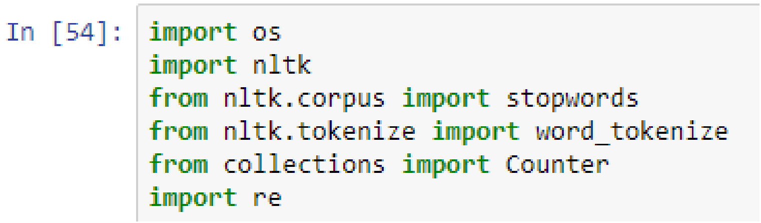
POSITIVE AND NEGATIVE WORDS IN DATA

Let’s proceed in analyzing the data into Sentiment Analysis.

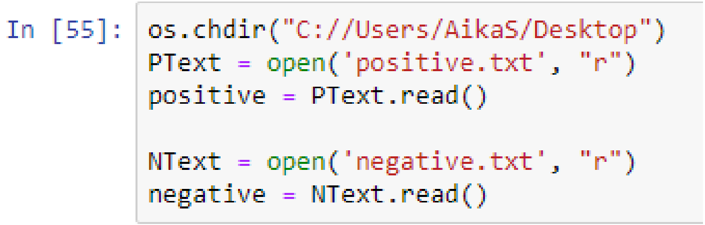
We must first import this packages that we’re going to use.



```
In [54]: import os
         import nltk
         from nltk.corpus import stopwords
         from nltk.tokenize import word_tokenize
         from collections import Counter
         import re
```

We will be having a set of data or a list of words that contain positive and negative words. The data was gathered from an existing dataset for sentiment analysis.

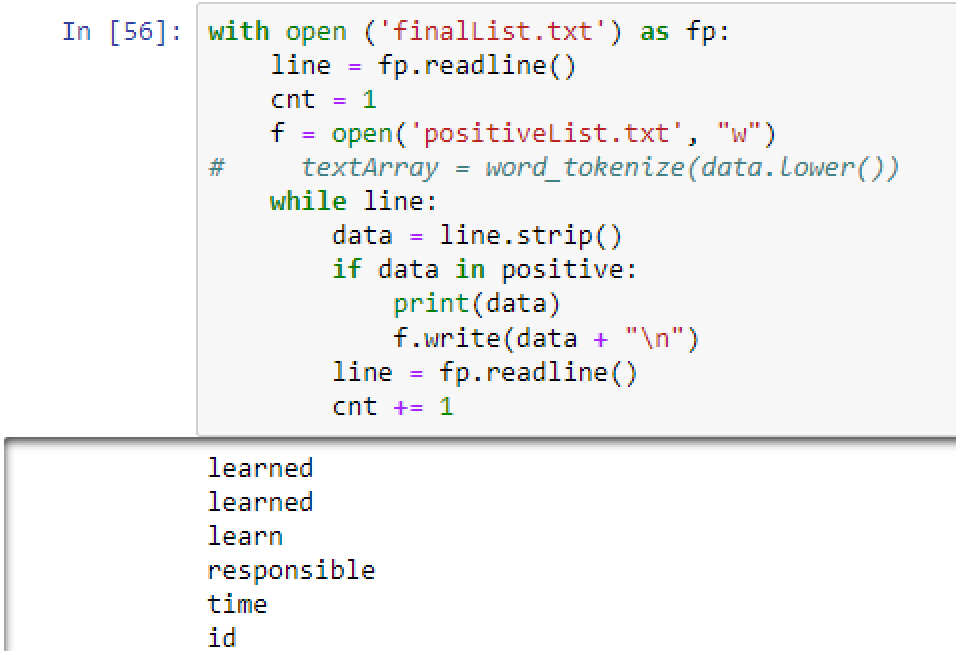
The same process that we did in the data processing, insert the file of the data in our Jupyter Notebook.



```
In [55]: os.chdir("C://Users/AikaS/Desktop")
         PText = open('positive.txt', "r")
         positive = PText.read()

         NText = open('negative.txt', "r")
         negative = NText.read()
```

The list of positive words is inserted in the “positive” variable and “negative” for our list of negative words. By doing this, it will be easier for us to call for the list of words.

Next, we’ll read the cleansed data from the previous module, then create another file which will contain the list of positive words from our finalList.txt



