

## EX.NO:4 Text Preprocessing and analytics Pipeline

### AIM:

To determine the text Preprocessing and data analytics Pipeline.

### CODE:

```
import pandas as pd
import re

nlp = Spacy. load ("en-core-web")
df = pd. read - csv ('amazon - revsu. csv')
Print (df [reviewText] load ())

# function to clean the test casting speech
def clean - text - space (text):
    if pd. isnull (text)
    return []
    text = text. lower ()
    text = re.sub (r ^ [^\w\s], " ")
    # Tokensize using spay
    doc = nlp (text)
    token = (token. text for token in doc if not
              token. is - stop and not token. is - pd)
    return token s.
```

Top is frequent word in Amazon Reviews

['', 3203] ('hide', 1447), ('int', 962) ('buds', 609)

(kinder', 1561) (sown, 473) ('like', 452)

(reel', 434), (great', 422) ('use', 420)

('tu', 380), ('select', 347) ('good', 529)

('device', 329), ('bo', 522)

(new - 2010 - 10) level . 2010 - 2010 = 0/0

old = old - new (new - old) = 0/0

Point (at [Amazon] level)

#function to return the list of words

old - new - list of words

at [Amazon] level

return

new - old - list of words

new - old - list of words

#function to return the list of words

at [Amazon] level

old - new - list of words

new - old - list of words

all - tokens = [token for token in df['cloud-  
tokens'] for token in tokens]

Print ("In TOP is frequent words in Amazon  
Review:")

Print (word - freq . most - conver (15))

RESULT :

The given Test Preprocessing and analytic  
pipeline has been excellent successfully.