PROJECT 22:

batchnumber 12

TITLE: OPINION MINING FOR HOTEL RATING THROUGH REVIEWS

> #Abstract: In this project, there will be a web application where user will provide review about hotels. This review will be in sentence form. System will extract certain keywords from the sentence and will minekeywords in database and system will rate the hotels based on the reviews of various users. We presented machine learning and Sentiment Word Net based method for opinion mining from hotel reviews and sentence relevance score based method for opinion summarization of hotel reviews. . The classified and summarized hotel review information helps web users to understand review contents easily in a short time.Opinion Mining for Hotel Review system that detects hidden sentiments in feedback of the customer andrates the feedback accordingly. The system uses opinion mining methodology in order to achieve desired functionality. Opinion mining for hotel reviews is a web application which gives review of the feedback thatis posted by various users. The System takes review of various users, based on the opinion, system willspecify whether the posted hotel is good, bad, or worst. We use a database of sentiment based keywords along with positivity or negativity weight in database and then based on these sentiment keywords mined inuser review is ranked. Using this application User will get to know which hotel is best and suitable for them.User can decide which hotel to accommodate before they reach the place

```
#dataset
```

+ Code + Text

ntlk-natural language tool kit . nltk is a platform used for building Python programs that work with human language data

```
import nltk
nltk.download('vader_lexicon')
```

```
[nltk_data] Downloading package vader_lexicon to /root/nltk_data...
True
```

using ntlk sentimant vader we are importing sentiment intensity analyzer to classify various text into positive and negative categories

```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from nltk.sentiment.vader import SentimentIntensityAnalyzer
sentiments = SentimentIntensityAnalyzer()

data = pd.read_csv("Hotel_Reviews.csv")
print(data.head())
```

```
                                       Hotel_Address  ...       lng
0   s Gravesandestraat 55 Oost 1092 AA Amsterdam ...  ...  4.915968
1   s Gravesandestraat 55 Oost 1092 AA Amsterdam ...  ...  4.915968
2   s Gravesandestraat 55 Oost 1092 AA Amsterdam ...  ...  4.915968
3   s Gravesandestraat 55 Oost 1092 AA Amsterdam ...  ...  4.915968
4   s Gravesandestraat 55 Oost 1092 AA Amsterdam ...  ...  4.915968

[5 rows x 17 columns]
/usr/local/lib/python3.7/dist-packages/nltk/twitter/__init__.py:20: UserWarning:
  warnings.warn("The twython library has not been installed. "
```

```
##NTLK-natural language tool kit-python module
```

```
data
```

	Hotel_Address	Additional_Number_of_Scoring	Review_Date	Average_Scor
0	s Gravesandestraat 55 Oost 1092 AA Amsterdam ...	194	8/3/2017	7
1	s Gravesandestraat 55 Oost 1092 AA Amsterdam ...	194	8/3/2017	7
2	s Gravesandestraat 55 Oost 1092 AA Amsterdam ...	194	7/31/2017	7
3	s Gravesandestraat 55 Oost 1092 AA Amsterdam ...	194	7/31/2017	7
4	s Gravesandestraat 55 Oost 1092 AA Amsterdam ...	194	7/24/2017	7
...	...	...	...	
2136	1 3 Queens Garden Westminster Borough London W...	1058	3/11/2017	7
2137	1 3 Queens Garden Westminster Borough London W...	1058	3/10/2017	7

```
data.dropna()
```

