

In [1]:

```
import pandas as pd
import numpy as np
import warnings
import re

# Visualisation
import matplotlib.pyplot as plt
import matplotlib
import seaborn as sns
from IPython.display import display

#nltk
from nltk.stem import WordNetLemmatizer
from nltk.sentiment.vader import SentimentIntensityAnalyzer
from nltk.sentiment.util import *
from nltk import tokenize
from wordcloud import WordCloud, STOPWORDS

matplotlib.style.use('ggplot')
pd.options.mode.chained_assignment = None
warnings.filterwarnings("ignore")

%matplotlib inline

tweets=pd.read_csv("Desktop/Next_Tweets.csv" , encoding = "ISO-8859-1")
```

In [4]:

```
tweets.head(5)
tweets["Tweet"][3]
```

Out[4]:

```
'@nextofficial Looking for any chairs in Boucle weave light dove grey plea
se x'
```

Preprocessing del the @ and other punctuation

In [5]:

```

tweets['tweetos'] = ''

#add tweetos first part
for i in range(len(tweets['Tweet'])):
    try:
        tweets['tweetos'][i] = tweets['Tweet'].str.split(' ')[i][0]
    except AttributeError:
        tweets['tweetos'][i] = 'other'

#Preprocessing tweetos. select tweetos contains 'RT @'
for i in range(len(tweets['Tweet'])):
    if tweets['tweetos'].str.contains('@')[i] == False:
        tweets['tweetos'][i] = 'other'

# remove URLs, and twitter handles
for i in range(len(tweets['Tweet'])):
    tweets['Tweet'][i] = " ".join([word for word in tweets['Tweet'][i].split()
                                   if 'http' not in word and '@' not in word and '<' not in
n word])

#tweets['Tweet'][6]

# put the text into lower case and delete the doublespace

tweets['Tweet'] = tweets['Tweet'].apply(lambda x: re.sub('[!@#$.;,:?&]', '', x.lower()))
tweets['Tweet'] = tweets['Tweet'].apply(lambda x: re.sub(' ', ' ', x))
tweets['Tweet'][13]

```

Out[5]:

```

'stop using maldon you could end up with a £60 parking ticket that won\x92
t refund'

```

Visualization with the wordcount

In [9]:

```
def wordcloud(tweets,col):
    stopwords = set(STOPWORDS)
    wordcloud = WordCloud(background_color="white",stopwords=stopwords,random_state = 2
016).generate(" ".join([i for i in tweets[col]]))
    plt.figure( figsize=(20,10), facecolor='k')
    plt.imshow(wordcloud)
    plt.axis("off")
    plt.title("Good Morning Datascience+")
wordcloud(tweets,'Tweet')
```



Sentiment Analysis

In [10]:

```
tweets['Tweet_lem'] = [''.join([WordNetLemmatizer().lemmatize(re.sub('[^A-Za-z]', ' ',
line)) for line in lists]).strip() for lists in tweets['Tweet']]
sid = SentimentIntensityAnalyzer()
tweets['sentiment_compound_polarity']=tweets.Tweet_lem.apply(lambda x:sid.polarity_scores(x)['compound'])
tweets['sentiment_neutral']=tweets.Tweet_lem.apply(lambda x:sid.polarity_scores(x)['neu
u'])
tweets['sentiment_negative']=tweets.Tweet_lem.apply(lambda x:sid.polarity_scores(x)['ne
g'])
tweets['sentiment_pos']=tweets.Tweet_lem.apply(lambda x:sid.polarity_scores(x)['pos'])
tweets['sentiment_type']=''
tweets.loc[tweets.sentiment_compound_polarity>0,'sentiment_type']='POSITIVE'
tweets.loc[tweets.sentiment_compound_polarity==0,'sentiment_type']='NEUTRAL'
tweets.loc[tweets.sentiment_compound_polarity<0,'sentiment_type']='NEGATIVE'

tweets_sentiment = tweets.groupby(['sentiment_type'])['sentiment_neutral'].count()
tweets_sentiment.rename("",inplace=True)
explode = (1, 0, 0)
plt.subplot(221)
tweets_sentiment.transpose().plot(kind='barh',figsize=(20, 20))
plt.title('Sentiment Analysis 1', bbox={'facecolor':'0.8', 'pad':0})
plt.subplot(222)
tweets_sentiment.plot(kind='pie',figsize=(20, 20),autopct='%1.1f%%',shadow=True,explode
=explode)
plt.legend(bbox_to_anchor=(1, 1), loc=3, borderaxespad=0.)
plt.title('Sentiment Analysis 2', bbox={'facecolor':'0.8', 'pad':0})
plt.show()
```

