

**School of Engineering and Technology**

Main Campus,Off Hennur-Bagalur Main Road,Chagalahatti,Bengaluru-562149

A

DISSERTATION REPORT ON

**“SENTINA”**

**Sentiment Based Twitter Reply Bot**

Submitted to

**CMR University**

**School Of Engineering and Technology,Bagalur**

for the partial fulfillment of the requirement for the award of the Degree of

**B.TECH**

**IN**

**COMPUTER SCIENCE**

Submitted by:

K Abhinandhan (REG.NO.**18BBTCS045)**

KN Prajwal Sai (REG.NO.**18BBTCS046)**

Karthik G (REG.NO.**18BBTCS047)**

Manoj Bahadur (REG.NO.**18BBTCS063)**

**Under the Guidance Of**

**Mr.Yogesh**

**and**

**Ms. Shivali Shakya**

**(Asst.Professor,CSE)**

**DEPARTMENT OF**

**COMPUTER SCIENCE AND ENGINEERING CMR UNIVERSITY**

**BAGALUR**

**2018-19**



**School of Engineering Technology**

Main Campus,Off Hennur-Bagalur Main Road,Chagalahatti,Bengaluru-562149

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**CERTIFICATE**

Certified that the project work entitled **SENTINA-Sentiment Based Twitter Reply Bot** carried out by **Mr Karthik G**, REG.NO **18BBTCS047** in partial fulfillment for the Award of Bachelor of Engineering / Bachelor of Technology in **COMPUTER SCIENCE AND TECHNOLOGY** of the CMR University, Bagalur during the year 2018-19. It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in the Report deposited in the departmental library. The project report has been approved as it satisfies the academic requirements in respect of Project work prescribed for the said Degree.

Signature of Guides Signature of HOD Signature of Dean

**External Viva**

Name of the examiners Signature

1. 1.

2. 2.

Date:

**DECLARATION**

I ,**KARTHIK G** ,student of CMR University, School Of Engineering and Technology,Bagalur hearby decalre that the dissertation entitled “**SENTINA-Sentiment Based Twitter Reply Bot”** embodies the reportvof my project carried out independently by me during the second semester of **B.TECH** in **COMPUTER SCIENCE,** under the supervision and guidance of **Mr.Yogesh and Ms. Shivali Shakya(Asst.Professor,CSE)** Department of Computer Science and Engineering and this work has been submitted for the partial fulfillment of the requirements for the Award of the B.Tech Degree.

I have not submitted the matter embodies to any other university or Institution for the Award of other Degree.

Date: 3 June 2019

Place: CMR University, Bengaluru-562149

**\*\*\*\*\*\***

**(REG.NO.188BBTCS047)**

Signature of the student

**ACKNOWLEDGEMENT**

I express my foremost gratitude to **Dr.M.K.Nagaraj**,**Dean,School of Engineeering and Technology** for his constant support.

I express my foremost gratitude to **Dr.Smitha Rao, HOD,Computer Science and Engineering Department** for her constant support.

I express my foremost gratitude to my guides **Mr.Yogesh and Ms.Shivali Shakya (Asst.Professor,CSE),** Department of Computer Science and Engineering , CMR University, Bagalur for his/her inspiration, adroit guidance, constant supervision, direction and discussion in successful completion of this dissertation.

My sincere thanks to all teaching and non-Teaching Staff of **Computer Science and Engineering** Department for all the facilities provided, without which, I could not have progressed with my work. Thanks to my parents who have been a great source of strength in the completion of this dissertation.

**\*\*\*\*\*\***

**(18BBTCS047)**

**ABSTRACT**

Social media have received more attention nowadays. Public and private opinion about a wide variety of subjects are expressed and spread continually via numerous social media. Twitter is one of the social media that is gaining popularity. Twitter offers organizations a fast and effective way to analyze customers' perspectives toward the critical to success in the market place. Developing a program for sentiment analysis is an approach to be used to computationally measure customers' perceptions. This paper reports on the design of a sentiment analysis, extracting a vast amount of tweets. Prototyping is used in this development. Results classify customers' perspective via tweets into positive and negative, which is represented in a pie chart and html page. However, the program has planned to develop on a web application system, but due to limitation of Django which can be worked on a Linux server or LAMP, for further this approach need to be done.

