Python Code: Airline Sentiment Analysis to understand Users' Perspective

Social media such as twitter has gained popularity nowadays. Many people connect using social media such as Twitter and provide their opinion on a wide variety of subjects. Analyzing opinion expressed by people on various subjects play a very important role in market analysis and to understand the feedback given by the users on services. This analysis also helps to understand where one is standing among various competing companies.

In this article, one such example is explored. The tweets related to airlines are collected for the purpose of analysis. You can obtain various keys required for accessing twitter by creating a new app and generate four access codes such as consumer key, consumer secret, access token, and access token secret. For further help visit https://developer.twitter.com/en/apps.

The python modules required for tweet preprocessing are

Use these keys

```
consumer_key= 'your key'
consumer_secret= 'your key'
access_token= 'your key'
access_token_secret= 'your key'
```

Next you can invoke twitter API:

```
authemtication = tw.OAuthHandler(consumer_key, consumer_secret)
authemtication.set_access_token(access_token, access_token_secret)
api = tw.API(authemtication, wait_on_rate_limit=True)
```

It is possible to search tweet on a given term as:

```
searchKeyWord= "@easyjet"
tweetsOfKeyword = tw.Cursor(api.search, q=searchKeyWord, lang="en",since='2018-11-01').items(10)
for tweet in tweetsOfKeyword:
  print([tweet.created_at.strftime("%Y-%m-%d %H:%M"), tweet.text])
```

Examples of tweets collected are:

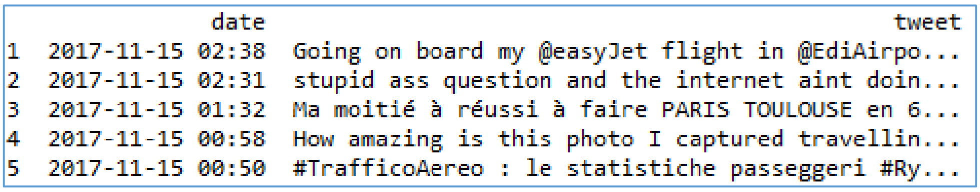
```
2017-11-15 02:38;"Going on board my @easyJet flight in @EdiAirport this afternoon
2017-11-15 02:31;"stupid ass question and the internet aint doing SHIT but can i bring hair straighteners
in a cabin bag when flying with eastjet? #ok #flying #Easyjet"
2017-11-15 01:32;"Ma moitié à réussi à faire PARIS TOULOUSE en 6 heures #easyjet easylate? En #TGV
Atlantique il aurait gagné en temps et confort ! Bad choice"
2017-11-15 00:58;"How amazing is this photo I captured travelling from Glasgow to Belfast!!
Outstanding views
```

```
2017-11-15 00:50;"#TrafficoAereo : le statistiche passeggeri #Ryanair e #EasyJet di Settembre 2017
https://www. trasportinfo.com/2017/11/14/tra ffico-aereo-statistiche-ryanair-easyjet-settembre-2017/
… #LowCost #Statistiche pic.twitter.com/cHLb4qLmwT"
2017-11-15 00:21;"#Easyjet wil tegen 2037 elke vlucht onder twee uur elektrisch doen http://www.
dutchcowboys.nl/technology/eas yjet-wil-tegen-2037-elke-vlucht-onder-twee-uur-elektrisch-doen … |
#Tech"
```

We can read into a dataframe the text of tweets as:

```
data=pd.read_csv('intweets.txt', sep=';' , header=None)
data.columns=['date','tweet']
```

The data which is a two columns one for date another for tweet. The data.head() would display

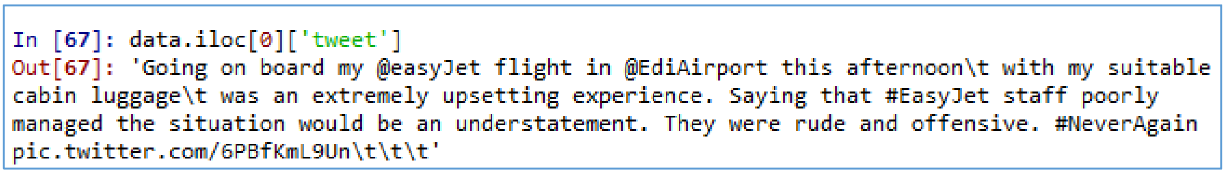


```
               date                                            tweet
1  2017-11-15 02:38  Going on board my @easyJet flight in @EdiAirpo...
2  2017-11-15 02:31  stupid ass question and the internet aint doin...
3  2017-11-15 01:32  Ma moitié à réussi à faire PARIS TOULOUSE en 6...
4  2017-11-15 00:58  How amazing is this photo I captured travellin...
5  2017-11-15 00:50  #TrafficoAereo : le statistiche passeggeri #Ry...
```

