

TASK #1: INTRODUCTION TO GUIDED PROJECT

PROJECT CARD

GOAL:

- *Build, train, test and deploy an Artificial intelligence (AI) model to predict sentiment from thousands of tweets. Sentiment prediction involves understanding of people feelings about a product or service.*

TOOLS:

- *Anaconda, Python, Scikit-Learn, Matplotlib, Seaborn*

PRACTICAL REAL-WORLD APPLICATION:

- *AI/ML sentiment analysis tools empower companies to automatically predict whether their customers are happy or not. The process could be done automatically without having humans manually review thousands of tweets/reviews.*

DATA:

- **INPUTS:**
 - *Twitter tweets (text data)*
- **OUTPUT:**
 - *Sentiment (0 or 1)*



Photo Credit: <https://www.stockvault.net/photo/259626/customer-satisfaction-survey>

- Natural language processors (NLP) work by converting words (text) into numbers.
- These numbers are then used to train an AI/ML model to make predictions.
- Predictions could be sentiment inferred from social media posts and product reviews.
- AI/ML-based sentiment analysis is crucial for companies to automatically predict whether their customers are happy or not.
- The process could be done automatically without having humans manually review thousands of tweets and customer reviews.
- In this case study, we will analyze thousands of Twitter tweets to predict people's sentiment.

TWEET



"Good morning everyone! Such a beautiful sunny day!"

"Don't fly on xx airline, their customer service is horrendous"

SENTIMENT

POSITIVE (LABEL = 0)

NEGATIVE (LABEL = 1)

SENTIMENT
ANALYSIS
(NLP MODEL)



AI-POWERED NLP APPLICATIONS

- Check MonkeyLearn and watch the video: <https://monkeylearn.com/>

Social Media Monitoring to understand what makes customers angry/happy

Customer Service with Personalized messages

Brand Monitoring over multiple outlets such as Facebook, Instagram, Google maps..etc.

Market Research and analysis of company's competition

Analyze Employees voice on Glassdoor for example

ENHANCED GUIDED PROJECT FEATURES



Project includes several practice opportunities with detailed solution explanation in a video format



Reading materials and a quiz are provided so students can understand the fundamentals of Natural Language Processing (NLP) techniques and their practical applications in the industry.



A final practical portfolio project is provided so students can showcase their skills in AI/ML to future employers. Students will be provided with a brand-new dataset and are asked to apply 3 different AI/ML algorithms.

```
In [1]: import pandas as pd  
import numpy as np  
import seaborn as sns  
import matplotlib.pyplot as plt  
from jupyterthemes import jtplot  
jtplot.style(theme='monokai', context='notebook', ticks=True, grid=False)  
# setting the style of the notebook to be monokai theme  
# this line of code is important to ensure that we are able to see the x and y axes clearly  
# If you don't run this code line, you will notice that the xlabel and ylabel on any plot is black on black
```

```
In [2]: # Load the data  
tweets_df = pd.read_csv('twitter.csv')
```

```
In [3]: tweets_df
```

