Import Necessary Libraries

In [40]:
```
import pandas as pd
import numpy as np
import re
import matplotlib.pyplot as plt
from wordcloud import WordCloud
from nltk.tokenize import word_tokenize
from nltk.corpus import stopwords
import nltk
from nltk.stem import WordNetLemmatizer
from nltk.corpus import stopwords
wordnet = WordNetLemmatizer()
import re
from nltk.tokenize import sent_tokenize
from sklearn.feature_extraction.text import TfidfVectorizer
import requests
from bs4 import BeautifulSoup as bs
```

In [41]:
```
nltk.download('stopwords')
nltk.download('punkt')
nltk.download('wordnet')
```

```
[nltk_data] Downloading package stopwords to
[nltk_data]     C:\Users\91998\AppData\Roaming\nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
[nltk_data] Downloading package punkt to
[nltk_data]     C:\Users\91998\AppData\Roaming\nltk_data...
[nltk_data]   Package punkt is already up-to-date!
[nltk_data] Downloading package wordnet to
[nltk_data]     C:\Users\91998\AppData\Roaming\nltk_data...
[nltk_data]   Package wordnet is already up-to-date!
```

Out[41]: True

Data collection

2.1 connecting amazon and download reviews

In [42]:
```
oneplus_reviews=[]
```

In [43]:
```
for i in range (1,31):
    op=[]
    url ="https://www.amazon.in/Test-Exclusive-747/product-reviews/B07DJCVTDN/ref=cm_cr_dp_d_show_all_btm?ie=UTF8&reviewerType=all_reviews"+str(i)
    response = requests.get(url)
    soup = bs(response.content,"html.parser")
    reviews = soup.findAll("span",attrs = {"class","a-size-base review-text review-text-content"})
    for i in range(len(reviews)):
        op.append(reviews[i].text)
    oneplus_reviews = oneplus_reviews+op
```

2.2 Downloading review from Amazon.in for the phone oneplus 7

In [44]:
```
oneplus_reviews = list(set(oneplus_reviews))
```

Cleaning the text

In [45]:
```
txt_upd = ' '.join(oneplus_reviews)
```

In [46]:
```
txt_upd = re.sub("[^A-Za-z" "]+"," ",txt_upd).lower() #remove special character
txt_upd = re.sub("[0-9" "]+"," ",txt_upd).lower() #remove numbers
txt_upd = re.sub(r'^https?:\/\/.*[\r\n]*', '', txt_upd).lower() #remove hyperlink
```

In [47]:
```
text_tokens = word_tokenize(txt_upd)
```

In [48]:
```
tokens_without_sw = [word for word in text_tokens if not word in stopwords.words()]
```

Create the DF

In [18]:
```
tf = TfidfVectorizer()
```

In [19]:
```
text_tf = tf.fit_transform(tokens_without_sw)
```

In [21]:
```
feature_names = tf.get_feature_names()
dense = text_tf.todense()
denselist = dense.tolist()
df =pd.DataFrame(denselist, columns=feature_names)
```

In [22]:
```
df
```

Out[22]:

	adapter	adaptor	added	always	amazing	amazon	ambiance	amoeld	amoled	android	...	work	worried	worst	would	wounds	writing	year	years	youtube	zoom
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	0.0	0.0	0.0
1	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	0.0	0.0
2	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	0.0	0.0
3	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	0.0	0.0
4	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	0.0	0.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
879	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	0.0	0.0
880	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	0.0	0.0
881	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	0.0	0.0
882	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	0.0	0.0
883	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	0.0	0.0

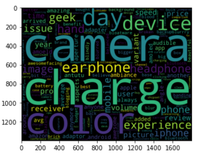
884 rows × 510 columns

In [23]:
```
word_list = ' '.join(df)
```

In [24]:
```
wordcloud = WordCloud(background_color='black',
                      width=1800,
                      height=1400).generate(word_list)
```

In [25]:
```
plt.imshow(wordcloud)
```

Out[25]: <matplotlib.image.AxesImage at 0x2e0acfd8520>



sentimental Analysis

In [27]:
```
with open("positive-words.txt","r") as pw:
    positive_words = pw.read().split("\n")
```

In [28]:
```
positive_words = positive_words[35:]
```

In [29]:
```
with open("negative-words.txt","r", encoding='latin-1') as nw:
    negative_words = nw.read().split("\n")
```

In [30]:
```
negative_words = negative_words[35:]
```

In [31]:
```
txt_neg_in_nw = ' '.join([word for word in df if word in negative_words])
```

In [32]:
```
wordcloud_neg = WordCloud(
                      background_color='black',
                      width=1800,
                      height=1400
                     ).generate(txt_neg_in_nw)
```

In [33]:
```
txt_pos_in_pw = ' '.join([word for word in df if word in positive_words])
```

In [34]:
```
wordcloud_pos = WordCloud(
                      background_color='black',
                      width=1800,
                      height=1400
                     ).generate(txt_pos_in_pw)
```

Conclusion

In [35]:
```
plt.imshow(wordcloud_neg)
```

Out[35]: <matplotlib.image.AxesImage at 0x2e0adece5e0>

In [36]:
```
plt.imshow(wordcloud_pos)
```

Out[36]: <matplotlib.image.AxesImage at 0x2e0adf2c8e0>

In [ ]:
```
```