A

Mini Project report on

SENTIMENT ANALYSIS REVIEWS

Submitted in partial fulfillment of the requirements for the

Award of the degree of

BACHELOR OF TECHNOLOGY
IN
INFORMATION TECHNOLOGY

By

Padamati Jashwanth Reddy	**(16881A1233)**
D. Ram Charan	**(16881A1212)**
Sreerangam Pradeep	**(16881A1250)**

Under the Guidance of

Dr. T. Raghunadha Reddy **M.Tech,Ph.D.**

Associate Professor

Department of Information Technology

DEPARTMENT OF INFORMATION TECHNOLOGY

VARDHAMAN COLLEGE OF ENGINEERING

(AUTONOMOUS)

(Affiliated to JNTUH, Approved by AICTE and Accredited by NBA)

Shamshabad - 501 218, Hyderabad

2016-2020

VARDHAMAN COLLEGE OF ENGINEERING

(AUTONOMOUS)

Shamshabad - 501 218, Hyderabad

DEPARTMENT OF INFORMATION TECHNOLOGY

CERTIFICATE

This is to certify that the project report entitled, “**Sentiment Analysis Reviews”,** done by **Padamati Jashwanth Reddy (16881A1233), D. Ram Charan (16881A1212) and Sreerangam Pradeep (16881A1250)** Submitted to the department of Information Technology in partial fulfillment of the requirements for the Degree of **BACHELOR OF TECHNOLOGY** in **Information Technology** from **Vardhaman College of Engineering (AUTONOMOUS), Hyderabad**, during the period 2016-2020. It is certified that he/she has completed the project satisfactorily.

Signature of Supervisor:
Dr. T.Raghunadha Reddy M.Tech,Ph.D.
Associate Professor,
Dept. of Information Technology,
Vardhaman College of Engineering,
Hyderabad

Head of the Department:
Dr. M. Gopi Chand M.Tech,Ph.D..
Professor & Head,
Dept. of Information Technology,
Vardhaman College of Engineering,
Hyderabad

ACKNOWLEDGEMENT

The satisfaction that accompanies the successful completion of the task would be put in complete without the mention of the people who made it possible, whose constant guidance and encouragement crown all the efforts with success.

We wish to express our deep sense of gratitude to **Dr. T. Raghunadha Reddy**, Associate Professor, Department Information Technology, Vardhaman College of Engineering, for his able guidance and useful suggestions, which helped us in completing the project work, in time.

We are particularly thankful to **Dr. M. Gopi Chand**, Professor& Head of Department, Information Technology for his guidance, intense support and encouragement, which helped us to mold our project into a successful one.

We show gratitude to our honorable Principal **Dr. S. Sai Satyanarayana Reddy**, for providing all the facilities and support.

We avail this opportunity to express our deep sense of gratitude and heartfelt thanks to all the members of management, for providing congenial atmosphere to complete this project successfully.

We also thank all our teaching and non-teaching staff members of Information Technology department for their valuable support and generous advice. Finally thanks to all our friends and family members for their continuous support and enthusiastic help.

Padamati Jashwanth Reddy	**(16881A1233)**
D. Ram Charan	**(16881A1212)**
Sreerangam Pradeep	**(16881A1250)**

ABSTRACT

Rise of fraudulent products on the internet has been observed since a while and finding the right and relevant product that suits the needs of a consumer has become extremely strenuous. This project aims to provide a solution to this problem. On a road to solve it, we want to develop a solution on a specific site. For this purpose, we have selected YouTube as our target. The analysis has to be done based on user feedback. To achieve this, the user should provide the link for which he wants to analyze. This link is the target of our project. In order to get proper feedback, we consider the comments, likes, and dislikes of that particular video.

To get the data for the analysis we use web-scrap concept. By web-scraping, the contents of the link provided by the user (target) we obtain all the contents of that particular webpage including the comments, like and dislikes. We scrap contents using Selenium. Selenium is one of the most widely used open source Web UI (User Interface) automation testing suite. Selenium supports automation across different browsers, platforms and programming languages. Selenium helps us to scrape the data dynamically. We then store the scraped content in Python. The data that is scraped and stored is being analyzed to determine whether the data is a positive response or a negative one.

This sentiment analysis is done using Python Language, in which we use packages such as Text Blob to analyze the data and to perform sentiment analysis. By using this all the data is analyzed and the respective result is stored. Based on this result a relevant response is given to the user specifying whether the given link contains a video of positive or negative response. By this the user can decide whether to watch that particular video or not and save time by ignoring negative response video.

CONTENTS

Chapter –1

INTRODUCTION

1.1 Motivation:

Rise of fraudulent products on the internet has been observed since a while and finding the right and relevant product that suits the needs of a consumer has become extremely strenuous. This project aims to provide a solution to this problem. On a road to solve it, we want to develop a solution on a specific site. For this purpose, we have selected Youtube as our target. The analysis has to be done based on user feedback. To achieve this the user should provide the link for which he wants to analyze. This link is the target of our project. In order to get proper feedback, we consider the comments, likes, and dislikes of that particular video.

1.2 Problem Statement:

In the present day situation we have very less time to know the review of any large video. For example consider any educational video which is of duration more than an hour. For knowing the review of the video would take more time to know the review of that particular video. This is the problem being faced by many end users which encouraged us to take up this issue and give a solution to it.

1.3 Objectives of the Project:

The main objective of the project is to develop a prototype of Sentiment Analysis Reviewer using technologies, namely, Flask, Selenium and Text Blob . Sentiment Analysis Reviewer can discover and extract the reviews associated with any online Youtube Video Link. It can answer complex queries for knowing the review of the video and thus assist the users to make intelligent decisions which traditional decision support systems cannot. By providing effective review, it also helps to reduce the time taken to know the review of the video. To enhance visualization and ease of interpretation, it displays the results in graphical forms.

1.4 Scope of the project:

Here the scope of the project is that integration of review analysis support with computer-based users comments could reduce human effort to know the review of the video. This suggestion is promising as data modeling and analysis tools.The main objective of this project is to develop a prototype Sentiment Analysis Reviewer using technologies, namely Flask, Selenium, Text Blob. So it provides effective reviews, it also helps to reduce time to know the review of the video and also enhances visualization and ease of interpretation. With immense knowledge and accurate data in that field. Such work brings together all available past and current data, as a basis on which to develop reasonable expectations about the future.

Chapter - 2

LITERATURE SURVEY

2.1 EXISTING SYSTEM:

There are systems which use scraping content from the website using Selenium, Beautiful Soup and so on, but there is no system which is used for giving review to the scrapped content. There are many sentiment analysis techniques already present which are generally used to process a group of data given and generate the result after processing the data whether it is positive or negative or neutral. For example there is this analysis where a statement is categorised based on the number of positive and negative words.

2.3 DISADVANTAGES OF EXISTING SYSTEM

Following are the disadvantages of the existing system which perform sentiment analysis:

- ❖ Can not process the information dynamically as per our need.
- ❖ Can not show the results in a visual manner so customers may get confused due to the text as visual representation gives more clarity.

2.2 PROPOSED SYSTEM:

As the problem statement describes the problem for which we came up with a solution where the end users will be getting the review of the youtube video for which the end user wants to know the review. For this we consider the youtube video link given by the end user and open that video in separate window and start scraping the content of the video. Here we scrap the comments,likes and dislikes which helps us to give the review report. For this we used the technologies like Flask, Selenium and Text blob.

Chapter – 3
SYSTEM ANALYSIS

3.1 SOFTWARE REQUIREMENTS SPECIFICATION:

3.1.1 Hardware requirements:

Processor	:	Any Update Processor
Ram	:	4 GB
Hard Disk	:	500 MB
Server	:	1

3.1.2 Software requirements:

Operating System	:	Windows 7 / Windows 8 /Windows 10 / Ubuntu.
Technology	:	Python Interpreter
Web Technologies	:	Html, Html-5, JavaScript, CSS
Web Server	:	Flask

3.2 Technologies And Flowchart:

3.2.1 Introduction To Python:

Python is a high-level, interpreted programming language developed in the late 1980s by Guido van Rossum at the National Research Institute for Mathematics and Computer Science in the Netherlands. It was mainly developed for emphasis on code readability and its syntax allows programmers to express concepts in few lines of code.

Python is derived from programming languages such as ABC, Modula 3, small talk, Algol-68. Van Rossum picked the name Python for the new language from a TV show, Monty Python's Flying Circus.

3.2.1.1 FEATURES OF PYTHON

1. Easy to learn and use: Python is easy to learn and use. It is developer friendly and high-level programming language.
2. Expressive Language: Python language is more expressive means that it is more understandable and readable.

3. Interpreted Language: Python is an interpreted language i.e. interpreter executes the code line by line at a time. This makes debugging easy and thus suitable for beginners.
4. Cross-platform Language: Python can run equally on different platforms such as Windows, Linux, Unix and Macintosh etc. So, we can say that Python is a portable language.
5. Free and Open Source: Python language is freely available at https://www.python.org/. The source code is also available. Therefore it is open source.
6. Object-Oriented Language: Python supports object-oriented language and concepts of classes and objects come into existence.
7. Extensible: It implies that other languages such as C/C++ can be used to compile the code and thus it can be used further in our python code.
8. GUI Programming Support: Graphical user interface can be developed using Python.

3.2.1.2 APPLICATIONS OF PYTHON

There are many applications of Python some of them are as follows:

1. Web Development.
2. Machine Learning.
3. Data Analysis.
4. Scripting.

3.2.3 Python Interpreter:

The default encoding for a Python source file is UTF-8. This is a *Unicode Standard variable-width character encoding;* it can encode 1,112,064 valid code points in Unicode using up to four 8-bit bytes. Using this encoding, we can use characters of most languages – we can use these in string literals, comments, and identifiers [1].

Since the standard library makes use of ASCII characters only, you must declare the use of this encoding to your editor. This is to ensure that all such characters display without a problem. The font should be such that supports all characters in the file.

On your machine, you can find your interpreter at an address like:

C:\Python36

Or it may reside on the location you selected at the time of installation. Add path using this command:

```
set path=%path%;C:\python36
```

On Windows, when you want to run the Python interpreter in the shell, you can type the following:

```
$python
```

To get out of the interpreter in disassembling the Bytecode shell, you can type:

```
>>> quit()
```

if you want, you can save your Python code as a script and execute it using the interpreter:

```
$python demo.py
```

3.2.4 Flask:

Flask App routing. App routing is used to map the specific URL with the associated function that is intended to perform some task. It is used to access some particular page like FlaskTutorial in the web application.

