Twitter Sentiment Analysis

**Prepared by**: Sandeep Singh

**Contents**

[Twitter Sentimental Analysis 3](#_Toc520464585)

[Introduction 3](#_Toc520464586)

[Pre-Requisites 3](#_Toc520464587)

[Installations 3](#_Toc520464588)

[MongoDB 3](#_Toc520464589)

[Jupyter Notebook 7](#_Toc520464590)

[Libraries Used 7](#_Toc520464591)

[Code 8](#_Toc520464592)

[Execution Approach 12](#_Toc520464593)

[Screenshots 14](#_Toc520464594)

[The Code 14](#_Toc520464595)

[CSV Files 14](#_Toc520464596)

[Database 15](#_Toc520464597)

[References 15](#_Toc520464598)

[GitHub link 15](#_Toc520464599)

# 

# Twitter Sentimental Analysis

# Introduction

Big data is a buzzword in IT which involves dealing with high volume of varied data coming at a very high velocity. One such real world scenario is of Twitter, the most trending form of social media. In today’s world, social media plays a vital role in determining how well any business is doing. The idea of this project is to perform sentiment analysis on various tweets which can be related to any business. For example, classifying tweets as positive, negative or neutral, one can determine the inclination of the business with respect to users’ tweets.

The project focuses on downloading the tweets from twitter, cleanse the data, save the data in a database and finally perform sentiment analysis on the tweets. The topic of search used in this project is “**Air Canada**”. Database used for the implementation of this project is MongoDB.

# Pre-Requisites

The following pre-requisites must be taken care of while implementing this project:

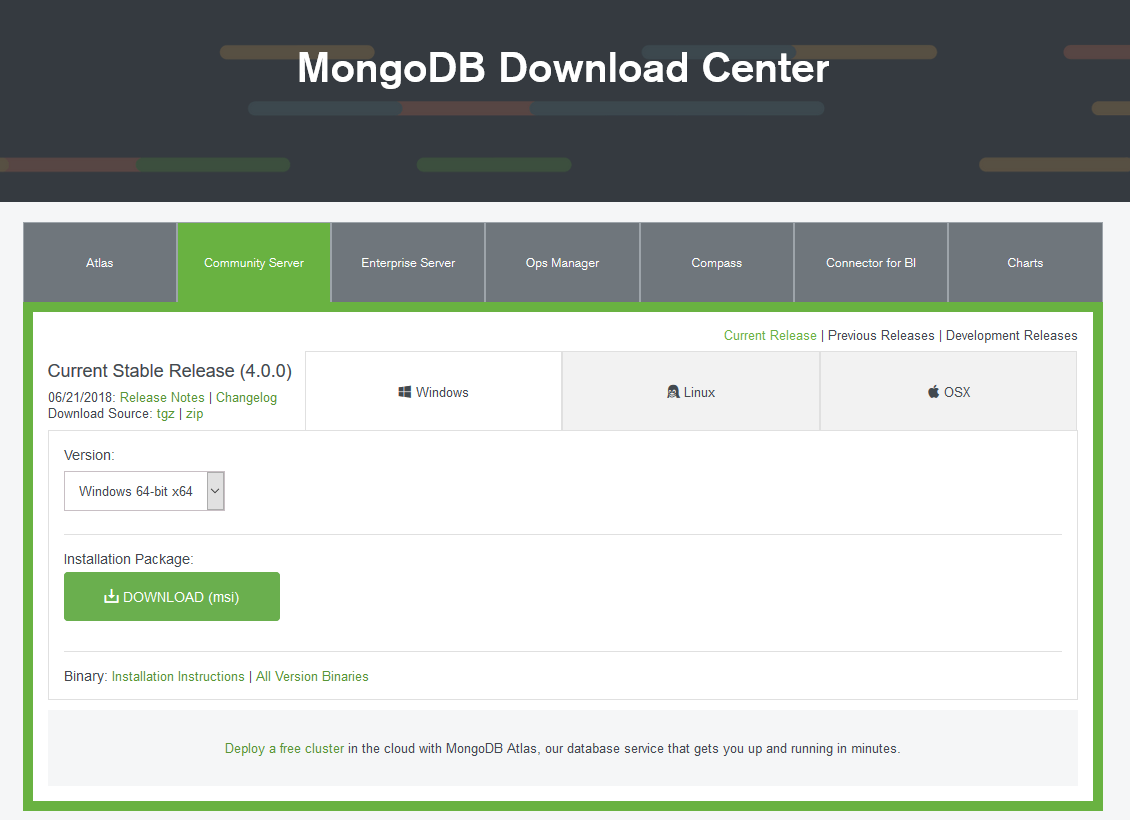
* MongoDB must be installed.
* Jupyter Notebook must be installed.
* All required libraries must be installed.