In this Modern Era customer satisfaction is very important for the reputation of the service or the company. Due to the wide spread internet people will express there reviews and issues mostly through online social medias. It is expected from the customer that the service provider to reply for their expressions. It is a humongous task for a person to go through all the mentions and replies.

**INDEX**

|  |  |  |
| --- | --- | --- |
| **TITLE**  **NO** | **CONTENTS** | **PAGE NO** |
| 1. | DECLARATION | 3 |
| 2. | ACKNOWLEDGEMENT | 4 |
| 3. | ABSTRACT | 5 |
| 4. | INDEX | 6 |
| 5. | LIST OF FIGURES | 7 |
| 6. | CHAPTER 1- INTRODUCTION | 8-9 |
| 7. | CHAPTER 2-COMPONENTS | 10-11 |
| 8. | CHAPTER 3- PROCEDURE | 12-13 |
| 9. | CHAPTER 4-RESULT AND APPLICATIONS | 14-15 |
| 10. | CONCLUSION | 16 |
| 11. | APPENDICES | 17-19 |
| 12. | REFERENCE | 20 |

**LIST OF FIGURES**

|  |  |  |
| --- | --- | --- |
| **FIGURE NO** | **NAME OF THE FIGURE** | **PAGE NO** |
| FIG 1 | Installing Tweepy python library | 10 |
| FIG 2 | Installing Textblob python library | 11 |
| FIG 3 | Running code in Terminal | 13 |
| FIG 4 | Result of Project | 14 |

CHAPTER 1

**INTRODUCTION**

**SENTINA** is a sentiment based tweet replying bot. It is based on the principle of Sentiment Analysis which is a topic of Natural Language Processing. Natural Language is one of the major subfield of Computer science or Artificial Intelligence.

The basic function of our App or Python Script is to read the tweets mentioned to a particular profile in Twitter.  The read tweets are analyzed to find the sentiment of the tweet. Sentiment here range between Positive, Negative and Neutral. Once the sentiment of the tweet is analyzed, we reply to the tweet automatically without manual input using pre-written messages. The pre-written messages are marked for different sentiments, which will be used accordingly when the sentiment is analyzed.

**1.1 Natural Language Processing**

Natural Language Processing is a branch of artificial intelligence that deals with analyzing, understanding and generating the languages that humans use naturally in order tointerface with computers in both written and spoken contexts using natural human languages instead of computer languages. One of the challenges inherent in natural language processing is teaching computers to understand the way humans learn and use language.

**1.2 Sentiment Analysis**

Sentiment Analysis, also called opinion mining or emotion AI, is the process of determining whether a piece of writing is positive, negative, or neutral. A common use case for this technology is to discover how people feel about a particular topic. Sentiment analysis is widely applied to reviews and social media for a variety of applications.

There can be two approaches to sentiment analysis:

1. Lexicon-based methods

2. Machine Learning-based methods.

In this problem, we will be using a Lexicon-based method.

**1.3 Lexicon-based method**

The lexicon based approach is based on the assumption that the contextual sentiment orientation is the sum of the sentiment orientation of each word or phrase.

A sentiment classifier takes a piece of plan text as input, and makes a classification decision on whether its contents are positive or negative. For simplicity, let’s assume that input text is known a priori to be opinionated (which we could obtain by filtering input text through another classifier that detects opinionated text from neutral ones).

**Key metrics you can track**

* Hashtag & campaign tracking – shares, reach, engagement, mentions
* Sentiment analysis – what’s driving negativity & positivity
* Image recognition - protect your trademark & reputation
* Google Analytics + Talkwalker for social media ROI
* Virality – track how your content spreads across the web & social
* Influencer marketing – identify industry influencers & brand ambassadors

**1.4 Lexicon-based method’s impact**

* Centralize all your social media data with one tool. Rather than logging in and out of multiple social media analytics tools, add your social media profiles and those of competing brands to a single dashboard. You’ll be able to analyze key metrics from your customers, campaigns, competitors, and the industry as a whole.
* Centralizing your social media accounts, along with those of your competitors, will allow you to choose the stats that matter and draw comparisons. Bringing actionable insights to improve your social marketing strategy.