	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?ðŸ˜ðŸ˜ðŸ˜ðŸ˜ðŸ˜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()
```

	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']
```

```
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?ðŸ˜ðŸ˜ðŸ˜ðŸ˜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 #3: PERFORM DATA EXPLORATION

In [8]:

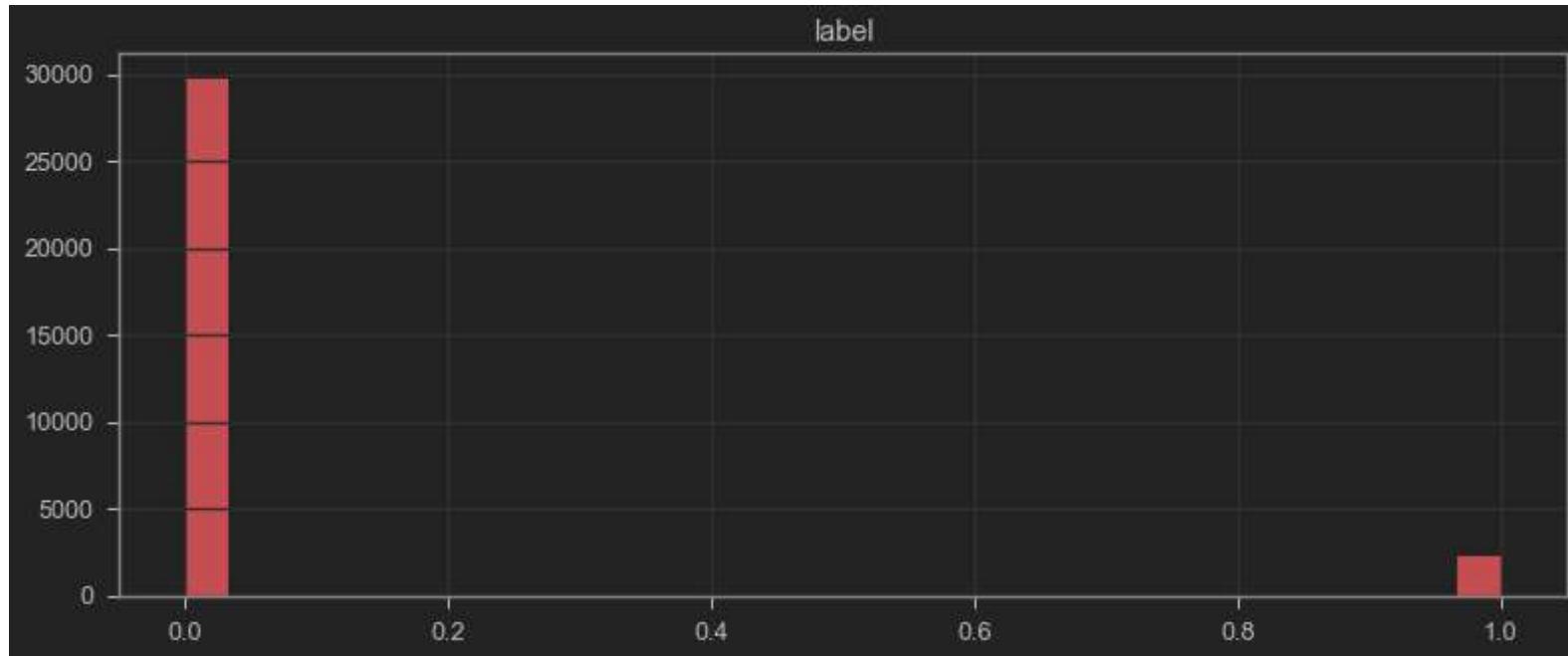
```
sns.heatmap(tweets_df.isnull(), yticklabels = False, cbar = False, cmap="Blues")
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x13f8f6e0748>
```



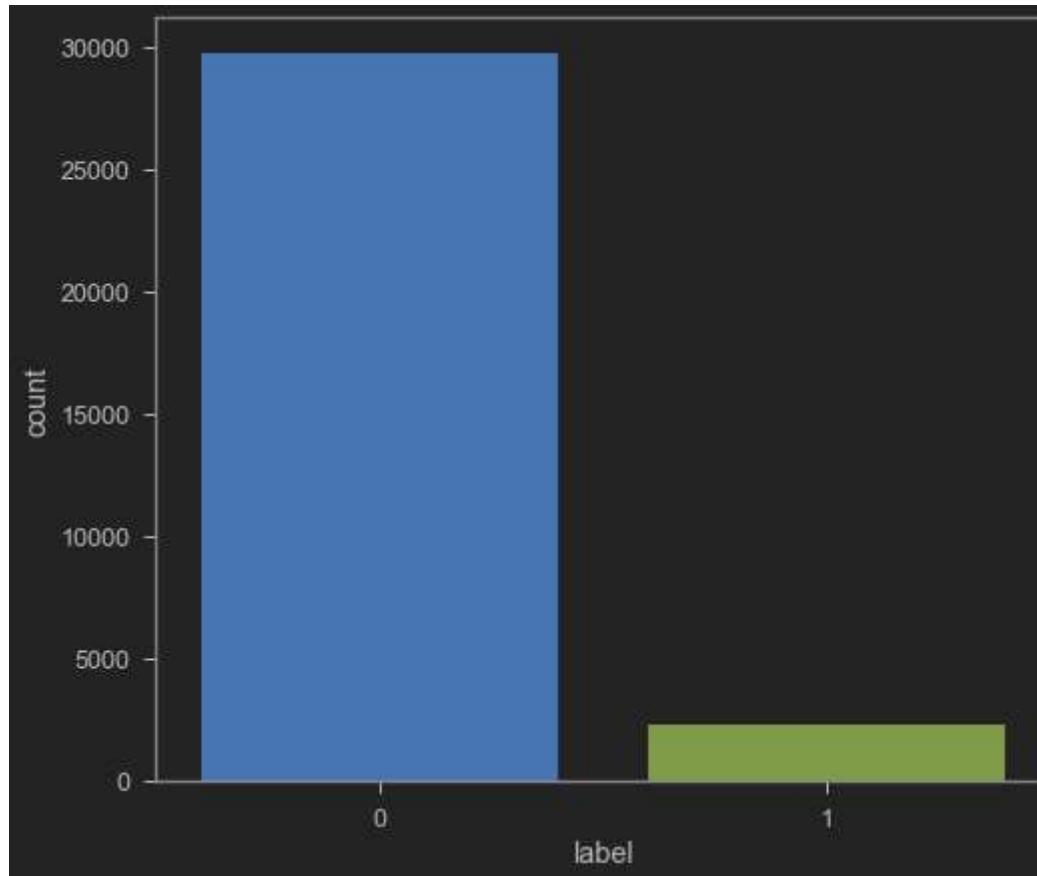
```
In [9]: tweets_df.hist(bins = 30, figsize = (13,5), color = 'r')
```