	Hotel_Address	Additional_Number_of_Scoring	Review_Date	Average_Scor
0	s Gravesandestraat 55 Oost 1092 AA Amsterdam ...	194	8/3/2017	7
1	s Gravesandestraat 55 Oost 1092 AA Amsterdam ...	194	8/3/2017	7
2	s Gravesandestraat 55 Oost 1092 AA Amsterdam ...	194	7/31/2017	7
3	s Gravesandestraat 55 Oost 1092 AA Amsterdam ...	194	7/31/2017	7
4	s Gravesandestraat 55 Oost 1092 AA Amsterdam ...	194	7/24/2017	7
...	...	...	...	
2135	1 3 Queens Garden Westminster Borough London W...	1058	3/12/2017	7
2136	1 3 Queens Garden Westminster Borough London W...	1058	3/11/2017	7
	1 3 Queens			

```
data.shape
```

```
(2141, 17)
```

```
#checking weather there are any null values present in the dataset
```

```
data.isnull().sum()
```

```
Hotel_Address                                 0
Additional_Number_of_Scoring                  0
Review_Date                                   0
Average_Score                                 0
Hotel_Name                                    0
Reviewer_Nationality                          0
Negative_Review                               0
Review_Total_Negative_Word_Counts             1
Total_Number_of_Reviews                       1
Positive_Review                               1
Review_Total_Positive_Word_Counts             1
Total_Number_of_Reviews_Reviewer_Has_Given    1
Reviewer_Score                                1
Tags                                          1
days_since_review                             1
lat                                           1
lng                                           1
dtype: int64
```

```
data['Negative_Review'].describe()
```

```
count            2141
unique           1397
top       No Negative
freq              611
Name: Negative_Review, dtype: object
```

```
data['Positive_Review'].describe()
```

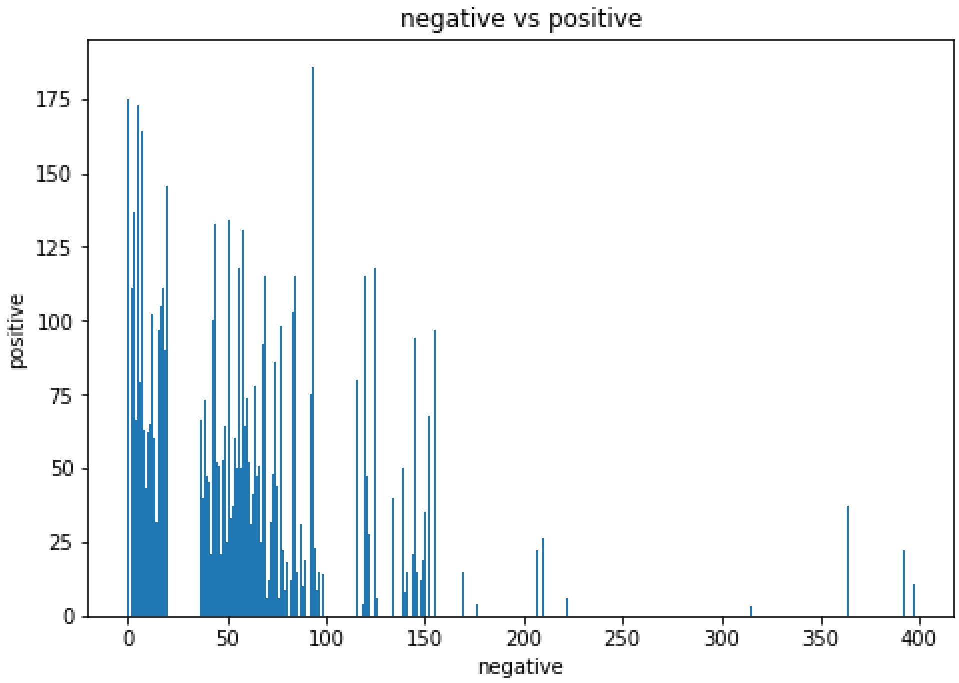
```
count            2140
```

```
unique           2005
top       No Positive
freq              103
Name: Positive_Review, dtype: object
```

