International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 09 Issue: 01 | Jan - 2025 SJIF Rating: 8.448 ISSN: 2582-3930

Automatic Youtube Comment Spam Detection using Navie Bayes and Logistic Regression

Mr.K.Praveen Kumar[1], M.Srija[2],P.Rakesh[3],P.Sriram[4],S.Ajay[5]

[1]Mr.K. Praveen Kumar (assistant professor)
[2]M.Srija Department of Computer Science and Engineering (Joginpally BR Engineering College)
[3]P.Rakesh Department of Computer Science and Engineering (Joginpally BR Engineering College)
[4]P.Sriram Department of Computer Science and Engineering (Joginpally BR Engineering College)
[5]S.Ajay Department of Computer Science and Engineering (Joginpally BR Engineering College)

ABSTRACT

During recent years, the volume and relevance of user-generated content on platforms such as YouTube have created a predicament in spam comment detection and filtering. This paper proposes an automated system for spam comment detection using Naive Bayes and Logistic Regression algorithms. After preprocessing, the spam comments go through tokenization, stop-word removal and stemming, while feature extraction is done using TF-IDF (Term Frequency-Inverse Document Frequency), which translates the comments into a numerical representation. Labeled datasets of YouTube comments introduce spam or non-spam categories for Naive Bayes and Logistic Regression model training. Performance is evaluated using generalized metrics like accuracy, precision, recall, and F1-score. Both models detect spam effectively, while Logistic Regression performs at a clasp higher level than Naive Bayes, (Support vector machine),KNN(k-nearest neighbors). The resulting machine might belong to YouTube's comment moderation scheme, filtering spam automatically in real-time**,** optimizing user experience to engage in content-rich communication as against annoying content**.**

Key Words: Naïve Bayes, Logistic Regression,

TF-IDF, Youtube Comments

1.INTRODUCTION

User-generated content is tremendously popular on platforms like YouTube and has correspondingly increased the rates of user engagement and interaction. YouTube receives a lot of comments to the extent that moderation of the comments is neither practical nor scalable but requires automated solutions for efficient content moderation. This research paper solves the problem of spam detection in YouTube comments** with the application of machine learning techniques such as Naive Bayes and Logistic Regression. The algorithms are popular for the text classification task and can also create a base on which both algorithms would function in distinguishing spam and legitimate comments. The preprocessing methods of the proposed system are tokenization, stop word removal, stemming, and feature extraction through TF-IDF (Term Frequency-Inverse Document Frequency) to provide raw comment data with meaningful numbers. The main goal of this research is to develop a system that is automated to accurately classify the YouTube comments as "spam" or "non-spam." Because both models are trained on a labeled dataset, the capability to efficiently filter spam messages will be demonstrated tied to the evaluation. The results of this study would not only improve the quality of the user experience on YouTube, but they would also promote cleaner, more relevant interactions and relieve the pressure on the content moderators.

2.PROBLEM STATEMENT

YouTube is one of the major platforms of both video content and social networking where millions of comments go on record each day. These comments involve engagement between creators and viewers, yet they induce a significant amount of spam, including messages that are not in any way related to the topic, advertisements, spam comments, and repetitive posts. Therefore, spams invariably disrupt meaningful conversations and worse, decline the quality of community interactions on the platform.

YouTube comment moderation is tedious, expensive, and not scalable that collects millions of comments. Such limitations warrant the development of an automated approach that alleviates the processing of comments and would classify and filter them in real-time from spam contents. This paper deals with the problem of automated detection and classification of spam comments on YouTube using machine learning techniques. The objective is to build a system that can effectively discriminate spam comments from their authentic counterparts and thus improve user experiences, reduce the burden on moderators, and ensure a cleaner online environment as much as possible.

3. LITERATURE REVIEW

The literature on YouTube comment spam detection has evolved significantly, with numerous studies exploring various machine learning techniques. Early methods primarily focused on keyword-based approaches, which, though simple, struggled to accurately identify spam due to the variability and complexity of human language. As a result, researchers shifted toward machine learning models, such as Naive Bayes and Logistic Regression, which offer more flexibility in handling large datasets and complex patterns. Naive Bayes, in particular, has been widely used due to its efficiency, although it may struggle with nuanced spam patterns. Logistic Regression, on the other hand, has demonstrated better performance, particularly in handling complex data, as shown in studies by Soni et al. (2018). Additionally, more sophisticated models like Support Vector Machines (SVM) and deep learning techniques, including Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs), have been applied to improve detection accuracy, especially in scenarios involving large-scale datasets. However, despite the progress, challenges remain, such as the dynamic nature of spam, language diversity across platforms, and the need for contextual understanding of comments. Recent research has explored hybrid models combining various machine learning algorithms, as well as the use of deep learning and transfer learning to enhance model adaptability and performance. These advancements show promise in dealing with the evolving nature of spam and improving the effectiveness of detection systems across different languages and contexts.