# Installations

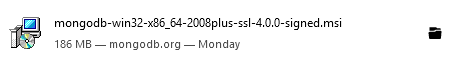
## MongoDB

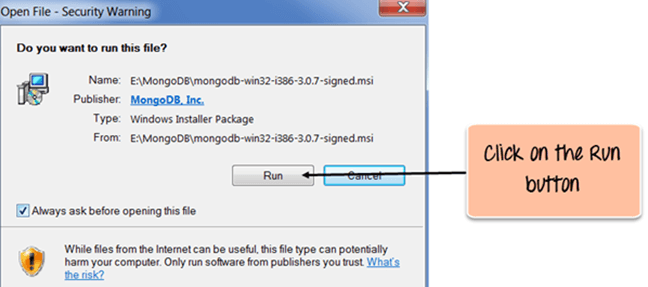
Appropriate versions of MongoDB must be installed while executing this project. However, Mongo DB (Community Server) was used during the development of this project.

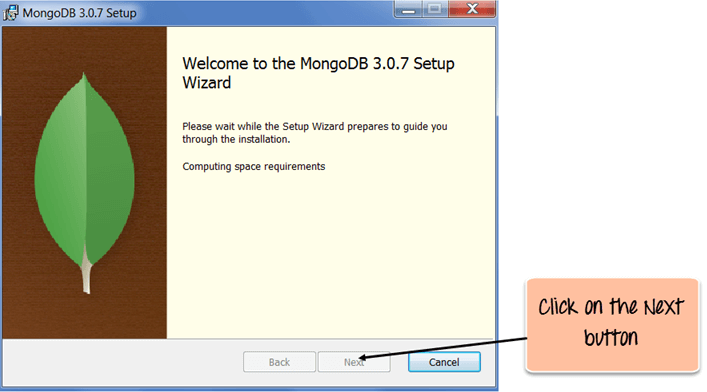
* [Link](https://www.mongodb.com/download-center#community) for downloading MongoDB (Community Server) for Windows.

