

# **A CASCADED ENSEMBLE BASED YOUTUBE SPAM COMMENTS DETECTION USING MACHINE LEARNING**

**A PROJECT REPORT**

*Submitted by*

**CHEGIREDY KEERTHANA [REGISTER NO: 211419104046]**

**SHALINI B [REGISTER NO: 211419104244]**

**SWATHI S [REGISTER NO: 211419104282]**

*in partial fulfillment for the award of the degree*

*of*

**BACHELOR OF ENGINEERING**

**IN**

**COMPUTER SCIENCE AND ENGINEERING**



**PANIMALAR ENGINEERING COLLEGE, CHENNAI-600123.**

**ANNA UNIVERSITY : CHENNAI 600 025**

**APRIL 2023**

**PANIMALAR ENGINEERING COLLEGE**  
**(An Autonomous Institution, Affiliated to Anna University,**  
**Chennai)**

**BONAFIDE CERTIFICATE**

Certified that this project report “**A CASCADED ENSEMBLE BASED YOUTUBE SPAM COMMENTS DETECTION USING MACHINE LEARNING**” is the bonafide work of “**CHEGIREDDY KEERTHANA [REGISTER NO: 211419104046], SHALINI B [REGISTER NO: 211419104244] and SWATHI S [REGISTER NO: 211419104282]**” who carried out the project work under my supervision.

**SIGNATURE**

**Dr.L.JABASHEELA,M.E.,Ph.D**  
**HEAD OF THE DEPARTMENT,**  
DEPARTMENT OF CSE,  
PANIMALAR ENGINEERING  
COLLEGE,  
NAZARATHPETTAI,  
POONAMALLEE,  
CHENNAI-600 123

**SIGNATURE**

**Dr.A.HEMLATHADHEVI, M.E., Ph.D.,**  
**ASSOCIATE PROFESSOR,**  
DEPARTMENT OF CSE,  
PANIMALAR ENGINEERING  
COLLEGE,  
NAZARATHPETTAI,  
POONAMALLEE,  
CHENNAI-600 123.

Certified that the above candidate(s) was/ were examined in the End Semester Project  
Viva-Voce Examination held on.....

**INTERNAL EXAMINER**

**EXTERNAL EXAMINER**

## **DECLARATION BY THE STUDENT**

We **CHEGIREDDY KEERTHANA [REGISTER NO: 211419104046], SHALINI B [REGISTER NO: 211419104244] and SWATHI S [REGISTER NO: 211419104282]** hereby declare that this project report titled “**A CASCADED ENSEMBLE BASED YOUTUBE SPAM COMMENTS DETECTION USING MACHINE LEARNING**”, under the guidance of **Dr. A. HEMLATHADHEVI, M.E., Ph.D., ASSOCIATE PROFESSOR** , is the orginial work done by us and we have not plagiarized or submitted to any other degree in any university by us.

**1. CHEGIREDDY  
KEERTHANA**

**2. SHALINI B**

**3. SWATHI S**

## ACKNOWLEDGEMENT

We would like to express our deep gratitude to our respected Secretary and Correspondent **Dr.P.CHINNADURAI, M.A., Ph.D.** for his kind words and enthusiastic motivation, which inspired us a lot in completing this project.

We express our sincere thanks to our beloved Directors **Tmt.C.VIJAYARAJESWARI, Dr.C.SAKTHI KUMAR,M.E.,Ph.D** and **Dr.SARANYASREE SAKTHI KUMAR B.E.,M.B.A.,Ph.D.**, for providing us with the necessary facilities to undertake this project.

We also express our gratitude to our Principal **Dr.K.Mani, M.E., Ph.D.** who facilitated us in completing the project.

We thank the Head of the CSE Department, **Dr. L.JABASHEELA , M.E.,Ph.D.,** for the support extended throughout the project.

We would like to thank my **Project Guide Dr. A. HEMLATHADHEVI, M.E., Ph.D., ASSOCIATE PROFESSOR** and all the faculty members of the Department of CSE for their advice and encouragement for the successful completion of the project.

**CHEGIREDY  
KEERTANA**

**SHALINI B**

**SWATHI S**

## **ABSTRACT**

YouTube possesses a spam filtering mechanism, although it consistently fails to deal with spam effectively. As a corollary, related studies on classifying YouTube spam comments have been done and classification experiments were performed. The subscribers of this social network might easily be impacted by these YouTube spam messages. Spam comments on YouTube have turned out to be a social issue and may spread more rapidly than the actual information. Among different kind of undesired content, YouTube is facing problems to manage the huge volume of undesired text comments posted by users that aim to self-promote their videos, or to disseminate malicious links to steal private data. To stop such kind of activities this spam detection can be supportive. Many machine learning approaches are utilized to analyze enormous, complicated data, assisting professionals in predicting YouTube spam comments. The most accurate model is applied to forecast the spam YouTube comments is known as cascade ensemble model. The algorithms that are used in this model are Logistic regression, Random Forest, Multinomial Naive Bayes algorithms.

**Keywords** – Machine learning, Spam or Ham, Cascade Ensemble model, Ensemble Hard and Ensemble Soft, Logistic Regression, Random Forest Classifier, Multinomial Naïve Bayes, Flask

# TABLE OF CONTENTS

CHAPTER NO.	TITLE	PAGE NO.
	<b>ABSTRACT</b>	iii
	<b>LIST OF TABLES</b>	viii
	<b>LIST OF FIGURES</b>	ix
	<b>LIST OF SYMBOLS, ABBREVIATIONS</b>	x
<b>1.</b>	<b>INTRODUCTION</b>	
	1.1 Problem Definition	2
	1.2 Scope of the project	2
<b>2.</b>	<b>LITERATURE SURVEY</b>	4
<b>3.</b>	<b>SYSTEM ANALYSIS</b>	
	3.1 Existing System	14
	3.2 Proposed system	14
	3.3 Feasibility Study	15
	3.4 Hardware Environment	17
	3.5 Software Environment	17
<b>4.</b>	<b>SYSTEM DESIGN</b>	
	4.1. ER diagram	19
	4.2 Data dictionary	19
	4.3 Data Flow Diagram	20
	4.4 UML Diagrams	23

<b>CHAPTER NO.</b>	<b>TITLE</b>	<b>PAGE NO.</b>
<b>5.</b>	<b>SYSTEM ARCHITECTURE</b>	
	5.1 Module Design Specification	30
	5.2 Algorithms	47
<b>6.</b>	<b>SYSTEM IMPLEMENTATION</b>	
	6.1 Client-side coding	56
	6.2 Server-side coding	62
<b>7.</b>	<b>SYSTEM TESTING / PERFORMANCE ANALYSIS</b>	
	7.1 Test Cases & Reports	91
<b>8.</b>	<b>CONCLUSION</b>	
	8.1 Conclusion	94
	8.2 Future Enhancements	94
	<b>APPENDICES</b>	
	A.1 Sample Screens	95
	A.2 Plagiarism Report	97
	<b>REFERENCES</b>	107

## LIST OF TABLES

<b>TABLE NO.</b>	<b>TABLE DESCRIPTION</b>	<b>PAGE NO.</b>
3.1	OPERATIONAL FEASIBILITY	16
5.1	DATASET DETAILS	32
5.2	PERFORMANCE OF THE ALGORITHMS	44

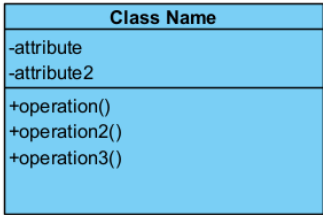
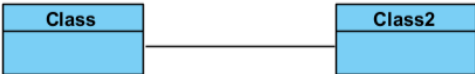

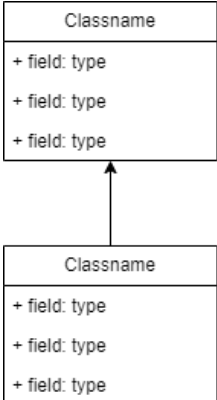


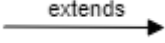



## **LIST OF FIGURES**

<b>FIG NO.</b>	<b>FIGURE DESCRIPTION</b>	<b>PAGE NO.</b>
3.1	OPERATIONAL FEASIBILITY GRAPH	17
4.1	ER DIAGRAM	19
4.2	DFD LEVEL 0	21
4.3	DFD LEVEL 1	22
4.4	DFD LEVEL 2	23
4.5	USE-CASE DIAGRAM	24
4.6	CLASS DIAGRAM	25
4.7	ACTIVITY DIAGRAM	26
4.8	SEQUENCE DIAGRAM	27
4.9	COLLABORATION DIAGRAM	28
5.1	SYSTEM ARCHITECTURE DIAGRAM	30
5.2	STRUCTURE OF THE DATASET	32
5.3	MODULE 1 SEQUENCE DIAGRAM	34
5.5	DATA VISUALIZATION	37

5.6	MODULE 2 SEQUENCE DIAGRAM	38
5.7	MODULE 3 SEQUENCE DIAGRAM	43
5.10	ESM-H	45
5.11	MODULE 4 SEQUENCE DIAGRAM	46
A.1	MAIN PAGE	95
A.2	ENTERING A HAM COMMENT	96
A.3	DETECTION OF HAM	96
A.4	ENTERING A SPAM COMMENT	97
A.5	DETECTION OF SPAM COMMENT	97

## LIST OF SYMBOLS

S.NO	NOTATION NAME	NOTATION	DESCRIPTION
1.	Class		Represents a collection of similar entities grouped together.
2.	Association		Associations represents a static relationships between the classes.
3.	Actor		It aggregates several classes into a single class.
4.	Aggregation		Interaction between the system and the external environment.

5.	Relation(uses)	<b>uses</b>	Used for additional process communication.
6.	Relation(extends)		Extends relationship is used when one use case is similar to another use case but does a bit more.
7.	Communication		Communication between various use cases.
8.	State		State of the process.
9.	Initial State		Initial state of the object.

10. Final state



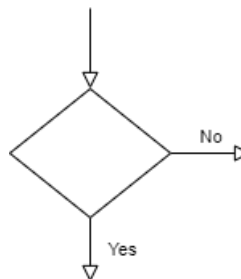
Final state of the object.

11. Control flow



Represents various control flow between the states.

12. Decision box



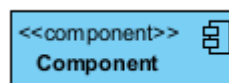
Represents decisions making process from a constraint.

13. Use case

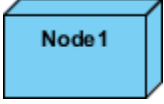
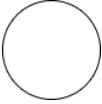




Interaction between the system and external environment.

14. Component



Represents physical modules which is a collection of components.

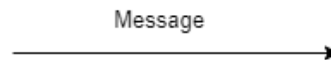
- |     |                       |  |   |
|-----|-----------------------|--|---|
| 15. | Node                  |     | Represents physical modules which are a collection of components.                                   |
| 16. | Data<br>Process/State |     | A circle in DFD represents a state or process which has been triggered due to some event or action. |
| 17. | External Entity       |  | Represents external entities such as keyboard, sensors, etc.,                                       |
| 18. | Transition            |  | Represents communication that occurs between processes.   |

19. Object Lifeline



Representation of the vertical dimensions of the objects communication.

20. Message



Represents the message exchanged.

## **LIST OF ABBREVIATIONS**

<b>S.NO</b>	<b>ABBREVIATION</b>	<b>EXPANSION</b>
1.	ESM-H	Ensemble Hard
2.	ESM-S	Ensemble Soft
3.	LR	Logistic Regression
4.	RF	Random Forest
5.	MNB	Multinomial Naïve Bayes



# **1. INTRODUCTION**

# **1. INTRODUCTION**

## **1.1 PROBLEM DEFINITION**

The way to detect spam comments is through cascaded ensemble method which will be the focus of the study. Cascaded Ensemble follows two methods ensemble hard voting and soft voting. Experimentation with several traditional machine learning models to set a baseline and then compare results to the state-of-the art deep networks to classify the stance between spam and ham comments. Spam comments can be come in many forms, including: outright false stories, fake giveaways or the stories which are developed to mislead and influence reader's opinion. While spam comments may have multiple forms, the effect that it can have on people, government and organizations may generally be negative since it differs from the facts. Detecting spam comments is hard for many reasons. Assessing the veracity of YouTube comments are a complex and cumbersome task, even for trained experts. These complexities make it a daunting task to classify spam comments.

## **1.2 SCOPE OF THE PROJECT**

There are numerous factors that contribute to the dissemination of YouTube Spam. The first is due to a lack of data among the public. YouTube Spam detection is attempted but the time-consuming human process is just too slow to prevent the false information. Detecting YouTube Spam automatically is a difficult task that defies present content-based analysis method. The cascaded ensemble methodology will have a huge detrimental influence on the public. The automatic fact-checking procedure is the second reason.

## **2. LITERATURE SURVEY**

## 2. LITERATURE SURVEY

1. **TITLE:** A YouTube Spam Comments Detection Scheme Using Cascaded Ensemble Machine Learning Model

**AUTHOR:** H. Oh

**YEAR:** 2022

**DESCRIPTION:** Algorithms like Decision tree, Logistic Regression, Naïve Bayes and SVM are used. Ensemble Soft voting ESM-S is performed which enhances the accuracy of the classification.

**DISADVANTAGES:** Time consuming and TF-IDF technique is not used.

2. **TITLE:** Hybrid Ensemble Framework With Self-Attention Mechanism For Social Spam Detection On Imbalanced Data

**AUTHOR:** Sanjeev Rao, Anil Kumar Verma, Tarunpreet Bhatia

**YEAR:** 2023

**DESCRIPTION:** In the proposed framework, the datasets are balanced using NearMiss and SmoteTomek techniques to feed several machine-learning models. Later, the baseline ML models and proposed voting-based ensemble models are evaluated on imbalanced and balanced datasets.

**DISADVANTAGES:** OSN spam is a critical problem that needs to be addressed. Existing ML/DL- based approaches are ineffective due to inadequacies such as class imbalance problems, spam drift, lack of larger labeled datasets, contextual features, performance issues, etc.

3. **TITLE:** A review of spam email detection: analysis of spammer strategies and the dataset shift problem.

**AUTHOR:** Jáñez-Martino, F., Alaiz-Rodríguez, R., González-Castro, V

**YEAR:** 2023

**DESCRIPTION:** Spam email filtering has been tackled using different machine learning approaches including NB, SVM, Random Forest (RF) or Neural Networks (NN), among others.

**DISADVANTAGES:** Effective strategies able to tackle the dataset shift and adversarial manipulation problems are necessary in order to handle security attacks and detect spammer corruption in data.

**4. TITLE:** Youtube Spam Comments Detection

**AUTHOR:** Gubbala Pranathi , Sri.S.K.Alisha, Sri.V.Bhaskara Murthy

**YEAR:** 2022

**DESCRIPTION:** The survey for the spam comments detection methodology has been carried out using four Artificial Intelligence estimations – Logistic Regression, Ada Boost, Decision Tree and Random Forest. With the use of Neural Network, achieve an exactness of 91.65% and beat the present course of action by around 18%.

**DISADVANTAGES:** Due to over fitting in the data it is not accurate compared with other algorithms.

**5. TITLE:** A Hybrid Spam Detection Framework For Social Networks

**AUTHOR:** Oğuzhan ÇITLAK, Murat DÖRTERLER, İbrahim Alper DOĞRU

**YEAR:** 2022

**DESCRIPTION:** It is aimed to detect spam accounts on social network and the spam detection policy of these networks is intended to support. Keywords are Social networks, spam detection, short link analysis, machine learning, text analysis.

**DISADVANTAGES:** When analysing a large dataset, it may take a little more time due to these limitations.

6. **TITLE:** Spam SMS filtering based on text features and supervised machine learning techniques

**AUTHOR:** Muhammad Adeel Abid, Saleem Ullah, Muhammad Abubakar Siddique, Muhammad Faheem Mushtaq, Wajdi Aljedaani & Furqan Rustam

**YEAR:** 2022

**DESCRIPTION:** The support vector classifier, gradient boosting machine, random forest, Gaussian Naive Bayes, and logistics regression are applied on the spam and ham SMS dataset to evaluate the performance using accuracy, precision, recall, and F1 score.

**DISADVANTAGES:** The most recent works on filtering spam emails emphasising on the phases of feature extraction and feature selection to mitigate overfitting by reducing the dimensionality of the input space.

7. **TITLE:** Spam Message Detection Using Danger Theory And Krill Herd Optimization

**AUTHOR:** Aakanksha Sharaff a, Chandramani Kamal a, Siddhartha Porwal a, Surbhi Bhatia b, Kuljeet Kaur c, Mohammad Mehendi Hassan d

**YEAR:** 2022

**DESCRIPTION:** This paper uses a biologically inspired algorithm named Krill herd Optimization (KHO) for the task of feature selection and various optimization functions like Quing function, Sumsquare function, Levy function etc. are applied for enhancing its performance. The Dendritic Cell Algorithm

(DCA) is also incorporated with KHA as an added advantage towards achieving efficiency. Comparative results between Dendritic Cell Algorithm (DCA) with KHA and other spam filtering models have been shown in comparison with several state-of-the-art machine learning classifiers. The algorithms have been experimented by using varied optimization functions illustrated using visualization tools and results have been validated in the paper. The obtained results demonstrate an admissible accuracy of 96% that is calculated using different information retrieval metrics using recall, F-measure and precision.

**DISADVANTAGES:** In this graph, different models have been tested and evaluated, out of which DCA algorithm has been found most promising algorithm. DCA is an unsupervised method.

## **8. TITLE:** A Spam Transformer Model For Sms Spam Detection

**AUTHOR:** Xiaoxu Liu , Haoye Lu , And Amiya Nayak

**YEAR:** 2022

**DESCRIPTION:** In this paper, we aim to explore the possibility of the Transformer model in detecting the spam Short Message Service (SMS) messages by proposing a modified Transformer model that is designed for detecting SMS spam messages. The evaluation of our proposed spam Transformer is performed on SMS Spam Collection v.1 dataset and UtkML's Twitter Spam Detection Competition dataset, with the benchmark of multiple established machine learning classifiers and state-of-the-art SMS spam detection approaches. In comparison to all other candidates, our experiments on SMS spam detection show that the proposed modified spam Transformer has the optimal results on the accuracy, recall, and F1-Score with the values of

98.92%, 0.9451, and 0.9613, respectively. Besides, the proposed model also achieves good performance on the UtkMI's Twitter dataset, which indicates a promising possibility of adapting the model to other similar problem.

**DISADVANTAGES:** SMS spam detection and confirms the availability of the Transformer on this problem, the model is still far from optimal. There are some improved models based on the Transformer with more complex architecture such as GPT-3 [4] and BERT [5] that could be explored in the future.

## 9. **TITLE:** Context-Dependent Model For Spam Detection On Social Networks

**AUTHOR:** Ghanem, R., Erbay, H

**YEAR:** 2022

**DESCRIPTION:** There are two main training algorithms for Word2vec: Skip Gram and Continous Bag Of Words (CBOW). Fast Text, an extension of the word2vec model, is another common word embedding method developed by Facebook in 2016. In Fast Text, each word in the corpus is represented as an n-gram of characters, which helps to capture the meaning of shorter words and allows the embeddings to understand suffixes and prefixes. The other benefit of using the Fast Text model is the ability to work with the rare words that haven't seen them before during the training data.

**DISADVANTAGES:** Improvement of the architecture of the proposed model and conduct an empirical study to tune the hyperparameters. Traditional weighting methods and word embedding methods cannot handle the problem of short text classification effectively because of some previously mentioned limitations.



**10.TITLE:** Traditional And Context-Specific Spam Detection In Low Resource Settings

**AUTHOR:** Kawintiranon, K., Singh, L. & Budak, C.

**YEAR:** 2022

**DESCRIPTION:** The neural network model outperforms the traditional models with an F1 score of 0.91. Because spam training data sets are notoriously imbalanced, we also investigate the impact of this imbalance and show that simple Bag-of-Words models are best with extreme imbalance, but a neural model that fine-tunes using language models from other domains significantly improves the F1 score, but not to the levels of domain-specific neural models. This suggests that the strategy employed may vary depending upon the level of imbalance in the data set, the amount of data available in a low resource setting, and the prevalence of context-specific spam vs. traditional spam. Finally, we make our data sets available for use by the research community.

**DISADVANTAGES:** The structure of a neural network is disparate from the structure of microprocessors therefore required to be emulated. When an item of the neural network declines, it can continue without some issues by its parallel features.

**11.TITLE:** A Research on Efficient Spam Detection Technique for Iot Devices Using Machine Learning

**AUTHOR:** Aijaz Ali Khan, Rahul M. Mulajkar, Vajid N Khan, Shrinivas K. Sonkar, Dattatray G. Takale

**YEAR:** 2022

**DESCRIPTION:** The technique that has been suggested can identify the spam parameters that are affecting the devices connected to the internet of things. In order to get the best possible outcomes, the Internet of Things data set is used in the validation of the proposed strategy. The Internet of Things (IoT) makes it possible for real-world devices, regardless of where they are physically located, to merge with one another and deploy new technologies.