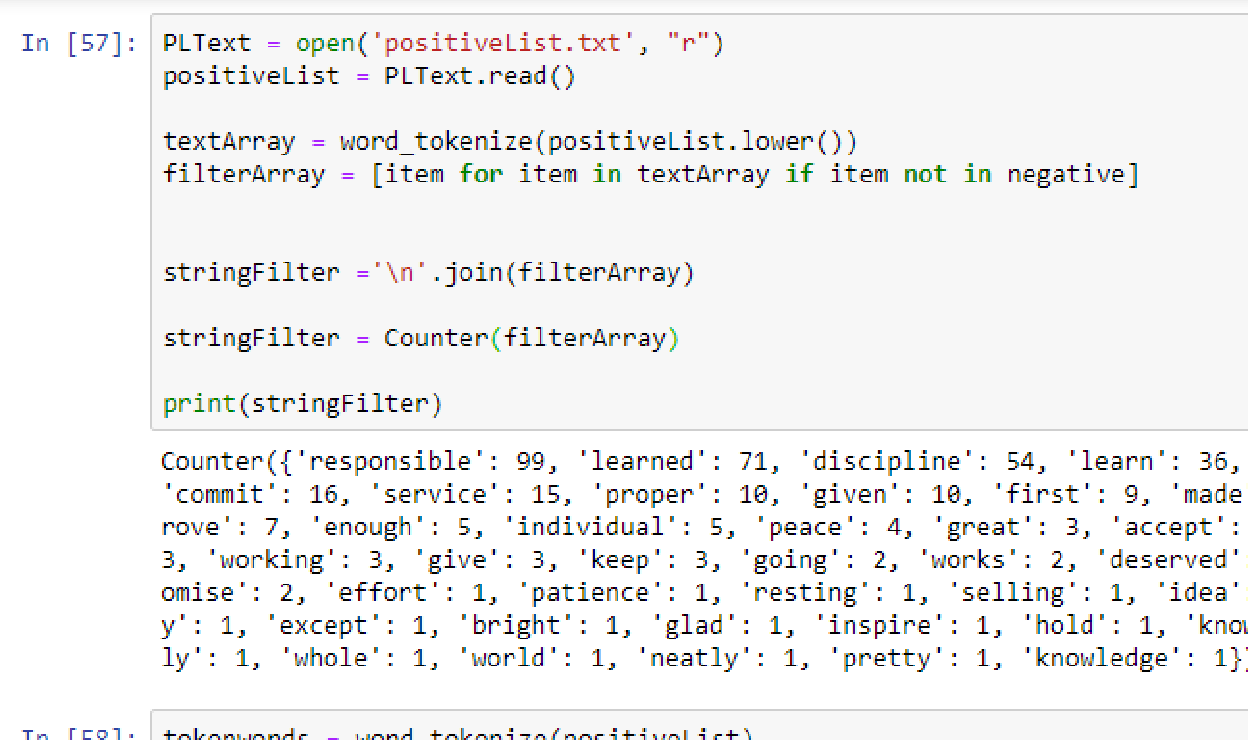
```
In [56]: with open ('finalList.txt') as fp:
             line = fp.readline()
             cnt = 1
             f = open('positiveList.txt', "w")
         #     textArray = word_tokenize(data.lower())
             while line:
                 data = line.strip()
                 if data in positive:
                     print(data)
                     f.write(data + "\n")
                 line = fp.readline()
                 cnt += 1
```

```
         learned
         learned
         learn
         responsible
         time
         id
```

We have identified the list of positive words from the finaList.txt but it may still contain words that are negative, so for that part, we need to remove negative words using negativeList.txt

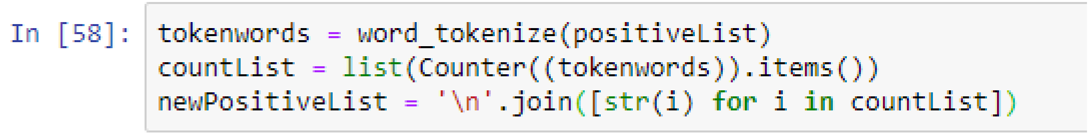
As we created a new file earlier, we're going to tokenize our list and lowercase before filtering the negative words then convert the token words into String by using *stringFilter = '\n'.join(filterArray)*

Then perform *Counter(filterArray)* to count all similar words.

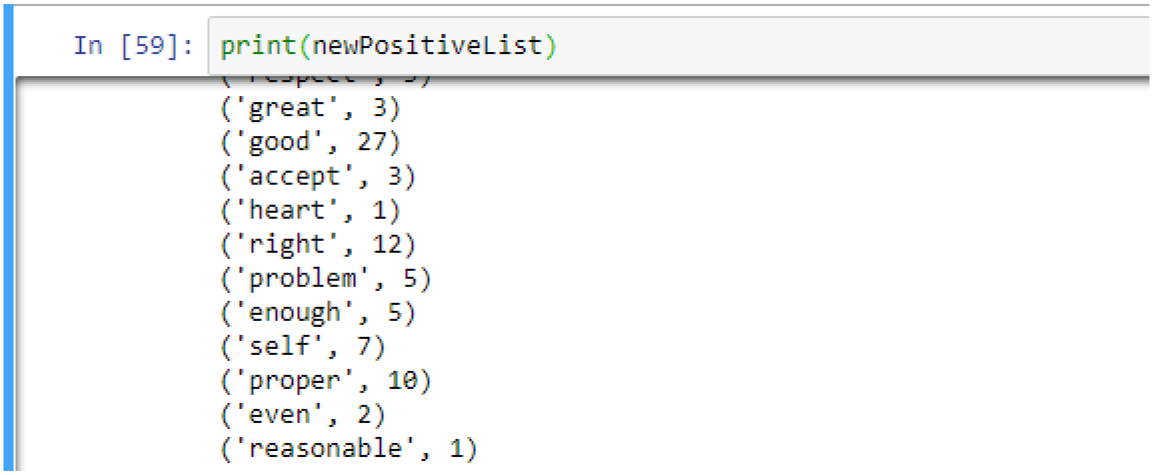


```
In [57]: PLText = open('positiveList.txt', "r")
         positiveList = PLText.read()

         textArray = word_tokenize(positiveList.lower())
         filterArray = [item for item in textArray if item not in negative]


         stringFilter ='\n'.join(filterArray)

         stringFilter = Counter(filterArray)

         print(stringFilter)

         Counter({'responsible': 99, 'learned': 71, 'discipline': 54, 'learn': 36,
         'commit': 16, 'service': 15, 'proper': 10, 'given': 10, 'first': 9, 'made
         rove': 7, 'enough': 5, 'individual': 5, 'peace': 4, 'great': 3, 'accept':
         3, 'working': 3, 'give': 3, 'keep': 3, 'going': 2, 'works': 2, 'deserved'
         omise': 2, 'effort': 1, 'patience': 1, 'resting': 1, 'selling': 1, 'idea'
         y': 1, 'except': 1, 'bright': 1, 'glad': 1, 'inspire': 1, 'hold': 1, 'kno
         ly': 1, 'whole': 1, 'world': 1, 'neatly': 1, 'pretty': 1, 'knowledge': 1}

In [58]: tokenwords = word_tokenize(positiveList)
```

There we have it, it may not be readable so let's transform it into a more readable result.