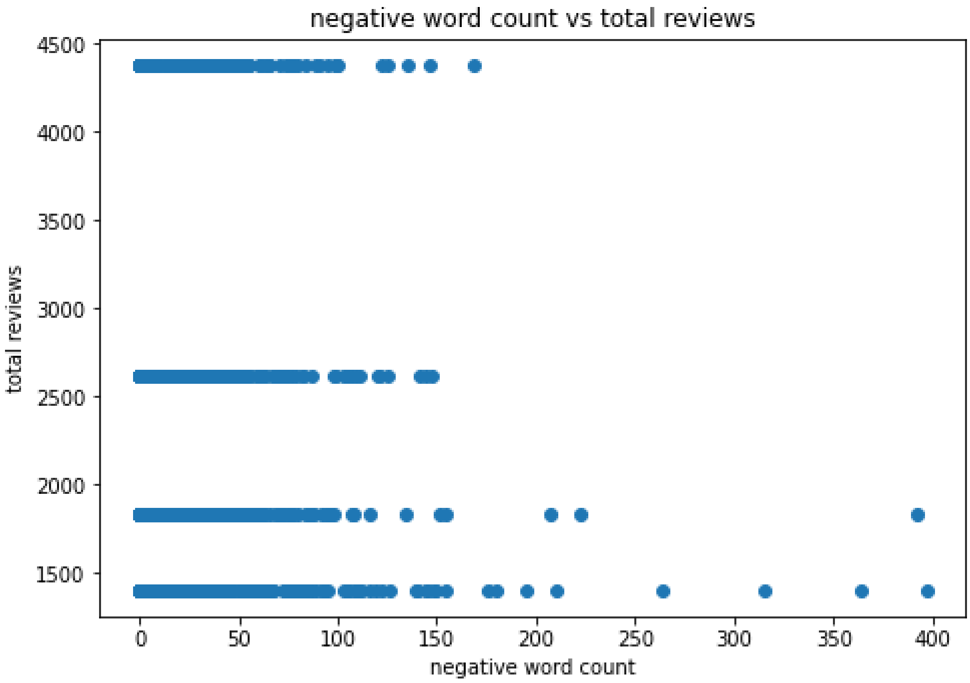
```
data.describe()
```

	Additional_Number_of_Scoring	Average_Score	Review_Total_Negative_Word_Co
count	2141.000000	2141.000000	2140.00
mean	476.427370	8.638020	18.80
std	251.419854	0.615859	31.07
min	194.000000	7.700000	0.00
25%	244.000000	7.700000	0.00
50%	639.000000	8.500000	8.00
75%	639.000000	9.200000	24.00
max	1058.000000	9.200000	397.00

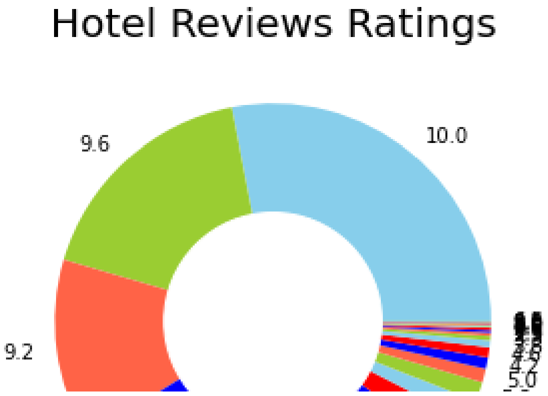
```
import matplotlib.pyplot as plt
fig=plt.figure()
ax=fig.add_axes([0,0,1,1])
attacks=data['Review_Total_Negative_Word_Counts']
defense = data['Review_Total_Positive_Word_Counts']
ax.bar(attacks,defense)
ax.set_xlabel('negative')
ax.set_ylabel('positive')
ax.set_title('negative vs positive')
plt.show()
```



```
import matplotlib.pyplot as plt
fig=plt.figure()
ax=fig.add_axes([0,0,1,1])
attacks=data['Review_Total_Negative_Word_Counts']
defense = data['Total_Number_of_Reviews']
ax.scatter(attacks,defense)
ax.set_xlabel('negative word count')
ax.set_ylabel('total reviews')
ax.set_title('negative word count vs total reviews')
plt.show()
```



```
#visualizing hotel review rating score


ratings = data["Reviewer_Score"].value_counts()
numbers = ratings.index
quantity = ratings.values

custom_colors = ["skyblue", "yellowgreen", 'tomato', "blue", "red"]
plt.figure(figsize=(5, 5))
plt.pie(quantity, labels=numbers, colors=custom_colors)
central_circle = plt.Circle((0, 0), 0.5, color='white')
fig = plt.gcf()
fig.gca().add_artist(central_circle)
plt.rc('font', size=12)
plt.title("Hotel Reviews Ratings", fontsize=20)
plt.show()
```



