NLP Report

Topic: Amazon Review Data of Health and Personal Care Products (http://jmcauley.ucsd.edu/data/amazon/)

Project Aim: Perform some sentiment analysis based on concepts covered in the course. Analyze and visualize interesting data insights.

Data Description: The data is in JSON format. Sample Json object is as follows,

{"reviewerID": "ALC5GH8CAMAI7",
"asin": "159985130X",
"reviewerName": "AnnN",
"helpful": [1, 1], "reviewText": "This is a great little gadget to have around. We've already used it to look for splinters and a few other uses. The light is great. It's a handy size. However, I do wish I'd bought one with a little higher magnification.",
"overall": 5.0,
"summary": "Handy little gadget",
"unixReviewTime": 1294185600,
"reviewTime": "01 5, 2011"}

Asin is the product ID, helpful have two values enclosed in a list form. It contains number of people who found this useful (1) or not useful (0).

Execution of the project Idea: This can be found in NLP_Project.ipynb file.

The steps used for execution of project are as follows:

- Import Dependencies
- Loading json file as pandas dataframe
- Data Analysis and Visualization
- Sentiment Analysis of words using sentinet.
    - Removing Stopwords and Punctuations.
    - Tokenization and pos tagging
    - Lemmatization of pos tagged tokens
    - Calculation of polarity of tokens

Analysis and Visualization of Data:

1. To check number of null values, present in data:

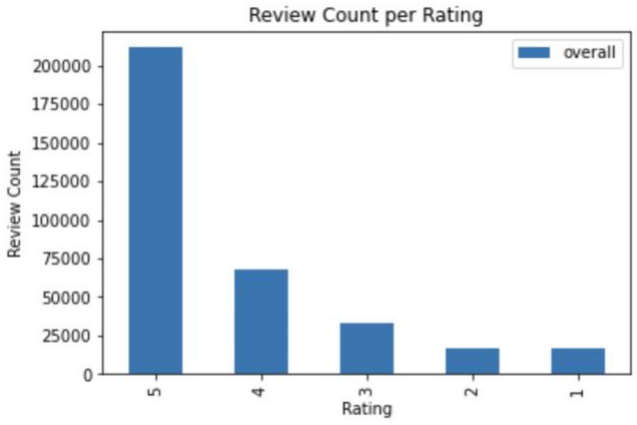
```
In [39]: df.isnull().sum()

Out[39]: reviewerID           0
         asin                 0
         reviewerName      3051
         helpful              0
         reviewText           0
         overall              0
         summary              0
         unixReviewTime       0
         reviewTime           0
         length_review        0
         Date                 0
         dtype: int64
```

2. Count and plot number of reviews for ratings ranging from 1 to 5.

```
In [4]: #count no. of reviews based on overall rating

        n1=pd.DataFrame(df['overall'].value_counts())
        n1
```

Out[4]:

	overall
5	211633
4	68168
3	33254
2	16754
1	16546



Products with rating as 5 had highest number of reviews. This can explain inclination towards positive sentiment score.

3. Top10 products with highest rating:

```
In [13]: # products with highest rating (Reference: https://iu.instructure.com/courses/2058564
cal2=df.iloc[:,[1,5]]

n2=pd.DataFrame(cal2.groupby('asin')['overall'].mean())

high_rate1 = n2.sort_values(by=['overall'], ascending=False).nlargest(10, 'overall')
high_rate1
```

Out[13]:

asin	overall
B000I4AIUA	5.0
B005XIDPFQ	5.0
B00FH0WUQ0	5.0
B00FH1CO9M	5.0
B000FP04CO	5.0
B001O5T4LQ	5.0
B005XQWVI0	5.0
B005XJCUSI	5.0
B005XCX84K	5.0
B0016P1VT2	5.0

4. Classifying reviews as positive and negative based on rating

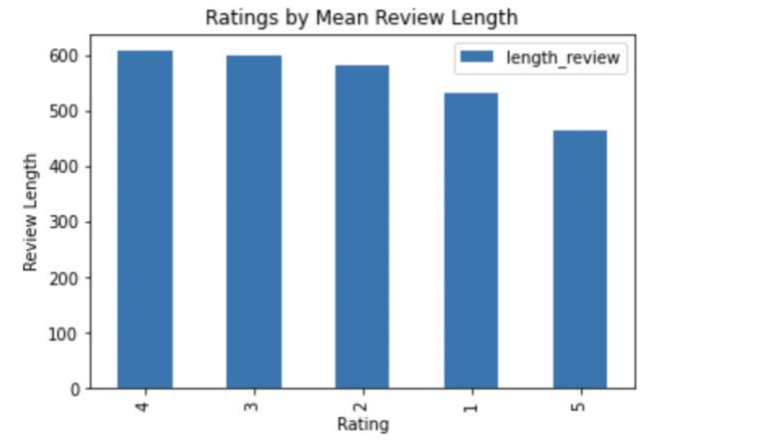
```
In [18]: print("Total positive reviews are:",len(positive))
Out[18]: 279801

In [21]: print("Total negative reviews are:",len(negative))
         Total negative reviews are: 66554
```

More than 80% of the reviews present in dataset are positive. These reviews are classified as positive and negative based on the rating. For products with rating more than and equal to 3.5 are classified as positive whereas the latter is considered as negative.

