YouTube Comment Classification Outlier Detection Algorithms

Project By:

Ayaz khan
Data Science Enthusiast
Email: khan.15@iitj.ac.in
GitHub:
LinkedIn:

ABSTRACT:

In this project, several methods were applied, including text preprocessing, vectorization using TF-IDF, and dimensionality reduction with Truncated SVD. Unsupervised outlier detection models such as the Z-Score, Local Outlier Factor (LOF), and One-Class SVM were employed to detect spam based on patterns in the data. An ensemble method combined the results of these models for more robust spam classification. Finally, a Decision Tree classifier was trained on the detected outliers to label remaining comments. The results demonstrate that these techniques, especially the ensemble approach, can effectively identify spam comments and provide a scalable solution for content moderation on social media platforms.

KEYWORDS:

Outlier detection, TF-IDF vectorization, Truncated SVD, Local Outlier Factor (LOF), One-Class SVM, Z-Score method, Decision Tree classifier, Ensemble method, Natural Language Processing (NLP), Unsupervised learning.

INTRODUCTION:

With the rapid growth of user-generated content on platforms like YouTube, managing and moderating comments has become increasingly challenging. A significant portion of these comments is spam, which disrupts meaningful interactions and degrades the user experience.

By leveraging Natural Language Processing (NLP) and unsupervised learning methods, this project aims to efficiently classify spam comments. Various preprocessing techniques were applied to prepare the text data, followed by vectorization and dimensionality reduction. The primary focus was on implementing outlier detection models, such as the Z-Score method, Local Outlier Factor (LOF), and One-Class SVM, to identify spam comments. An ensemble approach was used to combine the strengths of these models, and a Decision Tree classifier was employed to label the remaining comments. This report details the entire machine learning pipeline, discusses the rationale behind the chosen models, and evaluates the effectiveness of the approach in classifying spam comments at scale.

DATA FETCH:

The dataset consists of a large number of comments for different YouTube channels and on different videos. One YouTube channel data was considered for further work for spam classification. We then performed Sampling to reduce the size of the dataset.

MACHINE LEARNING PIPELINE

A Machine Learning pipeline automates the workflow of an entire machine learning activity. It can be accomplished by facilitating the transformation and correlation of a data sequence within a model that can be examined to yield the output. The pipeline followed in the project is:

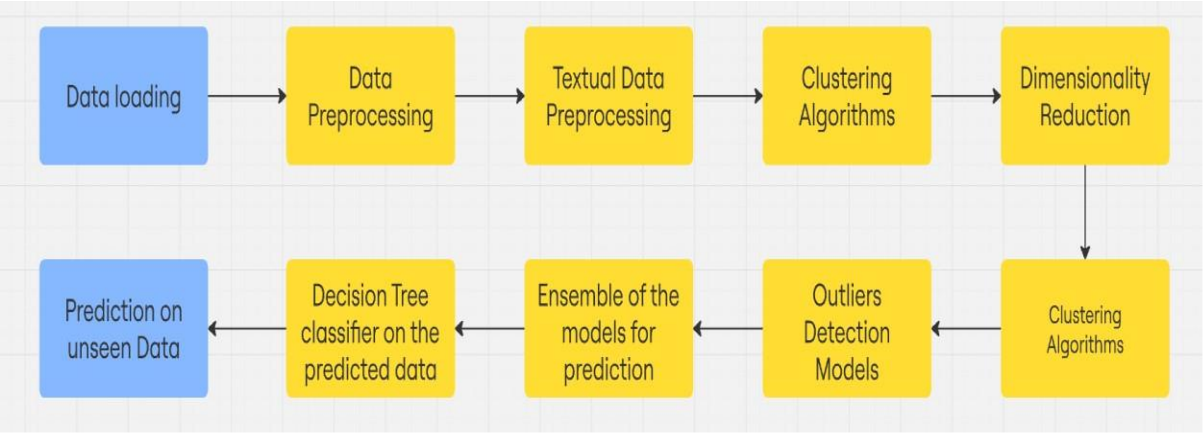
Data loading
Data Preprocessing
Textual Data Preprocessing
Clustering Algorithms
Dimensionality Reduction
Clustering Algorithms
Outliers Detection Models
Ensemble of the models for prediction
Decision Tree classifier on the predicted data
Prediction on unseen Data

Fig: ML Pipeline

DATA PREPROCESSING AND FEATURE SELECTION

Handling Null Values

The initial dataset had significant null values in several columns. To resolve this, null values were removed based on the "Comment (Actual)" feature. This reduced the dataset size a bit.