**DISADVANTAGES:** Enhancement the reliability and safety of Internet of Things devices by taking into account environmental and contextual factors.

**12.TITLE:** A Method for SMS Spam Message Detection Using Machine Learning.

**AUTHOR:** Saeed, Vaman

**YEAR:** 2022

**DESCRIPTION:** This study investigates the effectiveness of various supervised machine learning algorithms, such as the J48, K-Nearest Neighbors (KNN), and Decision Tree (DT), in identifying spam and ham communications. Experiments showed that the Decision Tree method obtained higher accuracy than other machine learning classifiers. The performance of the DT model is superior to that of other models, achieving an accuracy of 97.05%.

**DISADVANTAGES:** Limitations of decision trees is that they are largely unstable compared to other decision predictors.

**13.TITLE:** Detection of Abusive Bengali Comments for Mixed Social Media Data Using Machine Learning

**AUTHOR:** Sherin Sultana,Md Omur Faruk Redoy,Jabir Al Nahian,Jabir Al Nahian,Sheikh Abujar

**YEAR:** 2022

**DESCRIPTION:** Many people use social media for their livelihood. Social media has a lot of influence on our life from different aspects. Although there are many positive aspects, the trend of negative comments on social media has become a serious problem these days. Through this study, we have detected bad comments made in the Bengali language on social media using machine learning algorithms and measured those performances. Although much work has been done on this issue in other languages, it is scarce in the Bengali language.

**DISADVANTAGES:** SVM algorithm is not suitable for large data sets. SVM does not perform very well when the data set has more noise i.e. target classes are overlapping.

**14. TITLE:** Detection of Offensive Comments for Textual Data Using Machine Learning

**AUTHOR:** Rhea Hooda, Arunima Jaiswa,Isha Bansal, Mehak Jain, Pranjli Singh & Nitin Sachdeva

**YEAR:** 2022

**DESCRIPTION:** Social Media facilitates the sharing and spreading of information, thoughts, and ideas. However, just like any other innovation, it influences people in a harsh or another way. They have become a platform for spreading hatred, negative comments, and cyberbullying. Cyberbullying is

bullying that occurs via digital technologies for example social media, messaging platforms, gaming platforms, and mobile phones. Cyberbullying includes posting, sending or sharing negative, mean, and harmful content. A lot of efforts are being made by researchers for the detection of cyberbullying on social networking sites. The research in this paper focuses on detecting offensive comments for textual data.

**DISADVANTAGES:** The drawback was that some comments were not able to classify correctly which were sarcastic.

**15. TITLE:** Visualization Technology and Deep-Learning for Multilingual Spam Message Detection

**AUTHOR:** Hwabin Lee,Sua JeongORCID,Seogyong Cho andEunjung Choi

**YEAR:** 2022

**DESCRIPTION:** The study proposes a text-processing method and a string-imaging method. The CNN 2D visualization technology used in this paper can be applied to datasets of various languages by processing the data as images, so they can be equally applied to languages other than English.

**DISADVANTAGES:** In the case of the Korean dataset, there were difficulties in tokenizing according to spaces, as Korean has a variable word form, which changes form depending on the context, even if it is the same word.

### **3. SYSTEM ANALYSIS**

### **3. SYSTEM ANALYSIS**

#### **3.1 EXISTING SYSTEM**

YouTube performs spam comment filtering that is not quite effective. With YouTube comments, applying the same method (language modeling) doesn't work as the features of the data are different. It uses same method for spam comments detection that are used in detecting spam on websites. Features of YouTube comments represent less textual descriptions and information. YouTube spam comments is a major threat to democracy like influencing public opinion, and its impact cannot be understated particularly in the current socially and digitally connected society. Researchers from different disciplines like computer science, political science, information science, and linguistics have also studied the dissemination, detection, and mitigation of YouTube spam comments. However, it remains challenging to detect and prevent the dissemination of YouTube spam comments by the same mechanism used for filtering spam mails and spam websites. So this makes the current system inefficient. The drawbacks found in this system is that the performance accuracy is low and the TF-IDF technique is not used

#### **3.2 PROPOSED SYSTEM**

The proposed model is to build a machine learning model that is capable of classifying whether the YouTube comment is spam or not. The YouTube spam comments are considered to be widespread and controlling them is very difficult as the world is developing toward digital everyone now has access to internet and they can post whatever they want. So there is a greater chance for the people to get misguided. The machine learning is generally build to tackle these type of complicated task like it takes more amount of time to analyse

these type of data manually. The machine learning can be used to classify the YouTube spam comments by using the previous data and make them to understand the pattern and improve the accuracy of the model by adjusting parameters and use that model as the classification model. The method that is used to classify the comments is based on Cascade Ensemble model. There are two types of cascade ensemble model: ESM-H (Ensemble with hard voting) and ESM-S (Ensemble with soft voting). For YouTube spam comment detection the best method of ensemble voting is the hard voting. Using flask, a deployment model is constructed. A website where comment can be entered and the model identifies whether they are spam or ham.

### **3.3 FEASIBILITY STUDY**

The feasibility of the system has been studied from the various aspects like whether the system is feasible technically, operationally and economically. The present technology is found to be sufficient to meet the requirements of the system. This system is believed to work well when it is developed and installed. Hence, operational feasibility is achieved. Since the requirements for the project are easily available, it's headed with the intention to use the available resources to fulfill the system requirement. The detail feasibility study conducted is mentioned below:

- **TECHNICAL FEASIBILITY:**

The technology needed for the proposed system to develop is available. The work for the project is done with current equipment existing tools like python. The development of the system still using this technology if needed to upgrade. In future, if new technology is used like android app of this system it

is possible. Hence, the system that is developed will successfully satisfy the needs of the system for technical feasibility.

- **ECONOMIC FEASIBILITY:**

Since the system is developed as a part of project work, there is no manual cost to spend for the proposed system. Also all the resources are already available, it gives an indication that the system is economically possible for development. Economic justification is generally the "Bottom Line" consideration for most systems. The cost to conduct a full system investigation is negotiable because required information is collected from internet. This can run in system with normal hardware like desktop, laptop mobiles and so on. This system won't require extra specific software to use it. Hence, the project that is developed won't require enormous amount of money to be developed so it will be economically feasible.

- **OPERATIONAL FEASIBILITY:**

The interface will be user friendly and no training will be required to use the application. The solution proposed for this project is operationally workable and most likely convenient to solve the detection of spam comments. The accuracy comparison which determines the operational feasibility is compared below:

ALGORITHMS	EXISTING SYSTEM	PROPOSED SYSTEM
DT	84.67	0
LR	89	91.66
NB-B	87	0
SVM-L	89	0
SVM-R	88.5	
RF	0	100
MNB	0	97.74

TABLE NO.3.1 OPERATIONAL FEASIBILITY



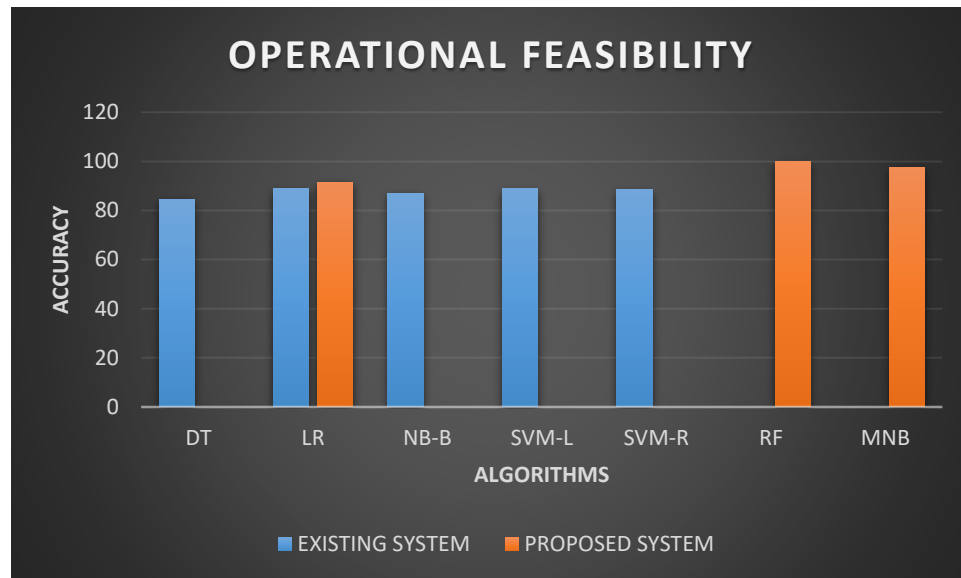


FIGURE 3.1. OPERATIONAL FEASIBILITY GRAPH

### 3.4 HARDWARE ENVIRONMENT:

- PROCESSOR : Intel i3
- HARD DISK : Minimum 80 GB
- RAM : Minimum 2 GB

### 3.5 SOFTWARE ENVIRONMENT:

- FRONT END : HTML
- BACK END : Anaconda with Jupyter Notebook
- OPERATING SYSTEM : WINDOWS 10
- LANGUAGE : PYTHON

## **4.SYSTEM DESIGN**

## 4. SYSTEM DESIGN

### 4.1 ER DIAGRAM

An entity relationship diagram (ERD), also known as an entity relationship model, is a graphical representation that depicts relationships among people, objects, places, concepts or events within an information technology (IT) system.

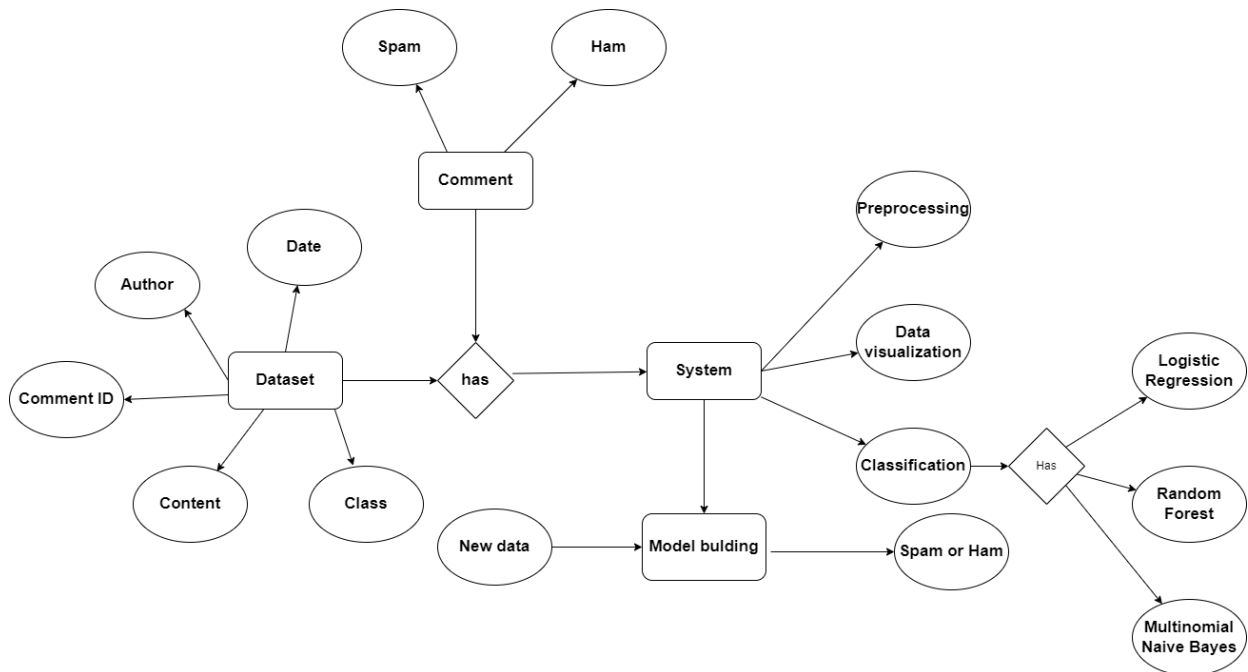


FIG NO. 4.1 ER DIAGRAM

### 4.2 DATA DICTIONARY

This is normally represented as the data about data. It is also termed as metadata some times which gives the data about the data stored in the database.

It defines each data term encountered during the analysis and design of a new system. Data elements can describe files or the processes. Following are some rules, which defines the construction of data dictionary entries:

1. Words should be defined to understand for what they need and not the variable need by which they may be described in the program.
2. Each word must be unique. We cannot have two definition of the same client.
3. Aliases or synonyms are allowed when two or more enters shows the same meaning. For example, a vendor number may also be called as customer number.
4. A self-defining word should not be decomposed. It means that the reduction of any information in to subpart should be done only if it is really required that is it is not easy to understand directly.

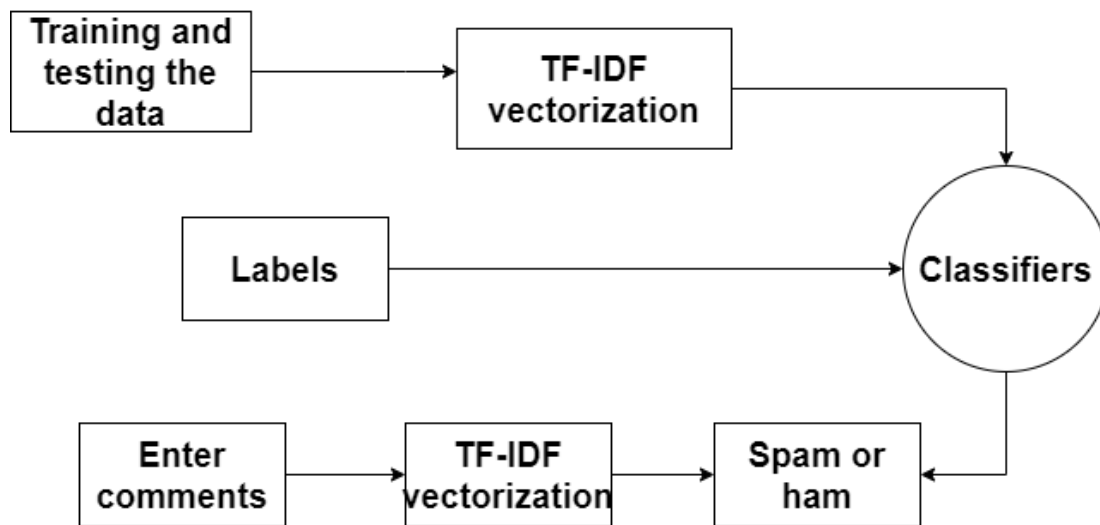
Data dictionary includes information such as the number of records in file, the frequency a process will run, security factor like pass word which user must enter to get excess to the information.

### **4.3 DATAFLOW DIAGRAMS**

A Data Flow Diagram (DFD) is a graphical representation of the “flow” of data through an information system. It differs from the flowchart as it shows the data flow instead of the control flow of the program. A data flow diagram can also be used for the visualization of the data processing. The DFD is designed to show how a system is divided into smaller portions and to highlight the flow of the data between those parts.

## LEVEL 0:

The Level 0 DFD shows how the system is divided into 'sub-systems' (processes), each of which deals with one or more of the data flows to or from an external agent, and which together provide all of the functionality of the system as a whole.

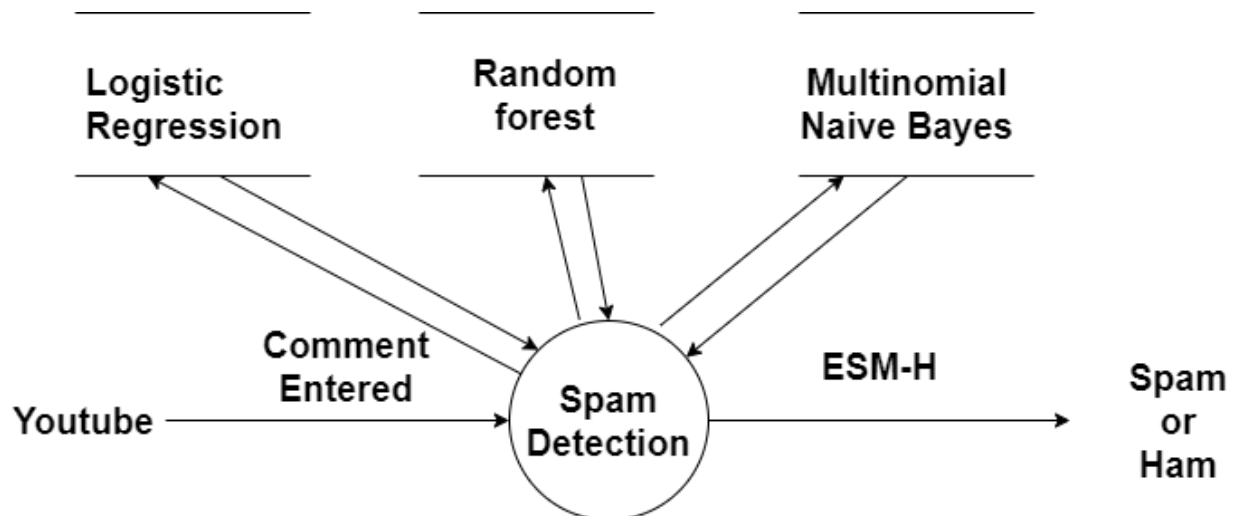


**FIG NO. 4.2 DFD LEVEL 0**

## LEVEL 1:

The next stage is to create the Level 1 Data Flow Diagram. This highlights the main functions carried out by the system. As a rule, to describe the system was using between two and seven functions - two being a simple system and seven being a

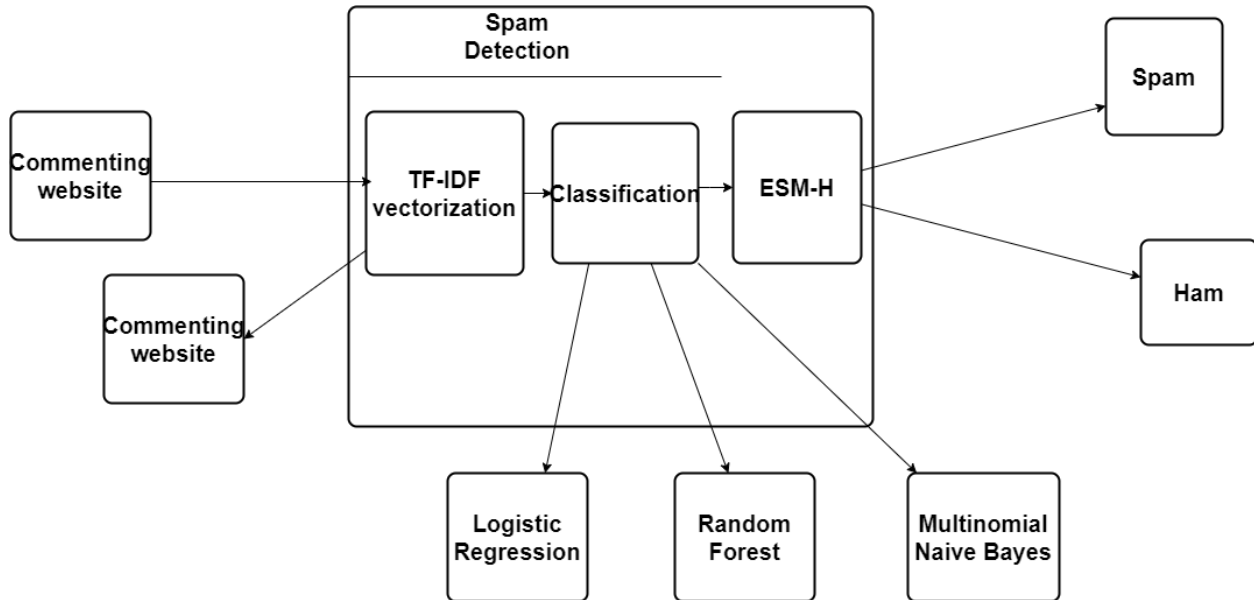
complicated system. This enables us to keep the model manageable on screen or paper



**FIG NO. 4.3 DFD LEVEL 1**

## **LEVEL 2:**

A Data Flow Diagram (DFD) tracks processes and their data paths within the business or system boundary under investigation. A DFD defines each domain boundary and illustrates the logical movement and transformation of data within the defined boundary. The diagram shows 'what' input data enters the domain, 'what' logical processes the domain applies to that data, and 'what' output data leaves the domain. Essentially, a DFD is a tool for process modeling and one of the oldest.

