In [1]:

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

In [2]:

```
#importing dataset

reviews_df = pd.read_csv('D:/Data Science for Business Package/5. Public Relations D

```

In [3]:

```
reviews_df.head()
```

Out[3]:

	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

In [4]:

```
reviews_df.describe()
```

Out[4]:

	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 [5]:

```
#checking NULL Values
reviews_df.isnull().sum().sum()
```

Out[5]:

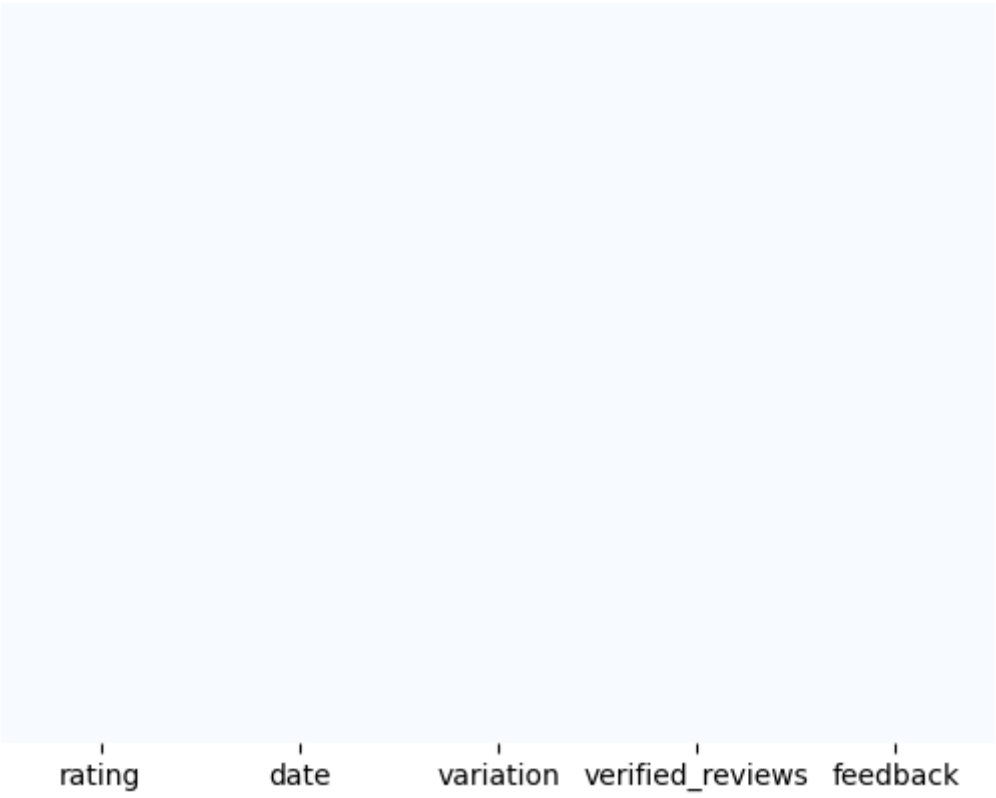
```
0
```

In [6]:

```
#visualiazing null values count if any
sns.heatmap(reviews_df.isnull(), yticklabels = False, cbar = False, cmap="Blues")
```

Out[6]:

```
<AxesSubplot: >
```



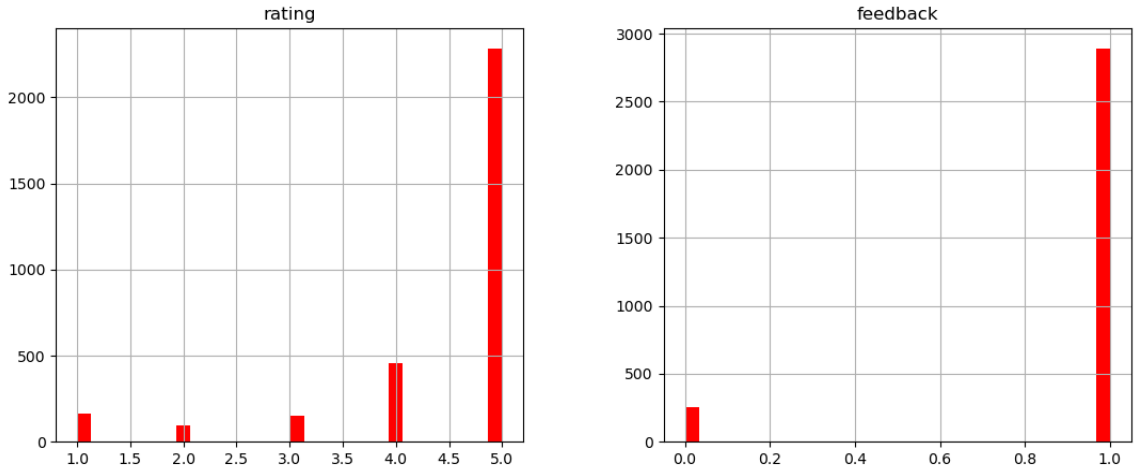
In [7]:

```
#plotting numerical column
reviews_df.hist(bins = 30, figsize = (13,5), color = 'r')

