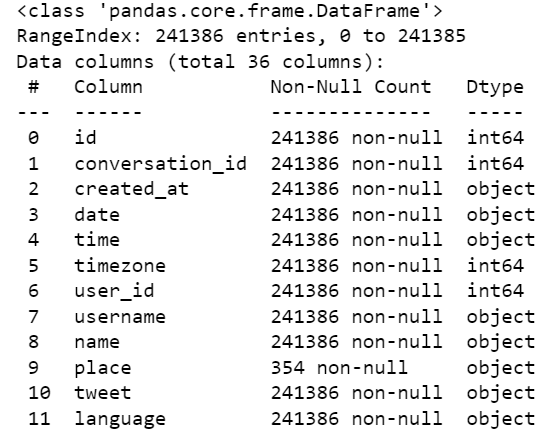
import numpy as np

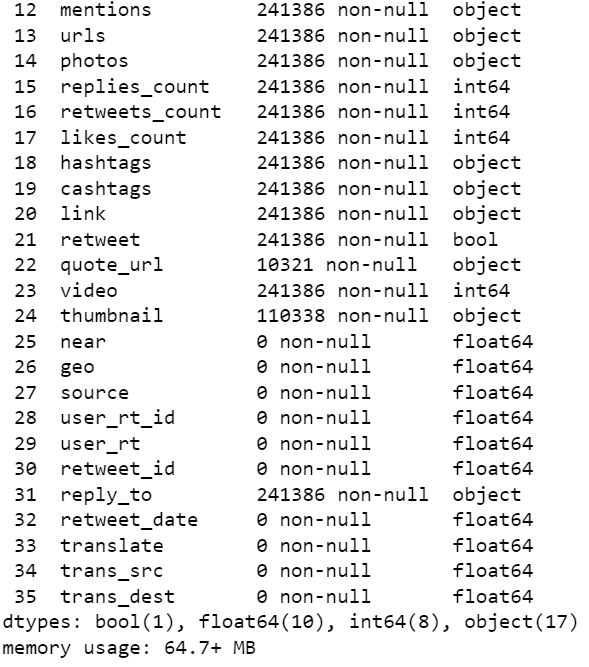
import pandas as pd

df = pd.read\_csv('data\_science.csv')

df.info()

df['tweet'][10]





df['tweet'][10]

output: 'Trends in #AI for next 5 years, including revenue, applications, and talent (#INFOGRAPHIC) ——————— #BigData #DataScience #MachineLearning #DeepLearning #ComputerVision #NLProc #DataLiteracy #AIStrategy #DigitalTransformation #EdgeAI #Edge #IoT #IIoT #IoTPL #IoTCommunity <https://t.co/mn7vFSgyyv'>

import nltk

nltk.download('vader\_lexicon')

from nltk.sentiment.vader import SentimentIntensityAnalyzer

sid = SentimentIntensityAnalyzer()

import re

import pandas as pd

import nltk

nltk.download('words')

words = set(nltk.corpus.words.words())

output: [nltk\_data] Downloading package vader\_lexicon to

[nltk\_data] C:\Users\AARTI\AppData\Roaming\nltk\_data...

[nltk\_data] Package vader\_lexicon is already up-to-date!

[nltk\_data] Downloading package words to

[nltk\_data] C:\Users\AARTI\AppData\Roaming\nltk\_data...

[nltk\_data] Package words is already up-to-date!

sentence = df['tweet'][0]

sid.polarity\_scores(sentence)['compound']

output: -0.1783

def cleaner(tweet):

tweet = re.sub("@[A-Za-z0-9]+","",tweet) #Remove @ sign

tweet = re.sub(r"(?:\@|http?\://|https?\://|www)\S+", "", tweet) #Remove http links

tweet = " ".join(tweet.split())

tweet = tweet.replace("#", "").replace("\_", " ") #Remove hashtag sign but keep the text

tweet = " ".join(w for w in nltk.wordpunct\_tokenize(tweet)

if w.lower() in words or not w.isalpha())

return tweet

df['tweet\_clean'] = df['tweet'].apply(cleaner)

word\_dict = {'manipulate':-1,'manipulative':-1,'jamescharlesiscancelled':-1,'jamescharlesisoverparty':-1,