These tweets need to be filtered and preprocessed before carrying out the sentiment computation. Usually a tweet may contain urls with www followed by text, @followed by some text are replaced by space. Also the word starting with number is not very much significant for sentiment analysis so they also replace by space. The following function will carry out these tasks:

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

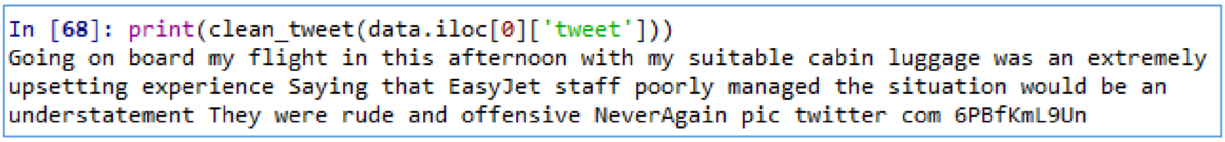
The following tweet is preprocessed.



```
In [67]: data.iloc[0]['tweet']
Out[67]: 'Going on board my @easyJet flight in @EdiAirport this afternoon\t with my suitable
cabin luggage\t was an extremely upsetting experience. Saying that #EasyJet staff poorly
managed the situation would be an understatement. They were rude and offensive. #NeverAgain
pic.twitter.com/6PBfKmL9Un\t\t\t'
```

The result of preprocessing can be obtained by

print(preprocess_tweet(data.iloc[0]['tweet']))

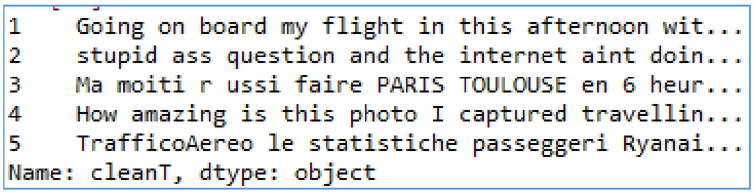


```
In [68]: print(clean_tweet(data.iloc[0]['tweet']))
Going on board my flight in this afternoon with my suitable cabin luggage was an extremely
upsetting experience Saying that EasyJet staff poorly managed the situation would be an
understatement They were rude and offensive NeverAgain pic twitter com 6PBfKmL9Un
```

In order to preprocess all the tweets present in data['tweet'] follow the code:

```
data['cleanT']=data['tweet'].apply(preprocess_tweet)
```

It will add a new column 'cleanT' into data frame after applying function preprocess_tweet() to each tweet. The data['cleanT'].head() would display as:

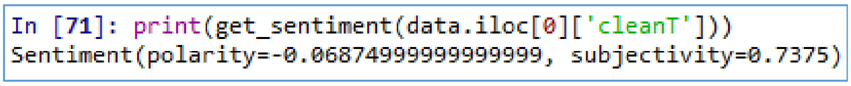


```
1    Going on board my flight in this afternoon wit...
2    stupid ass question and the internet aint doin...
3    Ma moiti r ussi faire PARIS TOULOUSE en 6 heur...
4    How amazing is this photo I captured travellin...
5    TrafficoAereo le statistiche passeggeri Ryanai...
Name: cleanT, dtype: object
```

Textblob is one of the sentiment analyzer. You can install the module as

pip install textblob

Various NLP tasks can be carried out using Textblob, which is built based on NLTK library. We can perform tokenization, lemmatization, POS tagging, converting word to singular or plural etc. Here Textblob is used for sentiment analysis. When Textblob called with text as argument, it will create an object, from which the sentiment information can be extracted as:



```
In [71]: print(get_sentiment(data.iloc[0]['cleanT']))
Sentiment(polarity=-0.06874999999999999, subjectivity=0.7375)
```

