

NLP

Natural Language Processing

Emotion Mining

P125 , Team - 2

Project | product | 2_EDA | Review | Text_P | 2_EDA | Amazo | Amazo | Colab | Untitle | 2_EDA | emotio | Text-M | Dashb | Assign | Bo x | + | - | X

← → ↺ ↻ ↵ 🔍 📄 📁 📂 📅 📆 📇 📈 📉 📊 📋 📌 📍 📎 📏 📐 📑 📒 📓 📔 📕 📖 📗 📙 📚 📛 📜 📝 📞 📟 📠 📡 📢 📣 📤 📥 📦 📧 📨 📩 📪 📫 📬 📭 📮 📯 📰 📱 📲 📳 📴 📵 📶 📷 📸 📹 📺 📻 📼 📽 📾 📿 📰 📱 📲 📳 📴 📵 📶 📷 📸 📹 📺 📻 📼 📽 📾 📿

https://www.amazon.in/Airdopes-441-Technology-Immersive-Resistance/dp/B084DS51NC/ref=cm_cr_ar_p_d_product_top?ie=UTF8

Explore Live Classes... Course: PCAP - Pro... New Tab Data Science: 7-9 ... Copy of Python ses... ASUS Software Port... MyASUS Software ... McAfee LiveSafe Dashboard | ExcelR GitHub - mwaskom...

Electronics > Headphones > In-Ear

6 VIDEOS

Boat Airdopes 441 Upto 30 Hours Playback, Boat Signature Sound, lwp Technology, lpx7, Type-C Interface and Capacitive Touch Controls V5.0 Bluetooth Truly Wireless In Ear Earbuds with Mic (Raging Red)

Visit the boAt Store

★★★★☆ 136,530 ratings | 1000+ answered questions

7 Days 1 Year Warranty Replacement

Currently unavailable.
We don't know when or if this item will be back in stock.
Select delivery location

Add to Wish List

Have one to sell?
Sell on Amazon

Deal of the Day

Save 56%
CROSSBEATS Neobuds Bluetooth...
₹2,199.00 ₹4,999.00 ✓prime

Sponsored

Colour: Raging Red

Style name: Airdopes 441

Airdopes 441 Airdopes 441 Pro Made In India Masaba IPL T-Rebel

Brand	BoAt
Ear Placement	In Ear
Colour	Raging Red
Connector Type	Wireless
Model Name	Airdopes 441

About this item

Roll over image to zoom in

24°C Partly sunny

ENG IN 08:22 15-06-2022

Extracting
sentiment
form
customer
reviews on
a product.

Natural Language Processing-

- NLP is a application of Artificial Intelligence of Computer Science.
- It is a inter mediation between human language and machine language.
- Text are unstructured data.
- Extracting high quality information from text.
- We do text optimization by removing unwanted word, symbols and lemmatize the text.
- After optimize the text we go for NLP by converting text to number then model creation and prediction.

Emotion Mining-

- A major task that the NLP (Natural Language Processing) has to follow is Sentiments analysis.
- Emotion mining from text refers to the detection of people's emotions based on observations of their writings.
- For finding whether the user's attitude is positive, neutral or negative, it captures each user's opinion and feelings about the corresponding product.
- Through this, needed changes can well be done on the product for better customer contentment by the companies.

Amazon Review

Web Scraping

- Amazon review on boat ear dopes.
- Scraping review from amazon website by using bs4 and request library.
- Requests will allow you to send HTTP requests using Python.

!pip install requests

`import` requests

- BeautifulSoup is a Python library that is used for web scraping purposes to pull the data out of HTML and XML files.
- It creates a parse tree from page source code that can be used to extract data in a hierarchical and more readable manner.

!pip install bs4

from bs4 `import` BeautifulSoup

```
1 def get_reviews(soup):
2     reviews = soup.find_all('div', {'data-hook': 'review'})
3     try:
4         for item in reviews:
5             review = (item.find('span', {'data-hook': 'review-body'}), text.strip())
6             reviewlist.append(review)
7     except:
8         pass

1 for x in range(1,20):
2     soup = html_code(f'https://www.amazon.in/Airdopes-441-Technology-Immersive-Resistance/product-reviews/B084D551NC/ref=cm_')
3     print(f'Getting page:')
4     get_reviews(soup)
5     print(len(reviewlist))
6     if not soup.find('li', {'class': 'a-disabled a-last'}):
7         pass
8     else:
9         break

10 df = pd.DataFrame(reviewlist)
11 df.to_excel('earbud.xlsx', index=False)
12 print('Fin.')
```

Getting page:

10

Getting page:

20

Getting page:

30

Getting page:

40

Getting page:

50

Getting page:

60

- We check for null value and duplicate data in the review file.
- We apply tokenization, normalisation, lemmatisation and transform text into Data Structure.

