twitter data

December 11, 2020

FRE 7871_I Fall 2020 HW 2 Part 1

- Jinfeng Hong jh6011
- Eric Sun zs861
- Raye Shen rs6981

[17]:
```
import os
import tweepy as tw
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import warnings
import time
from datetime import datetime
import re
from glob import glob
warnings.filterwarnings('ignore')
```

1 Get data

[2]:
```
consumer_key= 'xxx'
consumer_secret= 'xxx'
access_token= 'xxx'
access_token_secret= 'xxx'
auth = tw.OAuthHandler(consumer_key, consumer_secret)
auth.set_access_token(access_token, access_token_secret)
api = tw.API(auth, wait_on_rate_limit=True)
```

1.0.1 Daily Trade: 2000

[3]:
```
startDates = ["2020-11-29","2020-11-30"] + ["2020-12-"+str(i) for i in␣
↪range(1,5)]
endDates = ["2020-11-30"] + ["2020-12-"+str(i) for i in range(1,6)]
files = ["20201129","20201130"]+["202012" + str(i) for i in range(1,5)]
tickers =␣
↪["EIX","GOOGL","HPE","KHC","EBAY","LIN","MMM","MSCI","NEM","NRG","OKE","PLD"]
```

[4]:
```
for ticker in tickers:
#     search_words = "${} -filter:retweets".format(ticker)
    search_words = "${}".format(ticker)

    for i in range(len(startDates)):
        print("{}: {}{} start".format(i,ticker,files[i]))
        tweets = tw.Cursor(api.search,
                           q=search_words,
                           lang="en",
                           since=startDates[i],
                           until = endDates[i]).items(2000)

        tweet_list = [[tweet.text,tweet.created_at] for tweet in tweets]
        df = pd.DataFrame(tweet_list,columns=['tweets','date'])
        df.to_csv("./stocks3/{}_{}.csv".format(ticker,files[i]),␣
 ↪encoding='utf-8',index=False)
        print("{}: {}{} finish".format(i,ticker,files[i]))
```

```
0: EIX20201129 start
0: EIX20201129 finish
```

2 Preprocessing documents

[172]:
```
paths = glob(".\\stocks2\\*")
```

[219]:
```
df = pd.DataFrame(columns=['tweets','date','ticker'])
for path in paths:
    temp = pd.read_csv(path)

    # remove link and tag @
    temp['tweets'] = temp['tweets'].apply(lambda text: ' '.join(re.
 ↪sub("(@[A-Za-z0-9]+)|([^0-9A-Za-z \t])|(\w+:\/\/\S+)", " ", text).split()))

    # remove duplicates
    temp.drop_duplicates(subset="tweets",keep='last',inplace=True)

    # create a new date
    temp.date = temp.date.apply(lambda x: x.split()[0])

    # create a ticker column
    temp['ticker']=path.split("\\")[-1].split('_')[0]

    # lower case of all words
    temp.tweets = temp.tweets.str.lower()

    # remove numbers
```

```
    temp.tweets=temp.tweets.str.replace(r"\d",'')

    # remove single characters
    temp.tweets = temp.tweets.str.replace(r'\W*\b\w{1}\b', '')

    # remove additional space from string
    temp.tweets = temp.tweets.str.replace(' +', ' ')

    # remove stopwords
    STOPWORDS = "i, me, my, myself, we, our, ours, ourselves, you, you're,␣
 ↪you've, you'll, you'd, your, yours, \
    yourself, yourselves, he, him, his, himself, she, she's, her, hers,␣
 ↪herself, it, it's, its, itself, \
    they, them, their, theirs, themselves, what, which, who, whom, this, that,␣
 ↪that'll, these, those,\
    a, an, the, d, ll, m, o, re, ve, y, ".split(', ')

    def remove_stopwords(text):
        return " ".join([word for word in str(text).split() if word not in␣
 ↪STOPWORDS])

    temp.tweets = temp.tweets.apply(lambda text: remove_stopwords(text))

    df = df.append(temp,ignore_index=True)
#     df=df['tweets'].str.split(' ',n=-1,expand=False)
    df = df.sort_values(by=['ticker'])
    df.reset_index(drop=True,inplace=True)
df
```

[219]:

	tweets	date	ticker
0	ebay will see same kind of numbers improvement…	2020-11-29	EBAY
1	rt htsc ev nickel play about to run big they o…	2020-12-02	EBAY
2	rt cybermonday take off everything mercadomagi…	2020-12-01	EBAY
3	ebay nov calls up alerted at on nov pm peak af…	2020-12-03	EBAY
4	ebay entry target stop below	2020-12-03	EBAY
…	…	…	…
4143	wed squared below and day price channel with g…	2020-12-03	PLD
4144	top reit stocks with market cap over billion i…	2020-12-02	PLD
4145	pld in downtrend stochastic indicator sits in …	2020-11-29	PLD
4146	pld day moving average broke below day moving …	2020-12-03	PLD
4147	swing em if want em thestrat inside week adi u…	2020-11-30	PLD

[4148 rows x 3 columns]

2.1 vader package

[220]:
```
from vaderSentiment.vaderSentiment import SentimentIntensityAnalyzer
```

[221]:
```
# Create a SentimentIntensityAnalyzer object.
sid_obj = SentimentIntensityAnalyzer()

scores = {"neg":[],"neu":[],"pos":[],"compound":[]}

for text in df.tweets:
    score = sid_obj.polarity_scores(text)
    scores['neg'].append(score['neg'])
    scores['neu'].append(score['neu'])
    scores['pos'].append(score['pos'])
    scores['compound'].append(score['compound'])

df2 = pd.DataFrame(scores)
```

[222]:
```
result = pd.concat([df,df2],axis=1)
result
```

[222]:

	tweets	date	ticker	neg	neu	pos	compound
0	ebay will see same kind of numbers improvement…	2020-11-29	EBAY	0.000	0.769	0.231	0.4588
1	rt htsc ev nickel play about to run big they o…	2020-12-02	EBAY	0.000	0.893	0.107	0.3400
2	rt cybermonday take off everything mercadomagi…	2020-12-01	EBAY	0.000	1.000	0.000	0.0000
3	ebay nov calls up alerted at on nov pm peak af…	2020-12-03	EBAY	0.000	0.845	0.155	0.2960
4	ebay entry target stop below	2020-12-03	EBAY	0.355	0.645	0.000	-0.2960
…	…	…	…	…	…	…	…
4143	wed squared below and day price channel with g…	2020-12-03	PLD	0.109	0.674	0.218	0.4767
4144	top reit stocks with market cap over billion i…	2020-12-02	PLD	0.000	0.886	0.114	0.2023
4145	pld in downtrend stochastic indicator sits in …	2020-11-29	PLD	0.000	1.000	0.000	0.0000
4146	pld day moving average broke below day moving …	2020-12-03	PLD	0.157	0.843	0.000	-0.4215
4147	swing em if want em thestrat inside week adi u…	2020-11-30	PLD	0.000	0.929	0.071	0.0772

```
[4148 rows x 7 columns]
```

[223]:
```
result.to_csv("result_full.csv")
```