In [1]:
```
#Importing all the Libraries


import numpy as np
import pandas as pd
import nltk
from nltk.sentiment.vader import SentimentIntensityAnalyzer
import re
import transformers
import warnings
warnings.filterwarnings('ignore')
warnings.warn('this will not show')
pd.set_option('display.max_columns',None)
```

```
Neither PyTorch nor TensorFlow >= 2.0 have been found.Models won't be available and only tokenizers, co
nfigurationand file/data utilities can be used.
```

In [2]:
```
#reading the two datasets

data1= pd.read_csv(r"C:\Users\groni\OneDrive\Desktop\Sentiment Analysis - Case Study\Customer Reviews - (
data2=pd.read_csv(r"C:\Users\groni\OneDrive\Desktop\Sentiment Analysis - Case Study\Customer Reviews - C(
```

In [3]:
```
# Dropping two extra columns from two data sets

d1=data1.drop(['reviews.didPurchase'],axis=1)
d2=data2.drop(['reviews.dateAdded'],axis=1)
```

In [4]:
```
# Join two data sets

data=pd.concat([d1,d2])
```

In [5]:
```
# Export the joined data to change the 'reviews.text' column format to text

data.to_csv('final.csv')
```

In [6]:
```
#again import the data

data=pd.read_csv('final.csv')
```

In [7]:
```
#lower_casing the reviews
data['reviews.text']=data['reviews.text'].str.lower()
```

In [8]:
```

data['reviews.text']=data['reviews.text'].astype(str)
```

In [9]:
```
#sentence_tokenizing
data['sen_tok']=0
for i in range(len(data)):

    data['sen_tok'][i]=nltk.sent_tokenize(data['reviews.text'][i])
```

In [10]:
```
#word_tokenizing
data['word_tok']=0
for i in range(len(data)):

    data['word_tok'][i]=nltk.word_tokenize(data['reviews.text'][i])
```

In [11]:
```
data['cleaned']=0
for i in range(len(data)):

    for words in data['word_tok'][i]:
        clean=[]
        for w in words:
            res=re.sub(r'[^a-z^A-Z]','',w)
            if res !=" ":
                clean.append(res)
            data['cleaned'][i]=clean
```

In [12]:
```
from nltk.tokenize import sent_tokenize ,word_tokenize
import nltk
nltk.download('punkt')
#sent_tokenize
sent_tok1=[]
for sent in data['reviews.text']:
    sent=sent_tokenize(sent)
    sent_tok1.append(sent)
```

```
[nltk_data] Downloading package punkt to
[nltk_data]     C:\Users\groni\AppData\Roaming\nltk_data...
[nltk_data]   Package punkt is already up-to-date!
```

In [13]:
```
#importing stopwords and lemmatizer
from nltk.corpus import stopwords
from nltk.stem import WordNetLemmatizer
lemitizer=WordNetLemmatizer()
```

In [14]:
```
#lemmatizing & removing stopwords
data['cleaned']=' '
for i in range(len(data)):
    words=nltk.word_tokenize(data['reviews.text'][i])
    words=[lemitizer.lemmatize(word) for word in words if word not in set(stopwords.words('english'))]
    data['cleaned'][i]=words
```

In [15]:
```
#Again join for polarising

data['join']=' '
for p in range(len(data)):
    l=" "
    for u in data['cleaned'][p]:
        l=l+' '+u
    data['join'][p]=l
```

In [16]:
```
#importing sentiment analyzer
from nltk.corpus import stopwords
from nltk.sentiment import SentimentIntensityAnalyzer

o = SentimentIntensityAnalyzer()
```

In [17]:
```
#sentiment analysis
data['positive']=[o.polarity_scores(a)['pos'] for a in data['join']]
data['negative']=[o.polarity_scores(b)['neg'] for b in data['join']]
data['neutral']=[o.polarity_scores(c)['neu'] for c in data['join']]
data['compound']=[o.polarity_scores(d)['compound'] for d in data['join']]
```

In [18]:
```
# sentiment of the sentence

data['neutral_1/0']=0
data['positive_1/0']=0
data['negative_1/0']=0
for i in range(len(data)):
    if (data['positive'][i]> data['negative'][i] and data['positive'][i]> data['neutral'][i]):
        data['positive_1/0'][i]=1
    elif (data['negative'][i]> data['positive'][i] and data['negative'][i]> data['neutral'][i]):
        data['negative_1/0'][i]=1
    else:
        data['neutral_1/0'][i]=1
```

In [19]:
```
#creating_senrtiment_column
data['senti']=" "
for i in range(len(data)):
    if (data['positive'][i]> data['negative'][i] and data['positive'][i]> data['neutral'][i]):
        data['senti'][i]='positive'
    elif (data['negative'][i]> data['positive'][i] and data['negative'][i]> data['neutral'][i]):
        data['senti'][i]='negative'
    else:
        data['senti'][i]='neutral'
```

In [21]:
```
data.to_csv('finally.csv')
```