```
import pandas as pd
import numpy as np

df = pd.read_csv("Alexa-Dataset.csv")

df.head()
```

```
   rating       date         variation  \
0       5  31-Jul-18  Charcoal Fabric
1       5  31-Jul-18  Charcoal Fabric
2       4  31-Jul-18    Walnut Finish
3       5  31-Jul-18  Charcoal Fabric
4       5  31-Jul-18  Charcoal Fabric

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

```
df.shape
```

```
(3150, 5)
```

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

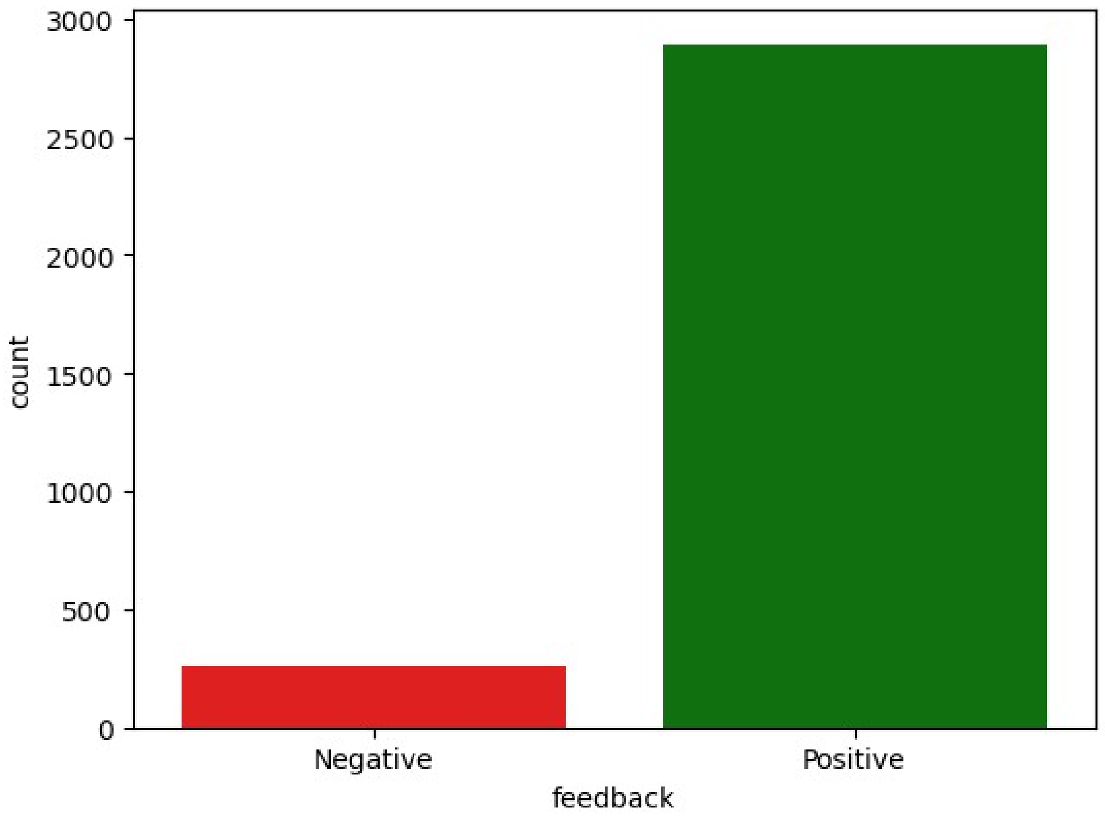
```
rating              0
date                0
variation           0
verified_reviews    1
feedback            0
dtype: int64
```

Plot a graph of Positive and Negative Feedback (1 = Positive Feedback, 0 = Negative Feedback)