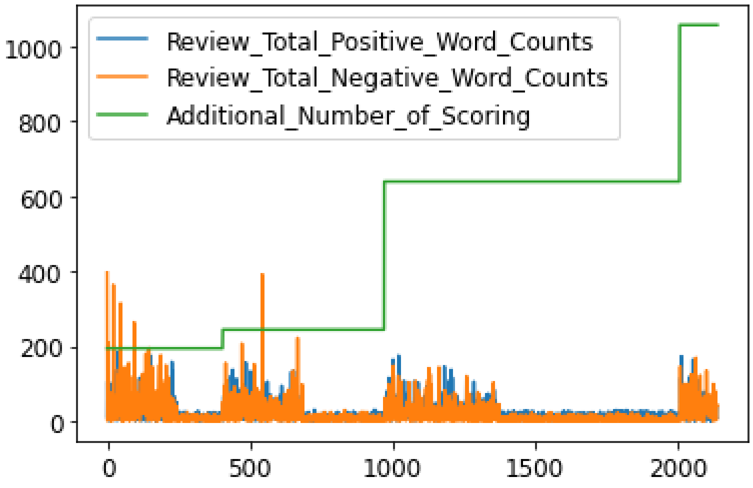
```
#highest rating btw postive rating , negative rating and neutral
```

```
x = sum(data["Review_Total_Positive_Word_Counts"])
y = sum(data["Review_Total_Negative_Word_Counts"])
z = sum(data["Additional_Number_of_Scoring"])

def sentiment_score(a, b, c):
    if (a>b) and (a>c):
        print("Positive 😊 ")
    elif (b>a) and (b>c):
        print("Negative 😠 ")
    else:
        print("Neutral 🙂 ")
sentiment_score(x, y, z)
```

```
Neutral 🙂
```

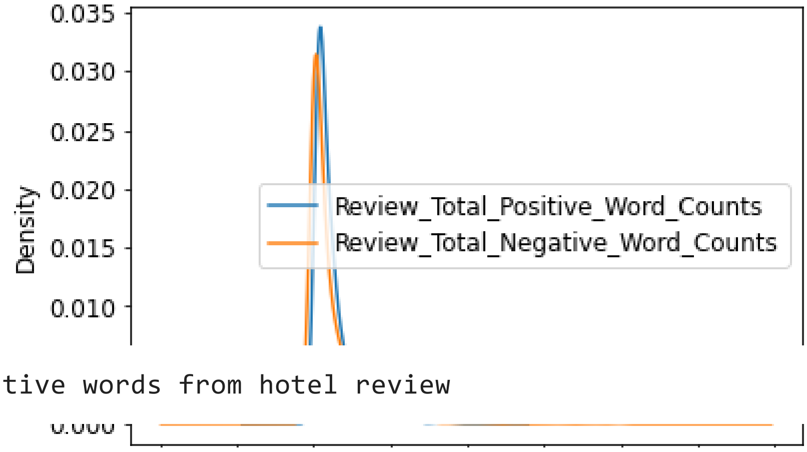
```
dataset2 = data[["Review_Total_Positive_Word_Counts","Review_Total_Negative_Word_Counts
dataset2.plot()
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f1da01b11d0>
```