```

Out[7]:

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



In [8]:

```
#Length of the messages

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

```

Out[8]:

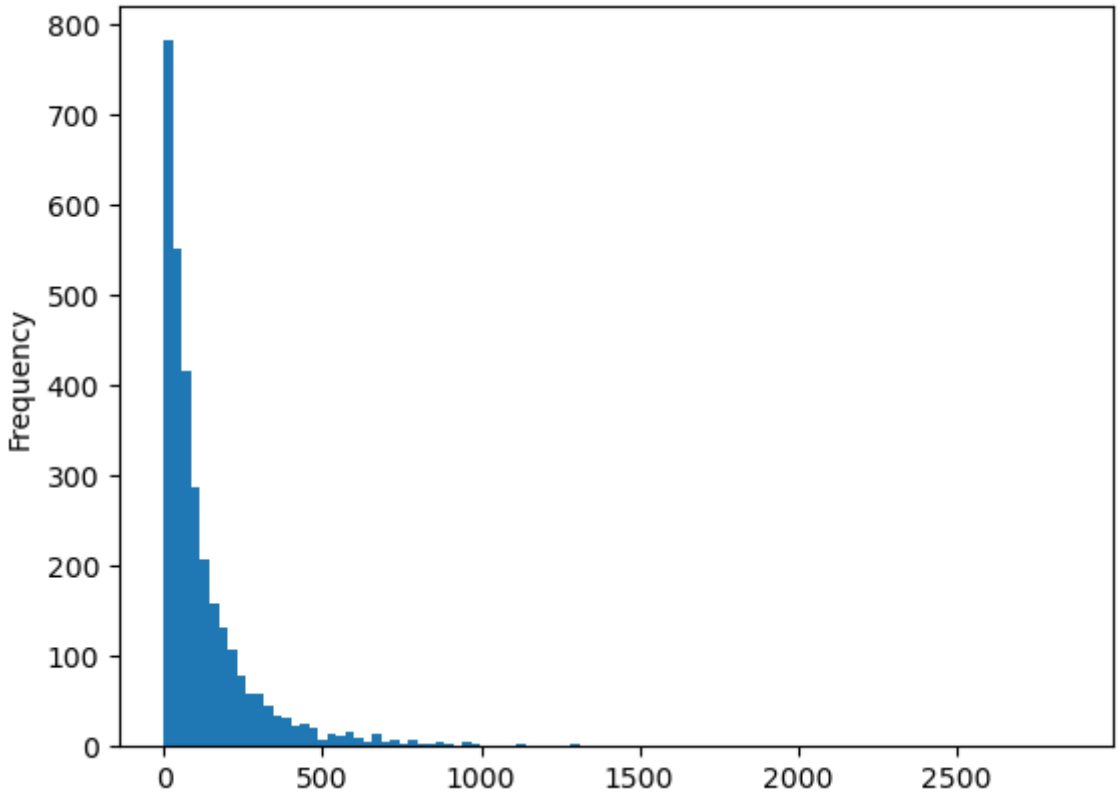
	rating	date	variation	verified_reviews	feedback	length
0	5	31-Jul-18	Charcoal Fabric	Love my Echo!	1	13
1	5	31-Jul-18	Charcoal Fabric	Loved it!	1	9
2	4	31-Jul-18	Walnut Finish	Sometimes while playing a game, you can answer...	1	195
3	5	31-Jul-18	Charcoal Fabric	I have had a lot of fun with this thing. My 4 ...	1	172
4	5	31-Jul-18	Charcoal Fabric	Music	1	5

In [9]:

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

Out[9]:

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



In [10]:

```
# Let's see the longest message
reviews_df[reviews_df['length'] == (reviews_df['length'].max())]['verified_reviews']
```

Out[10]:

"Incredible piece of technology.I have this right center of my living room on an island kitchen counter. The mic and speaker goes in every direction and the quality of the sound is quite good. I connected the Echo via Bluet ooth to my Sony soundbar on my TV but find the Echo placement and 360 soun d more appealing. It's no audiophile equipment but there is good range and decent bass. The sound is more than adequate for any indoor entertaining a nd loud enough to bother neighbors in my building. The knob on the top wor ks great for adjusting volume. This is my first Echo device and I would im agine having to press volume buttons (on the Echo 2) a large inconvenience and not as precise. For that alone I would recommend this over the regular Echo (2nd generation).The piece looks quality and is quite sturdy with som e weight on it. The rubber material on the bottom has a good grip on the g ranite counter-- my cat can even rub her scent on it without tipping it ov er.This order came with a free Philips Hue Bulb which I installed along wi th an extra one I bought. I put the 2 bulbs into my living room floor lam p, turned on the light, and all I had to do was say "Alexa, connect my devices". The default names for each bulb was assigned as "First l ight" and "Second light", so I can have a dimmer floor lamp if I just turned on/off one of the lights by saying "Alexa, turn off the second light". In the Alexa app, I created a 'Group' with "First l ight" and "Second light" and named the group "The light&#3 4;, so to turn on the lamp with both bulbs shining I just say "Alexa, turn on The light".I was surprised how easily the bulbs connected to t he Echo Plus with its built in hub. I thought I would have to buy a hub br idge to connect to my floor lamp power plug. Apparently there is some tech nology built directly inside the bulb! I was surprised by that. Awesome.Yo u will feel like Tony Stark on this device. I added quite a few "Skill s" like 'Thunderstorm sounds' and 'Quote of the day' . Alexa always lo ads them up quickly. Adding songs that you hear to specific playlists on A mazon Music is also a great feature.I can go on and on and this is only my second day of ownership.I was lucky to buy this for $100 on Prime Day, but I think for $150 is it pretty expensive considering the Echo 2 is only $10 0. In my opinion, you will be paying a premium for the Echo Plus and you h ave to decide if the value is there for you:1) Taller and 360 sound unit. 2) Volume knob on top that you spin (I think this is a huge benefit over b uttons)3) Built in hub for Hue bulbs. After researching more, there are so me cons to this setup if you plan on having more advanced light setups. Fo r me and my floor lamp, it's just perfect.I highly recommend it and will b uy an Echo dot for my bedroom now."

In [11]:

```
# Let's see the smallest message
reviews_df[reviews_df['length'] == (reviews_df['length'].min())]['verified_reviews']
```

Out[11]:

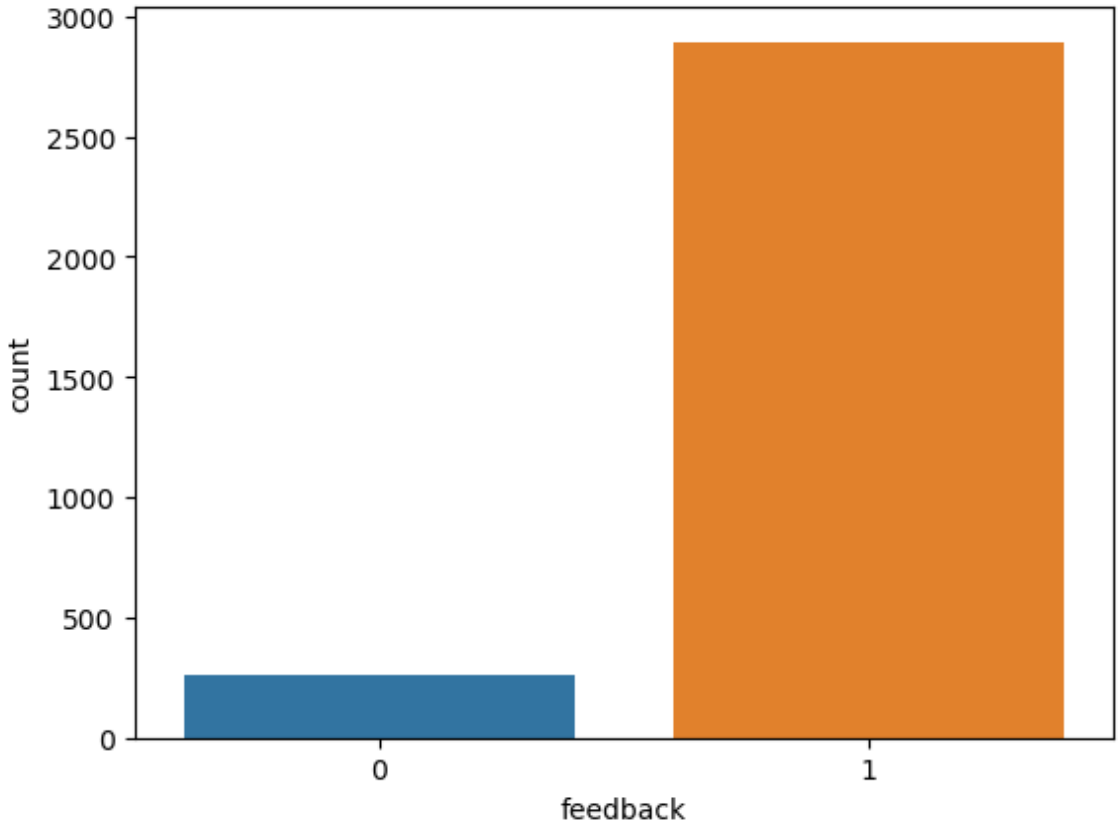
'😍'