```
import matplotlib.pyplot as plt
import seaborn as sns

sns.countplot(x=df["feedback"], palette=["red", "green"])
plt.xticks(ticks=[0, 1], labels=["Negative", "Positive"])
```

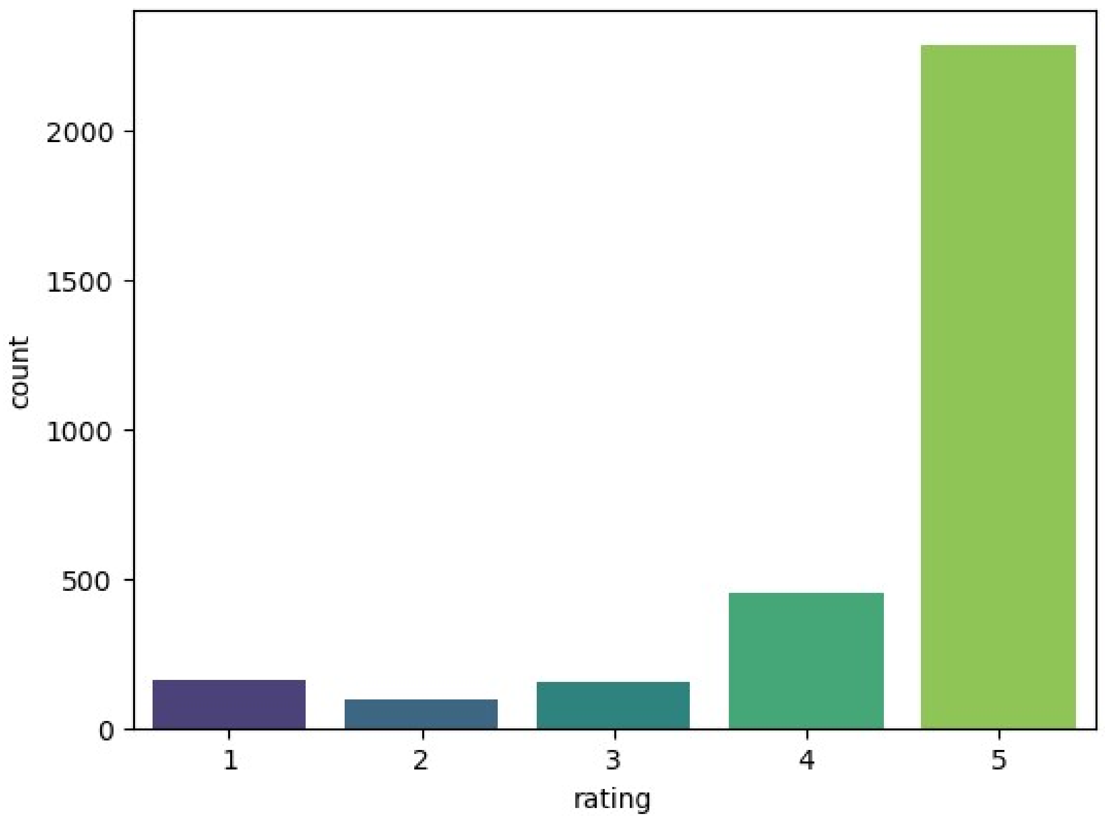
```
C:\Users\Harish\AppData\Local\Temp\ipykernel_15584\4214687043.py:1:
FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be
removed in v0.14.0. Assign the `x` variable to `hue` and set
`legend=False` for the same effect.

  sns.countplot(x=df["feedback"], palette=["red", "green"])
```

```
([<matplotlib.axis.XTick at 0x1b9c2415e20>,
  <matplotlib.axis.XTick at 0x1b9c2415d00>],
 [Text(0, 0, 'Negative'), Text(1, 0, 'Positive')])
```



Plot the graph of Ratings distribution.

```
sns.countplot(x=df["rating"], palette="viridis")

C:\Users\Harish\AppData\Local\Temp\ipykernel_15584\1266232429.py:1:
FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be
removed in v0.14.0. Assign the `x` variable to `hue` and set
`legend=False` for the same effect.

  sns.countplot(x=df["rating"], palette="viridis")

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



```
import re
import string
import nltk
```

Convert the review text into lowercase

```
Alexa_dataset_lower = df.apply(lambda x: x.astype(str).str.lower())
Alexa_dataset_lower.head()

  rating       date         variation  \
0      5  31-jul-18  charcoal fabric
1      5  31-jul-18  charcoal fabric
2      4  31-jul-18    walnut finish
3      5  31-jul-18  charcoal fabric
4      5  31-jul-18  charcoal fabric

                                    verified_reviews feedback
0                                      love my echo!        1
1                                          loved it!        1
2  sometimes while playing a game, you can answer...        1
3  i have had a lot of fun with this thing. my 4 ...        1
4                                              music        1
```

Remove all punctuations from review text.

```
import string
string.punctuation
```

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

In below code first check if text is a string or not and if it is not string then return empty value

```
import string

def remove_punctuation(text):
    if isinstance(text, str):
        return "".join([char for char in text if char not in
string.punctuation])
    return ""

df["clean_msg"] = df["verified_reviews"].apply(remove_punctuation)

df["clean_msg"].head()
```

```
0                                                Love my Echo
1                                                    Loved it
2    Sometimes while playing a game you can answer ...
3    I have had a lot of fun with this thing My 4 y...
4                                                       Music
Name: clean_msg, dtype: object
```