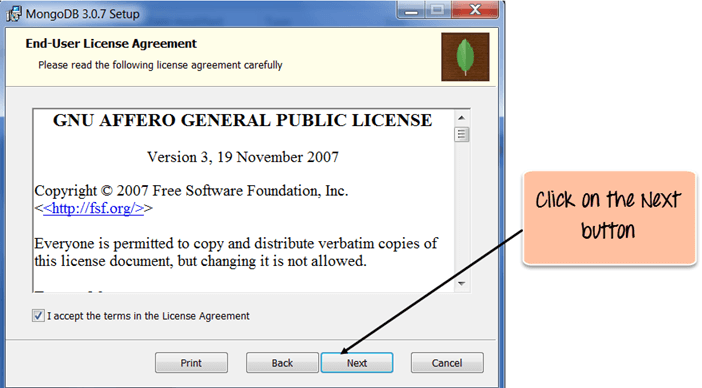


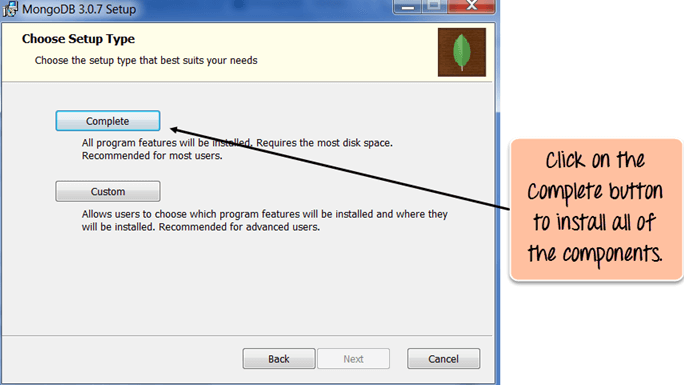
* Run the installation file

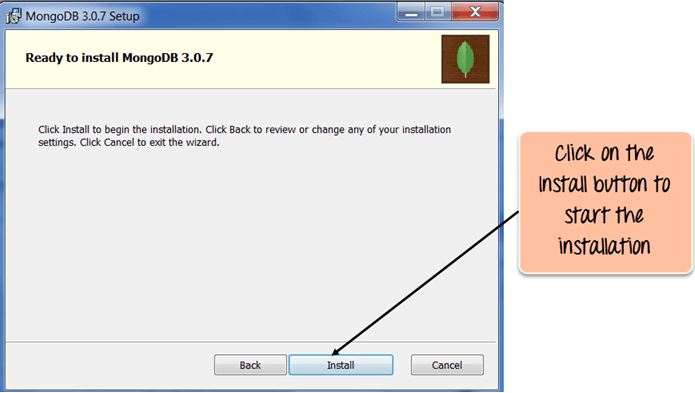


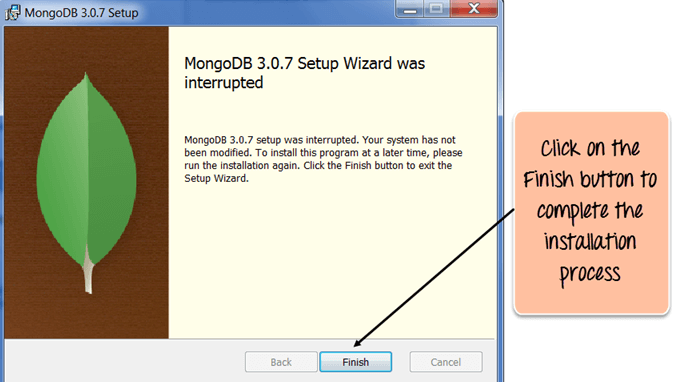












## Jupyter Notebook

In order to download Jupyter Notebook, Anaconda must be downloaded from this [link](https://www.anaconda.com/download/). All the default options must be selected to complete the installation. Upon completion, you can search for Jupyter Notebook and start coding. A link has also been posted in references to follow the installations.



\*\* The code can be run on any python complier provided the [listed libraries](#_Libraries_Used) must be installed.

# Libraries Used

* **Tweepy** -> to connect to twitter API and download the tweets
* **Pymongo** -> to connect to MongoDB and perform CRUD operations
* **Textblob** -> to perform sentiment analysis on the tweets
* **Pandas** -> to create and manage dataframe
* **Csv** -> to save the downloaded tweets in csv format
* **Json** -> to convert the csv file to json format before saving into MongoDB
* **Jdc** -> to split across multiple cells in Jupyter Notebook

python3 -m pip install <library\_name>

Above mentioned command can be used to install the libraries.

# Code

import pymongo

from pymongo import MongoClient

import tweepy

import csv

import pandas as pd

import json

import re

from textblob import TextBlob

#initializing the relevant keys for tweets downloads

def \_\_init\_\_(self):

self.consumer\_key = 'Your\_Consumer\_Key'

self.consumer\_secret = 'Your\_Consumer\_Secret'

self.access\_token = 'Your\_Access\_Token'

self.access\_token\_secret = 'Your\_Access\_Token\_Secret'

#search for the tweets based on the search field and save them in a csv file.

def searchTweets(self):

auth = tweepy.OAuthHandler(self.consumer\_key, self.consumer\_secret)

auth.set\_access\_token(self.access\_token, self.access\_token\_secret)

api = tweepy.API(auth,wait\_on\_rate\_limit=True)

csvFile = open('rawTweets.csv', 'a')

csvWriter = csv.writer(csvFile)