In [12]:

```
#plotting feedback
sns.countplot(x='feedback', data = reviews_df)
```

Out[12]:

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



In [13]:

```
#positive feedback
positive = reviews_df[reviews_df['feedback']==1]
positive
```

Out[13]:

	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 [14]:

```
#negative feedback
negative = reviews_df[reviews_df['feedback']==0]
negative
```

Out[14]:

	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

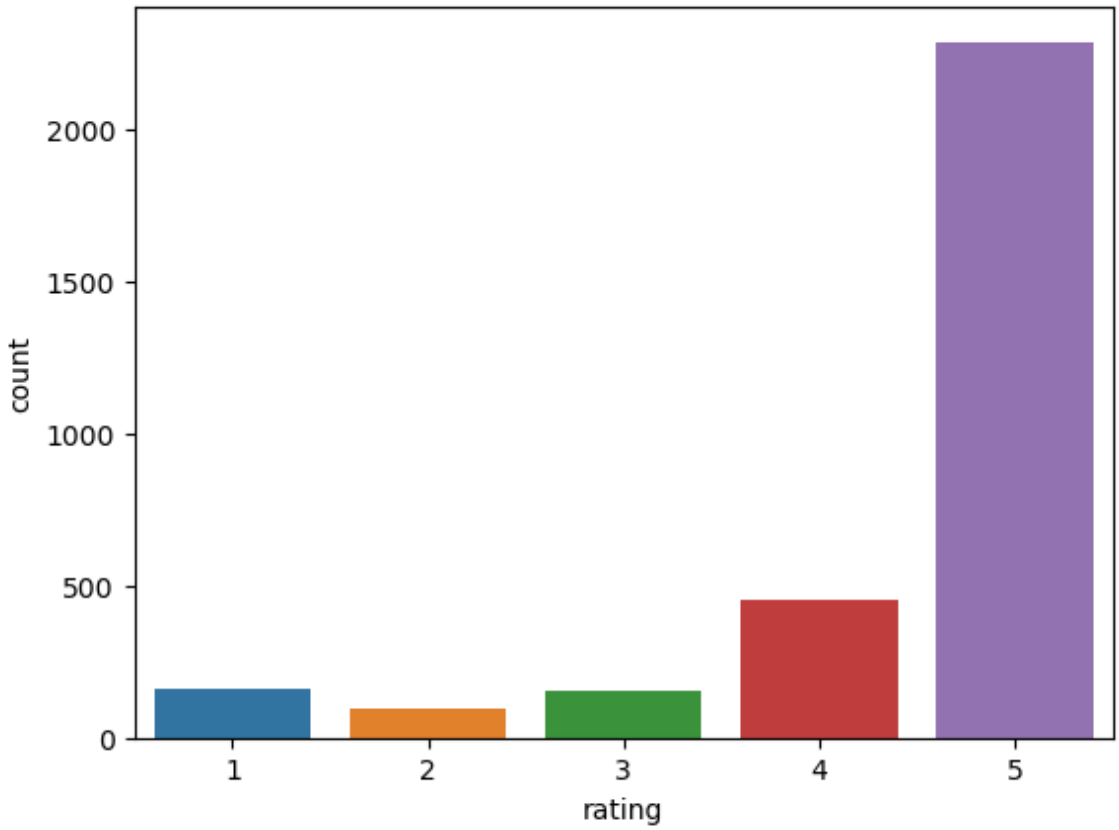
257 rows × 6 columns

In [15]:

```
#plotting rating column
sns.countplot(x = 'rating', data = reviews_df)
```

Out[15]:

