untitled17

April 14, 2023

[1]:
```python
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

[2]:
```python
# Load the datasets
train_df = pd.read_csv('train_data.csv')
test_df = pd.read_csv('test_data.csv')
```

[3]:
```python
train_df.head()
```

[3]:
```
                                                name   brand  \
0  All-New Fire HD 8 Tablet, 8" HD Display, Wi-Fi…  Amazon
1        Amazon - Echo Plus w/ Built-In Hub - Silver  Amazon
2  Amazon Echo Show Alexa-enabled Bluetooth Speak…  Amazon
3  Fire HD 10 Tablet, 10.1 HD Display, Wi-Fi, 16 …  Amazon
4  Brand New Amazon Kindle Fire 16gb 7" Ips Displ…  Amazon

                                        categories  \
0  Electronics,iPad & Tablets,All Tablets,Fire Ta…
1  Amazon Echo,Smart Home,Networking,Home & Tools…
2  Amazon Echo,Virtual Assistant Speakers,Electro…
3  eBook Readers,Fire Tablets,Electronics Feature…
4  Computers/Tablets & Networking,Tablets & eBook…

             primaryCategories              reviews.date  \
0                  Electronics  2016-12-26T00:00:00.000Z
1         Electronics,Hardware  2018-01-17T00:00:00.000Z
2         Electronics,Hardware  2017-12-20T00:00:00.000Z
3  Office Supplies,Electronics  2017-08-04T00:00:00.000Z
4                  Electronics  2017-01-23T00:00:00.000Z

                                      reviews.text  \
0  Purchased on Black FridayPros - Great Price (e…
1  I purchased two Amazon in Echo Plus and two do…
2  Just an average Alexa option. Does show a few …
3  very good product. Exactly what I wanted, and …
```

```
4  This is the 3rd one I've purchased. I've bough…

              reviews.title sentiment
0            Powerful tablet  Positive
1  Amazon Echo Plus AWESOME  Positive
2                   Average   Neutral
3               Greattttttt  Positive
4             Very durable!  Positive
```

[4]:
```
test_df.head()
```

[4]:
```
                                                name   brand  \
0  Fire Tablet, 7 Display, Wi-Fi, 16 GB - Include…  Amazon
1  Amazon Echo Show Alexa-enabled Bluetooth Speak…  Amazon
2  All-New Fire HD 8 Tablet, 8" HD Display, Wi-Fi…  Amazon
3  Brand New Amazon Kindle Fire 16gb 7" Ips Displ…  Amazon
4  Amazon Echo Show Alexa-enabled Bluetooth Speak…  Amazon

                                          categories     primaryCategories  \
0  Fire Tablets,Computers/Tablets & Networking,Ta…           Electronics
1  Computers,Amazon Echo,Virtual Assistant Speake…  Electronics,Hardware
2  Electronics,iPad & Tablets,All Tablets,Fire Ta…           Electronics
3  Computers/Tablets & Networking,Tablets & eBook…           Electronics
4  Computers,Amazon Echo,Virtual Assistant Speake…  Electronics,Hardware

               reviews.date  \
0  2016-05-23T00:00:00.000Z
1  2018-01-02T00:00:00.000Z
2  2017-01-02T00:00:00.000Z
3  2017-03-25T00:00:00.000Z
4  2017-11-15T00:00:00.000Z

                                        reviews.text  \
0  Amazon kindle fire has a lot of free app and c…
1  The Echo Show is a great addition to the Amazo…
2  Great value from Best Buy. Bought at Christmas…
3  I use mine for email, Facebook ,games and to g…
4  This is a fantastic item & the person I bought…

                      reviews.title
0                very handy device
1        Another winner from Amazon
2  simple to use and reliable so far
3                        Love it!!!
4                        Fantastic!
```

[5]:
```
test_hid = pd.read_csv('test_data_hidden.csv')
test_hid.head()
```

[5]:
```
                                                name   brand  \
0  Fire Tablet, 7 Display, Wi-Fi, 16 GB - Include…  Amazon
1  Amazon Echo Show Alexa-enabled Bluetooth Speak…  Amazon
2  All-New Fire HD 8 Tablet, 8" HD Display, Wi-Fi…  Amazon
3  Brand New Amazon Kindle Fire 16gb 7" Ips Displ…  Amazon
4  Amazon Echo Show Alexa-enabled Bluetooth Speak…  Amazon

                                          categories     primaryCategories  \
0  Fire Tablets,Computers/Tablets & Networking,Ta…           Electronics
1  Computers,Amazon Echo,Virtual Assistant Speake…  Electronics,Hardware
2  Electronics,iPad & Tablets,All Tablets,Fire Ta…           Electronics
3  Computers/Tablets & Networking,Tablets & eBook…           Electronics
4  Computers,Amazon Echo,Virtual Assistant Speake…  Electronics,Hardware

               reviews.date  \
0  2016-05-23T00:00:00.000Z
1  2018-01-02T00:00:00.000Z
2  2017-01-02T00:00:00.000Z
3  2017-03-25T00:00:00.000Z
4  2017-11-15T00:00:00.000Z

                                        reviews.text  \
0  Amazon kindle fire has a lot of free app and c…
1  The Echo Show is a great addition to the Amazo…
2  Great value from Best Buy. Bought at Christmas…
3  I use mine for email, Facebook ,games and to g…
4  This is a fantastic item & the person I bought…

                       reviews.title sentiment
0                    very handy device  Positive
1           Another winner from Amazon  Positive
2  simple to use and reliable so far  Positive
3                          Love it!!!  Positive
4                          Fantastic!  Positive
```

[6]:
```
#Create a master dataset by combining train and test dataset
df=pd.concat([train_df,test_hid],axis=0)
df.head()
```

[6]:
```
                                                name   brand  \
0  All-New Fire HD 8 Tablet, 8" HD Display, Wi-Fi…  Amazon
1        Amazon - Echo Plus w/ Built-In Hub - Silver   Amazon
2  Amazon Echo Show Alexa-enabled Bluetooth Speak…  Amazon
3  Fire HD 10 Tablet, 10.1 HD Display, Wi-Fi, 16 …  Amazon
```

```
4  Brand New Amazon Kindle Fire 16gb 7" Ips Displ…  Amazon

                                          categories  \
0  Electronics,iPad & Tablets,All Tablets,Fire Ta…
1  Amazon Echo,Smart Home,Networking,Home & Tools…
2  Amazon Echo,Virtual Assistant Speakers,Electro…
3  eBook Readers,Fire Tablets,Electronics Feature…
4  Computers/Tablets & Networking,Tablets & eBook…

             primaryCategories              reviews.date  \
0                  Electronics  2016-12-26T00:00:00.000Z
1         Electronics,Hardware  2018-01-17T00:00:00.000Z
2         Electronics,Hardware  2017-12-20T00:00:00.000Z
3  Office Supplies,Electronics  2017-08-04T00:00:00.000Z
4                  Electronics  2017-01-23T00:00:00.000Z

                                         reviews.text  \
0  Purchased on Black FridayPros - Great Price (e…
1  I purchased two Amazon in Echo Plus and two do…
2  Just an average Alexa option. Does show a few …
3  very good product. Exactly what I wanted, and …
4  This is the 3rd one I've purchased. I've bough…

             reviews.title sentiment
0          Powerful tablet  Positive
1  Amazon Echo Plus AWESOME  Positive
2                  Average   Neutral
3              Greattttttt  Positive
4            Very durable!  Positive
```

[7]:
```python
#Check for the target value counts
df['sentiment'].value_counts()
```

[7]:
```
Positive    4686
Neutral      197
Negative     117
Name: sentiment, dtype: int64
```

[8]:
```python
df.isnull().sum()
```

[8]:
```
name                  0
brand                 0
categories            0
primaryCategories     0
reviews.date          0
reviews.text          0
reviews.title        13
```

```
sentiment             0
dtype: int64
```

[9]:

```
df.isnull().sum()
```

[9]:

```
name                  0
brand                 0
categories            0
primaryCategories     0
reviews.date          0
reviews.text          0
reviews.title        13
sentiment             0
dtype: int64
```

[10]:

```
df[df['reviews.title'].isnull()==True]
```

[10]:

```
                                                  name   brand  \
834   Amazon Echo Show Alexa-enabled Bluetooth Speak…  Amazon
1268  Amazon Echo Show Alexa-enabled Bluetooth Speak…  Amazon
1695  Amazon Echo Show Alexa-enabled Bluetooth Speak…  Amazon
1824  Amazon Echo Show Alexa-enabled Bluetooth Speak…  Amazon
2786  Amazon Fire TV with 4K Ultra HD and Alexa Voic…  Amazon
2822  Amazon Echo Show Alexa-enabled Bluetooth Speak…  Amazon
2933  Amazon Echo Show Alexa-enabled Bluetooth Speak…  Amazon
3103  Amazon Echo Show Alexa-enabled Bluetooth Speak…  Amazon
3224  Amazon Echo Show Alexa-enabled Bluetooth Speak…  Amazon
3690  Amazon Echo Show Alexa-enabled Bluetooth Speak…  Amazon
222   Amazon Echo Show Alexa-enabled Bluetooth Speak…  Amazon
366   Amazon Echo Show Alexa-enabled Bluetooth Speak…  Amazon
570   Amazon Echo Show Alexa-enabled Bluetooth Speak…  Amazon

                                            categories     primaryCategories  \
834   Computers,Amazon Echo,Virtual Assistant Speake…  Electronics,Hardware
1268  Amazon Echo,Virtual Assistant Speakers,Electro…  Electronics,Hardware
1695  Computers,Amazon Echo,Virtual Assistant Speake…  Electronics,Hardware
1824  Computers,Amazon Echo,Virtual Assistant Speake…  Electronics,Hardware
2786  Amazon SMP,TV, Video & Home Audio,Electronics,…           Electronics
2822  Computers,Amazon Echo,Virtual Assistant Speake…  Electronics,Hardware
2933  Amazon Echo,Virtual Assistant Speakers,Electro…  Electronics,Hardware
3103  Amazon Echo,Virtual Assistant Speakers,Electro…  Electronics,Hardware
3224  Computers,Amazon Echo,Virtual Assistant Speake…  Electronics,Hardware
3690  Computers,Amazon Echo,Virtual Assistant Speake…  Electronics,Hardware
222   Amazon Echo,Virtual Assistant Speakers,Electro…  Electronics,Hardware
366   Computers,Amazon Echo,Virtual Assistant Speake…  Electronics,Hardware
570   Computers,Amazon Echo,Virtual Assistant Speake…  Electronics,Hardware
```

```
                  reviews.date  \
834   2017-12-29T16:56:05.000Z
1268  2017-12-29T16:56:05.000Z
1695  2018-09-01T19:51:34.000Z
1824  2018-06-06T20:46:55.000Z
2786  2017-11-30T21:40:30.000Z
2822  2018-08-16T23:06:42.000Z
2933  2018-01-06T15:03:52.000Z
3103  2018-04-06T23:51:32.000Z
3224  2017-12-30T18:26:19.000Z
3690  2018-04-06T23:51:32.000Z
222   2017-12-30T00:50:33.000Z
366   2017-12-21T00:19:38.000Z
570   2017-11-18T16:48:39.000Z