Tokenization-

- We split the sentences or document into chunks into meaningful words.
- It generally comprise word and number but we can include punctuation mark, symbols etc.

```
import nltk
from nltk.tokenize import word_tokenize
nltk.download('punkt')
token = word_tokenize(join)
token

[nltk_data] Downloading package punkt to
[nltk_data] C:\Users\lavanya\AppData\Roaming\nltk_data...
[nltk_data] Package punkt is already up-to-date!
['really',
 'impressed',
 'with',
 'the',
 'fastest',
 'delivery',
 'of',
 'amazon',
 'i',
 'ordered',
 'it',
 'on',
 'yesterday',
 'at',
 'and',
 'they',
 'delivered',
 'me',
 'at',
 'on',
 'the',
 'same',
 'day',
 'it',
```

Normalisation-

- Transform various form of same word into a common form.
- We remove all the punctuation marks, special character and unwanted characters, extra spaces etc.
- Transform all words to lower case .
Ex-HAPPY, Happy, happY = happy

Stop word-

- Stop word have no significant importance or they carry very little useful information on the natural data so we remove the stop word.
Ex- 'a', 'the', 'is', 'are' etc
- We remove 'no' and 'not' from stopword as they might change the meaning of some review.

```
import nltk
nltk.download('stopwords')
from nltk.corpus import stopwords
stop = stopwords.words('english')
```

Stemming and Lemmatization-

- Stemming and lemmatization are methods used by search engines and chat bots to analyze the meaning behind a word.
- Stemming uses the stem of the word.
- Algorithms work by cutting off the end or the beginning of the word, taking into account a list of common prefixes and suffixes that can be found in an inflected word. Stemmed word might have or don't have proper meaning.

Ex- run – run , running –run, ran –ran, runs – run, easily – easili , fairly – fairli

Lemmatization-

Lemmatization uses the context in which the word is being used.

It considers the morphological analysis of the words. To do so, it is necessary to have detailed dictionaries which the algorithm can look through to link the form back to its lemma.