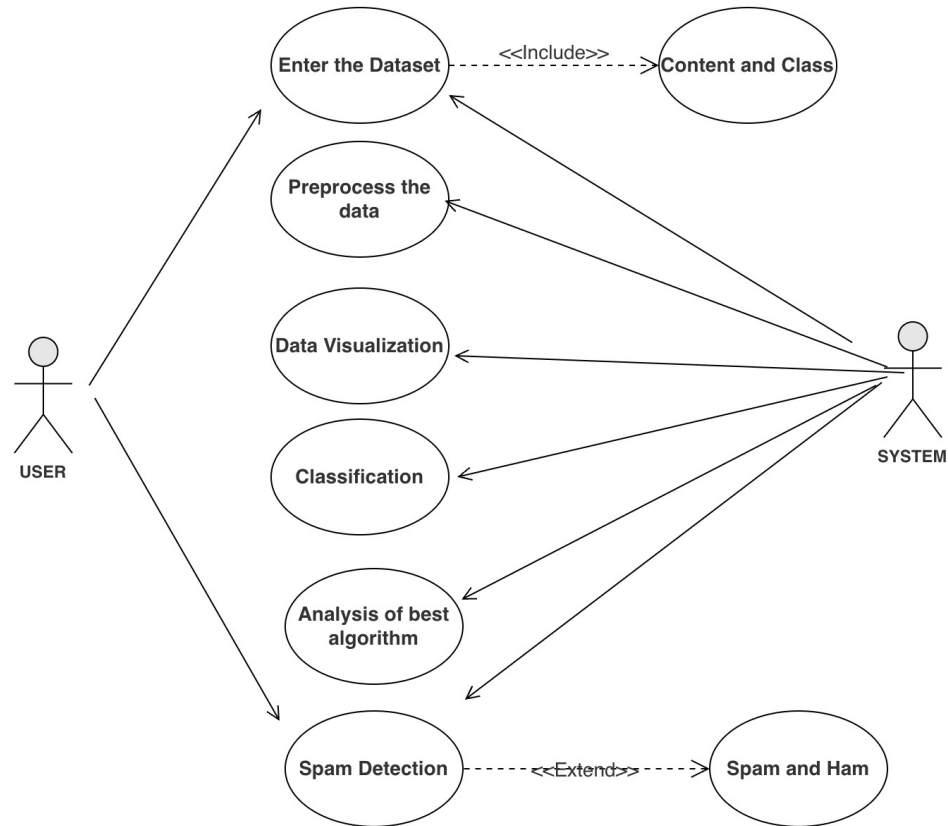


**FIG NO. 4.4 DFD LEVEL 2**

## 4.4UML DIAGRAMS

### 4.4.1 USE-CASE DIAGRAM:

Use-case diagrams describe the high-level functions and scope of a system. These diagrams also identify the interactions between the system and its actors. The use cases and actors in use-case diagrams describe what the system does and how the actors use it, but not how the system operates internally.

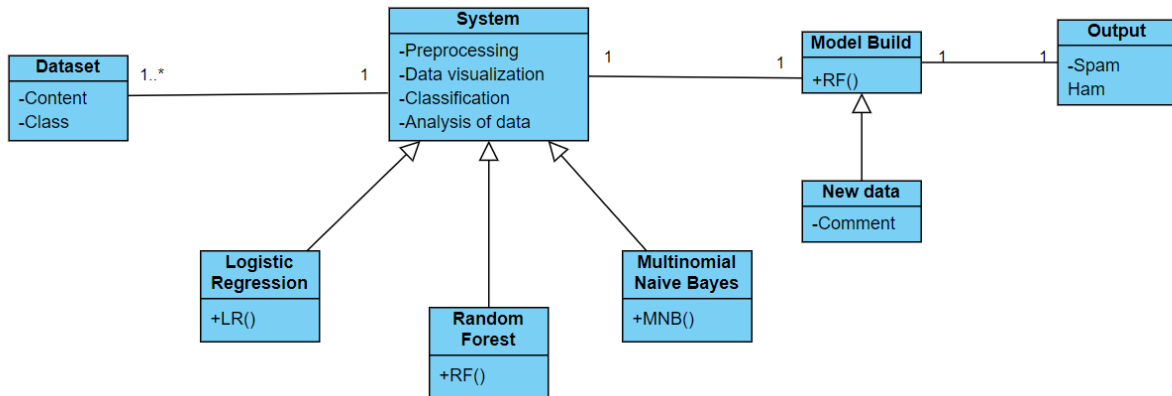


**FIG NO. 4.5 USE-CASE DIAGRAM**

## 4.5.2. CLASS DIAGRAM

The class diagram is the main building block of object-oriented modeling. It is used for general conceptual modeling of the structure of the application, and for detailed modeling, translating the models into programming code. Class diagrams can also be used for data modeling.

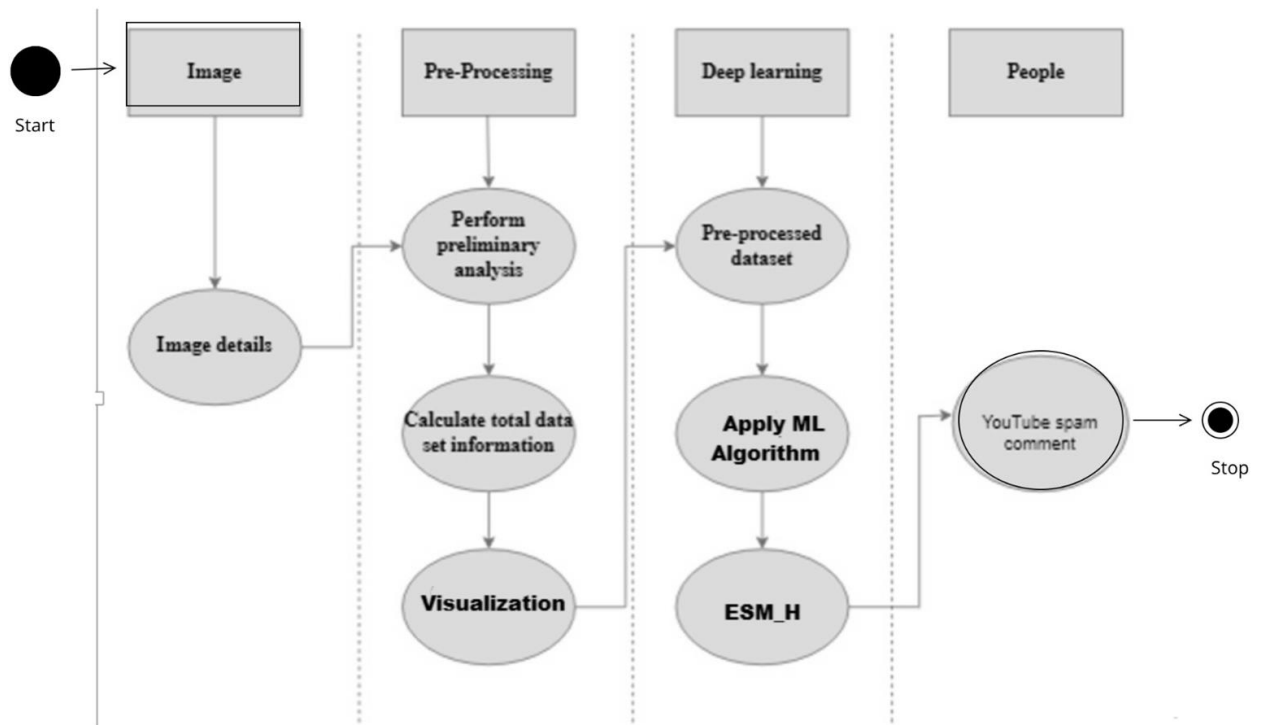




**FIG NO. 4.6 CLASS DIAGRAM**

### 4.5.3. ACTIVITY DIAGRAM

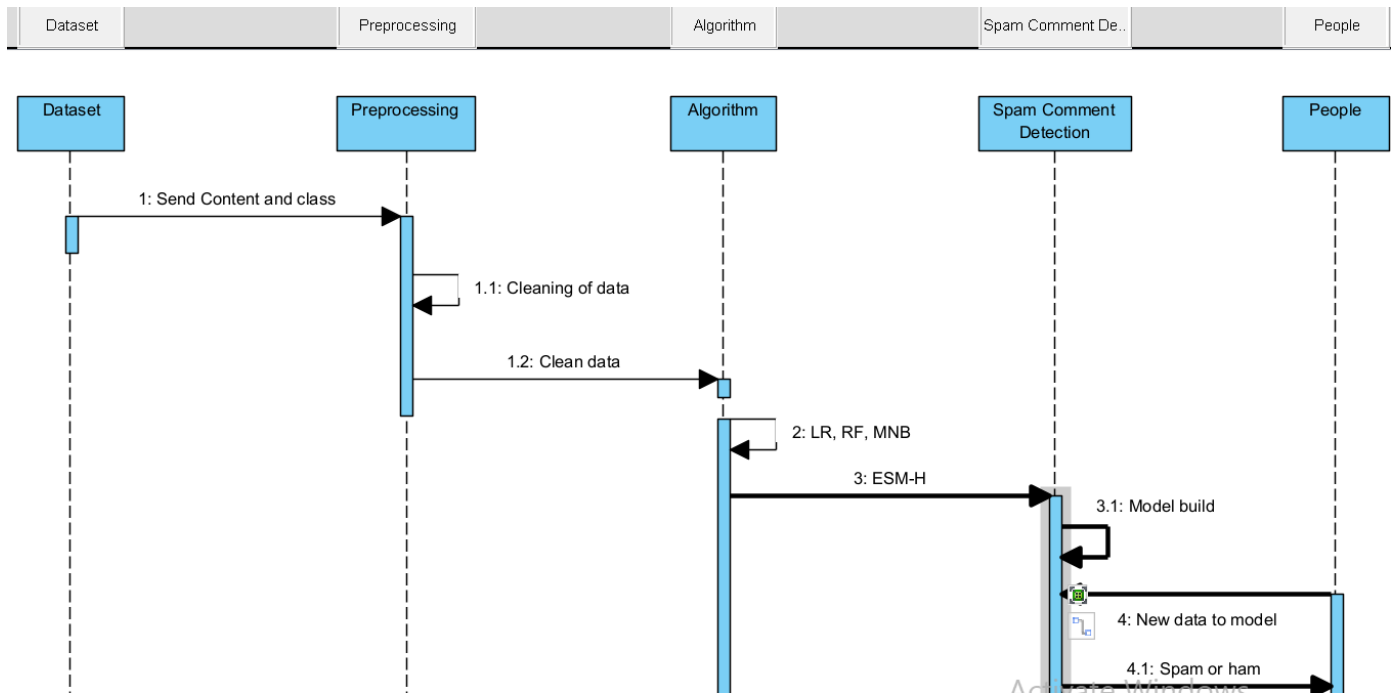
Activity diagrams are not only used for visualizing dynamic nature of a system but they are also used to construct the executable system by using forward and reverse engineering techniques. Activity diagram is some time considered as the flow chart. Although the diagrams looks like a flow chart but it is not. It shows different flow like parallel, branched, concurrent and single.



**FIG NO. 4.7 ACTIVITY DIAGRAM**

#### 4.5.4. SEQUENCE DIAGRAM

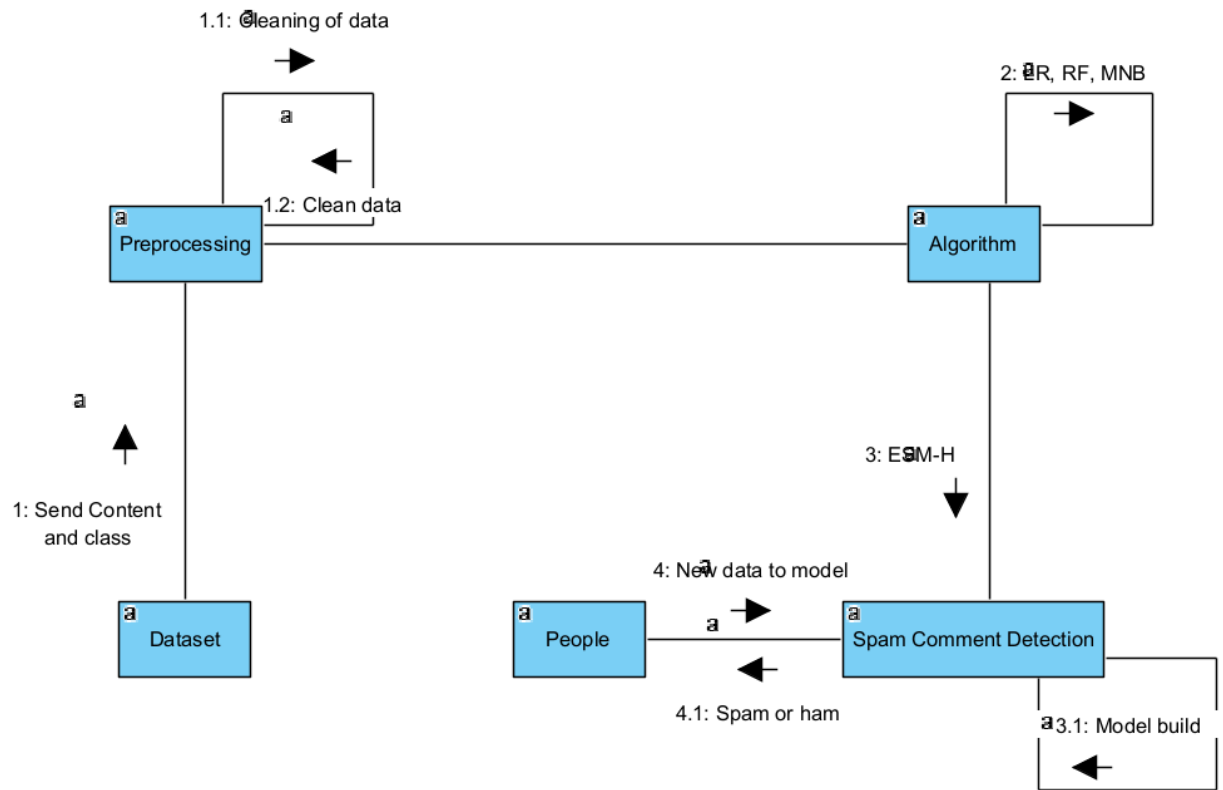
A sequence diagram is a type of interaction diagram because it describes how—and in what order—a group of objects works together. These diagrams are used by software developers and business professionals to understand requirements for a new system or to document an existing process.



**FIG NO. 4.8 SEQUENCE DIAGRAM**

### 4.5.5 COLLABORATION DIAGRAM

A collaboration diagram, also known as a communication diagram, is an illustration of the relationships and interactions among software objects in the Unified Modeling Language (UML). These diagrams can be used to portray the dynamic behavior of a particular use case and define the role of each object.



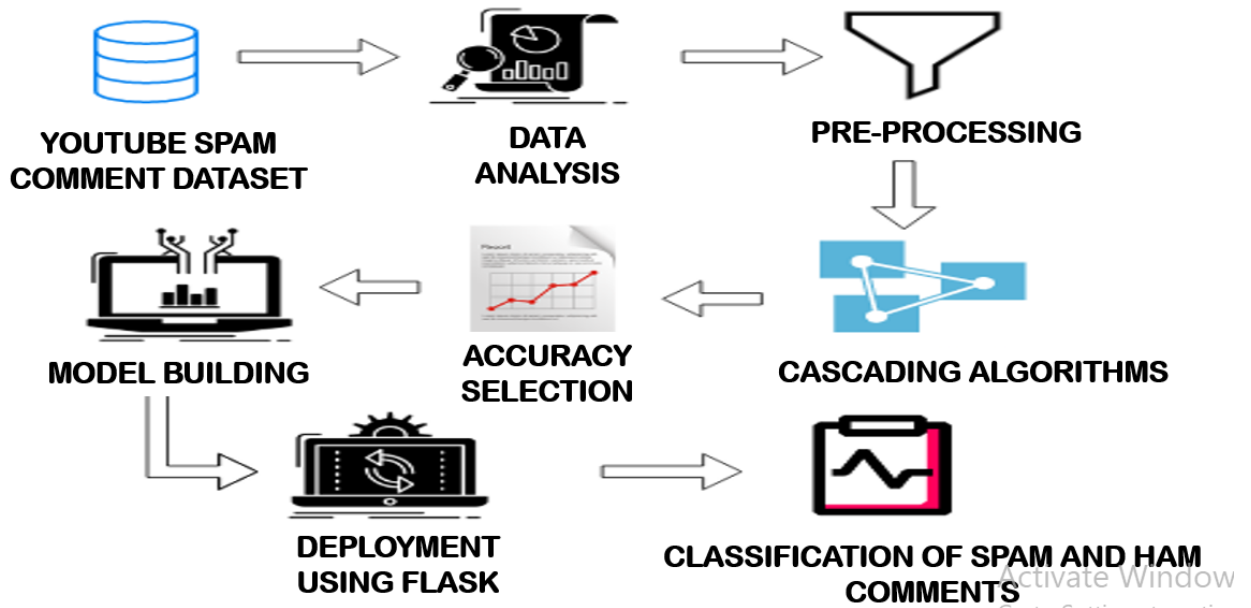
**FIG NO. 4.9 COLLABORATION DIAGRAM**

## **5.SYSTEM ARCHITECTURE**

## 5. SYSTEM ARCHITECTURE

### 5.1 MODULE DESIGN SPECIFICATION

#### SYSTEM ARCHITECTURE



**FIG NO. 5.1 SYSTEM ARCHITECTURE DIAGRAM**

Countering the spam comment phenomenon has become one of the most important challenges for YouTube. This demands a holistic approach to analyzing heterogeneous data and storing the results.

#### MODULE DESCRIPTION

There are four modules in this methodology. They are:

1. YouTube ID train model
2. Data Analysis
3. Classifications
4. Analysis Performance

## 1. YOUTUBE ID TRAIN MODEL

Machine learning data works better with numerical features so we have to convert text data into numerical columns. So we have to preprocess the text and that is called natural language processing. We can't use text data directly because it has some unusable words and special symbols and many more things. If we used it directly without cleaning then it is very hard for the ML algorithm to detect patterns in that text and sometimes it will also generate an error. So that we have to always first clean text data.

These are composed of comments from five well-known music videos. We solely make use of named classes and comment material. Each training and testing of the five data sets as shown in Table.1 can result in overfitting, where the five classifiers perform well only on that data and do not apply well to comment data in other videos. We thus include all five video datasets in this work in order to interpret the conclusion.

	<i>Spam</i>	<i>Ham</i>	<i>Total</i>
<i>Psy</i>	<i>175</i>	<i>175</i>	<i>350</i>
<i>KatyPerry</i>	<i>175</i>	<i>202</i>	<i>350</i>
<i>LMFAO</i>	<i>236</i>	<i>303</i>	<i>438</i>
<i>Eminem</i>	<i>245</i>	<i>203</i>	<i>448</i>
<i>Shakira</i>	<i>174</i>	<i>196</i>	<i>470</i>
<i>Total</i>	<i>1005</i>	<i>978</i>	<i>1983</i>

**TABLE 5.1. DATASET DETAILS**

The datasets is given as the input which is the comments from famous music videos. They contain YouTube ID, comment author, date, comment content, and labeled class (0: Ham or 1: Spam).Comment content and labeled class are only used. This dataset is used to train the model.

COMMENT_ID	AUTHOR	DATE	CONTENT	CLASS
LZQPQhLyRh80UYxN	Julius NM	2013-11-07T06:20:48	Huh, anyway check	1
uaDWhIGQYNQ96lu			out this you[tube]	
Cg-AYWqNPjpU			channel: ...	
LZQPQhLyRh_C2cTt	adam riyati	2013-11-07T12:37:15	Hey guys check out	1
d9MvFRJedxydaVW-			my new channel	
2sNg5Diuo4A			and our firs...	

**FIG NO. 5.2 STRUCTURE OF THE DATASET**

The datasets are text-based, hence pre-processing is used before applying machine learning. Using the PortStemmer function of the nltk library, stop words are removed and tokens are listed with comments. Finally, we utilise TF-IDF vectorization and count the frequency of token occurrence.