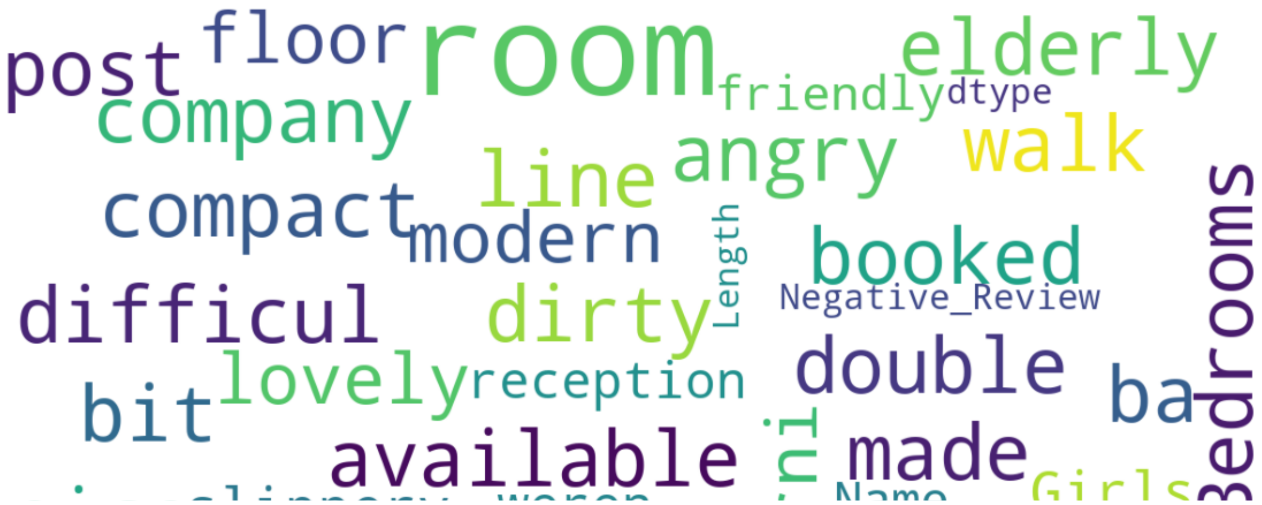
```
data[["Review_Total_Positive_Word_Counts", "Review_Total_Negative_Word_Counts"]].plot.k
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f1da01a0d10>
```



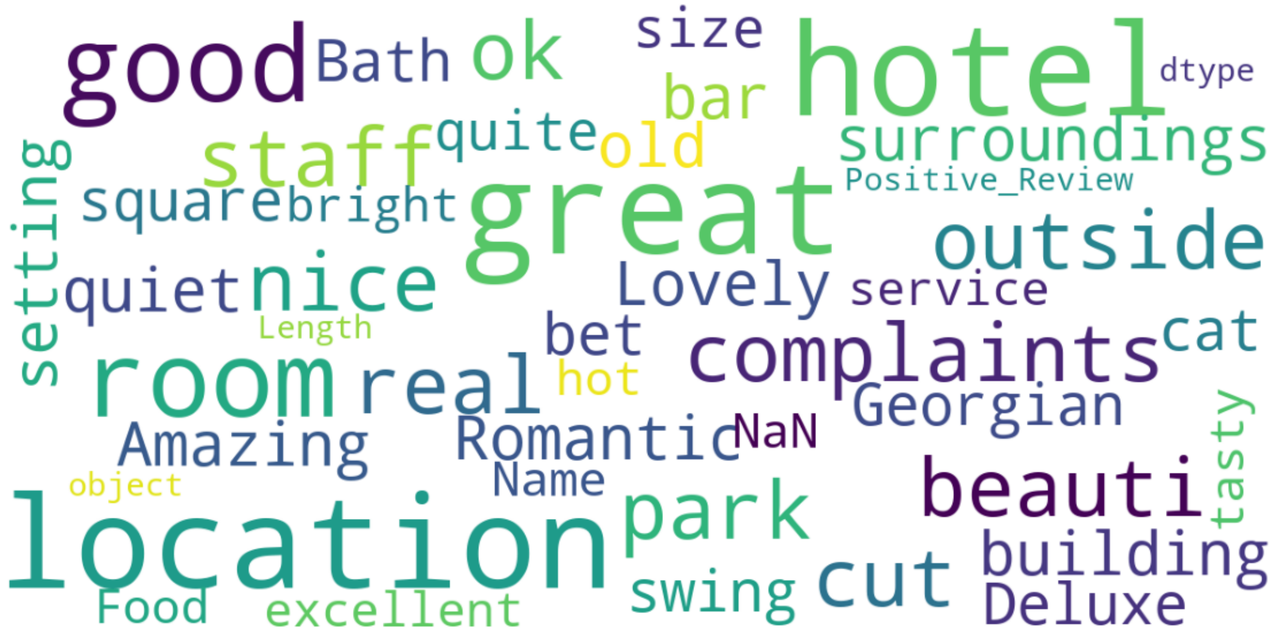
```
#negative words from hotel review
```

```
from wordcloud import WordCloud
import matplotlib.pyplot as plt

def show_wordcloud(data, title = None):
    wordcloud = WordCloud(
        background_color = 'white',
        max_words = 200,
        max_font_size = 40,
        scale = 3,
        random_state = 42
    ).generate(str(data))

    fig = plt.figure(1, figsize = (20, 20))
    plt.axis('off')
    if title:
        fig.suptitle(title, fontsize = 20)
        fig.subplots_adjust(top = 2.3)

    plt.imshow(wordcloud)
    plt.show()

# print wordcloud
show_wordcloud(data["Negative_Review"])
```



```
#positive words from hotel review
```

```
show_wordcloud(data["Positive_Review"])
```