CHAPTER 2

**COMPONENTS**

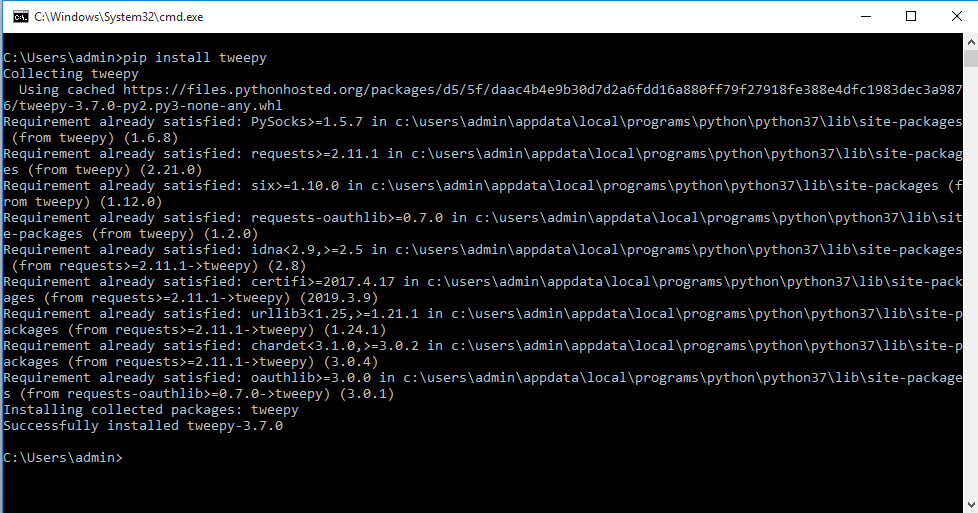
**Software Requirements**

**1. Tweepy**

Tweepy is open-sourced, hosted on GitHub and enables Python to communicate with Twitter platform and use its API.

At the time of writing, the current version of tweepy is 1.13. It was released on January 17, and offers various bug fixes and new functionality compared to the previous version. The 2.x version is being developed but it is currently unstable so a huge majority of the users should use the regular version. Installing tweepy is easy

**pip install tweepy**

****

**Fig:1 Installing Tweepy python library**

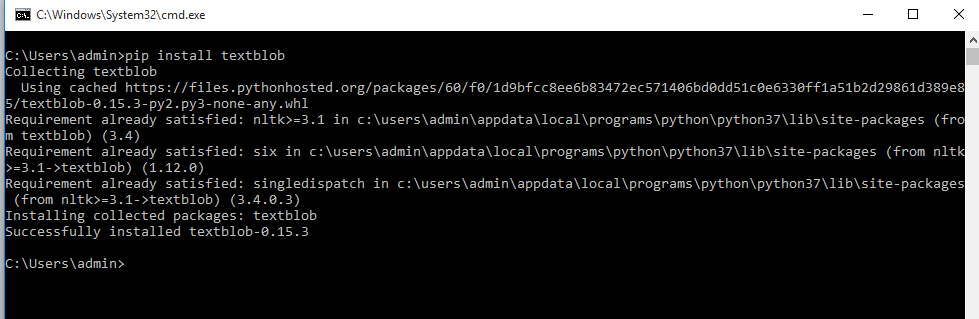
**2. TextBlob**

*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.

## Sentiment Analysis

The [sentiment](https://textblob.readthedocs.io/en/dev/api_reference.html#textblob.blob.TextBlob.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.

**pip install textblob**

****

**Fig:2 Installing Textblob python library**

CHAPTER 3

**PROCEDURE**

We follow these 3 major steps in our program:

* Authorize twitter API client.
* Make a GET request to Twitter API to fetch tweets for a particular query.
* Parse the tweets. Classify each tweet as positive, negative or neutral.
* Based on the situation replying to those tweets.
* First of all, we create a **TwitterClient** class. This class contains all the methods to interact with Twitter API and parsing tweets. We use **\_\_init\_\_** function to handle the authentication of API client.
* In **get\_tweets** function, we use:  
   fetched\_tweets = self.api.search(q = query, count = count)  
    
   to call the Twitter API to fetch tweets.
* In **get\_tweet\_sentiment** we use textblob module.  
   analysis = TextBlob(self.clean\_tweet(tweet))  
    