Feature Selection:

Features such as "Comment Time" and "User ID" were deemed irrelevant and were dropped.

TEXT PREPROCESSING

Textual data preprocessing was performed using the NLTK library. The steps involved:

- Stopword Removal
- Punctuation Removal
- Removing Numeric Data
- Lemmatization
- Tokenization

A custom function text_process() was created to handle these tasks efficiently.

VECTORIZATION

Textual data from the "Comment (Actual)" field was vectorized using the **TfidfVectorizer**. This method was chosen because it captures the importance of terms relative to the document (comments), thereby improving the spam detection capabilities of the model.

DIMENSIONALITY REDUCTION:

Since the dataset had a large number of sparse features, Truncated SVD (Singular Value Decomposition) was used to reduce the dimensionality. This technique was preferred over PCA because it can handle sparse matrices more effectively, which is typical in NLP tasks like this.

MODELS TRAINING:

Z-Score Method:

The **Z-score** method is a statistical technique that identifies outliers by measuring the deviation of each data point from the mean. The formula for calculating the Z-score is:

$$Z=\frac{X-\mu}{\sigma}$$

Where:

- X is the value of the data point.
- μ is the mean of the dataset.
- σ is the standard deviation of the dataset.

In the project, the Z-score method was applied to the vectorized features of the comments. Points with Z-scores greater than a certain threshold (e.g., 3 or -3) were considered outliers. The advantage of using this method is its simplicity and effectiveness for detecting anomalies in normally distributed data. However, the method might struggle in high-dimensional spaces, which is why it was combined with more advanced techniques.

Local Outlier Factor (LOF):

The **Local Outlier Factor (LOF)** algorithm is a density-based method that identifies outliers based on their local density compared to that of their neighbors. LOF assigns each data point an outlier score, which indicates how isolated the point is from its surroundings. The key parameters of the LOF method are:

- **n_neighbors**: This defines the number of neighbors considered when calculating the local density.
- **Contamination**: This specifies the proportion of outliers in the dataset, guiding the model on how many outliers to expect.

In this project, LOF was particularly effective for identifying comments that have a sparse representation in the feature space. Spam comments often exhibit different lexical or structural patterns, which the LOF method captures by comparing the density of a comment's neighbourhood with that of its peers. The LOF algorithm flagged comments with a lower local density as potential outliers, i.e., spam.

One-Class SVM (Support Vector Machine):

The **One-Class SVM** is an unsupervised algorithm that learns a decision boundary in the feature space. The algorithm works by mapping the data into a higher-dimensional space using a kernel trick and learning the boundary that encapsulates the normal data points. Data points that lie outside this boundary are classified as outliers.

Key parameters for One-Class SVM include:

- **Kernel function**: Typically, the radial basis function (RBF) kernel was used, as it performs well in non-linear spaces.
- **Nu**: This parameter controls the upper bound on the fraction of outliers and the lower bound on the fraction of support vectors, essentially controlling how aggressive the model is in detecting outliers.

Ensemble Method:

Each of the outlier detection methods—Z-score, LOF, and One-Class SVM—has its strengths and limitations when applied individually. To overcome these limitations, we employed an **ensemble approach**, where the results of the three models were combined to make the final spam detection decision. Specifically, comments were flagged as spam if they were classified as outliers by at least two of the three models.

Decision Tree Classification:

After identifying spam comments, the labelled data was used to train a **Decision Tree classifier**. This classifier was chosen for its interpretability and ability to handle the noisy and unbalanced nature of the data. The classifier was trained on a sample of 20,000 comments, and the remaining comments were labelled using this model. This semi-supervised approach helped scale the spam detection process across the large dataset.

Conclusion:

This project developed a robust spam comment detection pipeline using outlier detection methods and a decision tree classifier. By applying text preprocessing, vectorization, and dimensionality reduction techniques, we improved the efficiency and accuracy of the model. The ensemble of multiple outlier detection models ensured that the spam comments were reliably identified, while the Decision Tree classifier provided a scalable solution for labelling large datasets.

The pipeline can be further enhanced by experimenting with more advanced classifiers and finetuning hyperparameters for greater accuracy.