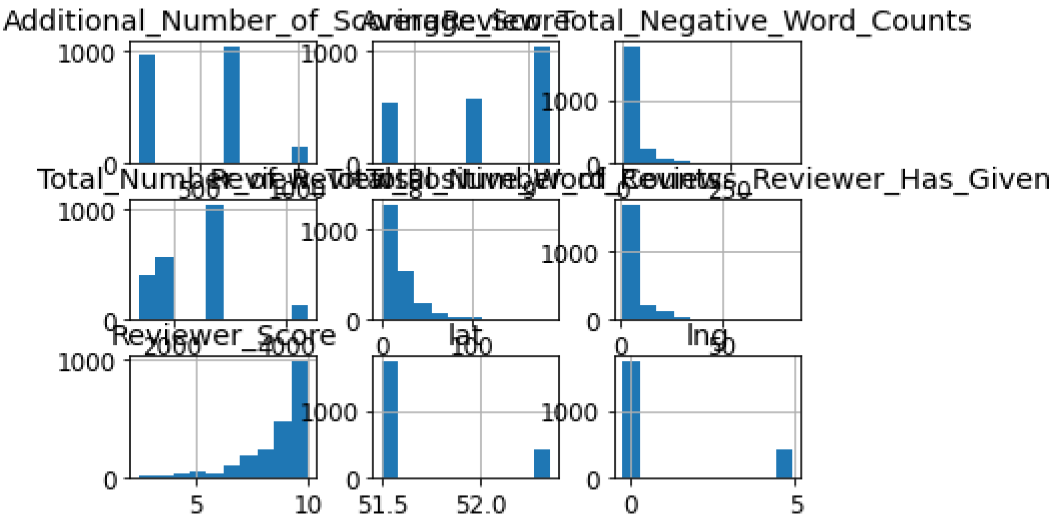


```
data['Review_Total_Negative_Word_Counts'].mean()
```

```
    18.801869158878503
```

```
data['Review_Total_Positive_Word_Counts'].mean()

     21.298130841121495
```

```
import matplotlib.pyplot as plt
data.hist()
plt.show()
```



```
#correlation btw negative word cound and positive word count
```

