08_Sentiment Analysis

May 7, 2022

From a business standpoint, it is very important to understand how customer feedback is on the products/services they offer to improvise on the products/service for customer satisfaction.

We have a dataset for Amazon food reviews. Let's use that data and extract insight out of it. You can download the data from www.kaggle.com/snap/amazon-fine-food-reviews.

[1]:
```
# Import necessary libraries
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

#Read the data
df = pd.read_csv('data/Reviews.csv', nrows=100)

# Look at the top 5 rows of the data
df.head(5)
```

[1]:

	Id	ProductId	UserId	ProfileName	\
0	1	B001E4KFG0	A3SGXH7AUHU8GW	delmartian	
1	2	B00813GRG4	A1D87F6ZCVE5NK	dll pa	
2	3	B000LQOCH0	ABXLMWJIXXAIN	Natalia Corres "Natalia Corres"	
3	4	B000UA0QIQ	A395BORC6FGVXV	Karl	
4	5	B006K2ZZ7K	A1UQRSCLF8GW1T	Michael D. Bigham "M. Wassir"	

	HelpfulnessNumerator	HelpfulnessDenominator	Score	Time	\
0	1	1	5	1303862400	
1	0	0	1	1346976000	
2	1	1	4	1219017600	
3	3	3	2	1307923200	
4	0	0	5	1350777600	

	Summary	Text
0	Good Quality Dog Food	I have bought several of the Vitality canned d…
1	Not as Advertised	Product arrived labeled as Jumbo Salted Peanut…
2	"Delight" says it all	This is a confection that has been around a fe…
3	Cough Medicine	If you are looking for the secret ingredient i…
4	Great taffy	Great taffy at a great price. There was a wid…

[2]:
```
# Understand the data types of the columns
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 100 entries, 0 to 99
Data columns (total 10 columns):
 #   Column                  Non-Null Count  Dtype
---  ------                  --------------  -----
 0   Id                      100 non-null    int64
 1   ProductId               100 non-null    object
 2   UserId                  100 non-null    object
 3   ProfileName             100 non-null    object
 4   HelpfulnessNumerator    100 non-null    int64
 5   HelpfulnessDenominator  100 non-null    int64
 6   Score                   100 non-null    int64
 7   Time                    100 non-null    int64
 8   Summary                 100 non-null    object
 9   Text                    100 non-null    object
dtypes: int64(5), object(5)
memory usage: 7.9+ KB
```

[3]:
```
# Looking at the summary of the reviews.
df.Summary.head()
```

[3]:
```
0    Good Quality Dog Food
1        Not as Advertised
2    "Delight" says it all
3           Cough Medicine
4              Great taffy
Name: Summary, dtype: object
```

[4]:
```
# Looking at the description of the reviews
df.Text.head()
```

[4]:
```
0    I have bought several of the Vitality canned d…
1    Product arrived labeled as Jumbo Salted Peanut…
2    This is a confection that has been around a fe…
3    If you are looking for the secret ingredient i…
4    Great taffy at a great price.  There was a wid…
Name: Text, dtype: object
```

[5]:
```
# Import libraries
from nltk.corpus import stopwords
from textblob import TextBlob
from textblob import Word

# Lower casing and removing punctuations
```

```
df['Text'] = df['Text'].apply(lambda x: " ".join(x.lower() for x in x.split()))
df['Text'] = df['Text'].str.replace('[^\w\s]','')

# Removal of stop words
stop = stopwords.words('english')
df['Text'] = df['Text'].apply(lambda x: " ".join(x for x in x.split() if x not␣
 ↪in stop))

# Spelling correction
df['Text'] = df['Text'].apply(lambda x: str(TextBlob(x).correct()))

# Lemmatization
df['Text'] = df['Text'].apply(lambda x: " ".join([Word(word).lemmatize() for␣
 ↪word in x.split()]))

df.Text.head(5)
```

