# CommentSecure: YouTube Spam Comment Filtering using Ensemble Learning

DISSERTATION

Submitted in partial fulfillment of the requirements of the MTech in Data Science and Engineering Degree programme

By

(Jeevan Madhukar Chavan) (2021SC04033)

Under the supervision of (Savio Coelho)

BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE

Pilani (Rajasthan) INDIA

(March, 2024)

# BIRLA INSTISTUTE OF TECHNOLOGY & SCIENCE, PILANI FIRST SEMEST 2023-24

**Dissertation Title: CommentSecure: YouTube Spam Comment Filtering using Ensemble Learning**

**Name of Student: Jeevan Madhukar Chavan**

**Name of Supervisor: Savio Coelho**

**ID No. of Student: 2021SC04033**

**EVALUATION DETAILS**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **EC**  **No.** | **Component** | **Weightage** | **Comments**  (Technical Quality, Originality, Approach,  Progress, Business value) | **Marks Awarded** |
| 1 | Dissertation Outline | 10% | Abstract – Outline Document Abstract is well-constructed, providing a concise yet comprehensive overview of the project. It effectively captures the problem statement, methodology, and anticipated outcomes, setting a clear expectation for readers | 100% |
| 2. | Mid-Sem Progress Seminar  Viva  Work Progress | 10%  5%  15% | The introduction is commendable, laying a strong foundation by effectively highlighting the significance of spam comment filtering on YouTube.  The literature review is well-structured,  data handling and modeling section could benefit from more specific details, particularly regarding dataset information, transparent bias handling, and a clear presentation of data preprocessing steps | 100% |

|  |  |
| --- | --- |
| **Organizational Mentor Name:** | Savio Coelho |
| **Qualification** | BE IT- 13 Years of experience |
| **Designation & Address** | Vice President - JPMorgan Chase & co, Mumbai |
| **Email Address** | [savio.coelho84@gmail.com](mailto:savio.coelho84@gmail.com) |
| **Signature** | **Savio Coelho** |
| **Date** | **30 Jan 2024** |

# Abstract

**Keywords**: Comment moderation, Classification techniques, Naive Bayes, KNN, SVM, ensemble learning

YouTube has played a pivotal role as a vital social media platform for video sharing since its establishment. However, the platform's open nature exposes users to malicious intentions, such as the dissemination of malware and profanity, often manifesting in the comment section. Despite the presence of YouTube's built-in spam control tool, its efficacy falls short in addressing the dynamic landscape of malicious and spam content within comments.

In response to this challenge, our project conducts a comprehensive evaluation of top- performance classification techniques, with a focus on incorporating ensemble learning methods to fortify YouTube's comment moderation capabilities. Through a rigorous statistical analysis, Naive Bayes, K-Nearest Neighbors (KNN), and Support Vector Machines (SVMs) emerge as statistically equivalent performers.

Building upon these findings and the power of ensemble learning, we introduce " CommentSecure" – an innovative system designed to filter comments on YouTube in real- time. This system represents a cutting-edge solution, leveraging advanced classification techniques, including ensemble learning, to effectively identify and mitigate spam, profanity and malicious content within the dynamic comment environment of YouTube.

**Chapter 1**

**Introduction**

Recently, the vast majority of Internet usage has been directed to the social media platforms. The reasons and causes for such implication are obviously related to the content provided by these platforms. Individuals, especially youth, do not tend to read from e-books, long web pages, and blogs, rather they prefer to read short texts or watch videos. Social media platforms provide the exact match of such purpose; for instance, short tweets, videos, posts and even comments. According to recent statistics, the most visited social platforms are Facebook followed by YouTube. According to the very same statistic, YouTube appears to have 1.5 billion active users per month, and their viewers spend, on average, more than an hour of watching per day only from mobile devices. Nevertheless, according to Amazon Web Traffic Statistics, a subsidiary of Amazon, YouTube is ranked as the most popular social media platform. Social media platforms are being continuously developed, and interesting features are consistently added for the ultimate user benefit. In this context, as a typical case, YouTube has enabled its users to have a share of the advertisement income based on Count per Click (CPC) or, in some cases, Count per View (CPV). As a result, many YouTubers have become more influential than their traditional media counterparts. To a considerable extent, not only the monetary benefit that matters but also the granted freedom, less content restriction and the selective nature of the videos. Further, the current Internet speed and accessibility level make YouTube videos an interesting target for watching. In spite of the mentioned advantages, YouTube suffers from varies disadvantages that deteriorate the quality of its contents. Among these disadvantages is the malicious and spam contents disseminated on its video comment field. Even though comment can be disabled, yet the ultimate benefit of the shared content dictates that the comments should be enabled. This is because comments can provide a very useful contribution to the shared video such as improvement suggestions, kind support, positive criticisms, and huge data repository for future analysis. Enabling the comments on YouTube means accepting the fact that a large number of comments will be considered as spam. Essentially, YouTube comment spam refers to any irrelevant, advertisement, unsolicited comments posted on YouTube.