### **TF-IDF VECTORIZER:**

The Count Vectorizer gives a basic method to both tokenize an assortment of content archives and fabricate a jargon of known words, yet additionally to encode new reports utilizing that jargon. Tfidf-Vectorizer : (Term Frequency \* Inverse Document Frequency). 1.Term Frequency: The number of times a word appears in a document divided by the total number of words in the document. Every document has its own term frequency. 2. Inverse Document Frequency: The log of the number of documents divided by the number of documents that contain the word  $w$ . Inverse data frequency determines the weight of rare words across all documents in the corpus. The procedure to calculate TF and IDF is:



$$tf-idf(t, d) = tf(t, d) * idf(t)$$

Where

*tf* is the term frequency

*idf* is the inverse document frequency

tf and idf can be calculated as follows

$$TF_{i,j} = n_{i,j} / \sum_k n_{i,j}$$

$$IDF = \log(N/n)$$

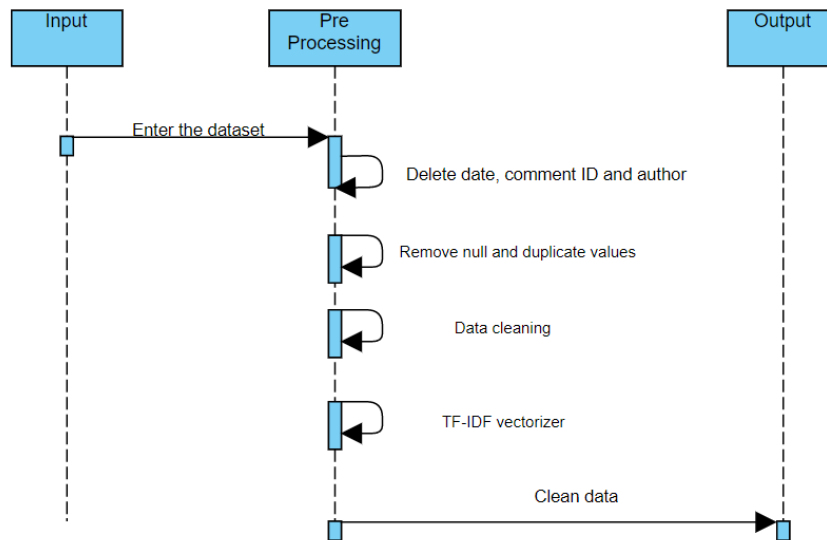
Where

N is the number of archives

n is the number of archives a term t has appeared in word.

TF-IDF tries to highlight words that are more important (appear more frequently) in a particular document using word frequency. The TD-IDF Vectorizer will convert the documents into tokens, learn the vocabulary and inverse document frequency weightings, and allow you to encode new documents.

## MODULE DIAGRAM



**FIG NO. 5.3. MODULE 1 SEQUENCE DIAGRAM**

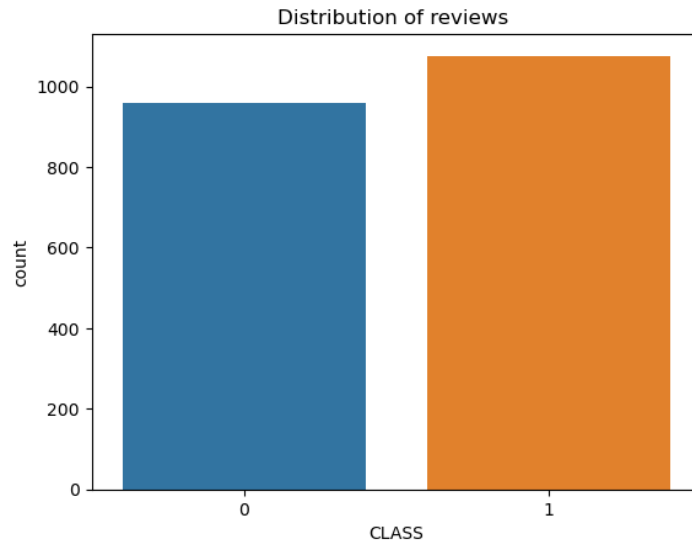
## 2. DATA ANALYSIS

The graphical depiction of information and data in pictorial or graphical formats, such as charts, graphs, and maps, is known as data visualization. Tools for data visualization offer a simple approach to spot and comprehend trends, data patterns, and outliers. Tools and methods for data visualization are crucial for processing vast volumes of data and making data-driven choices. The idea of utilizing visuals to comprehend data has been around for a very long time. Charts, tables, graphs, maps, and dashboards are the standard kinds of data visualization. Finding data patterns is data visualization's most crucial accomplishment. For all, when all the data is presented to user in a visual format rather than a table, it is much simpler to spot data trends. Data visualization puts data into perspective by illustrating its significance in relation to other factors. It

shows where certain data references stand in relation to the bigger picture. A data story may also be told to audiences through data visualization. The visualization may be used to convey a tale and guide the audience to an obvious conclusion while presenting the data facts in an understandable format.

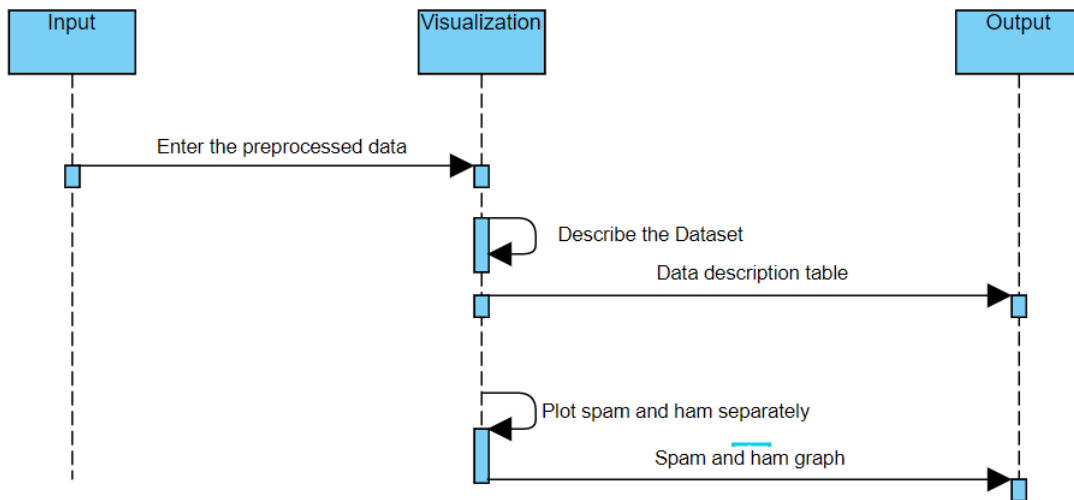
Data visualization is an important skill in applied statistics and machine learning. Statistics does indeed focus on quantitative descriptions and estimations of data. Data visualization provides an important suite of tools for gaining a qualitative understanding. This can be helpful when exploring and getting to know a dataset and can help with identifying patterns, corrupt data, outliers, and much more. With a little domain knowledge, data visualizations can be used to express and demonstrate key relationships in plots and charts that are more visceral and stakeholders than measures of association or significance. Data visualization and exploratory data analysis are whole fields themselves and it will recommend a deeper dive into some the books mentioned at the end.

Sometimes data does not make sense until it can look at in a visual form, such as with charts and plots. Being able to quickly visualize of data samples and others is an important skill both in applied statistics and in applied machine learning. It will discover the many types of plots that you will need to know when visualizing data in Python and how to use them to better understand your own data.



**FIG NO. 5.4 DATA VISUALIZATION**

## MODULE DIAGRAM



**FIG NO. 5.5 MODULE 2 SEQUENCE DIAGRAM**

### **3. CLASSIFICATIONS**

It is important to compare the performance of multiple different machine learning algorithms consistently and it will discover to create a test harness to compare multiple different machine learning algorithms in Python with scikit-learn. It can use this test harness as a template on your own machine learning problems and add more and different algorithms to compare. Each model will have different performance characteristics. Using resampling methods like cross validation, you can get an estimate for how accurate each model may be on unseen data. It needs to be able to use these estimates to choose one or two best models from the suite of models that you have created. When have a new dataset, it is a good idea to visualize the data using different techniques in order to look at the data from different perspectives. The same idea applies to model selection. You should use a number of different ways of looking at the estimated accuracy of your machine learning algorithms in order to choose the one or two to finalize. A way to do this is to use different visualization methods to show the average accuracy, variance and other properties of the distribution of model accuracies.

The key to a fair comparison of machine learning algorithms is ensuring that each algorithm is evaluated in the same way on the same data and it can achieve this by forcing each algorithm to be evaluated on a consistent test harness.

#### **SPLIT THE DATA:**

Splitting the data is the most essential step in machine learning. The procedure involves taking a dataset and dividing it into two subsets. The first subset is used to fit the model and is referred to as the training dataset. The second subset is not used to train the model; instead, the input element of the dataset is provided to the model, then predictions are made and compared to the expected values. This second dataset

is referred to as the test dataset. We train our model on the trainset and test our data on the testing set. We split our data in train and test using the `train_test_split` function from Scikit learn. The idea of “sufficiently large” is specific to each predictive modelling problem. It means that there is enough data to split the dataset into train and test datasets and each of the train and test datasets are suitable representations of the problem domain. Conversely, the train-test procedure is not appropriate when the dataset available is small. The reason is that when the dataset is split into train and test sets, there will not be enough data in the training dataset for the model to learn an effective mapping of inputs to outputs. There will also not be enough data in the test set to effectively evaluate the model performance. The estimated performance could be overly optimistic (good) or overly pessimistic (bad).

There is no optimal split percentage. A split percentage is chosen that meets the project’s objectives with considerations include, Computational cost in training the model. Computational cost in evaluating the model. Training set representativeness. Test set representativeness. Nevertheless, the split of the dataset is taken as Train=70% and Test=30% for Logistic Regression, Train=60 and Test=40 for Random Forest Classifier and Train=80% and Test=20% for Multinomial Naïve Bayes Classifier.

### **Performance Metrics to calculate:**

**False Positives (FP):** A person who will pay predicted as defaulter. When actual class is no and predicted class is yes. E.g. if actual class says this passenger did not survive but predicted class tells you that this passenger will survive.

**False Negatives (FN):** A person who default predicted as payer. When actual class is yes but predicted class in no. E.g. if actual class value indicates that this passenger survived and predicted class tells you that passenger will die.

**True Positives (TP):** A person who will not pay predicted as defaulter. These are the correctly predicted positive values which means that the value of actual class is yes and the value of predicted class is also yes. E.g. if actual class value indicates that this passenger survived and predicted class tells you the same thing.

**True Negatives (TN):** A person who default predicted as payer. These are the correctly predicted negative values which means that the value of actual class is no and value of predicted class is also no. E.g. if actual class says this passenger did not survive and predicted class tells you the same thing.

$$\text{True Positive Rate}(TPR) = TP / (TP + FN)$$

$$\text{False Positive rate}(FPR) = FP / (FP + TN)$$

**Accuracy:** The Proportion of the total number of predictions that is correct otherwise overall how often the model predicts correctly defaulters and non-defaulters.

**Accuracy calculation:**

$$\text{Accuracy} = (TP + TN) / (TP + TN + FP + FN)$$

Accuracy is the most intuitive performance measure and it is simply a ratio of correctly predicted observation to the total observations. One may think that, if we have high accuracy then our model is best. Yes, accuracy is a great measure but only when you have symmetric datasets where values of false positive and false negatives are almost same.

Precision: The proportion of positive predictions that are actually correct.

$$\textit{Precision} = TP / (TP + FP)$$

Precision is the ratio of correctly predicted positive observations to the total predicted positive observations. The question that this metric answer is of all passengers that labelled as survived, how many actually survived? High precision relates to the low false positive rate. We have got 0.788 precision which is pretty good.

**Recall:** The proportion of positive observed values correctly predicted. (The proportion of actual defaulters that the model will correctly predict)

$$\textit{Recall} = TP / (TP + FN)$$

Recall(Sensitivity) - Recall is the ratio of correctly predicted positive observations to the all observations in actual class - yes.

**F1 Score** is the weighted average of Precision and Recall. Therefore, this score takes both false positives and false negatives into account. Intuitively it is not as easy to understand as accuracy, but F1 is usually more useful than accuracy, especially if you have an uneven class distribution. Accuracy works best if false positives and false negatives have similar cost. If the cost of false positives and false negatives are very different, it's better to look at both Precision and Recall.



$$F1\ Score = 2*(Recall * Precision) / (Recall + Precision)$$

The below different algorithms are compared:

1. Logistic Regression
2. Random Forest Classifier
3. Multinomial Naïve Bayes

## 1. LOGISTIC REGRESSION

True Positive (TP) = 209

True Negative (TN) = 8

False Positive (FP) = 28

False Negative (FN) = 187

True Positive Rate(TPR) = TP / (TP + FN) = 209/(209+187)=0.52777

False Positive rate(FPR) = FP / (FP + TN) =28/(28+8)=0.77777

Accuracy = (TP + TN) / (TP + TN + FP + FN) = (209+8)/(209+8+28+187)  
= 0.91

Precision = TP/ (TP+FP) = 209/(209+28)= 0.88

Recall = TP/(TP+FN) = 209/(209+187) = 0.96

F1 Score = 2\*(Recall \* Precision) / (Recall + Precision)

=2\*(0.96\*0.88)/(0.96+0.88)

=0.92



**FIG NO. 5.6 CLASSIFICATION OF LR**

## **2. RANDOM FOREST CLASSIFIER**

True Positive (TP) = 596

True Negative (TN) = 0

False Positive (FP) = 0

False Negative (FN) = 596

True Positive Rate(TPR) =  $TP / (TP + FN) = 596 / (596 + 596) = 0.5$

False Positive rate(FPR) =  $FP / (FP + TN) = 0 / (0 + 0) = 0$

Accuracy =  $(TP + TN) / (TP + TN + FP + FN) = (596 + 0) / (596 + 0 + 0 + 0) = 1.0$

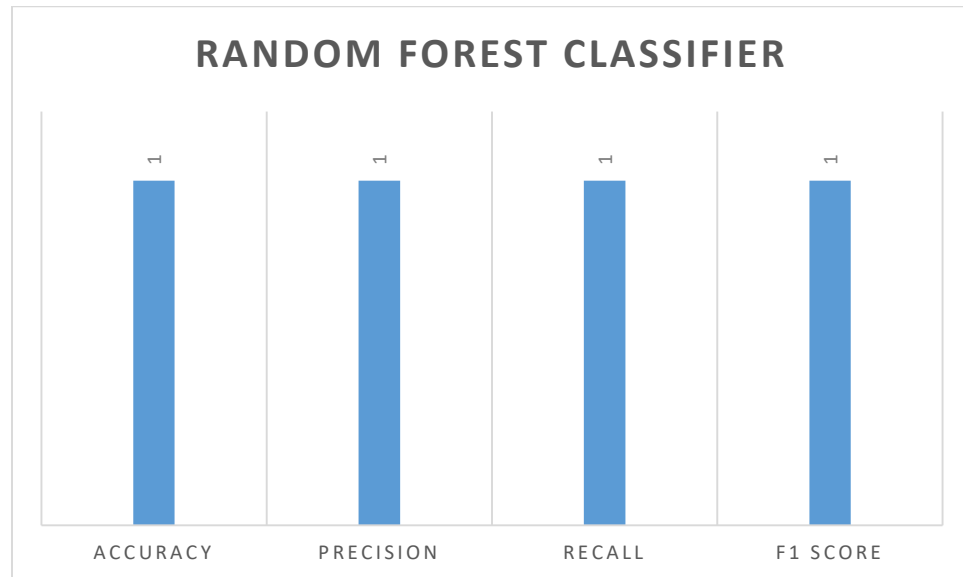
Precision =  $TP / (TP + FP) = 596 / (596 + 0) = 1.0$

Recall =  $TP / (TP + FN) = 596 / (596 + 0) = 1.0$

$$\text{F1 Score} = 2 * (\text{Recall} * \text{Precision}) / (\text{Recall} + \text{Precision})$$

$$= 2 * (1.0 * 1.0) / (1.0 + 1.0)$$

$$= 1.0$$



**FIG NO. 5.7 CLASSIFICATION OF RF**

### 3. MULTINOMIAL NAÏVE BAYES

$$\text{True Positive (TP)} = 292$$

$$\text{True Negative (TN)} = 13$$

$$\text{False Positive (FP)} = 0$$

$$\text{False Negative (FN)} = 271$$

$$\text{True Positive Rate (TPR)} = \text{TP} / (\text{TP} + \text{FN}) = 292 / (292 + 271) = 0.51$$

$$\text{False Positive rate (FPR)} = \text{FP} / (\text{FP} + \text{TN}) = 0 / (0 + 13) = 0$$

$$\text{Accuracy} = (\text{TP} + \text{TN}) / (\text{TP} + \text{TN} + \text{FP} + \text{FN}) = (292+13)/(292+13+0+271) \\ = 0.97$$

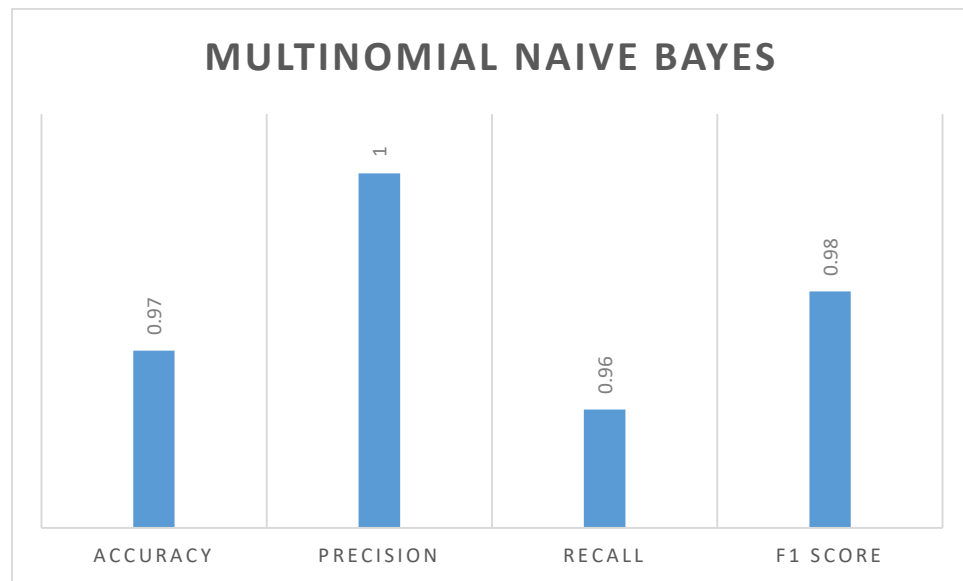
$$\text{Precision} = \text{TP} / (\text{TP} + \text{FP}) = 292 / (292 + 0) = 1.0$$

$$\text{Recall} = \text{TP} / (\text{TP} + \text{FN}) = 292 / (292 + 271) = 0.96$$

$$\text{F1 Score} = 2 * (\text{Recall} * \text{Precision}) / (\text{Recall} + \text{Precision})$$

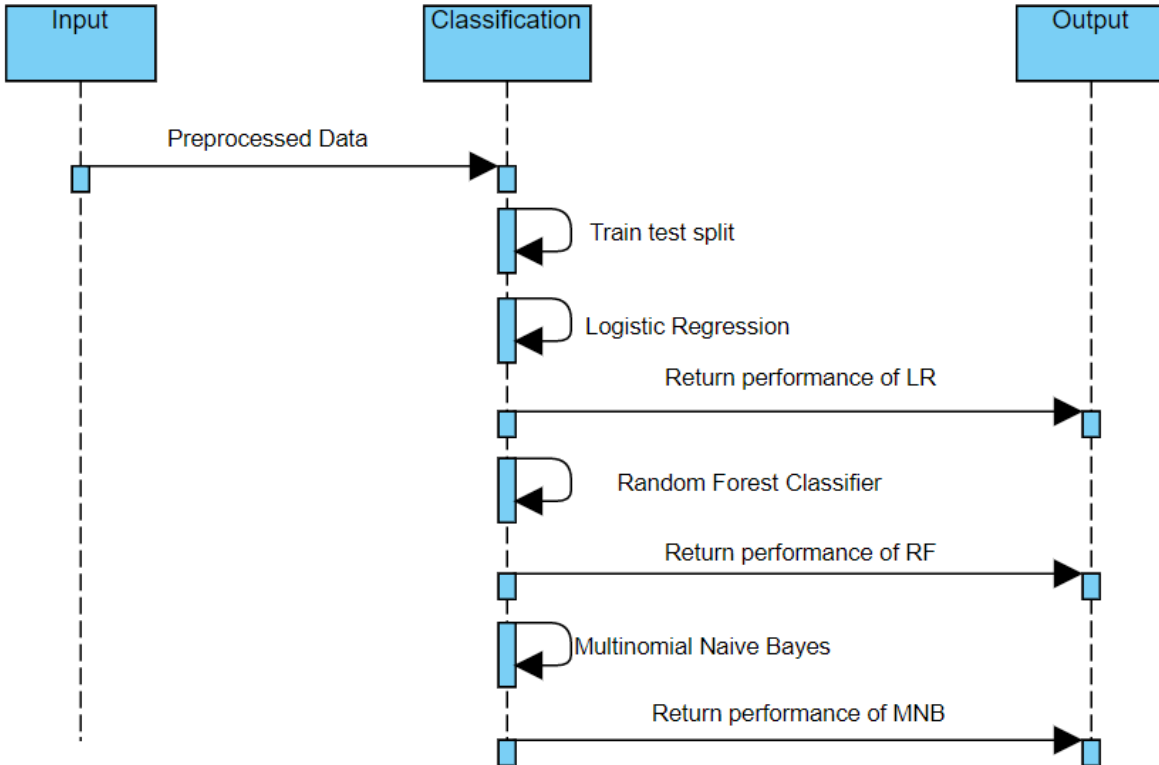
$$= 2 * (0.96 * 1.0) / (0.96 + 1.0)$$

$$= 0.98$$



**FIG NO. 5.8 CLASSIFICATION OF MNB**

## MODULE DIAGRAM



**FIG NO. 5.9 MODULE 3 SEQUENCE DIAGRAM**

## 4. ANALYSIS PERFORMANCE

Five metrics are employed for assessment, including Acc (Accuracy rate), precision, recall, f1-score, and support, with three methods (Logistic Regression, Multinomial Naive Bayes, and Random Forest) being used. Because of this, the ESM-H model performed best in terms of accuracy, precision recall, f1-score, and support. Thus, the cascaded ensemble model approach is used. From the

Ensemble Hard voting performed it is concluded that Random Forest has the best results. Hence it is chosen for deployment.