   TextBlob is actually a high level library built over top of NLTK library. First we call **clean\_tweet** method to remove links, special characters, etc. from the tweet using some simple regex.  
   Then, as we pass **tweet** to create a **TextBlob** object, following processing is done over text by textblob library:  
  + Tokenize the tweet ,i.e split words from body of text.
  + Remove stopwords from the tokens.(stopwords are the commonly used words which are irrelevant in text analysis like I, am, you, are, etc.)
  + Do POS( part of speech) tagging of the tokens and select only significant features/tokens like adjectives, adverbs, etc.
  + Pass the tokens to a **sentiment classifier** which classifies the tweet sentiment as positive, negative or neutral by assigning it a polarity between -1.0 to 1.0

**Here is how sentiment classifier is created:**

* + **TextBlob** uses a Movies Reviews dataset in which reviews have already been labelled as positive or negative.
  + Positive and negative features are extracted from each positive and negative review respectively.
  + Training data now consists of labelled positive and negative features. This data is trained on a Naive Bayes Classifier.

Then, we use **sentiment.polarity** method of **TextBlob** class to get the polarity of tweet between -1 to 1.  
 Then, we classify polarity as:  
  
 **if analysis.sentiment.polarity > 0:**

**return 'positive'**

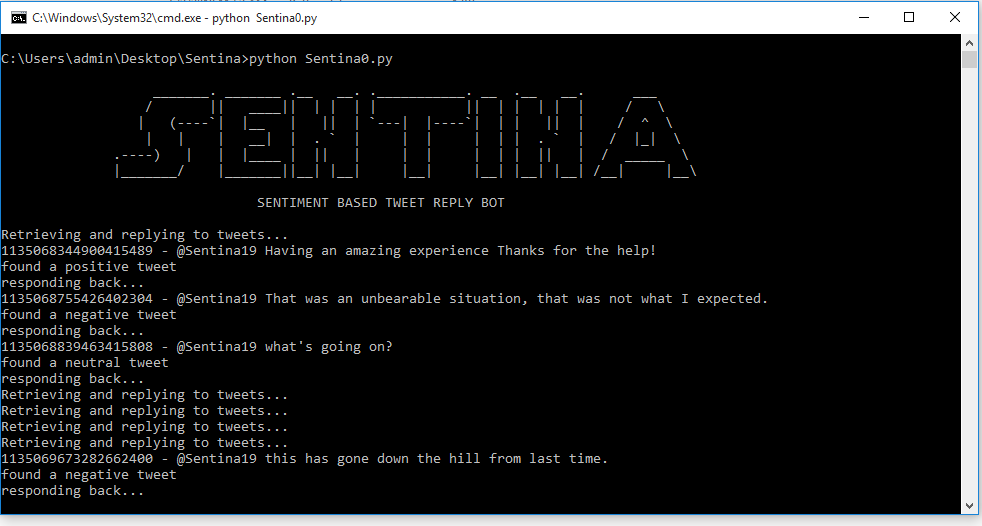
**elif analysis.sentiment.polarity == 0:**

**return 'neutral'**

**else:**

**return 'negative'**

Finally, parsed tweets are returned. Then, we can do various types of statistical analysis on the tweets. For example, in the above program, we tried to find the percentage of positive, negative and neutral tweets about a query.