5. Mean review length for every rating:

Out[30]:

overall	length_review
4	607.854976
3	599.108919
2	583.640444
1	533.546597
5	464.339295



There is not much difference in word count of review of products with highest and lowest rating i.e., 1 and 5. This can help explain extreme satisfaction and dissatisfaction with the product through analyzing sentiments.

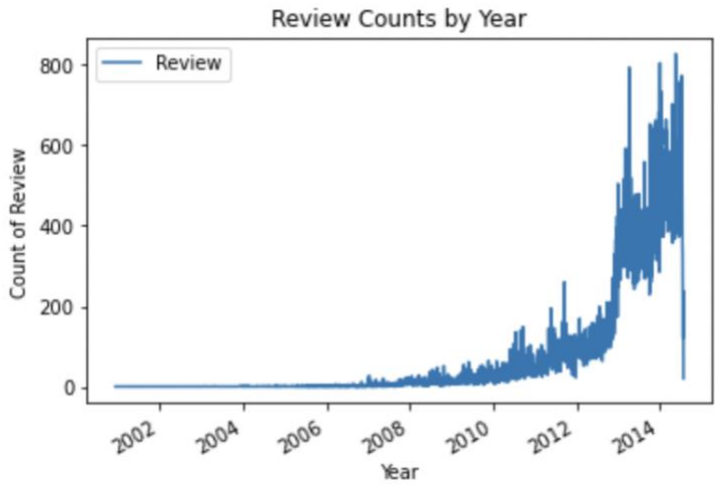
6. Products with highest purchase rate:

Out[7]:

	asin
B0037KMI0U	1089
B0010JLMO8	767
B001KXZ808	699
B0049LUI9O	528
B000GIPJY8	475
B001F51VRK	469
B000NL0T1G	434
B001F51VS4	431
B004YHKUXC	427
B007UZNS5W	422

Product ID along with the count showing its purchase rate is shown above.

7. Review Counts by Year



Reviewing of products tremendously increased between mid of 2012 to late 2014.

8. Result of Sentiment Analysis by calculation of polarity of words in reviews using Sentinet.

```
In [45]: #wordnet and lemma part referred: https://www.guru99.com/stemming-lemmatization-python-nltk.html and Assignment 4

from nltk.corpus import wordnet
from nltk.corpus import sentiwordnet as swn
from nltk.stem import WordNetLemmatizer
def conv(tag):
    if tag.startswith('J'):
        return wordnet.ADJ
    elif tag.startswith('N'):
        return wordnet.NOUN
    elif tag.startswith('R'):
        return wordnet.ADV
    elif tag.startswith('V'):
        return wordnet.VERB
    return None
score = 0.0
lemmatizer = WordNetLemmatizer()
for i in list_pos_tokens:
    for word, tag in i:
        #print(word, tag)
        word_t = conv(tag)
        if word_t not in (wordnet.NOUN, wordnet.ADJ, wordnet.ADV):
            continue
        lemma = lemmatizer.lemmatize(word, pos=word_t)
        synsets = wordnet.synsets(lemma, pos=word_t)
        if not synsets:
            continue
        synset = synsets[0]
        swn_synset = swn.senti_synset(synset.name())
        #print(swn_synset)
        score += swn_synset.pos_score() - swn_synset.neg_score()
print (score)

288960.8889999987
```

```
<great.s.01: PosScore=0.0 NegScore=0.0>
<small.a.01: PosScore=0.0 NegScore=0.375>
<appliance.n.01: PosScore=0.0 NegScore=0.0>
<already.r.01: PosScore=0.125 NegScore=0.0>
<expression.n.01: PosScore=0.0 NegScore=0.0>
<splinter.n.01: PosScore=0.0 NegScore=0.0>
<light.a.01: PosScore=0.0 NegScore=0.25>
<great.s.01: PosScore=0.0 NegScore=0.0>
<handy.s.01: PosScore=0.125 NegScore=0.125>
<size.n.01: PosScore=0.125 NegScore=0.375>
<however.r.01: PosScore=0.125 NegScore=0.5>
<idaho.n.01: PosScore=0.0 NegScore=0.0>
<small.a.01: PosScore=0.0 NegScore=0.375>
<high.a.01: PosScore=0.125 NegScore=0.25>
<magnification.n.01: PosScore=0.0 NegScore=0.0>
<travel.n.01: PosScore=0.0 NegScore=0.0>
<occasional.s.01: PosScore=0.0 NegScore=0.0>
<reappraisal.n.01: PosScore=0.125 NegScore=0.0>
<magnifier.n.01: PosScore=0.0 NegScore=0.0>
```

Conclusion: Thus, through this project by implementing above steps sentiment analysis by calculating polarity of reviews was successfully executed.