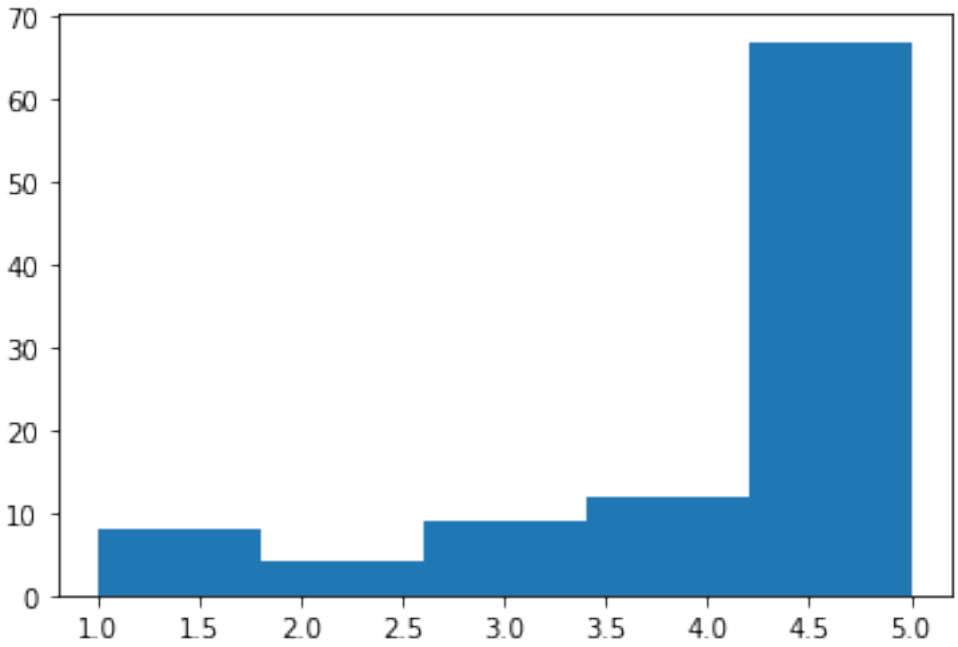
```
C:\Users\ADMINI~1\AppData\Local\Temp/ipykernel_20160/2814826275.py:8:
FutureWarning: The default value of regex will change from True to False in a
future version.
  df['Text'] = df['Text'].str.replace('[^\w\s]','')
```

[5]:
```
0    bought several vitality canned dog food produc…
1    product arrived labelled lumbo halted peanutst…
2    connection around century light pillow city ge…
3    looking secret ingredient robitussin believe f…
4    great staff great price wide assortment mummy …
Name: Text, dtype: object
```

[6]:
```
# Create a new data frame "reviews" to perform exploratory data analysis upon␣
 ↪that
reviews = df

# Dropping null values
reviews.dropna(inplace=True)

# The histogram reveals this dataset is highly unbalanced towards high rating.
reviews.Score.hist(bins=5,grid=False)
plt.show()
print(reviews.groupby('Score').count().Id)
```




```
Score
1     8
2     4
3     9
4    12
5    67
Name: Id, dtype: int64
```

[7]:
```
# To make it balanced data, we sampled each score by the lowest n-count from␣
 ↪above. (i.e. 29743 reviews scored as '2')

score_1 = reviews[reviews['Score'] == 1].sample(n=4)
score_2 = reviews[reviews['Score'] == 2].sample(n=4)
score_3 = reviews[reviews['Score'] == 3].sample(n=4)
score_4 = reviews[reviews['Score'] == 4].sample(n=4)
score_5 = reviews[reviews['Score'] == 5].sample(n=4)

# Here we recreate a 'balanced' dataset.
reviews_sample = pd.concat([score_1,score_2,score_3,score_4,score_5],axis=0)
reviews_sample.reset_index(drop=True,inplace=True)

# Printing count by 'Score' to check dataset is now balanced.
print(reviews_sample.groupby('Score').count().Id)
```

```
Score
1    4
```

```
2    4
3    4
4    4
5    4
Name: Id, dtype: int64
```

[8]:
```
# Let's build a word cloud looking at the 'Summary'  text
from wordcloud import WordCloud

# Wordcloud function's input needs to be a single string of text.
# Here I'm concatenating all Summaries into a single string.
# similarly you can build for Text column

reviews_str = reviews_sample.Summary.str.cat()

wordcloud = WordCloud(background_color='white').generate(reviews_str)
plt.figure(figsize=(10,10))
plt.imshow(wordcloud,interpolation='bilinear')
plt.axis("off")
plt.show()
```

[9]:
```
# Now let's split the data into Negative (Score is 1 or 2) and Positive (4 or␣
↪#5) Reviews.
negative_reviews = reviews_sample[reviews_sample['Score'].isin([1,2]) ]
positive_reviews = reviews_sample[reviews_sample['Score'].isin([4,5]) ]

# Transform to single string
negative_reviews_str = negative_reviews.Summary.str.cat()
```

```
positive_reviews_str = positive_reviews.Summary.str.cat()
```

[10]:
```
# Create wordclouds
wordcloud_negative = WordCloud(background_color='white').
 ↪generate(negative_reviews_str)
wordcloud_positive = WordCloud(background_color='white').
 ↪generate(positive_reviews_str)

# Plot
fig = plt.figure(figsize=(10,10))
ax1 = fig.add_subplot(211)
ax1.imshow(wordcloud_negative,interpolation='bilinear')
ax1.axis("off")
ax1.set_title('Reviews with Negative Scores',fontsize=20)
ax2 = fig.add_subplot(212)
ax2.imshow(wordcloud_positive,interpolation='bilinear')
ax2.axis("off")
ax2.set_title('Reviews with Positive Scores',fontsize=20)
plt.show()
```

VADER Sentiment Analysis :

VADER (Valence Aware Dictionary and sEntiment Reasoner) is a lexicon and rule-based sentiment analysis tool that is specifically attuned to sentiments expressed in social media. VADER uses a combination of A sentiment lexicon is a list of lexical features (e.g., words) which are generally labeled according to their semantic orientation as either positive or negative. VADER not only tells about the Positivity and Negativity score but also tells us about how positive or negative a sentiment is.