                                           reviews.text reviews.title  \
834   Best New Adult Toy in years! Wish I had purcha…           NaN
1268  Best New Adult Toy in years! Wish I had purcha…           NaN
1695  I bought the echo show for my mom for her birt…           NaN
1824  this is pretty cool, we love ours, we listen t…           NaN
2786  Really cool device! Instantly noticed the diff…           NaN
2822  I love the Echo show. I have found so many use…           NaN
2933  Awesome so far. Have used it as alarm clock, s…           NaN
3103  This was bought for a gift. But it looks nice…           NaN
3224  Delivered on time and it looked good will hook…           NaN
3690  This was bought for a gift. But it looks nice…           NaN
222   Absolutely Love the echo Show! It is in my kit…           NaN
366   Its a lot more then we expected.this is a wond…           NaN
570   I love it. It does so much and is so easy to u…           NaN

     sentiment
834   Positive
1268  Positive
1695  Positive
1824  Positive
2786  Positive
2822  Positive
2933  Positive
3103  Positive
3224  Positive
3690  Positive
222   Positive
366   Positive
570   Positive
```

[11]:
```
#Labelling the target variable
df['sentiment']=df['sentiment'].map({'Negative':0,'Neutral':1,'Positive':2})
```

[12]:
```
df.head()
```

[12]:
```
                                                name   brand  \
0  All-New Fire HD 8 Tablet, 8" HD Display, Wi-Fi…  Amazon
1        Amazon - Echo Plus w/ Built-In Hub - Silver  Amazon
2  Amazon Echo Show Alexa-enabled Bluetooth Speak…  Amazon
3  Fire HD 10 Tablet, 10.1 HD Display, Wi-Fi, 16 …  Amazon
4  Brand New Amazon Kindle Fire 16gb 7" Ips Displ…  Amazon

                                          categories  \
0  Electronics,iPad & Tablets,All Tablets,Fire Ta…
1  Amazon Echo,Smart Home,Networking,Home & Tools…
2  Amazon Echo,Virtual Assistant Speakers,Electro…
3  eBook Readers,Fire Tablets,Electronics Feature…
4  Computers/Tablets & Networking,Tablets & eBook…

            primaryCategories              reviews.date  \
0                 Electronics  2016-12-26T00:00:00.000Z
1        Electronics,Hardware  2018-01-17T00:00:00.000Z
2        Electronics,Hardware  2017-12-20T00:00:00.000Z
3  Office Supplies,Electronics  2017-08-04T00:00:00.000Z
4                 Electronics  2017-01-23T00:00:00.000Z

                                        reviews.text  \
0  Purchased on Black FridayPros - Great Price (e…
1  I purchased two Amazon in Echo Plus and two do…
2  Just an average Alexa option. Does show a few …
3  very good product. Exactly what I wanted, and …
4  This is the 3rd one I've purchased. I've bough…

            reviews.title  sentiment
0         Powerful tablet          2
1  Amazon Echo Plus AWESOME          2
2                 Average          1
3             Greattttttt          2
4           Very durable!          2
```

[13]:
```
#Remove the missing values
test_df.isnull().sum()
```

[13]:
```
name                 0
brand                0
categories           0
primaryCategories    0
reviews.date         0
reviews.text         0
reviews.title        3
```

```
dtype: int64
```

[14]:
```
#Since all the sentiment values are positive lets replace the NaN values with␣
↪good
df['reviews.title'].fillna('good',inplace=True)
```

[15]:
```
test_df.dropna(inplace=True)
```

[16]:
```
a=df['reviews.text'].values
a[:3]
```

[16]:
```
array(['Purchased on Black FridayPros - Great Price (even off sale)Very powerful
and fast with quad core processors Amazing soundWell builtCons -Amazon ads,
Amazon need this to subsidize the tablet and will remove the adds if you pay
them $15.Inability to access other apps except the ones from Amazon. There is a
way which I was able to accomplish to add the Google Play storeNet this is a
great tablet for the money',
       'I purchased two Amazon in Echo Plus and two dots plus four fire sticks
and the hub Philips hue for lamp for the family at Christmas 2017. I‚Äôm so
happy with these purchases and learning so much with Alexa. You can start your
daily routine with Alexa and program it to whatever you would like to include
news weather music horoscope ALSO you can start your day off with a compliment
and I think is very important. Alexa gave me the BEST CHILI RECIPE I MEAN THE
BEST it‚Äôs called Chili I. I want my husband to use Alexa to stay organized for
business dates and reminders. This is the way to go',
       'Just an average Alexa option. Does show a few things on screen but still
limited.'],
      dtype=object)
```

[17]:
```
b=test_df['reviews.text'].values
b[:3]
```

[17]:
```
array(['Amazon kindle fire has a lot of free app and can be used by any one that
wants to get online anywhere',
       'The Echo Show is a great addition to the Amazon family. Works just like
the Echo, but with a 7" screen. Bright vibrant display. Rich clear sound. Works
great with Arlo security cameras. Excellent smart home addition. Just hope
Google and Amazon start playing nice with each other soon so youtube will work
again.',
       'Great value from Best Buy. Bought at Christmas sale.'],
      dtype=object)
```

[18]:
```
import warnings
warnings.filterwarnings('ignore')
```

1 Convert the reviews in Tf-Idf score

[19]:
```python
import re
#remove punctaitions and special chracters
df['reviews.text'] = df['reviews.text'].str.replace('[^\w\s]','')
text=df['reviews.text'].values
```

[20]:
```python
len(text)
```

[20]:
```
5000
```

[21]:
```python
text[:2]
```

[21]:
```
array(['Purchased on Black FridayPros  Great Price even off saleVery powerful
and fast with quad core processors Amazing soundWell builtCons Amazon ads Amazon
need this to subsidize the tablet and will remove the adds if you pay them
15Inability to access other apps except the ones from Amazon There is a way
which I was able to accomplish to add the Google Play storeNet this is a great
tablet for the money',
       'I purchased two Amazon in Echo Plus and two dots plus four fire sticks
and the hub Philips hue for lamp for the family at Christmas 2017 IÄôm so happy
with these purchases and learning so much with Alexa You can start your daily
routine with Alexa and program it to whatever you would like to include news
weather music horoscope ALSO you can start your day off with a compliment and I
think is very important Alexa gave me the BEST CHILI RECIPE I MEAN THE BEST
itÄôs called Chili I I want my husband to use Alexa to stay organized for
business dates and reminders This is the way to go'],
      dtype=object)
```

[22]:
```python
text_lower=[i.lower() for i in text]
text_lower[:3]
```

[22]:
```
['purchased on black fridaypros  great price even off salevery powerful and fast
with quad core processors amazing soundwell builtcons amazon ads amazon need
this to subsidize the tablet and will remove the adds if you pay them
15inability to access other apps except the ones from amazon there is a way
which i was able to accomplish to add the google play storenet this is a great
tablet for the money',
 'i purchased two amazon in echo plus and two dots plus four fire sticks and the
hub philips hue for lamp for the family at christmas 2017 iäôm so happy with
these purchases and learning so much with alexa you can start your daily routine
with alexa and program it to whatever you would like to include news weather
music horoscope also you can start your day off with a compliment and i think is
very important alexa gave me the best chili recipe i mean the best itäôs called
chili i i want my husband to use alexa to stay organized for business dates and
reminders this is the way to go',
 'just an average alexa option does show a few things on screen but still
```

limited']

[23]:
```
from nltk.tokenize import word_tokenize
text_token=[word_tokenize(word) for word in text_lower]
print(text_token[:3])
```

```
[['purchased', 'on', 'black', 'fridaypros', 'great', 'price', 'even', 'off',
'salevery', 'powerful', 'and', 'fast', 'with', 'quad', 'core', 'processors',
'amazing', 'soundwell', 'builtcons', 'amazon', 'ads', 'amazon', 'need', 'this',
'to', 'subsidize', 'the', 'tablet', 'and', 'will', 'remove', 'the', 'adds',
'if', 'you', 'pay', 'them', '15inability', 'to', 'access', 'other', 'apps',
'except', 'the', 'ones', 'from', 'amazon', 'there', 'is', 'a', 'way', 'which',
'i', 'was', 'able', 'to', 'accomplish', 'to', 'add', 'the', 'google', 'play',
'storenet', 'this', 'is', 'a', 'great', 'tablet', 'for', 'the', 'money'], ['i',
'purchased', 'two', 'amazon', 'in', 'echo', 'plus', 'and', 'two', 'dots',
'plus', 'four', 'fire', 'sticks', 'and', 'the', 'hub', 'philips', 'hue', 'for',
'lamp', 'for', 'the', 'family', 'at', 'christmas', '2017', 'iäôm', 'so',
'happy', 'with', 'these', 'purchases', 'and', 'learning', 'so', 'much', 'with',
'alexa', 'you', 'can', 'start', 'your', 'daily', 'routine', 'with', 'alexa',
'and', 'program', 'it', 'to', 'whatever', 'you', 'would', 'like', 'to',
'include', 'news', 'weather', 'music', 'horoscope', 'also', 'you', 'can',
'start', 'your', 'day', 'off', 'with', 'a', 'compliment', 'and', 'i', 'think',
'is', 'very', 'important', 'alexa', 'gave', 'me', 'the', 'best', 'chili',
'recipe', 'i', 'mean', 'the', 'best', 'itäôs', 'called', 'chili', 'i', 'i',
'want', 'my', 'husband', 'to', 'use', 'alexa', 'to', 'stay', 'organized', 'for',
'business', 'dates', 'and', 'reminders', 'this', 'is', 'the', 'way', 'to',
'go'], ['just', 'an', 'average', 'alexa', 'option', 'does', 'show', 'a', 'few',
'things', 'on', 'screen', 'but', 'still', 'limited']]
```

[24]:
```
from nltk.corpus import stopwords
stop_nltk=stopwords.words('english')

def del_words(text):
    res=[word for word in text if word not in stop_nltk]
    return res
```

[25]:
```
#After removing stop words
text_nostop=[del_words(i) for i in text_token]
print(text_nostop[:3])
```

```
[['purchased', 'black', 'fridaypros', 'great', 'price', 'even', 'salevery',
'powerful', 'fast', 'quad', 'core', 'processors', 'amazing', 'soundwell',
'builtcons', 'amazon', 'ads', 'amazon', 'need', 'subsidize', 'tablet', 'remove',
'adds', 'pay', '15inability', 'access', 'apps', 'except', 'ones', 'amazon',
'way', 'able', 'accomplish', 'add', 'google', 'play', 'storenet', 'great',
'tablet', 'money'], ['purchased', 'two', 'amazon', 'echo', 'plus', 'two',
'dots', 'plus', 'four', 'fire', 'sticks', 'hub', 'philips', 'hue', 'lamp',
```