In this scenario, this we present a comprehensive performance evaluation of several well-known machine learning techniques that can be applied to automatically filter such undesired comments. Our main goal is to find promising methods and settings that can be used in an online tool developed to detect undesired text comments posted on YouTube, besides to offer new public datasets and good base- lines for future comparisons.

* 1. **Problem Statement**

One of the major Challenge on YouTube is lead owners of famous channels facing problems in comments section of their videos as Spam Comments with aim to self- promote their videos or disseminate viruses and malwares which disable comment section for that channel.

* 1. **Objectives**
     + To develop a spam comment monitoring System which will detect disseminate viruses and malwares links in comment section of channel.
     + To detect the self-promoting / Abuse comments in video sections
     + To Label comments as Spam or Ham.
     + To protect users from malicious activities
     + To show that ensemble method is feasible to get more accurate results
  2. **Motivation:**

The motivation behind ensemble learning lies in addressing the limitations of individual models. Different models may capture different aspects of the underlying patterns in the data or make different errors. By combining these diverse models, ensemble methods aim to create a more reliable and generalizable prediction.

**Chapter 2**

**Literature Survey**

**G. Mishne, D. Carmel and R. Lempel, “Blocking Blog Spam with Language Model Disagreement,” AIRWeb vol. 5, pp. 1-6, 2005[1].**

Comment-related studies have started since more than a decade ago. One of the earliest studies in this field is the work of Mishne et al. In their study, the authors developed a relatively simple-coded model that requires no training for detecting comment spam (links) in blog pages comments. The developed model was totally based on the used language to identify the spam contents. The study took advan- tage of the language context while considering the possibility of comment spam; generally, comment spam violates the solid semantic of the language. Even though the developed model produced 7.5 percent of false positive rate and 11 percent of false negative rate, yet the used corpus was extremely limited which made it far from being a practical comprehensive model.

**Ashwin Rajadesingan and Anand Mahendran, “Comment Spam Classifica- tion in Blogs through Comment Analysis and Comment-Blog Post Relation- ships,”**

In their paper, they considered the problem of spamming in blogs. In blogs, spammers usually target commenting systems which are provided by the authors to facilitate interaction with the readers. Unfortunately, spammers abuse these commenting systems by posting irrelevant and unsolicited content in the form of spam comments. Thus, we propose a novel methodology to classify comments into spam and non-spam using previously-undescribed features including certain blog post-comment relationships. Experiments conducted using our methodology produced a spam detection accuracy of 94.82% with a precision of 96.50% and a recall of 95.80%.

**A. Sureka, “Mining user comment activity for detecting forum spammers in YouTube,” arXiv preprint arXiv:1103.5044, 2011[3].**

One of the studies exploited the concept of text mining in YouTube comment spam- ming was proposed by Author. The idea here is based on designing an automatic detection of comment spamming activity on YouTube. The author used text-mining approach to discover the spamming behavior. Essentially, the assumption was made that if multiple comments on the same video or repeatedly posted comments on unrelated videos do exist, then the user would be flagged as a spammer. The de- veloped method was empirically tested on a dataset of 13,000 comments extracted from 240 different users. The obtained results from the study presented a high per- centage of spam flags for a number of commenters, most of which were actually spammers. However, the main problem with the developed method is that it was tested on a very small sample data, making the method’s efficiency questionable.

**Chapter 3**

**Requirements and Analysis**

A software requirements specification (SRS) is a description of a software system to be developed. It lays out functional and non-functional requirements, and may include a set of use cases that describe user interactions that the software must provide.