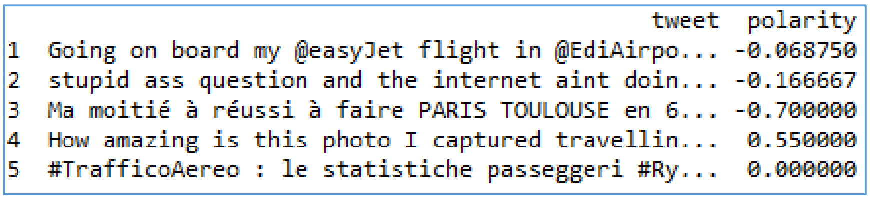
Analyze the sentiment of all the tweet using code:

```
data['sentiment']=data['cleanT'].map(get_sentiment)
```

Tokenize the tweet into its various words and then normalize the words of a tweet by using stemming technique in which word is brought to its root word. Then combined back to a string format and polarity is assigned using sentiment analysis of Textblob:

```
porter=PorterStemmer()
data['tokens']=data['cleanT'].apply(lambda x:x.split())
data['stemmed']=data['tokens'].apply(lambda x: [porter.stem(i) for i in x])
data['cleanStemmedT']=data['stemmed'].apply(lambda x:' '.join(i for i in x))
data['polarity']=data['sentiment'].apply(lambda x: x.polarity)
print(data[['tweet','polarity']].head())
```

Output is:

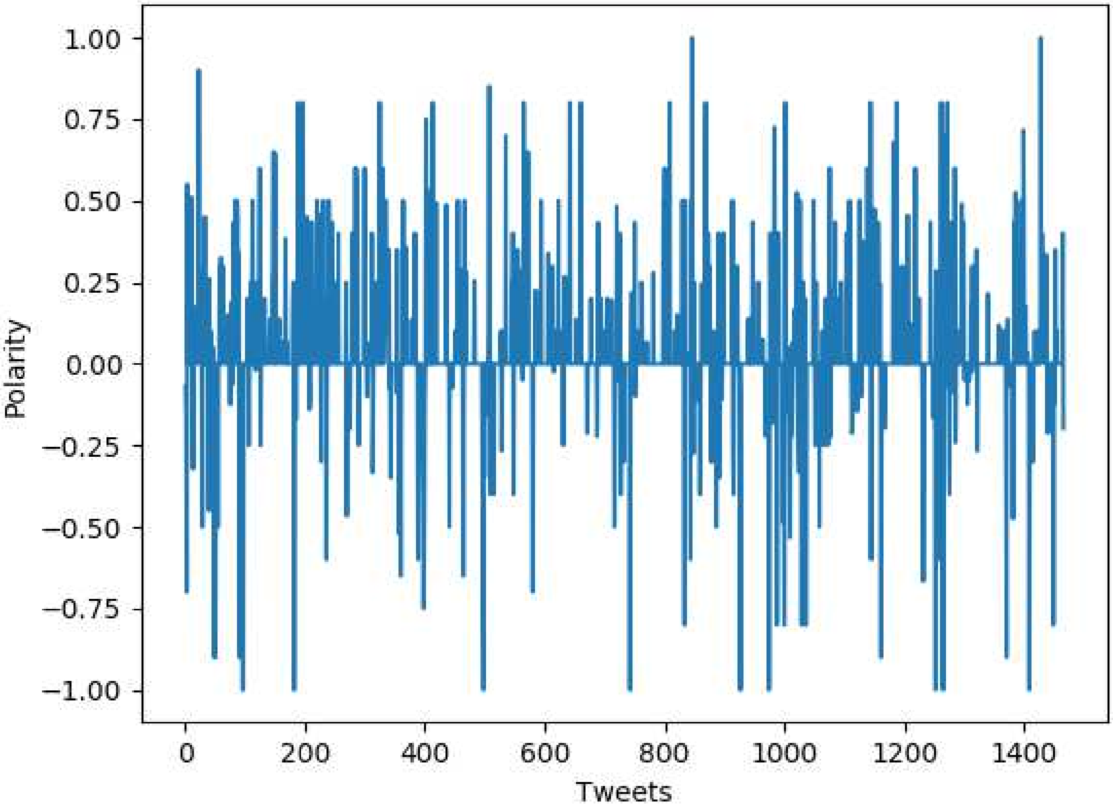


```
                                               tweet  polarity
1  Going on board my @easyJet flight in @EdiAirpo... -0.068750
2  stupid ass question and the internet aint doin... -0.166667
3  Ma moitié à réussi à faire PARIS TOULOUSE en 6... -0.700000
4  How amazing is this photo I captured travellin...  0.550000
5  #TrafficoAereo : le statistiche passeggeri #Ry...  0.000000
```

We can analyze the fluctuation of polarity for the given data set as:

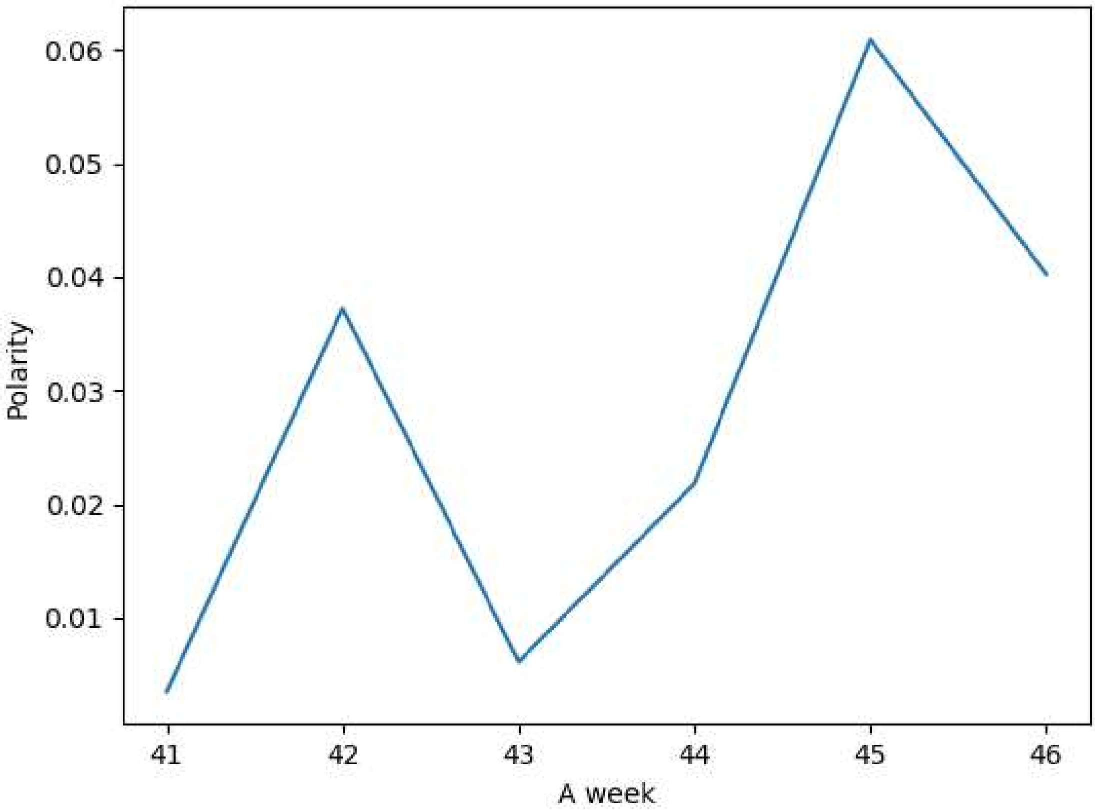
```
fig=plt.figure()
```

```
ax=plt.axes()
ax.plot(data['polarity'])
plt.xlabel('Tweets')
plt.ylabel('Polarity')
plt.show()
```