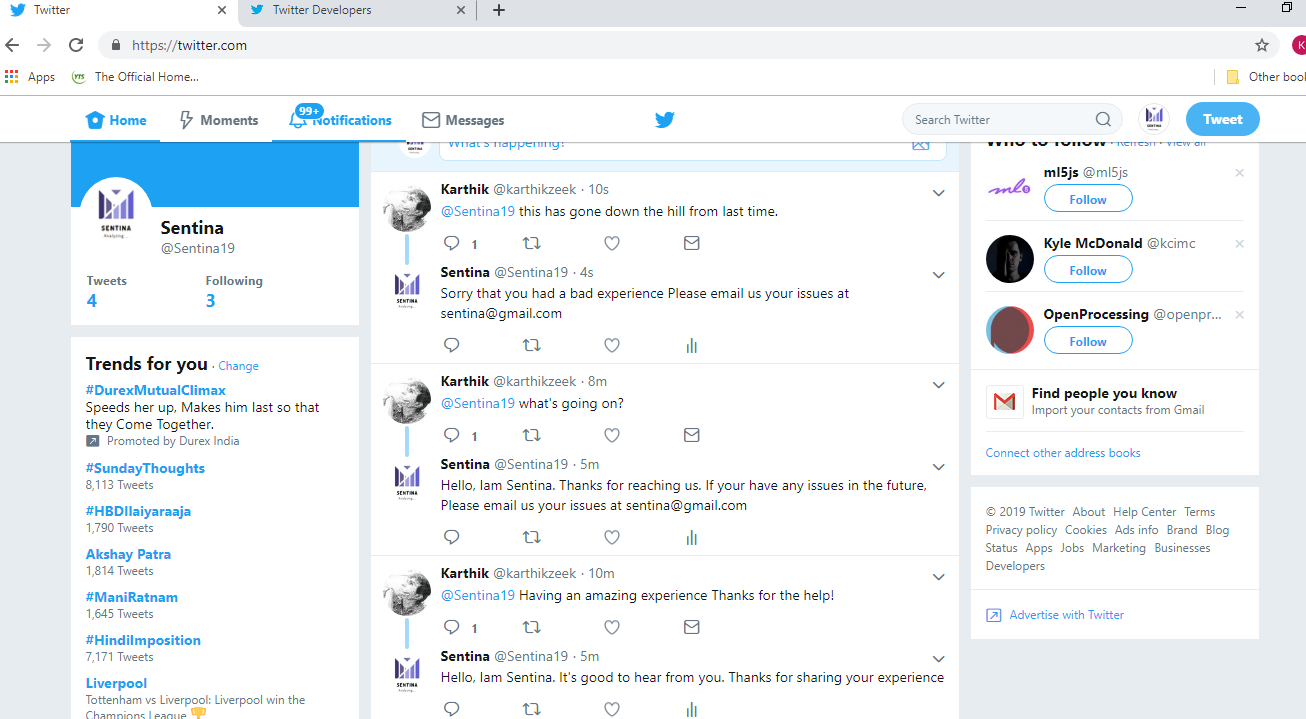


**Fig:3 Running code in Terminal**

CHAPTER 4

**RESULTS AND APPLICATIONS**

When the app or the Python script is run the reply bot starts reading the tweets. The tweets in the time line of the profile linked to the given authorization keys are read. The read tweets are returned to the code as a string input. The string is than sentiment analyzed using the Textblob library’s polarity function. Polarity of the tweet gives the sentiment associated with the tweet. Once the sentiment of the tweet is analyzed, we send a reply to the tweeter using @mentions. The replies are sent based on the sentiment. Replies are pre-written strings marked to different sentiment. All this happens automatically once the code is run. Therefore we can run this code in cloud to automate the tweet replying.



**Fig: 4 Result of Project**

**APPLICATIONS**

* **Reputation management** - or you could also call it brand monitoring. We all know how much good reputation means these days when the majority of us check social media reviews as well as review sites before making a purchase decision.   
    
  Negative reviews put people off and how you handle can define your future as a business. You could either ignore them (highly not recommended), act rude and make your situation even worse, or apologize for whatever caused a person to write a negative opinion and do your best to make up for it.
* **Customer support** - Social media are channels of communication with your customers these days, and whenever they’re unhappy about something related to you, whether or not it’s your fault, they’ll call you out on Facebook/Twitter/Instagram.  
    
  People nowadays expect brands to respond on social media almost immediately, and if you’re not quick enough, you might as well see them moving on to your competitors instead of waiting for your reply.
* **Competitor monitoring** - chances are some of your competitors are getting bad press online. It’s where you could step in as long as you’re aware of those negative mentions. During these times we could automate to reply for the user to try our product or service

**CONCLUSION**

In this Modern Era customer satisfaction is very important for the reputation of the service or the company. Due to the wide spread internet people will express there reviews and issues mostly through online social medias. It is expected from the customer that the service provider to reply for their expressions. It is a humongous task for a person to go through all the mentions and replies.

Methods like this can be used to filter out the unwanted and not necessary to take care of messages. Still replying to the customer accordingly for the concerns.

Using this particular method for automating replies will highly reduce man power requirement and will save lot of time.