```
'family', 'christmas', '2017', 'iäôm', 'happy', 'purchases', 'learning', 'much',
'alexa', 'start', 'daily', 'routine', 'alexa', 'program', 'whatever', 'would',
'like', 'include', 'news', 'weather', 'music', 'horoscope', 'also', 'start',
'day', 'compliment', 'think', 'important', 'alexa', 'gave', 'best', 'chili',
'recipe', 'mean', 'best', 'itäôs', 'called', 'chili', 'want', 'husband', 'use',
'alexa', 'stay', 'organized', 'business', 'dates', 'reminders', 'way', 'go'],
['average', 'alexa', 'option', 'show', 'things', 'screen', 'still', 'limited']]
```

[26]:
```
text_str=[" ".join(i) for i in text_nostop]
text_str[:3]
```

[26]:
```
['purchased black fridaypros great price even salevery powerful fast quad core
processors amazing soundwell builtcons amazon ads amazon need subsidize tablet
remove adds pay 15inability access apps except ones amazon way able accomplish
add google play storenet great tablet money',
 'purchased two amazon echo plus two dots plus four fire sticks hub philips hue
lamp family christmas 2017 iäôm happy purchases learning much alexa start daily
routine alexa program whatever would like include news weather music horoscope
also start day compliment think important alexa gave best chili recipe mean best
itäôs called chili want husband use alexa stay organized business dates
reminders way go',
 'average alexa option show things screen still limited']
```

[27]:
```
from sklearn.feature_extraction.text import CountVectorizer
bow_transformer=CountVectorizer().fit(df['reviews.text'])
bow_transformer.vocabulary_
```

[27]:
```
{'purchased': 4211,
 'on': 3693,
 'black': 750,
 'fridaypros': 2252,
 'great': 2444,
 'price': 4089,
 'even': 1926,
 'off': 3668,
 'salevery': 4583,
 'powerful': 4041,
 'and': 415,
 'fast': 2062,
 'with': 5963,
 'quad': 4235,
 'core': 1317,
 'processors': 4130,
 'amazing': 386,
 'soundwell': 4952,
 'builtcons': 883,
 'amazon': 389,
```

```
'ads': 270,
'need': 3545,
'this': 5401,
'to': 5463,
'subsidize': 5136,
'the': 5354,
'tablet': 5237,
'will': 5941,
'remove': 4424,
'adds': 256,
'if': 2717,
'you': 6057,
'pay': 3847,
'them': 5361,
'15inability': 29,
'access': 189,
'other': 3746,
'apps': 486,
'except': 1951,
'ones': 3699,
'from': 2261,
'there': 5372,
'is': 2896,
'way': 5862,
'which': 5921,
'was': 5839,
'able': 175,
'accomplish': 203,
'add': 243,
'google': 2398,
'play': 3956,
'storenet': 5085,
'for': 2205,
'money': 3452,
'two': 5595,
'in': 2763,
'echo': 1776,
'plus': 3982,
'dots': 1684,
'four': 2232,
'fire': 2141,
'sticks': 5070,
'hub': 2682,
'philips': 3896,
'hue': 2685,
'lamp': 3052,
'family': 2045,
```

```
'at': 549,
'christmas': 1066,
'2017': 54,
'iäôm': 2943,
'so': 4890,
'happy': 2517,
'these': 5381,
'purchases': 4214,
'learning': 3094,
'much': 3494,
'alexa': 332,
'can': 942,
'start': 5034,
'your': 6066,
'daily': 1408,
'routine': 4550,
'program': 4155,
'it': 2902,
'whatever': 5912,
'would': 6014,
'like': 3155,
'include': 2770,
'news': 3575,
'weather': 5873,
'music': 3508,
'horoscope': 2656,
'also': 372,
'day': 1430,
'compliment': 1206,
'think': 5394,
'very': 5757,
'important': 2748,
'gave': 2310,
'me': 3336,
'best': 718,
'chili': 1055,
'recipe': 4336,
'mean': 3338,
'itäôs': 2938,
'called': 926,
'want': 5824,
'my': 3522,
'husband': 2696,
'use': 5701,
'stay': 5049,
'organized': 3738,
'business': 896,
```

'dates': 1423,
'reminders': 4419,
'go': 2364,
'just': 2977,
'an': 412,
'average': 588,
'option': 3728,
'does': 1660,
'show': 4773,
'few': 2102,
'things': 5393,
'screen': 4636,
'but': 899,
'still': 5073,
'limited': 3165,
'good': 2382,
'product': 4134,
'exactly': 1944,
'what': 5910,
'wanted': 5825,
'3rd': 85,
'one': 3696,
'ive': 2939,
'bought': 815,
'all': 347,
'of': 3666,
'nieces': 3588,
'no': 3595,
'case': 976,
'compares': 1181,
'has': 2530,
'held': 2575,
'protected': 4182,
'many': 3308,
'times': 5448,
'dropping': 1728,
'light': 3136,
'weight': 5889,
'wish': 5960,
'wifi': 5938,
'download': 1695,
'online': 3703,
'7year': 139,
'old': 3686,
'daughter': 1424,
'saved': 4601,
'up': 5669,

'by': 912,
'her': 2585,
'brother': 861,
'8gb': 152,
'about': 176,
'year': 6042,
'earlier': 1750,
'new': 3568,
'she': 4748,
'needed': 3546,
'more': 3466,
'space': 4956,
'os': 3745,
'bit': 748,
'clunky': 1124,
'less': 3109,
'intuitive': 2870,
'then': 5368,
'higher': 2601,
'priced': 4090,
'tablets': 5253,
'tough': 5506,
'beat': 673,
'speed': 4976,
'isnt': 2897,
'web': 5876,
'or': 3733,
'certain': 1002,
'gaming': 2307,
'fine': 2132,
'books': 787,
'parental': 3822,
'controls': 1291,
'are': 499,
'helpful': 2579,
'take': 5267,
'time': 5439,
'figure': 2114,
'out': 3754,
'require': 4458,
'steps': 5061,
'than': 5345,
'seem': 4684,
'necessary': 3544,
'change': 1015,
'overall': 3774,
'ok': 3680,

'first': 2150,
'kids': 3003,
'considering': 1257,
'adults': 274,
'ebooks': 1773,
'basic': 649,
'otherwise': 3748,
'better': 724,
'quality': 4236,
'efficiency': 1802,
'son': 4916,
'room': 4539,
'upgrade': 5679,
'memory': 3367,
'allow': 356,
'games': 2301,
'speakers': 4965,
'could': 1327,
'be': 670,
'located': 3205,
'position': 4027,
'mom': 3447,
'been': 692,
'wanting': 5827,
'get': 2327,
'ipad': 2885,
'myself': 3523,
'might': 3400,
'expensive': 1986,
'as': 520,
'well': 5895,
'cheap': 1038,
'not': 3616,
'run': 4556,
'chrome': 1069,
'stuff': 5120,
'returned': 4505,
'store': 5083,
'got': 2403,
'replace': 4437,
'sons': 4924,
'love': 3249,
'adultchild': 273,
'profile': 4151,
'ability': 173,
'have': 2537,
'replacement': 4440,

```
'warranty': 5836,
'came': 934,
'handy': 2508,
'make': 3291,
'sure': 5184,
'monitor': 3456,
'prevent': 4085,
'unwanted': 5667,
'app': 459,
'choice': 1057,
'colors': 1145,
'kindles': 3020,
'own': 3783,
'gift': 2332,
'replaces': 4441,
'kindle': 3014,
'hd': 2545,
'we': 5865,
'childrens': 1052,
'aging': 305,
'ipads': 2886,
'theyre': 5385,
'purchase': 4210,
'fact': 2026,
'tv': 5583,
'movies': 3486,
'watch': 5850,
'offline': 3675,
'easy': 1764,
'set': 4725,
'surprisingly': 5197,
'useful': 5705,
'ben': 711,
'gen': 2318,
'definitely': 1475,
'recommend': 4349,
'before': 694,
'seams': 4659,
'work': 5988,
'model': 3439,
'only': 3704,
'because': 680,
'sale': 4577,
'built': 882,
'that': 5350,
'strip': 5105,
'lights': 3148,
```

```
'andi': 416,
'didnäôt': 1553,
'say': 4608,
'sounds': 4950,
'baby': 608,
'talk': 5271,
'melol': 3360,
'little': 3190,
'hesitant': 2593,
'spend': 4982,
'am': 380,
'glad': 2352,
'did': 1551,
'always': 378,
'answers': 436,
'when': 5916,
'commands': 1161,
'noticed': 3629,
'probably': 4121,
'95': 160,
'right': 4524,
'any': 442,
'command': 1160,
'given': 2349,
'see': 4681,
'some': 4903,
'briefings': 846,
'picturevideo': 3926,
'thats': 5352,
'nice': 3583,
'too': 5481,
'video': 5763,
'call': 925,
'planning': 3951,
'getting': 2329,
'grandma': 2428,
'who': 5929,
'technologically': 5302,
'advanced': 277,
'sit': 4826,
'table': 5235,
'grandkids': 2425,
'worth': 6010,
'every': 1933,
'penny': 3859,
'exceed': 1947,
'expectations': 1980,
```

'control': 1285,
'they': 5383,
'item': 2905,
'those': 5408,
'prime': 4104,
'almost': 364,
'must': 3518,
'keeps': 2984,
'place': 3942,
'small': 4871,
'enough': 1874,
'carry': 970,
'purse': 4222,
'open': 3712,
'whole': 5930,
'world': 5998,
'while': 5923,
'waiting': 5810,
'appointments': 477,
'listen': 3180,
'sign': 4792,
'unlimited': 5650,
'pandora': 3809,
'device': 1536,
'home': 2630,
'enjoys': 1871,
'its': 2927,
'picture': 3924,
'sound': 4942,
'easier': 1758,
'viewing': 5774,
'adjustment': 263,
'fit': 2153,
'buy': 905,
'again': 295,
'grand': 2414,
'picked': 3917,
'network': 3564,
'short': 4766,
'order': 3734,
'idea': 2711,
'implemented': 2745,
'setup': 4731,
'enable': 1841,
'voice': 5795,
'ios': 2883,
'unclear': 5618,

'connected': 1239,
'equipment': 1899,
'wife': 5936,
'our': 3751,
'kitchen': 3026,
'thing': 5392,
'different': 1563,
'communicate': 1168,
'directly': 1590,
'facetime': 2021,
'rather': 4273,
'having': 2543,
'someone': 4907,
'deal': 1444,
'breaker': 836,
'means': 3340,
'though': 5409,
'alot': 367,
'instead': 2830,
'running': 4559,
'dog': 1666,
'somewhere': 4915,
'lot': 3240,
'awesome': 596,
'needs': 3551,
'internet': 2861,
'streaming': 5097,
'dont': 1675,
'everything': 1938,
'however': 2677,
'within': 5964,
'updated': 5675,
'often': 3677,
'happened': 2513,
'top': 5490,
'above': 177,
'mini': 3411,
'now': 3635,
'gives': 2350,
'89': 150,
'worked': 5990,
'expected': 1981,
'highly': 2605,
'looks': 3229,
'quick': 4246,
'specifically': 4972,
'aspect': 530,