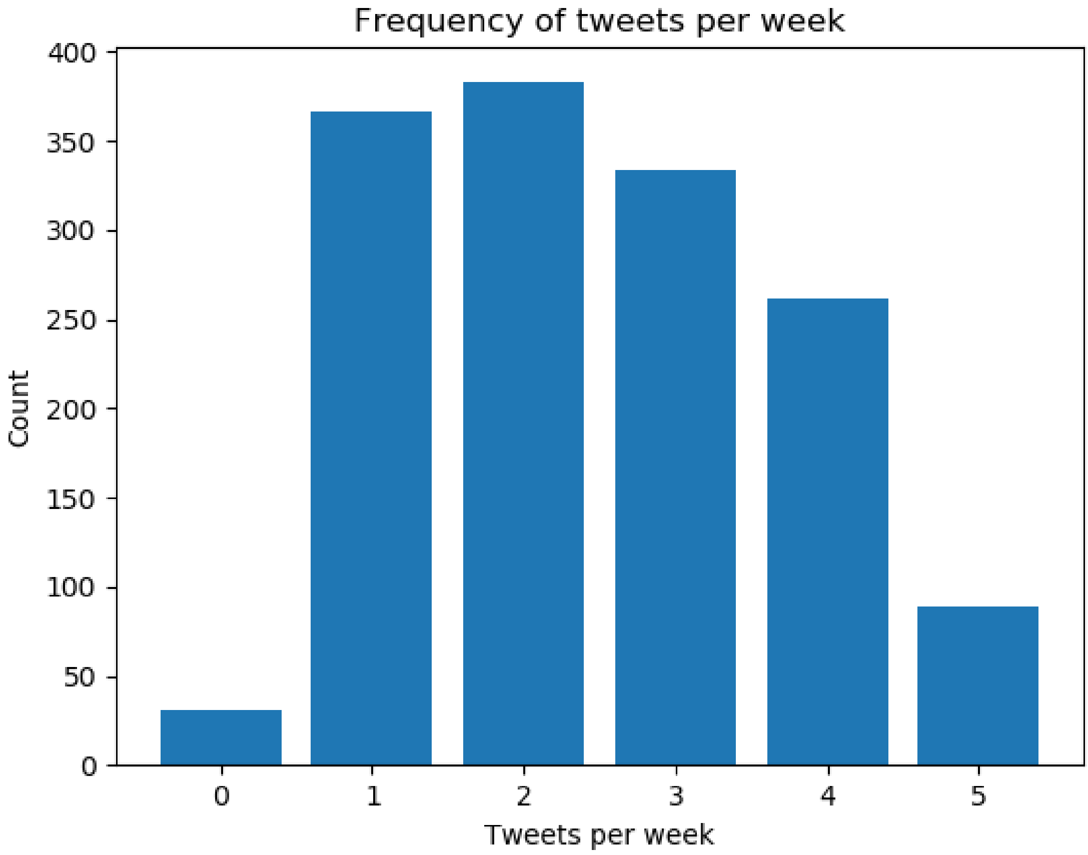
It is also important to know weekly progress in terms of polarity, whether it is going up or down. This can be done using the following code

```
data['date']=pd.to_datetime(data['date'])
countOfTweetsPerWeek=data['polarity'].groupby(data['date'].dt.week).count()
meanOfPolarityPerWeek=data['polarity'].groupby(data['date'].dt.week).mean()
fig=plt.figure()
ax=plt.axes()
ax.plot(meanOfPolarityPerWeek)
plt.xlabel('A week')
plt.ylabel('Polarity')
plt.show()
```



The frequency of tweet per week can be determined and a bar chart for same is drawn using code:

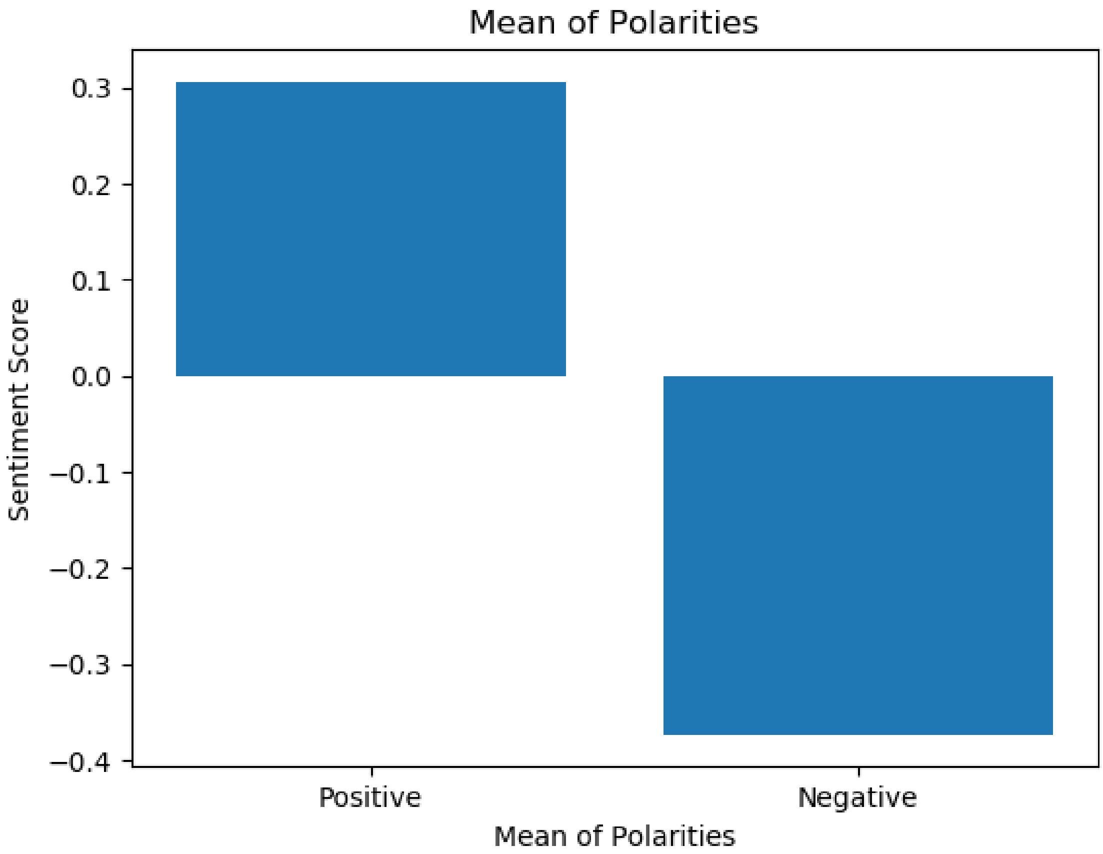
```
index=np.arange(len(countOfTweetsPerWeek))
plt.bar(index,countOfTweetsPerWeek)
plt.xlabel('Tweets per week')
plt.ylabel('Count')
plt.title('Frequency of tweets per week')
plt.show()
```



We can compute the mean values of positive polarities and mean value of negative polarities. It determines for given time duration whether people have given more positive comments or negative comments.