# Hardware Requirements

* + - Processor -Intel I5 core
    - Speed - 1.1 GHz
    - RAM - 512 MB (min)
    - Hard Disk - 50 GB
    - Keyboard - Standard Keyboard
    - Mouse - Two or Three Button Mouse
    - Monitor - LED Monitor

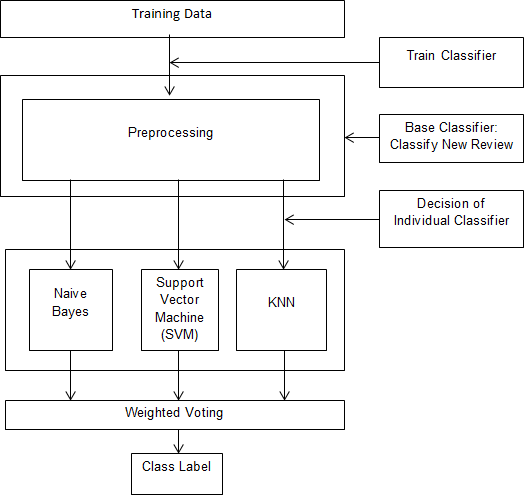
# Software Requirements

* + - Operating System – Windows 7/10
    - Programming Language - Python
    - Software Version - python 3.7 or above
    - Tools - pycharm
    - Front End – HTML

**Chapter 4**

**System Architecture and Use Diagram**

* 1. **System Architecture**



**Figure 4.1:** System Architecture

Above diagram shows the overall architecture of the proposed system. Based on these elementary requirements, the first step in the methodology is to prepare the datasets for analysis purpose. The used datasets are extracted from YouTube using YouTube Data API.

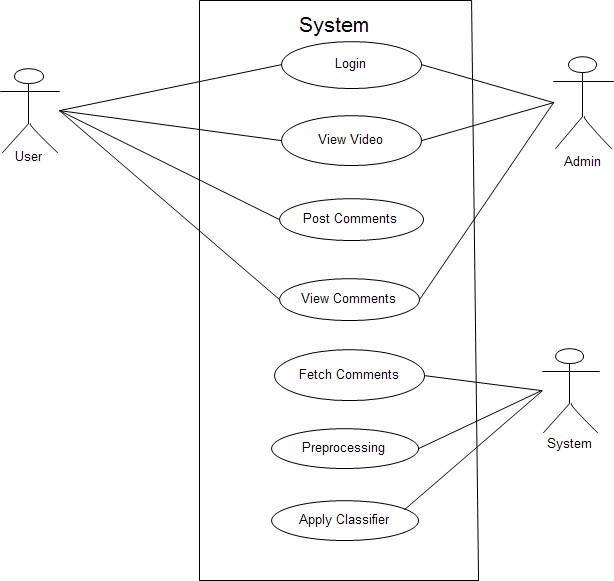
All datasets are combined into one file which composites the followings:

* + 1. Profile ID,
    2. Commenter ID,
    3. Date,
    4. Comment
    5. Classification

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

The first stage in data processing is isolating comments column from the dataset as one vector to be processed solely. The comments were regarded as bags of words and processed on this basis. No further preprocessing was done to keep the originality of the text. Essentially, each word was considered as a set of terms, and each term is composing a word of two or more alphabetical, underscore or a number.In addition, SVM,naive Bayes and KNN were used for classification pur- pose.

* 1. **Use-Case Diagram**

A use case diagram is a graphical representation of a user’s interaction with the system and depicting the specifications of a use case. A use case diagram can show the different types of users of a system and the various ways in which they interact with the system. Use case diagrams are used to gather the requirements of a system

including internal and external influences. These requirements are mostly design requirements. So when a system is analyzed to gather its functionality use cases are prepared and actors are identified.

**Chapter 5**

**Methodology and Algorithms**

* 1. **Methodology**

The project involves a comprehensive evaluation of three classification techniques: Naive Bayes, KNN, and SVM. These techniques are selected based on their widespread use and effectiveness in text classification tasks. Ensemble learning is introduced to harness the strengths of each individual classifier, creating a more robust and accurate system.

* + 1. **Data Collection:**

To conduct a comprehensive evaluation of comment moderation techniques, a diverse and representative dataset of YouTube comments was collected. This dataset included comments with varying content types, such as text, emojis, and multimedia elements. Efforts were made to ensure the inclusion of both benign and malicious comments to create a balanced training and testing set.

5.1.2 Preprocessing:

The collected dataset underwent thorough preprocessing to standardize and clean the text data. This involved tasks such as lowercasing, removal of special characters, and stemming to ensure consistency and reduce dimensionality. Emphasis was placed on maintaining the context and semantics of the comments during preprocessing to preserve the nuances of user-generated content.