The method has its disadvantages which should be mentioned at this point.

Sentiment Analysis is a branch of Natural language processing which is still under development. We cannot expect one hundred percent accuracy. There can be times where Sarcasm can be taken as a negative tweet. The analysis doesn’t analyze the context of the tweet but interpret based on the words used. Most of the data conducted on the accuracy of the analysis suggest that the method is close to eighty percent.

The system can be trained using data sets and increase the accuracy.

**APPENDICES**

**Main**

import tweepy

import time

# Sentiment analysis using Text blob

from analysis import get\_tweet\_sentiment

# NOTE: I put my keys in the keys.py to separate them

# from this main file.

# Please refer to keys\_format.py to see the format.

from keys import \*

# NOTE: flush=True is just for running this script

# with PythonAnywhere's always-on task.

print(name\_tag, flush=True)

auth = tweepy.OAuthHandler(CONSUMER\_KEY, CONSUMER\_SECRET)

auth.set\_access\_token(ACCESS\_KEY, ACCESS\_SECRET)

api = tweepy.API(auth)

FILE\_NAME = 'last\_seen\_id.txt'

def retrieve\_last\_seen\_id(file\_name):

f\_read = open(file\_name, 'r')

last\_seen\_id = int(f\_read.read().strip())

f\_read.close()

return last\_seen\_id

def store\_last\_seen\_id(last\_seen\_id, file\_name):

f\_write = open(file\_name, 'w')

f\_write.write(str(last\_seen\_id))

f\_write.close()

return

def reply\_to\_tweets():

print('Retrieving and replying to tweets...', flush=True)

# DEV NOTE: use 1060651988453654528 for testing.

last\_seen\_id = retrieve\_last\_seen\_id(FILE\_NAME)

# NOTE: We need to use tweet\_mode='extended' below to show

# all full tweets (with full\_text). Without it, long tweets

# would be cut off.

mentions = api.mentions\_timeline(

last\_seen\_id,

tweet\_mode='extended')

for mention in reversed(mentions):

print(str(mention.id) + ' - ' + mention.full\_text, flush=True)

last\_seen\_id = mention.id

store\_last\_seen\_id(last\_seen\_id, FILE\_NAME)

if get\_tweet\_sentiment(mention.full\_text.lower()) == 'positive':

print('found a positive tweet')

print('responding back...')

api.update\_status('@' + mention.user.screen\_name + " " +

"Hello, Iam Sentina. It's good to hear from you. " +

"Thanks for sharing your experience", mention.id)

if get\_tweet\_sentiment(mention.full\_text.lower()) == 'negative':

print('found a negative tweet')

print('responding back...')

api.update\_status('@' + mention.user.screen\_name + " " + 'Sorry that you had a bad experience' +

" Please email us your issues at sentina@gmail.com", mention.id)

if get\_tweet\_sentiment(mention.full\_text.lower()) == 'neutral':

print('found a neutral tweet')

print('responding back...')

api.update\_status('@' + mention.user.screen\_name + " " +

"Hello, Iam Sentina. Thanks for reaching us. If your have any issues in the future,"

+ " Please email us your issues at sentina@gmail.com", mention.id)

while True:

reply\_to\_tweets()

time.sleep(15)

**Analysis**

from textblob import TextBlob

def get\_tweet\_sentiment(tweet):

# Utility function to classify sentiment of passed tweet using textblob's sentiment method

# create TextBlob object of passed tweet text

analysis = TextBlob(tweet)

# set sentiment

if analysis.sentiment.polarity > 0:

return 'positive'

elif analysis.sentiment.polarity == 0:

return 'neutral'

else:

return 'negative'

**REFERENCE**

<https://www.python.org/>

<https://textblob.readthedocs.io/en/dev/quickstart.html#sentiment-analysis>

<https://tweepy.readthedocs.io/en/latest/>

<https://github.com/tweepy/tweepy>

[https://csdojo.io/twitter](https://www.youtube.com/redirect?redir_token=jibljxtVNvHaUGGCb0PvLT5BygR8MTU1OTU1Mjg5M0AxNTU5NDY2NDkz&q=https%3A%2F%2Fcsdojo.io%2Ftwitter&v=W0wWwglE1Vc&event=video_description)