

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import string
from wordcloud import WordCloud
```

```
df = pd.read_csv('../input/amazon-fine-food-reviews/Reviews.csv')
```



```
-----
FileNotFoundError                                Traceback (most recent call last)
<ipython-input-12-c1b26c13b81b> in <cell line: 1>()
----> 1 df = pd.read_csv('../input/amazon-fine-food-reviews/Reviews.csv')
```

4 frames

```
/usr/local/lib/python3.10/dist-packages/pandas/io/common.py in
get_handle(path_or_buf, mode, encoding, compression, memory_map, is_text,
errors, storage_options)
```

```
    857         if ioargs.encoding and "b" not in ioargs.mode:
    858             # Encoding
--> 859             handle = open(
    860                 handle,
    861                 ioargs.mode,
```

```
FileNotFoundError: [Errno 2] No such file or directory: '../input/amazon-
fine-food-reviews/Reviews.csv'
```

```
df.head(10)
```



```
-----  
NameError                                Traceback (most recent call last)  
<ipython-input-13-4a676c3d8c05> in <cell line: 1>()  
----> 1 df.head(10)  
  
NameError: name 'df' is not defined
```

```
print(df.shape)  
print(df.isnull().values.any())  
df.dropna(axis = 0 , inplace = True)  
print(df.shape)
```



```
-----  
NameError                                Traceback (most recent call last)  
<ipython-input-14-5f5c37626150> in <cell line: 1>()  
----> 1 print(df.shape)  
      2 print(df.isnull().values.any())  
      3 df.dropna(axis = 0 , inplace = True)  
      4 print(df.shape)  
  
NameError: name 'df' is not defined
```

```
df.drop_duplicates(subset=['Score','Text'],keep='first',inplace=True)  
print(df.shape)  
df.head(10)
```



```
-----  
NameError                                Traceback (most recent call last)  
<ipython-input-15-58852d896849> in <cell line: 1>()  
----> 1 df.drop_duplicates(subset=  
    ['Score','Text'],keep='first',inplace=True)  
      2 print(df.shape)  
      3 df.head(10)  
  
NameError: name 'df' is not defined
```

```
plt.figure(figsize=(10,10))  
ax = sns.countplot(x=df["Score"], data=df, order = df["Score"].value_counts().index )  
for p, label in zip(ax.patches, df["Score"].value_counts()):  
    ax.annotate(label, (p.get_x()+0.25, p.get_height()+0.5))
```

```
df.groupby('ProductId').count()  
df_products = df.groupby('ProductId').filter(lambda x: len(x) >= 400)  
df_product_groups = df_products.groupby('ProductId')  
#Count of products and groups  
print(len(df_products))  
print(len(df_product_groups))
```

```
plt.figure(figsize=(20,20))
sns.countplot(y="ProductId", hue="Score", data=df_products);
```

```
df.groupby('UserId').count()
df_users = df.groupby('UserId').filter(lambda x: len(x) >= 100)
df_userGroup = df_users.groupby('UserId')
print("Number of Users:" + str(len(df_userGroup)))
df_products = df_users.groupby('ProductId')
print("Number of products:" + str(len(df_products)))
```

```
from nltk.tokenize import word_tokenize
from nltk.tokenize import sent_tokenize
from nltk.stem import WordNetLemmatizer
from nltk.corpus import stopwords
```

```
def remove_Stopwords(text ):
    stop_words = set(stopwords.words('english'))
    words = word_tokenize( text.lower() )
    sentence = [w for w in words if not w in stop_words]
    return " ".join(sentence)
```

```
def lemmatize_text(text):
    wordlist=[]
    lemmatizer = WordNetLemmatizer()
    sentences=sent_tokenize(text)
    for sentence in sentences:
        words=word_tokenize(sentence)
        for word in words:
            wordlist.append(lemmatizer.lemmatize(word))
    return ' '.join(wordlist)
```

```
def clean_text(text ):
    delete_dict = {sp_character: '' for sp_character in string.punctuation}
    delete_dict[' '] = ' '
    table = str.maketrans(delete_dict)
    text1 = text.translate(table)
    textArr= text1.split()
    text2 = ' '.join([w for w in textArr])

    return text2.lower()
```

```
mask = (df["Score"] == 1) | (df["Score"] == 2)
df_rating1 = df[mask]
mask = (df["Score"]==4) | (df["Score"]==5) | (df["Score"]==3)
df_rating2 = df[mask]
print(len(df_rating1))
print(len(df_rating2))
```

```
df_rating1['Text'] = df_rating1['Text'].apply(clean_text)
df_rating1['Text'] = df_rating1['Text'].apply(remove_Stopwords)
df_rating1['Text'] = df_rating1['Text'].apply(lemmatize_text)
df_rating2['Text'] = df_rating2['Text'].apply(clean_text)
df_rating2['Text'] = df_rating2['Text'].apply(remove_Stopwords)
df_rating2['Text'] = df_rating2['Text'].apply(lemmatize_text)
df_rating1['Num_words_text'] = df_rating1['Text'].apply(lambda x:len(str(x).split()))
df_rating2['Num_words_text'] = df_rating2['Text'].apply(lambda x:len(str(x).split()))
```

```
wordcloud = WordCloud(background_color="white",width=1600, height=800).generate(' '.join
plt.figure( figsize=(20,10), facecolor='k')
```

```
plt.imshow(wordcloud)
```

```
wordcloud = WordCloud(background_color="white",width=1600, height=800).generate(' '.join  
plt.figure( figsize=(20,10), facecolor='k')  
plt.imshow(wordcloud)  
plt.axis("off")
```

```
import spacy  
nlp=spacy.load('en_core_web_sm')  
from spacy import displacy  
doc=nlp(u'The blue pen was over the oval table.')
```

```
displacy.render(doc, style='dep')
```

```
doc1=nlp(u'I am Namrata Kapoor and I love NLP.')  
options={'distance':110,'compact':'True', 'color':'white','bg':'#FF5733','font':'Times'}  
displacy.render(doc1, style='dep',options=options)
```