4.METHODOLOGY

4.1 Datasets Collection: A dataset comprising many YouTube comments is gathered, either by scraping the publicly available YouTube comment by following the YouTube terms of service or using some already available labeled data for spam and non-spam comment. The dataset must be diversified to represent the actual world of YouTube comments, consisting of spam, irrelevant, abusive, and true non-spam comments.

- **Comment Labeling**: Each comment in the dataset is labeled either as spam (irrelevant, promotional, abusive, unsolicited content) or non-spam (related, meaningful, and associated with the video content). A labeled dataset is important to train and also evaluate the spam detection models.

4.2. Preparing Data:

Preprocessing deals with preparing raw text data for machine learning algorithms.

- **Cleaning Text**: Removes unnecessary characters, HTML tags, URLs, and punctuation marks from the comments and cleans them up.
- **Tokenization**: Breaking text into very small items called tokens, which will, by default, create the smallest possible segments-a word is a token-by the machine learning model. For example, for the comment "Great video!", it will be tokenized into ["Great", "video"].
- **Stop Words Removal**: This eliminates common words such as "is," "and," or "the," which do not add significant value to the classification. This cuts down the size of the data.
- **Stemming/Lemmatization**: Turning words to root or base form; for instance, "running" becomes "run" while "better" becomes "good". Hence, standardization will treat words with similar meanings as one feature.
- **Lower casing**: Lower case makes all the text so that 'Spam' and 'spam' behave as one same word; redundancy is avoided by this.

3.3. Feature Extraction: After cleaning, the cleaned text data will be transformed into a numerical form

that the machine learning algorithms can process. This is done through Feature Extraction

- **TF-IDF (Term Frequency-Inverse Document Frequency):** TF-IDF is used in transforming the text data into a representation in vector space. Measures how important a word is to a document in a collection. TF-IDF helps to emphasize rare, meaningful words (such as keywords belonging to spam content) while down-weighting frequently occurring, less informative words. Term Frequency (TF): The frequency.

4.4.Model Training

After extracting features, we train the machine learning models to classify a comment as spam or not-spam:

- **Naive Bayes Classifier:** Naive Bayes is a probabilistic model given by Bayes' Theorem which assumes that features are independent conditionally: it computes the probability that a comment is spam as per the frequency of words used in it. The model is suited to text classification problems and is effective when dealing with high dimensional data.
- **Logistic Regression:** A linear classifier is Logistic regression which classifies the comments as spam or non-spam by modeling the probability that a given comment is spam or not spam. Since it establishes a linear relation between the features extracted (words, presence of links, etc.) and the outcome (spam or non-spam), in order to obtain the final result, the output probability must be threshold to either of the two classes.
- **SVM (Support Vector Machine)**: SVM is best to apply in high dimensional feature spaces; it works wonders on such spam data, which are non-linearlly separable. But it needs to tune its hyperparameters carefully and at the same time requiring a lot of computational budget.
- **KNN (k-Nearest Neighbors**): KNN is simple to implement, is able to work well on data that is quite small without too many irrelevant features, but loses such advantage on larger datasets due to its high computation during the prediction phase.
- **Random forest:** give a complete and flexible solution especially for big datasets with many features. It balances both performance and interpretability since it is capable of giving a lot of valuable insights about feature importance, thus making it great for spam detection tasks.

4.5. Evaluation of the Models

A number of metrics would be used in evaluating the two models.

Train/Test Split: The data set is split into a training set (usually 70%-80%) and a test set (20%-30%). It uses the training set to fit models and the test set to evaluate the performance of the model.

Cross Validation: For robustness, k-fold cross-validation can be implemented, where the dataset is divided into k subsets and the data is repeatedly test and train k times on different combinations of subsets.

4.6 Evaluation Metrics

- **Accuracy:** This is the proportion of all comments (spam and non-spam) that are correctly classified.