#creating header for our csv file

writer = csv.DictWriter(csvFile, fieldnames = ["Tweet ID", "UserName", "UserScreenName","UserLocation","UserDescription","UserLanguage","UserFollowers","UserFriends","Tweet Creation Time","Tweet Text"])

writer.writeheader()

#writing the downloaded tweets in csv file

for tweet in tweepy.Cursor(api.search,q="#unitedAIRLINES",count=100,lang="en",since="2017-04-03").items():

csvWriter.writerow([tweet.id,tweet.user.name.encode("utf-8"),tweet.user.screen\_name.encode("utf-8"),tweet.user.location.encode("utf-8"),tweet.user.description.encode("utf-8"),tweet.user.lang.encode("utf-8"),tweet.user.followers\_count,tweet.user.friends\_count,tweet.created\_at, tweet.text.encode('utf-8')])

csvFile.close()

#regex function to clean the downloaded tweets

def cleanSingleTweet(self,tweet):

return ' '.join(re.sub("(@[A-Za-z0-9]+)|([^0-9A-Za-z \t])|(\w+:\/\/\S+)", " ", tweet).split())

def cleanAllTweets(self):

df = pd.read\_csv("rawTweets.csv") #read csv with pandas df

df["UserName"]= df["UserName"].str[2:-1] #remove unwanted letters such as 'b'

df["UserScreenName"]= df["UserScreenName"].str[2:-1]

df["UserLocation"]=df["UserLocation"].str[2:-1]

df["UserDescription"]=df["UserDescription"].str[2:-1]

df["UserLanguage"]=df["UserLanguage"].str[2:-1]

df["Tweet Text"]=df["Tweet Text"].str[2:-1]

df["Clean Tweet Text"] = df["Tweet Text"].apply(self.cleanSingleTweet)

#print(df.head()) #Uncomment to check the contents of the dataframe

df.to\_csv("cleanTweets.csv", index = False) #Save the cleaned tweets in csv file

def saveTweetsToDB(self):

#read from csv and save the contents in desired database (mongoDB in this case)

client = MongoClient('localhost', 27017)

myDB = client['twitterDB']

collection\_name = 'tweets\_collection'

db\_cm = myDB[collection\_name]

data = pd.read\_csv("cleanTweets.csv")

data\_json = json.loads(data.to\_json(orient='records'))

db\_cm.remove()

db\_cm.insert(data\_json)

def sentimentAnalysis(self,tweet): #a function to calculate the polarity in order to determine the sentiment of the tweet

analysis = TextBlob(tweet)

tweetPolarity = analysis.sentiment.polarity

if tweetPolarity > 0:

return 'Positive'

elif tweetPolarity < 0:

return 'Negative'

else:

return 'Neutral'

def allTweetsAnalysis(self):

client = MongoClient('localhost', 27017)

#allDBs = client.list\_database\_names()

myDB = client['twitterDB']

collection\_name = 'tweets\_collection'

dbCollection = myDB[collection\_name]

documents =dbCollection.find()

#print(documents)

#tweet = [] #Use these lists if you want to check sentiment analysis by textblob

#result = []

for document in documents:

tempPolarity = self.sentimentAnalysis(document["Tweet Text"])

dbCollection.update({"\_id": document["\_id"]}, {"$set": {"SentimentAnalysis": tempPolarity}})

#tweet.append(document['Tweet Text'])

#result.append(tempPolarity)

#print(result)

def main():

t = Twitter()

t.searchTweets()

t.cleanAllTweets()

t.saveTweetsToDB()

t.allTweetsAnalysis()

main()

# Execution Approach

* **Import libraries**

We begin with importing all the required libraries for this project.

* **Define Class**

Next, we define a class 'Twitter' to incorporate all the required functions for this project. We define a **init** method and mention all the keys required to connect to twitter API

* **Search and Download Tweets**

The searchTweets() method is defined to connect to twitter API, search for tweets relating to specific topic and save them in a csv file. This will be a raw file containing all the information regarding the tweets in an unformatted order.

* **Clean Individual Tweets**

cleanSingleTweet() method is written to perform cleaning operation using regular expressions on each tweet. This method is later called in cleanAllTweets() method.

* **Clean Entire Data**

cleanAllTweets() method cleans all the fields as well as the tweets field to be used later for sentiment analysis. This cleaned data is stored as a csv file.

* **Save Data into MongoDB Database**

The cleaned csv file is imported as a pandas dataframe. The dataframe is converted to json format. A connection is established with mongodb and a new database and collection is created. The json format data is then inserted to the mongoDB database.

* **Sentiment Analysis of Individual Tweet**

sentimentAnalysis() method takes single tweet as input and returns a result (positive negative or neutral) based on the polarity generated using textblob library.

* **Sentiment Analysis of all Tweets**

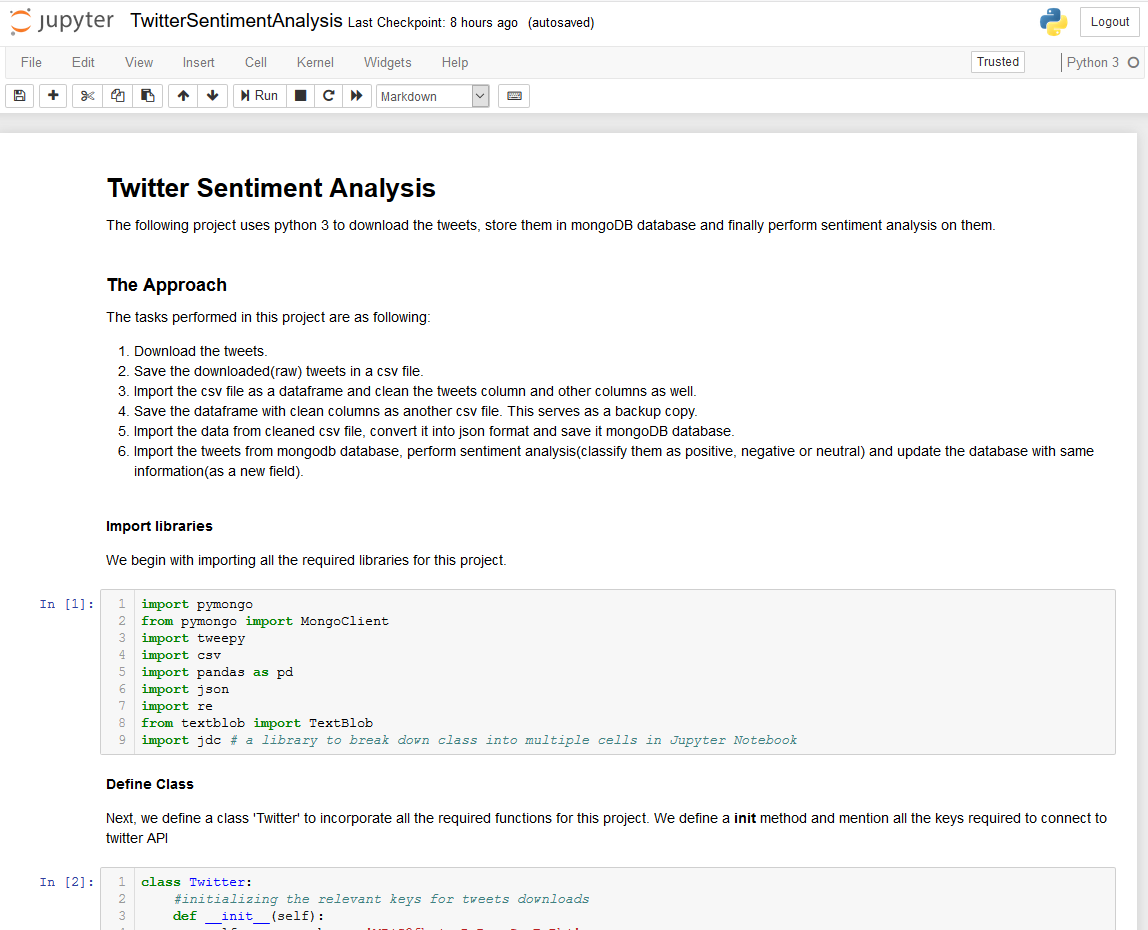
allTweetsAnalysis() method is used to extract all the tweets from the database and perform sentiment analysis on them. Finally, a new field representing the sentiment of each tweet is generated and the mongodb collection is updated.