MEASURES	LR	RF	MNB
ACCURACY	0.91	1	0.97
PRECISION	0.88	1	1
RECALL	0.96	1	0.96
F1-SCORE	0.92	1	0.98

TABLE 5.2. PERFORMANCE OF THE ALGORITHMS

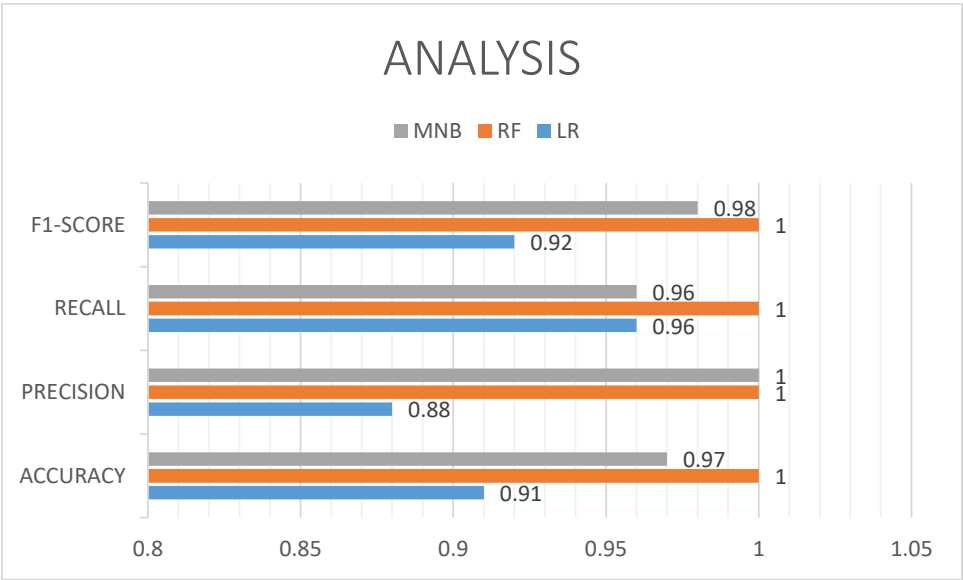
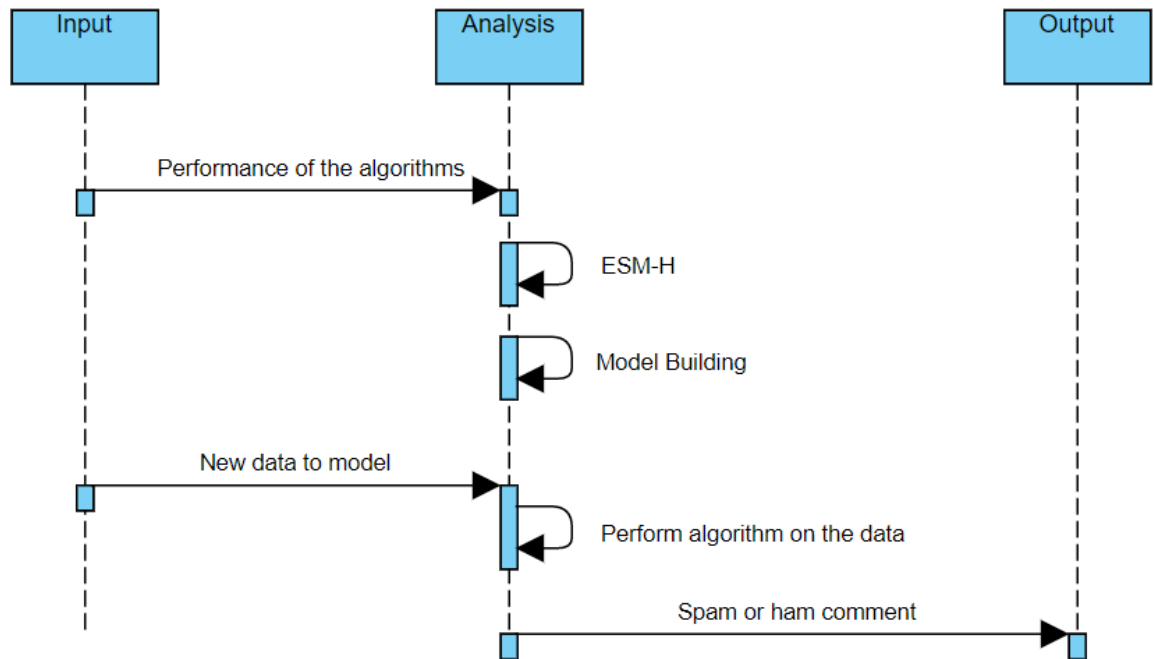


FIG NO. 5.10 ESM-H

## MODULE DIAGRAM



**FIG NO. 5.11 MODULE 4 SEQUENCE DIAGRAM**

## 5.2. ALGORITHMS

There are three algorithms used in this method. They are:

1. Logistic Regression
2. Random Forest Classifier
3. Multinomial Naïve Bayes Classifier

### 1. LOGISTIC REGRESSION

Logistic regression is a member of the supervised machine learning framework in the domain of artificial intelligence. It is also regarded as a

discriminative model since it makes an effort to discriminate between different classes (or categories). Unlike a generative algorithm, like naive bayes, it is unable to produce information of the class that it is attempting to predict, such as a picture. For obtaining the beta coefficients for the model, logistic regression maximises the log likelihood function. When seen in the perspective of machine learning, this modifies slightly. In machine learning, the loss function is the negative log likelihood, and the global maximum is found using gradient descent. Moreover, logistic regression is susceptible to overfitting, especially when the model contains a large number of predictor variables. When a model has high dimensionality, regularisation is often employed to penalise big coefficients in the parameters.

Logistic regression is commonly used for prediction and classification problems. It is used to classify based on 1 and 0. One of the most widely utilized Machine Learning algorithms, within the category of Supervised Learning, is logistic regression. With a predetermined set of independent factors, it is used to predict the categorical dependent variable. With a categorical dependent variable, the output is predicted via logistic regression. As a result, the result must be a discrete or categorical value. Rather of providing the precise values of 0 and 1, it provides the probabilistic values that fall between 0 and 1. It can be either Yes or No, 0 or 1, true or false, etc. With the exception of how they are applied, logistic regression and linear regression are very similar. Whereas logistic regression is used to solve classification difficulties, linear regression is used to solve regression problems.

In Logistic regression, instead of fitting a regression line, we fit a "S" shaped logistic function, which predicts two maximum values (0 or 1). The



logistic function's curve demonstrates the likelihood of several things, like whether or not the cells are malignant, etc. Since it can classify new data using both continuous and discrete datasets, logistic regression is a key machine learning approach.

The linear regression equation yields the logistic regression equation. The following are the mathematical steps to get Logistic Regression equations:

A straight line's equation may be expressed as:

$$y_0 = b_0 + b_1x_1 + b_2x_2 + \dots + b_nx_n$$

In Logistic Regression,  $y$  can only range between 0 and 1, thus multiply the following equation by  $(1-y)$  to account for this:

$$y/(1-y); 0 \text{ for } y=0 \text{ and infinity for } y=1$$

However, we want a range between  $-\text{[infinity]}$  and  $+\text{[infinity]}$ . If we take the equation's logarithm, it becomes:

$$\log[y/(1-y)] = b_0 + b_1x_1 + b_2x_2 + b_3x_3 + \dots + b_nx_n$$

where:

$\log$  is the logarithm function

$y$  is the probability of “success”

$b_0$  is the  $y$  intercept

$b_n$  is the first order logistic regression coefficient of the  $n$ th predictor

$x_n$  is the value of the  $n$ th predictor

## **2. RANDOM FOREST CLASSIFIER**

One of the most well-liked and often applied algorithms among data scientists is random forest. Supervised machine learning algorithms like random forest are frequently employed in classification and regression issues. Using various samples, it constructs decision trees and uses their average for classification and majority vote for regression. The Random Forest Algorithm's ability to handle data sets with continuous variables, as in regression, and categorical variables, as in classification, is one of its most crucial qualities. For classification and regression tasks, it performs better. Some decision trees may predict the proper output, while others may not, since the random forest mixes numerous trees to forecast the class of the dataset. Yet when all the trees are combined, they forecast the right result. Hence, the following two presumptions for an improved Random forest classifier: In order for the classifier to predict accurate results rather than an assumed outcome, there should be some real values in the feature variable of the dataset. The predictions from each tree must have very low correlations.

The Random Forest Algorithm consists of many decision trees with the same nodes in each, but different leaves based on different input. To get an answer that represents the average of all these decision trees, it combines the outcomes of several decision trees. A decision tree is the name given to the procedure in the field of machine learning. The process begins with a node, which branches to another node, and so on until one reaches a leaf. In order to assist in classifying the data, a node poses a query. A branch symbolises the various directions that this node might lead to. A decision tree's leaf is the point at which there are no more branches, or a node. The random forest

method employs labelled data to learn how to categorise unlabeled data; it is an example of supervised learning. Engineers frequently utilise the Random Forest Algorithm because it may be used to address classification and regression difficulties.

In order to determine how the data branches from each node when utilising the Random Forest Algorithm to solve regression issues, the mean squared error (MSE) is used.

$$MSE = 1/N \sum_{i=1}^N (f_i - y_i)^2$$

*Where*

*N is the number of data points*

*f<sub>i</sub> is the value returned by the model and*

*y<sub>i</sub> is the actual value for data point i.*

To determine which branch is the best choice for the forest, this algorithm estimates the distance between each node and the expected actual value. In this case, f<sub>i</sub> is the value the decision tree returned, and y<sub>i</sub> is the value of the data point you are testing at a particular node.

When performing Random Forests based on classification data, one should know that they are often using the Gini index, or the formula used to decide how nodes on a decision tree branch.

$$Gini = 1 - \sum_{i=1}^C (p_i)^2$$

*Here,*

*p<sub>i</sub> stands for the class's observed relative frequency in the dataset,*

*C for the total number of classes.*

This formula calculates the Gini of each branch on a node based on the class and probability, indicating which branch is more likely to occur. Here,  $p_i$  represents the relative frequency of the class that is being observed in the dataset and  $c$  represents the number of classes.

Entropy may also be used to predict the branching patterns of decision tree nodes.

$$\text{Entropy} = 1 - \sum_{i=1}^C (p_i) * \log_2(p_i)$$

*Where,*

*Log is the logarithmic function*

*$p_i$  stands for the class's observed relative frequency in the dataset,*

*$C$  for the total number of classes.*

Entropy uses the likelihood of a particular outcome to determine which branch the node should take. It is more mathematically complex than the Gini index since a logarithmic function is utilised to calculate it.

The formula for Random Forest classifier is

$$RFf_i = (\sum_{j \in \text{all trees}} \text{norm}f_{ij}) / T$$

Where:

$RFf_i$  sub(i) = the importance of feature  $i$  calculated from all trees in the Random

Forest model

$\text{normfi sub}(ij)$  = the normalized feature importance for  $i$  in tree  $j$

$T$  = total number of trees

### 3. MULTINOMIAL NAÏVE BAYES CLASSIFIER

Popular in Natural Language Processing is the Bayesian learning method known as the Multinomial Naive Bayes algorithm (NLP). Using the Bayes principle, the algorithm makes an accurate prediction about the tag of a text, such as an email or news article. It calculates the probability of each tag for a certain sample and produces the tag with the highest probability. Each feature being classified by the Naive Bayes classifier is distinct from every other feature, which unites the several algorithms that make up the classifier. The inclusion or removal of one characteristic does not depend on the presence or absence of another feature. Data that cannot be represented numerically can be classified using the multinomial model. The vastly decreased complexity is its key benefit. It gives the capacity to carry out classification tasks with less training data and without having to constantly retrain.

Popular in Natural Language Processing is the Bayesian learning method known as the Multinomial Naive Bayes algorithm (NLP). The Bayes theorem is used by the software to infer the tag of a text, such as an email or a news article. It determines the likelihood of each tag for a certain sample and produces the tag with the highest likelihood. Each feature being classified by the Naive Bayes classifier is distinct from every other feature, which unites the several algorithms that make up the classifier. The presence or absence of

a feature has no influence on whether another feature is included or not. It is simple to put in place because all that is required is the calculation of probability. Both continuous and discrete data can be used in this strategy. Real-time applications may be predicted using this simple method. Large datasets may be handled with ease and it is very scalable. The likelihood of one feature occurring has no bearing on the likelihood of the other feature occurring. The Naive Bayes method is an effective technique for examining text input and resolving issues involving several classes. It is vital to first understand the Bayes theorem concept since the Naive Bayes theorem is built on it. The Bayes theorem, which was created by Thomas Bayes, calculates the probability of an event happening based on information about its circumstances. We determine the chance of class A when predictor B is provided. Naive Bayes can outperform the most potent alternatives for small sample sets. It is utilised in many different sectors since it is quite durable, simple to use, quick, and accurate.

The formula of Multinomial Naïve Bayes algorithm can be written as:

$$P(\alpha/\beta) = P(\alpha) * P(\beta / \alpha)/P(\beta)$$

Where:

$P(\beta)$  = prior probability of  $\beta$

$P(\alpha)$  = prior probability of class  $\alpha$

$P(\beta | \alpha)$  = occurrence of predictor  $\beta$  given class  $\alpha$  probability

## **6. SYSTEM IMPLEMENTATION**

## 6.1 CLIENT SIDE CODING

### home.html

```
<!DOCTYPE html>

<html>

<head>

    <title>Home</title>

    <link rel="stylesheet" type="text/css" href="{{ url_for('static',
filename='css/bootstrap.min.css') }}">

    <style>

        .back{

            background-image: url('{{ url_for('static',
filename='image/download.jpg') }}');

            background-repeat: no-repeat;

            background-attachment: fixed;

            background-size: 100% 100%;

        }

        .white{

            color:white;

        }
```



```
.space{  
  
margin:10px 30px;  
  
padding:10px 10px;  
  
background: lightblue;  
  
width:500px  
  
}  
  
.gap{  
  
padding:10px 20px;  
  
}  
  
#text{  
  
    box-sizing: border-box;  
  
    height: 100px;  
  
    width: -20%;  
  
}  
  
</style>
```

```
</head>
```

```
<body class="back">
```

```
<header class="jumbotron" style="height:100px;">
```

```
<div class="container">
```

<center>

<h1>YOUTUBE SPAM OR HAM</h1>

</center>

</div>

</header>

<center>

<div class="card ml-container" style="width:40%" >

```
<form class="form-group" action="{ { url_for('predict') } }"
method="POST">

<label class="black" for="">Enter The Description Here</label>

<!-- <input type="text" name="comment"/> -->

<textarea name="message" class="space form-control" rows="0"
cols="" id="text">

</textarea>

<br/>
```

```
        <input type="submit" class="btn btn-success btn-block"
style="width:350px;padding:20px" value="Predict">
```

```
    </form>
```

```
</div>
```

```
</center>
```

```
</body>
```

```
</html>
```

### **result.html**

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
    <title></title>
```

```
    <link rel="stylesheet" type="text/css" href="{{ url_for('static',
filename='css/bootstrap.min.css') }}">
```

```
    <style>
```

```
        .back{
```

```
background-image: url("{ { url_for('static',  
filename='image/images.jpg') } }");
```

```
background-repeat: no-repeat;
```

```
background-attachment: fixed;
```

```
background-size: 100% 100%;
```

```
}
```

```
center{
```

```
padding-top:10%;
```

```
}
```

```
a{
```

```
color:red;
```

```
}
```

```
</style>
```

```
</head>
```

```
<body class="back">
```

```
<header class="jumbotron" style="height: 100px;">
```

```
<div class="container">
```

```
<h2 style="text-align:center">Youtube Spam or Ham</h2>
```

</div>

</header>

<center>

<div class="card" style="width:30%">

<p style="color:red;font-size:20;text-align: center;"><b>Your  
Result</b></p>

<div class="results">

{% if prediction == 0 % }

<h2 style="color:rgb(233, 3, 33);">Ham</h2>

{% elif prediction == 1 % }

<h2 style="color:red;">SPAM</h2>

{% endif % }

</div>

</center>

</div>

<a href="{{ url\_for('home')}}">Go back</a>

</body>

</html>

## 6.2 SERVER SIDE CODING

### Flask.ipynb

```
from flask import Flask,render_template,url_for,request
```

```
import pandas as pd
```

```
import joblib
```

```
# load the model from disk
```

```
clf = joblib.load("LogRe.pkl")
```

```
cv = joblib.load("LogReTV.pkl")
```

```
app = Flask(__name__)
```

```

@app.route('/')

def home():

    return render_template('home.html')


@app.route('/predict',methods=['POST'])

def predict():


    if request.method == 'POST':

        message = request.form['message']

        data = [message]

        print(data)

        vect = cv.transform(data).toarray()

        my_prediction = clf.predict(vect)

        print(my_prediction)

    return render_template('result.html',prediction = my_prediction)


if __name__ == '__main__':

    app.run(debug=False,port=7000,host='localhost')

```

## M1.ipynb

```
#!/usr/bin/env python
```

```
# coding: utf-8
```

```
# In[1]:
```

```
import warnings
```

```
warnings.filterwarnings('ignore')
```

```
# In[2]:
```

```
import pandas as pd
```

```
# In[3]:
```

```
psy = pd.read_csv('Youtube01-Psy.csv')
```

```
katty = pd.read_csv('Youtube02-KatyPerry.csv')
```

```
Lmfao = pd.read_csv('Youtube03-LMFAO.csv')
```

```
Eminem = pd.read_csv('Youtube04-Eminem.csv')
```

```
shakira = pd.read_csv('Youtube04-Eminem.csv')
```

```
# In[4]:
```

```
data=pd.concat([psy,katty,Lmfao,Eminem,shakira])
```

```
# In[5]:
```

```
data.shape
```

```
# In[6]:
```



```
data.head(10)
```

```
# In[7]:
```

```
del data['DATE']
```

```
# In[8]:
```

```
data
```

```
# In[9]:
```

```
data.isnull().sum()
```

```
# In[10]:
```

```
data.duplicated()
```

```
# In[11]:
```

```
data.duplicated().sum()
```

```
# In[12]:
```

```
data=data.drop_duplicates()
```

```
# In[13]:
```

```
data.shape
```

```
# In[14]:
```

```
data.columns
```

```
# In[15]:
```

```
del data['COMMENT_ID']  
del data['AUTHOR']
```

```
# In[16]:
```

```
data
```

```
# In[17]:
```

```
data['CLASS'].value_counts()
```

```
# In[18]:
```

```
data.reset_index(inplace=True)
```

```
# In[19]:
```

```
import re  
from nltk.corpus import stopwords  
from nltk.stem.porter import PorterStemmer  
ps=PorterStemmer()
```

```
corpus=[]  
for i in range(0, len(data)):
```

```
review=re.sub('[a-zA-Z][0-9]', ' ', str(data['CONTENT'][i]))
review=review.lower()
review=review.split()
```

```
review=[ps.stem(word) for word in review if not word in
stopwords.words('english')]
review=' '.join(review)
corpus.append(review)
```

```
corpus
```

```
# In[20]:
```

```
df=data
```

```
# In[21]:
```

```
type(df['CONTENT'].loc[100])
```

```
# In[22]:
```

```
df.info()
```

```
# In[23]:
```

```
# Creating the TFIDF model
from sklearn.feature_extraction.text import TfidfVectorizer
tv=TfidfVectorizer(max_features=1500,ngram_range=(1,2))
X=tv.fit_transform(corpus).toarray()
```