Ex-saw VERB see

eighteen NUM eighteen

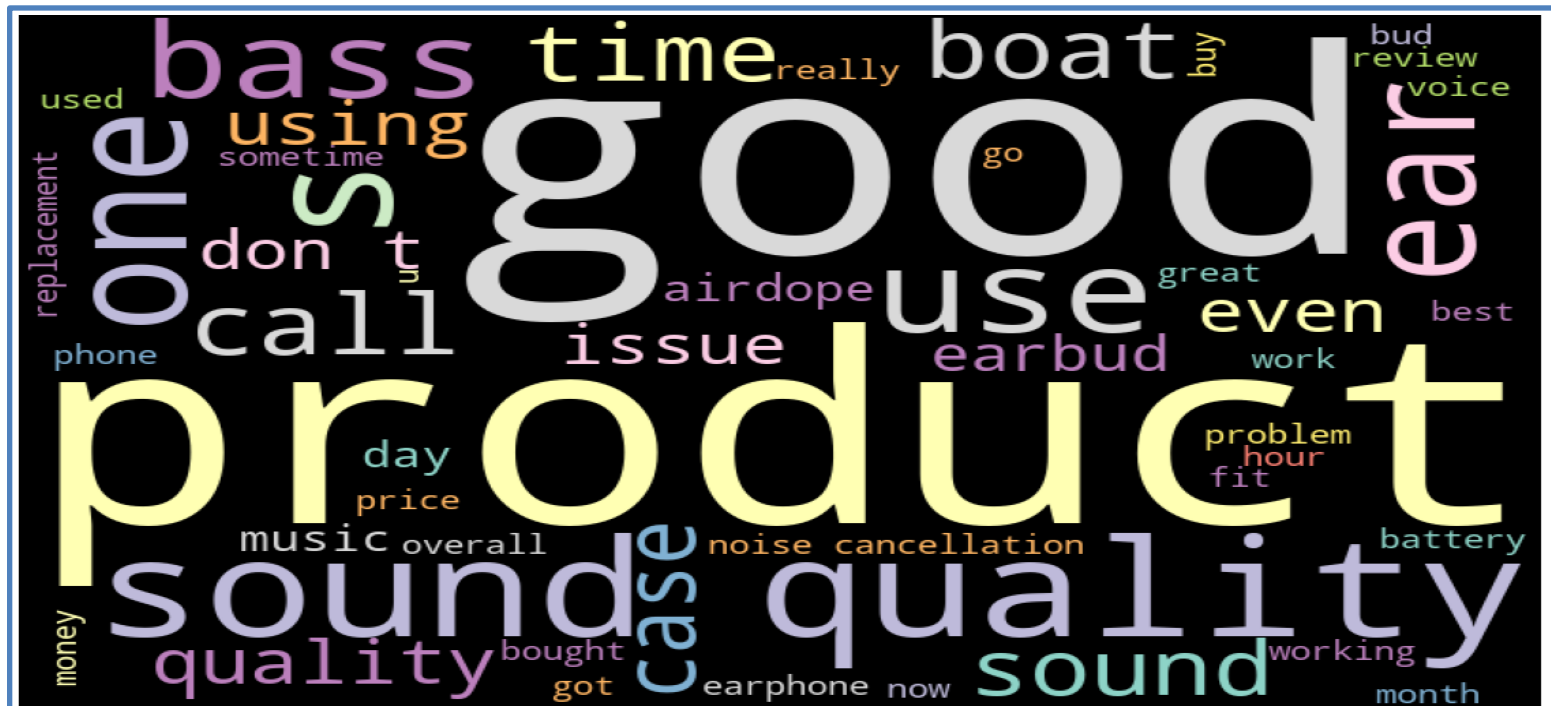
mice NOUN mouse

today NOUN today

Dataset-

Unnamed: 0		Reviews	sentence	clean_text	polarity	sentiment
0	0	Really impressed with the fastest delivery of ...	Really impressed with the fastest delivery of ...	really impressed with the fastest delivery of ...	0.5469	positive
1	1	This is the second time I am writing this revi...	This is the second time I am writing this revi...	this is the second time i am writing this revi...	0.1152	positive
2	2	1. Quite good look2. Better noise cancellation...	1. Quite good look2. Better noise cancellation...	quite good look better noise cancellation good...	0.2518	positive
3	3	its a good set of ear buds but1. the quality o...	its a good set of ear buds but1. the quality o...	its a good set of ear buds but the quality of ...	0.0104	positive
4	4	I have been researching about getting a nice T...	I have been researching about getting a nice T...	i have been researching about getting a nice t...	0.1836	positive
...
3595	3595	After searching for too many TWS finally I bou...	After searching for too many TWS finally I bou...	after searching for too many tws finally i bou...	0.2417	positive
3596	3596	The casing is little boxy but still okay to ca...	The casing is little boxy but still okay to ca...	the casing is little boxy but still okay to ca...	0.4153	positive
3597	3597	Worst product of my life, when i use boat aird...	Worst product of my life, when i use boat aird...	worst product of my life when i use boat airdr...	-0.3667	negative

- ```
%matplotlib inline
```



## Sentimental Score-

- It is a scaling system that reflects the emotional depth of emotions in a piece of text via numbers.
- We have use following libraries for sentimental score-
  1. Affin
  2. Textlob
  3. Vader

## Textlob-

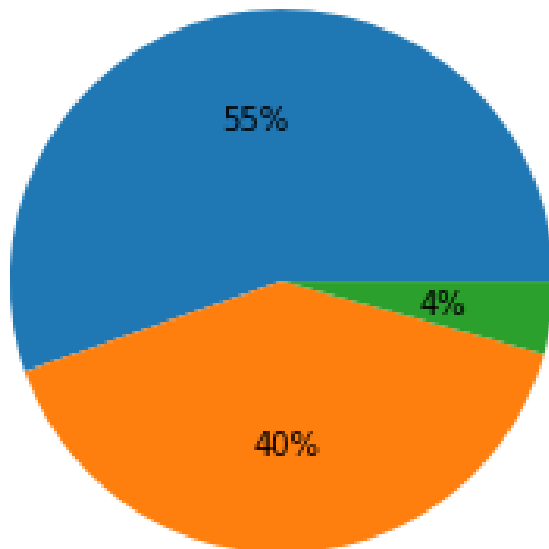
- Textlob return polarity value between -1.0 to 1.0 in float, where 0 is nutral

```
from textblob import TextBlob
from textblob import Word
```