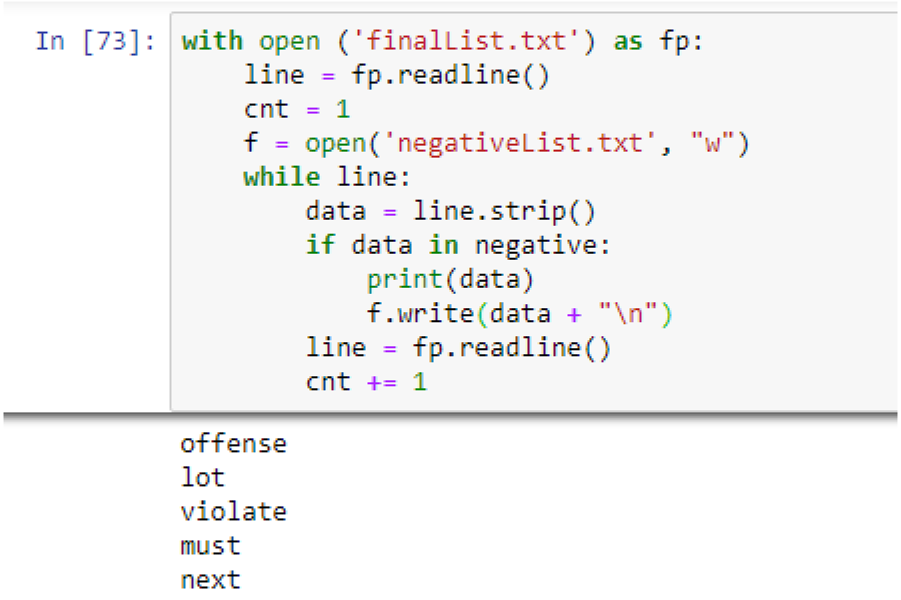


```
In [58]: tokenwords = word_tokenize(positiveList)
         countList = list(Counter((tokenwords)).items())
         newPositiveList = '\n'.join([str(i) for i in countList])
```



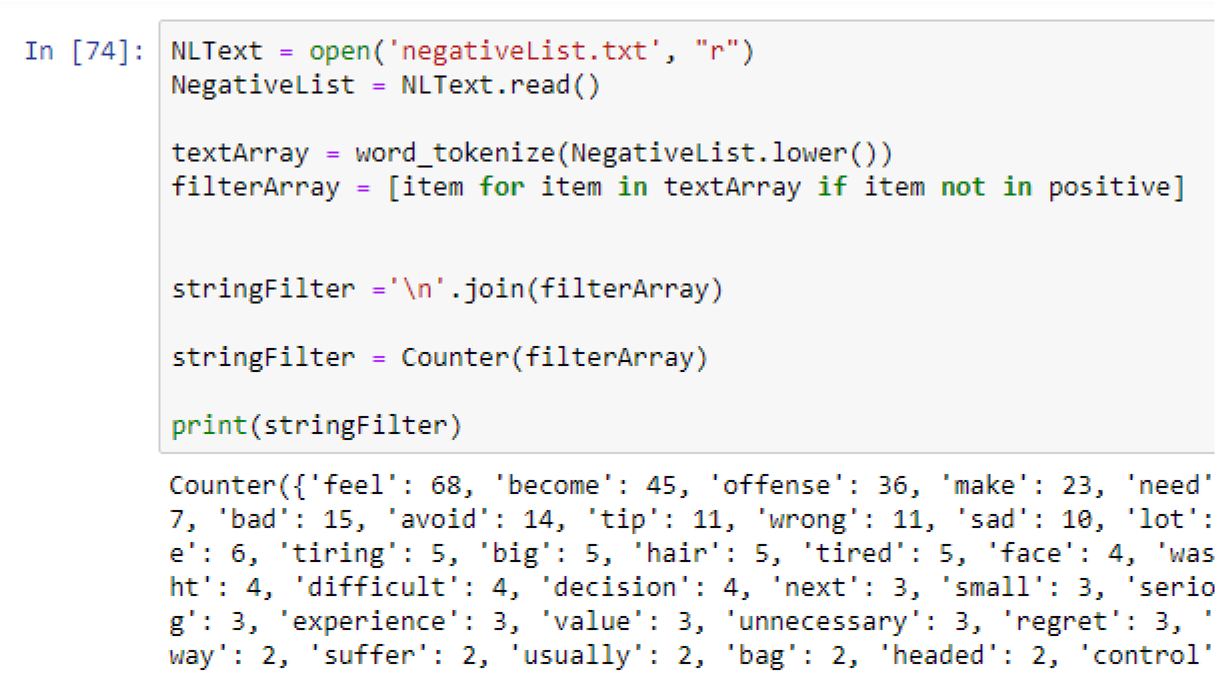
```
In [59]: print(newPositiveList)

         ('respect', 3)
         ('great', 3)
         ('good', 27)
         ('accept', 3)
         ('heart', 1)
         ('right', 12)
         ('problem', 5)
         ('enough', 5)
         ('self', 7)
         ('proper', 10)
         ('even', 2)
         ('reasonable', 1)
```