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

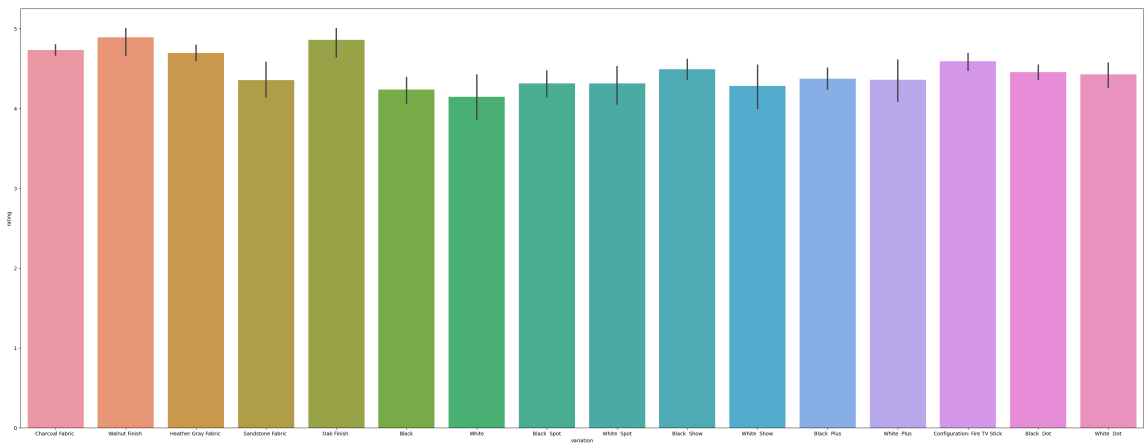


In [16]:

```
#visualizing variation vs rating
plt.figure(figsize = (40,15))
sns.barplot(x = 'variation', y='rating', data = reviews_df)
```

Out[16]:

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



In [17]:

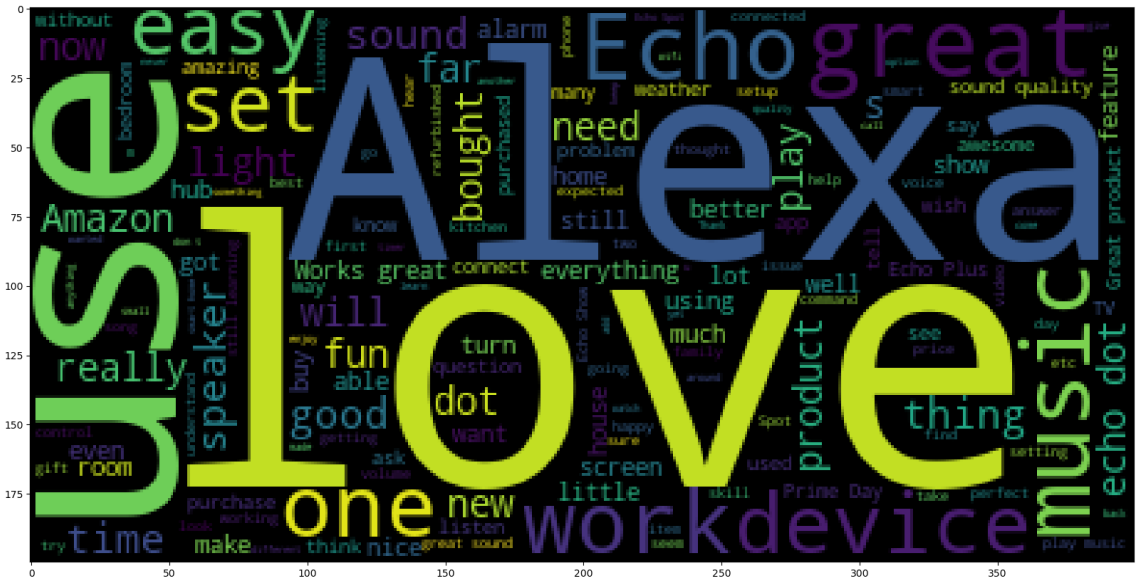
```
#converting verified_reviews into list
sentences = reviews_df['verified_reviews'].tolist()
#len(sentences)
sentences_as_one_string =" ".join(sentences)

from wordcloud import WordCloud

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

Out[17]:

```
<matplotlib.image.AxesImage at 0x1cb54a48a60>
```



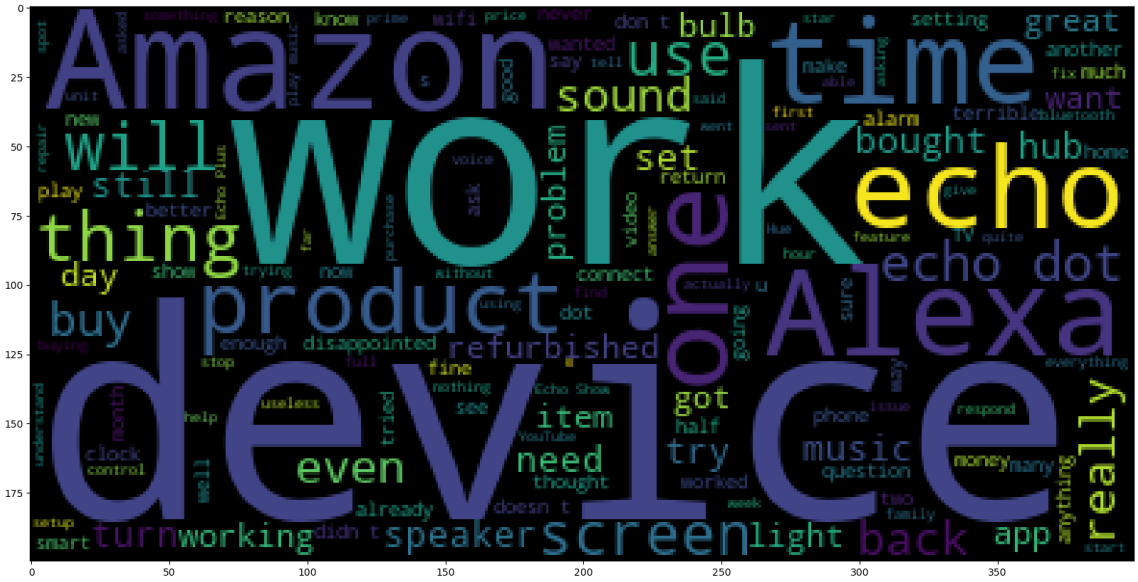
In [18]:

```
#converting negative verified_reviews into list
negative_list = negative['verified_reviews'].tolist()
negative_sentences_as_one_string = " ".join(negative_list)
plt.figure(figsize=(20,20))
plt.imshow(WordCloud().generate(negative_sentences_as_one_string))

```

Out[18]:

```
<matplotlib.image.AxesImage at 0x1cb5668a6a0>
```



In [19]:

```
#converting positive verified_reviews into list
positive_list = positive['verified_reviews'].tolist()
positive_sentences_as_one_string = " ".join(positive_list)
plt.figure(figsize=(20,20))
plt.imshow(WordCloud().generate(positive_sentences_as_one_string))

```

Out[19]:

```
<matplotlib.image.AxesImage at 0x1cb564151c0>
```

In [20]:

```
#Data Cleaning
reviews_df = reviews_df.drop(['date', 'rating', 'length'],axis=1)
reviews_df.head()
```

Out[20]:

	variation	verified_reviews	feedback
0	Charcoal Fabric	Love my Echo!	1
1	Charcoal Fabric	Loved it!	1
2	Walnut Finish	Sometimes while playing a game, you can answer...	1
3	Charcoal Fabric	I have had a lot of fun with this thing. My 4 ...	1
4	Charcoal Fabric	Music	1

In [21]:

```
#encoding
variation_dummies = pd.get_dummies(reviews_df['variation'], drop_first = True)
variation_dummies
```

Out[21]:

	Black Dot	Black Plus	Black Show	Black Spot	Charcoal Fabric	Configuration: Fire TV Stick	Heather Gray Fabric	Oak Finish	Sandstone Fabric	Wal Fin
0	0	0	0	0	1	0	0	0	0	
1	0	0	0	0	1	0	0	0	0	
2	0	0	0	0	0	0	0	0	0	
3	0	0	0	0	1	0	0	0	0	
4	0	0	0	0	1	0	0	0	0	
...	...	...	...	...	...	...	...	...	...	
3145	1	0	0	0	0	0	0	0	0	
3146	1	0	0	0	0	0	0	0	0	
3147	1	0	0	0	0	0	0	0	0	
3148	0	0	0	0	0	0	0	0	0	
3149	1	0	0	0	0	0	0	0	0	

3150 rows × 15 columns

In [22]:

```
#dropping variation
reviews_df.drop(['variation'], axis=1, inplace=True)
```

In [23]:

```
#concat two column
reviews_df = pd.concat([reviews_df, variation_dummies], axis=1)
reviews_df
```

Out[23]:

	verified_reviews	feedback	Black Dot	Black Plus	Black Show	Black Spot	Charcoal Fabric	Configuration: Fire TV Stick	Heatl Gı Fab
0	Love my Echo!	1	0	0	0	0	1	0	
1	Loved it!	1	0	0	0	0	1	0	
2	Sometimes while playing a game, you can answer...	1	0	0	0	0	0	0	
3	I have had a lot of fun with this thing. My 4 ...	1	0	0	0	0	1	0	
4	Music	1	0	0	0	0	1	0	
...	...	...	...	...	...	...	...	...	
3145	Perfect for kids, adults and everyone in betwe...	1	1	0	0	0	0	0	
3146	Listening to music, searching locations, check...	1	1	0	0	0	0	0	
3147	I do love these things, i have them running my...	1	1	0	0	0	0	0	
3148	Only complaint I have is that the sound qualit...	1	0	0	0	0	0	0	
3149	Good	1	1	0	0	0	0	0	

3150 rows × 17 columns

In [24]:

```
#countvector
#from sklearn.feature_extraction.text import CountVectorizer
#sample_data = ['This is the first document.','This document is the second document.