'pedophile':-1,'pedo':-1,'cancel':-1,'cancelled':-1,'cancel culture':0.4,'teamtati':-1,'teamjames':1,

'teamjamescharles':1,'liar':-1}

import nltk

nltk.download('vader\_lexicon')

from nltk.sentiment.vader import SentimentIntensityAnalyzer

sid = SentimentIntensityAnalyzer()

sid.lexicon.update(word\_dict)

list1 = []

for i in df['tweet\_clean']:

list1.append((sid.polarity\_scores(str(i)))['compound'])

df['sentiment'] = pd.Series(list1)

def sentiment\_category(sentiment):

label = ''

if(sentiment>0):

label = 'positive'

elif(sentiment == 0):

label = 'neutral'

else:

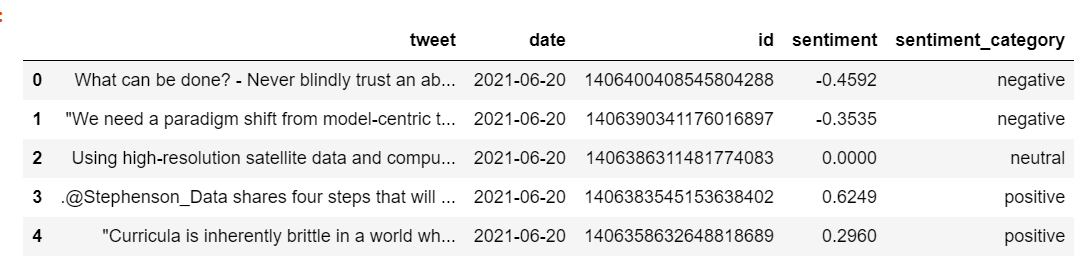
label = 'negative'

return(label)

df['sentiment\_category'] = df['sentiment'].apply(sentiment\_category)

df = df[['tweet','date','id','sentiment','sentiment\_category']]

df.head()



neg = df[df['sentiment\_category']=='negative']

neg = neg.groupby(['date'],as\_index=False).count()

pos = df[df['sentiment\_category']=='positive']

pos = pos.groupby(['date'],as\_index=False).count()

pos = pos[['date','id']]

neg = neg[['date','id']]

import plotly.graph\_objs as go

fig = go.Figure()

for col in pos.columns:

fig.add\_trace(go.Scatter(x=pos['date'], y=pos['id'],

name = col,

mode = 'markers+lines',

line=dict(shape='linear'),

connectgaps=True,

line\_color='green'

)

)

for col in neg.columns:

fig.add\_trace(go.Scatter(x=neg['date'], y=neg['id'],

name = col,

mode = 'markers+lines',

line=dict(shape='linear'),

connectgaps=True,

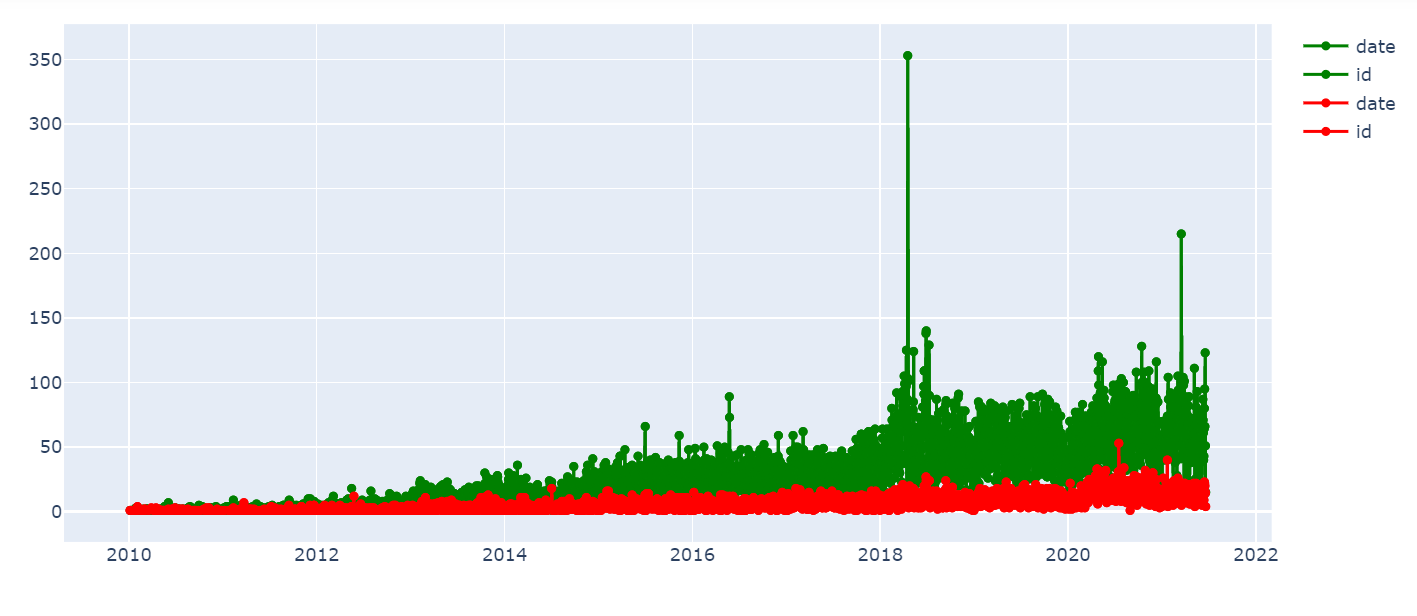
line\_color='red'

)

)

fig.show()

output:



newdf = df[(df['date']>='2019-05-01') & (df['date']<='2019-06-29')]

neg = newdf[newdf['sentiment\_category']=='negative']

neg = neg.groupby(['date'],as\_index=False).count()

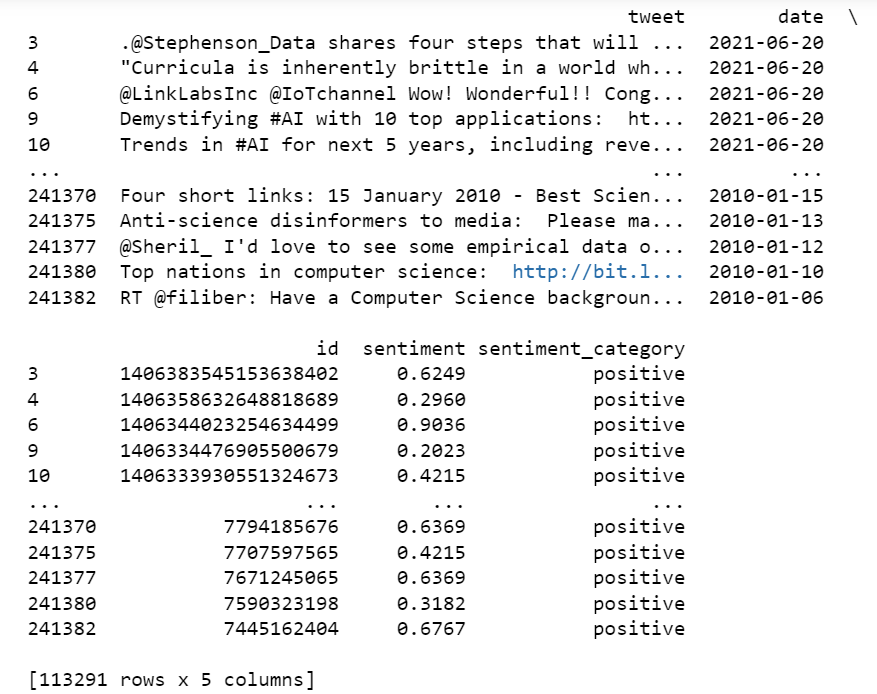
pos = newdf[newdf['sentiment\_category']=='positive']

pos = pos.groupby(['date'],as\_index=False).count()

pos = pos[['date','id']]

neg = neg[['date','id']]

print(df[df['sentiment\_category']=='positive'])



print(df[df['sentiment\_category']=='negative'])