```
array([[<matplotlib.axes._subplots.AxesSubplot object at 0x0000013F91B35B08>]],  
      dtype=object)
```



```
In [10]: sns.countplot(tweets_df['label'], label = "Count")
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x13f91c05bc8>
```



```
In [11]: # Let's get the length of the messages  
tweets_df['length'] = tweets_df['tweet'].apply(len)
```

In [12]: | tweets df

	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?ðŸ˜ðŸ˜ðŸ˜ðŸ˜ðŸ˜ðŸ˜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()
```

	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]

'i love you '
```

PRACTICE OPPORTUNITY #2:

PRACTICE OPPORTUNITY

1. Using “tweets_df” DataFrame, view the tweet that has the average number of characters
2. Plot the histogram of the “length” column

```
In [15]: # Let's view the message with mean length
```

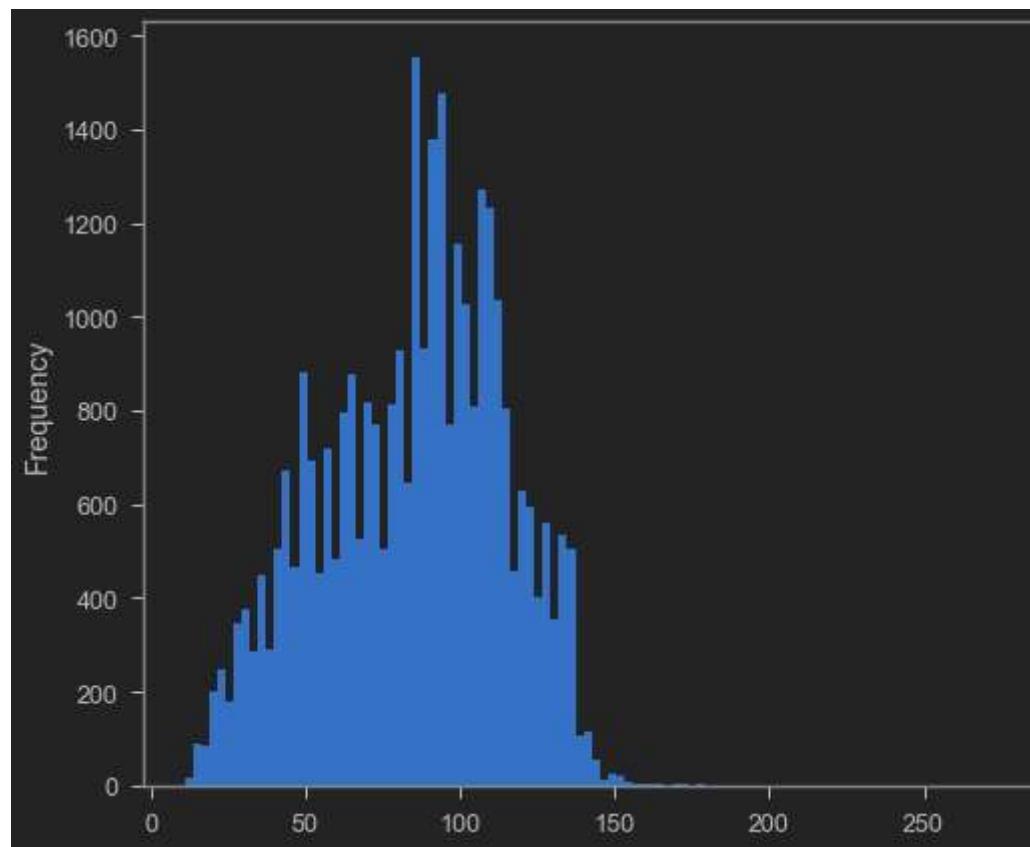
```
tweets_df[tweets_df['length'] == 84]['tweet'].iloc[0]
```