Flask offers suggestions, but doesn't enforce any dependencies or project layout. It is up to the developer to choose the tools and libraries they want to use. There are many extensions provided by the community that make adding new functionality easy.

Routing Modern web applications use meaningful URLs to help users. Users are more likely to like a page and come back if the page uses a meaningful URL they can remember and use to directly visit a page. Use the route() decorator to bind a function to a URL [2].

To build a URL to a specific function, use the url_for() function. It accepts the name of the function as its first argument and any number of keyword arguments, each corresponding to a variable part of the URL rule. Unknown variable parts are appended to the URL as query parameters.

Web applications use different HTTP methods when accessing URLs. You should familiarize yourself with the HTTP methods as you work with Flask. By default, a route only answers to GET requests. You can use the methods argument of the route() decorator to handle different HTTP methods.

3.2.5 Selenium:

Selenium is one of the most widely used open source Web UI (User Interface) automation testing suite. Selenium supports automation across different browsers, platforms and programming languages.

Selenium can be easily deployed on platforms such as Windows, Linux, Solaris and Macintosh.It supports a variety of programming languages through the use of drivers specific to each language.Languages supported by Selenium include C#, Java, Perl, PHP, Python and Ruby.Currently, Selenium Web driver is most popular with Java and C#. Selenium test scripts can be coded in any of the supported programming languages and can be run directly in most modern web browsers. Browsers supported by Selenium include Internet Explorer, Mozilla Firefox, Google Chrome and Safari.

The primary new feature in Selenium 2.0 is the integration of the WebDriver API. WebDriver is designed to provide a simpler, more concise programming interface in addition to addressing some limitations in the Selenium-RC API. Selenium-WebDriver was developed to better support dynamic web pages where elements of a page may change without the page itself being reloaded [4]. WebDriver's goal is to supply a well-designed object-oriented API that provides improved support for modern advanced web-app testing problems.

Selenium-WebDriver makes direct calls to the browser using each browser's native support for automation. How these direct calls are made, and the features they support depends on the browser you are using. Information on each 'browser driver' is provided later in this chapter.

For those familiar with Selenium-RC, this is quite different from what you are used to. Selenium-RC worked the same way for each supported browser. It 'injected' javascript functions into the browser when the browser was loaded and then used its javascript to drive the AUT within the browser. WebDriver does not use this technique. Again, it drives the browser directly using the browser's built in support for automation.

3.2.6 NATURAL LANGUAGE PROCESSING:

Natural language processing (NLP) is a subfield of computer science and artificial intelligence concerned with the interactions between computers and human (natural) languages. It is used to apply machine learning algorithms to text and speech.

Generally, we use NLP to create systems like speech recognition, Document summarization, Machine translation, Spam detection, Named entity recognition, Question answering, Autocomplete, Predictive text and so on.

Some of the best examples of Natural language processing applications are Cortana, Siri, google assistant.

3.2.7 SENTIMENT ANALYSIS:

Sentiment Analysis is the process of determining whether a piece of writing is positive, negative or neutral. A sentiment analysis system for text analysis combines Natural language processing (NLP) and machine learning techniques to assign weighted sentiment scores to the entities, topics, themes and categories within a sentence or phrase [5].

Usually, besides identifying the opinion, these systems extract attributes of the expression like

- ❖ Polarity: if the speaker expresses a positive or negative opinion,
- ❖ Subject: the thing that is being talked about,
- ❖ Opinion holder: the person, or entity that expresses the opinion.

3.2.8 TextBlob:

TextBlob is a Python (2 and 3) library for processing textual data. It provides a simple API for diving into common natural language processing (NLP) tasks such as part-of-speech tagging, noun phrase extraction, sentiment analysis, classification, translation, and more.It stands on the giant shoulders of NLTK and pattern, and plays nicely with both.

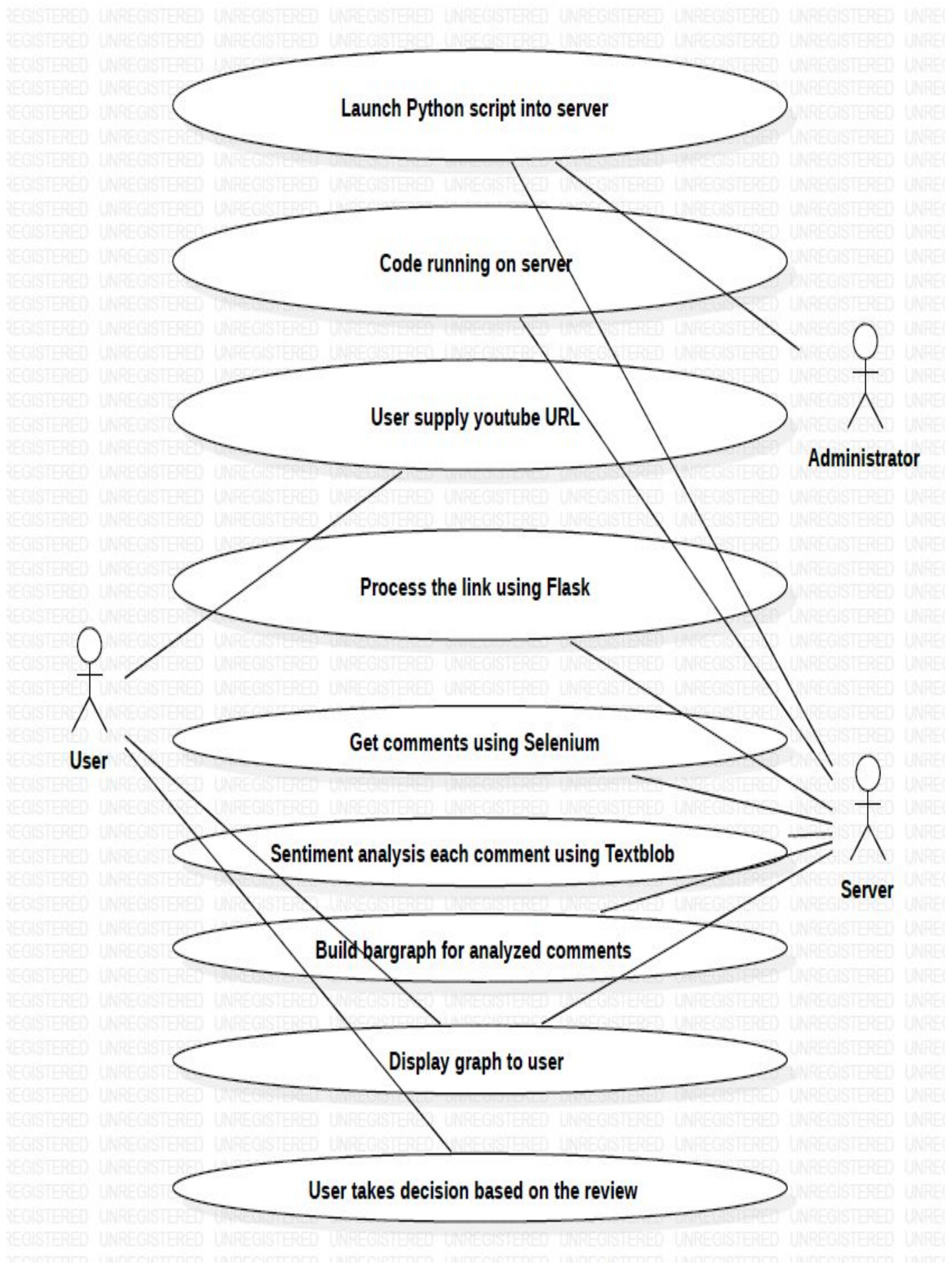
The sentiment property returns a namedtuple of the form Sentiment(polarity, subjectivity). The polarity score is a float within the range [-1.0, 1.0]. The subjectivity is a float within the range [0.0, 1.0] where 0.0 is very objective and 1.0 is very subjective [3].

Each word in TextBlob.words or Sentence.words is a Word object (a subclass of unicode) with useful methods, e.g. for word inflection. Words can be lemmatized by calling the lemmatize method.

New in version 0.5.0. The textblob.sentiments module contains two sentiment analysis implementations, PatternAnalyzer (based on the pattern library) and NaiveBayesAnalyzer (an NLTK classifier trained on a movie reviews corpus). The default implementation is PatternAnalyzer, but you can override the analyzer by passing another implementation into a TextBlob’s constructor.

3.3 UML Diagram:

3.3.1 Use Case Diagram:


Launch Python script into server
Code running on server
User supply youtube URL
Administrator
Process the link using Flask
User
Get comments using Selenium
Sentiment analysis each comment using Textblob
Server
Build bargraph for analyzed comments
Display graph to user
User takes decision based on the review

Actors Involved:

User: The user gives the YouTube link to get the required output.

Administrator: The administrator will handle all the scraped comments and likes.

Server: The server will handle the script which is executed and helps to run the whole project.

Use Cases Involved:

Launch Python script into server: Here the python script will be launched into the server.

Code running on server: The launched script will be running on the server.

User supply youtube URL: The user will supply youtube url for getting the review of the video.

Process the link using Flask: The given link will be processed using Flask.

Get comments using Selenium: The comments of the given youtube link will be scraped using Selenium.

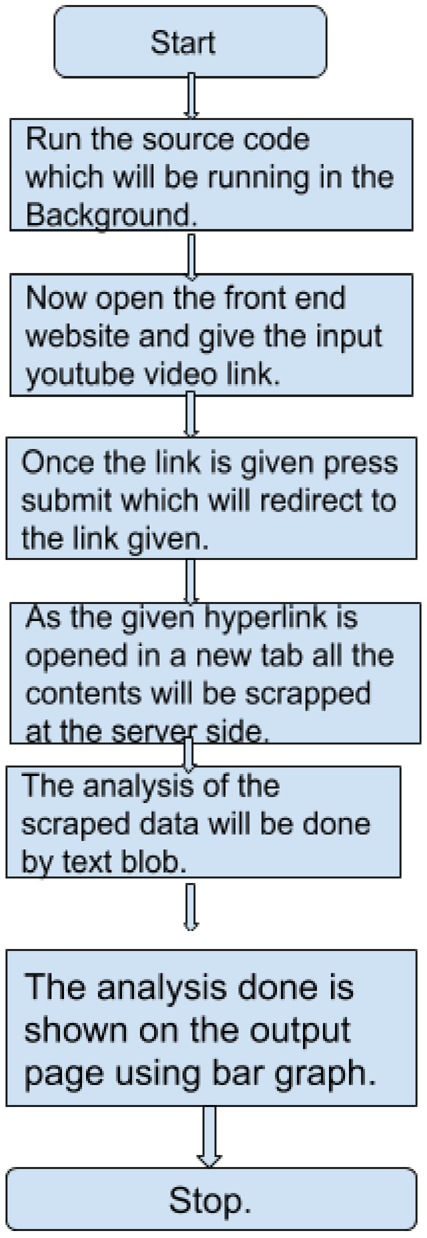
Sentiment analysis each comment using Textblob: Sentiment analysis of the scraped comments will be done using TexBlob.