```
'life': 3130,
'performs': 3876,
'pleasantly': 3970,
'surprised': 5194,
'performance': 3872,
'amazons': 396,
'help': 2577,
'rutine': 4562,
'asking': 526,
'simple': 4806,
'specific': 4971,
'flash': 2161,
'traffic': 5522,
'commute': 1171,
'tool': 5483,
'tech': 5295,
'enthusiast': 1885,
'meet': 3354,
'stylus': 5129,
'pen': 3858,
'same': 4586,
'storage': 5082,
'mine': 3408,
'reading': 4292,
'surfing': 5188,
'shows': 4777,
'dark': 1416,
'reason': 4311,
'hulu': 2689,
'netflix': 3562,
'wonderful': 5976,
'comes': 1154,
'free': 2239,
'fantastic': 2052,
'office': 3673,
'personal': 3887,
'assistant': 537,
'totally': 5499,
'friendly': 2254,
'cost': 1322,
'investment': 2877,
'tap': 5278,
'essentially': 1915,
'smaller': 4872,
'version': 5754,
'capability': 949,
'doubles': 1688,
```

'portable': 4021,
'bluetooth': 768,
'speaker': 4961,
'do': 1652,
'ask': 523,
'anything': 448,
'works': 5995,
'prefer': 4057,
'college': 1138,
'students': 5117,
'supplement': 5173,
'their': 5359,
'says': 4612,
'children': 1050,
'therapy': 5371,
'practice': 4047,
'taking': 5270,
'awhile': 600,
'how': 2676,
'organize': 3737,
'took': 5482,
'over': 3772,
'approve': 484,
'such': 5143,
'charging': 1031,
'account': 205,
'registering': 4383,
'settings': 4728,
'beautiful': 677,
'camera': 936,
'photos': 3907,
'pixelated': 3939,
'inexpensive': 2799,
'him': 2611,
'learn': 3092,
'step': 5057,
'nabi': 3525,
'he': 2554,
'thrilled': 5416,
'skype': 4844,
'already': 370,
'battery': 662,
'outstanding': 3770,
'really': 4308,
'book': 780,
'worry': 6006,
'size': 4834,

```
'wasnt': 5844,
'something': 4909,
'friday': 2251,
'thought': 5410,
'why': 5933,
'soon': 4928,
'understood': 5630,
'enhances': 1864,
'being': 703,
'seeing': 4682,
'adjusting': 262,
'shopping': 4765,
'list': 3179,
'touch': 5501,
'walking': 5817,
'blurbs': 771,
'sleek': 4846,
'counter': 1332,
'anyone': 445,
'older': 3687,
'absolutely': 180,
'loves': 3252,
'features': 2081,
'market': 3314,
'value': 5733,
'incredible': 2781,
'questions': 4244,
'timers': 5446,
'etc': 1917,
'heart': 2564,
'1000s': 3,
'recommendation': 4350,
'simarly': 4804,
'screenless': 4641,
'auto': 575,
'entertainment': 1882,
'friends': 2258,
'knowledge': 3034,
'jokes': 2962,
'tons': 5478,
'fun': 2275,
'weekend': 5882,
'sorcerers': 4934,
'box': 819,
'displays': 1630,
'magical': 3280,
'pictures': 3925,
```

```
'bright': 848,
'pretty': 4083,
'kid': 2996,
'ecosystem': 1789,
'nothing': 3625,
'spectacular': 4974,
'majorly': 3290,
'wrong': 6027,
'biggest': 736,
'flaw': 2165,
'geared': 2313,
'found': 2231,
'searching': 4663,
'lock': 3208,
'playing': 3962,
'child': 1048,
'hits': 2620,
'lots': 3242,
'helping': 2581,
'back': 611,
'into': 2865,
'movie': 3484,
'relaxing': 4399,
'guided': 2478,
'activated': 227,
'commercials': 1166,
'correct': 1319,
'multiple': 3500,
'devices': 1540,
'household': 2673,
'comparable': 1178,
'competitors': 1190,
'geeks': 2316,
'squad': 5011,
'monday': 3451,
'april': 494,
'goes': 2371,
'along': 366,
'perfect': 3866,
'had': 2489,
'tabletwhy': 5259,
'bigger': 735,
'pick': 3916,
'weary': 5872,
'hang': 2509,
'makes': 3292,
'member': 3361,
```

'enjoy': 1865,
'options': 3731,
'exploring': 1998,
'skills': 4840,
'alarm': 324,
'informs': 2805,
'todays': 5467,
'forcast': 2208,
'local': 3203,
'events': 1929,
'decent': 1453,
'mindless': 3406,
'another': 432,
'accessory': 195,
'house': 2670,
'uses': 5716,
'recipes': 4337,
'cooking': 1307,
'making': 3293,
'grocery': 2458,
'lists': 3184,
'listens': 3183,
'crisp': 1371,
'since': 4813,
'using': 5721,
'washed': 5841,
'twice': 5591,
'throw': 5421,
'ticking': 5430,
'80': 141,
'automated': 578,
'alexawe': 340,
'each': 1749,
'ereader': 1903,
'pricy': 4100,
'specs': 4973,
'voyage': 5800,
'far': 2054,
'pixels': 3940,
'procesor': 4126,
'costs': 1325,
'difference': 1560,
'between': 727,
'lighter': 3140,
'brighter': 850,
'leds': 3102,
'longer': 3221,

```
'youre': 6067,
'cover': 1345,
'drains': 1706,
'faster': 2065,
'without': 5965,
'youll': 6061,
'read': 4283,
'hours': 2668,
'water': 5856,
'resistant': 4473,
'hence': 2583,
'name': 3527,
'plan': 3948,
'raining': 4261,
'outside': 3769,
'bath': 657,
'currently': 1391,
'offer': 3669,
'mind': 3405,
'tag': 5265,
'bad': 626,
'both': 809,
'numerous': 3643,
'educational': 1798,
'available': 585,
'10': 0,
'yr': 6075,
'granddaughter': 2420,
'wants': 5829,
'unable': 5610,
'find': 2129,
'instore': 2832,
'excited': 1957,
'delivered': 1488,
'simply': 4812,
'easily': 1760,
'miss': 3428,
'5yo': 111,
'facebook': 2020,
'shop': 4764,
'search': 4660,
'brand': 829,
'tasks': 5287,
'complaints': 1196,
'left': 3103,
'plane': 3949,
'return': 4504,
```

'50': 100,
'days': 1437,
'later': 3074,
'due': 1737,
'years': 6045,
'newer': 3570,
'technology': 5304,
'side': 4787,
'connects': 1247,
'mobil': 3435,
'phone': 3901,
'people': 3860,
'chose': 1064,
'disappointed': 1600,
'86': 147,
'father': 2069,
'finds': 2131,
'figuring': 2117,
'cant': 945,
'external': 2007,
'pdfs': 3852,
'properly': 4176,
'either': 1812,
'zooms': 6083,
'provides': 4197,
'experience': 1987,
'install': 2821,
'loved': 3250,
'quieter': 4250,
'ever': 1931,
'looking': 3227,
'hands': 2505,
'tied': 5432,
'äúdropping': 6092,
'inäù': 2882,
'likenit': 3158,
'cradle': 1355,
'charge': 1024,
'user': 5712,
'big': 734,
'large': 3060,
'unfair': 5634,
'rate': 4272,
'cause': 989,
'taken': 5268,
'homework': 2639,
'worthless': 6011,

```
'regular': 4386,
'poor': 4008,
'excuse': 1964,
'chat': 1034,
'bathroom': 658,
'pool': 4006,
'places': 3945,
'where': 5918,
'may': 3333,
'info': 2802,
'newest': 3571,
'hottest': 2666,
'skeptical': 4838,
'project': 4163,
'videos': 5767,
'f450': 2018,
'drone': 1723,
'youtube': 6071,
'wont': 5981,
'sell': 4700,
'products': 4143,
'further': 2288,
'testing': 5332,
'iphone': 2889,
'android': 418,
'most': 3470,
'functions': 2284,
'echos': 1781,
'perform': 3871,
'unless': 5648,
'calls': 928,
'check': 1044,
'move': 3481,
'feet': 2096,
'outlet': 3761,
'hugh': 2688,
'expense': 1985,
'others': 3747,
'opened': 3713,
'know': 3032,
'employees': 1839,
'needespecially': 3548,
'jissela': 2956,
'dot': 1683,
'clear': 1096,
'were': 5903,
'huge': 2686,
```

'teaching': 5292,
'turned': 5578,
'advertising': 285,
'browser': 866,
'turn': 5577,
'determine': 1531,
'downloaded': 1696,
'enjoyed': 1867,
'certainly': 1003,
'fiance': 2104,
'enjoyes': 1868,
'excellent': 1950,
'changes': 1018,
'form': 2220,
'adore': 269,
'flatness': 2164,
'brightness': 853,
'daylight': 1434,
'imo': 2741,
'actually': 235,
'yes': 6050,
'operate': 3717,
'wirelessit': 5955,
'operates': 3718,
'handsäyou': 2507,
'speak': 4960,
'microphone': 3391,
'button': 901,
'hold': 2624,
'down': 1691,
'rechargeable': 4331,
'base': 646,
'unit': 5641,
'wonders': 5979,
'didnt': 1552,
'browsing': 867,
'fancy': 2050,
'fits': 2154,
'perfectly': 3869,
'pricepff': 4092,
'cheaper': 1040,
'2nd': 67,
'owned': 3784,
'present': 4075,
'65': 117,
'issues': 2901,
'functionality': 2281,

'approved': 485,
'content': 1275,
'elderly': 1813,
'xmass': 6035,
'electronic': 1815,
'competent': 1187,
'pleased': 3972,
'grandsons': 2435,
'together': 5472,
'send': 4704,
'messages': 3376,
'grandparents': 2432,
'once': 3694,
'besti': 722,
'cameras': 937,
'extra': 2009,
'last': 3064,
'nephew': 3557,
'stick': 5068,
'meantime': 3342,
'likes': 3159,
'gadgets': 2293,
'knew': 3029,
'august': 570,
'charger': 1027,
'connection': 1241,
'loose': 3232,
'exchange': 1954,
'problem': 4122,
'going': 2376,
'brothers': 862,
'sister': 4823,
'rethinking': 4500,
'let': 3113,
'walk': 5815,
'spoken': 4994,
'word': 5985,
'equally': 1897,
'impressive': 2755,
'gradually': 2412,
'arenäôt': 506,
'blinded': 758,
'feels': 2094,
'affordable': 290,
'class': 1087,
'grandson': 2433,
'monitoring': 3457,