```
# In[28]:
```

X

```
# In[25]:
```

```
X.shape
```

```
# In[26]:
```

```
y=pd.get_dummies(df['CLASS'])  
y=y.iloc[:,1].values
```

```
# In[27]:
```

y

## **M2.ipynb**

```
#!/usr/bin/env python  
# coding: utf-8
```

```
# # EDA of visualization and training a model by given attributes
```

```
# In[1]:
```

```
#import library packages  
import pandas as pd  
import matplotlib.pyplot as plt  
import seaborn as sns  
import numpy as np
```

```
# In[2]:
```

```
import warnings
```

```
warnings.filterwarnings("ignore")
```

```
# In[3]:
```

```
psy = pd.read_csv('Youtube01-Psy.csv')  
katty = pd.read_csv('Youtube02-KatyPerry.csv')  
Lmfao = pd.read_csv('Youtube03-LMFAO.csv')  
Eminem = pd.read_csv('Youtube04-Eminem.csv')  
shakira = pd.read_csv('Youtube04-Eminem.csv')
```

```
# In[4]:
```

```
data=pd.concat([psy,katty,Lmfao,Eminem,shakira])
```

```
# In[5]:
```

```
del data['DATE']  
del data['COMMENT_ID']  
del data['AUTHOR']
```

```
# In[6]:
```

```
data
```

```
# In[7]:
```

```
data.columns
```

```
# In[8]:
```

```
data.groupby('CONTENT').describe()
```

```
# In[9]:
```

```
#plotting graph for distribution
sns.countplot(x = "CLASS", data = data)
data.loc[:, 'CLASS'].value_counts()
plt.title('Distribution of reviews ')
```

```
# In[10]:
```

```
data['CLASS'].unique()
```

```
# Training model:
```

```
# In[11]:
```

```
#!pip install nltk
```

```
# In[12]:
```

```
import nltk
nltk.download('stopwords')
df=data
```

```
# In[13]:
```

```
# divide the set in training and test
from sklearn.model_selection import train_test_split
X,X_test,y,y_test =
train_test_split(df.loc[:, 'CONTENT'],df['CLASS'],test_size=0.2)
```

```
# In[14]:
```

```
X.info()
```

```
# In[15]:
```

```
from wordcloud import WordCloud
```

```
positive=' '.join(X.loc[y==0,'CONTENT'].values)
```

```
ham_text = WordCloud(background_color='white',max_words=2000,width = 800,  
height = 800).generate(positive)
```

```
negative=' '.join(X.loc[y==1,'CONTENT'].values)
```

```
spam_text = WordCloud(background_color='black',max_words=2000,width = 800,  
height = 800).generate(negative)
```

```
plt.figure(figsize=[30,50])
```

```
plt.subplot(1,3,1)
```

```
plt.imshow(ham_text,interpolation='bilinear')
```

```
plt.title("")
```

```
plt.axis('off')
```

```
plt.subplot(1,3,2)
```

```
plt.imshow(spam_text, interpolation='bilinear')
```

```
plt.axis('off')
```

```
plt.title("")
```

### M3.ipynb

```
#!/usr/bin/env python
# coding: utf-8
```

```
# # Logistic regression
```

```
# In[1]:
```

```
#import library packages
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np
```

```
# In[2]:
```

```
import warnings
warnings.filterwarnings("ignore")
```

```
# In[3]:
```

```
psy = pd.read_csv('Youtube01-Psy.csv')
katty = pd.read_csv('Youtube02-KatyPerry.csv')
Lmfao = pd.read_csv('Youtube03-LMFAO.csv')
Eminem = pd.read_csv('Youtube04-Eminem.csv')
shakira = pd.read_csv('Youtube04-Eminem.csv')
```

```
# In[4]:
```

```
psy.shape, katty.shape, Lmfao.shape, Eminem.shape, shakira.shape
```

```
# In[5]:
```



```
data=pd.concat([psy,katty,Lmfao,Eminem,shakira])
```

```
# In[6]:
```

```
data.shape
```

```
# In[7]:
```

```
data
```

```
# In[8]:
```

```
del data['DATE']  
del data['COMMENT_ID']  
del data['AUTHOR']
```

```
# In[9]:
```

```
data
```

```
# In[10]:
```

```
data['CLASS'].value_counts()
```

```
# In[11]:
```

```
data.duplicated()
```

```
# In[12]:
```

```
data.duplicated().sum()
```

```
# In[13]:
```

```
data=data.drop_duplicates()
```

```
# In[14]:
```

```
data['CLASS'].value_counts()
```

```
# In[15]:
```

```
data.shape
```

```
# In[16]:
```

```
data.isnull().sum()
```

```
# In[17]:
```

```
data.reset_index(inplace=True)
```

```
# In[18]:
```

```
import re
from nltk.corpus import stopwords
from nltk.stem.porter import PorterStemmer
ps=PorterStemmer()
```

```

corpus=[]
for i in range(0, len(data)):
    review=re.sub('[a-zA-Z][0-9]', ' ', str(data['CONTENT'][i]))
    review=review.lower()
    review=review.split()

    review=[ps.stem(word) for word in review if not word in
stopwords.words('english')]
    review=' '.join(review)
    corpus.append(review)

```

```
corpus
```

```
# In[19]:
```

```
df=data
```

```
# In[20]:
```

```
type(df['CONTENT'].loc[100])
```

```
# In[21]:
```

```
df.info()
```

```
# In[22]:
```

```

# Creating the TFIDF model
from sklearn.feature_extraction.text import TfidfVectorizer
tv=TfidfVectorizer(max_features=1500,ngram_range=(1,2))
X=tv.fit_transform(corpus).toarray()

```

```
# In[23]:
```

```
X
```

```
# In[24]:
```

```
X.shape
```

```
# In[25]:
```

```
y=pd.get_dummies(df['CLASS'])  
y=y.iloc[:,1].values
```

```
# In[26]:
```

```
y
```

```
# In[ ]:
```

```
# In[27]:
```

```
## Since data is imbalanced  
## Trying over sampling
```

```
# from imblearn.over_sampling import RandomOverSampler
```

```
# rs=RandomOverSampler()  
# X,y=rs.fit_resample(X,y)
```

```
# X.shape,y.shape
```

```
# In[28]:
```

```
# Train Test Split
```

```
from sklearn.model_selection import train_test_split  
X_train,X_test,y_train,y_test=train_test_split(X,y,test_size=0.3,random_state=42)
```

```
# In[29]:
```

```
from sklearn.linear_model import LogisticRegression  
lr = LogisticRegression()  
lr.fit(X_train,y_train)
```

```
# In[30]:
```

```
predict = lr.predict(X_test)
```

```
# In[31]:
```

```
from sklearn.metrics import accuracy_score  
print('Accuracy of Logistic Regression',accuracy_score(y_test,predict)*100)
```

```
# In[32]:
```

```
from sklearn.metrics import confusion_matrix  
print('Confusion matrix of Logistic  
Regression\n',confusion_matrix(y_test,predict))
```

```
# In[33]:
```

```
from sklearn.metrics import classification_report
print('Classification report of Logistic
Regression\n\n',classification_report(y_test,predict))
```

```
# In[34]:
```

```
import joblib
joblib.dump(lr,'LogRe.pkl')
```

```
# In[35]:
```

```
import joblib
joblib.dump(tv,'LogReTV.pkl')
```

## **M4.ipynb**

```
#!/usr/bin/env python
# coding: utf-8
```

```
# # Random Forest
```

```
# In[1]:
```

```
#import library packages
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np
```

```
# In[2]:
```

```
import warnings
warnings.filterwarnings("ignore")
```

```
# In[3]:
```

```
psy = pd.read_csv('Youtube01-Psy.csv')
katty = pd.read_csv('Youtube02-KatyPerry.csv')
Lmfao = pd.read_csv('Youtube03-LMFAO.csv')
Eminem = pd.read_csv('Youtube04-Eminem.csv')
shakira = pd.read_csv('Youtube04-Eminem.csv')
```

```
# In[4]:
```

```
data=pd.concat([psy,katty,Lmfao,Eminem,shakira])
```

```
# In[5]:
```

```
del data['DATE']
del data['COMMENT_ID']
del data['AUTHOR']
```

```
# In[6]:
```

```
data
```

```
# In[7]:
```

```
data.duplicated()
```

```
# In[8]:
```

```
data.duplicated().sum()
```

```
# In[9]:
```

```
data=data.drop_duplicates()
```

```
# In[10]:
```

```
data.shape
```

```
# In[11]:
```

```
data.isnull().sum()
```

```
# In[12]:
```

```
data.reset_index(inplace=True)
```

```
# In[13]:
```

```
# Data cleaning and preprocessing
```

```
import re  
import nltk  
#nltk.download('stopwords')
```

```
# In[14]:
```

```
from nltk.corpus import stopwords  
from nltk.stem.porter import PorterStemmer
```



```

ps=PorterStemmer()

corpus=[]
for i in range(0, len(data)):
    review=re.sub('[a-zA-Z][0-9]', ' ', str(data['CONTENT'][i]))
    review=review.lower()
    review=review.split()

    review=[ps.stem(word) for word in review if not word in
stopwords.words('english')]
    review=' '.join(review)
    corpus.append(review)

corpus

# In[15]:

df=data

# In[16]:

type(df['CONTENT'].loc[100])

# In[17]:

df.info()

# In[18]:

# Creating the TFIDF model
from sklearn.feature_extraction.text import TfidfVectorizer
tv=TfidfVectorizer(max_features=6000,ngram_range=(1,2))
X=tv.fit_transform(corpus).toarray()

```

```
# In[19]:
```

```
X
```

```
# In[20]:
```

```
X.shape
```

```
# In[21]:
```

```
y=pd.get_dummies(df['CONTENT'])  
y=y.iloc[:,1].values
```

```
# In[22]:
```

```
y
```

```
# In[23]:
```

```
# Since data is imbalanced  
# Trying over sampling
```

```
from imblearn.over_sampling import RandomOverSampler
```

```
rs=RandomOverSampler()  
X,y=rs.fit_resample(X,y)
```

```
X.shape,y.shape
```

```
# In[24]:
```

```
# Train Test Split
```

```
from sklearn.model_selection import train_test_split  
X_train,X_test,y_train,y_test=train_test_split(X,y,test_size=0.4,random_state=0)
```

```
# In[25]:
```

```
from sklearn.ensemble import RandomForestClassifier  
RFC = RandomForestClassifier()  
RFC.fit(X_train,y_train)
```

```
# In[26]:
```

```
predict = RFC.predict(X_test)
```

```
# In[27]:
```

```
from sklearn.metrics import accuracy_score  
print('Accuracy of RandomForestClassifier',accuracy_score(y_test,predict)*100)
```

```
# In[28]:
```

```
from sklearn.metrics import confusion_matrix  
print('Confusion matrix of  
RandomForestClassifier\n',confusion_matrix(y_test,predict))
```

```
# In[29]:
```

```
from sklearn.metrics import classification_report
```

```
print('Classification report of  
RandomForestClassifier\n\n',classification_report(y_test,predict))
```

```
# In[30]:
```

```
import joblib  
joblib.dump(RFC,'rfc.pkl')
```

```
# In[31]:
```

```
import joblib  
joblib.dump(tv,'rfcTV.pkl')
```

## **M5.ipynb**

```
#!/usr/bin/env python  
# coding: utf-8
```

```
# # Multinomial Naive bayes
```

```
# In[1]:
```

```
#import library packages  
import pandas as pd  
import matplotlib.pyplot as plt  
import seaborn as sns  
import numpy as np
```

```
# In[2]:
```

```
import warnings  
warnings.filterwarnings("ignore")
```

```
# In[3]:
```

```
psy = pd.read_csv('Youtube01-Psy.csv')
katty = pd.read_csv('Youtube02-KatyPerry.csv')
Lmfao = pd.read_csv('Youtube03-LMFAO.csv')
Eminem = pd.read_csv('Youtube04-Eminem.csv')
shakira = pd.read_csv('Youtube04-Eminem.csv')
```

```
# In[4]:
```

```
data=pd.concat([psy,katty,Lmfao,Eminem,shakira])
```

```
# In[5]:
```

```
del data['DATE']
del data['COMMENT_ID']
del data['AUTHOR']
```

```
# In[6]:
```

```
data
```

```
# In[7]:
```

```
data.duplicated()
```

```
# In[8]:
```

```
data.duplicated().sum()
```

```
# In[9]:
```

```
data=data.drop_duplicates()
```

```
# In[10]:
```

```
data.shape
```

```
# In[11]:
```

```
data.isnull().sum()
```

```
# In[12]:
```

```
data.reset_index(inplace=True)
```

```
# In[13]:
```

```
# Data cleaning and preprocessing
```

```
import re  
import nltk  
nltk.download('stopwords')
```

```
# In[14]:
```

```
from nltk.corpus import stopwords  
from nltk.stem.porter import PorterStemmer  
ps=PorterStemmer()
```

```
corpus=[]  
for i in range(0, len(data)):
```

```

review=re.sub('[a-zA-Z][0-9]', ' ', str(data['CONTENT'][i]))
review=review.lower()
review=review.split()

review=[ps.stem(word) for word in review if not word in
stopwords.words('english')]
review=' '.join(review)
corpus.append(review)

corpus

# In[15]:

df=data

# In[16]:

type(df['CONTENT'].loc[100])

# In[17]:

df.info()

# In[18]:

# Creating the TFIDF model
from sklearn.feature_extraction.text import TfidfVectorizer
tv=TfidfVectorizer(max_features=3000,ngram_range=(1,2))
X=tv.fit_transform(corpus).toarray()

# In[19]:

```

X

```
# In[20]:
```

X.shape

```
# In[21]:
```

```
y=pd.get_dummies(df['CONTENT'])  
y=y.iloc[:,1].values
```

```
# In[22]:
```

y

```
# In[23]:
```

```
# Since data is imbalanced  
# Trying over sampling
```

```
from imblearn.over_sampling import RandomOverSampler
```

```
rs=RandomOverSampler()  
X,y=rs.fit_resample(X,y)
```

X.shape,y.shape

```
# In[24]:
```

```
# Train Test Split
```



```
from sklearn.model_selection import train_test_split
X_train,X_test,y_train,y_test=train_test_split(X,y,test_size=0.2,random_state=0)
```

```
# In[25]:
```

```
from sklearn.naive_bayes import MultinomialNB
MNB = MultinomialNB()
MNB.fit(X_train,y_train)
```

```
# In[26]:
```

```
predict = MNB.predict(X_test)
```

```
# In[27]:
```

```
from sklearn.metrics import accuracy_score
print('Accuracy of MultinomialNB',accuracy_score(y_test,predict)*100)
```

```
# In[28]:
```

```
from sklearn.metrics import confusion_matrix
print('Confusion matrix of MultinomialNB\n',confusion_matrix(y_test,predict))
```