Build bar graph for analyzed comments: The analyzed comments are presented using the bar graph.

Display graph to user: The built graph will be displayed to the user.

User takes decision based on the review: Based on the displayed bar graph the user takes the decision.

3.4 Flowchart:


Start
Run the source code which will be running in the Background.
Now open the front end website and give the input youtube video link.
Once the link is given press submit which will redirect to the link given.
As the given hyperlink is opened in a new tab all the contents will be scrapped at the server side.
The analysis of the scraped data will be done by text blob.
The analysis done is shown on the output page using bar graph.
Stop.

Chapter - 4
Implementation

4.1 Introduction:

As already introduced our project the main aim of our project is to provide the review of the video link given by the user. Here we use Open.html, output.html for taking the input and displaying the output. Then we have analysis.py application used to run on the server side.

4.2 Modules Design and Organization:

4.2.1 Open.html:

This is a front end web application which is used to take the given video link to the server end. The front end website consists of a textbox for inserting the video link, button to open the URL in a different tab. We used html as the programing language to develop the website [7]. The background coloring and the other designs are done using CSS [8]. We integrated both of these into one program to have a proper view of the website.

This particular application plays one of the most important roles to take the video link from the user and send it to the server end. The sent link will be taken and used as a source to scrap the comments and likes using Flask. Once the flask takes the link it gets opened in a new tab allows the server to scrap the comments.

The application consists of style tag where it contains body, headings h1,h2 , span and div tags. In body tag we use padding, text alignment, font family, background and background size. In h1, h2 we used font size, font family, webkit stroke, color, text shadow.

Later we added the required text for making the website user friendly.

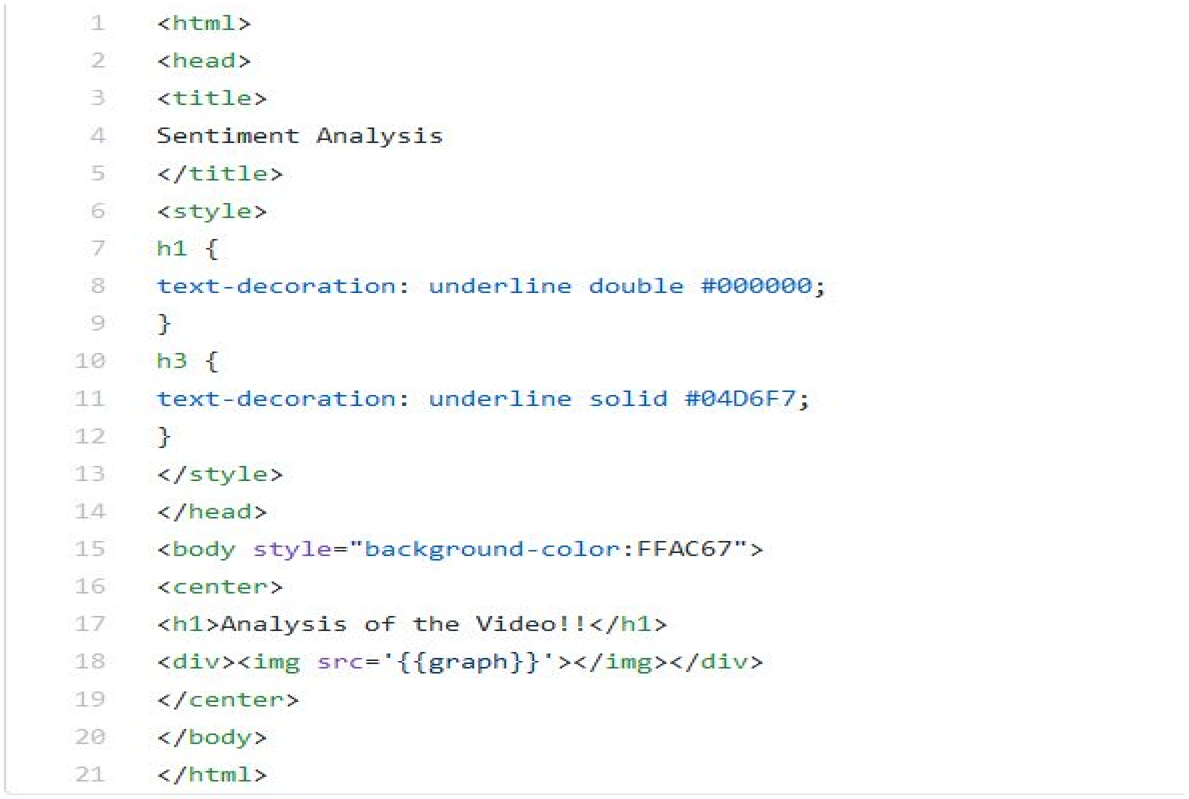
```
<html><style>
 body {
 padding-top: 80px;
 text-align: center;
 font-family: monaco, monospace;
 background: url(i2.png) 20%;
 background-size: cover;
}
h1, h2 {
 display: inline-block;
 background: ;
 color:#f2f2f2;
}
h1 {
 font-size: 35px;
 font-family:"Comic Sans MS", cursive, sans-serif;
 @-webkit-keyframes colorchange {
 0% {
   -webkit-text-stroke: 10px red;
 }
 100% {
   -webkit-text-stroke: 20px black;
 }
}

  color: white;
  Text-shadow:

      3px 3px 0 #000,
    -1px -1px 0 #000,
     1px -1px 0 #000,
     -1px 1px 0 #000,
      1px 1px 0 #000;
  color: white;
  text-shadow:
      3px 3px 0 #000,
    -1px -1px 0 #000,
     1px -1px 0 #000,
     -1px 1px 0 #000,
      1px 1px 0 #000;

 }
 #d1 {
 font-size: 30px;
 font-family:Courier New, monospace;
 -webkit-text-stroke: 1px black;
  color: white;
  text-shadow:
      3px 3px 0 #000,
    -1px -1px 0 #000,
     1px -1px 0 #000,
     -1px 1px 0 #000,
      1px 1px 0 #000;

 }

div{
height:100%;
width:70%;
}
</style>
<body style="background-color:BDFC65">
<div>
```

```
        <form method="POST" action="http://127.0.0.1:5000/res">

    <h1>Welcometo<br><span>Suggester</span></h1>
    <br><br>
        <h2>A Sentiment Analysis of Reviews</h2><br>
        <span id="d1">Target URL:</span><input type="text" size="50" id="url" name="url"><br><br>
        <button style="color:brown" onclick="redirect()">Analyze</button>
        </div>
        </body>
        </html>
```

4.2.2 Output.html:

This is a front end web application which is used to display the review in the form of the bar graph. The front end website of a bar graph which displays the output. We used html as the programing language to develop the website [7]. The background coloring and the other designs are done using CSS. We integrated both of these into one program to have a proper view of the website.

This particular application plays one of the most important roles to display the output in the form of bar graphs.



```
<html>
<head>
<title>
Sentiment Analysis
</title>
<style>
h1 {
text-decoration: underline double #000000;
}
h3 {
text-decoration: underline solid #04D6F7;
}
</style>
</head>
<body style="background-color:FFAC67">
<center>
<h1>Analysis of the Video!!</h1>
<div><img src='{{graph}}'></img></div>
</center>
</body>
</html>
```

4.2.3 Analysis.py:

This is a python code which is executed and ran in the server side. This code helps us to establish a connection between the front end web application and the server. This contains the libraries of Flask, Matplot, TextBlob and Selenium. The flask library is used to establish a connection between the front end application and the server using the chrome driver which helps us to open the link given by the user and lets the Selenium library to scrap the contents. The Selenium driver helps us to scrap the content of the youtube video link given by the user.

The Matplot library is used to represent the output in the form of a bar graph where the review of the youtube video is categorised into strong negative, negative, neutral, positive and strong positive. Each category is represented by bars in the bar graph. The x-axis consists of the categories of the reviews and the y-axis represents the percentage of each category based on the analysis of Text Blob [3].

The text blob library is used to find out the polarity and subjectivity of the words. The weight of each word or phrase of the scraped contents will be calculated by the individual weights of them.

All these together will help us to get the required output. Now lets see the flow of the program.

First we execute the Analysis.py code which will be running in the backend [1]. Then we open the Open.html front end web application which takes the input video link from the user and the button which is displayed will help the user to start the process of generating the review by scraping the content present in the youtube video link. Which helps us to calculate the subjectivity and polarity based on the weights of the phrases and words.

As the analysis of subjectivity and polarity is done we now transfer the data of each category which is represented in the form of bar graphs [6].

```
#importing necessary packages
#install packages which aren't available
from flask import Flask, request, render_template
import matplotlib.pyplot as plt
from textblob import TextBlob
import io
import base64
from selenium import webdriver
import time

#execution starts from main...
```