```
'dad': 1406,
'super': 5167,
'living': 3195,
'trying': 5569,
'switch': 5209,
'fastcapable': 2063,
'previous': 4087,
'models': 3441,
'alternative': 375,
'should': 4770,
'lose': 3235,
'break': 834,
'pocket': 3991,
'ordering': 3736,
'today': 5466,
'keep': 2982,
'buying': 907,
'improvement': 2758,
'feature': 2080,
'bringing': 856,
'us': 5695,
'rundown': 4557,
'morning': 3468,
'saying': 4609,
'made': 3276,
'car': 957,
'12': 15,
'thick': 5389,
'rubber': 4552,
'plays': 3965,
'long': 3220,
'mix': 3433,
'items': 2907,
'cord': 1316,
'pull': 4202,
'port': 4018,
'angle': 420,
'mow': 3491,
'yard': 6040,
'gadget': 2292,
'songs': 4920,
'various': 5739,
'reluctant': 4407,
'iäôve': 2944,
'package': 3792,
'phillips': 3898,
'bulb': 885,
```

```
'amazed': 382,
'addictive': 248,
'pairs': 3807,
'becomes': 682,
'indispensable': 2790,
'availability': 584,
'three': 5414,
'niece': 3587,
'protect': 4181,
'smart': 4875,
'tvs': 5587,
'firesticks': 2147,
'confusing': 1233,
'itsafe': 2928,
'sd': 4653,
'card': 958,
'holding': 2626,
'playimg': 3961,
'past': 3840,
'defective': 1469,
'bent': 715,
'hard': 2521,
'possible': 4033,
'color': 1139,
'16gb': 33,
'pics': 3921,
'substitute': 5139,
'minimal': 3413,
'usage': 5697,
'never': 3567,
'goingit': 2377,
'text': 5334,
'fonts': 2200,
'buttons': 903,
'pages': 3800,
'flip': 2174,
'compared': 1180,
'tapping': 5282,
'positive': 4030,
'reviews': 4514,
…}
```

[28]:
```
text_bow=bow_transformer.transform(df['reviews.text'])
print(text_bow)
```

```
(0, 29)       1
(0, 175)      1
```

```
(0, 189)      1
(0, 203)      1
(0, 243)      1
(0, 256)      1
(0, 270)      1
(0, 386)      1
(0, 389)      3
(0, 415)      2
(0, 486)      1
(0, 750)      1
(0, 883)      1
(0, 1317)     1
(0, 1926)     1
(0, 1951)     1
(0, 2062)     1
(0, 2205)     1
(0, 2252)     1
(0, 2261)     1
(0, 2398)     1
(0, 2444)     2
(0, 2717)     1
(0, 2896)     2
(0, 3452)     1
:     :
(4998, 4720)  1
(4998, 5350)  1
(4998, 5383)  1
(4998, 5463)  1
(4998, 5473)  1
(4998, 5839)  2
(4998, 5881)  1
(4998, 5963)  1
(4999, 415)   1
(4999, 520)   1
(4999, 1764)  1
(4999, 2150)  1
(4999, 2205)  2
(4999, 2332)  1
(4999, 2585)  2
(4999, 2902)  1
(4999, 3252)  1
(4999, 3472)  1
(4999, 3522)  1
(4999, 3535)  1
(4999, 4211)  1
(4999, 4748)  1
(4999, 5237)  1
(4999, 5401)  1
```

```
  (4999, 5463)  1
```

[29]:
```
#Applying TfIdf on the data
from sklearn.feature_extraction.text import TfidfVectorizer
vec=TfidfVectorizer(max_features=3000)
text_tfidf=vec.fit_transform(text_str)
```

[30]:
```
text_tfidf.shape
```

[30]: (5000, 3000)

[31]:
```
print(text_tfidf.shape)
print(text_tfidf)
```

```
(5000, 3000)
  (0, 1628)     0.16731758650340947
  (0, 1900)     0.12817868846808744
  (0, 1115)     0.16589213902602223
  (0, 128)      0.1789530847024756
  (0, 93)       0.14097607252511044
  (0, 2860)     0.16072917642173107
  (0, 1755)     0.20484950295660792
  (0, 894)      0.22014769401177406
  (0, 253)      0.12535884609674772
  (0, 98)       0.17327876925976912
  (0, 1840)     0.20065741877487597
  (0, 136)      0.23285792485716897
  (0, 2171)     0.24164528075897848
  (0, 2590)     0.17192461726925
  (0, 1670)     0.14502476492375543
  (0, 144)      0.21374868854604903
  (0, 200)      0.31746813935602636
  (0, 198)      0.16043145925851013
  (0, 643)      0.2723345698923521
  (0, 2063)     0.2723345698923521
  (0, 947)      0.16731758650340947
  (0, 1954)     0.23037582948569038
  (0, 877)      0.148993064573648
  (0, 1979)     0.11520744137879667
  (0, 1135)     0.15367223257563953
  :     :
  (4998, 40)    0.27214996467608527
  (4998, 1863)  0.23944139247708154
  (4998, 2870)  0.2406975694541218
  (4998, 2347)  0.23061243864223446
  (4998, 2270)  0.25082389365997887
  (4998, 2268)  0.23161416939616541
  (4998, 2681)  0.2644679390107173
```

```
  (4998, 1376)	0.32171709914943203
  (4998, 564)	0.21201615691963138
  (4998, 317)	0.1959734225450786
  (4998, 1044)	0.2326419220197183
  (4998, 787)	0.1762596627856057
  (4998, 1311)	0.19435824924993722
  (4998, 1082)	0.14570598215571237
  (4998, 651)	0.1821922323383668
  (4998, 1979)	0.13609804974618306
  (4998, 391)	0.218012418278566
  (4999, 1662)	0.4697280840242521
  (4999, 1638)	0.4931288017886686
  (4999, 1512)	0.2839168599669272
  (4999, 821)	0.24583516898194904
  (4999, 1086)	0.3251686289542549
  (4999, 990)	0.3732931174329456
  (4999, 2590)	0.21807249219454117
  (4999, 2051)	0.32015277230219186
```

[33]:
```python
from sklearn.naive_bayes import MultinomialNB
mnb=MultinomialNB()
model=mnb.fit(text_tfidf,df['sentiment'])
```

[34]:
```python
pred=model.predict(text_tfidf)
print(pred)
```

```
[2 2 2 … 2 2 2]
```

[35]:
```python
from sklearn.metrics import␣
 ↪confusion_matrix,classification_report,roc_auc_score,roc_curve
co_mat=confusion_matrix(df['sentiment'],pred)
print(co_mat)
```

```
[[   8    0  109]
 [   0    0  197]
 [   0    0 4686]]
```

[36]:
```python
from sklearn.model_selection import train_test_split
xtrain,xtest,ytrain,ytest=train_test_split(text_tfidf,df['sentiment'],test_size=0.
 ↪2,random_state=42)
```

[37]:
```python
print(xtrain.shape,ytrain.shape)
print(xtest.shape,ytest.shape)
```

```
(4000, 3000) (4000,)
(1000, 3000) (1000,)
```

[38]:
```python
mnb.fit(xtrain,ytrain)
```

[38]: MultinomialNB()

[39]:
```
y_pred=mnb.predict(xtest)
y_predicted=pd.Series(y_pred)
y_predicted.value_counts()
```

[39]:
```
2    999
0      1
dtype: int64
```

[40]:
```
confusion_matrix(ytest,y_pred)
```

[40]:
```
array([[  1,   0,  20],
       [  0,   0,  57],
       [  0,   0, 922]])
```

[41]:
```
print(classification_report(ytest,y_pred))
```

```
              precision    recall  f1-score   support

           0       1.00      0.05      0.09        21
           1       0.00      0.00      0.00        57
           2       0.92      1.00      0.96       922

    accuracy                           0.92      1000
   macro avg       0.64      0.35      0.35      1000
weighted avg       0.87      0.92      0.89      1000
```

2 Tackling Class Imbalance Problem:

[42]:
```
#Oversampling or undersampling can be used to tackle the class imbalance problem
from imblearn.over_sampling import SMOTE
smote=SMOTE()
```

[43]:
```
xtrain_smote,ytrain_smote=smote.fit_resample(xtrain,ytrain)
```

[44]:
```
from collections import Counter
print('Before smote:',Counter(ytrain))
print('After smote:',Counter(ytrain_smote))
```

```
Before smote: Counter({2: 3764, 1: 140, 0: 96})
After smote: Counter({2: 3764, 1: 3764, 0: 3764})
```

[45]:
```
#Now trying the classifier
mnb.fit(xtrain_smote,ytrain_smote)
```

[45]: MultinomialNB()

[46]:
```
y_pred=mnb.predict(xtest)
y_predicted=pd.Series(y_pred)
y_predicted.value_counts()
```

[46]:
```
2    849
1     95
0     56
dtype: int64
```

[47]:
```
#Check the performance metrics
confusion_matrix(ytest,y_pred)
```

[47]:
```
array([[ 11,   5,   5],
       [  8,  28,  21],
       [ 37,  62, 823]])
```

[48]:
```
print(classification_report(ytest,y_pred))
```

```
              precision    recall  f1-score   support

           0       0.20      0.52      0.29        21
           1       0.29      0.49      0.37        57
           2       0.97      0.89      0.93       922

    accuracy                           0.86      1000
   macro avg       0.49      0.64      0.53      1000
weighted avg       0.91      0.86      0.88      1000
```

[49]:
```
ypred_prob=mnb.predict_proba(xtest)
ypred_prob
```

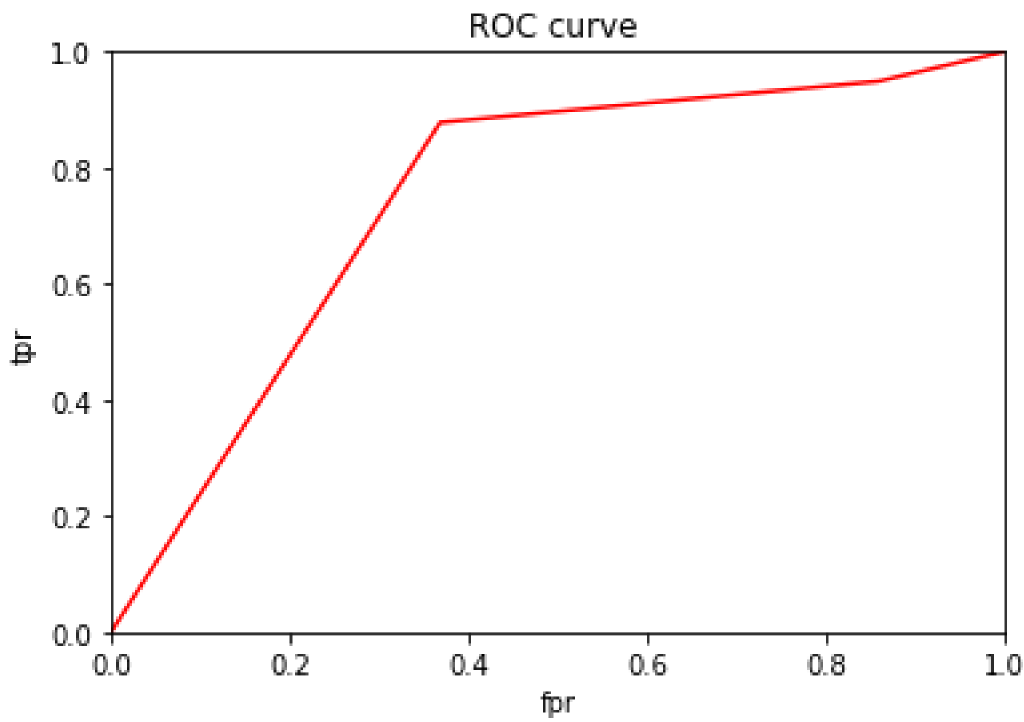
[49]:
```
array([[0.09559089, 0.2258333 , 0.67857581],
       [0.34486267, 0.32629888, 0.32883845],
       [0.01467373, 0.08400542, 0.90132085],
       …,
       [0.01008611, 0.13117393, 0.85873996],
       [0.00299987, 0.0275022 , 0.96949793],
       [0.01746939, 0.04267903, 0.93985158]])
```

[50]:
```
tpr,fpr,thresholds=roc_curve(ytest,y_pred,pos_label=1)
```

[51]:
```
print(tpr)
print(fpr)
print(thresholds)
```

```
[0.         0.87804878 0.94909862 1.        ]
[0.         0.36842105 0.85964912 1.        ]
[3 2 1 0]
```

[52]:
```
import matplotlib.pyplot as plt
%matplotlib inline
plt.xlim([0.0,1.0])
plt.ylim([0.0,1.0])
plt.title('ROC curve ')
plt.xlabel('fpr')
plt.ylabel('tpr')
plt.plot(fpr,tpr,color='red')
plt.show()
```



[53]:
```
roc_auc_score(ytest,ypred_prob,multi_class='ovr')
```

[53]: 0.8315592348320351

[54]:
```
#Use Tree-based classifiers like Random Forest and XGBoost
from sklearn.ensemble import RandomForestClassifier
```

[55]:
```
rfclf=RandomForestClassifier()
rfclf.fit(xtrain,ytrain)
```

[55]: RandomForestClassifier()

[56]:
```
ypred_rfclf=rfclf.predict(xtest)
ypred_rfclf1=pd.Series(ypred_rfclf)
ypred_rfclf1.value_counts()
```

[56]:
```
2    984
1     11
0      5
dtype: int64
```

[60]:
```
confusion_matrix (ytest,ypred_rfclf)
```

[60]:
```
array([[  5,   0,  16],
       [  0,  11,  46],
       [  0,   0, 922]])
```

[61]:
```
print(classification_report(ytest,ypred_rfclf))
```

```
              precision    recall  f1-score   support

           0       1.00      0.24      0.38        21
           1       1.00      0.19      0.32        57
           2       0.94      1.00      0.97       922

    accuracy                           0.94      1000
   macro avg       0.98      0.48      0.56      1000
weighted avg       0.94      0.94      0.92      1000
```

[62]:
```
rfclf.fit(xtrain_smote,ytrain_smote)
ypred_rfclf_smote=rfclf.predict(xtest)
confusion_matrix(ytest,ypred_rfclf_smote)
```

[62]:
```
array([[  5,   0,  16],
       [  0,  12,  45],
       [  0,   1, 921]])
```

[63]:
```
print(classification_report(ytest,ypred_rfclf_smote))
```

```
              precision    recall  f1-score   support

           0       1.00      0.24      0.38        21
           1       0.92      0.21      0.34        57
           2       0.94      1.00      0.97       922

    accuracy                           0.94      1000
   macro avg       0.95      0.48      0.56      1000
```

```
weighted avg       0.94      0.94      0.92      1000
```

[64]:
```
#Apply XG Boost
import xgboost as xg
model=xg.XGBClassifier()
```

[65]:
```
model.fit(xtrain_smote,ytrain_smote)
yxg_pred=model.predict(xtest)
```

[66]:
```
confusion_matrix(ytest,yxg_pred)
```

[66]:
```
array([[  6,   2,  13],
       [  4,  13,  40],
       [  5,  17, 900]])
```

[67]:
```
confusion_matrix(ytest,yxg_pred)
```

[67]:
```
array([[  6,   2,  13],
       [  4,  13,  40],
       [  5,  17, 900]])
```

[68]:
```
print(classification_report(ytest,yxg_pred))
```

```
              precision    recall  f1-score   support

           0       0.40      0.29      0.33        21
           1       0.41      0.23      0.29        57
           2       0.94      0.98      0.96       922

    accuracy                           0.92      1000
   macro avg       0.58      0.50      0.53      1000
weighted avg       0.90      0.92      0.91      1000
```

3 Model Selection

[69]:
```
#Apply multi-class SVM's and neural nets
from sklearn.metrics import accuracy_score
from sklearn.svm import SVC
svc=SVC()
```

[70]:
```
svc.fit(xtrain_smote,ytrain_smote)
```

[70]:
```
SVC()
```

[71]:
```
ypred_svc=svc.predict(xtest)
```

[72]:
```
accuracy_score(ytest,ypred_svc)
```

[72]: 0.937

[73]:
```
print(classification_report(ytest,ypred_svc))
```

```
              precision    recall  f1-score   support

           0       1.00      0.24      0.38        21
           1       0.92      0.19      0.32        57
           2       0.94      1.00      0.97       922

    accuracy                           0.94      1000
   macro avg       0.95      0.48      0.56      1000
weighted avg       0.94      0.94      0.92      1000
```

[74]:
```
#Use possible ensemble techniques like: XGboost + oversampled_multinomial_NB
import xgboost as xg
from sklearn.naive_bayes import GaussianNB
nb=GaussianNB()
model=xg.XGBClassifier(base_estimator=GaussianNB())
model.fit(xtrain_smote,ytrain_smote)
```

[74]:
```
XGBClassifier(base_estimator=GaussianNB(), base_score=0.5, booster=None,
              colsample_bylevel=1, colsample_bynode=1, colsample_bytree=1,
              gamma=0, gpu_id=-1, importance_type='gain',
              interaction_constraints=None, learning_rate=0.300000012,
              max_delta_step=0, max_depth=6, min_child_weight=1, missing=nan,
              monotone_constraints=None, n_estimators=100, n_jobs=0,
              num_parallel_tree=1, objective='multi:softprob', random_state=0,
              reg_alpha=0, reg_lambda=1, scale_pos_weight=None, subsample=1,
              tree_method=None, validate_parameters=False, verbosity=None)
```

[75]:
```
ypred_xg_nb=model.predict(xtest)
```

[76]:
```
accuracy_score(ytest,ypred_xg_nb)
```

[76]: 0.919

[77]:
```
print(classification_report(ytest,ypred_xg_nb))
```

```
              precision    recall  f1-score   support

           0       0.40      0.29      0.33        21
           1       0.41      0.23      0.29        57
           2       0.94      0.98      0.96       922
```

```
    accuracy                           0.92      1000
   macro avg       0.58      0.50      0.53      1000
weighted avg       0.90      0.92      0.91      1000
```

4 Assign a score to the sentence sentiment (sentiment score). Use this engineered feature in the model and check for improvements

[79]:
```python
from textblob import TextBlob
def getTextPolarity(txt):
    return TextBlob(txt).sentiment.polarity
```

[80]:
```python
df['Polarity'] = df['reviews.text'].apply(getTextPolarity)
```

[81]:
```python
df.head()
```

[81]:
```
                                                name   brand  \
0  All-New Fire HD 8 Tablet, 8" HD Display, Wi-Fi…  Amazon
1        Amazon - Echo Plus w/ Built-In Hub - Silver  Amazon
2  Amazon Echo Show Alexa-enabled Bluetooth Speak…  Amazon
3  Fire HD 10 Tablet, 10.1 HD Display, Wi-Fi, 16 …  Amazon
4  Brand New Amazon Kindle Fire 16gb 7" Ips Displ…  Amazon

                                          categories  \
0  Electronics,iPad & Tablets,All Tablets,Fire Ta…
1  Amazon Echo,Smart Home,Networking,Home & Tools…
2  Amazon Echo,Virtual Assistant Speakers,Electro…
3  eBook Readers,Fire Tablets,Electronics Feature…
4  Computers/Tablets & Networking,Tablets & eBook…

            primaryCategories              reviews.date  \
0                 Electronics  2016-12-26T00:00:00.000Z
1        Electronics,Hardware  2018-01-17T00:00:00.000Z
2        Electronics,Hardware  2017-12-20T00:00:00.000Z
3  Office Supplies,Electronics  2017-08-04T00:00:00.000Z
4                 Electronics  2017-01-23T00:00:00.000Z

                                        reviews.text  \
0  Purchased on Black FridayPros  Great Price eve…
1  I purchased two Amazon in Echo Plus and two do…
2  Just an average Alexa option Does show a few t…
3  very good product Exactly what I wanted and a …
4  This is the 3rd one Ive purchased Ive bought o…

          reviews.title  sentiment  Polarity
0        Powerful tablet          2  0.363542
```

```
1  Amazon Echo Plus AWESOME          2  0.458214
2                    Average         1 -0.140476
3                Greattttttt         2  0.690000
4              Very durable!         2  0.187500
```

[83]:
```
def getTextAnalysis(a):
    if a < 0:
        return "Negative"
    elif a == 0:
        return "Neutral"
    else:
        return "Positive"
df['Score'] = df['Polarity'].apply(getTextAnalysis)
```

[84]:
```
df.head()
```

[84]:
```
                                                name   brand  \
0  All-New Fire HD 8 Tablet, 8" HD Display, Wi-Fi…  Amazon
1        Amazon - Echo Plus w/ Built-In Hub - Silver  Amazon
2  Amazon Echo Show Alexa-enabled Bluetooth Speak…  Amazon
3  Fire HD 10 Tablet, 10.1 HD Display, Wi-Fi, 16 …  Amazon
4  Brand New Amazon Kindle Fire 16gb 7" Ips Displ…  Amazon

                                          categories  \
0  Electronics,iPad & Tablets,All Tablets,Fire Ta…
1  Amazon Echo,Smart Home,Networking,Home & Tools…
2  Amazon Echo,Virtual Assistant Speakers,Electro…
3  eBook Readers,Fire Tablets,Electronics Feature…
4  Computers/Tablets & Networking,Tablets & eBook…

             primaryCategories              reviews.date  \
0                  Electronics  2016-12-26T00:00:00.000Z
1         Electronics,Hardware  2018-01-17T00:00:00.000Z
2         Electronics,Hardware  2017-12-20T00:00:00.000Z
3  Office Supplies,Electronics  2017-08-04T00:00:00.000Z
4                  Electronics  2017-01-23T00:00:00.000Z

                                        reviews.text  \
0  Purchased on Black FridayPros  Great Price eve…
1  I purchased two Amazon in Echo Plus and two do…
2  Just an average Alexa option Does show a few t…
3  very good product Exactly what I wanted and a …
4  This is the 3rd one Ive purchased Ive bought o…

              reviews.title  sentiment  Polarity     Score
0           Powerful tablet          2  0.363542  Positive
1  Amazon Echo Plus AWESOME          2  0.458214  Positive
```

```
2                    Average          1 -0.140476  Negative
3                 Greatttttttt        2  0.690000  Positive
4               Very durable!         2  0.187500  Positive
```

[85]:
```
#Labelling the target variable
df['Score']=df['Score'].map({'Negative':0,'Neutral':1,'Positive':2})
```

[86]:
```
from sklearn.model_selection import train_test_split
xtrain1,xtest1,ytrain1,ytest1=train_test_split(text_tfidf,df['Score'],test_size=0.
 ↪2,random_state=42)
```

[87]:
```
#Use Tree-based classifiers like Random Forest and XGBoost
from sklearn.ensemble import RandomForestClassifier
rfclf=RandomForestClassifier()
rfclf.fit(xtrain1,ytrain1)
ypred1_rfclf=rfclf.predict(xtest1)
```

[88]:
```
print(classification_report(ytest1,ypred1_rfclf))
```

```
              precision    recall  f1-score   support

           0       0.92      0.26      0.40        47
           1       0.50      0.14      0.22        65
           2       0.91      0.99      0.95       888

    accuracy                           0.90      1000
   macro avg       0.78      0.46      0.52      1000
weighted avg       0.88      0.90      0.87      1000
```

[89]:
```
xtrain1_smote,ytrain1_smote=smote.fit_resample(xtrain1,ytrain1)
from collections import Counter
print('Before smote:',Counter(ytrain1))
print('After smote:',Counter(ytrain1_smote))
```

```
Before smote: Counter({2: 3585, 1: 226, 0: 189})
After smote: Counter({2: 3585, 1: 3585, 0: 3585})
```

[90]:
```
#Use possible ensemble techniques like: XGboost + oversampled_multinomial_NB
import xgboost as xg
from sklearn.naive_bayes import GaussianNB
nb=GaussianNB()
model=xg.XGBClassifier(base_estimator=GaussianNB())
model.fit(xtrain1,ytrain1)
```

[90]:
```
XGBClassifier(base_estimator=GaussianNB(), base_score=0.5, booster=None,
              colsample_bylevel=1, colsample_bynode=1, colsample_bytree=1,
```

```
              gamma=0, gpu_id=-1, importance_type='gain',
              interaction_constraints=None, learning_rate=0.300000012,
              max_delta_step=0, max_depth=6, min_child_weight=1, missing=nan,
              monotone_constraints=None, n_estimators=100, n_jobs=0,
              num_parallel_tree=1, objective='multi:softprob', random_state=0,
              reg_alpha=0, reg_lambda=1, scale_pos_weight=None, subsample=1,
              tree_method=None, validate_parameters=False, verbosity=None)
```

[92]:
```
ypred1_xg_nb=model.predict(xtest1)
accuracy_score(ytest1,ypred1_xg_nb)
```

[92]:
```
0.916
```

5 Adding sentiment score didn't gave any improved results.

6 Applying LSTM

[98]:
```
from tensorflow import keras
from tensorflow.keras.layers import Dense, Embedding, LSTM,GRU
from tensorflow.keras.optimizers import Adam,SGD
from keras.models import Sequential
from keras.preprocessing.text import Tokenizer
import tensorflow as tf
from keras import layers
from keras_preprocessing.sequence import pad_sequences
from keras.callbacks import EarlyStopping
from sklearn.model_selection import train_test_split
```

[100]:
```
most_common_words=1000
max_len=100
tokenizer = Tokenizer(most_common_words)
tokenizer.fit_on_texts(text_str)
sequences=tokenizer.texts_to_sequences(text_str)
word_index = tokenizer.word_index
print('Found %s unique tokens.' % len(word_index))
X = pad_sequences(sequences, maxlen=max_len)
```

```
Found 5998 unique tokens.
```

[101]:
```
X_train,X_test,y_train,y_test=train_test_split(X,
df['sentiment'],
test_size=0.2,
random_state=101)
```

[102]:
```
emb_dim=128
batch_size=256
```

```
print((X_train.shape, y_train.shape, X_test.shape, y_test.shape))
```

```
((4000, 100), (4000,), (1000, 100), (1000,))
```

[104]:
```
from tensorflow.keras.optimizers import SGD
model = Sequential()
model.add(Embedding(most_common_words, emb_dim, input_length=X.shape[1]))
#model.add(SpatialDropout1D(0.7))
model.add(LSTM(64, dropout=0.7, recurrent_dropout=0.7))
model.add(Dense(3, activation='softmax'))
model.compile(optimizer='adam', loss='sparse_categorical_crossentropy',␣
 ↪metrics=['acc'])
```

[108]:
```
history = model.fit(X_train, y_train,
epochs=5,
batch_size=batch_size,
validation_split=0.2,
callbacks=[EarlyStopping(monitor='val_loss',patience=7,min_delta=0.01)])
```

```
Epoch 1/5
13/13 [==============================] - 10s 560ms/step - loss: 0.7975 - acc:
0.8844 - val_loss: 0.2689 - val_acc: 0.9425
Epoch 2/5
13/13 [==============================] - 7s 533ms/step - loss: 0.2963 - acc:
0.9394 - val_loss: 0.2820 - val_acc: 0.9425
Epoch 3/5
13/13 [==============================] - 7s 542ms/step - loss: 0.2741 - acc:
0.9394 - val_loss: 0.2567 - val_acc: 0.9425
Epoch 4/5
13/13 [==============================] - 7s 533ms/step - loss: 0.2666 - acc:
0.9394 - val_loss: 0.2577 - val_acc: 0.9425
Epoch 5/5
13/13 [==============================] - 7s 527ms/step - loss: 0.2633 - acc:
0.9394 - val_loss: 0.2534 - val_acc: 0.9425
```

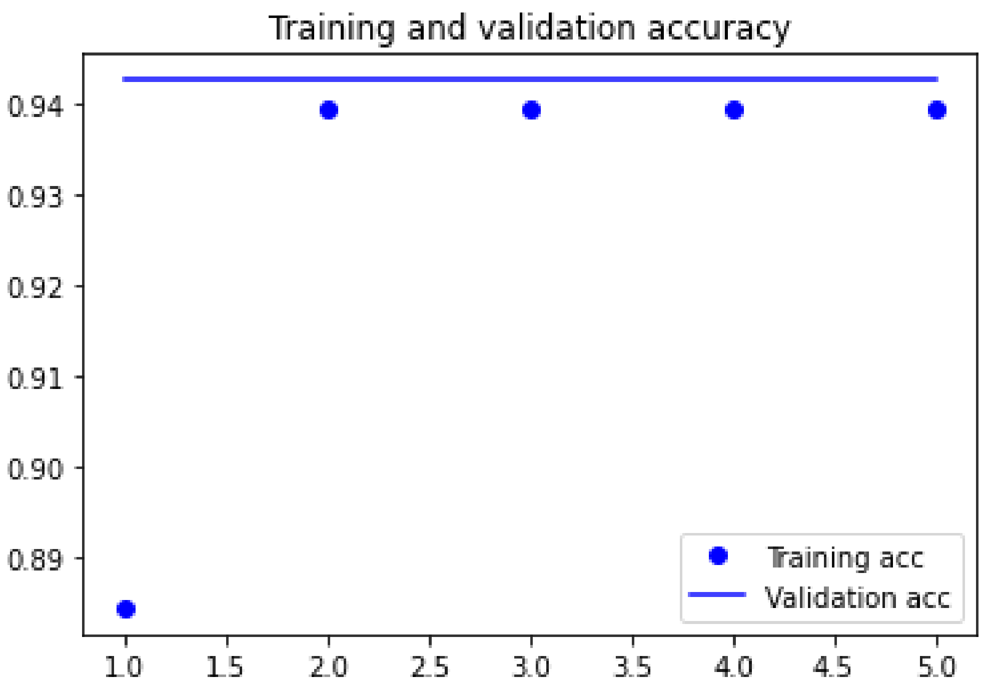
[109]:
```
X.shape
```

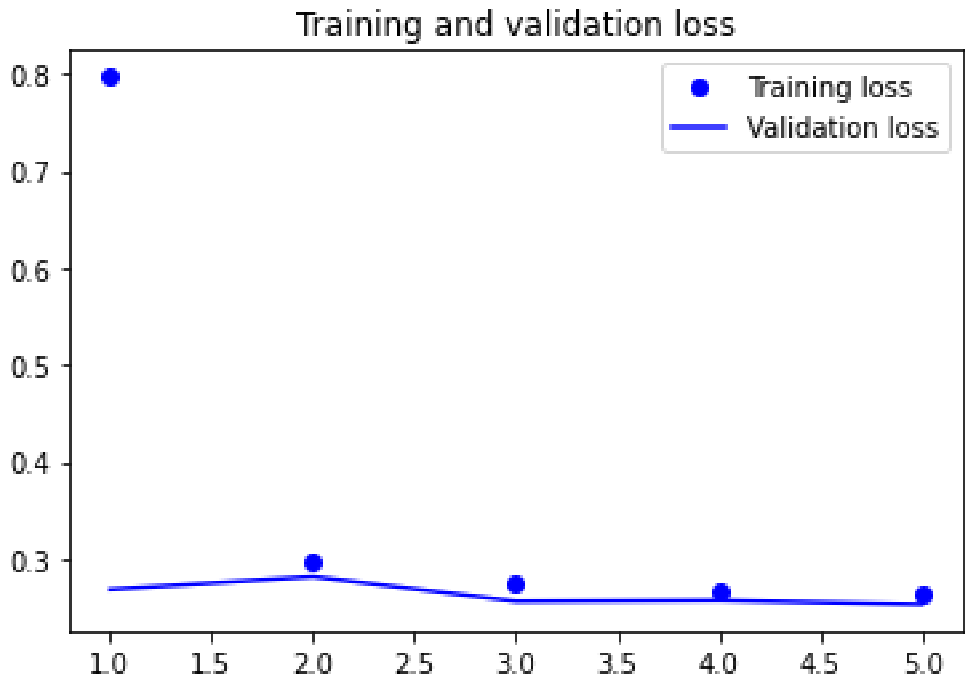
[109]:
```
(5000, 100)
```

[111]:
```
accr = model.evaluate(X_test,y_test)
print('Test set\n Loss: {:0.3f}\n Accuracy: {:0.3f}'.format(accr[0],accr[1]))
```

