YouTube Comment Classification

Using Clustering Algorithms

Project By:

Kachhawa Goutham
Data Science Enthusiast
Email: Kachhawa.1@iitj.ac.in
GitHub:
LinkedIn:

Abstract:

This project explores the application of clustering algorithms to classify YouTube comments. With the increasing volume of unstructured data generated through social media, clustering techniques offer a scalable approach for organizing and analysing large datasets. In this project, several unsupervised learning methods, including K-Means and hierarchical clustering, were implemented to groups comments based on similarity, without predefined labels already being set. The results show that clustering algorithms can successfully sort YouTube comments into groups, which can help with sentiment analysis, finding spam, and getting people involved in the community.

Keywords:

YouTube Comment Classification, Clustering Algorithms, K-Means Clustering, Hierarchical Clustering, Unsupervised Learning, Natural Language Processing (NLP), Sentiment Analysis

Introduction:

The project demonstrate how clustering algorithms can be used to organize large-scale, unstructured text data, offering a practical solution to companies, content creators looking to extract actionable insights from their audience's comments. This process not only aids in better understanding user feedback but can also be used for broader applications like improving user experience, detecting spam, and performing sentiment analysis. This report outlines the methodology, results, and potential applications of clustering YouTube comments, highlighting the benefits and challenges of applying unsupervised learning techniques in real-world scenarios.

The dataset consists of YouTube comments from different YouTubers. The primary objective of this dataset is to identify a cluster that contains all the spam comments.

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.

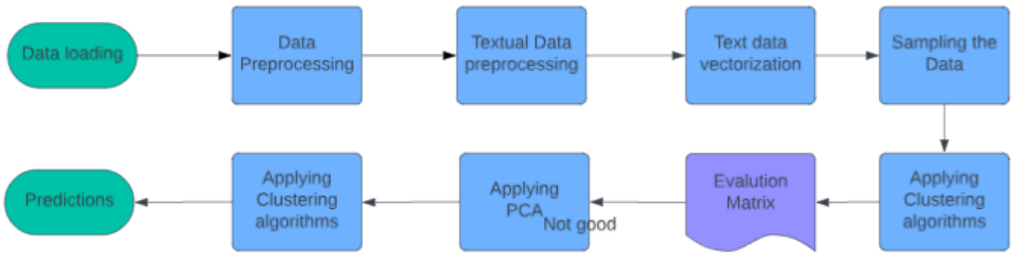

Fig: ML Pipeline

DATA PREPROCESSING AND FEATURE SELECTION

A real-world dataset generally contains noises, missing values, and may be in an unusable format which cannot be directly used for machine learning models. Therefore, Data Pre-processing is a required task for cleaning the data and making it suitable for a machine learning model, which also increases the accuracy and efficiency of a machine learning model. The dataset was checked for null values, and the null rows were dropped. We check the number of null-valued rows and drop them.

Feature selection means selecting the essential features for further pipeline tasks from the original dataset and dropping the rest. Since the sole purpose of the project is to detect spam comments, we select only the comments displayed as a feature and leave the rest.

TEXT PREPROCESSING

Whenever we have textual data, we need to apply several pre-processing steps to the data to transform words into numerical features that work with machine learning algorithms. With the help NLTK library performed the following task for pre-processing the text comments:

1)Text lowercase

2)Remove Numbers

3)Remove punctuation

4)remove whitespaces

5)Remove default stopwords

6)Stemming: Stemming is the process of getting the root form of a word. Stem or root is the part to which inflectional affixes (-ed, -ize, -de, -s, etc.) are added.

7) Lemmatization: Like stemming, lemmatization also converts a word to its root form. The only difference is that lemmatization ensures that the root word belongs to the language. We will get valid words if we use lemmatization.

VECTORIZATION

A generic natural language processing (NLP) model is a combination of multiple mathematical and statistical steps. It usually starts with raw text and ends with a model that can predict outcomes. Text cleaning, or normalization, is one of the most important steps of any NLP task. It includes removing unwanted data, converting words to their base forms (stemming or lemmatization), and vectorization.

There are three major methods for performing vectorization on text data:

1. Count Vectorizer

2. TF-IDF

3. Word2Vec

Here for vectorization TF-IDF was used to convert word to vectors. TF-IDF is an abbreviation for Term Frequency Inverse Document Frequency. This is very common algorithm to transform text into a meaningful representation of numbers which is used to fit machine algorithm for prediction. Count Vectorizer give number of frequencies with respect to index of vocabulary whereas tfidf consider overall documents of weight of words.