5.1.3 Feature Extraction:

For the classification models, the preprocessed comments were transformed into numerical feature vectors. Techniques such as TF-IDF (Term Frequency-Inverse Document Frequency) were employed to represent the importance of words within the comments. This feature extraction process aimed to convert textual information into a format suitable for training machine learning models.

5.1.4 Model Selection:

Three classification techniques, namely Naive Bayes, K-Nearest Neighbors (KNN), and Support Vector Machines (SVM), were chosen for their proven effectiveness in text classification tasks. These algorithms were implemented using popular machine learning libraries, and their hyperparameters were tuned through cross-validation to optimize performance.

5.1.5 Ensemble Learning:

Ensemble learning was introduced to harness the collective intelligence of multiple classifiers. A weighted voting system was employed, where the decisions of Naive Bayes, KNN, and SVM were combined to produce a final classification. The weights assigned to each classifier were optimized based on their individual performance on the validation set.

5.1.6 Evaluation Metrics:

To assess the performance of the models, standard classification metrics were used, including precision, recall, and F1 score. Precision measures the accuracy of positive predictions, recall gauges the model's ability to capture all positive instances, and F1 score provides a balance between precision and recall.

5.1.7 Experimental Setup:

The dataset was split into training and testing sets to evaluate the models' generalization performance. Cross-validation was employed during training to minimize overfitting and ensure robustness. Additionally, a validation set was used for hyperparameter tuning and ensemble learning optimization.

5.1.8 Real-time Implementation:

The final ensemble learning model was integrated into the "CommentSecure" system for real-time YouTube comment moderation. The system continuously monitors and filters incoming comments, leveraging the trained classifiers to swiftly identify and mitigate spam, profanity, and malicious content.

5.1.9 Ethical Considerations:

Throughout the methodology, ethical considerations were prioritized, and steps were taken to ensure the responsible use of the models. Bias detection and mitigation strategies were implemented to prevent the algorithms from exhibiting discriminatory behavior.

This detailed methodology aimed to provide a transparent and replicable approach to evaluating and implementing comment moderation techniques on the YouTube platform. The combination of rigorous model selection, ensemble learning, and ethical considerations contributes to the robustness and reliability of the proposed system, "CommentSecure."

* 1. **Algorithms**

We will make use of Naïve Bayes, K-nearest neighbor and SVN algorithm for the purpose of spam detection in YouTube videos. Using these algorithms, we achieve results that are more efficient and accurate in the classification of a given video as a spam.

* + 1. **Naive Bayes**

The Bayesian Classification represents a supervised learning method as well as a statistical method for classification. Assumes an underlying probabilistic model and it allows us to capture uncertainty about the model in a principled way by determining probabilities of the outcomes. It can solve diagnostic and predictive problems. This Classification is named after Thomas Bayes (1702- 1761), who proposed the Bayes Theorem. Bayesian classification provides practical learning algorithms and prior knowledge and observed data can be combined. Bayesian Classification provides a useful perspective for understanding and evaluating many learning algorithms. It calculates explicit probabilities for hypothesis and it is robust to noise in input data. Naive Bayes classifiers are highly scalable, requiring a number of parameters linear in the number of variables (features/predictors) in a learning problem. Maximum likelihood training can be done by evaluating a closed-form expression, which takes linear time, rather than by expensive iterative approximation as used for many other types of classifiers. Naïve Bayes is a simple technique for constructing classifiers: models that assign class labels to problem instances, represented as vectors of feature values, where the class labels are drawn from some finite set. It is not a single algorithm for training such classifiers, but a family of algorithms based on a common principle: all Naïve Bayes classifiers assume that the value of a particular feature is independent of the value of any other feature, given the class variable. For example, a fruit may be considered to be an apple if it is red, round, and about 10 cm in diameter. A Naive Bayes classifier considers each of these features to contribute independently to the prob- ability that this fruit is an apple, regardless of any possible correlations between the color, roundness and diameter features. For some types of probability models, Naïve Bayes classifiers can be trained very efficiently in a supervised learning set- ting. In many practical applications, parameter estimation for Naïve Bayes models uses the method of maximum likelihood; in other words, one can work with the Naive Bayes model without accepting Bayesian probability or using any Bayesian methods.

* + 1. **KNN**

K nearest neighbors is a simple algorithm that stores all available cases and classifies new cases based on a similarity measure (e.g., distance functions). the k- nearest neighbors algorithm (k-NN) is a non-parametric method used for classification and regression. In both cases, the input consists of the k closest training examples in the feature space. The output depends on whether k-NN is used for classification or regression.