```
32/32 [==============================] - 1s 26ms/step - loss: 0.3106 - acc:
0.9260
Test set
 Loss: 0.311
 Accuracy: 0.926
```

[112]:
```python
acc = history.history['acc']
val_acc = history.history['val_acc']
loss = history.history['loss']
val_loss = history.history['val_loss']
epochs = range(1, len(acc) + 1)
plt.plot(epochs, acc, 'bo', label='Training acc')
plt.plot(epochs, val_acc, 'b', label='Validation acc')
plt.title('Training and validation accuracy')
plt.legend()
plt.figure()
plt.plot(epochs, loss, 'bo', label='Training loss')
plt.plot(epochs, val_loss, 'b', label='Validation loss')
plt.title('Training and validation loss')
plt.legend()
plt.show()
```





[113]:
```
X = np.reshape(X, (X.shape[0],X.shape[1],1))
X.shape
```

[113]: (5000, 100, 1)

[115]:
```
model=Sequential()
model.add(layers.GRU(units=64,
dropout=0.5,
recurrent_dropout=0.5,
input_shape=(X.shape[1],1)))
model.add(layers.Dense(3))
sgd = SGD(lr=0.001, decay=1e-6)
model.compile(loss='sparse_categorical_crossentropy', optimizer=sgd,␣
 ↪metrics=['acc'])
```

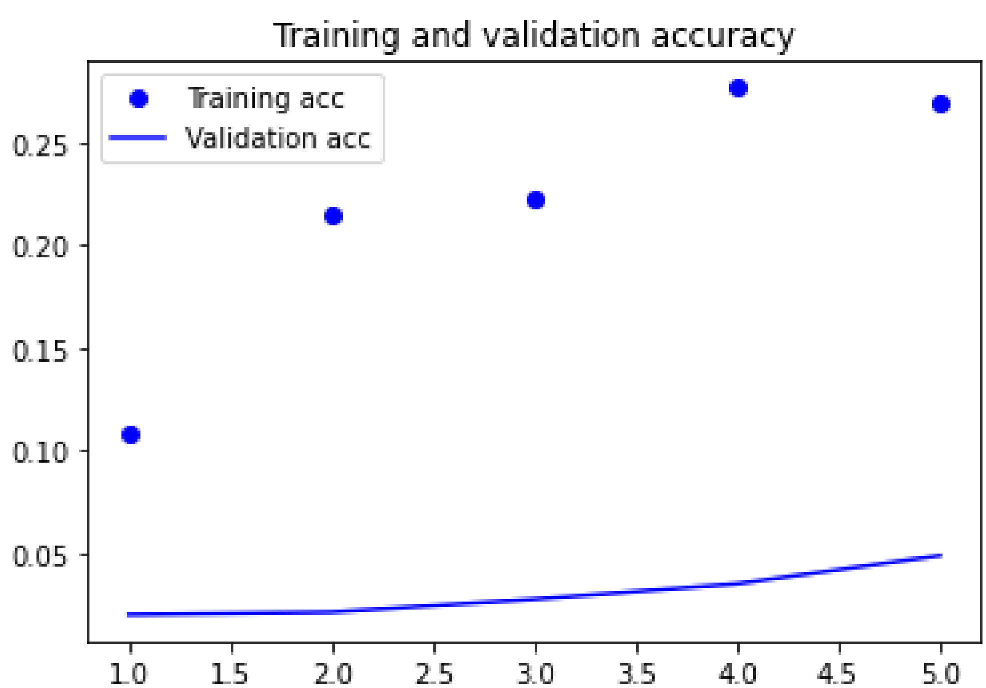
[116]:
```
history = model.fit(X_train, y_train,
epochs=5,
batch_size=batch_size,
validation_split=0.2)
```

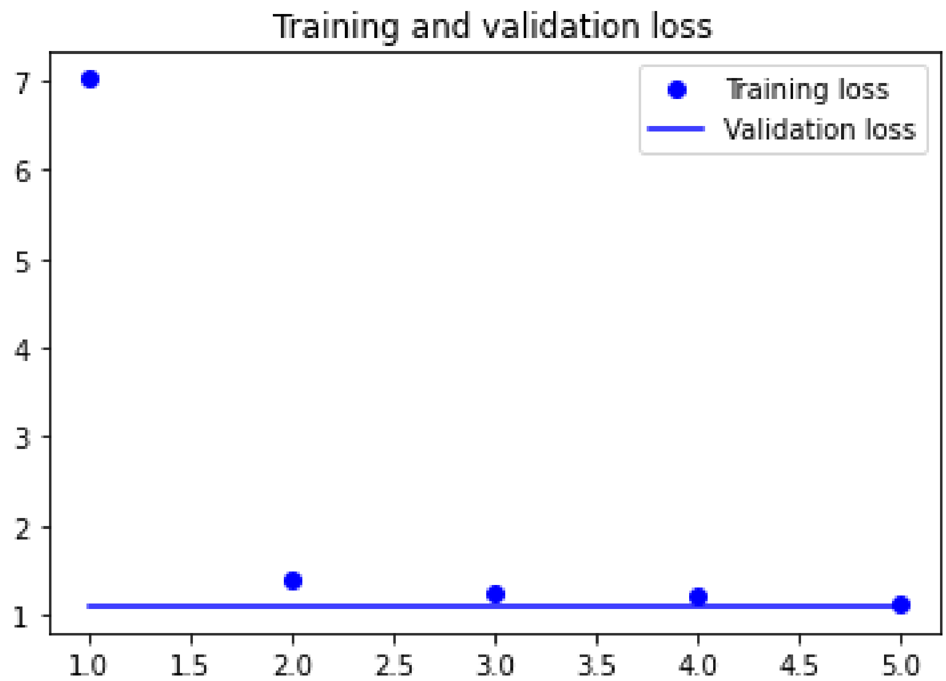
```
Epoch 1/5
13/13 [==============================] - 5s 258ms/step - loss: 7.0236 - acc:
0.1081 - val_loss: 1.0986 - val_acc: 0.0200
Epoch 2/5
13/13 [==============================] - 3s 233ms/step - loss: 1.3824 - acc:
```

```
0.2144 - val_loss: 1.0986 - val_acc: 0.0213
Epoch 3/5
13/13 [==============================] - 3s 233ms/step - loss: 1.2407 - acc:
0.2231 - val_loss: 1.0986 - val_acc: 0.0275
Epoch 4/5
13/13 [==============================] - 3s 231ms/step - loss: 1.2111 - acc:
0.2769 - val_loss: 1.0986 - val_acc: 0.0350
Epoch 5/5
13/13 [==============================] - 3s 228ms/step - loss: 1.1340 - acc:
0.2700 - val_loss: 1.0986 - val_acc: 0.0487
```

[117]:

```python
acc = history.history['acc']
val_acc = history.history['val_acc']
loss = history.history['loss']
val_loss = history.history['val_loss']
epochs = range(1, len(acc) + 1)
plt.plot(epochs, acc, 'bo', label='Training acc')
plt.plot(epochs, val_acc, 'b', label='Validation acc')
plt.title('Training and validation accuracy')
plt.legend()
plt.figure()
plt.plot(epochs, loss, 'bo', label='Training loss')
plt.plot(epochs, val_loss, 'b', label='Validation loss')
plt.title('Training and validation loss')
plt.legend()
plt.show()
```





7 Compare the accuracy of neural nets with traditional ML based algorithms: In both LSTM andGRU model the accuracy obtained are 94.25 whereas in ML algorithms likeRandomForestClassiSfier and SVM an accuracy of 95% was achieved. As a result, the MLmodels performed better than the NN models in this case.

8 Topic Modeling

[118]:
```
#vec=TfidfVectorizer()
#text_tfidf=vec.fit_transform(text_str)
X1=text_tfidf
from sklearn.decomposition import LatentDirichletAllocation
from pprint import pprint
import pyLDAvis
import pyLDAvis.sklearn
import matplotlib.pyplot as plt
%matplotlib inline

lda_model=LatentDirichletAllocation(n_components=12)
```

```
lda_output = lda_model.fit_transform(X1)
print(lda_model)
```

```
LatentDirichletAllocation(n_components=12)
```

[119]:
```
#See model parameters
pprint(lda_model.get_params())
```

```
{'batch_size': 128,
 'doc_topic_prior': None,
 'evaluate_every': -1,
 'learning_decay': 0.7,
 'learning_method': 'batch',
 'learning_offset': 10.0,
 'max_doc_update_iter': 100,
 'max_iter': 10,
 'mean_change_tol': 0.001,
 'n_components': 12,
 'n_jobs': None,
 'perp_tol': 0.1,
 'random_state': None,
 'topic_word_prior': None,
 'total_samples': 1000000.0,
 'verbose': 0}
```

[120]:
```
pyLDAvis.enable_notebook()
panel = pyLDAvis.sklearn.prepare(lda_model,X1,vec)
panel
```

[120]:
```
PreparedData(topic_coordinates=              x         y  topics  cluster
Freq
topic
1     -0.134930  0.051886       1        1  13.998058
7      0.083982  0.176955       2        1  12.614024
2      0.003444 -0.002841       3        1  12.233623
10    -0.202118  0.028712       4        1  12.173189
9      0.052512  0.003970       5        1   8.981147
11     0.056805  0.001583       6        1   6.502153
3      0.060450 -0.033754       7        1   6.391026
8     -0.023275 -0.073598       8        1   5.892591
4      0.046768 -0.032605       9        1   5.461780
0     -0.005336 -0.062515      10        1   5.417551
5      0.043083  0.010183      11        1   5.396284
6      0.018615 -0.067977      12        1   4.938575, topic_info=
Term       Freq       Total Category  logprob  loglift
825         echo  86.000000   86.000000  Default  30.0000  30.0000
1512       loves  97.000000   97.000000  Default  29.0000  29.0000
```

```
2132  recommend  49.000000   49.000000  Default  28.0000  28.0000
1748        old  67.000000   67.000000  Default  27.0000  27.0000
2969       year  54.000000   54.000000  Default  26.0000  26.0000
…         …        …          …       …        …       …
1509       love   4.362432  152.295957  Topic12  -5.2281  -0.5447
2050   purchase   3.534155   35.576893  Topic12  -5.4386   0.6989
2880       well   3.565537   57.800037  Topic12  -5.4298   0.2224
1748        old   3.590986   67.636986  Topic12  -5.4227   0.0724
407       books   3.462123   55.524773  Topic12  -5.4592   0.2331

[814 rows x 6 columns], token_table=      Topic      Freq      Term
term
2        10  0.543486      1000
12       10  0.492959        13
13       12  0.824880        14
19        9  0.763927       16g
21        6  0.516332        18
…     …        …        …
2981      7  0.604081     young
2983      2  0.728113  youngest
2990      2  0.806671        yr
2990      5  0.100834        yr
2991      4  0.732026       yrs

[2123 rows x 3 columns], R=30, lambda_step=0.01, plot_opts={'xlab': 'PC1',
'ylab': 'PC2'}, topic_order=[2, 8, 3, 11, 10, 12, 4, 9, 5, 1, 6, 7])
```

[ ]: