Mini Project: Sentiment Analysis of YouTube Comments Using CRISP-DM

TEAM MEMBERS:

VEERAPU RAJU----------BU22CSEN0100246

CHINTHA KRISHNA BALAJI-------BU22CSEN0101063

BOMMANA HARSHITHA----------BU22CSEN0101289

TALARI LOKESH KUMAR----------BU22CSEN0101360

C. NAYUM AKTHAR ----------BU22CSEN0101361

D.NIKHIL REDDY----------BU22CSEN0101541

Machine Learning Techniques and Applications

GITAM (DEEMED TO BE UNIVERSITY), BENGALURU CAMPUS

October 2024

1. Problem Statement and Business Need

Problem Statement:

In the rapidly growing world of social media, YouTube has become one of the most popular platforms for video sharing. Content creators on YouTube rely on user engagement to gauge the success of their videos. With thousands of comments posted on popular videos, manually analyzing the sentiment (positive, neutral, or negative) of each comment becomes an overwhelming task.

This project aims to use sentiment analysis on YouTube comments to automatically classify them into sentiment categories. Sentiment analysis will provide actionable insights to content creators by helping them understand how their audience feels about their content.

Business Need:

Sentiment analysis is crucial for content creators and businesses who want to:

Enhance audience engagement: Understanding user feedback helps creators tailor their content to audience preferences.

Improve content quality: Negative feedback can pinpoint areas that need improvement.

Drive decision-making: Positive sentiment trends can help guide future content strategies.

Using machine learning models to automatically analyze comments can save time, improve response rates, and provide meaningful insights for business decisions.

2. Data Loading and Pre-processing Steps

Data Loading:

The dataset used for this project contains YouTube comments data, which was sourced from [Kaggle/UCI Machine Learning Repository]. The dataset includes the following key features:

comment_text: The text of the comment.

likes: Number of likes the comment received.

views: Total views on the associated video.

dislikes: Number of dislikes.

comment_author: Name of the person who commented.

upload_date: Date the comment was posted.

The target variable, Sentiment, is derived from the comment_text using a sentiment analysis library like TextBlob, which provides a polarity score.

Pre-processing Steps:

Data pre-processing is a vital step in preparing the dataset for machine learning models. The following steps were performed:

Handling Missing Values: Missing values in comment_text were checked and removed since comments are the key input for sentiment analysis. Without the text, sentiment classification would be impossible.

Text Pre-processing:

Tokenization: The comments were split into individual words for analysis.

Stop-word Removal: Common words (e.g., "is", "the", "and") that do not contribute meaning were removed.

Stemming and Lemmatization: Words were reduced to their base forms (e.g., "running" becomes "run") to unify word variants.

Feature Scaling: Numerical features like likes and views were scaled using MinMaxScaler to bring them within a uniform range, ensuring that larger values do not disproportionately influence the model.

Encoding Categorical Variables: Categorical variables such as comment_author were encoded using Label Encoding to convert text data into numerical form.

3. Model Creation and Algorithm Choice

Algorithm Selection:

For this project, **Logistic Regression** was selected as the primary machine learning model. Logistic regression is widely used for classification tasks, particularly binary and multi-class classification problems, such as sentiment analysis. The decision to use Logistic Regression was based on the following factors:

Simplicity: Logistic Regression is straightforward to implement and interpret, making it ideal for a first-pass model.

Efficiency: It performs well on smaller datasets and can be trained quickly without the need for extensive computational resources.

Interpretability: The coefficients of the logistic regression model provide clear insight into how each feature (or word) contributes to the predicted sentiment class.

Text Classification: Logistic Regression has been shown to perform well on text classification tasks, making it a good fit for sentiment analysis.

Model Training:

The dataset was split into a training set (80%) and a testing set (20%) to ensure the model's ability to generalize to new, unseen data.

Training Set: The training set was used to train the logistic regression model by fitting the relationships between the text features (vectorized comments) and the target sentiment labels.

Testing Set: The testing set was held out to evaluate the performance of the trained model on unseen data.

Hyperparameters such as max_iter (maximum iterations) were tuned to ensure the model converged properly.

4. Model Performance Evaluation

Evaluation Metrics:

To evaluate the performance of the model, the following metrics were used:

Accuracy:

Accuracy measures the percentage of correct predictions out of all predictions made.

Precision:

Precision indicates how many of the predicted positive instances were actually positive. High precision means that the model makes few false positive predictions.

Recall:

Recall measures how many actual positive instances were correctly identified by the model. High recall means the model misses few positive instances.

F1 Score:

F1 Score is the harmonic mean of precision and recall, balancing the two metrics. It is especially useful when dealing with imbalanced classes.

Confusion Matrix:

A confusion matrix was used to visualize the performance of the model by showing how many instances of each sentiment class were correctly classified, and how many were misclassified.

Results:

After evaluating the model on the testing set, the following performance results were observed:

Accuracy: The logistic regression model achieved an accuracy of [insert accuracy %] on the test set.

Confusion Matrix: The confusion matrix showed that [number] positive comments, [number] neutral comments, and [number] negative comments were correctly classified, while [number] instances were misclassified.

Model Strengths and Weaknesses:

Strengths:

The logistic regression model performed well in classifying positive and neutral comments, achieving high precision and recall for these classes.

Weaknesses:

The model struggled with distinguishing between neutral and negative comments, likely due to the subtle differences in sentiment expression within these classes. This suggests that the dataset may require more examples of negative comments, or that more complex models (such as neural networks) could yield better results.

Conclusion:

The sentiment analysis model developed in this project successfully classified YouTube comments into positive, neutral, and negative sentiment categories with a reasonable level of accuracy. While the logistic regression model provided strong baseline results, further improvements could be made by experimenting with more advanced models like Random Forest or deep learning techniques such as neural networks. Overall, the project demonstrates the potential for automating sentiment analysis to provide valuable insights for content creators and businesses.