### Screen Shots/Graphs/Results

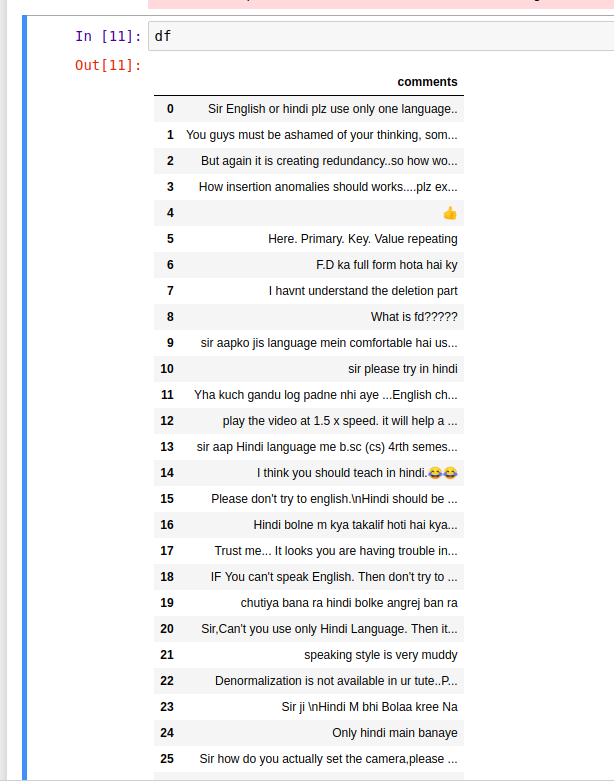
1)

**Dataset extracted:**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Video ID | Number of Likes | Number of Dislikes | Number of Views | Publish Date | Duration | Upload Time(UTC) | Video Quality |
| YJqkG9BV6hU | 1.2K | 255 | 260K | 02/12/2016 | 0:36:00 | 09:04:13 | 720p |
| UrYLYV7WSHM | 9.5K | 455 | 1M | 14 Aug 15 | 0:19:01 | 20:54:06 | 720p |
| InF1bpXCn0 | 714 | 63 | 62K | 29 Mar 18 | 0:53:30 | 14:09:33 | 720p |
| 7B9FnIIIsQc | 724 | 66 | 110K | 4 Sep 16 | 1:02:35 | 13:29:39 | 720p |
| K\_Rr5B9jXyo | 677 | 134 | 221K | 29 Oct 15 | 0:37:22 | 08:36:54 | 720p |
| xoTyrdT9SZI | 2.3K | 47 | 282K | 07/10/17 | 08:11 | 16:09:05 | 1080p |
| oylHRgBDfNc | 4.3K | 170 | 455K | 23/09/16 | 13:34 | 13:05:00 | 480p |
| wRoABkgPocI | 1.1K | 105 | 200K | 07/07/16 | 07:40 | 12:26:05 | 1080p |
| 5GDTIUVlHB8 | 502 | 6 | 25k | 16/02/18 | 12:51 | 07:20:00 | 720p |
| qTR8QnYXHvQ | 105 | 2 | 15K | 25/01/18 | 08:53 | 10:20:00 | 720p |

2)

Data Scrapped from web and used as dataframe of comments:



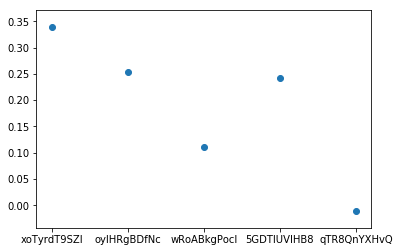
3)

Datasets formed into dataframes :



4)

Polarity graph of first five videoids



5)

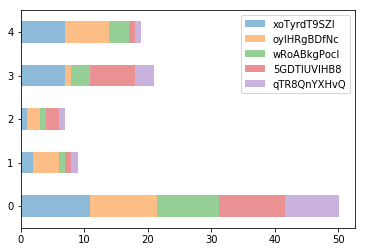
Factors describing each video[0-Comments

1-Likes

2-Duration

3-Quality

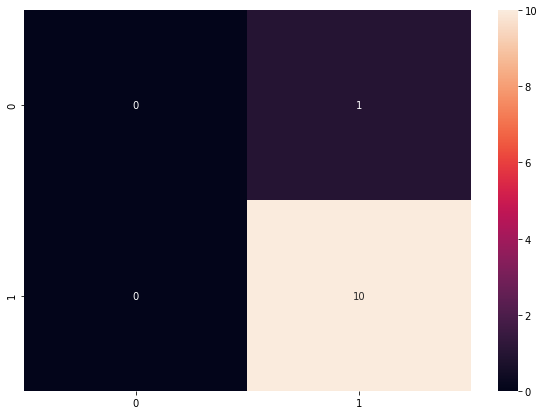
4-Views]



6)Giving colors according to the rating of videos for videos in [1]



7)Accuracy Heatmap



**Validation checks :**

**Providing credits to comments**

Accuracy is one metric for evaluating classification models. Informally,accuracy is the fraction of predictions our model got right. Formally, accuracy has the following definition.