#vectorizer = CountVectorizer()
#X = vectorizer.fit_transform(sample_data)
#print(vectorizer.get_feature_names_out())
#print(X.toarray())
```

In [25]:

```
#performs (1) remove punctuation, (2) remove stopwords
import string
import nltk
from nltk.corpus import 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()
    return Test_punc_removed_join_clean
```

In [26]:

```
reviews_df_clean = reviews_df['verified_reviews'].apply(message_cleaning)
print(reviews_df_clean)
```

```
0                                          [Love, Echo]
1                                               [Loved]
2       [Sometimes, playing, game, answer, question, c...
3       [lot, fun, thing, 4, yr, old, learns, dinosaur...
4                                               [Music]
                              ...
3145                    [Perfect, kids, adults, everyone]
3146    [Listening, music, searching, locations, check...
3147    [love, things, running, entire, home, TV, ligh...
3148    [complaint, sound, quality, isnt, great, mostl...
3149                                             [Good]
Name: verified_reviews, Length: 3150, dtype: object
```

In [27]:

```
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'])
print(vectorizer.get_feature_names_out())
print("Array\n",reviews_countvectorizer.toarray())
print("\nShape",reviews_countvectorizer.shape)
```

```
['072318' '1' '10' ... '😬' '😳' '🤯']
Array
 [[0 0 0 ... 0 0 0]
 [0 0 0 ... 0 0 0]
 [0 0 0 ... 0 0 0]
 ...
 [0 0 0 ... 0 0 0]
 [0 0 0 ... 0 0 0]
 [0 0 0 ... 0 0 0]]

Shape (3150, 5211)
```

In [28]:

```
#Drop the column
reviews_df.drop(['verified_reviews'], axis=1, inplace=True)
#new dataframe
reviews = pd.DataFrame(reviews_countvectorizer.toarray())
#concatenate dataframes
reviews_df = pd.concat([reviews_df, reviews], axis=1)
reviews_df

```

Out[28]:

	feedback	Black Dot	Black Plus	Black Show	Black Spot	Charcoal Fabric	Configuration: Fire TV Stick	Heather Gray Fabric	Oak Finish	Sand F
0	1	0	0	0	0	1	0	0	0	
1	1	0	0	0	0	1	0	0	0	
2	1	0	0	0	0	0	0	0	0	
3	1	0	0	0	0	1	0	0	0	
4	1	0	0	0	0	1	0	0	0	
...	...	...	...	...	...	...	...	...	...	
3145	1	1	0	0	0	0	0	0	0	
3146	1	1	0	0	0	0	0	0	0	
3147	1	1	0	0	0	0	0	0	0	
3148	1	0	0	0	0	0	0	0	0	
3149	1	1	0	0	0	0	0	0	0	

3150 rows × 5227 columns

In [29]:

```
#data preparation
X = reviews_df.drop(['feedback'],axis=1)
X.columns = X.columns.astype(str)
y = reviews_df['feedback']
```

In [30]:

```
#splitting data
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)

#naive bayes classification
from sklearn.naive_bayes import MultinomialNB

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

Out[30]:

MultinomialNB

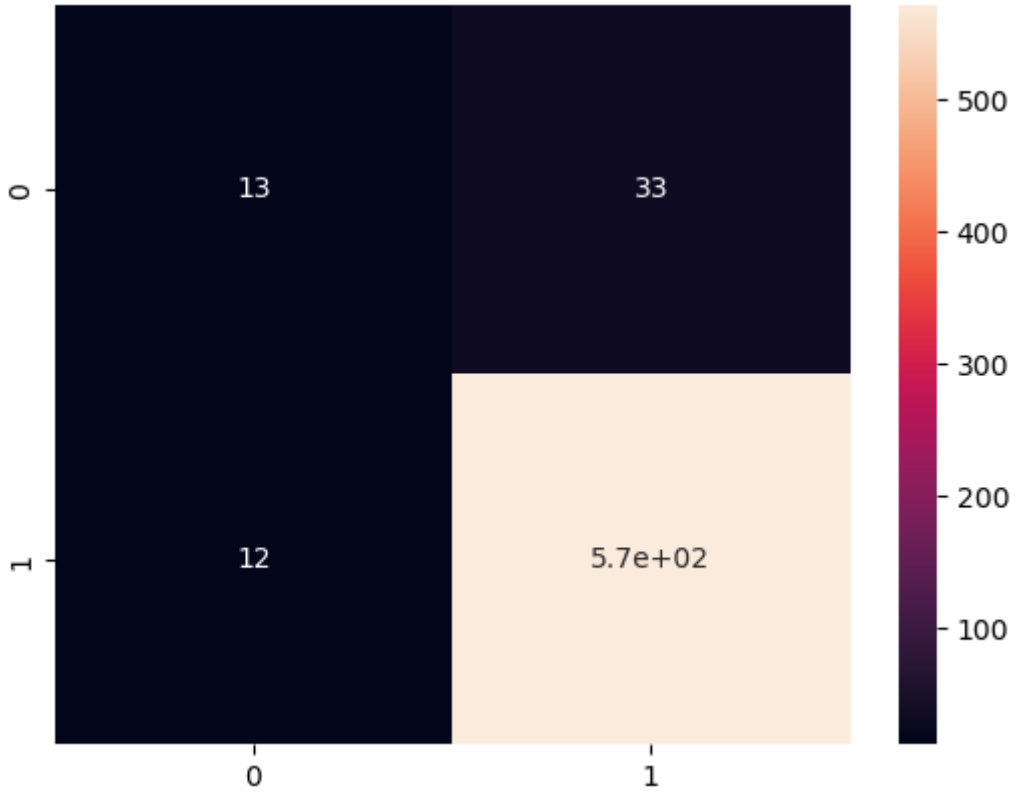
MultinomialNB()

