**AIDS Lab**

**EXPERIMENT NO. 5**

**Aim**: To build a Cognitive Analytics for personalization of Customer service application/Insurance/Healthcare Application/Smarter Cities/Government etc.

**Theory**:

Sentiment analysis is a machine learning technique that can analyze comments about your brand and your competition for opinion polarity (positive, negative, neutral, and beyond). With 2 billion monthly active users, YouTube is the 2nd most popular social media platform and, based on total visitors and page views, 2nd only to Google as the world’s most visited website. Yet, it goes largely untapped when it comes to opinion mining and social listening.

YouTube users continue to grow by the day, and YouTube comments can provide a wealth of information and actionable insights for any brand. All those unguarded opinions and open customer feedback are free for the taking when you have the right tools in place.

Process of sentiment analysis on Youtube comments:

1. Data Labelling: The dataset is unlabelled because of using API, you can only extract the comments but not the polarity. Polarity is something that can identify the emotion of a particular sentence by using the words present in that.
2. Lowercase the comments: It becomes important as it makes the data more effective to produce a better result but if not converted then the system treats them as two different words which might be redundant information and leads to producing different results than the desired one.
3. Strip: Now using strip() method by python, remove all the trailing spaces from the comments.
4. Stop words: It contains words of less importance, i.e., commonly used words like is, am, the, are, a, etc. as they don’t add any fruitful information that is required for analysis. So, by removing these kinds of words, our data become more concise as having fewer features but significant.
5. Data splitting: As of now, you are working with the one dataset, but for making predictions of sentiment analysis or can say that for testing the accuracy of the model trained there will require a test data set/validation data set.
6. Feature extraction from Text Data: Now it’s time to extract features out of text data, i.e. conversion to integer values or floating-point values as a machine learning model can’t be applied directly to the text data.

Now you have got the pre-processed data consisting of training and testing dataset. So the next step is the selection of an appropriate machine learning algorithm for classification. For the sentiment classification into two classes positive and negative, Logistic Regression, a machine learning classification algorithm will be used and achieve an accuracy score higher than and more reliable than accuracy.

**Code and Output**:

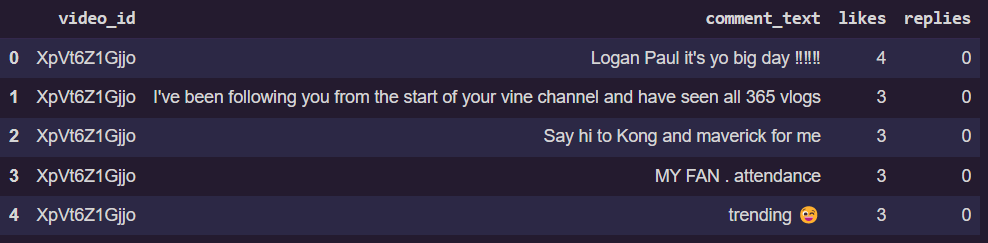
This dataset consists of the Top 200 trending videos from youtube and characteristics of those videos like number of likes , number of comments ,number of subscribers etc. We can use these data to know which sort of videos are trending and how much time does it take for the videos to come to treding statuts once they are uploaded and how long will they be in that position and what are comments on the videos etc.

| import numpy as np import pandas as pd import matplotlib.pyplot as plt import seaborn as sns import re %matplotlib inline import warnings warnings.filterwarnings("ignore") |
| --- |

| pd.set\_option('display.max\_columns', None) US\_comments = pd.read\_csv('UScomments.csv', error\_bad\_lines=False) |
| --- |

Analysis and Data Cleaning

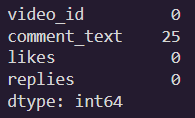
| US\_comments.head() |
| --- |



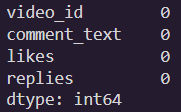
| US\_comments.shape |
| --- |



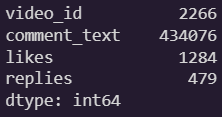
| US\_comments.isnull().sum() |
| --- |



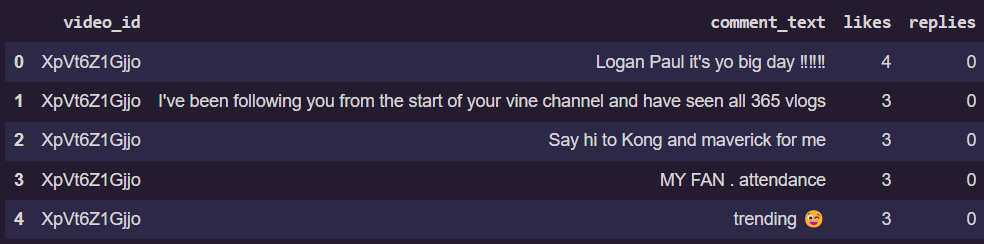
| US\_comments.dropna(inplace=True) US\_comments.isnull().sum() |
| --- |



| US\_comments.nunique() |
| --- |



| US\_comments.drop(41587, inplace=True) US\_comments = US\_comments.reset\_index().drop('index',axis=1) US\_comments.likes = US\_comments.likes.astype(int) US\_comments.replies = US\_comments.replies.astype(int) US\_comments.head() |
| --- |



Removing Punctuations, Numbers and Special Characters

| US\_comments['comment\_text'] = US\_comments['comment\_text'].str.replace("[^a-zA-Z#]", " ") |
| --- |

Removing short words

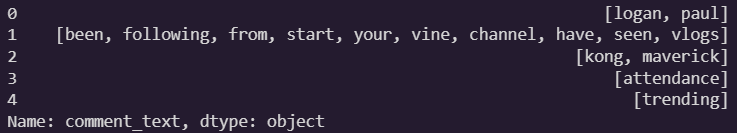
| US\_comments['comment\_text'] = US\_comments['comment\_text'].apply(lambda x: ' '.join([w for w in x.split() if len(w)>3])) |
| --- |

Changing the text to lowercase

| US\_comments['comment\_text'] = US\_comments['comment\_text'].apply(lambda x:x.lower()) |
| --- |

Tokenization

| tokenized\_tweet = US\_comments['comment\_text'].apply(lambda x: x.split()) tokenized\_tweet.head() |
| --- |

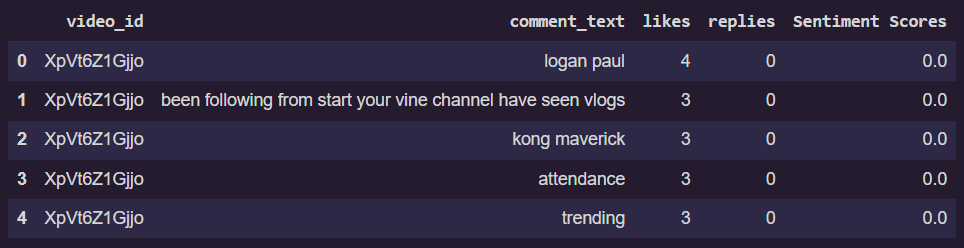


Sentiment Analysis on the US Comments

| import nltk nltk.download('vader\_lexicon') from nltk.sentiment.vader import SentimentIntensityAnalyzer sia = SentimentIntensityAnalyzer() |
| --- |

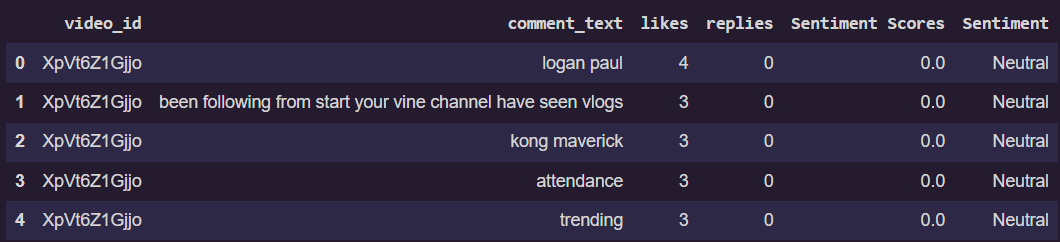
Setting The Sentiment Scores

| US\_comments['Sentiment Scores'] = US\_comments['comment\_text'].apply(lambda x:sia.polarity\_scores(x)['compound']) US\_comments.head() |
| --- |

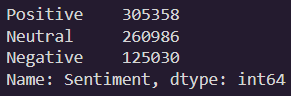


Classifying the Sentiment scores as Positive, Negative and Neutral

| US\_comments['Sentiment'] = US\_comments['Sentiment Scores'].apply(lambda s : 'Positive' if s > 0 else ('Neutral' if s == 0 else 'Negative')) US\_comments.head() |
| --- |



| US\_comments.Sentiment.value\_counts() |
| --- |



**Conclusion**:

Thus we studied an overview of how to perform sentiment analysis on Youtube comments and implemented Cognitive Analytics for personalization of Customer service applications.