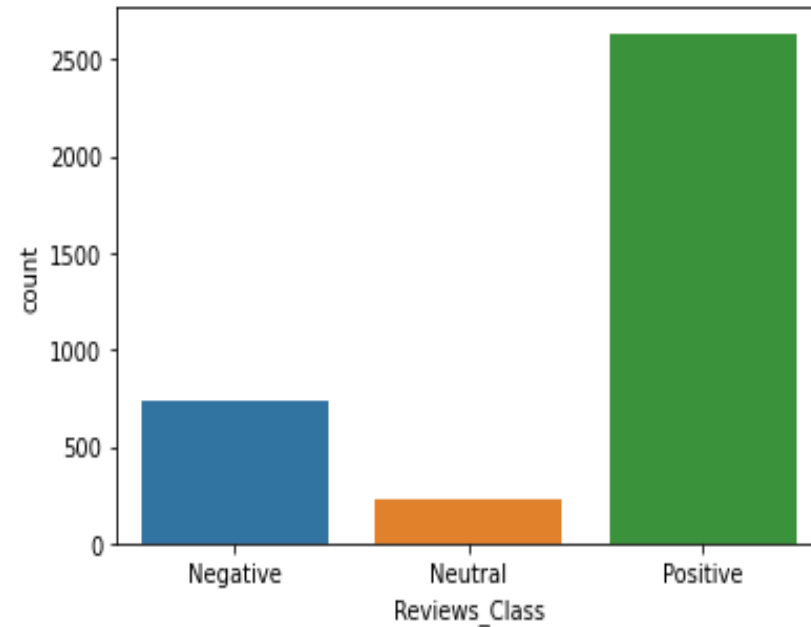
```
Reviews['polarity'] = Reviews["Reviews1"].apply(lambda x: TextBlob(x).sentiment[0])
Reviews[['Reviews1', 'polarity']].head(5)
```

|   | Reviews1                                          | polarity |
|---|---------------------------------------------------|----------|
| 0 | really impressed fastest delivery amazon order... | 0.657143 |
| 1 | second time writing review sure earlier detail... | 0.157843 |
| 2 | quite good look better noise cancellation good... | 0.287755 |
| 3 | good set ear bud quality sound get bad connect... | 0.052500 |
| 4 | researching getting nice tws bud everyday usem... | 0.241146 |

```
plt.pie(df['Reviews_Class'].value_counts(), labels=('Positive', 'Negative', 'Neutral'), autopct='%.2f')
plt.legend();
```



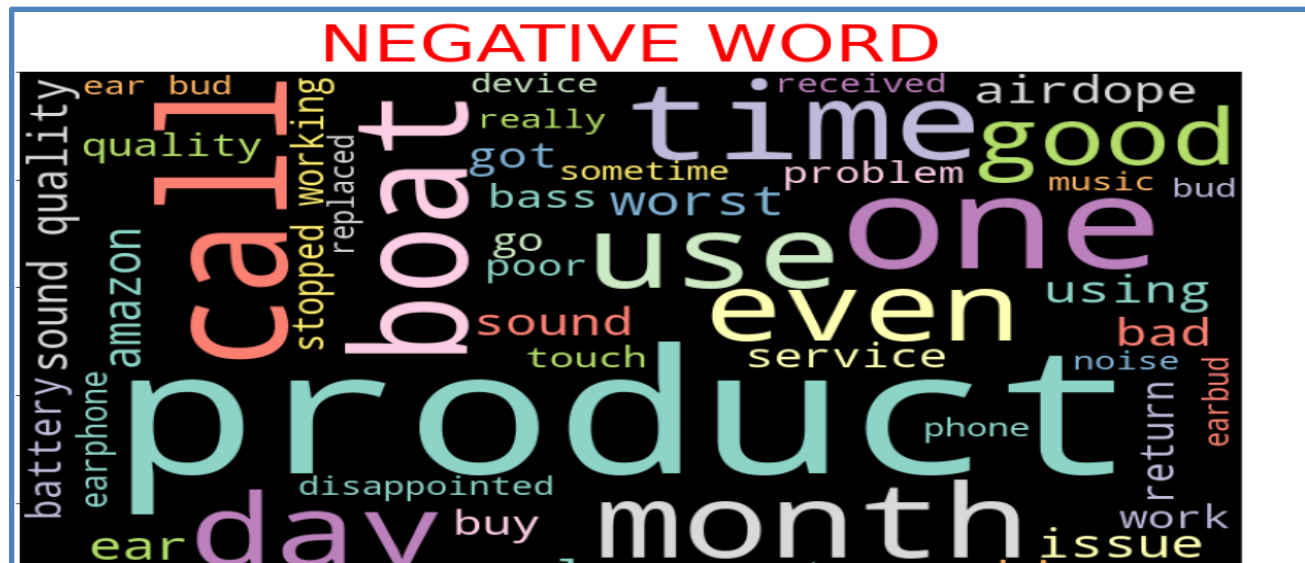
```
sns.countplot(x='Reviews_Class', data=df);
```



