Refence:

<https://www.youtube.com/watch?v=1gQ6uG5Ujiw>

go to minute 1:20:18

from tweepy import API

from tweepy import Cursor

from tweepy.streaming import StreamListener

from tweepy import OAuthHandler

from tweepy import Stream

from textblob import TextBlob

import twitter\_credentials

import matplotlib.pyplot as plt

import numpy as np

import pandas as pd

import re

# # # # TWITTER CLIENT # # # #

class TwitterClient():

def \_\_init\_\_(self, twitter\_user=None):

self.auth = TwitterAuthenticator().authenticate\_twitter\_app()

self.twitter\_client = API(self.auth)

self.twitter\_user = twitter\_user

def get\_twitter\_client\_api(self):

return self.twitter\_client

def get\_user\_timeline\_tweets(self, num\_tweets):

tweets = []

for tweet in Cursor(self.twitter\_client.user\_timeline, id=self.twitter\_user).items(num\_tweets):

tweets.append(tweet)

return tweets

def get\_friend\_list(self, num\_friends):

friend\_list = []

for friend in Cursor(self.twitter\_client.friends, id=self.twitter\_user).items(num\_friends):

friend\_list.append(friend)

return friend\_list

def get\_home\_timeline\_tweets(self, num\_tweets):

home\_timeline\_tweets = []

for tweet in Cursor(self.twitter\_client.home\_timeline, id=self.twitter\_user).items(num\_tweets):

home\_timeline\_tweets.append(tweet)

return home\_timeline\_tweets

# # # # TWITTER AUTHENTICATER # # # #

class TwitterAuthenticator():

def authenticate\_twitter\_app(self):

auth = OAuthHandler(twitter\_credentials.CONSUMER\_KEY, twitter\_credentials.CONSUMER\_SECRET)

auth.set\_access\_token(twitter\_credentials.ACCESS\_TOKEN, twitter\_credentials.ACCESS\_TOKEN\_SECRET)

return auth

# # # # TWITTER STREAMER # # # #

class TwitterStreamer():

"""

Class for streaming and processing live tweets.

"""

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

self.twitter\_autenticator = TwitterAuthenticator()

def stream\_tweets(self, fetched\_tweets\_filename, hash\_tag\_list):

# This handles Twitter authetification and the connection to Twitter Streaming API

listener = TwitterListener(fetched\_tweets\_filename)

auth = self.twitter\_autenticator.authenticate\_twitter\_app()

stream = Stream(auth, listener)

# This line filter Twitter Streams to capture data by the keywords:

stream.filter(track=hash\_tag\_list)

# # # # TWITTER STREAM LISTENER # # # #

class TwitterListener(StreamListener):

"""

This is a basic listener that just prints received tweets to stdout.

"""

def \_\_init\_\_(self, fetched\_tweets\_filename):

self.fetched\_tweets\_filename = fetched\_tweets\_filename

def on\_data(self, data):

try:

print(data)

with open(self.fetched\_tweets\_filename, 'a') as tf:

tf.write(data)

return True

except BaseException as e:

print("Error on\_data %s" % str(e))

return True

def on\_error(self, status):

if status == 420:

# Returning False on\_data method in case rate limit occurs.

return False

print(status)

class TweetAnalyzer():

"""

Functionality for analyzing and categorizing content from tweets.

"""

def clean\_tweet(self, tweet):

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

def analyze\_sentiment(self, tweet):

analysis = TextBlob(self.clean\_tweet(tweet))

if analysis.sentiment.polarity > 0:

return 1

elif analysis.sentiment.polarity == 0:

return 0

else:

return -1

def tweets\_to\_data\_frame(self, tweets):

df = pd.DataFrame(data=[tweet.text for tweet in tweets], columns=['tweets'])

df['id'] = np.array([tweet.id for tweet in tweets])

df['len'] = np.array([len(tweet.text) for tweet in tweets])

df['date'] = np.array([tweet.created\_at for tweet in tweets])

df['source'] = np.array([tweet.source for tweet in tweets])

df['likes'] = np.array([tweet.favorite\_count for tweet in tweets])

df['retweets'] = np.array([tweet.retweet\_count for tweet in tweets])

return df

if \_\_name\_\_ == '\_\_main\_\_':

twitter\_client = TwitterClient()

tweet\_analyzer = TweetAnalyzer()

api = twitter\_client.get\_twitter\_client\_api()

tweets = api.user\_timeline(screen\_name="realDonaldTrump", count=200)

df = tweet\_analyzer.tweets\_to\_data\_frame(tweets)

df['sentiment'] = np.array([tweet\_analyzer.analyze\_sentiment(tweet) for tweet in df['tweets']])

print(df.head(10))