* **Calling the main() method**

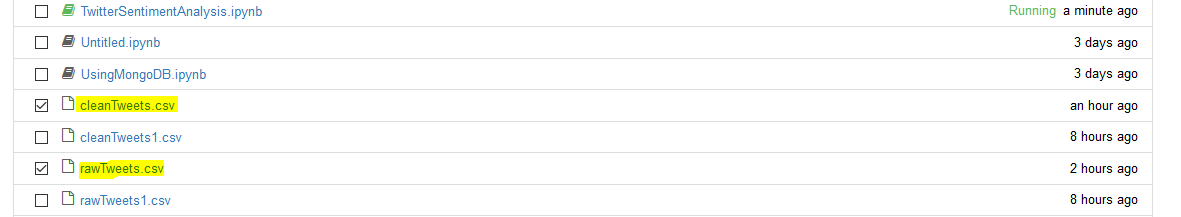
Lastly, we define main() method to declare the object of 'Twitter' class. We finally call main() method which in turn calls the relevant methods defined in class Twitter.

# Screenshots

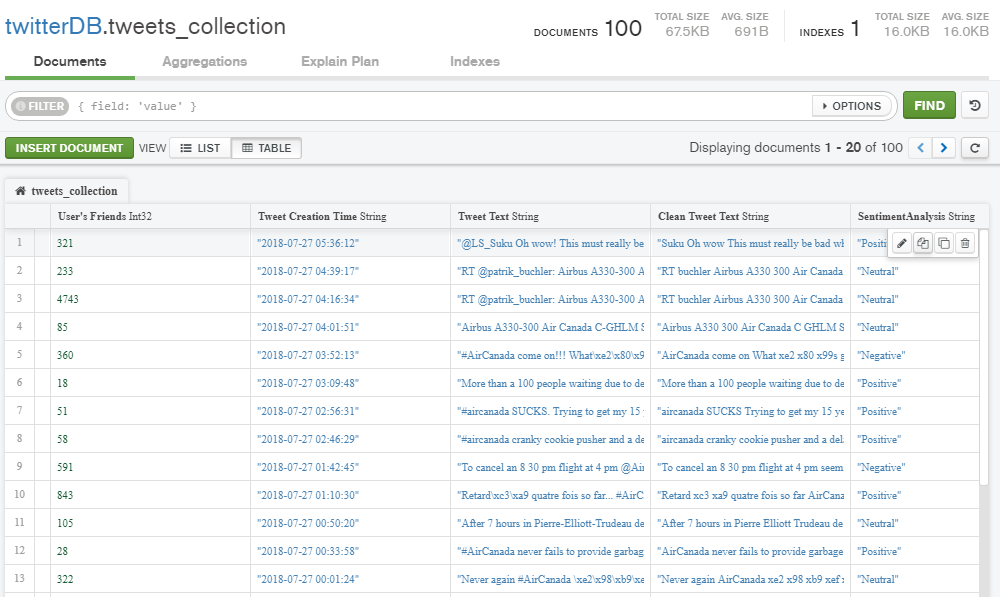
## The Code



## CSV Files



## Database



# References

* <https://www.guru99.com/installation-configuration-mongodb.html>
* <http://jupyter.org/install>
* <http://docs.tweepy.org/en/v3.5.0/>
* <https://api.mongodb.com/python/current/api/>
* <http://textblob.readthedocs.io/en/dev/classifiers.html>
* <https://www.debuggex.com/cheatsheet/regex/python>
* <https://gist.github.com/mprajwala/849b5909f5b881c8ce6a>

# GitHub link

* <https://github.com/sandeep-43/python/tree/master/TwitterSentimentAnalysis>

\*\*\*\*\*