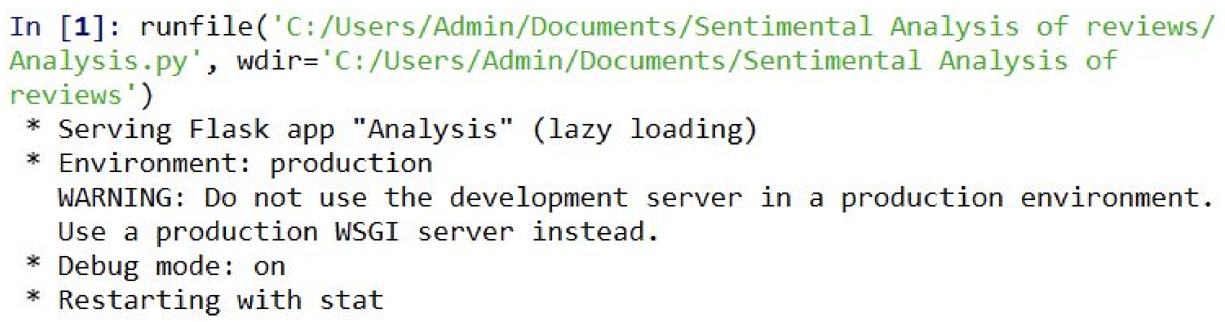
```
#creating app for the Flask
app=Flask(__name__, static_url_path='/')

#default route for methods GET and POST
@app.route("/",methods=['GET','POST'])
#function for the above route
def index():
        return(render_template('open.html'))#displaying open.html to the user

#route for result
@app.route("/res",methods=['GET','POST'])
#function for the above route
def scrape():
        url=request.form.get('url')h
        #download and place the chrome driver in the local
        #pass the relavent path to the driver
        driver=webdriver.Chrome(executable_path='C:/Users/Admin/Documents/Sentimental\
        Analysis of reviews/templates/driver/chromedriver.exe')
        #open the url using the driver
        driver.get(url)
        #defining the pause for every scroll that we see in below code
        pause=5 #setting pause time
        driver.execute_script('window.scrollTo(1, 500);')
        # scroll to 500 pixel on the monitor where 500 is height
        time.sleep(pause) #waiting for pause sec
        video = driver.find_element_by_css_selector("video")
        #accessing video using css selector
        driver.execute_script("arguments[0].muted = true;", video)
        #muting the video
        driver.execute_script("window.scrollTo(0,750);")
        #scroll to 750 pixel on the monitor where 750 is height
        time.sleep(pause)#waiting for pause sec
        driver.execute_script("window.scrollTo(0,1000);")
        # scroll to 750 pixel on the monitor where 750 is height
        time.sleep(pause)#waiting for pause sec
        driver.execute_script("window.scrollTo(0,1500);")
        # scroll to 1500 pixel on the monitor where 1500 is height
        lastHeight =3000
        #setting lastheight
```

```
i=0
while i<10:#runs for either 10 instructions
	driver.execute_script("window.scrollTo(0,"+str(lastHeight)+");")
	# scroll to lastheight variable pixel on the monitor
	time.sleep(pause)#waiting for pause sec
	newHeight = driver.execute_script("return \
	document.documentElement.scrollHeight")
	if newHeight == lastHeight:
		break
	division=newHeight-lastHeight# difference of heights
	lastHeight+=division#update the last height for the scroll
	driver.execute_script("window.scrollTo(0,"+str(lastHeight)+");")
	#scroll to last height
	lastHeight = newHeight
	#update the lastheight
	i+=1.  #iteration icremental

'''Using Xpath along with id'''
comment_div=driver.find_element_by_xpath('//*[@id="contents"]')
#getting division of comments
comments=comment_div.find_elements_by_xpath('//*[@id="content-text"]')
#getting comments under comments division
likes=comment_div.find_elements_by_xpath('//*[@id="vote-count-middle"]')
# getting likes under comments division
likeslist=[]#To store all the likes count for all the comments collected

'''initialization'''
no_of_positive=0
no_of_negetive=0
no_of_deep_positive=0
no_of_deep_negetive=0
no_of_neutral=0
for like in likes:
	#validating for only digits
	val_like=''.join(filter(lambda x: x.isdigit(),like.text))
	#assigning for empty strings
	if(len(val_like)==0):
		likeslist.append(0)
	else:
```

```
                likeslist.append(int(val_like))#conversion to values

    for i in range(len(comments)):
        comment=comments[i]
        #for every comment
        anal=TextBlob(comment.text)#getting the text of comment collected
        '''Getting polarity for all the comments and classifying them accordingly'''
        if anal.sentiment.polarity>0 and anal.sentiment.polarity<0.4:
            no_of_positive+=1+likeslist[i]
        elif anal.sentiment.polarity>=0.4:
            no_of_deep_positive+=1+likeslist[i]
        elif anal.sentiment.polarity==0:
            no_of_neutral+=1+likeslist[i]
        elif anal.sentiment.polarity<=-0.4:
            no_of_deep_negetive+=1+likeslist[i]
        else:
            no_of_negetive+=1+likeslist[i]
    '''building the graph based on analysis'''
    labels=['Strongly Positive','Positive','Neutral','Negetive','Strong Negetive']
    values=[no_of_deep_positive, no_of_positive, no_of_neutral, no_of_negetive, \
    no_of_deep_negetive]
    plt.bar(labels,values,color=['blue', 'cyan','green','yellow','red'])
    plt.xlabel('Type of Comments', fontsize=18)
    plt.ylabel('No of Comments', fontsize=18)
    plt.xticks(labels, fontsize=6, rotation=0,fontstyle='italic')
    plt.title('Analysis of Comments')
    '''saving the image in the format of png image and creating path for the graph'''
    img = io.BytesIO()
    plt.savefig(img, format='png')
    img.seek(0)
    graph_url = base64.b64encode(img.getvalue()).decode()
    plt.close()
    graph='data:image/png;base64,{}'.format(graph_url)
    driver.close()
    #sending graph path to the output.html page
    return(render_template('output.html',graph=graph))
if(__name__=="__main__"):
    app.run(debug=True)
```

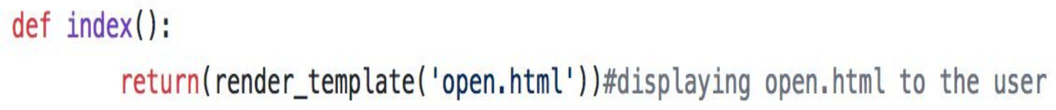
4.2.4 Algorithm:

Step1: We need to run the python script using python interpreter. Then open the corresponding page in the any of the browser. The following is shown after execution of the script [1].



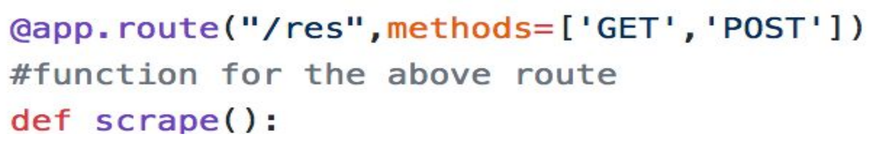
```
In [1]: runfile('C:/Users/Admin/Documents/Sentimental Analysis of reviews/
Analysis.py', wdir='C:/Users/Admin/Documents/Sentimental Analysis of
reviews')
 * Serving Flask app "Analysis" (lazy loading)
 * Environment: production
   WARNING: Do not use the development server in a production environment.
   Use a production WSGI server instead.
 * Debug mode: on
 * Restarting with stat
```

Step2: The script executes @app.route("/",methods=['GET','POST']), then the def under it. The script returns the open.html will be returned onto the browser at client end which has requested the server [2].



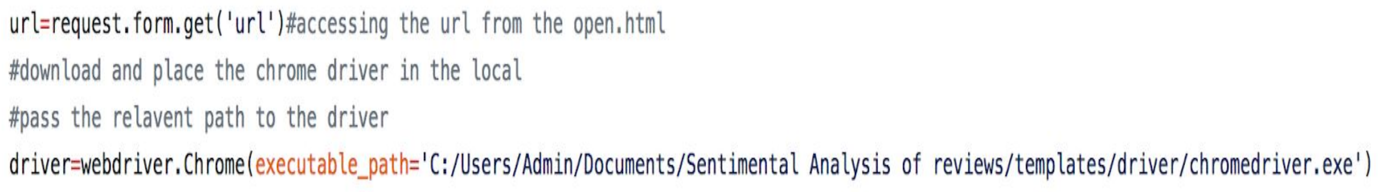
```
def index():
        return(render_template('open.html'))#displaying open.html to the user
```

Step3: The user needs to give the youtube link at the input space in the open.html. After successful submission the then @app.route("/res",methods=['GET','POST']) gets executes and then the def under it [2].



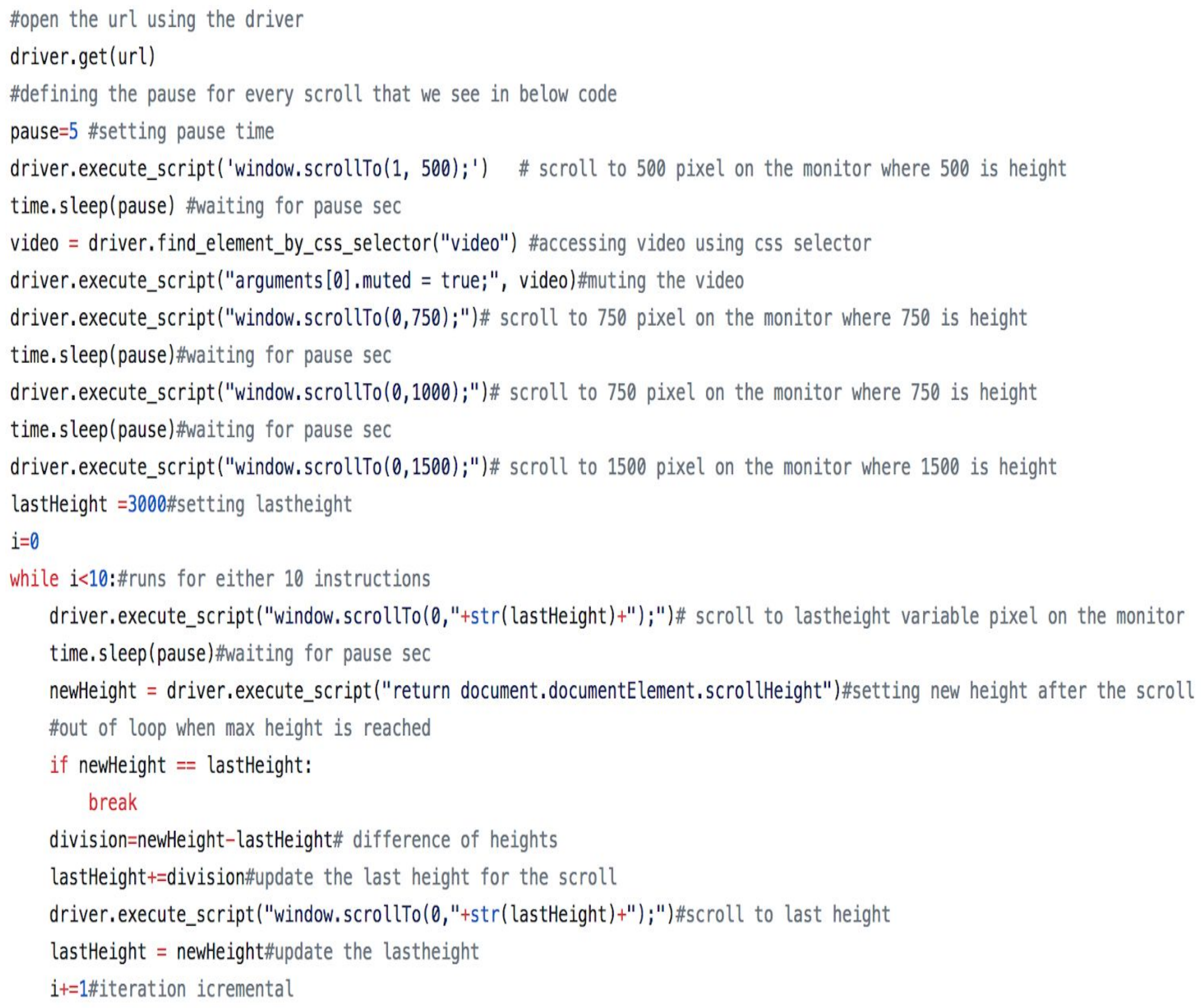
```
@app.route("/res",methods=['GET','POST'])
#function for the above route
def scrape():
```

Step4: In the execution the URL will be accessed and stored under the url variable. Then the driver for chrome is opened at the server end. The url is opened in the chrome using the driver.