Now let’s perform the same process for the Negative using negativeList.txt



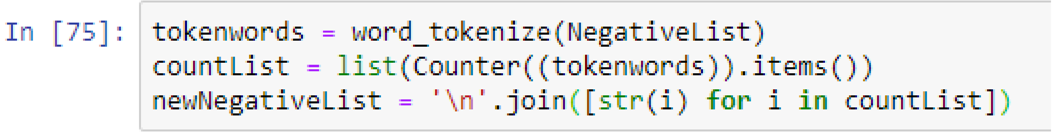
```
In [73]: with open ('finalList.txt') as fp:
             line = fp.readline()
             cnt = 1
             f = open('negativeList.txt', "w")
             while line:
                 data = line.strip()
                 if data in negative:
                     print(data)
                     f.write(data + "\n")
                 line = fp.readline()
                 cnt += 1
```

```
offense
lot
violate
must
next
```



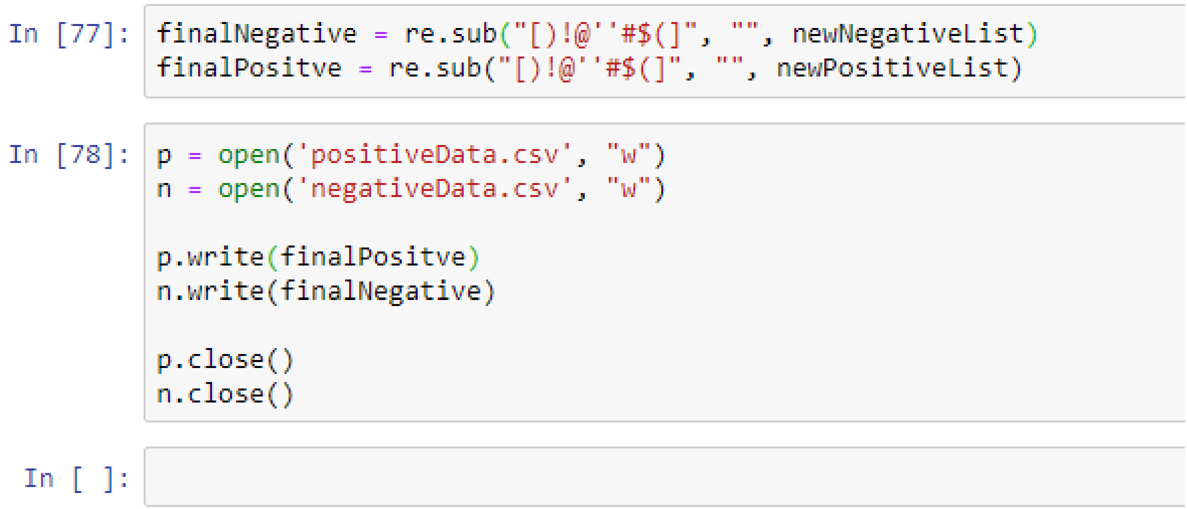
```
In [74]: NLText = open('negativeList.txt', "r")
         NegativeList = NLText.read()

         textArray = word_tokenize(NegativeList.lower())
         filterArray = [item for item in textArray if item not in positive]


         stringFilter ='\n'.join(filterArray)

         stringFilter = Counter(filterArray)

         print(stringFilter)
```

```
Counter({'feel': 68, 'become': 45, 'offense': 36, 'make': 23, 'need'
7, 'bad': 15, 'avoid': 14, 'tip': 11, 'wrong': 11, 'sad': 10, 'lot':
e': 6, 'tiring': 5, 'big': 5, 'hair': 5, 'tired': 5, 'face': 4, 'was
ht': 4, 'difficult': 4, 'decision': 4, 'next': 3, 'small': 3, 'serio
g': 3, 'experience': 3, 'value': 3, 'unnecessary': 3, 'regret': 3, '
way': 2, 'suffer': 2, 'usually': 2, 'bag': 2, 'headed': 2, 'control'
```



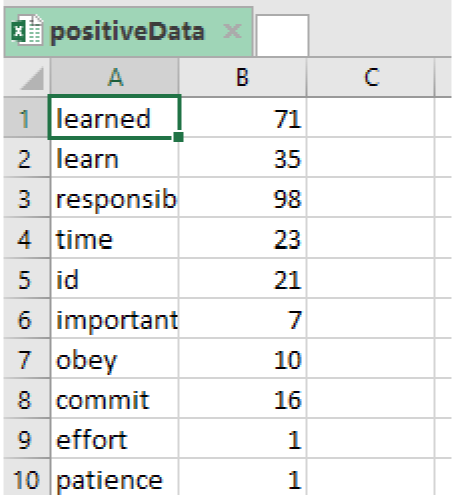
```
In [75]: tokenwords = word_tokenize(NegativeList)
         countList = list(Counter((tokenwords)).items())
         newNegativeList = '\n'.join([str(i) for i in countList])
```