Accuracy=Total Number of PredictionsNumber of Correct Predictions=TP+TN+FP+FNTP+TN

TP = True Positives: The number of correctly predicted positive instances (e.g., spam comments correctly classified as spam).

TN = True Negatives: The number of correctly predicted negative instances (e.g., non-spam comments correctly classified as non-spam).

FP = False Positives: The number of negative instances incorrectly classified as positive (e.g., non-spam comments incorrectly classified as spam).

FN = False Negatives: The number of positive instances incorrectly classified as negative (e.g., spam comments incorrectly classified as non-spam).

- **Precision:** This is the proportion of spam comments which are actually spam. Thus, it measures correctness in the positive predictions (spam).

  Precision=TP+FP/TP

- **Recall:** The true spam detection ability of this model in percentage. It tells how well

this model captures the reallife spam comments.

Recall=TP+FN/TP

F-1 Score: The harmonic mean measure of precision and recall. This is an equanimous measure when both precision and recall matter. These evaluation metrics were important as far as comparing the performance of Naive Bayes and Logistic Regression in addition to knowing both individual model efficiencies in spam detection.

4.7. Model Optimization

- **Hyperparameter Tuning**: fine-tunings for hyperparameters. Most of the hyper parameter can be set to improve accuracy. Naive Bayes also has advantage of improving the work through smoothing parameters while regularization might be a fine optimal to Logistic Regression.
- **Grid Search/Random surface**: These are systematic searching algorithms for the best combinations in hyperparameters.
- **Feature Selection:** The importance of each feature is evaluated and most relevant ones are selected to enhance performance of the models and lower overfitting levels.

4.8. Real-time Integration and Deployment

Once models have been trained and evaluated, it is possible to fit them to YouTube comment moderation workflows for the spam detection system.

- Training models can be hosted on a server from where it will pick new comments and process them for spam or non spam classification in real-time.
- **Continuous Monitoring**: This system can also be set to continuously monitor the accuracy and retrain the data model by periodically adding new data to it.

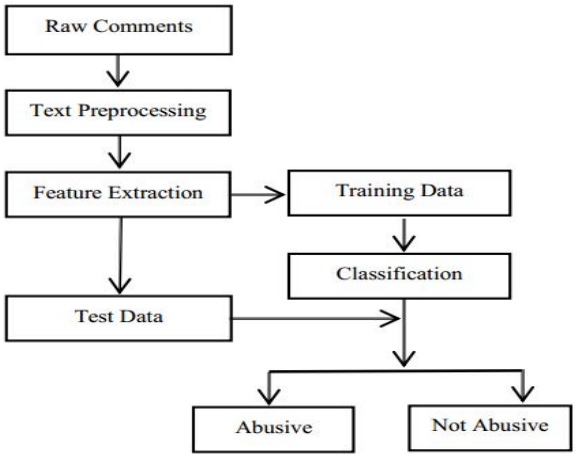
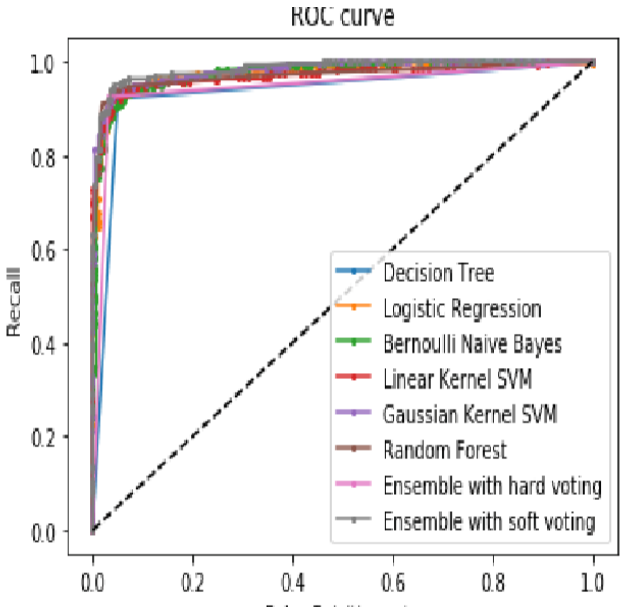


fig flowchart for spam comments

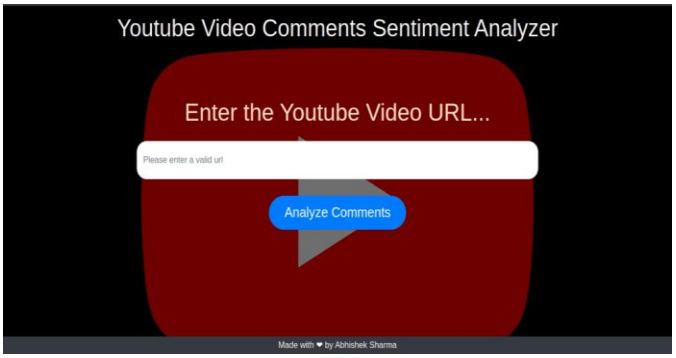
5.RESULTS

In the Results section, the theory behind the model evaluation revolves around assessing the performance of machine learning algorithms using a set of standard metrics. The most commonly used metrics for evaluating classification models like Naive Bayes and Logistic Regression, SVM, KNN, Random Forest are accuracy**,** precision**,** recall, and F1-score, each offering a different perspective on model performance.

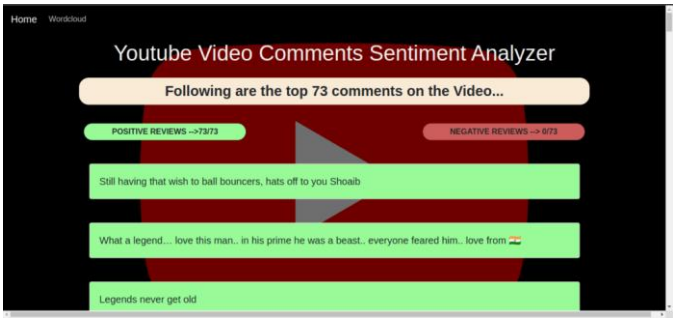
5.1 Roc Curve The Proposed Classifiers Scheme



6.2 Home Screen



6.3 results screen



6.CONCLUSION

A number of machine learning algorithms - like Naive Bayes, logistic regression, random forests, support vector machines (SVM), and k-nearest neighbors (KNN) can be effectively applied to carry out spam detection tasks, depending on the advantages each method provides. It has been shown that Naive Bayes is an efficient classifier that works best in the area of text classification especially for a large number of instances in a high-dimensional data set. Logistic regression is a viable method to determine whether a case is either present or absent in the outcome and strikes a balance between interpretability and accuracy when profile matching is required. Random forests are more robust and are proven to manage complex relationships within data leading frequently to very high accuracy. SVM is well known for its precision in classification with reference to data with high dimensional feature spaces, while KNN is a simple and intuitive approach but slow due to the amount of computational power required during testing while performing adequately on any size dataset. The algorithm that is to be used depends on factors such as size of the dataset, computing resource availability, and level of importance placed by the user on metrics like accuracy, precision, and recall.

7.REFERENCES

[1]. Mitchell, T. M. (1997). Machine Learning. McGraw-Hill."This book provides a comprehensive introduction to machine learning techniques, including algorithms like Naive Bayes, Logistic Regression, and others used in spam detection."

[2]. Rennie, J. D. M., Shih, L., Teevan, J., & Karger, D. (2003). Tackling the Poor Assumptions of Naive Bayes Text Classifiers. Proceedings of the 20th International Conference on Machine Learning (ICML).

[3]. Breiman, L. (2001). Random Forests. Machine Learning, 45(1), 5-32."This seminal paper introduces the Random Forest algorithm and demonstrates its effectiveness in classification tasks, including spam detection."

[4].Cortes, C., &Vapnik, V. (1995). *Support Vector Networks*. Machine Learning, 20(3), 273-297."This foundational paper explains Support Vector Machines (SVM) and their application to complex classification problems like spam filtering."

[5].Cover, T. M., & Hart, P. E. (1967). Nearest Neighbor Pattern Classification. IEEE Transactions on Information Theory, 13(1), 21-27."This paper introduces the K-Nearest Neighbors (KNN) algorithm, explaining its principles and use in pattern classification, such as spam detection."

[6].James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). An Introduction to Statistical Learning: *with Applications in R*. Springer."This book covers a broad range of machine learning methods, including Logistic Regression and Random Forests, with practical applications in data analysis."

[7].Zhang, H. (2004). The Naive Bayes Classifier: A Tutorial. UCI Machine Learning Repository. "This tutorial provides a step-by-step guide on implementing the Naive Bayes algorithm for text classification, which is often applied in spam filtering."