In [31]:

```
#evaluation
from sklearn.metrics import classification_report, confusion_matrix
y_predict_test = NB_classifier.predict(X_test)
cm = confusion_matrix(y_test, y_predict_test)
sns.heatmap(cm, annot=True)
```

Out[31]:

<AxesSubplot: >



In [32]:

```
#Logistic Regression
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score
model = LogisticRegression()
model.fit(X_train, y_train)
y_pred = model.predict(X_test)
```

In [33]:

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

print('Accuracy {} %'.format( 100 * accuracy_score(y_pred, y_test)))
```

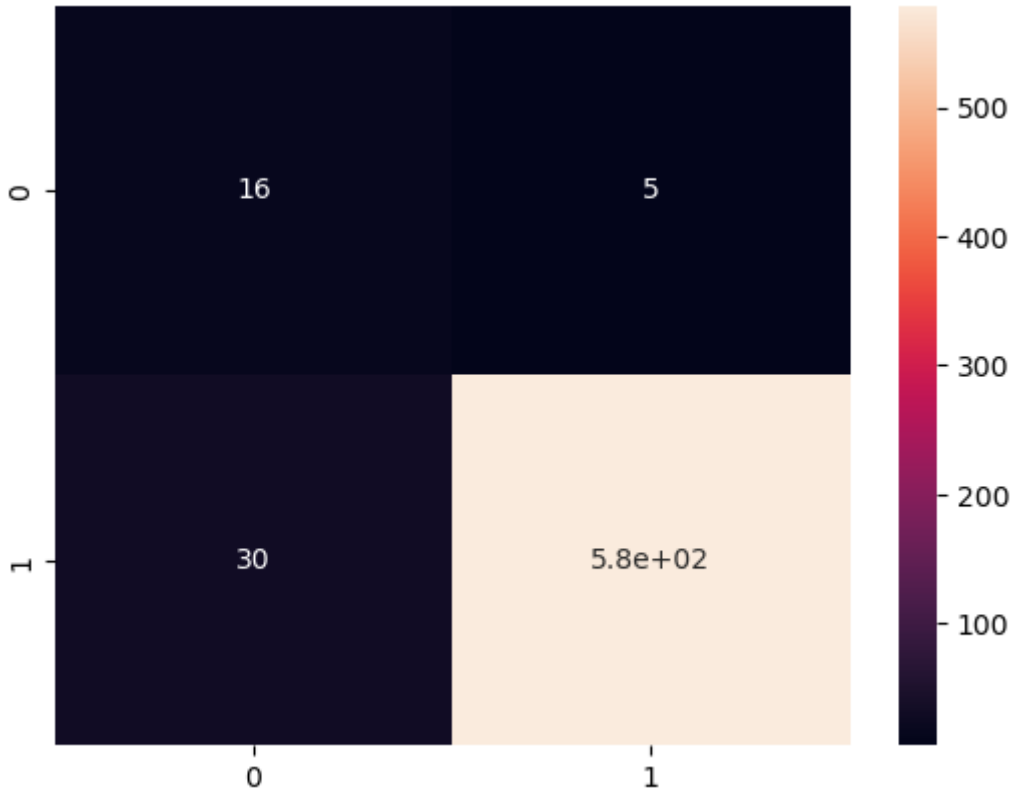
```
Accuracy 94.44444444444444 %
```

In [34]:

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

Out[34]:

```
<AxesSubplot: >
```



In [35]:

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

```

```
              precision    recall  f1-score   support

           0       0.76      0.35      0.48        46
           1       0.95      0.99      0.97       584

    accuracy                           0.94       630
   macro avg       0.86      0.67      0.72       630
weighted avg       0.94      0.94      0.93       630
```

In [ ]:

```

```