In k-NN classification, the output is a class membership. An object is classified by a majority vote of its neighbors, with the object being assigned to the class most common among its k nearest neighbors (k is a positive integer, typically small). If k = 1, then the object is simply assigned to the class of that single nearest neighbor. In k-NN regression, the output is the property value for the object. This value is the average of the values of its k nearest neighbors. k-NN is a type of instance-based learning, or lazy learning, where the function is only approximated locally and all computation is deferred until classification. The k-NN algorithm is among the simplest of all machine learning algorithms.

Both for classification and regression, a useful technique can be used to assign weight to the contributions of the neighbors, so that the nearer neighbors contribute more to the average than the more distant ones. For example, a common weighting scheme consists in giving each neighbor a weight of 1/d, where d is the distance to the neighbor.

The neighbors are taken from a set of objects for which the class (for k-NN classification) or the object property value (for k-NN regression) is known. This can be thought of as the training set for the algorithm, though no explicit training step is required.

* + 1. **SVM**

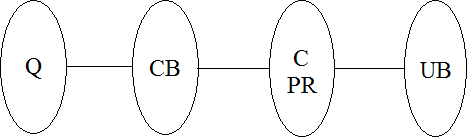
Support vector machine is supervised learning algorithm which can be used for both regression and classification. In this algorithm, system plots each data item as point in n-dimensional space with the value of a particular coordinate. After that classification by finding the hyper-plane that differentiate the two classes very well. Support Vectors are simply the co-ordinates of individual observation.

Support vectors are simply the best segregates the two classes. SVM is one of the best known methods in pattern classification and image classification. It is designed to separate of a set of training images two different classes, (x1, y1), (x2, y2), ..., (xn, yn) where xi in Rd, d-dimensional feature space, and yi in -1,+1, the class label, with i=1..n [1]. SVM builds the optimal separating hyper planes based on a kernel function (K). All images, of which feature vector lies on one side of the hyper plane, are belong to class -1 and the others are belong to class +1.

* 1. **Mathematical model**

The following terms shows in detail working of project.

1. **Mapping Diagram**



Where,

Q = User entered comment CB = Fetch Comment

C = Preprocessing PR = Classification **B] Set Theory**

* 1. Let S be as system which find entered URL is blacklist or not. S = *{*In, P, Op, φ*}*
  2. Identify Input In as In = *{*Q*}*

Where,

Q = User entered comment

* 1. Identify Process P as P = *{*CB, C, PR*}* Where,

CB = Fetch Comment C = Preprocessing

PR = Classification

* 1. Identify Output Op as Op = *{*UB*}*

Where, UB = U

After classification the request, system decides particular comment is spam or not.

φ = Failures and Success conditions

**Failures:**

1. Huge database can lead to more time consumption to get the information.
2. Hardware failure.
3. Software failure.

**Success:**

1. Search the required information from available in Datasets.
2. User gets result very fast according to their needs.

**Space Complexity:**

The space complexity depends on Presentation and visualization of discovered pat- terns. More the storage of data more is the space complexity.

**Time Complexity:**

Check No. of patterns available in the datasets= n

If (n¿1) then retrieving of information can be time consuming. So the time complexity of this algorithm is O*∧*n.

**Chapter 6**

**Conclusion**

We came to a conclusion that we will make use of combination of Naive Bayes, KNN and SVN algorithm using ensemble method for the purpose of spam detection in YouTube videos. Using these algorithms we hope to overcome the disadvantages faced by the existing systems and achieve results that are more efficient and accurate in the classification of a given comment as a spam.

**References**

* + - 1. G. Mishne, D. Carmel and R. Lempel, “Blocking Blog Spam with Language Model Disagreement,” AIRWeb vol. 5, pp. 1-6, 2005.
      2. C. Smitashree and J. G. Breslin, “User Sentiment Detection: a YouTube Use Case,” The 21st National Conference on Artificial Intelligence and Cognitive Science, 2010.
      3. A. Sureka, “Mining user comment activity for detecting forum spammers in YouTube,” arXiv preprint arXiv:1103.5044, 2011.
      4. P. Heymann, G. Koutrika, and H. Garcia-Molina, “Fighting spam on social web sites: A survey of approaches and future challenges”, IEEE Internet Computing, 11(6):36–45,2007