Introduction

For this simple excercise we arre requireed to use tweepy to access twitter extract information from it perform a sentiment analysis and display our findings I will perform and display the results of sentiment analysis using textblob and the NaiveBayes analyser. The twitter accounts chosen are:

- Dave2D: tech youtuber
- JColeNC: the twitter handle of the hip-hop rapper Jermaine Cole
- MKBHD: tech youtuber
- BBCNews: one of the accounts for the broadcasting company

Connecting to the twitter developer account

In [1]:
```
import tweepy
import keys
```

In [2]:
```
auth = tweepy.OAuthHandler(keys.consumer_key, keys.consumer_secret)
```

In [3]:
```
auth.set_access_token(keys.access_token,keys.access_token_secret)
```

In [4]:
```
api= tweepy.API(auth,wait_on_rate_limit=True,wait_on_rate_limit_notify=True)
```

Now that we have connected to twitter we can go ahead and get the information we need for our accounts.

Performing Sentiment Analysis Using TextBlob

In [5]:
```
from textblob import TextBlob
```

In [6]:
```
dave_acc =  api.get_user('Dave2D')
jcole_acc =  api.get_user('JColeNC')
mkbhd_acc =  api.get_user('MKBHD')
bbc_acc =  api.get_user('BBCNews')
```

In [7]:
```
dave_acc.name
```

Out[7]:
```
'Dave Lee'
```

In [8]:
```
jcole_acc.name
```

Out[8]:
```
'J. Cole'
```

In [9]:
```
mkbhd_acc.name
```

Out[9]:
```
'Marques Brownlee'
```

In [10]:
```
bbc_acc.name
```

Out[10]:
```
'BBC News (UK)'
```

Getting the statuses in text

In [11]:
```
bbc_status = bbc_acc.status.text
bbc_status
```

Out[11]:
```
"Titanic letter by 'brave' pastor John Harper sells for £42k https://t.co/ari
xgMzj38"
```

In [12]:
```
jcole_status = jcole_acc.status.text
jcole_status
```

Out[12]:
```
'☁☁☁☁☁☁☁☁☁ DREAMER ☁☁☁☁☁☁☁☁☁ https://t.co/zeWsSTFXSt'
```

In [13]:
```
mkbhd_status = mkbhd_acc.status.text
mkbhd_status
```

Out[13]:
```
'@UltraLinx How’s the wrap holding up? Looks incredibly clean here'
```

In [14]:
```
dave_status = dave_acc.status.text
dave_status
```

Out[14]:
```
'@excessorizeme @imPatrickT that croissant thing looks crazy cool'
```

Analyzing

BBC

In [15]:
```
blob_bbc = TextBlob(bbc_status)
blob_bbc
```

Out[15]:
```
TextBlob("Titanic letter by 'brave' pastor John Harper sells for £42k http
s://t.co/arixgMzj38")
```

In [16]:
```
blob_bbc.sentiment
```

Out[16]:
```
Sentiment(polarity=0.8, subjectivity=1.0)
```

JCole

In [17]:
```
blob_cole = TextBlob(jcole_status)
blob_cole
```

Out[17]:
```
TextBlob("☁☁☁☁☁☁☁☁☁ DREAMER ☁☁☁☁☁☁☁☁☁ https://t.co/ze
WsSTFXSt")
```

In [18]:
```
blob_cole.sentiment
```

Out[18]:
```
Sentiment(polarity=0.0, subjectivity=0.0)
```

MKBHD

In [19]:
```
blob_mkbhd = TextBlob(mkbhd_status)
blob_mkbhd
```

Out[19]:
```
TextBlob("@UltraLinx How’s the wrap holding up? Looks incredibly clean here")
```

In [20]:
```
blob_mkbhd.sentiment
```

Out[20]:
```
Sentiment(polarity=0.3666666666666667, subjectivity=0.7000000000000001)
```

Dave2D

In [21]:
```
blob_dave = TextBlob(dave_status)
blob_dave
```

Out[21]:
```
TextBlob("@excessorizeme @imPatrickT that croissant thing looks crazy cool")
```

In [22]:
```
blob_dave.sentiment
```

Out[22]:
```
Sentiment(polarity=-0.125, subjectivity=0.775)
```

Performing Sentiment Analysis Using NaiveBayesAnalyser

In [23]:
```
from textblob.sentiments import NaiveBayesAnalyzer
```

In [24]:
```
def blob_info(b):
    p = b.polarity
    s = b.subjectivity
    if p == 0.0:
        pr = "Neutral"
    elif p > 0:
        pr = "Positive"
    elif p < 0:
        pr = "Negative"
    if s == 0.0:
        sr = "Subjective"
    else:
        sr = "Objective"
    print('Polarity: {}({})\nSubjectivity: {}({})'.format(pr,p,sr,s))
```

In [25]:
```
def sentiment(userlist):
    for u in userlist:
        current = api.get_user(u)
        print(current.name)
        print(current.status.text)
        blob_info(TextBlob(current.status.text))
        print(TextBlob(current.status.text, analyzer=NaiveBayesAnalyzer()).
        sentiment)
        print('---------------------------------')
```

In [26]:
```
sentiment(['Dave2D','MKBHD','JColeNC','BBCNews'])
```

```
Dave Lee
@excessorizeme @imPatrickT that croissant thing looks crazy cool
Polarity: Negative(-0.125)
Subjectivity: Objective(0.775)
Sentiment(classification='neg', p_pos=0.10646539641654, p_neg=0.8935346035834
612)
---------------------------------
Marques Brownlee
@UltraLinx How’s the wrap holding up? Looks incredibly clean here
Polarity: Positive(0.3666666666666667)
Subjectivity: Objective(0.7000000000000001)
Sentiment(classification='pos', p_pos=0.5893107533046477, p_neg=0.41068924669
53528)
---------------------------------
J. Cole
☁☁☁☁☁☁☁☁☁ DREAMER ☁☁☁☁☁☁☁☁☁ https://t.co/zeWsSTFXSt
Polarity: Neutral(0.0)
Subjectivity: Subjective(0.0)
Sentiment(classification='neg', p_pos=0.16666666666666646, p_neg=0.8333333333
33333)
---------------------------------
BBC News (UK)
Titanic letter by 'brave' pastor John Harper sells for £42k https://t.co/arix
gMzj38
Polarity: Positive(0.8)
Subjectivity: Objective(1.0)
Sentiment(classification='pos', p_pos=0.9311272609078344, p_neg=0.06887273909
216433)
---------------------------------
```

Account	Status	Classification	Positive	Negative
Dave Lee	@excessorizeme @imPatrickT that croissant thing looks crazy cool	Negative	0.10646539641654	0.8935346035834612
Marques Brownlee	@UltraLinx How’s the wrap holding up? Looks incredibly clean here	Positive	0.5893107533046477	0.4106892466953528
J. Cole	☁☁☁☁☁☁☁☁☁ DREAMER ☁☁☁ ☁☁☁☁☁☁ https://t.co/zeWsSTFXSt (https://t.co/zeWsSTFXSt)	Neutral	0.16666666666666646	0.833333333333
BBC News (UK)	Titanic letter by 'brave' pastor John Harper sells for £42k https://t.co/arixgMzj38 (https://t.co/arixgMzj38)	Positive	0.9311272609078344	0.06887273909216433

Conclusion

The sentiment analysis on the twitter statuses highlights afew limitations to the analysers we have:

```
* Emojis cloud the ability of the analyser to best interprete the text
* The analysers however advanced aren't able to pick up on certain slang such as 'c
razy cool' being flagged as negative instead of a positive
```

In [ ]: