NLP: Twitter Sentimental Analysis

In [1]:

```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
```

In [2]:

```
# Load the data
tweets_df = pd.read_csv(r'C:\Users\Manish Kumar\Documents\twitter.csv')
```

In [3]:

```
tweets_df
```

Out[3]:

	id	label	tweet
0	1	0	@user when a father is dysfunctional and is s...
1	2	0	@user @user thanks for #lyft credit i can't us...
2	3	0	bihday your majesty
3	4	0	#model i love u take with u all the time in ...
4	5	0	factsguide: society now #motivation
...	...	...	...
31957	31958	0	ate @user isz that youuu?ð ð ð ð ð ð...
31958	31959	0	to see nina turner on the airwaves trying to...
31959	31960	0	listening to sad songs on a monday morning otw...
31960	31961	1	@user #sikh #temple vandalised in in #calgary,...
31961	31962	0	thank you @user for you follow

31962 rows × 3 columns

In [4]:

```
tweets_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 31962 entries, 0 to 31961
Data columns (total 3 columns):
 #   Column  Non-Null Count  Dtype 
---  ------  --------------  ----- 
 0   id      31962 non-null  int64 
 1   label   31962 non-null  int64 
 2   tweet   31962 non-null  object
dtypes: int64(2), object(1)
memory usage: 749.2+ KB
```

In [5]:

```
tweets_df.describe()
```

Out[5]:

	id	label
count	31962.000000	31962.000000
mean	15981.500000	0.070146
std	9226.778988	0.255397
min	1.000000	0.000000
25%	7991.250000	0.000000
50%	15981.500000	0.000000
75%	23971.750000	0.000000
max	31962.000000	1.000000

In [6]:

```
tweets_df['tweet']
```

Out[6]:

```
0         @user when a father is dysfunctional and is s...
1        @user @user thanks for #lyft credit i can't us...
2                                      bihday your majesty
3        #model   i love u take with u all the time in ...
4                   factsguide: society now    #motivation
                               ...                        
31957    ate @user isz that youuu?ð   ð   ð   ð   ð   ð...
31958      to see nina turner on the airwaves trying to...
31959    listening to sad songs on a monday morning otw...
31960    @user #sikh #temple vandalised in in #calgary,...
31961                     thank you @user for you follow  
Name: tweet, Length: 31962, dtype: object
```

In [7]:

```
# Drop the 'id' column
tweets_df = tweets_df.drop(['id'], axis=1)
```

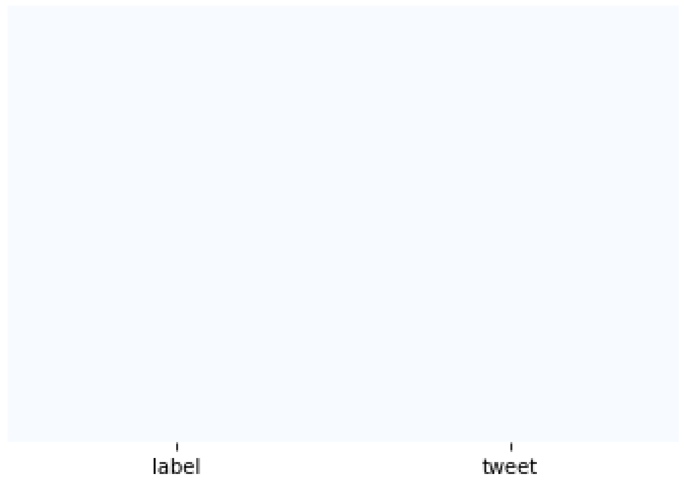
TASK #2: PERFORM DATA EXPLORATION

In [8]:

```
sns.heatmap(tweets_df.isnull(), yticklabels = False, cbar = False, cmap="Blues")
```

Out[8]:

```
<AxesSubplot:>
```

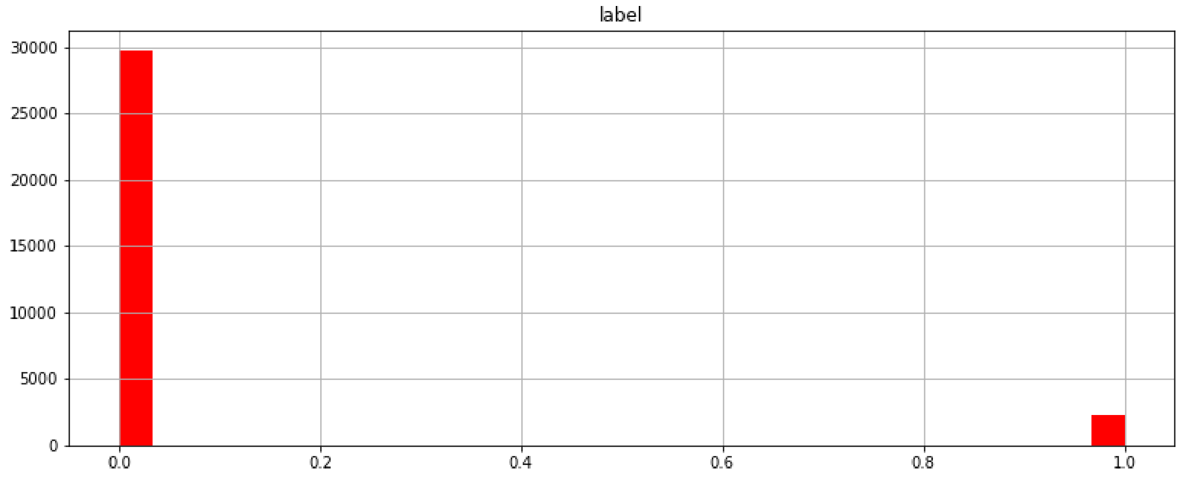


In [9]:

```
tweets_df.hist(bins = 30, figsize = (13,5), color = 'r')
```

Out[9]:

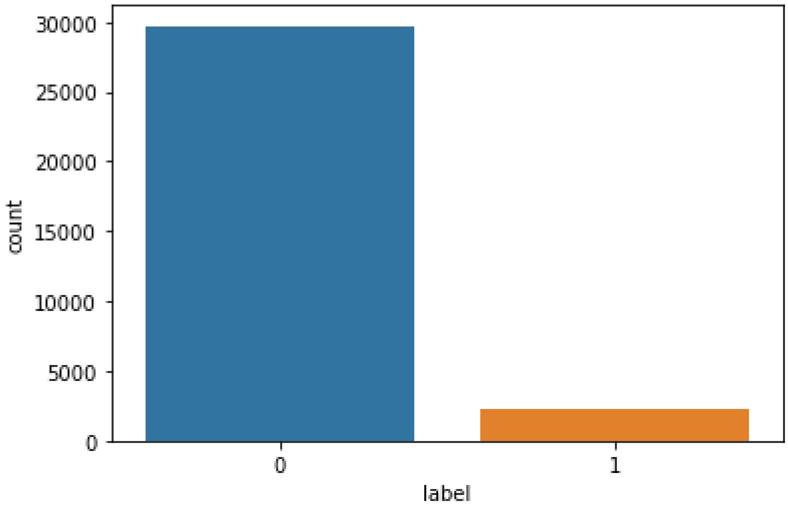
```
array([[<AxesSubplot:title={'center':'label'}>]], dtype=object)
```



In [10]:

```
sns.countplot(tweets_df['label'], label = "Count") ;
```

```
M:\ANAC\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the
following variable as a keyword arg: x. From version 0.12, the only valid po
sitional argument will be `data`, and passing other arguments without an exp
licit keyword will result in an error or misinterpretation.
  warnings.warn(
```



In [11]:

```
# Let's get the length of the messages
tweets_df['length'] = tweets_df['tweet'].apply(len)
```

In [12]:

```
tweets_df
```

Out[12]:

	label	tweet	length
0	0	@user when a father is dysfunctional and is s...	102
1	0	@user @user thanks for #lyft credit i can't us...	122
2	0	bihday your majesty	21
3	0	#model i love u take with u all the time in ...	86
4	0	factsguide: society now #motivation	39
...	...	...	...
31957	0	ate @user isz that youuu?ð ð ð ð ð ð...	68
31958	0	to see nina turner on the airwaves trying to...	131
31959	0	listening to sad songs on a monday morning otw...	63
31960	1	@user #sikh #temple vandalised in in #calgary,...	67
31961	0	thank you @user for you follow	32

31962 rows × 3 columns

In [13]:

```
tweets_df.describe()
```

Out[13]:

	label	length
count	31962.000000	31962.000000
mean	0.070146	84.739628
std	0.255397	29.455749
min	0.000000	11.000000
25%	0.000000	63.000000
50%	0.000000	88.000000
75%	0.000000	108.000000
max	1.000000	274.000000

In [14]:

```
# Let's see the shortest message
tweets_df[tweets_df['length'] == 11]['tweet'].iloc[0]
```

Out[14]:

```
'i love you '
```

In [15]:

```
# Let's view the message with mean length
tweets_df[tweets_df['length'] == 84]['tweet'].iloc[0]
```

Out[15]:

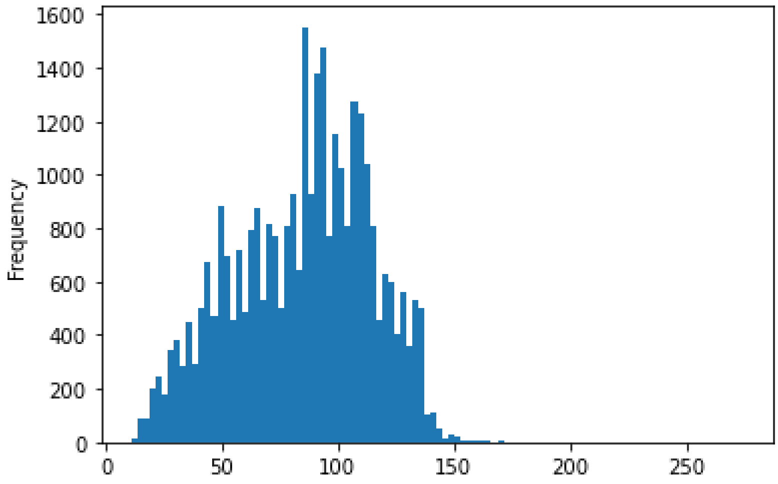
```
'my mom shares the same bihday as @user   bihday snake! see you this weekend
ð\x9f\x99\x8cð\x9f\x8f¼'
```

In [16]:

```
# Plot the histogram of the length column
tweets_df['length'].plot(bins=100, kind='hist')
```

Out[16]:

```
<AxesSubplot:ylabel='Frequency'>
```



TASK #3: PLOT THE WORDCLOUD

In [17]:

```
positive = tweets_df[tweets_df['label']==0]
positive
```

Out[17]:

	label	tweet	length
0	0	@user when a father is dysfunctional and is s...	102
1	0	@user @user thanks for #lyft credit i can't us...	122
2	0	bihday your majesty	21
3	0	#model i love u take with u all the time in ...	86
4	0	factsguide: society now #motivation	39
...	...	...	...
31956	0	off fishing tomorrow @user carnt wait first ti...	61
31957	0	ate @user isz that youuu?ð ð ð ð ð ð...	68
31958	0	to see nina turner on the airwaves trying to...	131
31959	0	listening to sad songs on a monday morning otw...	63
31961	0	thank you @user for you follow	32

29720 rows × 3 columns

In [18]:

```
negative = tweets_df[tweets_df['label']==1]
negative
```

Out[18]:

	label	tweet	length
13	1	@user #cnn calls #michigan middle school 'buil...	74
14	1	no comment! in #australia #opkillingbay #se...	101
17	1	retweet if you agree!	22
23	1	@user @user lumpy says i am a . prove it lumpy.	47
34	1	it's unbelievable that in the 21st century we'...	104
...	...	...	...
31934	1	lady banned from kentucky mall. @user #jcpenn...	59
31946	1	@user omfg i'm offended! i'm a mailbox and i'...	82
31947	1	@user @user you don't have the balls to hashta...	112
31948	1	makes you ask yourself, who am i? then am i a...	87
31960	1	@user #sikh #temple vandalised in in #calgary,...	67

2242 rows × 3 columns

In [19]:

```
sentences = tweets_df['tweet'].tolist()
len(sentences)
```

Out[19]:

```
31962
```

In [20]:

```
sentences_as_one_string =" ".join(sentences)
```

In [21]:

```
sentences_as_one_string
```

Out[21]:

```
' @user when a father is dysfunctional and is so selfish he drags his kids
into his dysfunction.   #run @user @user thanks for #lyft credit i can\'t
use cause they don\'t offer wheelchair vans in pdx.    #disapointed #getth
anked   bihday your majesty #model   i love u take with u all the time in
urð\x9f\x93±!!! ð\x9f\x98\x99ð\x9f\x98\x8eð\x9f\x91\x84ð\x9f\x91\x85ð\x9f
\x92¦ð\x9f\x92¦ð\x9f\x92¦    factsguide: society now    #motivation [2/2]
huge fan fare and big talking before they leave. chaos and pay disputes wh
en they get there. #allshowandnogo    @user camping tomorrow @user @user @
user @user @user @user @user dannyâ\x80¦ the next school year is the year
for exams.ð\x9f\x98¯ can\'t think about that ð\x9f\x98\xad #school #exams
#hate #imagine #actorslife #revolutionschool #girl we won!!! love the lan
d!!! #allin #cavs #champions #cleveland #clevelandcavaliers  â\x80¦   @use
r @user welcome here !  i\'m   it\'s so #gr8 !   â\x86\x9d #ireland consum
er price index (mom) climbed from previous 0.2% to 0.5% in may   #blog #si
lver #gold #forex we are so selfish. #orlando #standwithorlando #pulseshoo
ting #orlandoshooting #biggerproblems #selfish #heabreaking   #values #lov
e # i get to see my daddy today!!   #80days #gettingfed @user #cnn calls #
michigan middle school \'build the wall\' chant \'\' #tcot   no comment!
```

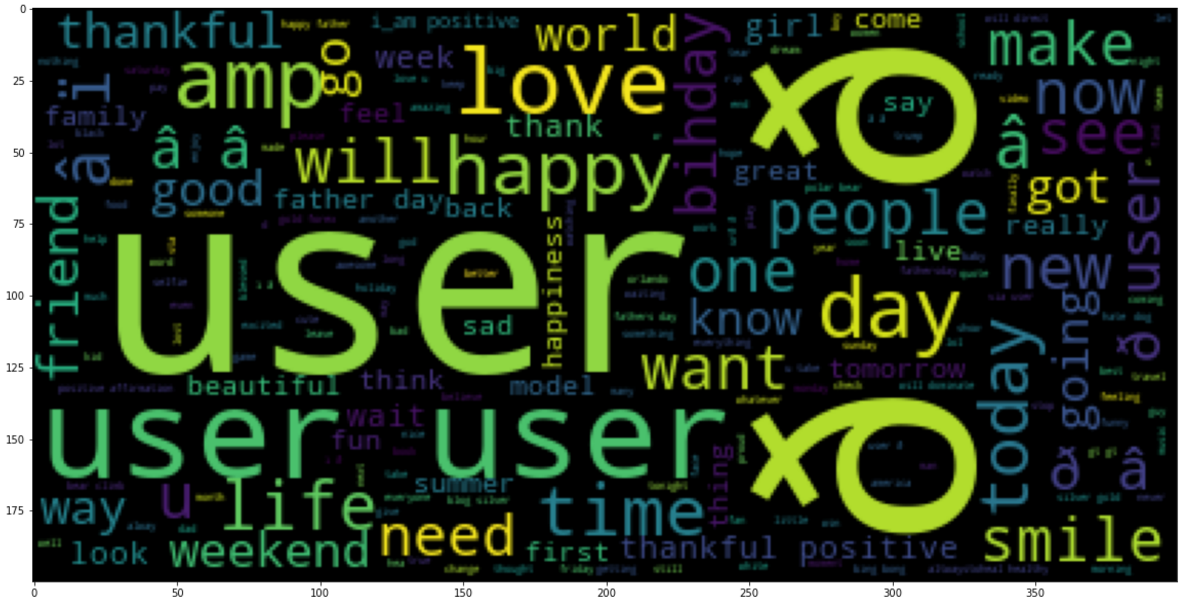
In [ ]:

In [22]:

```
from wordcloud import WordCloud
plt.figure(figsize=(20,20))
plt.imshow(WordCloud().generate(sentences_as_one_string))
```

Out[22]:

```
<matplotlib.image.AxesImage at 0x1c9bd8492e0>
```

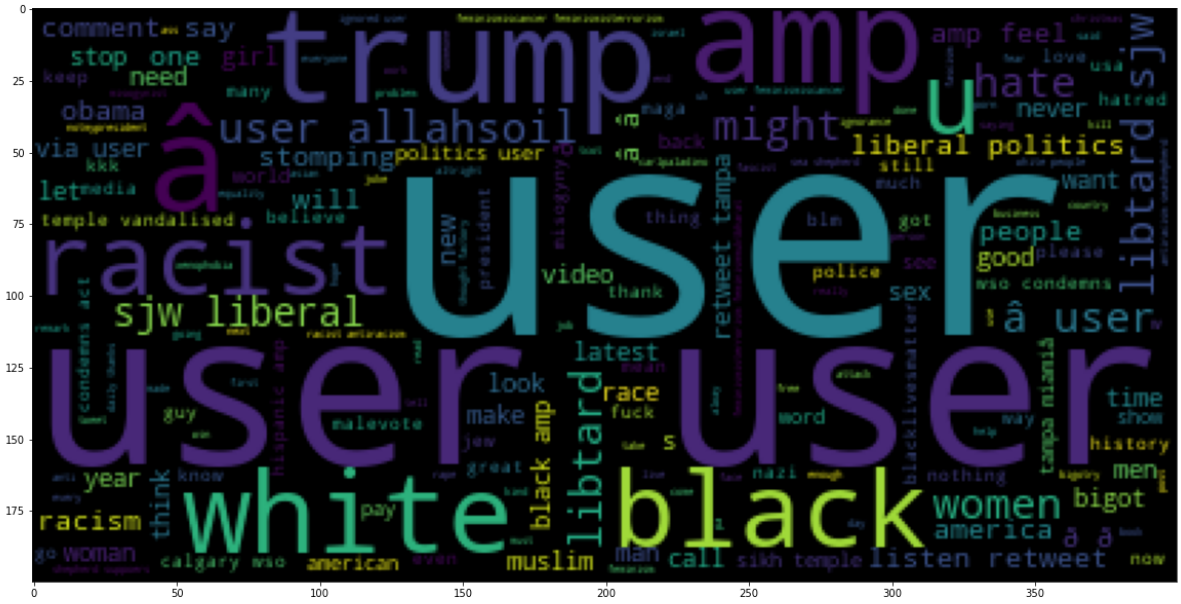


In [23]:

```
negative_list = negative['tweet'].tolist()
negative_list
negative_sentences_as_one_string = " ".join(negative_list)
plt.figure(figsize=(20,20))
plt.imshow(WordCloud().generate(negative_sentences_as_one_string))
```

Out[23]:

```
<matplotlib.image.AxesImage at 0x1c9c1bf5280>
```



TASK #3: PERFORM DATA CLEANING - REMOVE PUNCTUATION FROM TEXT

In [24]:

```
import string
string.punctuation
```

Out[24]:

```
'!"#$%&\'()*+,-./:;<=>?@[\\]^_`{|}~'
```

In [25]:

```
Test = '$I love AI & Machine learning!!'
Test_punc_removed = [char for char in Test if char not in string.punctuation]
Test_punc_removed_join = ''.join(Test_punc_removed)
Test_punc_removed_join
```

Out[25]:

```
'I love AI  Machine learning'
```

In [26]:

```
Test = 'Good morning beautiful people :)... I am having fun learning Machine learning and A
```

In [27]:

```
Test_punc_removed = [char for char in Test if char not in string.punctuation]
Test_punc_removed
```

Out[27]:

```
['G',
 'o',
 'o',
 'd',
 ' ',
 'm',
 'o',
 'r',
 'n',
 'i',
 'n',
 'g',
 ' ',
 'b',
 'e',
 'a',
 'u',
 't',
 'i',
 'f',
 'u',
 'l',
 ' ',
 'p',
 'e',
 'o',
 'p',
 'l',
 'e',
 ' ',
 ' ',
 'I',
 ' ',
 'a',
 'm',
 ' ',
 'h',
 'a',
 'v',
 'i',
 'n',
 'g',
 ' ',
 'f',
 'u',
 'n',
 ' ',
 'l',
 'e',
 'a',
 'r',
 'n',
 'i',
 'n',
 'g',
 ' ',
 'M',
 'a',
 'c',
 'h',
 'i',
 'n',
 'e',
 ' ',
 'l',
 'e',
 'a',
 'r',
 'n',
 'i',
 'n',
 'g',
 ' ',
 'a',
 'n',
 'd',
 ' ',
 'A',
 'I']
```

In [28]:

```
# Join the characters again to form the string.
Test_punc_removed_join = ''.join(Test_punc_removed)
Test_punc_removed_join
```

Out[28]:

```
'Good morning beautiful people  I am having fun learning Machine learning an
d AI'
```

TASK 4: PERFORM DATA CLEANING - REMOVE STOPWORDS

In [29]:

```
import nltk # Natural Language tool kit
nltk.download('stopwords')

# You have to download stopwords Package to execute this command
from nltk.corpus import stopwords
stopwords.words('english')
```

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

In [30]:

```
Test_punc_removed_join = 'I enjoy coding, programming and Artificial intelligence'
Test_punc_removed_join_clean = [word for word in Test_punc_removed_join.split() if word.low
```

In [31]:

```
Test_punc_removed_join_clean # Only important (no so common) words are left
```

Out[31]:

```
['enjoy', 'coding,', 'programming', 'Artificial', 'intelligence']
```

In [32]:

```
Test_punc_removed_join
```

Out[32]:

```
'I enjoy coding, programming and Artificial intelligence'
```

TASK 5: PERFORM COUNT VECTORIZATION (TOKENIZATION)

In [33]:

```
from sklearn.feature_extraction.text import CountVectorizer
sample_data = ['This is the first paper.','This document is the second paper.','And this is

vectorizer = CountVectorizer()
X = vectorizer.fit_transform(sample_data)
```

In [34]:

```
print(vectorizer.get_feature_names())
```

```
['and', 'document', 'first', 'is', 'one', 'paper', 'second', 'the', 'third',
'this']
```

In [35]:

```
print(X.toarray())
```

```
[[0 0 1 1 0 1 0 1 0 1]
 [0 1 0 1 0 1 1 1 0 1]
 [1 0 0 1 1 0 0 1 1 1]
 [0 0 1 1 0 1 0 1 0 1]]
```

In [36]:

```
mini_challenge = ['Hello World','Hello Hello World','Hello World world world']

# mini_challenge = ['Hello World', 'Hello Hello Hello World world', 'Hello Hello World worl

vectorizer_challenge = CountVectorizer()
X_challenge = vectorizer_challenge.fit_transform(mini_challenge)
print(X_challenge.toarray())
```

```
[[1 1]
 [2 1]
 [1 3]]
```

TASK #6: CREATE A PIPELINE TO REMOVE PUNCTUATIONS, STOPWORDS AND PERFORM COUNT VECTORIZATION

In [37]:

```
# Let's define a pipeline to clean up all the messages
# The pipeline performs the following: (1) remove punctuation, (2) remove stopwords

def message_cleaning(message):
    Test_punc_removed = [char for char in message if char not in string.punctuation]
    Test_punc_removed_join = ''.join(Test_punc_removed)
    Test_punc_removed_join_clean = [word for word in Test_punc_removed_join.split() if word
    return Test_punc_removed_join_clean
```

In [38]:

```
# Let's test the newly added function
tweets_df_clean = tweets_df['tweet'].apply(message_cleaning)
```

In [39]:

```
print(tweets_df_clean[5]) # show the cleaned up version
```

```
['22', 'huge', 'fan', 'fare', 'big', 'talking', 'leave', 'chaos', 'pay', 'di
sputes', 'get', 'allshowandnogo']
```

In [40]:

```
print(tweets_df['tweet'][5]) # show the original version
```

```
[2/2] huge fan fare and big talking before they leave. chaos and pay dispute
s when they get there. #allshowandnogo
```

In [41]:

```
from sklearn.feature_extraction.text import CountVectorizer
# Define the cleaning pipeline we defined earlier
vectorizer = CountVectorizer(analyzer = message_cleaning, dtype = np.uint8)
tweets_countvectorizer = vectorizer.fit_transform(tweets_df['tweet'])
```

In [42]:

```
print(vectorizer.get_feature_names())
```

```
['0', '0000001', '00027', '001', '0035', '00h30', '01', '0115', '0161', '0
19', '01926889917', '02', '0265', '0266808099', '02900', '03', '030916',
'03111880779', '032', '033', '0345', '039', '04', '045', '04k', '05', '050
6823156', '06', '06052016', '0606', '060616', '0608', '0608wed', '0609',
'0610', '061116', '0612', '0613', '0616', '0617', '0618', '0618saturday7mo
nthscouple', '0618â\x99¡', '0620', '06202016', '0622', '0624', '06Â', '0
7', '07000', '07040', '07044', '07150', '07190', '07400', '07468', '0750
0', '076', '07788427999', '07800', '07840', '07850', '07870', '07900', '07
930', '07950', '08', '0806', '080616', '088b', '08â\x80¦', '09', '0906201
6', '0933m', '09600', '0k', '0shares', '0tolerancemovie', '0ð\x9f\x98¥â\x9
8¹ï¸\x8f', '1', '10', '100', '1000', '100000', '10003', '10007', '1000gift
s', '1000th', '1000x', '1000yr', '1000â\x82¬', '1001', '1001000s', '1001
4', '10021', '10025', '10040', '100616', '10064', '100d', '100daysofcode',
'100daysofpigpaintings', '100daysoftea', '100faces', '100happydays', '100h
appydaysâ\x80¦', '100happysongs', '100juiceð\x9f\x8d\x8dð\x9f\x8d\x93ð\x9f
\x8d\x87ð\x9f\x8d\x92ð\x9f\x8d\x91ð\x9f\x8d\x8b', '100k', '100ml', '100p
m', '100yr', '100â\x80\x99s', '101', '10125', '1014', '10143hr', '1015',
'1017', '1019', '101dalmatians', '101daysofsmiles', '101ð\x9f\x98\x89ð\x9f
\x98\x89ð\x9f\x8e\x89ð\x9f\x8e\x89ð\x9f\x92¥ð\x9f\x92¥', '1027', '102816',
```

In [43]:

```
print(tweets_countvectorizer.toarray())
```

```
[[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]]
```

In [44]:

```
tweets_countvectorizer.shape
```

Out[44]:

```
(31962, 47386)
```

In [45]:

```
X = pd.DataFrame(tweets_countvectorizer.toarray())
```

In [46]:

```
X
```

Out[46]:

	0	1	2	3	4	5	6	7	8	9	...	47376	47377	47378	47379	47380	47381	473
0	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	
1	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	
2	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	
3	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	
4	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
31957	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	
31958	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	
31959	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	
31960	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	
31961	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	

31962 rows × 47386 columns

In [47]:

```
y = tweets_df['label']
```

TASK #7: TRAIN AND EVALUATE A NAIVE BAYES CLASSIFIER MODEL

In [48]:

```
X.shape
```

Out[48]:

```
(31962, 47386)
```

In [49]:

```
y.shape
```

Out[49]:

```
(31962,)
```

In [50]:

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2)
```

Using Naive Bayes

In [51]:

```
from sklearn.naive_bayes import MultinomialNB

NB_classifier = MultinomialNB()
NB_classifier.fit(X_train, y_train)
```

Out[51]:

```
MultinomialNB()
```

In [52]:

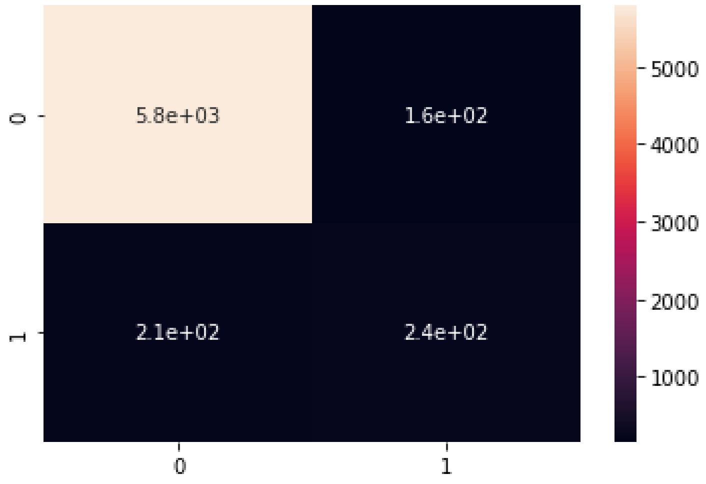
```
from sklearn.metrics import classification_report, confusion_matrix
```

In [53]:

```
# Predicting the Test set results
y_predict_test = NB_classifier.predict(X_test)
cm = confusion_matrix(y_test, y_predict_test)
sns.heatmap(cm, annot=True)
```

Out[53]:

```
<AxesSubplot:>
```



In [54]:

```
print(classification_report(y_test, y_predict_test))
```

```
              precision    recall  f1-score   support

           0       0.97      0.97      0.97      5948
           1       0.60      0.54      0.57       445

    accuracy                           0.94      6393
   macro avg       0.78      0.76      0.77      6393
weighted avg       0.94      0.94      0.94      6393
```

In [ ]:

```

```

In [105]:

```
from sklearn.linear_model import LogisticRegression
```

In [106]:

```
 #Training the model
lr = LogisticRegression()
lr.fit(X_train, y_train)
```

Out[106]:

```
LogisticRegression()
```

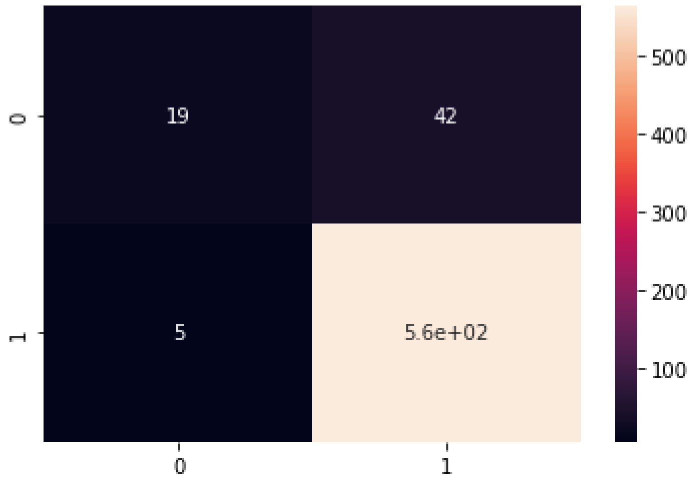
Using Logistic Regression

In [107]:

```
# Predicting the Test set results
y_predict_test = lr.predict(X_test)
cm = confusion_matrix(y_test, y_predict_test)
sns.heatmap(cm, annot=True)
```

Out[107]:

```
<AxesSubplot:>
```



In [108]:

```
print(classification_report(y_test, y_predict_test))
```

```
              precision    recall  f1-score   support

           0       0.79      0.31      0.45        61
           1       0.93      0.99      0.96       569

    accuracy                           0.93       630
   macro avg       0.86      0.65      0.70       630
weighted avg       0.92      0.93      0.91       630
```

Accuracies:-

Naive Bayes :- 0.94

Logistic Regression:- 0.93

Using Support vector Machine (SVM)

In [59]:

```
# from sklearn import svm
```

In [60]:

```
 #Training the model
# lr = svm.SVC()
```

In [61]:

```
# lr.fit(X_train, y_train)
```

Amazon Echo Customer Reviews

Goal-

Build train, test and artificial intelligence (AI) model to predict sentiment from real Amazon Echo customer reviews.

Tool: Anaconda, Python, Scikit-learn, Matplotlib, Seaborn

In [62]:

```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
```

In [63]:

```
reviews_df = pd.read_csv(r"C:\Users\Manish Kumar\Documents\ML\amazon_reviews.csv")
reviews_df
```

Out[63]:

	rating	date	variation	verified_reviews	feedback
0	5	31-Jul-18	Charcoal Fabric	Love my Echo!	1
1	5	31-Jul-18	Charcoal Fabric	Loved it!	1
2	4	31-Jul-18	Walnut Finish	Sometimes while playing a game, you can answer...	1
3	5	31-Jul-18	Charcoal Fabric	I have had a lot of fun with this thing. My 4 ...	1
4	5	31-Jul-18	Charcoal Fabric	Music	1
...	...	...	...	...	...
3145	5	30-Jul-18	Black Dot	Perfect for kids, adults and everyone in betwe...	1
3146	5	30-Jul-18	Black Dot	Listening to music, searching locations, check...	1
3147	5	30-Jul-18	Black Dot	I do love these things, i have them running my...	1
3148	5	30-Jul-18	White Dot	Only complaint I have is that the sound qualit...	1
3149	4	29-Jul-18	Black Dot	Good	1

3150 rows × 5 columns

In [64]:

```
reviews_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3150 entries, 0 to 3149
Data columns (total 5 columns):
 #   Column            Non-Null Count  Dtype 
---  ------            --------------  ----- 
 0   rating            3150 non-null   int64 
 1   date              3150 non-null   object
 2   variation         3150 non-null   object
 3   verified_reviews  3150 non-null   object
 4   feedback          3150 non-null   int64 
dtypes: int64(2), object(3)
memory usage: 123.2+ KB
```

In [65]:

```
reviews_df.describe()
```

Out[65]:

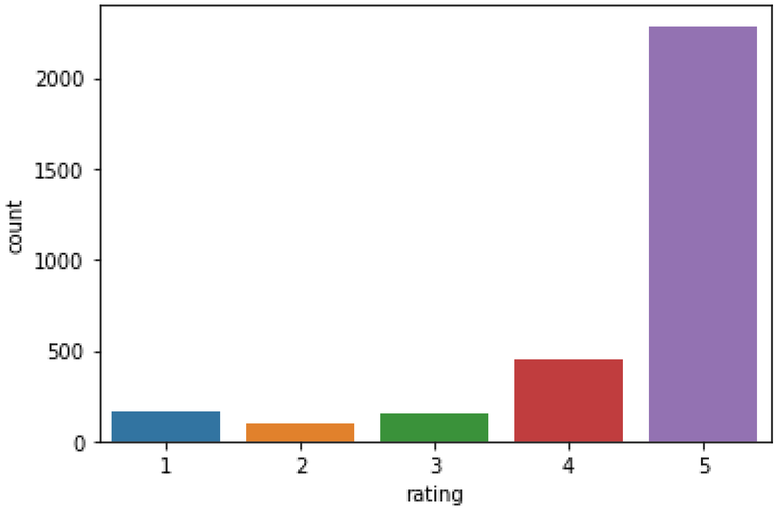
	rating	feedback
count	3150.000000	3150.000000
mean	4.463175	0.918413
std	1.068506	0.273778
min	1.000000	0.000000
25%	4.000000	1.000000
50%	5.000000	1.000000
75%	5.000000	1.000000
max	5.000000	1.000000

In [66]:

```
# Plot the count plot for the ratings
sns.countplot(x = reviews_df['rating'])
```

Out[66]:

```
<AxesSubplot:xlabel='rating', ylabel='count'>
```



In [67]:

```
reviews_df ['length'] = reviews_df['verified_reviews'].apply(len)
```

In [68]:

```
reviews_df
```

Out[68]:

	rating	date	variation	verified_reviews	feedback	length
0	5	31-Jul-18	Charcoal Fabric	Love my Echo!	1	13
1	5	31-Jul-18	Charcoal Fabric	Loved it!	1	9
2	4	31-Jul-18	Walnut Finish	Sometimes while playing a game, you can answer...	1	195
3	5	31-Jul-18	Charcoal Fabric	I have had a lot of fun with this thing. My 4 ...	1	172
4	5	31-Jul-18	Charcoal Fabric	Music	1	5
...	...	...	...	...	...	...
3145	5	30-Jul-18	Black Dot	Perfect for kids, adults and everyone in betwe...	1	50
3146	5	30-Jul-18	Black Dot	Listening to music, searching locations, check...	1	135
3147	5	30-Jul-18	Black Dot	I do love these things, i have them running my...	1	441
3148	5	30-Jul-18	White Dot	Only complaint I have is that the sound qualit...	1	380
3149	4	29-Jul-18	Black Dot	Good	1	4

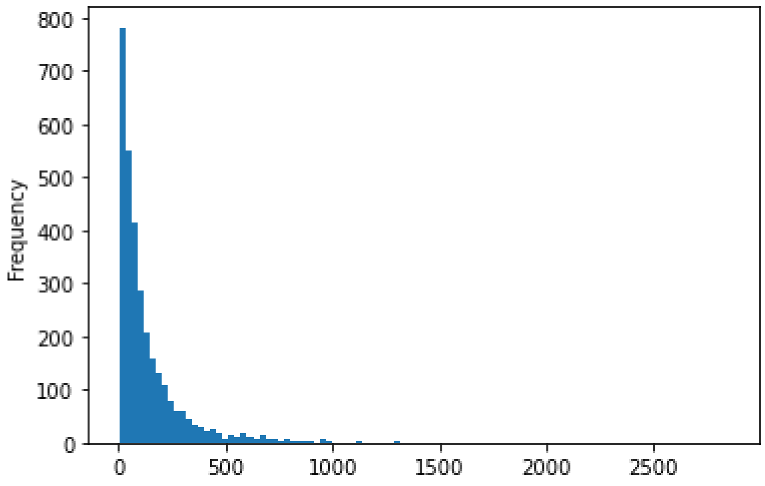
3150 rows × 6 columns

In [69]:

```
reviews_df['length'].plot(bins=100, kind = 'hist')
```

Out[69]:

```
<AxesSubplot:ylabel='Frequency'>
```



In [70]:

```
reviews_df.describe()
```

Out[70]:

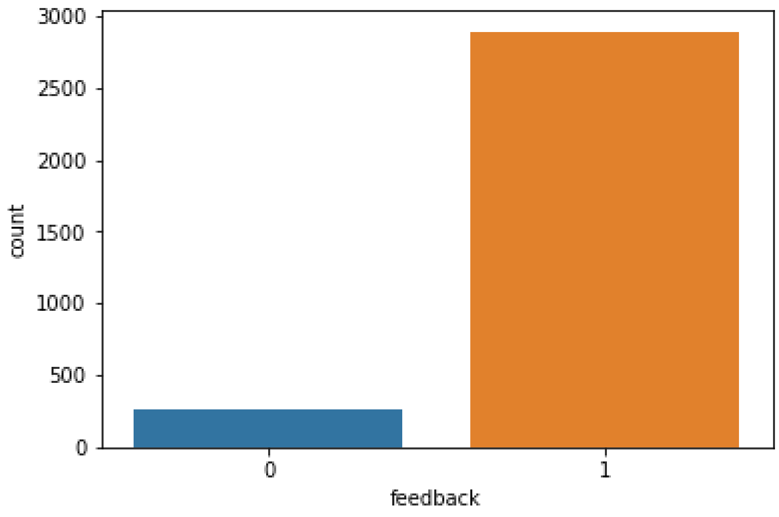
	rating	feedback	length
count	3150.000000	3150.000000	3150.000000
mean	4.463175	0.918413	132.049524
std	1.068506	0.273778	182.099952
min	1.000000	0.000000	1.000000
25%	4.000000	1.000000	30.000000
50%	5.000000	1.000000	74.000000
75%	5.000000	1.000000	165.000000
max	5.000000	1.000000	2851.000000

In [71]:

```
sns.countplot(x = reviews_df['feedback'])
```

Out[71]:

```
<AxesSubplot:xlabel='feedback', ylabel='count'>
```



Plotting Wordcloud

In [72]:

```
# Obtain only the positive reviews
positive = reviews_df[reviews_df['feedback'] == 1]
positive
```

Out[72]:

	rating	date	variation	verified_reviews	feedback	length
0	5	31-Jul-18	Charcoal Fabric	Love my Echo!	1	13
1	5	31-Jul-18	Charcoal Fabric	Loved it!	1	9
2	4	31-Jul-18	Walnut Finish	Sometimes while playing a game, you can answer...	1	195
3	5	31-Jul-18	Charcoal Fabric	I have had a lot of fun with this thing. My 4 ...	1	172
4	5	31-Jul-18	Charcoal Fabric	Music	1	5
...	...	...	...	...	...	...
3145	5	30-Jul-18	Black Dot	Perfect for kids, adults and everyone in betwe...	1	50
3146	5	30-Jul-18	Black Dot	Listening to music, searching locations, check...	1	135
3147	5	30-Jul-18	Black Dot	I do love these things, i have them running my...	1	441
3148	5	30-Jul-18	White Dot	Only complaint I have is that the sound qualit...	1	380
3149	4	29-Jul-18	Black Dot	Good	1	4

2893 rows × 6 columns

In [73]:

```
# Obtain only the negative reviews
negative = reviews_df[reviews_df['feedback'] == 0]
negative
```

Out[73]:

	rating	date	variation	verified_reviews	feedback	length
46	2	30-Jul-18	Charcoal Fabric	It's like Siri, in fact, Siri answers more acc...	0	163
111	2	30-Jul-18	Charcoal Fabric	Sound is terrible if u want good music too get...	0	53
141	1	30-Jul-18	Charcoal Fabric	Not much features.	0	18
162	1	30-Jul-18	Sandstone Fabric	Stopped working after 2 weeks ,didn't follow c...	0	87
176	2	30-Jul-18	Heather Gray Fabric	Sad joke. Worthless.	0	20
...	...	...	...	...	...	...
3047	1	30-Jul-18	Black Dot	Echo Dot responds to us when we aren't even ta...	0	120
3048	1	30-Jul-18	White Dot	NOT CONNECTED TO MY PHONE PLAYLIST :(	0	37
3067	2	30-Jul-18	Black Dot	The only negative we have on this product is t...	0	240
3091	1	30-Jul-18	Black Dot	I didn’t order it	0	17
3096	1	30-Jul-18	White Dot	The product sounded the same as the emoji spea...	0	210

257 rows × 6 columns

In [74]:

```
#Convert to list format
sentences = positive['verified_reviews'].tolist()
len(sentences)
```

Out[74]:

```
2893
```

In [75]:

```
sentences_as_one_string = " ".join(sentences)
```

In [76]:

```
sentences_as_one_string
```

Out[76]:

```
'Love my Echo! Loved it! Sometimes while playing a game, you can answer a
question correctly but Alexa says you got it wrong and answers the same as
you.  I like being able to turn lights on and off while away from home. I
have had a lot of fun with this thing. My 4 yr old learns about dinosaurs,
i control the lights and play games like categories. Has nice sound when p
laying music as well. Music I received the echo as a gift. I needed anothe
r Bluetooth or something to play music easily accessible, and found this s
mart speaker. Can’t wait to see what else it can do. Without having a cell
phone, I cannot use many of her features. I have an iPad but do not see th
at of any use.  It IS a great alarm.  If u r almost deaf, you can hear her
alarm in the bedroom from out in the living room, so that is reason enough
to keep her.It is fun to ask random questions to hear her response.  She d
oes not seem to be very smartbon politics yet. I think this is the 5th one
I\'ve purchased. I\'m working on getting one in every room of my house. I
really like what features they offer specifily playing music on all Echos
and controlling the lights throughout my house. looks great Love it! I’ve
listened to songs I haven’t heard since childhood! I get the news, weathe
r, information! It’s great! I sent it to my 85 year old Dad, and he talks
```

In [77]:

```
pip install wordcloud
```

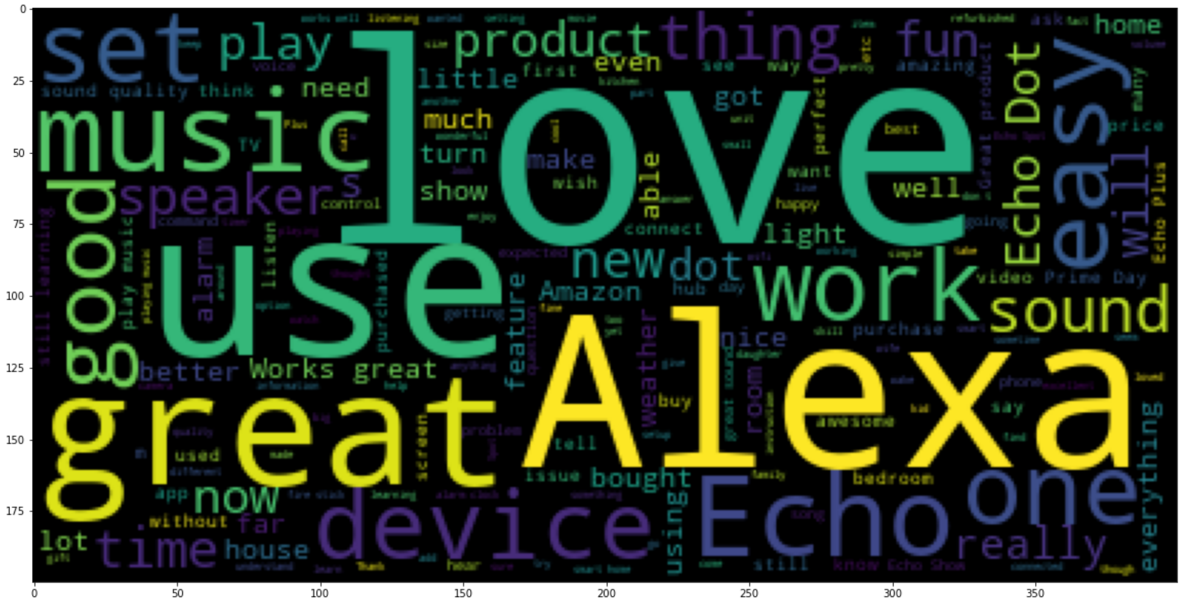
```
Requirement already satisfied: wordcloud in m:\anac\lib\site-packages (1.8.
1)
Requirement already satisfied: pillow in m:\anac\lib\site-packages (from wor
dcloud) (8.4.0)
Requirement already satisfied: numpy>=1.6.1 in m:\anac\lib\site-packages (fr
om wordcloud) (1.20.3)
Requirement already satisfied: matplotlib in m:\anac\lib\site-packages (from
wordcloud) (3.4.3)
Requirement already satisfied: kiwisolver>=1.0.1 in m:\anac\lib\site-package
s (from matplotlib->wordcloud) (1.3.1)
Requirement already satisfied: cycler>=0.10 in m:\anac\lib\site-packages (fr
om matplotlib->wordcloud) (0.10.0)
Requirement already satisfied: pyparsing>=2.2.1 in m:\anac\lib\site-packages
(from matplotlib->wordcloud) (3.0.4)
Requirement already satisfied: python-dateutil>=2.7 in m:\anac\lib\site-pack
ages (from matplotlib->wordcloud) (2.8.2)
Requirement already satisfied: six in m:\anac\lib\site-packages (from cycler
>=0.10->matplotlib->wordcloud) (1.16.0)
Note: you may need to restart the kernel to use updated packages.
```

In [78]:

```
from wordcloud import WordCloud

plt.figure(figsize=(20,20))
plt.imshow(WordCloud().generate(sentences_as_one_string))
```

Out[78]:

```
<matplotlib.image.AxesImage at 0x1c9c1e7e730>
```

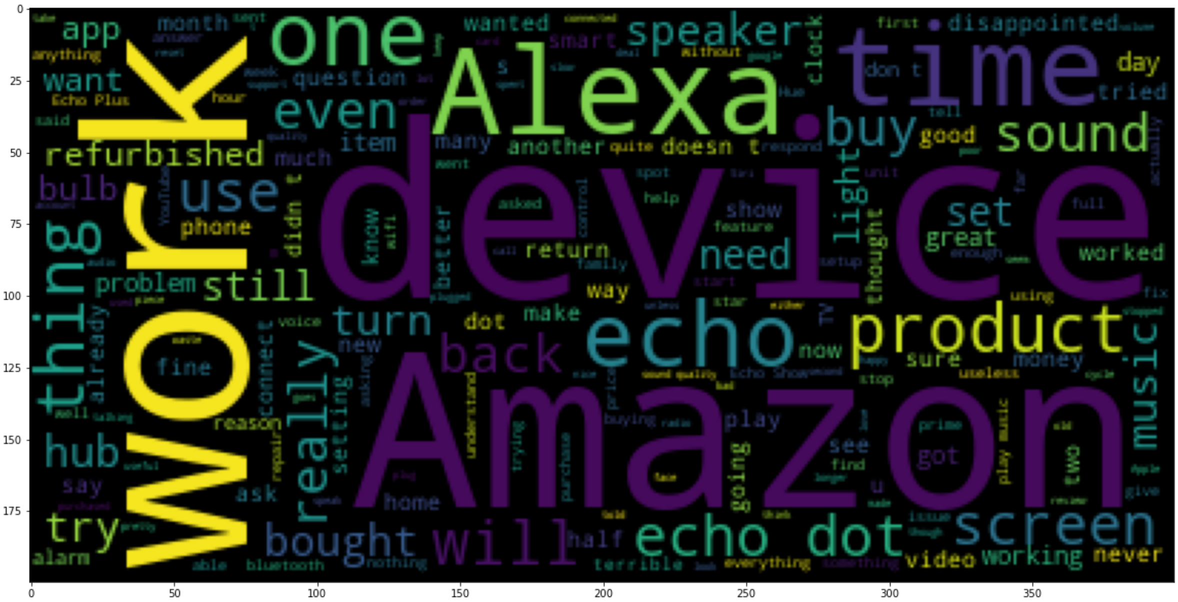


In [79]:

```
sentences = negative['verified_reviews'].tolist()
len(sentences)
sentences_as_one_string =" ".join(sentences)
plt.figure(figsize = (20,20))
plt.imshow(WordCloud().generate(sentences_as_one_string))
```

Out[79]:

```
<matplotlib.image.AxesImage at 0x1c9c1e1d7f0>
```



PERFORM DATA CLEANING

In [80]:

```
# Let's define a pipeline to clean up all the messages
# The pipeline performs the following: (1) remove punctuation, (2) remove stopwords

def message_cleaning(message):
    Test_punc_removed = [char for char in message if char not in string.punctuation]
    Test_punc_removed_join = ''.join(Test_punc_removed)
    Test_punc_removed_join_clean = [word for word in Test_punc_removed_join.split() if word
    return Test_punc_removed_join_clean
```

In [81]:

```
# Let's test the newly added function
reviews_df_clean = reviews_df['verified_reviews'].apply(message_cleaning)
```

In [82]:

```
# show the original review
print(reviews_df['verified_reviews'][5])
```

```
I received the echo as a gift. I needed another Bluetooth or something to pl
ay music easily accessible, and found this smart speaker. Can’t wait to see
what else it can do.
```

In [83]:

```
# show the cleaned up version
print(reviews_df_clean[5])
```

```
['received', 'echo', 'gift', 'needed', 'another', 'Bluetooth', 'something',
'play', 'music', 'easily', 'accessible', 'found', 'smart', 'speaker', 'Ca
n’t', 'wait', 'see', 'else']
```

In [84]:

```
from sklearn.feature_extraction.text import CountVectorizer
# Define the cleaning pipeline we defined earlier
vectorizer = CountVectorizer(analyzer = message_cleaning)
reviews_countvectorizer = vectorizer.fit_transform(reviews_df['verified_reviews'])
```

In [85]:

```
print(vectorizer.get_feature_names())
```

```
['072318', '1', '10', '100', '1000', '100X', '1010', '1030pm', '11', '1100
sf', '1220', '129', '12am', '15', '150', '19', '1964', '1990s', '1990’s',
'1GB', '1rst', '1st', '2', '20', '200', '2000', '2017', '2030', '229', '2
3', '2448', '247', '24GHZ', '24ghz', '25', '29', '299', '2999', '2Origina
l', '2nd', '2or', '2package', '3', '30', '300', '30so', '334', '34', '342n
d', '3434', '34A34', '34Alexa', '34Alexa34', '34Certified', '34Computer3
4', '34Dot34', '34Drop', '34First', '34Hub', '34I', '34Im', '34NEVER', '34
Philips', '34Play', '34Second', '34Skills34', '34Tell', '34The', '34Thing
s', '34Thongs', '34Try', '34Whats', '34alexa34', '34card34', '34cycle', '3
4cycle34', '34fixes34', '34fun34', '34group34', '34hear34', '34hmm', '34hm
mm', '34it34', '34late', '34learn', '34light34', '34lights34', '34listen3
4', '34minor', '34outlet34', '34personal34', '34she34', '34show', '34smar
t', '34smart34', '34sorry', '34spying34', '34the', '34thick34', '34thing
s', '34this', '34trouble', '34try', '34turn', '34visual34', '34wake34me',
'34warehouse34', '35', '360', '39', '399', '3999', '3Dots', '3rd', '3xs',
'4', '40000', '45', '48', '4K', '4am', '4k', '4th', '5', '50', '54', '5GH
Z', '5GHz', '5am”', '5ghz', '5th', '6', '600', '62', '672', '6th', '7', '7
000', '70s', '75', '7900', '8', '80s', '81', '83', '85', '88', '888', '8G
B', '9', '90', '91', '911', '99', 'A1', 'A19', 'ABC', 'ABSOLUTELY', 'AF',
```

In [86]:

```
print(reviews_countvectorizer.toarray())
```

```
[[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]]
```

In [87]:

```
reviews_countvectorizer.shape
```

Out[87]:

```
(3150, 5211)
```

In [88]:

```
reviews = pd.DataFrame(reviews_countvectorizer.toarray())
```

In [89]:

```
X = reviews
```

In [90]:

```
y = reviews_df['feedback']
y
```

Out[90]:

```
0       1
1       1
2       1
3       1
4       1
       ..
3145    1
3146    1
3147    1
3148    1
3149    1
Name: feedback, Length: 3150, dtype: int64
```

In [ ]:

```

```

TRAIN AND TEST AI/ML MODELS

In [91]:

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2)
```

Using Naive Bayes Classifier

In [92]:

```
from sklearn.naive_bayes import MultinomialNB

NB_classifier = MultinomialNB()
NB_classifier.fit(X_train, y_train)
```

Out[92]:

```
MultinomialNB()
```

In [93]:

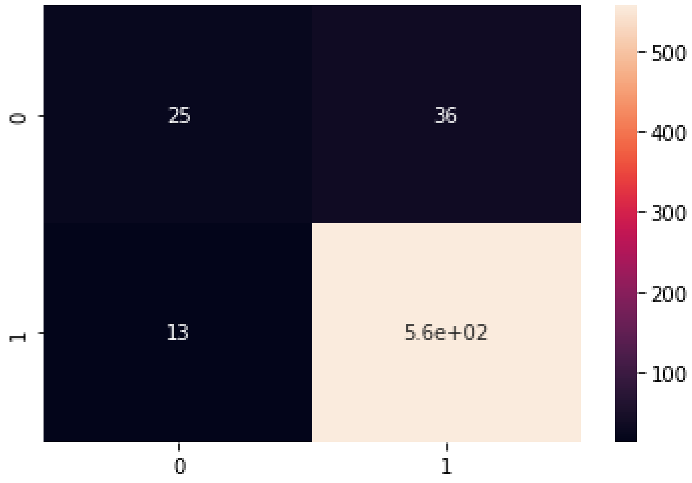
```
from sklearn.metrics import classification_report, confusion_matrix
```

In [94]:

```
# Predicting the Test set results
y_predict_test = NB_classifier.predict(X_test)
cm = confusion_matrix(y_test, y_predict_test)
sns.heatmap(cm, annot=True)
```

Out[94]:

```
<AxesSubplot:>
```



In [95]:

```
print(classification_report(y_test, y_predict_test))
```

```
              precision    recall  f1-score   support

           0       0.66      0.41      0.51        61
           1       0.94      0.98      0.96       569

    accuracy                           0.92       630
   macro avg       0.80      0.69      0.73       630
weighted avg       0.91      0.92      0.91       630
```

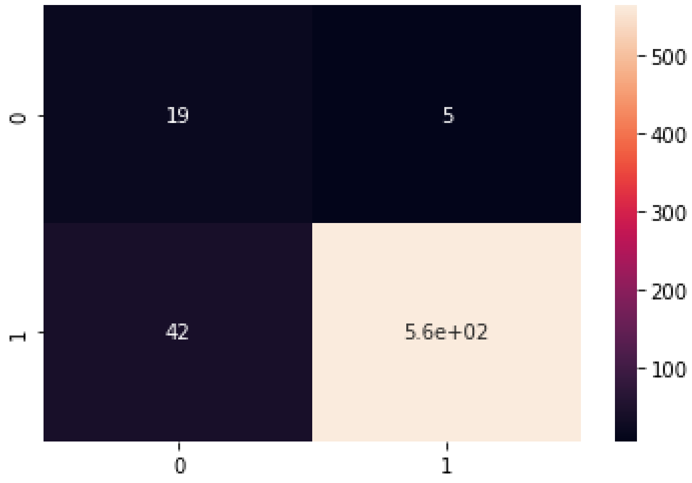
Using Logistic Regression

In [96]:

```
from sklearn.linear_model import LogisticRegression

model = LogisticRegression()
model.fit(X_train, y_train)

y_pred = model.predict(X_test)

cm = confusion_matrix(y_pred, y_test)
sns.heatmap(cm, annot = True)
```

Out[96]:

```
<AxesSubplot:>
```



In [97]:

```
print(classification_report(y_test, y_pred))
```

```
              precision    recall  f1-score   support

           0       0.79      0.31      0.45        61
           1       0.93      0.99      0.96       569

    accuracy                           0.93       630
   macro avg       0.86      0.65      0.70       630
weighted avg       0.92      0.93      0.91       630
```

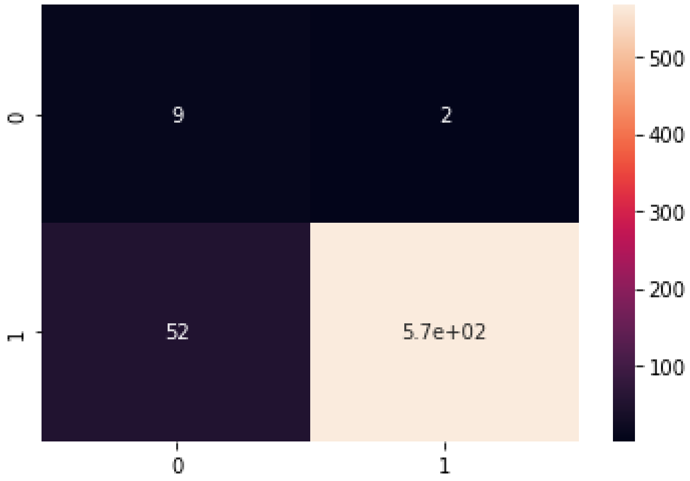
Using Gradient Boosting Classifier

In [98]:

```
from sklearn.ensemble import GradientBoostingClassifier

model = GradientBoostingClassifier()
model.fit(X_train, y_train)

y_pred = model.predict(X_test)

cm = confusion_matrix(y_pred, y_test)
sns.heatmap(cm, annot = True)
```

Out[98]:

```
<AxesSubplot:>
```



In [99]:

```
print(classification_report(y_test, y_pred))
```

```
              precision    recall  f1-score   support

           0       0.82      0.15      0.25        61
           1       0.92      1.00      0.95       569

    accuracy                           0.91       630
   macro avg       0.87      0.57      0.60       630
weighted avg       0.91      0.91      0.89       630
```

Result(Accuracies):

Naive Bayes Classifier = 0.92

logistic Regression = 0.93

Gradient Boosting = 0.91

---END--

In [ ]:

```

```