```
# In[29]:
```

```
from sklearn.metrics import classification_report
print('Classification report of
MultinomialNB\n\n',classification_report(y_test,predict))
```

## **7. SYSTEM TESTING**

## 7. SYSTEM TESTING

### 7.1 TEST CASES AND REPORTS

TEST CASE ID	TEST CASE/ACTION TO BE PERFORMED	EXPECTED RESULT	ACTUAL RESULT	FAIL/PASS
1	Check if dataset available	Dataset is uploaded	Dataset is uploaded	Pass
2	Clean the data	Unwanted symbols and words removed	Unwanted symbols and words removed	Pass
3	Split the data	Training and testing set created	Training and testing set created	Pass
4	Tfidf Vectorizer	Finds term frequency and inverse document frequency	Finds term frequency and inverse document frequency	Pass
5	Classifier algorithm(LR,RF,MNB)	Predict the probability of categorical dependent variable	Predict the probability of categorical	Pass

			dependent variable	
6	Classification Metrics	Confusion matrix and accuracy score provided	Confusion matrix and accuracy score provided	Pass
7	Save the model	The results are stored for future references	The results are stored for future references	Pass

## **8. CONCLUSION**

## **8. CONCLUSION**

### **8.1 CONCLUSION**

A Cascaded Ensemble Machine Learning Model-based method for detecting spam comments on YouTube, which have just experienced remarkable increase, is presented. It reviewed previous research on the screening of spam comments on YouTube and carried out classification tests using three different machine learning methods (Logistic regression, Multinomial Nave Bayes, Random Forest, and an Ensemble model (Ensemble with hard voting) etc.). The experimental findings demonstrated that the ESM-H model put forward in this study performed best across five assessment metrics. In contrast to other research , we developed a novel model that used fewer algorithms but produced better performance outcomes. The ensemble model was also used to analyse movies from various categories. It was discovered that the ESM-H model had the best support, precision, f1-score, and inaccuracy results.

### **8.2 FUTURE ENHANCEMENTS**

Further study is anticipated to demonstrate that this approach of spam identification can function on gifs and photos, allowing it to be used to other social media platforms including WhatsApp, Facebook, Twitter, Instagram, and others.

## APPENDICES

### A.1 SAMPLE SCREENS

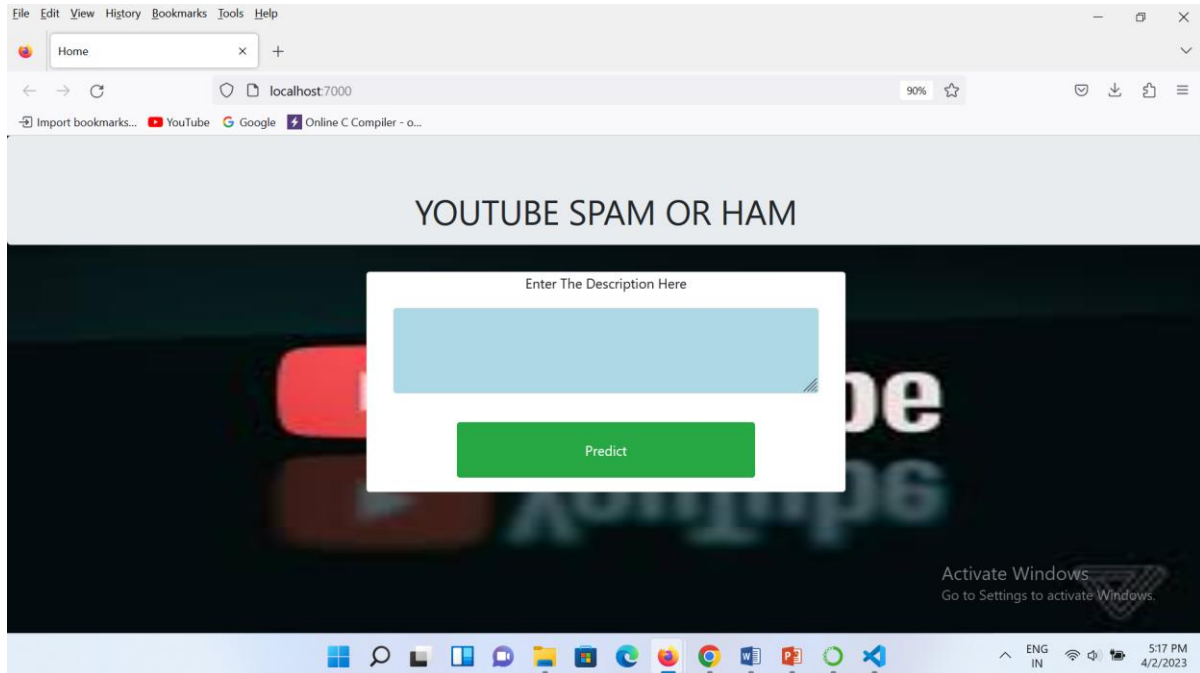


FIG NO. A.1 MAIN PAGE

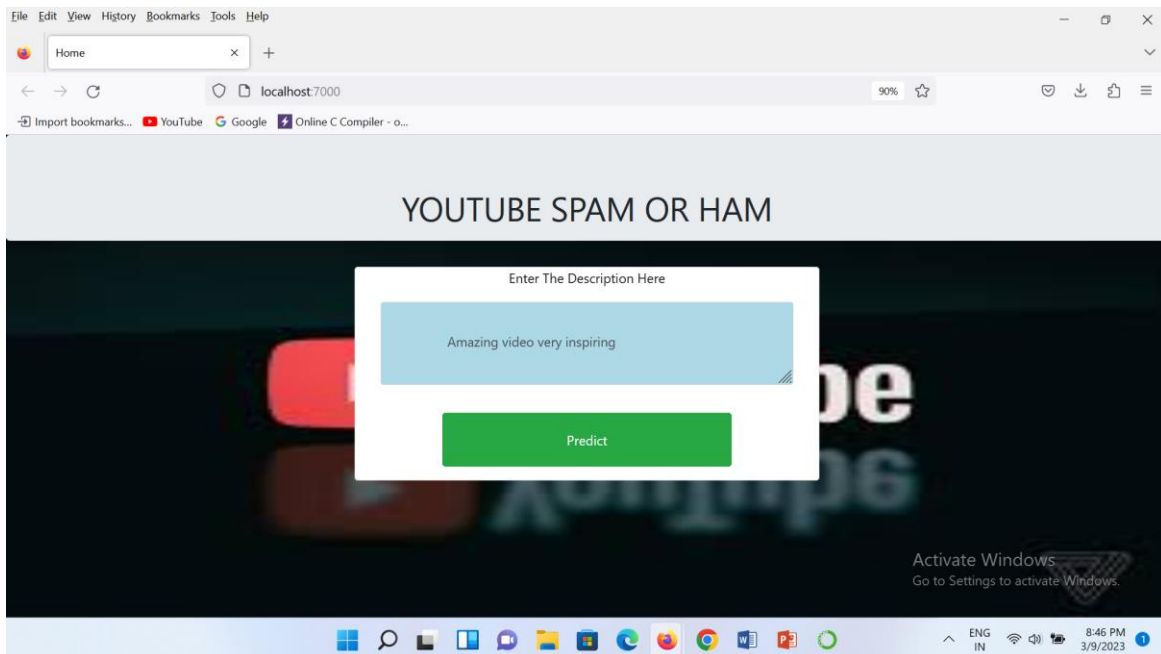
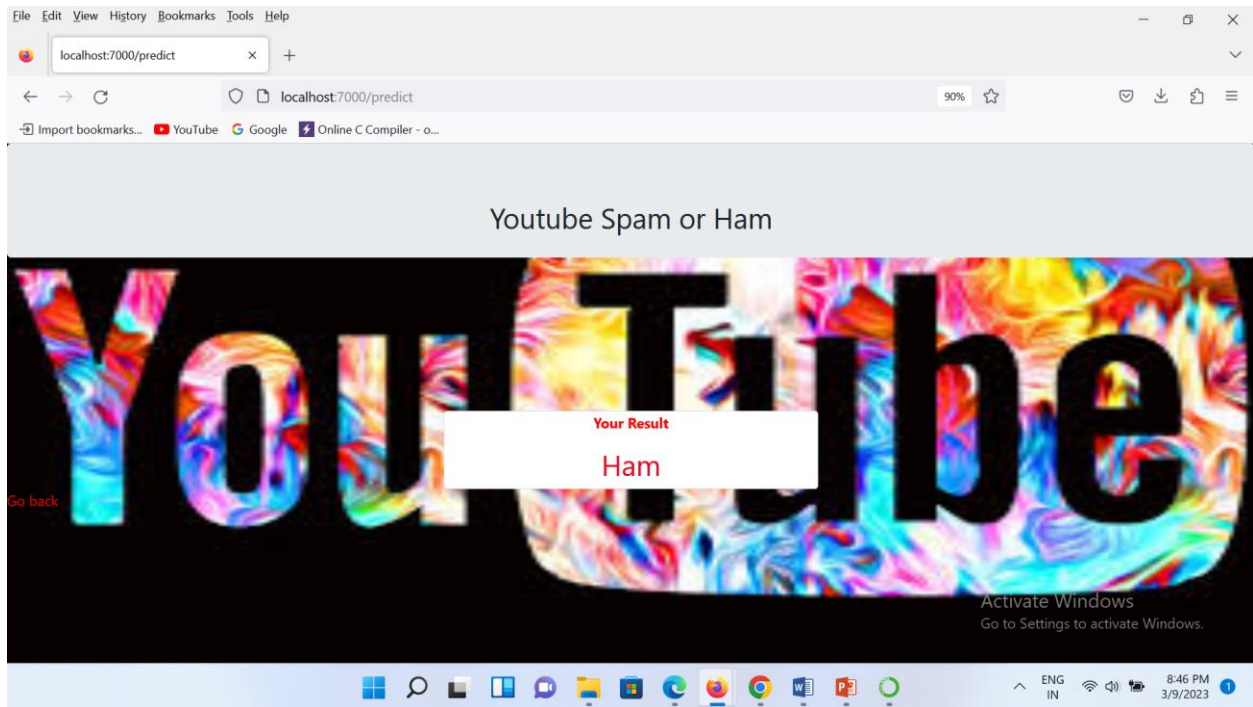
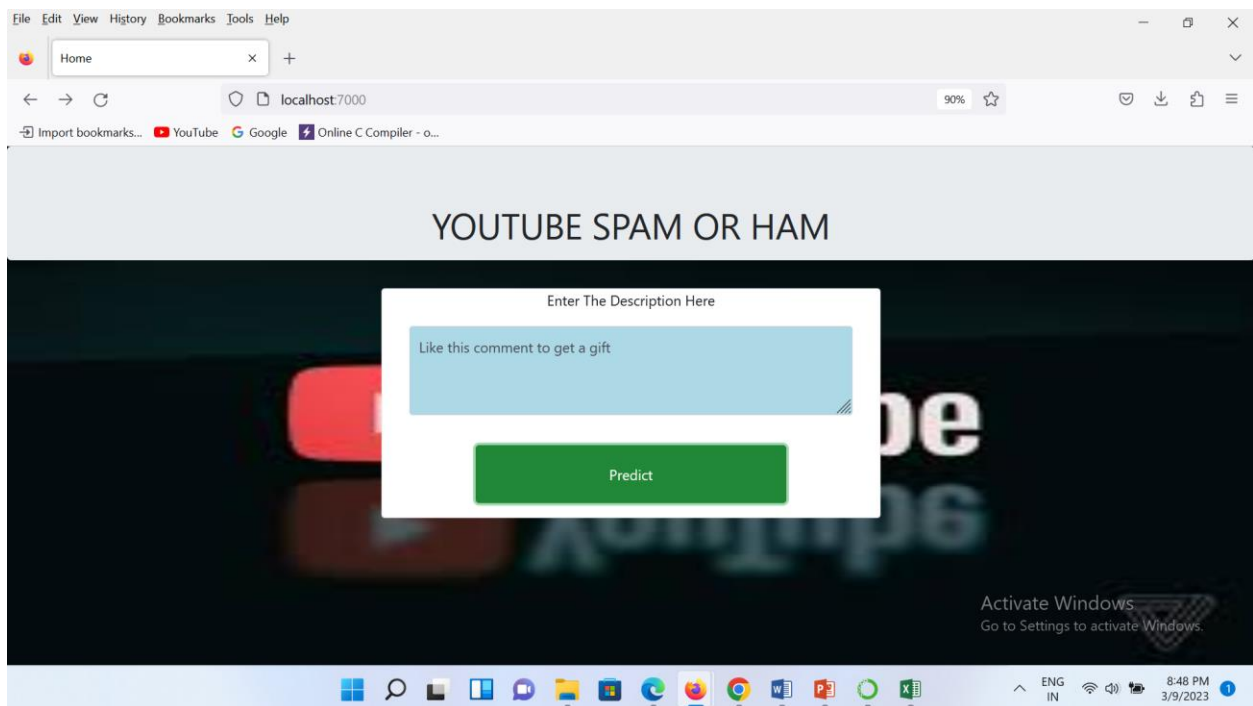


FIG NO. A.2 ENTERING A HAM COMMENT

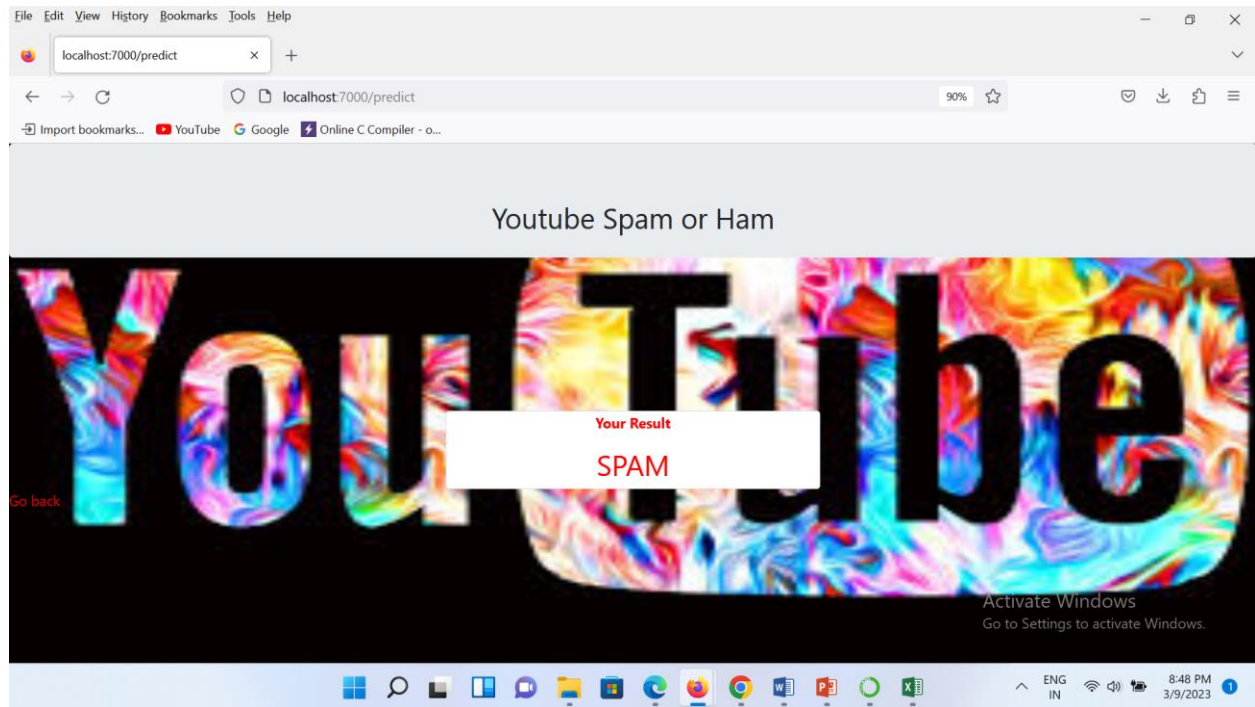


**FIG NO. A.3 DETECTION OF HAM**



**FIG NO. A.4 ENTERING A SPAM COMMENT**





**FIG NO A.5 DETECTION OF SPAM COMMENT**

## **A.2 PLAGIARISM REPORT**

Date: 05-04-2023

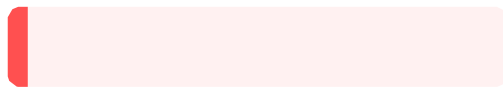
# PLAGIARISM SCAN REPORT

PLAGIARISED

4%

UNIQUE

96%



Word Count: 4767

Plagiarized Sentences: 9

Total Characters: 31262

Unique Sentences: 236

## Content

A CASCADED ENSEMBLE BASED YOUTUBE SPAM COMMENTS DETECTION USING MACHINE LEARNING Chegiredy Keerthana Department of CSE Panimalar Engineering College Chennai, India chegiredykeerthana.2001@gmail.com Shalini B Department of CSE Panimalar Engineering College Chennai, India shalinibaskar12@gmail.com Dr. A. Hemlathadhevi, M.E., Ph.d., Department of CSE Panimalar Engineering College Chennai, India hemlathadhevi@gmail.com Swathi S Department of CSE Panimalar Engineering College Chennai, India swathisathiya2001@gmail.com

**ABSTRACT**

YouTube possesses a spam filtering mechanism, although it consistently fails to deal with spam effectively. As a corollary, related studies on classifying YouTube spam comments have been done and classification experiments were performed. The subscribers of this social network might easily be impacted by these YouTube spam messages. Spam comments on YouTube have turned out to be a social issue and may spread more rapidly than the actual information. YouTube has had problems reducing the amount of malicious text posted by people trying to promote their videos or spread malicious links to steal their personal information. Many machine learning approaches are utilized to analyze enormous, complicated data, assisting professionals in predicting YouTube spam comments. The most accurate model is applied to forecast the spam YouTube comments is known as cascade ensemble model. The algorithms that are used in this model are Logistic regression, Random Forest, Multinomial Naive Bayes algorithms. Keywords Machine learning, Spam or Ham, Cascade Ensemble model, Ensemble Hard and Ensemble Soft, Logistic Regression, Random Forest Classifier, Multinomial Naïve Bayes, Flask I.

**Introduction**

Machine learning makes use of past data to forecast the data. The procedures of training and prediction both make use of machine learning algorithms. The training data are put into an algorithm, and the system makes predictions based on the new test data. Machine learning may be categorized into three major groups. Unsupervised, supervised, and reinforcement learning are the three different forms of learning. The supervised learning algorithm receives the input data as well as the

labelling that goes with it. The data must first be labelled by a person. Learning without supervision has no labels. This process must be used to identify how the input data are clustered. In order to improve performance, reinforcement learning actively interacts with its environment and absorbs both positive and negative input. When learning is done under supervision, an algorithm is used to determine the mapping function from the input to the output, which is  $y = f(X)$ . The objective is to estimate the mapping function as closely as possible so that you can forecast the output variables ( $y$ ) for fresh input data ( $X$ ). Algorithms for supervised machine learning use techniques including support vector machines, decision trees, logistic regression, and multi-class classification. The data used to train the algorithm must be labelled with the proper responses in order for supervised learning to take place. Classification difficulties may be further divided into supervised learning issues. Building a clear model that can predict the value of the dependent attribute from the attribute variables is the aim of this task. The dependent feature for categorical categorization is numerical, which distinguishes the two tasks. A classification model pertains to infer knowledge from the values that were seen. A categorization model will attempt to forecast the value of one or more outputs given one or more inputs. When the output variable is a category, such as "red" or "blue," there is a classification challenge. II. RELATED WORK Analysis on identifying spam content concentrates on an array of categories. Many research looked into website spam. Spammers target YouTube with low-quality content or advertisements as it becomes more and more popular as a platform for sharing videos. The number of spammers harming the YouTube community is growing, making it interesting to do study on how to identify them. Algorithms like Decision Tree, Logistic Regression, Nave Bayes, and SVM are utilised in references [1]. ESM-S ensemble soft voting is used. increases the classification's precision. Gubbala Pranati et al. [2] noted that as online social networks have improved, spammers have come to understand how easy it is to utilise them to trick people into engaging in harmful acts by leaving spam comments on videos. In this study, spam detection is done while looking through YouTube comments. According to references [3], the growing usage of social media has made these platforms the target of malevolent individuals. Despite the fact that social networks have their own spam detection systems, these systems occasionally fail to stop spam from entering their networks. The security and functionality of users on these networks are threatened by spam messages and contents. This study suggests a three-part strategy for detecting spam accounts. In the suggested framework, short link analysis, machine learning, and text analysis are all combined. To begin, a dataset was constructed with this objective in mind, and the characteristics of spam accounts were identified. Then, the link analysis component was used to examine the hyperlinks in the messages in this dataset. The machine learning component's properties were used to model it. Moreover, text analysis was used to examine the communications that users of social networks sent to one another. The suggested paradigm was used in a web-based application. The experimental research made possible by the framework led to the conclusion that it performed with a performance of 95.69%. The F-measure and accuracy assessment metrics were used to determine the effectiveness of this article while accounting for the sensitive content rate. On social networks, it aims to identify spam accounts, and these networks' spam detection policies are meant to help this. Social networks, spam identification, short link analysis, machine learning, and text analysis are some of the keywords. The datasets are balanced using NearMiss and SmoteTomek approaches in the suggested framework of [4] to feed several machine-learning models. The suggested voting-based ensemble models and standard ML models are next assessed on unbalanced and balanced datasets. For the proposed deep learning-based hybrid approaches,

word embeddings from GloVe and FastText are created on a balanced combined dataset, and then they are fed into a deep neural network made up of Conv1D and Bi-directional Recurrent Neural Network Layers with the Self-Attention Mechanism for improved context understanding and successful results. This study compares the best hybrid techniques for identifying social spam using unbalanced social network data. Also, a self-attention technique, deep learning with hyper-parameter optimisation, word embeddings, and machine learning ensembles are comprehensively contrasted. The suggested hybrid framework with deep learning-based approaches offers improved performance, according to tests and comparisons with existing methods. To extract characteristics from data, methods like Term Frequency-Inverse Document Frequency (TF-IDF) and bag-of-words are utilised as described in [5]. Due to the imbalance in the utilised SMS dataset, we applied over- and under-sampling strategies

The spam and ham SMS dataset is used to the support vector classifier, gradient boosting machine, random forest, Gaussian Naive Bayes, and logistics regression to assess the performance using F1 score. The experiment's findings indicate that random forest classification successfully categorises spam and ham SMSes with 99% accuracy. Using TF-IDF features and the oversampling approach, the suggested model is successfully trained to distinguish between the SMS categories of Ham and Spam. For the spam email dataset, the performance of the suggested technique was also assessed with significant 99% accuracy. For the purpose of feature selection in this study [6], a biologically inspired algorithm termed Krill herd Optimization (KHO) is used. Its performance is improved using a variety of optimisation functions such as the Quing function, Sumsquare function, Levy function, etc. In order to increase efficiency, the Dendritic Cell Algorithm (DCA) is also merged with KHA. In contrast to numerous cutting-edge machine learning classifiers, results between Dendritic Cell Algorithm (DCA) using KHA and other spam filtering models have been demonstrated. The methods were tested using a variety of optimisation functions that were visualised utilising tools, and the outcomes were confirmed in the study. The findings show an acceptable accuracy of 96%, which was assessed using several information retrieval measures, including recall, F-measure, and precision. The authors of this study [7] propose a modified Transformer model specifically made for identifying SMS spam messages in order to investigate the potential of the Transformer model in recognising the spam Short Message Service (SMS) messages. Using a baseline of several existing machine learning classifiers and cutting-edge SMS spam detection techniques, the proposed spam Transformer is evaluated using the SMS Spam Collection v.1 dataset and UtkMI's Twitter Spam Detection Competition dataset. The trials on SMS spam detection demonstrate that the proposed improved spam Transformer outperforms all other alternatives in terms of accuracy, recall, and F1-Score, with scores of 98.92%, 0.9451, and 0.9613, respectively. Also, the suggested model performs well on the UtkMI Twitter dataset, indicating a promising prospect of applying the model to additional problems of a similar kind. Word2vec has two primary training algorithms according to [8]: Skip Gram and Continuous Bag of Words (CBOW). Another popular word embedding technique was Rapid Text, a development of the word2vec model, created by Facebook in 2016. To better capture the meaning of shorter words, Fast Text represents each word in the corpus as an n-gram of letters. Each word in the corpus of Fast Text is represented as an n-gram of characters, allowing the embeddings to comprehend suffixes and prefixes and aiding in the understanding of shorter words. The Quick Text model's capacity to deal with uncommon words that haven't been seen in training data is another advantage. With an F1 score of 0.91, the neural network model in [9] surpasses the conventional models. Due to the infamously unbalanced nature of spam training data sets, we also look into the effects of this

unbalance and demonstrate that simple Bag-of-Words models perform best in these situations. However, a neural model that fine-tunes using language models from other domains significantly raises the F1 score, though not to the levels of domain-specific neural models. This implies that the approach taken may change based on the degree of imbalance in the data set, the quantity of data accessible in a low resource scenario, and other factors. Consequently, we provide the data sets for the scientific community to utilise. [10] claims that a variety of machine learning techniques, including NB, SVM, Random Forest (RF), and Neural Networks (NN), have been used to combat spam email filtering. The precision stays over 83.50 percent. 83.50% throughout the board. The method outlined in [11] can be used to determine the spam parameters influencing the internet of things-connected devices. The proposed technique is validated using the Internet of Things data set in order to achieve the best results. The usefulness of different supervised machine learning algorithms, including the J48, K-Nearest Neighbors (KNN), and Decision Tree (DT), in detecting spam and ham transmissions is examined in this work [12]. Studies revealed that the Decision Tree technique outperformed other machine learning classifiers in terms of accuracy. Several individuals utilise social media for their livelihood, according to [13]. Social media has a significant impact on our lives in many different ways. Although there are a lot of advantages, there is currently a big issue with the rise of nasty social media comments. In this work, we used machine learning algorithms to find offensive remarks posted in the Bengali language on social media and evaluated their effectiveness. Even though this topic has received a lot of attention in other languages, Bengali has seen little of it. According to [14], social media makes it easier to share and disseminate knowledge, ideas, and thoughts. Yet, it has an impact on individuals, either negatively or positively, just like any other innovation. They have developed into a forum for the dissemination of prejudice, unfavorable remarks, and cyberbullying. Bullying that takes place online, such as on social media, messaging services, gaming platforms, or mobile devices is known as cyberbullying. Posting, transmitting, or distributing offensive, hurtful, or unpleasant information constitutes cyberbullying. Researchers are trying very hard to find instances of cyberbullying on social networking platforms. The study presented in this paper focuses on identifying offensive textual remarks. A text-processing approach and a string-imaging method are suggested by the work from [15]. By processing the data into pictures, the CNN 2D visualisation method employed in this study may be similarly applied to datasets of different languages, including languages other than English.

### III. EXISTING SYSTEM

Spam comment screening on YouTube is not very efficient. With YouTube comments, applying the same method (language modelling) doesn't work as the features of the data are different. It employs the same technique for detecting spam in comments as it does for detecting spam on webpages. Less textual explanations and information are represented by features of YouTube comments. In especially in the modern socially and technologically linked culture, YouTube spam comments pose a serious threat to democracy by swaying public opinion. Researchers from a variety of fields, including computer science, political science, information science, and linguistics, have also investigated how to identify and reduce spam comments on YouTube. Unfortunately, using the same technique for filtering spam emails and websites still makes it difficult to find and stop the spread of YouTube spam comments. As a result, the existing system is ineffective.

### IV. PROPOSED SYSTEM

Building a machine learning model to determine if a YouTube remark is spam or not is the recommended solution. As everyone now has access to the internet and can post whatever they want, spam comments on YouTube are seen as being prevalent and are very tough to police. Hence, there is a higher likelihood that individuals would be misled. Machine learning is typically designed to handle these kind of complex tasks because it takes more



time to manually analyse these types of data. By using the prior data to identify the pattern and increase the model's accuracy by changing parameters, machine learning may be used to categories YouTube spam comments. This model is then utilized as the classification model. The Cascade Ensemble model provides the basis for the strategy used to categories the comments. Cascade ensemble models come in two different flavors: ESM-H (Ensemble with Hard Voting) and ESM-S. (Ensemble with soft voting). The hard voting ensemble voting technique is the most effective for detecting spam comments on YouTube. A deployment model is built using flask. A website that accepts comments and uses a model to determine whether they are spam or not.

A. LOGISTIC REGRESSION In the field of artificial intelligence, logistic regression is a component of the supervised machine learning framework. It is also viewed as a discriminatory paradigm since it tries to differentiate between various classes (or categories). In contrast to a generative algorithm, such as naive bayes, it is unable to create information of the class that it is attempting to forecast, such as an image. Logistic regression maximises the log likelihood function to provide the beta coefficients for the model. This changes slightly when seen from the viewpoint of machine learning. The negative log likelihood serves as the loss function in machine learning, and gradient descent is used to locate the global maximum. Also, overfitting can occur in logistic regression, particularly when the model has a lot of predictor variables. Regularization is frequently used to penalise large coefficients in the parameters of high dimensional models.