Remove emoticons and emojis from the text

```
def preprocess_text(text):
    if isinstance(text, str):
        return re.sub(r"[^\w\s]", "", text)
    return ""
df["cleaned_reviews"] = df["verified_reviews"].apply(preprocess_text)

df["cleaned_reviews"].head()
```

```
0                                                Love my Echo
1                                                    Loved it
2    Sometimes while playing a game you can answer ...
3    I have had a lot of fun with this thing My 4 y...
4                                                       Music
Name: cleaned_reviews, dtype: object
```

Tokenize the review text into words.

```
def tokenization(text):
    tokens = re.split('W+',text)
    return tokens
df['msg_tokenied']= df['cleaned_reviews'].apply(lambda x:
tokenization(x))
```

```
df['msg_tokenied']

0                                          [Love my Echo]
1                                              [Loved it]
2       [Sometimes while playing a game you can answer...
3       [I have had a lot of fun with this thing My 4 ...
4                                                 [Music]
                              ...
3145    [Perfect for kids adults and everyone in between]
3146    [Listening to music searching locations checki...
3147    [I do love these things i have them running my...
3148    [Only complaint I have is that the sound quali...
3149                                               [Good]
Name: msg_tokenied, Length: 3150, dtype: object
```

Remove the Stopwords from the tokenized text.

```
from nltk.corpus import stopwords
nltk.download('stopwords')

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

True

STOPWORDS = set(stopwords.words('english'))
def remove_stopwords(text):
    """custom function to remove the stopwords"""
    return " ".join([word for word in str(text).split() if word not in
STOPWORDS])

df["review_stop"] = df["msg_tokenied"].apply(lambda text:
remove_stopwords(text))
df["review_stop"]

0                                            ['Love Echo']
1                                             ['Loved it']
2       ['Sometimes playing game answer question corre...
3       ['I lot fun thing My 4 yr old learns dinosaurs...
4                                              ['Music']
                              ...
3145             ['Perfect kids adults everyone between']
3146    ['Listening music searching locations checking...
3147    ['I love things running entire home TV lights ...
3148    ['Only complaint I sound quality isnt great I ...
3149                                             ['Good']
Name: review_stop, Length: 3150, dtype: object
```

Perform stemming & lemmatization on the review text.

```
from nltk.stem.porter import PorterStemmer

stemmer = PorterStemmer()
def stem_words(text):
    return " ".join([stemmer.stem(word) for word in text.split()])

df["review_stemmed"] = df["review_stop"].apply(lambda text:
stem_words(text))
df["review_stemmed"]

0                                             ['love echo']
1                                               ['love it']
2       ['sometim play game answer question correctli ...
3       ['i lot fun thing my 4 yr old learn dinosaur c...
4                                                 ['music']
                              ...
3145                 ['perfect kid adult everyon between']
3146    ['listen music search locat check time look we...
3147    ['i love thing run entir home tv light thermos...
3148    ['onli complaint i sound qualiti isnt great i ...
3149                                               ['good']
Name: review_stemmed, Length: 3150, dtype: object

from nltk.stem import WordNetLemmatizer
nltk.download('wordnet')

[nltk_data] Downloading package wordnet to
[nltk_data]     C:\Users\Harish\AppData\Roaming\nltk_data...
[nltk_data]   Package wordnet is already up-to-date!

True

lemmatizer = WordNetLemmatizer()
def lemmatize_words(text):
    return " ".join([lemmatizer.lemmatize(word) for word in
text.split()])

df["review_lemmatized"] = df["review_stop"].apply(lambda text:
lemmatize_words(text))

df["review_lemmatized"]

0                                             ['Love Echo']
1                                              ['Loved it']
2       ['Sometimes playing game answer question corre...
3       ['I lot fun thing My 4 yr old learns dinosaur ...
4                                                 ['Music']
                              ...
3145                ['Perfect kid adult everyone between']
3146    ['Listening music searching location checking ...
3147    ['I love thing running entire home TV light th...
```

```
3148    ['Only complaint I sound quality isnt great I ...
3149                                             ['Good']
Name: review_lemmatized, Length: 3150, dtype: object
```

Perform the word vectorization on review text using Bag of Words technique.

```
from sklearn.feature_extraction.text import CountVectorizer

reviews = df["review_lemmatized"].astype(str)

bow_vectorizer = CountVectorizer()
bow_matrix = bow_vectorizer.fit_transform(reviews)

bow_matrix.shape
```

```
(3150, 4264)
```

Create representation of Review Text by calculating Term Frequency and Inverse Document Frequency (TF-IDF)

```
from sklearn.feature_extraction.text import TfidfVectorizer

tfidf_vectorizer = TfidfVectorizer()
tfidf_matrix = tfidf_vectorizer.fit_transform(reviews)

tfidf_matrix.shape
```

```
(3150, 4264)
```