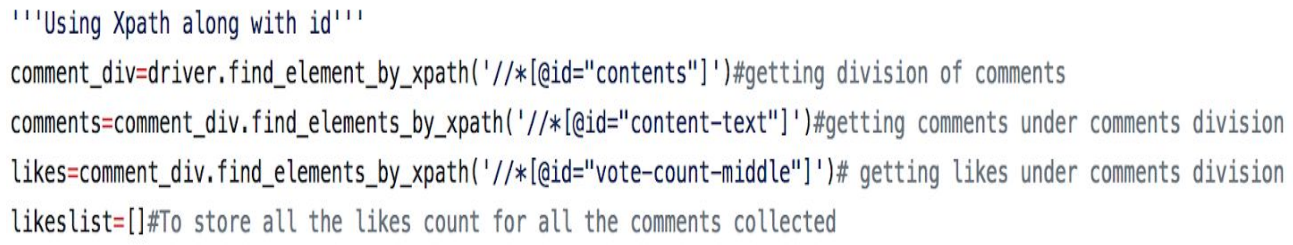


```
url=request.form.get('url')#accessing the url from the open.html
#download and place the chrome driver in the local
#pass the relavent path to the driver
driver=webdriver.Chrome(executable_path='C:/Users/Admin/Documents/Sentimental Analysis of reviews/templates/driver/chromedriver.exe')
```

Step5: Once the URL is runned in the chrome, the scroll operation takes place using the Selenium with a wait time of 5 sec. This Selenium technology will allow us to scrap the website contents dynamically by scrolling down after the given wait time. So, once the wait time is done the website will be scrolled down iteratively [4]. To make the comments scrapped efficient and maintain comment scraped and time taken to it balanced, we are limiting the scraping to 10 iterations. Once the iterations are done the scraping gets stopped.

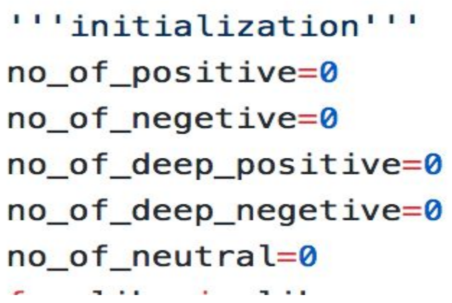


```
#open the url using the driver
driver.get(url)
#defining the pause for every scroll that we see in below code
pause=5 #setting pause time
driver.execute_script('window.scrollTo(1, 500);')   # scroll to 500 pixel on the monitor where 500 is height
time.sleep(pause) #waiting for pause sec
video = driver.find_element_by_css_selector("video") #accessing video using css selector
driver.execute_script("arguments[0].muted = true;", video)#muting the video
driver.execute_script("window.scrollTo(0,750);")# scroll to 750 pixel on the monitor where 750 is height
time.sleep(pause)#waiting for pause sec
driver.execute_script("window.scrollTo(0,1000);")# scroll to 750 pixel on the monitor where 750 is height
time.sleep(pause)#waiting for pause sec
driver.execute_script("window.scrollTo(0,1500);")# scroll to 1500 pixel on the monitor where 1500 is height
lastHeight =3000#setting lastheight
i=0
while i<10:#runs for either 10 instructions
    driver.execute_script("window.scrollTo(0,"+str(lastHeight)+");")# scroll to lastheight variable pixel on the monitor
    time.sleep(pause)#waiting for pause sec
    newHeight = driver.execute_script("return document.documentElement.scrollHeight")#setting new height after the scroll
    #out of loop when max height is reached
    if newHeight == lastHeight:
        break
    division=newHeight-lastHeight# difference of heights
    lastHeight+=division#update the last height for the scroll
    driver.execute_script("window.scrollTo(0,"+str(lastHeight)+");")#scroll to last height
    lastHeight = newHeight#update the lastheight
    i+=1#iteration icremental
```



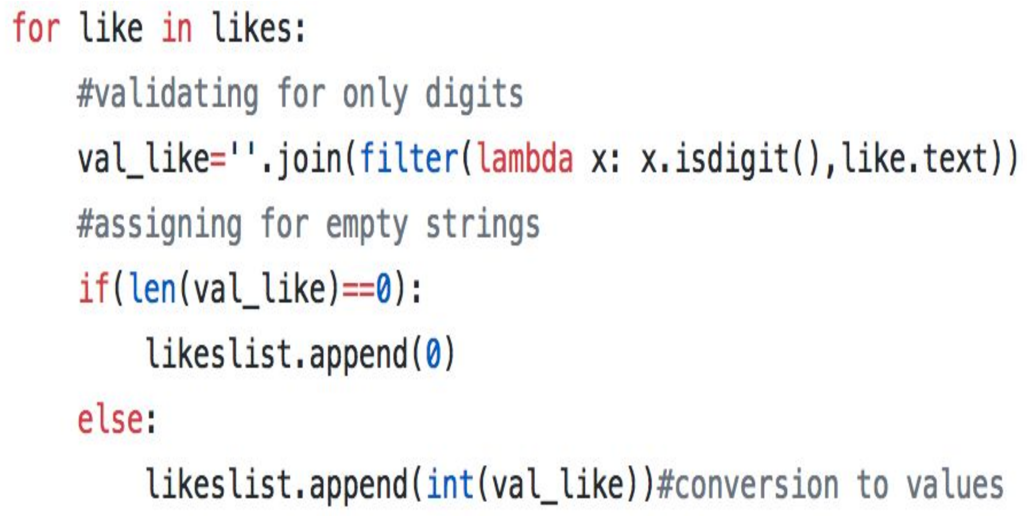
```
'''Using Xpath along with id'''
comment_div=driver.find_element_by_xpath('//*[@id="contents"]')#getting division of comments
comments=comment_div.find_elements_by_xpath('//*[@id="content-text"]')#getting comments under comments division
likes=comment_div.find_elements_by_xpath('//*[@id="vote-count-middle"]')# getting likes under comments division
likeslist=[]#To store all the likes count for all the comments collected
```

Step6: Assigning the initial values for calculation and to later used to display the results based on these values that are calculated later based on analysis and weights.



```
'''initialization'''
no_of_positive=0
no_of_negetive=0
no_of_deep_positive=0
no_of_deep_negetive=0
no_of_neutral=0
```

Step7: Filtering and conversion of text type of number of likes to int. The filtered text type of likes will be converted into int and will be used for finding the weights of each comment [1].



```
for like in likes:
    #validating for only digits
    val_like=''.join(filter(lambda x: x.isdigit(),like.text))
    #assigning for empty strings
    if(len(val_like)==0):
        likeslist.append(0)
    else:
        likeslist.append(int(val_like))#conversion to values
```

Step8: Every comment is converted into text for analysis. For the converted text the sentiment polarity is found and update classification category values [3].

```
for i in range(len(comments)):
    comment=comments[i]#for every comment
    anal=TextBlob(comment.text)#getting the text of comment collected
    '''Getting polarity for all the comments and classifying them accordingly'''
    if anal.sentiment.polarity>0 and anal.sentiment.polarity<0.4:
        no_of_positive+=1+likeslist[i]
    elif anal.sentiment.polarity>=0.4:
        no_of_deep_positive+=1+likeslist[i]
    elif anal.sentiment.polarity==0:
        no_of_neutral+=1+likeslist[i]
    elif anal.sentiment.polarity<=-0.4:
        no_of_deep_negetive+=1+likeslist[i]
    else:
        no_of_negetive+=1+likeslist[i]
```

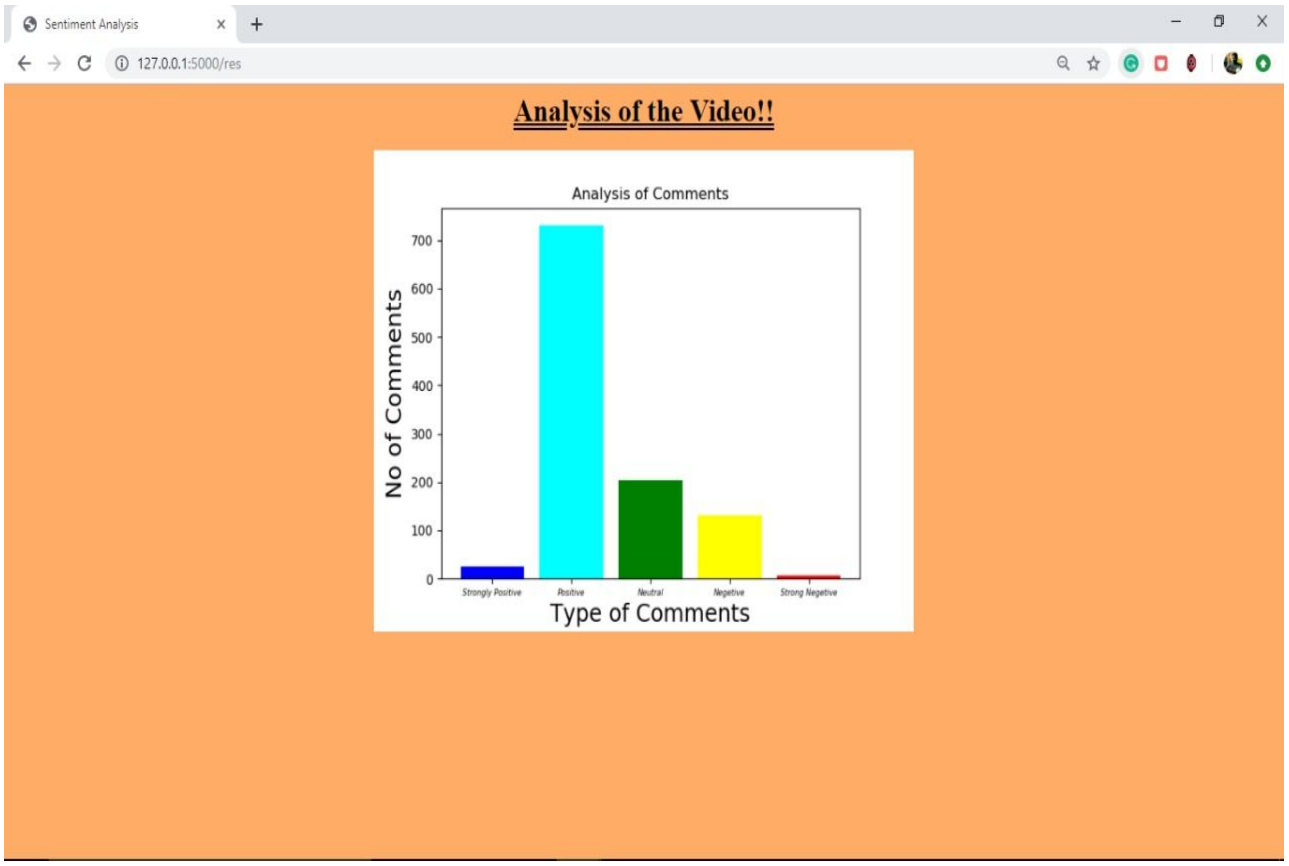
Step9: Once the sentiment polarity is categorized and calculated done, relevant bar graph is built [6].

```
'''building the graph based on analysis'''

labels=['Strongly Positive','Positive','Neutral','Negetive','Strong Negetive']
values=[no_of_deep_positive,no_of_positive,no_of_neutral,no_of_negetive,no_of_deep_negetive]
plt.bar(labels,values,color=['blue', 'cyan','green','yellow','red'])
plt.xlabel('Type of Comments', fontsize=18)
plt.ylabel('No of Comments', fontsize=18)
plt.xticks(labels, fontsize=6, rotation=0,fontstyle='italic')
plt.title('Analysis of Comments')
```