Next, we'll remove the symbols from the result then export the counted positive and negative into CSV files.



```
In [77]: finalNegative = re.sub("[)!@''#$(]", "", newNegativeList)
         finalPositve = re.sub("[)!@''#$(]", "", newPositiveList)

In [78]: p = open('positiveData.csv', "w")
         n = open('negativeData.csv', "w")

         p.write(finalPositve)
         n.write(finalNegative)

         p.close()
         n.close()

In [ ]:
```

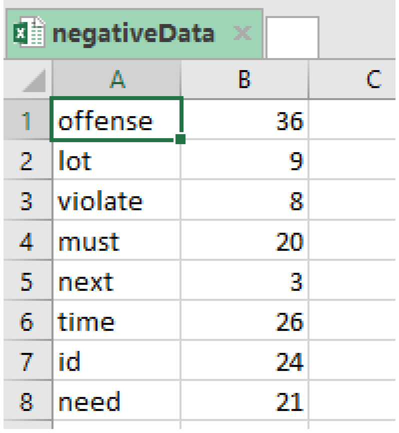
For postiveData.csv



positiveData

	A	B	C
1	learned	71	
2	learn	35	
3	responsib	98	
4	time	23	
5	id	21	
6	important	7	
7	obey	10	
8	commit	16	
9	effort	1	
10	patience	1	

For negativeData.csv



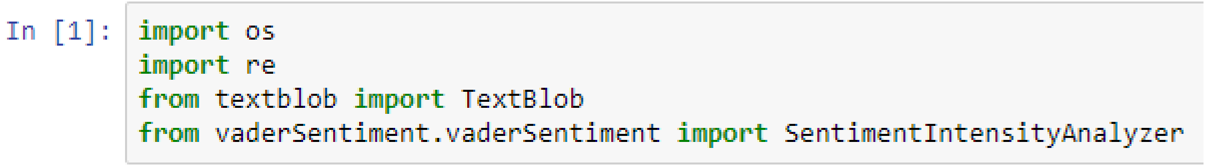
negativeData

	A	B	C
1	offense	36	
2	lot	9	
3	violate	8	
4	must	20	
5	next	3	
6	time	26	
7	id	24	
8	need	21	

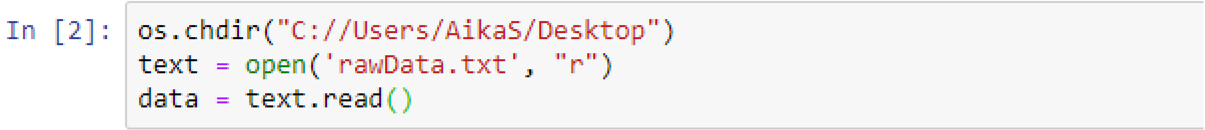
POLARITY AND SENTIMENT

We'll now proceed in identifying the polarity and sentiment of data.

First, import packages that we're going to use and open the raw data as well.



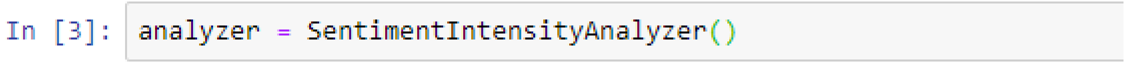
```
In [1]: import os
        import re
        from textblob import TextBlob
        from vaderSentiment.vaderSentiment import SentimentIntensityAnalyzer
```



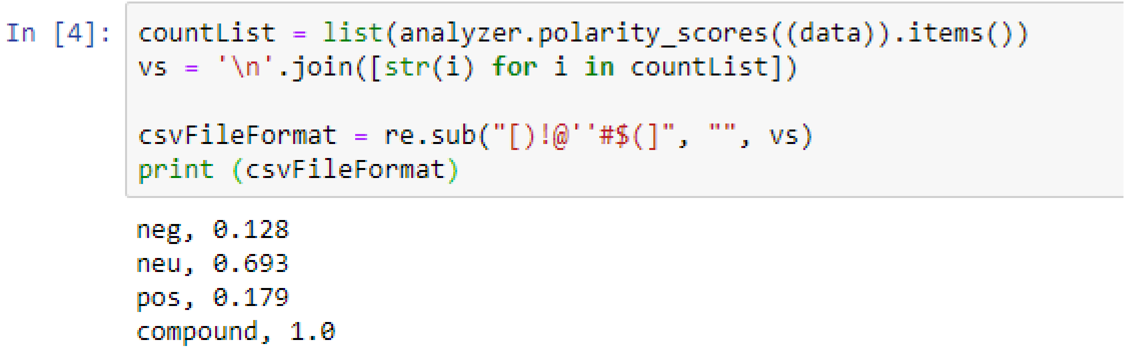
```
In [2]: os.chdir("C://Users/AikaS/Desktop")
        text = open('rawData.txt', "r")
        data = text.read()
```

The raw Data contains each line a sentence that we're going to process. Let's set a new variable

Analyzer = SentimentIntensityAnalyzer() method. After that, let's list the polarity scores of the data and remove the symbols other than the delimiter.