[11]:
```
!pip install vaderSentiment
```

```
Requirement already satisfied: vaderSentiment in
c:\users\administrator\anaconda3\lib\site-packages (3.3.2)
Requirement already satisfied: requests in
c:\users\administrator\anaconda3\lib\site-packages (from vaderSentiment)
(2.26.0)
Requirement already satisfied: certifi>=2017.4.17 in
c:\users\administrator\anaconda3\lib\site-packages (from
requests->vaderSentiment) (2021.10.8)
Requirement already satisfied: idna<4,>=2.5 in
c:\users\administrator\anaconda3\lib\site-packages (from
requests->vaderSentiment) (3.2)
Requirement already satisfied: urllib3<1.27,>=1.21.1 in
c:\users\administrator\anaconda3\lib\site-packages (from
requests->vaderSentiment) (1.26.7)
Requirement already satisfied: charset-normalizer~=2.0.0 in
c:\users\administrator\anaconda3\lib\site-packages (from
requests->vaderSentiment) (2.0.4)
```

[12]:
```
import seaborn as sns
plt.style.use('fivethirtyeight')
# Function for getting the sentiment
cp = sns.color_palette()
from vaderSentiment.vaderSentiment import SentimentIntensityAnalyzer
analyzer = SentimentIntensityAnalyzer()

# Generating sentiment for all the sentence present in the dataset
emptyline=[]
for row in df['Text']:
    vs=analyzer.polarity_scores(row)
    emptyline.append(vs)
```

[13]:
```
# Creating new dataframe with sentiments
df_sentiments=pd.DataFrame(emptyline)
df_sentiments.head(5)
```

[13]:

	neg	neu	pos	compound
0	0.000	0.503	0.497	0.9413
1	0.258	0.644	0.099	-0.5719
2	0.134	0.602	0.264	0.7880
3	0.000	0.854	0.146	0.4404
4	0.000	0.455	0.545	0.9186

compound score is the metric that calculates the sentiment score(-1 to 1):

- positive: >0.05
- neutral: > -0.05 and <0.05

- negative: <= -0.05

[14]:
```
# Merging the sentiments back to reviews dataframe
df_c = pd.concat([df.reset_index(drop=True), df_sentiments], axis=1)
df_c.head(3)
```

[14]:
```
   Id   ProductId          UserId                      ProfileName  \
0   1  B001E4KFG0  A3SGXH7AUHU8GW                       delmartian
1   2  B00813GRG4  A1D87F6ZCVE5NK                           dll pa
2   3  B000LQOCH0   ABXLMWJIXXAIN  Natalia Corres "Natalia Corres"

   HelpfulnessNumerator  HelpfulnessDenominator  Score        Time  \
0                     1                       1      5  1303862400
1                     0                       0      1  1346976000
2                     1                       1      4  1219017600

                Summary                                                Text  \
0  Good Quality Dog Food  bought several vitality canned dog food produc…
1      Not as Advertised  product arrived labelled lumbo halted peanutst…
2  "Delight" says it all  connection around century light pillow city ge…

     neg    neu    pos  compound
0  0.000  0.503  0.497    0.9413
1  0.258  0.644  0.099   -0.5719
2  0.134  0.602  0.264    0.7880
```

[15]:
```
# Convert scores into positive and negetive sentiments using some threshold
df_c['Sentiment'] = np.where(df_c['compound'] >= 0 , 'Positive', 'Negative')
df_c.head(3)
```

[15]:
```
   Id   ProductId          UserId                      ProfileName  \
0   1  B001E4KFG0  A3SGXH7AUHU8GW                       delmartian
1   2  B00813GRG4  A1D87F6ZCVE5NK                           dll pa
2   3  B000LQOCH0   ABXLMWJIXXAIN  Natalia Corres "Natalia Corres"

   HelpfulnessNumerator  HelpfulnessDenominator  Score        Time  \
0                     1                       1      5  1303862400
1                     0                       0      1  1346976000
2                     1                       1      4  1219017600

                Summary                                                Text  \
0  Good Quality Dog Food  bought several vitality canned dog food produc…
1      Not as Advertised  product arrived labelled lumbo halted peanutst…
2  "Delight" says it all  connection around century light pillow city ge…

     neg    neu    pos  compound Sentiment
0  0.000  0.503  0.497    0.9413  Positive
```

```
1  0.258  0.644  0.099   -0.5719  Negative
2  0.134  0.602  0.264    0.7880  Positive
```

[16]:
```
result=df_c['Sentiment'].value_counts()
print(result)
result.plot(kind='bar', rot=45)
```

```
Positive    91
Negative     9
Name: Sentiment, dtype: int64
```

[16]: <AxesSubplot:>