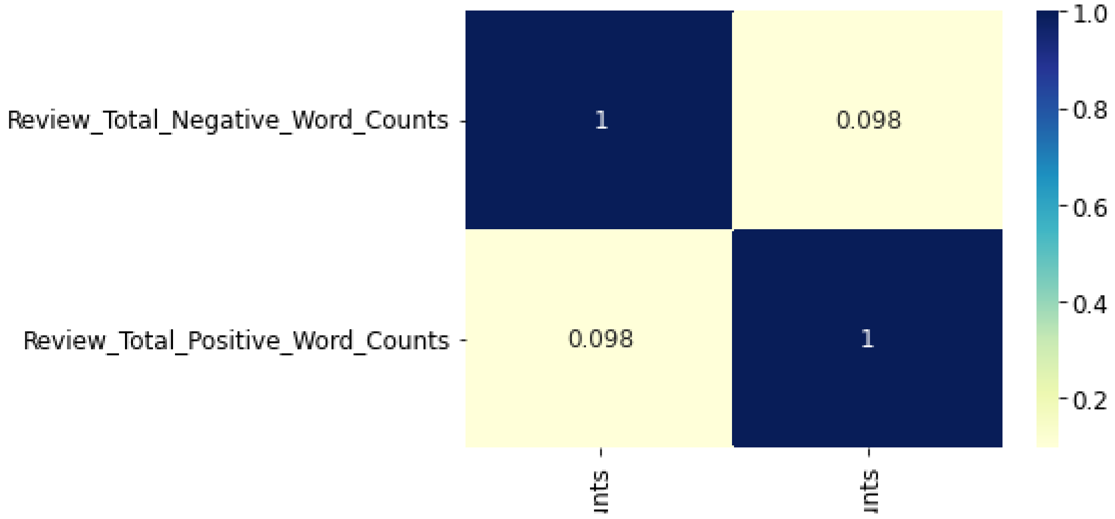
```
correlation = data[['Review_Total_Negative_Word_Counts','Review_Total_Positive_Word_Cou
```

```
correlation
```

	Review_Total_Negative_Word_Counts	Review_Tota
Review_Total_Negative_Word_Counts	1.000000	
Review_Total_Positive_Word_Counts	0.098358	

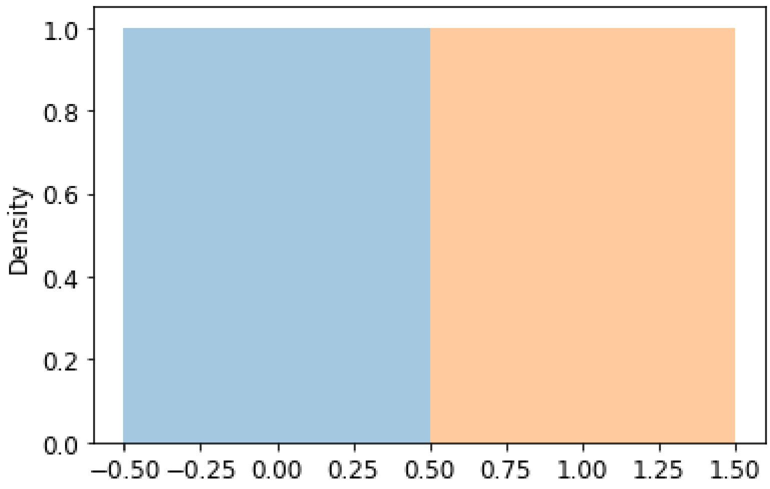
```
#visualizing correlation btw positive and negative word count using heat map
```

```
import seaborn as sb
dp = sb.heatmap(correlation, cmap="YlGnBu", annot=True)
```



```
import seaborn as sns

for x in [0, 1]:
    subset = data[data['Positive_Review'] == x]

    # Draw the density plot
    if x == 0:
        l = "Good reviews"
    else:
        l = "Bad reviews"
    sns.distplot(x,label=l)
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/distributions.py:2619: FutureWarni
  warnings.warn(msg, FutureWarning)
/usr/local/lib/python3.7/dist-packages/seaborn/distributions.py:316: UserWarning:
  warnings.warn(msg, UserWarning)
/usr/local/lib/python3.7/dist-packages/seaborn/distributions.py:2619: FutureWarni
  warnings.warn(msg, FutureWarning)
/usr/local/lib/python3.7/dist-packages/seaborn/distributions.py:316: UserWarning:
  warnings.warn(msg, UserWarning)
```



New Section

[ ] ↳ *4 cells hidden*