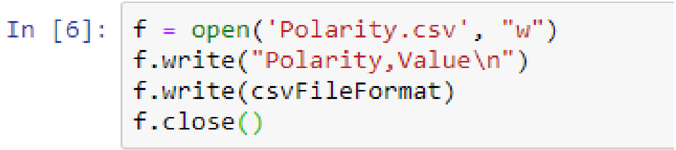


```
In [3]: analyzer = SentimentIntensityAnalyzer()
```



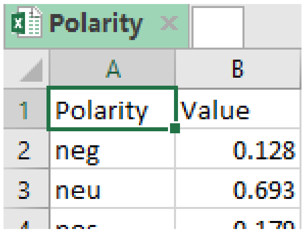
```
In [4]: countList = list(analyzer.polarity_scores((data)).items())
        vs = '\n'.join([str(i) for i in countList])

        csvFileFormat = re.sub("[)!@''#$(]", "", vs)
        print (csvFileFormat)

        neg, 0.128
        neu, 0.693
        pos, 0.179
        compound, 1.0
```

Export the result to csv file.



```
In [6]: f = open('Polarity.csv', "w")
        f.write("Polarity,Value\n")
        f.write(csvFileFormat)
        f.close()
```

Polarity.csv

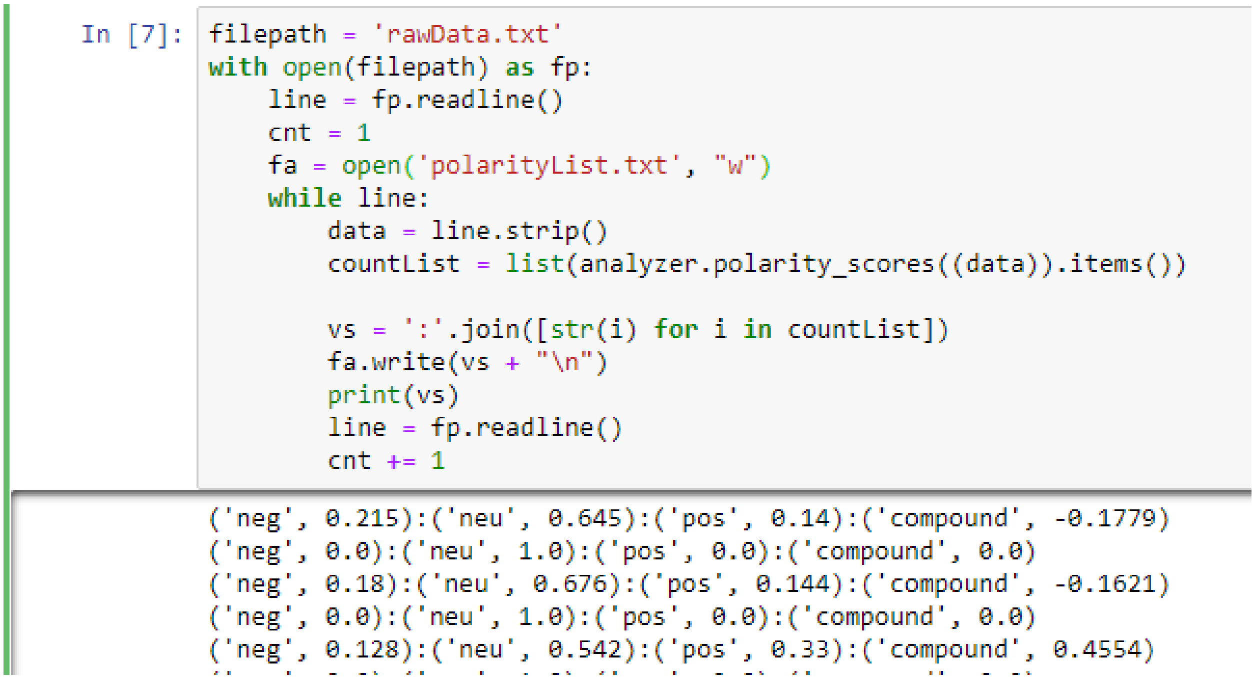


	A	B
1	Polarity	Value
2	neg	0.128
3	neu	0.693
4	pos	0.179

Now that we’ve have performed polarity for the whole file, let’s perform the same task for each line of the Raw Data. Let us first read the text file again and read each line of the data.

Put a looping to analyze each line to its polarity.

We’ll also export each line to text file polarityList.txt to save the results.

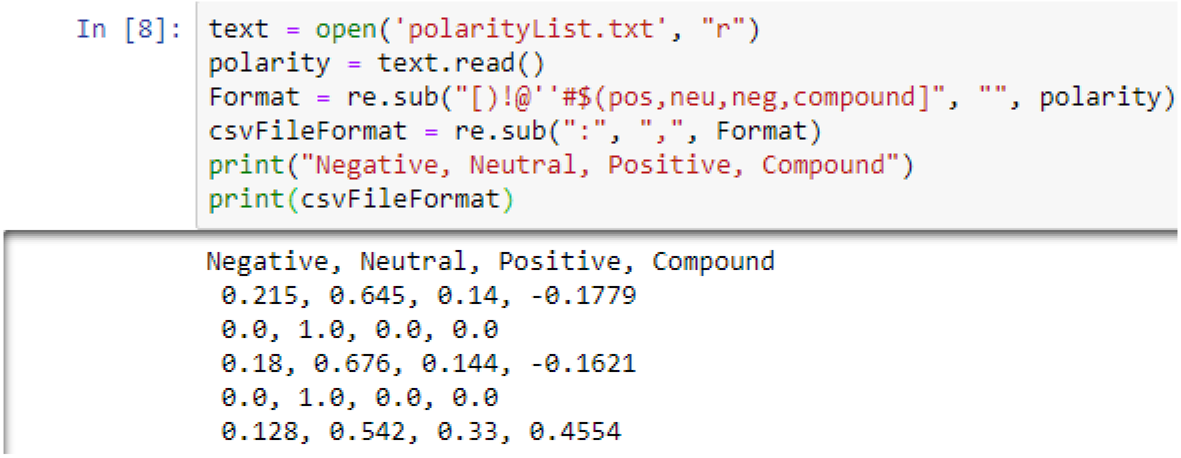


```
In [7]: filepath = 'rawData.txt'
        with open(filepath) as fp:
            line = fp.readline()
            cnt = 1
            fa = open('polarityList.txt', "w")
            while line:
                data = line.strip()
                countList = list(analyzer.polarity_scores((data)).items())

                vs = ':'.join([str(i) for i in countList])
                fa.write(vs + "\n")
                print(vs)
                line = fp.readline()
                cnt += 1
```

```
('neg', 0.215):('neu', 0.645):('pos', 0.14):('compound', -0.1779)
('neg', 0.0):('neu', 1.0):('pos', 0.0):('compound', 0.0)
('neg', 0.18):('neu', 0.676):('pos', 0.144):('compound', -0.1621)
('neg', 0.0):('neu', 1.0):('pos', 0.0):('compound', 0.0)
('neg', 0.128):('neu', 0.542):('pos', 0.33):('compound', 0.4554)
```

This may look very uncomfortable and inappropriate for the CSV format. Therefore, let’s transform this data for more reliable content before exporting to CSV file.

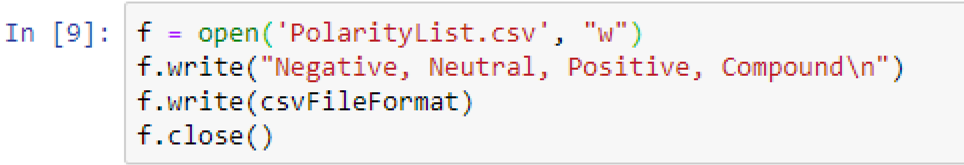
Remove the symbols also the repeated neg, neu, pos, and compound words then change the delimiter to proper delimiter for CSV Format.



```
In [8]: text = open('polarityList.txt', "r")
        polarity = text.read()
        Format = re.sub("[)!@''#$(pos,neu,neg,compound]", "", polarity)
        csvFileFormat = re.sub(":", ",", Format)
        print("Negative, Neutral, Positive, Compound")
        print(csvFileFormat)
```

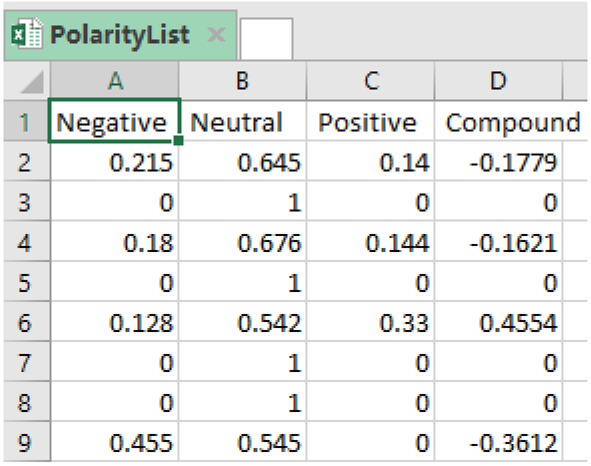
```
Negative, Neutral, Positive, Compound
 0.215, 0.645, 0.14, -0.1779
 0.0, 1.0, 0.0, 0.0
 0.18, 0.676, 0.144, -0.1621
 0.0, 1.0, 0.0, 0.0
 0.128, 0.542, 0.33, 0.4554
```

We can now export the result to CSV file.



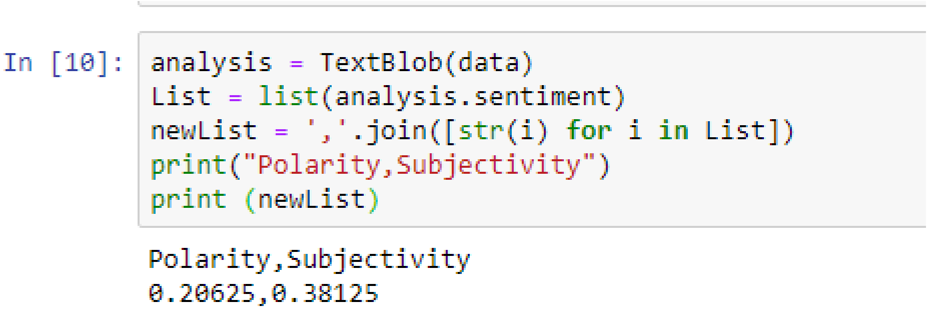
```
In [9]: f = open('PolarityList.csv', "w")
        f.write("Negative, Neutral, Positive, Compound\n")
        f.write(csvFileFormat)
        f.close()
```

PolarityList.csv file



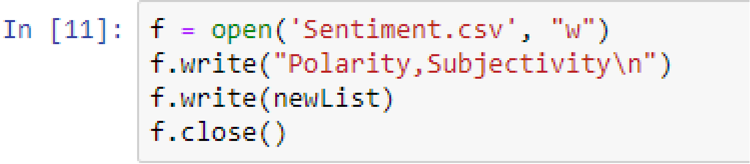
	A	B	C	D
1	Negative	Neutral	Positive	Compound
2	0.215	0.645	0.14	-0.1779
3	0	1	0	0
4	0.18	0.676	0.144	-0.1621
5	0	1	0	0
6	0.128	0.542	0.33	0.4554
7	0	1	0	0
8	0	1	0	0
9	0.455	0.545	0	-0.3612

Now let's perform an analysis using TextBlob for the sentiment. We'll execute the same process we've done earlier. Set *analysis = TextBlob(data)*



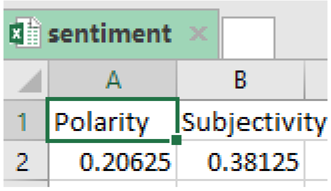
```
In [10]: analysis = TextBlob(data)
         List = list(analysis.sentiment)
         newList = ','.join([str(i) for i in List])
         print("Polarity,Subjectivity")
         print (newList)

         Polarity,Subjectivity
         0.20625,0.38125
```

The data for this one is processed into its polarity and subjectivity. Let's export this into CSV File.



```
In [11]: f = open('Sentiment.csv', "w")
         f.write("Polarity,Subjectivity\n")
         f.write(newList)
         f.close()
```

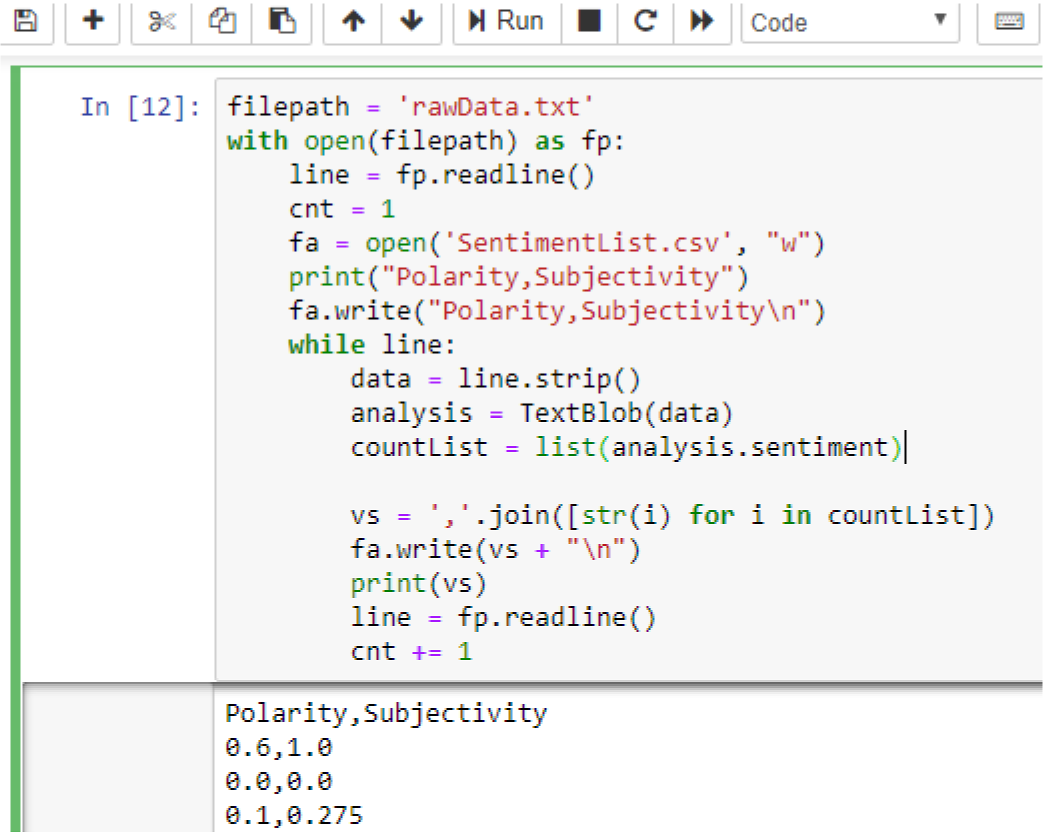
Sentiment.csv



	A	B
1	Polarity	Subjectivity
2	0.20625	0.38125

Next, we're going to analyze the sentiment of each line in the raw Data just like we've done earlier.

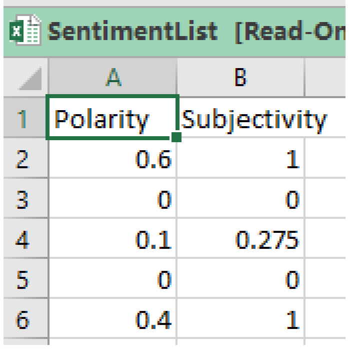
But this one will directly export into CSV file.



```
filepath = 'rawData.txt'
with open(filepath) as fp:
    line = fp.readline()
    cnt = 1
    fa = open('SentimentList.csv', "w")
    print("Polarity,Subjectivity")
    fa.write("Polarity,Subjectivity\n")
    while line:
        data = line.strip()
        analysis = TextBlob(data)
        countList = list(analysis.sentiment)

        vs = ','.join([str(i) for i in countList])
        fa.write(vs + "\n")
        print(vs)
        line = fp.readline()
        cnt += 1
```

```
Polarity,Subjectivity
0.6,1.0
0.0,0.0
0.1,0.275
```

SentimentList.csv



	A	B
1	Polarity	Subjectivity
2	0.6	1
3	0	0
4	0.1	0.275
5	0	0
6	0.4	1