Step10: Once the bar graph is built it is converted into png image format and a relevant graph url is constructed, this graph url is sent to output.html and output.html template is rendered and it is passed to the client end [2].

```
img = io.BytesIO()
plt.savefig(img, format='png')
img.seek(0)
graph_url = base64.b64encode(img.getvalue()).decode()
plt.close()
graph='data:image/png;base64,{}'.format(graph_url)
driver.close()
#sending graph path to the output.html page
return(render_template('output.html',graph=graph))
```

Step11: The bar graph in the form of image which represents the categories of comments of the video and the output is shown as the below image at the client end [2].



CHAPTER-5
TESTING AND VALIDATION

5.1 INTRODUCTION:

The development of software systems involves a series of production activities where opportunities for injection of human fallibilities are enormous. Errors may begin to occur at the very inception of the process where the objectives may be erroneously or imperfectly specified, as well as in later design and development stages. Because of the human inability to perform and communicate with perfection, software development is accompanied by a quality assurance activity.

5.2 TESTING TECHNIQUES:

Testing is the process of executing a program with the intention of finding errors. The various test strategies used for testing the software are as follows.

- ❖ Unit Testing
- ❖ Integration Testing
- ❖ Validation Testing
- ❖ System Testing

5.2.1 UNIT TESTING

Unit testing focuses on verification effort on the smallest unit of software design module. The main goal is to make sure that every source statement and the logic path has been executed correctly at least once. The output of this stage is the source code.

5.2.2 INTEGRATION TESTING

In Integration testing, we find errors that have occurred during the integration. After testing each module, which is then integrated into subsystems and then to form the entire system on which integration testing is performed. The goal of testing is to detect design errors while focusing on testing the interconnection between modules.

5.2.3 VALIDATION TESTING

This testing concentrates on confirming that the software is error-free in all respects. All the specified validations are verified and the software is subjected to hard-core testing. It also aims at determining the degree of deviation that exists in the software designed from the specification; they are listed out and are corrected.

5.2.4 SYSTEM TESTING

In this testing, the system is tested for the errors after coupling all the modules together. The system is tested against the specified requirements to see if all the requirements are met and the system performs as specified by the requirements.

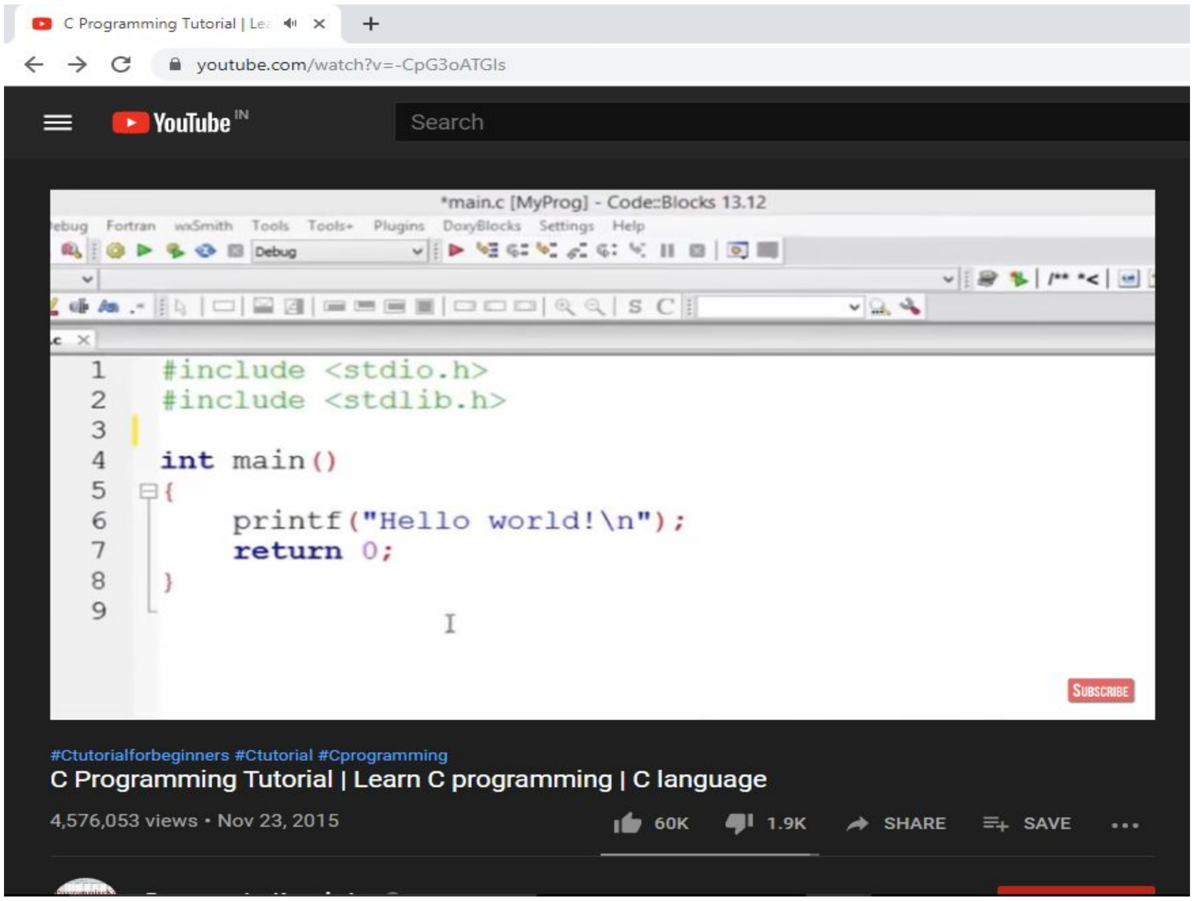
5.3 TEST CASES:

A test case is a set of conditions or variables under which a tester will determine whether a system under test satisfies requirements or works correctly. The process of developing test cases can help find problems in the requirements or design of an application. A test case is a specification of the inputs, execution conditions, testing procedure, and expected results that define a single test to be executed to achieve a particular software testing objective, such as to exercise a particular program path or to verify compliance with a specific requirement.

Following are the test cases that are used to test the code for relevant output. The outputs are cross checked manually.

Test Case 1-

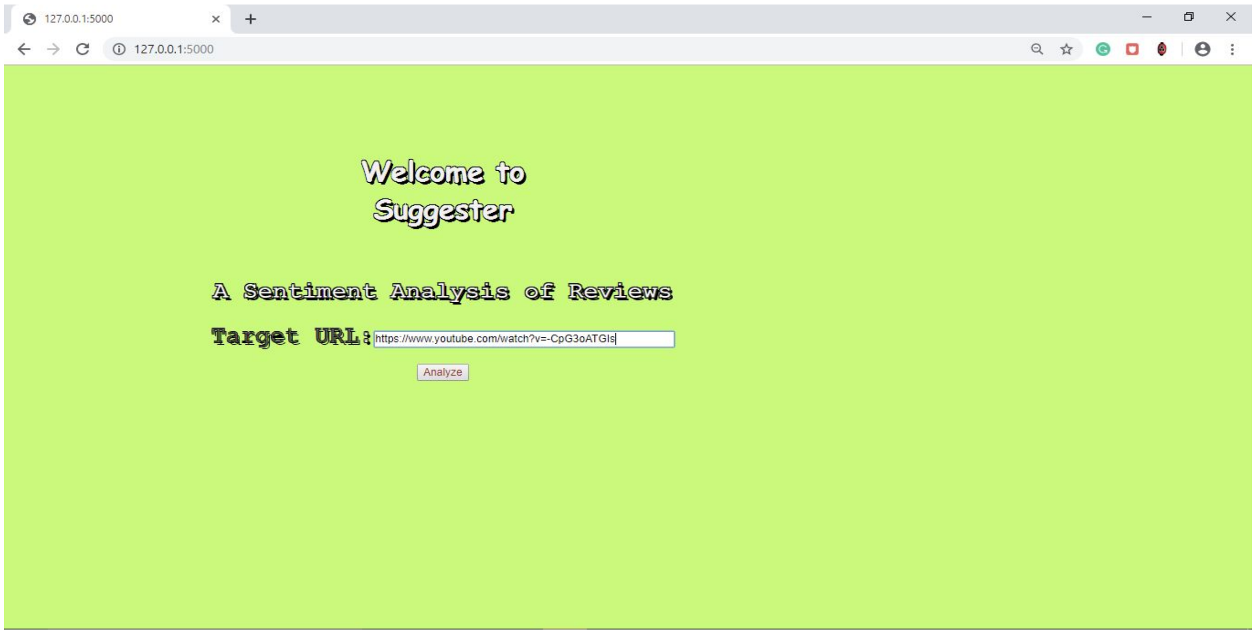
The following is the youtube video that is selected for our testing.
Youtube link:https://www.youtube.com/watch?v=-CpG3oATGIs


