```
!pip install Contractions
```

```
Collecting Contractions
  Downloading contractions-0.1.73-py2.py3-none-any.whl.metadata (1.2 kB)
Collecting textsearch>=0.0.21 (from Contractions)
  Downloading textsearch-0.0.24-py2.py3-none-any.whl.metadata (1.2 kB)
Collecting anyascii (from textsearch>=0.0.21->Contractions)
  Downloading anyascii-0.3.3-py3-none-any.whl.metadata (1.6 kB)
Collecting pyahocorasick (from textsearch>=0.0.21->Contractions)
  Downloading pyahocorasick-2.2.0-cp312-cp312-manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata
Downloading contractions-0.1.73-py2.py3-none-any.whl (8.7 kB)
Downloading textsearch-0.0.24-py2.py3-none-any.whl (7.6 kB)
Downloading anyascii-0.3.3-py3-none-any.whl (345 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 345.1/345.1 kB 5.2 MB/s eta 0:00:00
Downloading pyahocorasick-2.2.0-cp312-cp312-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (114 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 114.9/114.9 kB 4.3 MB/s eta 0:00:00
Installing collected packages: pyahocorasick, anyascii, textsearch, Contractions
Successfully installed Contractions-0.1.73 anyascii-0.3.3 pyahocorasick-2.2.0 textsearch-0.0.24
```

```
import re, string, unicodedata
import contractions
from bs4 import BeautifulSoup
import numpy as np
import pandas as pd
import nltk
nltk.download('punkt')
nltk.download('punkt_tab')
nltk.download('stopwords')
nltk.download('wordnet')
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize, sent_tokenize
from nltk.stem import WordNetLemmatizer
import seaborn as sns
import matplotlib.pyplot as plt
```

```
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]   Unzipping tokenizers/punkt.zip.
[nltk_data] Downloading package punkt_tab to /root/nltk_data...
[nltk_data]   Unzipping tokenizers/punkt_tab.zip.
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Unzipping corpora/stopwords.zip.
[nltk_data] Downloading package wordnet to /root/nltk_data...
```

```
data=pd.read_csv('/content/drive/MyDrive/Dataset/Reviews.csv')
```

```
data.head()
```

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

```
data.tail()
```

	Id	ProductId	UserId	ProfileName	HelpfulnessNumerator	HelpfulnessDenomina
568449	568450	B001EO7N10	A28KG5XORO54AY	Lettie D. Carter	0	
568450	568451	B003S1WTCU	A3I8AFVPEE8KI5	R. Sawyer	0	
568451	568452	B004I613EE	A121AA1GQV751Z	pksd "pk_007"	2	
568452	568453	B004I613EE	A3IBEVCTXKNOH	Kathy A. Welch "katwel"	1	
568453	568454	B001LR2CU2	A3LGQPJCZVL9UC	srfell17	0	

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 568454 entries, 0 to 568453
Data columns (total 10 columns):
 #   Column                  Non-Null Count   Dtype
---  ------                  --------------   -----
 0   Id                      568454 non-null  int64
 1   ProductId               568454 non-null  object
 2   UserId                  568454 non-null  object
 3   ProfileName             568428 non-null  object
 4   HelpfulnessNumerator    568454 non-null  int64
 5   HelpfulnessDenominator  568454 non-null  int64
 6   Score                   568454 non-null  int64
 7   Time                    568454 non-null  int64
 8   Summary                 568427 non-null  object
 9   Text                    568454 non-null  object
dtypes: int64(5), object(5)
memory usage: 43.4+ MB
```

```
data.shape
```

```
(568454, 10)
```

```
data.describe()
```

	Id	HelpfulnessNumerator	HelpfulnessDenominator	Score	Time
count	568454.000000	568454.000000	568454.00000	568454.000000	5.684540e+05
mean	284227.500000	1.743817	2.22881	4.183199	1.296257e+09
std	164098.679298	7.636513	8.28974	1.310436	4.804331e+07
min	1.000000	0.000000	0.00000	1.000000	9.393408e+08
25%	142114.250000	0.000000	0.00000	4.000000	1.271290e+09
50%	284227.500000	0.000000	1.00000	5.000000	1.311120e+09
75%	426340.750000	2.000000	2.00000	5.000000	1.332720e+09
max	568454.000000	866.000000	923.00000	5.000000	1.351210e+09

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

	0
Id	0
ProductId	0
UserId	0
ProfileName	26
HelpfulnessNumerator	0
HelpfulnessDenominator	0
Score	0
Time	0
Summary	27
Text	0

dtype: int64

```
data['sentiment'] = data['Score'].apply(lambda score:'Positive' if score > 3 else 'negative')
```

```
import seaborn as sns
```

```
from wordcloud import WordCloud,STOPWORDS
```

```
words=' '.join(data['Text'])
```

```
word_Cloud=WordCloud(stopwords=STOPWORDS,background_color='black',width=3000,height=2500).generate(wor
```

```
word_Cloud
```

```
<wordcloud.wordcloud.WordCloud at 0x7c55ad030560>
```

```
plt.figure(1,figsize=(12,6))
plt.imshow(word_Cloud)
```

```
plt.axis('off')
plt.show()
```


```
Cleanered_words=word_Cloud.process_text(words)
```

```
Cleanered_words
```

```
 'manufactured': 842,
'hand': 12327,
'cream': 2830,
'stumbled': 120,
'idea': 5445,
'making': 12125,
'lowfat': 78,
'sweeter': 1392,
```