-Positive

-Negative

-Neutral



## Features and Target –

- Here the features is the review column which we have convert to numeric format by using Tfidf fit\_transform .
- Target is the sentiment column.

|      | Reviews                                           | Reviews_value |
|------|---------------------------------------------------|---------------|
| 0    | really impressed fastest delivery amazon order... | 14            |
| 1    | second time writing review not sure earlier de... | 1             |
| 2    | quite good better noise cancellation good base... | 12            |
| 3    | good set ear buds quality sound gets bad conne... | 3             |
| 4    | researching getting nice tws buds everyday use... | 31            |
| ...  | ...                                               | ...           |
| 3595 | searching many tws finally bought n fully sati... | 1             |
| 3596 | casing little boxy still okay carry around tou... | 6             |
| 3597 | worst product life use boat airdropes call kee... | 0             |
| 3598 | im really disappointed sound quality bad thoug... | -8            |
| 3599 | want use music video good calls worst much noi... | -2            |

3600 rows × 2 columns

## TFIDF-

- Term frequency Inverse document frequency.
- We TFIDF to get words weightage and frequency on a given document.
- We can select max feature s and the max frequency from the document.

```

2 from sklearn.feature_extraction.text import TfidfVectorizer
3 vectorizer = TfidfVectorizer(stop_words='english', max_features= 1000, max_d
4 vectorizer

```

TfidfVectorizer(max\_df=0.5, max\_features=1000, stop\_words='english')

```

1 doc_vec = vectorizer.fit_transform(df["lemma"])
2 X=doc_vec

```

```

1 doc_vec.toarray()

```

```

array([[0., 0., 0., ..., 0., 0., 0.],
 [0., 0., 0., ..., 0., 0., 0.],
 [0., 0., 0., ..., 0., 0., 0.],
 ...,
 [0., 0., 0., ..., 0., 0., 0.],
 [0., 0., 0., ..., 0., 0., 0.],
 [0., 0., 0., ..., 0., 0., 0.]])

```

## Model-

- In NLP we create models using various supervised method-
  1. MultinomialNB
  2. Random forest
  3. Decision tree classifier
  4. SVC
  5. Logistic Regression
- From the above model we did train test split of 75:25, and calculate the accuracy, confusion matrix.

|   | Model                   | test_accuracy |
|---|-------------------------|---------------|
| 0 | Multinomial Naive Bayes | 0.7960        |
| 1 | Logistic Regression     | 0.8419        |
| 2 | Decision Tree           | 0.7528        |
| 3 | Random Forest           | 0.8545        |
| 4 | Ada Boost               | 0.7528        |