C Programming Tutorial | Le
youtube.com/watch?v=-CpG3oATGIs
YouTube IN
Search
*main.c [MyProg] - Code::Blocks 13.12
Debug
SUBSCRIBE
SHARE
SAVE

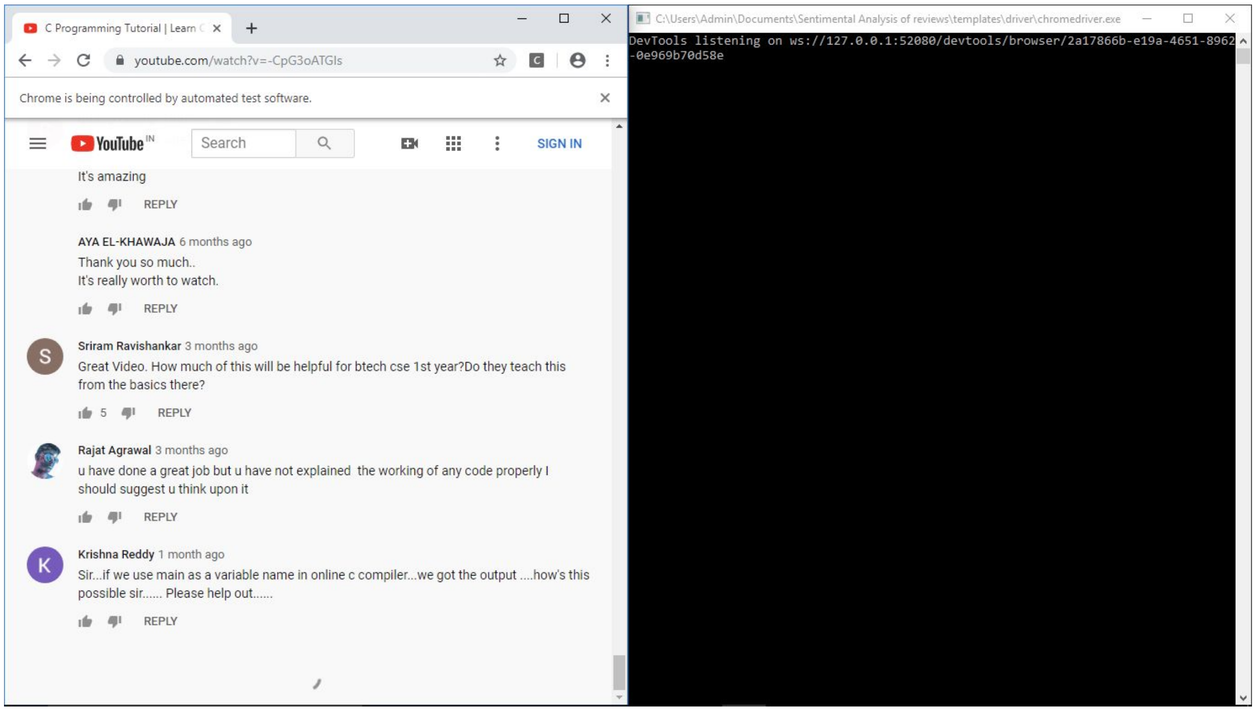
```c
#include <stdio.h>
#include <stdlib.h>

int main()
{
    printf("Hello world!\n");
    return 0;
}
```

#Ctutorialforbeginners #Ctutorial #Cprogramming
C Programming Tutorial | Learn C programming | C language
4,576,053 views • Nov 23, 2015 60K 1.9K

The following is the input screen that is shown to the user at client end and where the user enters the Youtube link.


127.0.0.1:5000
Welcome to Suggester
A Sentiment Analysis of Reviews
Target URL: https://www.youtube.com/watch?v=-CpG3oATGIs
Analyze

The following gets executed at the server end where the chrome driver is executed and the youtube comments gets scraped.


C Programming Tutorial | Learn C
youtube.com/watch?v=-CpG3oATGIs
Chrome is being controlled by automated test software.
YouTube
Search
SIGN IN
C:\Users\Admin\Documents\Sentimental Analysis of reviews\templates\driver\chromedriver.exe

It's amazing

AYA EL-KHAWAJA 6 months ago
Thank you so much..
It's really worth to watch.

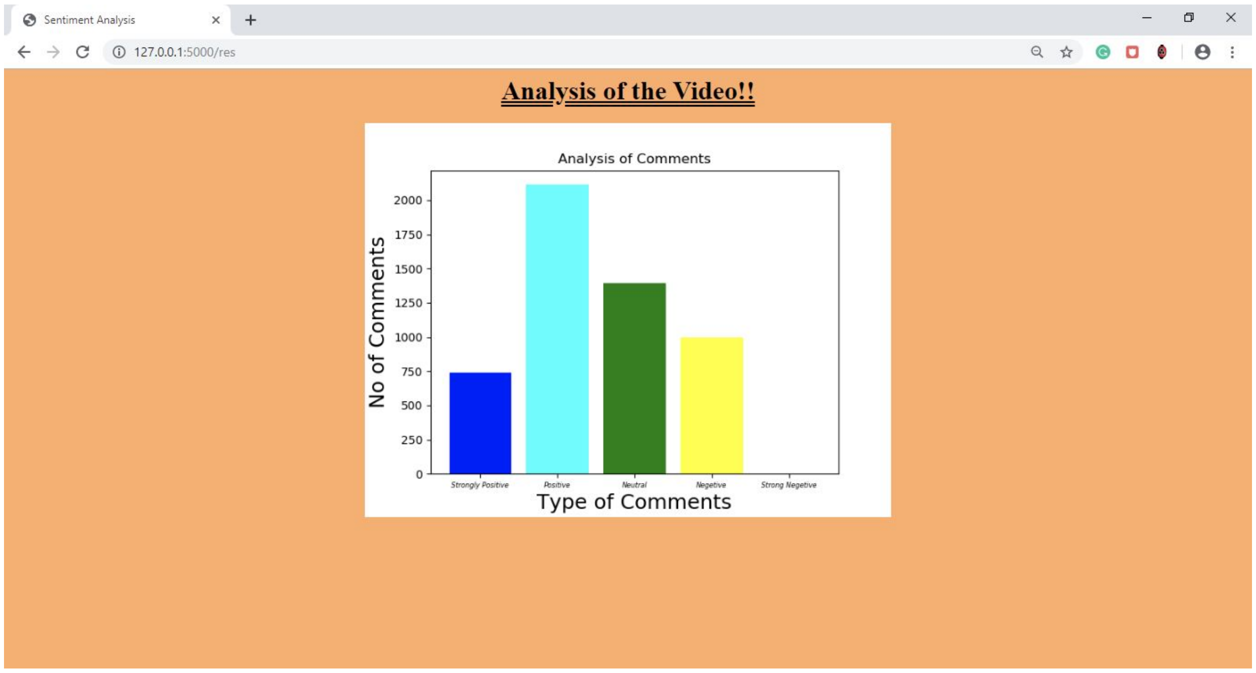
Sriram Ravishankar 3 months ago
Great Video. How much of this will be helpful for btech cse 1st year?Do they teach this from the basics there?

Rajat Agrawal 3 months ago
u have done a great job but u have not explained the working of any code properly I should suggest u think upon it

Krishna Reddy 1 month ago
Sir...if we use main as a variable name in online c compiler...we got the output ....how's this possible sir...... Please help out......

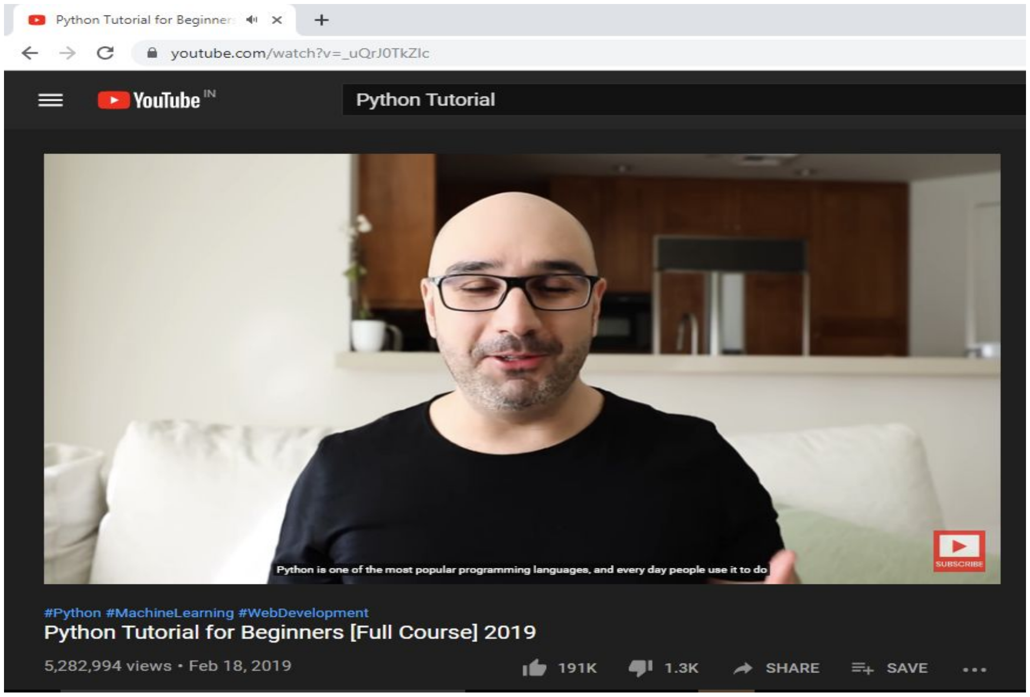
```
DevTools listening on ws://127.0.0.1:52080/devtools/browser/2a17866b-e19a-4651-8962
-0e969b70d58e
```

The output is displayed at the client end where the visualization is shown as follows.