B. RANDOM FOREST CLASSIFIER Random forest is one of the most popular and often used algorithms among data scientists. Random forest is commonly used in classification and regression problems. It builds decision trees from different samples, using their average for categorization and majority vote for regression. One of the most important features of the Random Forest Algorithm is its capability to handle data sets with continuous variables, as in regression, and categorical variables, as in classification. The performance is found better in the case of classification and regression tasks. Given that the random forest uses a variety of trees to estimate the dataset's class, some decision trees may predict the right output while others may not. Yet when all the trees are taken into account, they accurately predict the outcome. So, the following two premises for a better Random forest classifier are: For the classifier to anticipate actual outcomes as opposed to an assumed result, there should be some real values in the feature variable of the dataset. The correlations between the predictions of each tree must be very small.

C. MULTINOMIAL NAÏVE BAYES The Multinomial Naive Bayes algorithm, a Bayesian learning technique, is widely used in Natural Language Processing (NLP). The system accurately predicts the tag of a text, such as an email or news story, using the Bayes principle. The tag with the highest likelihood is produced after calculating the probabilities of each tag for a certain sample. The Naive Bayes classifier classifies features that are different from one another, which combines the several methods that make up the classifier. One characteristic's inclusion or exclusion is independent of the existence or absence of another attribute. Using the multinomial model, data that cannot be represented numerically may be categorised. Its primary benefit is the vastly decreased complexity. . With less training data and without having to repeatedly retrain, it enables classification jobs.

V. ARCHITECTURE DIAGRAM VI. USECASE DIAGRAM VII. WORKING OF PROPOSED METHODOLOGY The Cascaded Ensemble Machine Learning Model of YouTube Spam Comments Detection System was improved using three machine learning techniques: LR (Logistic Regression), RF (Random Forest), and MNB (Multinomial Naive Bayes). They did well and had a great level of confidence. We offer a model ensemble that integrates them, and we evaluate the efficiency. To categorise the spam data, stop words like articles (like the, a, and an) and pronouns (like I, you, and it) must be removed. No single approach works for

all datasets. The ensemble model performed well, therefore we kept looking at several ways to find the best classification algorithm, which made use of three machine learning philosophies (i.e., LR, MNB, RF). The ensemble model, the ESM-H is used, to train and test the dataset (Ensemble with hard voting). They make predictions and assess the group.

A. YOUTUBE ID TRAIN MODEL Datasets are made up of remarks from five popular music videos. We only utilise named classes and comment-related content. The five data sets listed in Table.1's training and testing can lead to overfitting, where the classifiers excel exclusively when applied to that data and struggle when applied to comment data from other videos. So, in order to understand the findings, we incorporate all five video datasets in our paper. Dataset Spam Ham Total Psy 175 175 350 KatyPerry 175 202 350 LMFAO 236 303 438 Eminem 245 203 448 Shakira 174 196 470 Total 1005 978 1983

Table 1: Dataset Details

The datasets, which contain comments from well-known music videos, are provided as input. They contain YouTube ID, comment author, date, comment content, and labelled class (0: Ham or 1: Spam). The only things utilised are comment text and identified class. The model is trained using this dataset.

Figure 1: Structure of a dataset

As the datasets are text-based, pre-processing is done before machine learning is applied. Stop words are eliminated, tokens are shown with comments, and stop words are deleted using the PortStemmer function of the nltk package. Last but not least, we count the frequency of token occurrence using TF-IDF vectorization.

B. DATA ANALYSIS Data visualisation is the graphical representation of information and data in pictorial or graphical representations, such as charts, graphs, and maps. Data visualisation tools provide a quick way to identify and understand trends, data patterns, and outliers. Processing massive amounts of data and making decision-based on that data requires the use of tools and methodologies for data visualisation. Visual aids for data comprehension have long been thought to be useful. The most common types of data visualisation are charts, tables, graphs, maps, and dashboards. The most important result of data visualisation is finding data patterns. It is considerably easier to identify data trends when all the data is provided to the user in a visual style rather than a table. By displaying the relevance of the data in relation to other aspects, data visualisation puts the data into perspective. It demonstrates how specific data references fit into the overall scheme. Data visualisation may also be used to communicate a data story to audiences. When displaying the data facts in an accessible way, the visualisation may be utilised to tell a story and lead the audience to a clear conclusion.

C. CLASSIFICATIONS

1. LOGISTIC REGRESSION For problems involving categorization and prediction, logistic regression is frequently utilised. On the basis of 1 and 0, it is used to categorise. Logistic regression is one of the most frequently used Machine Learning algorithms in the Supervised Learning domain. It is used to forecast the categorical dependent variable using a specified set of independent variables. Logistic regression is used to predict the output when the dependent variable is categorical. The outcome must thus be a discrete or categorical value. Either True for 0 or False for 1 which are possible outcomes. Logistic regression and linear regression are fairly similar, with the exception of how they are used. In contrast to logistic regression, which is used to address classification issues, linear regression addresses regression issues. In logistic regression, we fit a "S" shaped logistic function instead of a regression line, which predicts two maximum values (0 or 1). The logistic function's curve illustrates the possibility of various events, such as whether the cells are cancerous or not, among others. Using both continuous and discrete datasets to classify fresh data, logistic regression is a crucial machine learning technique. The logistic regression equation may be obtained from the linear regression equation. The mathematical procedures to obtain Logistic Regression equations are as follows: The equation that can be derived for a straight line can be

written as  $y_0 = b_0 + b_1x_1 + b_2x_2 + \dots + b_nx_n$ . In Logistic Regression,  $y$  can only be between 0 and 1, thus to account for this, multiply the following equation by  $(1-y)$ ; for  $y=0$  and infinity for  $y=1$ , respectively. The range that is need, however, should be between  $-\infty$  and  $+\infty$ . If we use the logarithm of the equation, it becomes:  $\log[y/(1-y)] = b_0 + b_1x_1 + b_2x_2 + b_3x_3 + \dots + b_nx_n$  where  $\log$  is the logarithmic function  $x$  represents the likelihood of "success." The  $y$  interrupt for  $b_n$  is  $b_0$ .  $n$ th predictor's first-order logistic regression coefficient  $x_n$  is the  $n$ th predictor's value significant 2.

**RANDOM FOREST CLASSIFIER** The decision trees used in the Random Forest Algorithm all have the identical nodes, but their leaves vary depending on the input. It integrates the results of many decision trees to get an answer that reflects the average of all these decision trees. The process is known as a decision tree in the field of machine learning. A node is where the process starts. The procedure starts with a node, branches to another node, and continues in this manner until one reaches a leaf. A node asks a question to aid in categorising the data. The numerous routes that this node may lead to are represented by a branch. The node or leaf of a decision tree is the location where there are no more branches. The random forest approach, a type of supervised learning, uses labelled data to teach computers how to classify unlabeled data. The Random Forest Algorithm is often used by engineers because it may be used to problems with classification and regression. When using the Random Forest Algorithm to address regression difficulties, the mean squared error (MSE) is employed to identify how the data branches from each node.  $MSE = 1/N \sum_{i=1}^N (f_i - y_i)^2$  Where  $N$  is the number of data points  $f_i$  is the value returned by the model and  $y_i$  is the actual value for data point  $i$ . One should be aware that the Gini index, or the algorithm used to determine how nodes on a decision tree branch are selected, is frequently employed when running Random Forests on classification data. Based on the class and likelihood, this formula determines the Gini of each branch on a node, showing which branch is more likely to occur.  $Gini = 1 - \sum C_i = 1 - (p_i)^2$  Here,  $p_i$  stands for the class's observed relative frequency in the dataset,  $C$  for the total number of classes. Decision tree node branching patterns may also be predicted using entropy.  $Entropy = 1 - \sum C_i = 1 - (p_i) \cdot \log_2(p_i)$  Where,  $\log$  is the logarithmic function  $p_i$  stands for the class's observed relative frequency in the dataset,  $C$  for the total number of classes. The branch that the node should take is determined by entropy using the likelihood of a specific result. As a logarithmic function is used to calculate it, it is more technically challenging than the Gini index. The Random Forest classifier's formula is  $RF_{fii} =$

$RF_{fii} = \frac{RF_{fii}}{\sum_{j=1}^J RF_{fij}}$  where  $RF_{fii}$  is the importance of feature  $i$  calculated from all trees in the Random Forest model  $norm_{fii} = \frac{RF_{fii}}{\sum_{j=1}^J RF_{fij}}$  is the normalized feature importance for  $i$  in tree  $j$

Multinomial Naive Bayes algorithm, a Bayesian learning technique, is widely used in Natural Language Processing (NLP). The programme use the Bayes theorem to determine the tag of a text, such an email or a news story. The tag with the highest likelihood is produced after calculating the likelihood of each tag for a certain sample. The Naive Bayes classifier classifies features that are different from one another, which combines the several methods that make up the classifier. No feature's inclusion or exclusion is influenced by the presence or absence of another feature. No feature's inclusion or exclusion is influenced by the presence or absence of another feature. As just the computation of probability is needed, it is easy to implement. This approach may be applied to both continuous and discontinuous data. This straightforward approach may be used to forecast real-time applications. It is incredibly scalable and can easily handle large datasets. The probability that one characteristic will occur has no influence on the probability that the other feature will occur. The Naive Bayes approach works well for looking at text input and addressing problems involving several classes. As the Naive Bayes theorem is based on the Bayes theorem, it is



essential to first comprehend this idea. Thomas Bayes developed the Bayes theorem, which determines the likelihood of an event occurring based on knowledge of its conditions. When predictor B is available, we calculate the probability of class A. For tiny sample sets, Naive Bayes can outperform the most effective alternatives. It is used in a wide range of industries because it is reliable, easy to use, rapid, and accurate. The Multinomial Naive Bayes algorithm's formula is as follows:  $P(\alpha/\beta) = P(\alpha) * P(\beta / \alpha) / P(\beta)$  Where:  $P(\beta)$  = prior probability of  $\beta$   $P(\alpha)$  = prior probability of class  $\alpha$   $P(\beta | \alpha)$  = occurrence of predictor  $\beta$  given class  $\alpha$  probability D. ANALYSIS PERFORMANCE Three methods—Logistic Regression, Multinomial Naive Bayes, and Random Forest—are used to analyse five metrics: Acc (Accuracy rate), precision, recall, f1-score, and support. As a result, the ESM-H model had the greatest accuracy, precision recall, f1-score, and support performance. As a result, the cascaded ensemble model method is employed. Algorithm Accuracy precision recall f1-score support LR 91.66 0.92 0.92 0.92 217 RF 100 1.00 1.00 1.00 1152 MNB 97.74 0.98 0.98 0.98 576 ESM-H 100 1.00 1.00 1.00 1152 Table 2: ESM-H The results of the Ensemble Hard vote show that Random Forest produces the greatest outcomes. It is therefore chosen for deployment. Figure 2: Performance of the algorithms VIII. RESULT Based on the conducted tests, it is anticipated that the most effective set of algorithms are the current classifiers that were frequently employed the characteristics in identifying comment spams. In a public test set of a better accuracy score method, the best accuracy is discovered. The one that was discovered is employed in the deployment that can assist in locating YouTube spam comments. IX. CONCLUSION A Cascaded Ensemble Machine Learning Model-based strategy for identifying spam comments on YouTube—which have recently shown a notable increase—is provided in this suggested study. Using three alternative machine learning techniques (logistic regression, multinomial Nave Bayes, random forest, and an ensemble model), it performed classification tests and examined prior research on the filtering of spam comments on YouTube (Ensemble with hard voting)etc.).The experimental results showed that the ESM-H model proposed in this work outperformed the competition on five evaluation measures. We created a unique model that, in contrast to earlier studies, utilised fewer algorithms yet gave superior performance results. Also, movies from various genres were examined using the ensemble model. The ESM-H model was shown to have the greatest outcomes for support, precision, f1-score, and inaccuracy. It is hoped that further research will show that this method of spam identification may work with gifs and photographs, enabling it to be used to other social media sites including WhatsApp, Facebook, Twitter, Instagram, and others.

## SOURCES

0% Plagiarized

A categorization model will attempt to forecast the value of one or more outputs given one or more inputs. Variety of Classification models exist including ...A categorization model will attempt to forecast the value of one or more outputs given one or more inputs.

<https://futureacad.com/certification-in-data-science-and-machine-learning-by-iit-mandi>

### 0% Plagiarized

by MA Abid · 2022 · Cited by 7 — The support vector classifier, gradient boosting machine, random forest, Gaussian Naive Bayes, and logistics regression are applied on the ...

<https://dl.acm.org/doi/abs/10.1007/s11042-022-12991-0>

### 0% Plagiarized

by H Oh · 2021 · Cited by 5 — They contain YouTube ID, comment author, date, comment content, and labeled class (0: Ham or 1: Spam). We only use comment content and ...

<https://ieeexplore.ieee.org/iel7/6287639/9312710/09580841.pdf>

### 0% Plagiarized

by S Ishikawa · 2023 — Compared with simplifying the model, it is more accurate to explain that the model is trained using this dataset and find an abstract pattern for inferences ...

<https://www.sciencedirect.com/science/article/pii/S1569843223000377>

### 0% Plagiarized

For problems involving categorization and prediction, logistic regression is frequently utilised. Some examples of these use cases are:.

<https://logicmojo.com/logistic-regression-machine-learning>

### 0% Plagiarized

In Logistic regression, instead of fitting a regression line, we fit an "S" shaped logistic function, which predicts two maximum values (0 or 1).

<https://www.javatpoint.com/logistic-regression-in-machine-learning>

### 0% Plagiarized

Where N is the number of data points,  $f_i$  is the value returned by the model and  $y_i$  is the actual value for datapoint  $i$ . This formula calculates the distance ...

<https://www.jetir.org/papers/JETIR2105860.pdf>

### 1% Plagiarized

May 11, 2018 ·  $RF_{fi}$  sub(i)= the importance of feature i calculated from all trees in the Random Forest model;  $norm_{fi}$  sub(ij)= the normalized feature importance for i in tree j; See method featureImportances in treeModels.scala. Conclusion. This goal of this model was to explain how Scikit-Learn and Spark implement Decision Trees and calculate Feature ...

<https://towardsdatascience.com/the-mathematics-of-decision-trees-random-forest-and-feature-importance-in-scikit-learn-and-spark-f2861df67e3>

## REFERENCES

- [1]H. Oh, "A YouTube Spam Comments Detection Scheme Using Cascaded Ensemble Machine Learning Model," in IEEE Access, vol. 9, pp. 144121-144128, 2022, doi: 10.1109/ACCESS.2022.3121508.
- [2]Gubbala Pranathi , Sri.S.K.Alisha, Sri.V.Bhaskara Murthy Mca Student, Senior Assistant Professor, Associate Professor Dept Of Mca B.V.Raju College, Bhimavaram et al, "YOUTUBE SPAM COMMENTS DETECTION",at Journal of Engineering Sciences , 2022, Vol 13 Issue 07,2022, ISSN:0377-9254
- [3] Çıtlak, Oğuzhan & Dörterler, Murat & Doğru, İbrahim, (2022). A Hybrid Spam Detection Framework for Social Networks. Journal of Polytechnic. 10.2339/politeknik.933785.
- [4] Sanjeev Rao, Anil Kumar Verma and Tarunpreet Bhatia, “Hybrid ensemble framework with self-attention mechanism for social spam detection on imbalanced data” ,Expert Systems with Applications,Volume 217,2023,119594,ISSN 0957-4174.
- [5] Abid, M.A., Ullah, S., Siddique, M.A. et al. Spam SMS filtering based on text features and supervised machine learning techniques. Multimed Tools Appl 81, 39853–39871 (2022). Doi: 10.1007/s11042-022-12991-0
- [6] Aakanksha Sharaff, Chandramani Kamal, Siddhartha Porwal, Surbhi Bhatia, Kuljeet Kaur, Mohammad Mehendi Hassan, et al, “Spam message detection

using Danger theory and Krill herd optimization”, Volume 199,2022 ,108453,ISSN 1389-1286.

[7] X. Liu, H. Lu and A. Nayak, "A Spam Transformer Model for SMS Spam Detection," in IEEE Access, vol. 9, pp. 80253-80263, 2022, doi: 10.1109/ACCESS.2022.3081479.

[8] Ghanem, R. and Erbay, H. Context-dependent model for spam detection on social networks. SN Appl. Sci. **2**, 1587 (2022).doi:10.1007/s42452-020-03374-x

[9] Kawintiranon, K., Singh, L. & Budak, C. Traditional and context-specific spam detection in low resource settings. Mach Learn **111**, 2515–2536 (2022).

[10] Jáñez-Martino, F., Alaiz-Rodríguez, R., González-Castro, V. et al. A review of spam email detection: analysis of spammer strategies and the dataset shift problem. Artif Intell Rev **56**, 1145–1173 (2023).  
<https://doi.org/10.1007/s10462-022-10195-4>

[11] Aijaz Ali Khan, Rahul M. Mulajkar, Vajid N Khan, Shrinivas K. Sonkar, Dattatray G. Takale et al. (2022). A Research on Efficient Spam Detection Technique for Iot Devices Using Machine Learning A Research on Efficient Spam Detection Technique for Iot Devices Using Machine Learning. NeuroQuantology. November 2022. 625-631. 10.48047/NQ.2022.20.18..

[12] Saeed and Vaman. (2023). A Method for SMS Spam Message Detection Using Machine Learning. Artificial Intelligence & Robotics Development Journal. 214-228. 10.52098/airdj.202366.

[13] Sherin Sultana, Md Omur Faruk Redoy, Jabir Al Nahian, Abu Kaisar Mohammad Masum , Sheikh Abujar et al., “Detection of Abusive Bengali Comments for Mixed Social Media Data Using Machine Learning” in Research Square at 2022.

[14] Hooda, R., Jaiswal, A., Bansal, I., Jain, M., Singh, P., Sachdeva, N. et al., (2022). Detection of Offensive Comments for Textual Data Using Machine Learning. In: Sugumaran, V., Upadhyay, D., Sharma, S. (eds) Advancements in Interdisciplinary Research. AIR 2022. Communications in Computer and

Information Science, vol 1738. Springer, Cham. [https://doi.org/10.1007/978-3-031-23724-9\\_20](https://doi.org/10.1007/978-3-031-23724-9_20).

[15] Lee, Hwabin, Sua Jeong, Seogyong Cho, Eunjung Choi, et al., "Visualization Technology and Deep-Learning for Multilingual Spam Message Detection" 2022, Electronics 12, no. 3: 582.