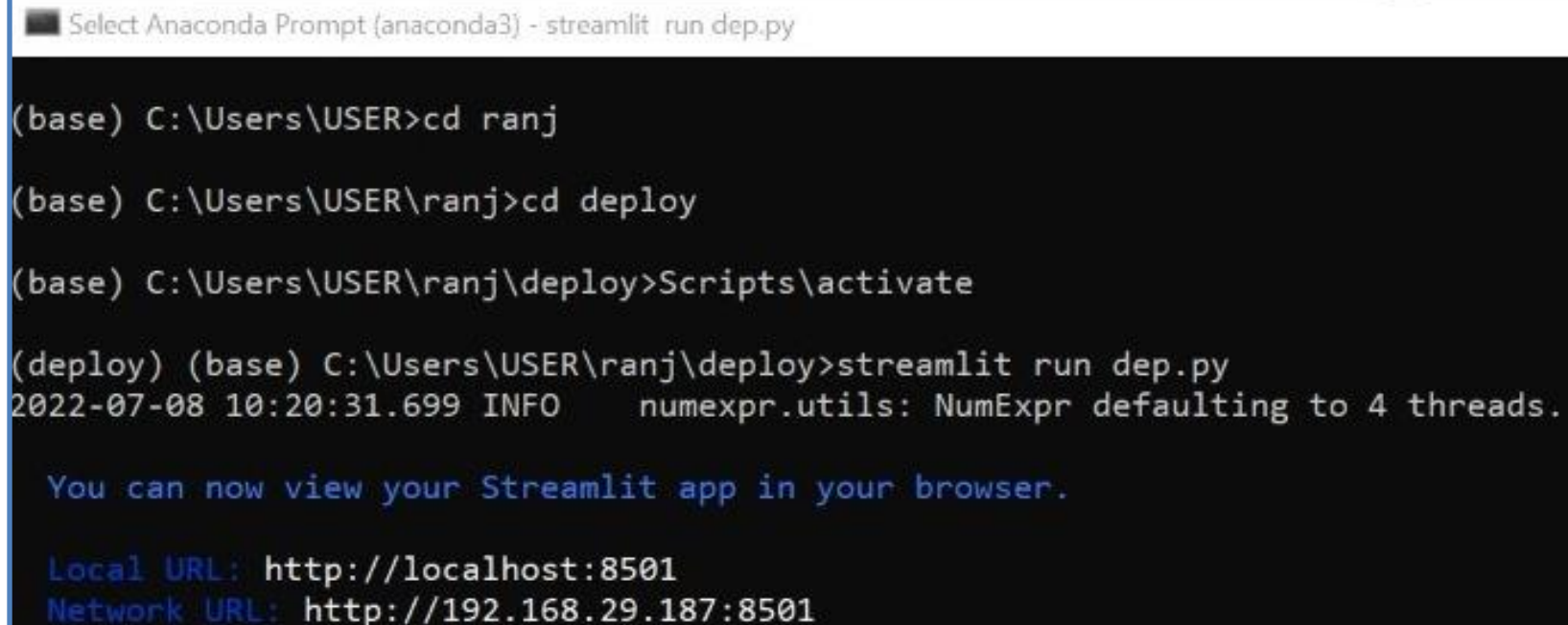
## Model Selection-

- We got best fit for data in MultinomialNB from the rest .
- Got the accuracy approx. 80%
- Then we decided to go with MultinomialNB model

- Created user define function to check whether model is predicting output properly or not
- Then imported pickle library to dump vectorizer and model (MultinomialNB) file
- Then created pipeline to deploy code to check product review
- In pipeline we have added two pickle file viz vectorizer and model file and then added punctuation, stemming, TextBlob lexicon Library code.

## Model Deployment-

- After creation of pipeline we open Anaconda command prompt
- Then entered required command to deploy our model



```
Select Anaconda Prompt (anaconda3) - streamlit run dep.py

(base) C:\Users\USER>cd ranj

(base) C:\Users\USER\ranj>cd deploy

(base) C:\Users\USER\ranj\deploy>Scripts\activate

(deploy) (base) C:\Users\USER\ranj\deploy>streamlit run dep.py
2022-07-08 10:20:31.699 INFO numexpr.utils: NumExpr defaulting to 4 threads.

You can now view your Streamlit app in your browser.

Local URL: http://localhost:8501
Network URL: http://192.168.29.187:8501
```

## Model Deployment-

- In Anaconda prompt we downloaded streamlit
- Then open Visual Studio Code (vs code), save the pipeline file
- Now run the code in anaconda prompt or vs code
- `streamlit run dep.py` (here .py is python file and dep could be any name)
- After execution of code it will take you to browser
- Now enter the review and get the result of sentiment and polarity.

## Conclusion-

- Now we have a clean data which can be use for worldcloud, Emotion analysis, Positive, negative, neutral statement, model creation for training and testing data.

- p125 team 2

Please enter your review

The first generation neckband was slightly treble heavy. This one has extra bass. But in no way is the b

Prediction

**Positive** 😄

The polarity of the review is: 0.2207

Thank you!