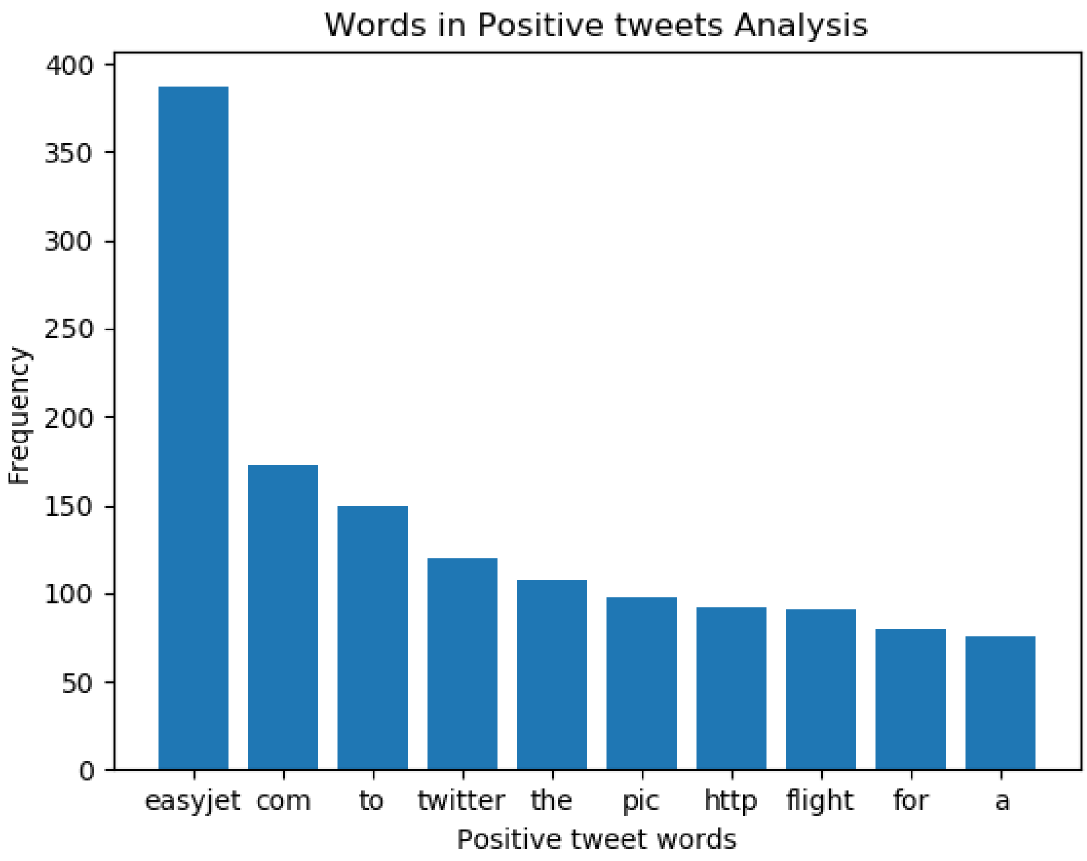
```
positivePolarity=data['polarity'][data['polarity']>0]
negativePolarity=data['polarity'][data['polarity']<0]
meanPositivePolarity=positivePolarity.mean()
meanNegativePolarity=negativePolarity.mean()
index=np.arange(2)
plt.bar(index,[meanPositivePolarity,meanNegativePolarity])
plt.xlabel('Mean of Polarities')
plt.ylabel('Sentiment Score')
plt.xticks(index,['Positive','Negative'])
plt.title('Mean of Polarities')
plt.show()
```

The bar chart is:



Which are words most frequently used in positive tweets can be found using the following code:

```
dataPositivePolarity=data['cleanStemmedT'][data['polarity']>0]
positiveTweetWords=[posTweet.split() for posTweet in dataPositivePolarity]
positiveTweetWords=sum(positiveTweetWords,[])
freqPosWords=nltk.FreqDist(positiveTweetWords)
dfPositiveTweetWords=pd.DataFrame({'word':list(freqPosWords.keys()),'frequency':list(freqPosWords.v
alues())})
top10PositiveTweetWords=dfPositiveTweetWords.nlargest(n=10,columns='frequency')
index=np.arange(len(top10PositiveTweetWords))
plt.bar(index,top10PositiveTweetWords['frequency'])
plt.xlabel('Positive tweet words')
plt.ylabel('Frequency')
plt.xticks(index,top10PositiveTweetWords['word'])
plt.title('Words in Positive tweets Analysis')
plt.show()
```



Conclusion

In this article, the twitter sentiment analysis is carried on the tweets collected for Easyjet. Initially, tweets extraction from the twitter account is described. Then preprocessing of tweets is described along with stemming and tokenization. The sentiment of tweets is computing using the Textblob module. Various visualizations are given such as bar chart for the polarity of each tweet, average progression of polarity per week, the frequency of tweets per week. Mean of positive polarity and negative polarities are computed and displayed using a bar chart. The most frequent words used in positive tweets are identified and their frequency is displayed.

Research Paper

The research on consumer loyalty evaluation using twitter data is appeared in:

http://thesai.org/Publications/ViewPaper?Volume=9&Issue=6&Code=IJACSA&SerialNo=52

Cite the Work

Please cite the following research paper:

Rida Khan, Siddhaling Urologin, "Airline Sentiment Visualization, Consumer Loyalty Measurement and Prediction Using Twitter Data", in International Journal of Advanced Computer Science and Applications (IJACSA), volume 9, issue 6, pp. 380-388, 2018.

Further Projects and Contact

www.researchreader.com

https://medium.com/@dr.siddhaling

Dr. Siddhaling Urolagin,
siddesh_u@yahoo.com