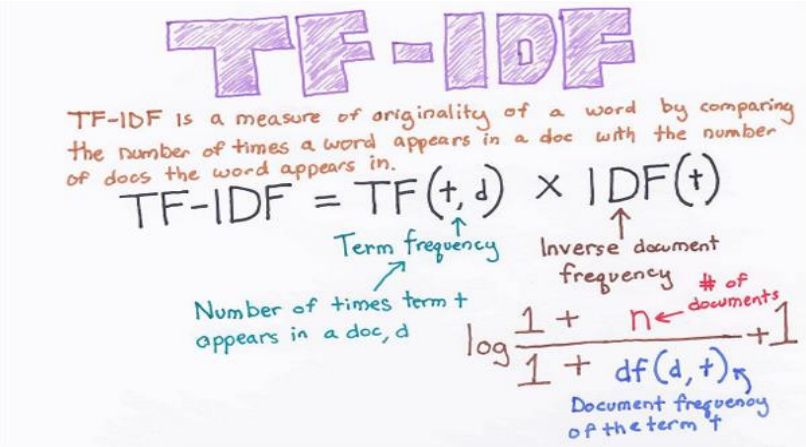

Fig: TF-IDF Interpretation

Source: https://medium.com/analytics-vidhya/demonstrating-calculation-of-tf-idf-from-sklearn-4f9526e7e78b

After vectorization, a new set of features in the form of floating-point values is obtained, which will be used for further procedures.

CLUSTERING (BEFORE PCA)

Clustering or cluster analysis is a machine learning technique, which groups the unlabelled dataset. It can be defined as "A way of grouping the data points into different clusters, consisting of similar data points. The objects with the possible similarities remain in a group that has less or no similarities with another group."

Here mainly popular Clustering algorithms that are widely used in machine learning were used:

1)k-means algorithm

2)DBSCAN algorithm

3)Agglomerative Hierarchical algorithm

K-Means algorithm: It classifies the dataset by dividing the samples into different clusters of equal variances. The number of clusters must be specified in this algorithm. It is fast with fewer computations required, with the linear complexity of O(n).

Result:

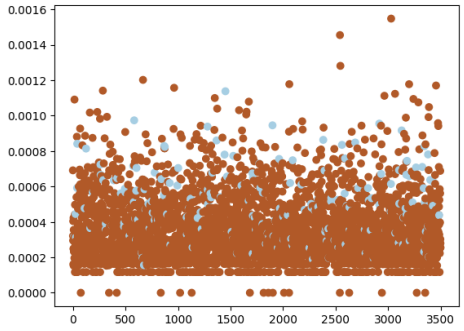

Fig: K-Means clustering predictions

DBSCAN algorithm: It stands for Density-Based Spatial Clustering of Applications with Noise. It is an example of a density-based model similar to the mean-shift, but with some remarkable advantages. In this algorithm, the areas of high density are separated by the areas of low density. Because of this, the clusters can be found in any arbitrary shape.

Result:

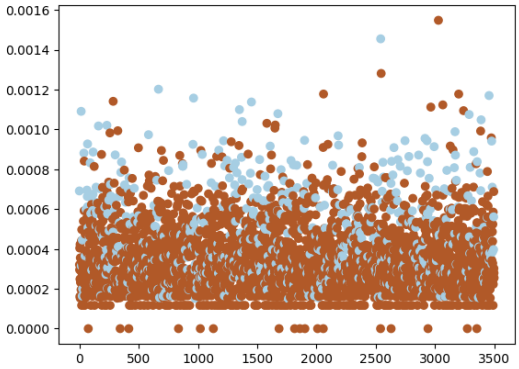

Fig: DBSCAN clustering predictions

HIERARCHICAL CLUSTERING: The Agglomerative hierarchical algorithm performs the bottom-up hierarchical clustering. In this, each data point is treated as a single cluster at the outset and then successively merged. The cluster hierarchy can be represented as a tree-structure.

Result:

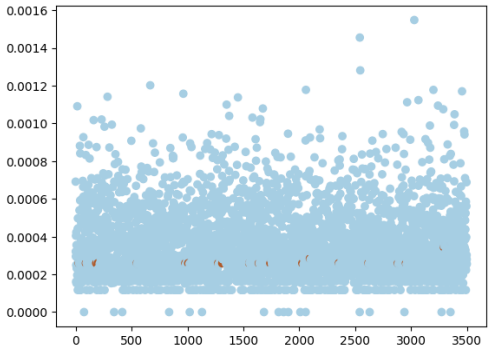

Fig: Hierarchical clustering predictions

SILHOUETTE SCORE:

Silhouette Coefficient or silhouette score is a metric used to calculate the goodness of a clustering technique. Its value ranges from -1 to 1.

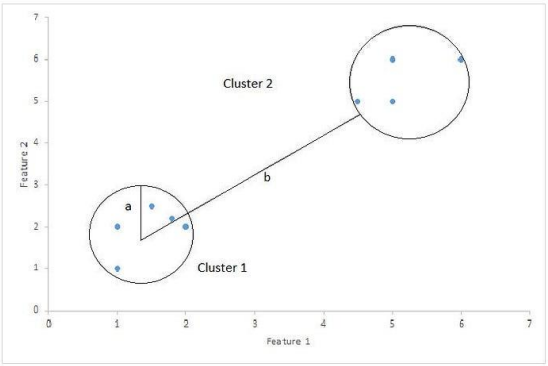

Fig: silhouette score visualization

Silhouette Score=(b-a)/max(a,b) ,Where

a=average intra-cluster distance, i.e., the distance between each point within a cluster

b= average inter-cluster distance, i.e., the average distance between all clusters

Results: Silhouette Score before applying PCA

Models	**Silhouette Score**
K-Means	0.0048
DBScan	0.0011
Agglomerative Clustering	0.0092

Applying PCA:

Principal Component Analysis is an unsupervised learning algorithm that is used for the dimensionality reduction in machine learning. It is a statistical process that converts the observations of correlated features into a set of linearly uncorrelated features with the help of orthogonal transformation. These new transformed features are called the Principal Components. PCA generally tries to find the lower-dimensional surface to project the high-dimensional data. The PCA algorithm is based on some mathematical concepts such as:

o Variance and Covariance

o from Eigenvalues and Eigen factors

Clustering after implementing PCA:

K-Means:

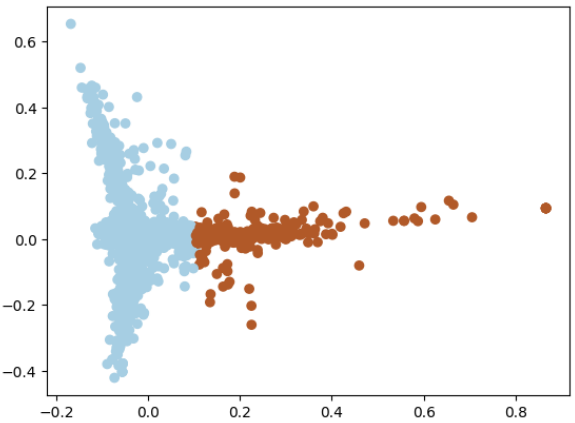

Fig: K-Means clustering predictions after applying PCA

DBSCAN:

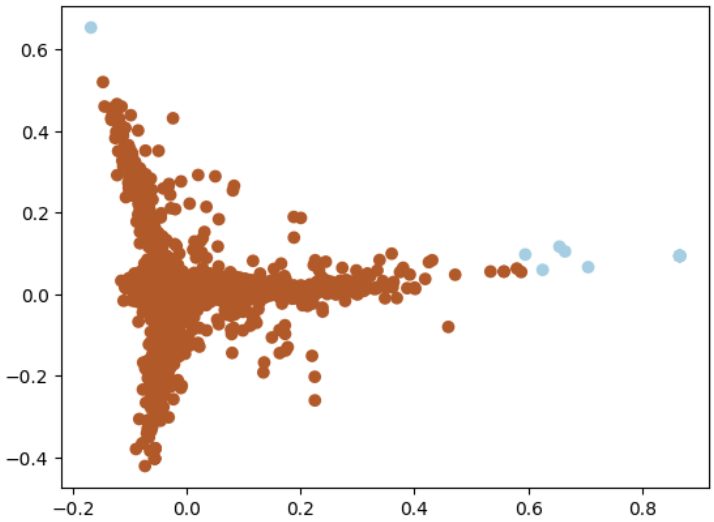

Fig: DBSCAN clustering predictions after applying PCA

HIERARCHICAL CLUSTERING:

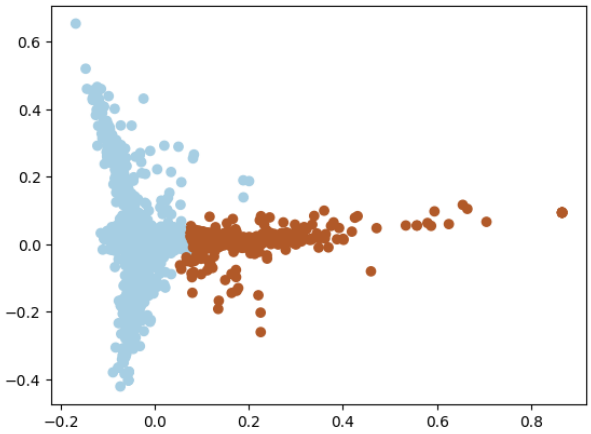

Fig: Hierarchical clustering predictions after applying PCA

Silhouette score:

Models	**Silhouette Score**
K-Means	0.6237
DBScan	0.8337
Agglomerative Clustering	0.6043

Results and Some Visualization:

Based on silhouette score after applying PCA K-Means algorithm is good for the data. Based on this the decision boundary is plotting and prediction are made.

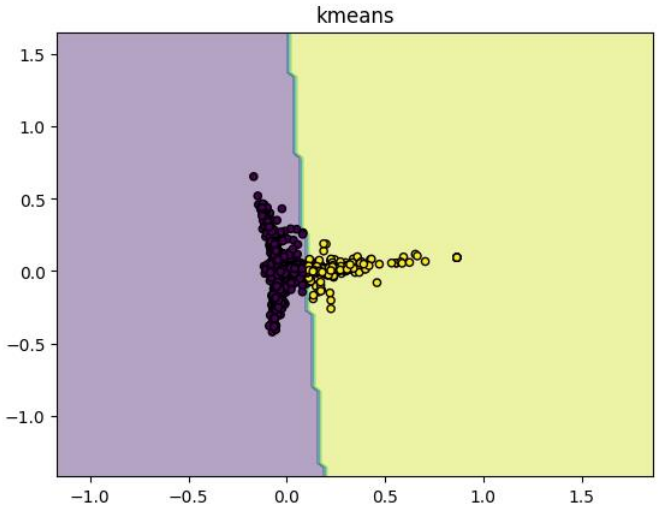

Fig: Decision boundary

Some visualization on the dataset considered. The number of comments considered was 3495. The spam and non-spam comments can be visualized using below plots.

we can use both K-Means and hierarchical clustering as the final model as the silhouette scores of both are close to each other. The silhouette score of K-Means came out to be comparable to agglomerative. Pie chart of K-Means:

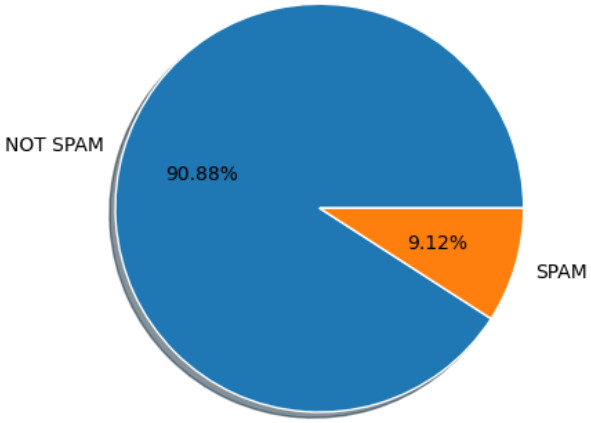

Fig: Predictions pie chart by K-Means

Pie chart of agglomerative clustering:

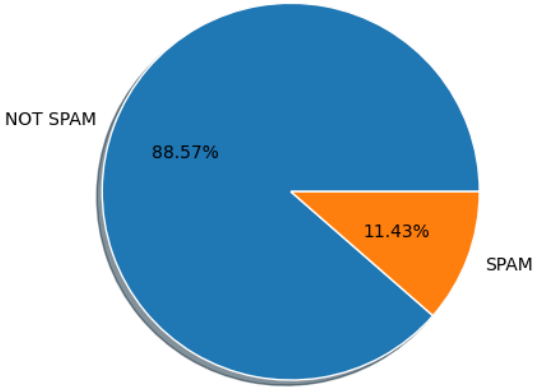

Fig: Predictions pie chart by agglomerative clustering

Conclusion:

The project aimed to classify YouTube comments into spam or non-spam categories, outlining the necessary steps to detect spam comments in YouTube videos using machine learning techniques. The machine learning pipeline included data fetch, data preprocessing, feature selection, vectorization, and text preprocessing to derive an appropriate dataset for the ML Model. CountVectorizer was used for vectorization, and the NLTK library was used for text preprocessing. The dataset consisted of comments from four YouTubers. The primary objective was to identify a cluster that contains all the spam comments and fix the issue.