file)

count = 0
for data in jsondata:
    if count == 0:
        header = data.keys()
        csv_writer.writerow(header)
        count += 1
    csv_writer.writerow(data.values())

data_file.close()
```

Classification of Comments as SPAM or HAM

```
import tensorflow as tf
import tensorflow_hub as hub
import pandas as pd
import seaborn as sns
import numpy as np
%matplotlib inline
from sklearn.model_selection import train_test_split
```

```
csv = pd.read_csv(
    "/content/data_yt.csv"
)

# DataFrame with only "Comment" and "Spam/Ham" column
df = csv[["comment_text", "class"]]

# Remove all comments that have no ratings
df = df.dropna()


df
```

```

from tensorflow.keras.preprocessing.sequence import pad_sequences
from tensorflow.keras.preprocessing.text import Tokenizer


from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense
from tensorflow.keras.layers import Flatten
from tensorflow.keras.layers import Dropout
from tensorflow.keras.layers import Embedding
from tensorflow.keras.callbacks import EarlyStopping


from sklearn.model_selection import train_test_split
```

```
model = "https://tfhub.dev/google/nnlm-en-dim50/2"
hub_layer = hub.KerasLayer(model, input_shape=[], dtype=tf.string, trainable=True)

model = tf.keras.Sequential()
model.add(hub_layer)
model.add(tf.keras.layers.Dense(16, activation='relu'))
model.add(tf.keras.layers.Dense(1, activation='sigmoid'))

print(model.summary())
```

```
early_stop = EarlyStopping(monitor='val_loss', mode='min', verbose=1, patience=10)

model.compile(optimizer='rmsprop', loss='binary_crossentropy', metrics=['accuracy'])

# fit the model
model.fit(X_train,
          y_train,
          epochs=40,
          batch_size=512,
          validation_data=(X_test, y_test),
          verbose=1,
          callbacks=[early_stop]
)
```

```
sms = [
       "So helpful! Thank you!",
"Excellent video. Well explained",
"Amazing!!!",
"unable to scrape myntra product details using scrapy?",
"Precise and no bullshit, well explained. Thank you!"
]
```

```
for test in sms:
    out = model.predict([test])
    print("%s - %f - %s" % ("Spam" if out[0] > 0.5 else "HAM", out[0], test))
    print()
```

```
list_comments = copy_comment['comment_text'].to_list()
```

```
for test in list_comments:
    out = model.predict([test])
    print("%s - %f - %s" % ("Spam" if out[0] > 0.5 else "HAM", out[0], test))
    print()
```

```
for test in list_comments:
    out = model.predict([test])
    print("%s - %f - %s" % (l1.append("1") if out[0] > 0.5 else l1.append("0"), out[0], test))

    print()
```

```
l1_df = pd.DataFrame({'col':l1})
print (l1_df)
```

```
l1_df.to_csv('file1.csv')
```

Calculating the accuracy of Algorithms

```python
import nltk
import pandas as pd
import re
from sklearn.feature_extraction.text import TfidfVectorizer, CountVectorizer
import string


import zipfile
import pickle

from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import TfidfVectorizer

from sklearn.naive_bayes import MultinomialNB

from sklearn.metrics import confusion_matrix, classification_report
```

```python
data = pd.read_csv('/content/drive/MyDrive/YoutubeSpamMergedData.csv')
data.columns = ['comment_id', 'author', 'date', 'content', 'class']
```

```python
def count_punct(text):
    count = sum([1 for char in text if char in string.punctuation])
    return round(count/(len(text) - text.count(" ")), 3)*100

data['body_len'] = data['content'].apply(lambda x: len(x) - x.count(" "))
data['punct%'] = data['content'].apply(lambda x: count_punct(x))
```

```python
stopwords = nltk.corpus.stopwords.words('english')
ps = nltk.PorterStemmer()

def clean_text(text):
    text = "".join([word.lower() for word in text if word not in string.punctuation])
    tokens = re.split('\W+', text)
    text = [ps.stem(word) for word in tokens if word not in stopwords]
    return text
```

```python
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import GridSearchCV

tfidf_vect = TfidfVectorizer(analyzer=clean_text)
X_tfidf = tfidf_vect.fit_transform(data['content'])
X_tfidf_feat = pd.concat([data[['body_len', 'punct%']].reset_index(drop=True), pd.DataFrame(X_tfidf.toarray())], axis=1)
```

```
rf = RandomForestClassifier(n_estimators=300, max_depth=None, n_jobs=-1)

start = time.time()
rf_model = rf.fit(X_train_vect, y_train)
end = time.time()
fit_time = (end - start)

start = time.time()
y_pred = rf_model.predict(X_test_vect)
end = time.time()
pred_time = (end - start)

precision, recall, fscore, train_support = score(y_test, y_pred, pos_label=1, average='binary')
print('Fit time: {} \nPredict time: {} \nPrecision: {} \nRecall: {} \nAccuracy: {}'.format(
    round(fit_time, 3), round(pred_time, 3), round(precision, 3), round(recall, 3), round((y_pred==y_test).sum()/len(y_pred),
```

```
from sklearn.metrics import plot_confusion_matrix

confusion_matrix = plot_confusion_matrix(rf, X_test_vect, y_test, display_labels=["spam", "ham"], values_format=".1f")
pyplot.title('Confusion Matrix of Random Forest Classification')
pyplot.show()
```

```
gb = GradientBoostingClassifier(n_estimators=150, max_depth=11)

start = time.time()
gb_model = gb.fit(X_train_vect, y_train)
end = time.time()
fit_time = (end - start)

start = time.time()
y_pred = gb_model.predict(X_test_vect)
end = time.time()
pred_time = (end - start)

precision, recall, fscore, train_support = score(y_test, y_pred, pos_label=1, average='binary')
print('Fit time: {} \nPredict time: {} \nPrecision: {} \nRecall: {} \nAccuracy: {}'.format(
    round(fit_time, 3), round(pred_time, 3), round(precision, 3), round(recall, 3), round((y_pred==y_test).sum()/len(y_pred),
```

```
confusion_matrix = plot_confusion_matrix(gb, X_test_vect, y_test, display_labels=["spam", "ham"], values_format=".1f")
pyplot.title('Confusion Matrix of Gradient Boosting Classification')
pyplot.show()
```