Accuracy = (No. Of correct prediction)/(Total no of prediction)

Accuracy = (TP+TN)/(TP+TN+FP+FN)

Where TP= True Positives,TN = True Negatives, FP = False Positives, and FN = False Negatives.

Here According to Fig[7]

TP=10,TN=0,FP=0,FN=1.

This data is trained on less data therefore it has less features to learn from but it is validated for this small data.

Here the Accuracy =(10+0)/(1+10+0+0)

=90.90%

**Proof that we are saving time of users:**

Based on Research done on first five videos for the topic:Normalization in Dbms:

When we tested our program with 3 users ,the average time taken to find the best video was taking around 17 min roughly which includes the skipping of non-intersting video based on normal psychology .But we give the usr the scale made by us to judge the ratings of the video then directly browsing the first video with highest rating + still having a trial with othe decreased rating videos , the user took 6.5 min to obsrve the best video. Therefor we saved our user’s time by a factor of 10.5 min .

Its a rough estimate of timmings as the software is not tested with variety of users.

Probably ,the user was getting more time to reach the best video and sometimes the user didn’t even scrolled enough to reach the best video.

### Coding:

**Source Code for finding the polarity of all the comments of one video individually**

#!/usr/bin/env python

# coding: utf-8

# In[1]:import urllib

# imorting urllib to use any url in code

# In[2]:import json

# In[3]:from pandas.io.json import json\_normalize

# # Normalize semi-structured json data into a flat table

# In[4]:url = "https://www.googleapis.com/youtube/v3/commentThreads?key=AIzaSyBwdx2rVaePw3a5ls0tMnF3Z\_dea\_c-NLM&textFormat=plainText&part=snippet&videoId=5GDTIUVlHB8&maxResults=100"

----------------------------------------------------[skipped]

# In[11]:df

# In[12]:df.to\_csv("video4.csv", index=False)

# In[13]:import csv

# In[14]:from textblob import TextBlob

# In[15]:infile = '/home/prachi/PycharmProjects/Mini\_project/video4.csv'

# In[16]:print("polarity")

with open(infile, 'r') as csvfile:

rows = csv.reader(csvfile)

for row in rows:

sentence = row[0]

blob = TextBlob(sentence)

print (blob.sentiment.polarity,",")

#

# In[17]:import pandas

df\_pol = pandas.read\_csv('video4\_pol.csv')

print(df\_pol)

# In[18]:df\_pol.sum(axis = 1, skipna = True)

# In[19]:a=[0.0 ,0.0 ,0.4 ,0.0 ,0.8 ,0.7500000000000001 ,0.0 ,0.0 ,0.2 ,0.0 ,0.0 ,0.0 ,0.2 ,0.78 ,0.15 ,0.0 ,1.0 ,0.8 ,1.0 ,0.3333333333333333 ,0.2904761904761905 ,0.0 ,0.0 ,0 ,0.1 ,0.4 ,-0.3214285714285714 ,0.0 ,0.20555555555555557 ,0.0 ,0.0 ,0.3333333333333333 ,

0.0 ,0.2 ,0.7 ,0.0 ,0.30000000000000004 ,0.0 ,0.0 ,0.3816666666666667 ,

0.18333333333333335 ,0.16666666666666666 ,0.78 ]

# In[21]:print(avg=(sum(a)/43))

# In[22]:avg = sum(a)/43

# In[23]:print(avg)

# 0.23564968623108157

**Source code for finding the accuracy of our model**

#!/usr/bin/env python

# coding: utf-8

# In[64]:

import numpy as np

import pandas as pd

import sklearn

import matplotlib as plt

import matplotlib.pyplot as plt

import seaborn as sns

-------------------------------------------------------[skipped]

# In[70]:corpus

# In[73]:

y = dataset.iloc[:, 2].values

y = y.astype(int)

y

# In[78]:

# Creating the Bag of Words model

from sklearn.feature\_extraction.text import CountVectorizer

# experiment with to get better results

cv = CountVectorizer(max\_features = 1500)

# X contains corpus (dependent variable)

X = cv.fit\_transform(corpus).toarray()

X

# y contains answers if review

# is positive or negative

y = dataset.iloc[:, 2].values

y = y.astype(int)

y

# In[79]:

# Splitting the dataset into

# the Training set and Test set

from sklearn.model\_selection import train\_test\_split

# experiment with "test\_size"

# to get better results

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size = 0.25)

------------------------------------------------------------

---------------------------[skipped]

# Predicting the Test set results

y\_pred = model.predict(X\_test)

# Making the Confusion Matrix

from sklearn.metrics import confusion\_matrix

cm = confusion\_matrix(y\_test, y\_pred) #only some steps---------------[skipped]

**Future Scope:**

As currently we are predicting over the fetched data,So we don’t have any mechanism to trace the fraud views,likes,subscribers etc.Therefore,in future we hope to give authenticate+best analyzed predictions produced over the videos by using fraud detection method.