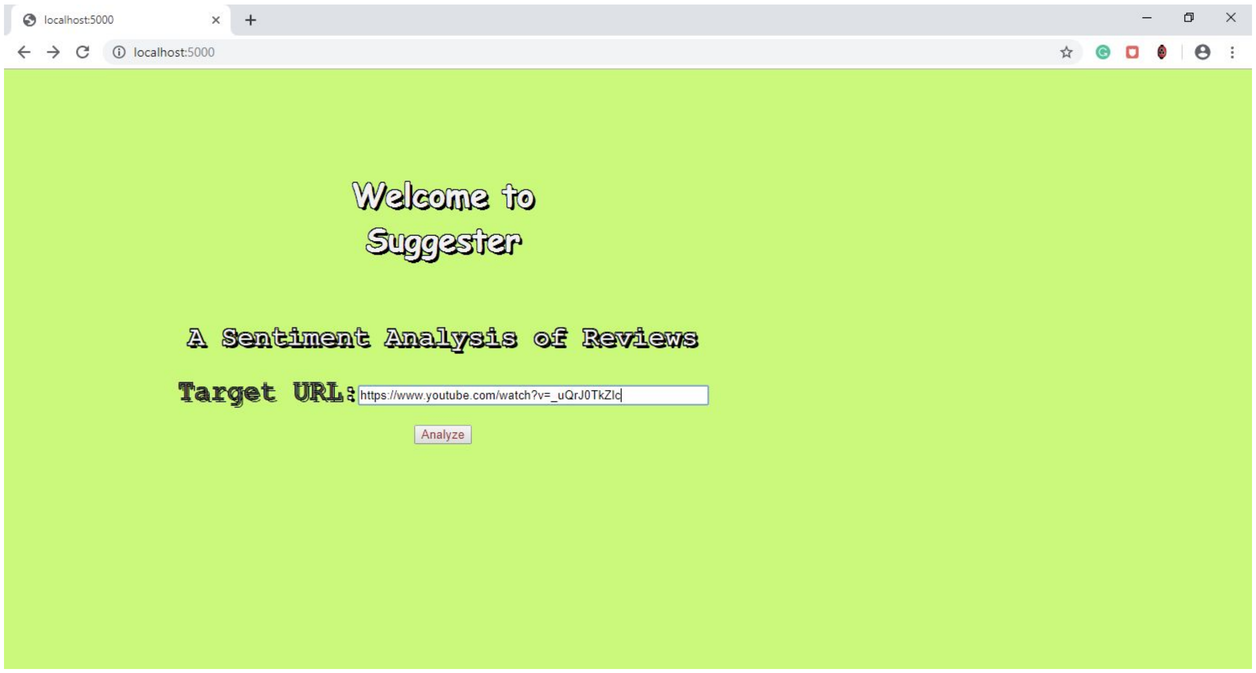


Test Case 2-

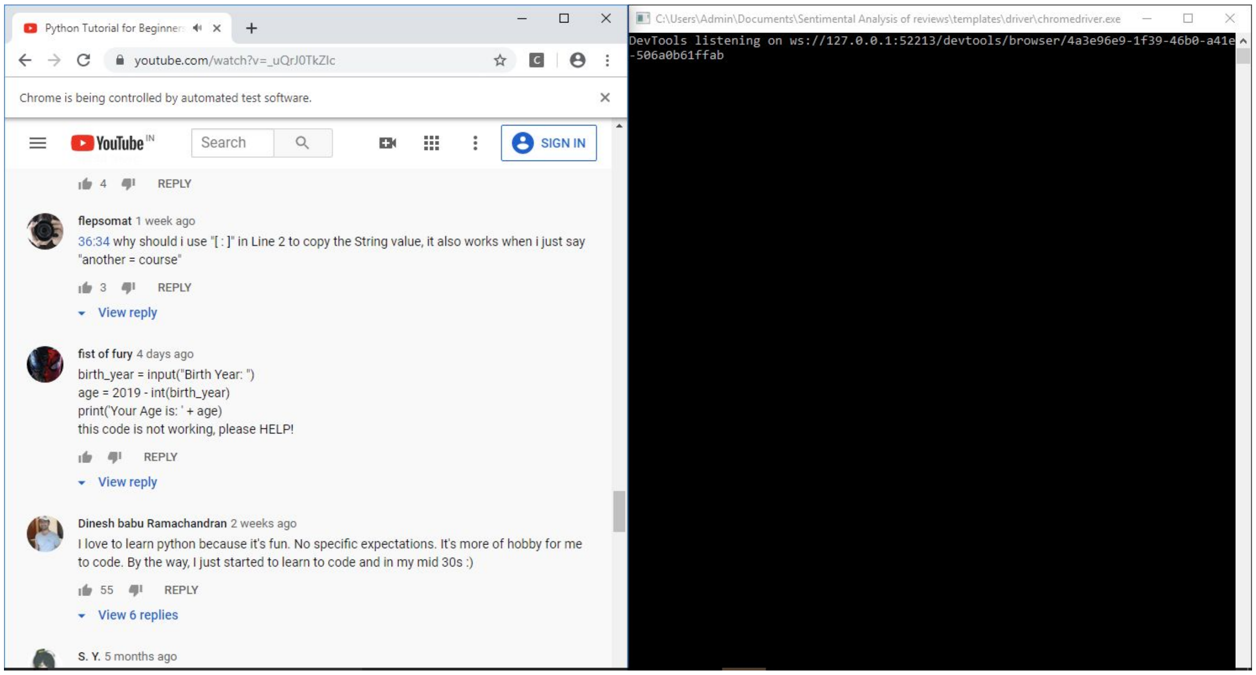
The following is the youtube video that is selected for our testing.
Youtube link:https://www.youtube.com/watch?v=_uQrJ0TkZlc


Python Tutorial for Beginners
youtube.com/watch?v=_uQrJ0TkZlc
YouTube IN
Python Tutorial
Python is one of the most popular programming languages, and every day people use it to do
#Python #MachineLearning #WebDevelopment
Python Tutorial for Beginners [Full Course] 2019
5,282,994 views • Feb 18, 2019
191K
1.3K
SHARE
SAVE

The following is the input screen that is shown to the user at client end and where the user enters the Youtube link.


localhost:5000
localhost:5000
Welcome to Suggester
A Sentiment Analysis of Reviews
Target URL:
https://www.youtube.com/watch?v=_uQrJ0TkZIc
Analyze

The following gets executed at the server end where the chrome driver is executed and the youtube comments gets scraped.


Python Tutorial for Beginner
youtube.com/watch?v=_uQrJ0TkZIc
YouTube IN
Search
SIGN IN
C:\Users\Admin\Documents\Sentimental Analysis of reviews\templates\driver\chromedriver.exe

Chrome is being controlled by automated test software.

4 REPLY

flepsomat 1 week ago
36:34 why should i use "[:]" in Line 2 to copy the String value, it also works when i just say "another = course"

3 REPLY
View reply

fist of fury 4 days ago
birth_year = input("Birth Year: ")
age = 2019 - int(birth_year)
print('Your Age is: ' + age)
this code is not working, please HELP!

REPLY
View reply

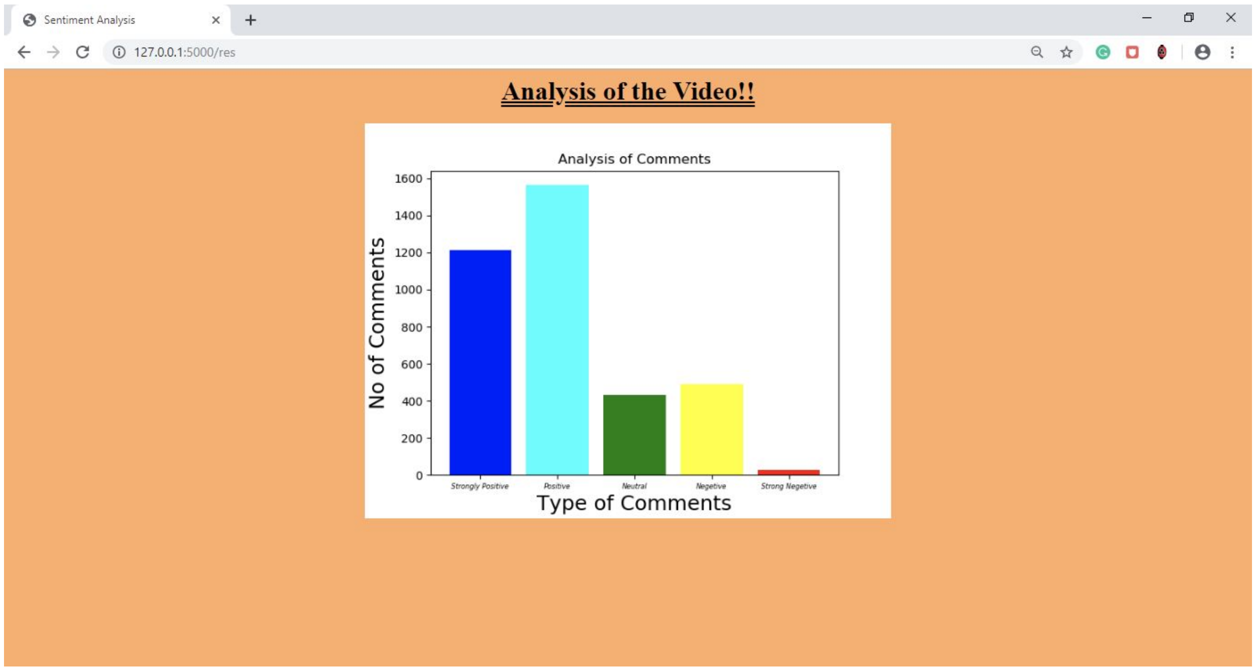
Dinesh babu Ramachandran 2 weeks ago
I love to learn python because it's fun. No specific expectations. It's more of hobby for me to code. By the way, I just started to learn to code and in my mid 30s :)

55 REPLY
View 6 replies

S. Y. 5 months ago

```
DevTools listening on ws://127.0.0.1:52213/devtools/browser/4a3e96e9-1f39-46b0-a41e
-506a0b61ffab
```

The output is displayed at the client end where the visualization is shown as follows.



The above test cases are executed and the outputs of them are successfully cross checked with the actual comments manually respectively.

CHAPTER – 6
CONCLUSION & FUTURE ENHANCEMENT

6.1 CONCLUSION

- We deployed python script on to a server using python interpreter to analyse the youtube comments and show appropriate presentation of how good the video is.
- Deployed code uses Flask, Selenium and TextBlob to present the visualization to the user for the given youtube link.
- This can be used in multiple areas, it is not purely restricted to Youtube.

6.2 PROJECT FUTURE ENHANCEMENT

This deployed script can be further improved to make analysis sites at the same time can be enhanced to analyse multiple languages other than English for example- Hindi, Japanese, Chinese etc. The scrapping time can be further reduced based on the internet connection at the server end which results in less latency and higher accuracy.

REFERENCES

1. https://docs.python.org/3
2. https://flask-doc.readthedocs.io/en/latest/
3. https://textblob.readthedocs.io/en/dev/
4. https://www.seleniumhq.org/docs/
5. https://towardsdatascience.com/sentiment-analysis-concept-analysis-and-applications-6c94d6f58c17
6. https://towardsdatascience.com/introduction-to-data-visualization-in-python-89a54c97fbed
7. https://www.w3schools.com/html/
8. https://css-tricks.com/almanac/properties/t/text-shadow/