```
'my mom shares the same bihday as @user    bihday snake! see you this weekend ð\x9f\x99\x8cð\x9f\x8f%'
```

```
In [16]: # Plot the histogram of the Length column
```

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

```
<matplotlib.axes._subplots.AxesSubplot at 0x13f92271d48>
```



TASK #4: PLOT THE WORDCLOUD

```
In [17]: positive = tweets_df[tweets_df['label']==0]  
positive
```

	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?ðŸ˜ðŸ˜ðŸ˜ðŸ˜ðŸ˜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
```

	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 #jcpen...	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)
```

31962

```
In [20]: sentences_as_one_string = ".join(sentences)
```

```
In [21]: sentences_as_one_string
```

' @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 #getthanked 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; factsguid e: society now #motivation [2/2] huge fan fare and big talking before they leave. chaos and pay disputes when they get there. #allsho wandnogo @user camping tomorrow @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 land!!! #allin #cavs #champions #cleveland #clevelandcavaliers â\x80; @user @user welcome here ! i'm it's so #gr8 ! â\x86\x9d #ir eland consumer price index (mom) climbed from previous 0.2% to 0.5% in may #blog #silver #gold #forex we are so selfish. #orlando #sta ndwithorlando #pulseshooting #orlandoshooting #biggerproblems #selfish #heabreaking #values #love # i get to see my daddy today!! #8 0days #gettingfed @user #cnn calls #michigan middle school \'build the wall\' chant \'\' #tcot no comment! in #australia #opkilling bay #seashepherd #helpcovedolphins #thecove #helpcovedolphins ouch...junior is angryð\x9f\x98\x90#got7 #junior #yugyoem #omg i am thankful for having a paner. #thankful #positive retweet if you agree! its #friday! ð\x9f\x98\x80 smiles all around via ig user: @user #cookies make people as we all know, essential oils are not made of chemicals. #euro2016 people blaming ha for conceded goal was it fat rooney who gave away free kick knowing bale can hit them from there. sad little dude.. #badday #coneofshame #cats #pissed #funny #laughs product of the day: happy man #wine tool who's it's the #weekend? time to open up &; drink up! @user @user lumpy says i am a . prove it lumpy. @user #tgif #ff to my #gamedev #indiedev #indiegamedev #squad! @user @user @user beautiful sign by vendor 80 for \$45.00!! #upsideofflorida #shopalysas #love @user all #smiles when #media is !! ð\x9f\x98\x9cð\x9f\x98\x88 #pressconference in #antalya #turkey ! sunday #throwback love! ð\x9f\x98\x8að\x9f\x98\x98â\x9dñi,\x8f we had a great panel on the mediat ization of the public service #ica16 happy father's day @user ð\x9f\x92\x93ð\x9f\x92\x93ð\x9f\x92\x93ð\x9f\x92\x93ð\x9f\x92\x93 50 people went t o nightclub to have a good night and 1 man's actions means those people are lost to their families forever #rip#orlando i have never ha d a chance to vote for a presidential candidate i was excited about and this cycle looks to be no different. #alohafriday #time does #not #exist #positivevibes #hawaiian @user @user @user rip to the fellow nohern ireland fan who sadley passed away tonigh t!.. gawa, forever singing and cheering on fire it was a hard monday due to cloudy weather. disabling oxygen production for today. #go odnight #badmonday it's unbelievable that in the 21st century we'd need something like this again #teaventure #xenophobia #tau lone

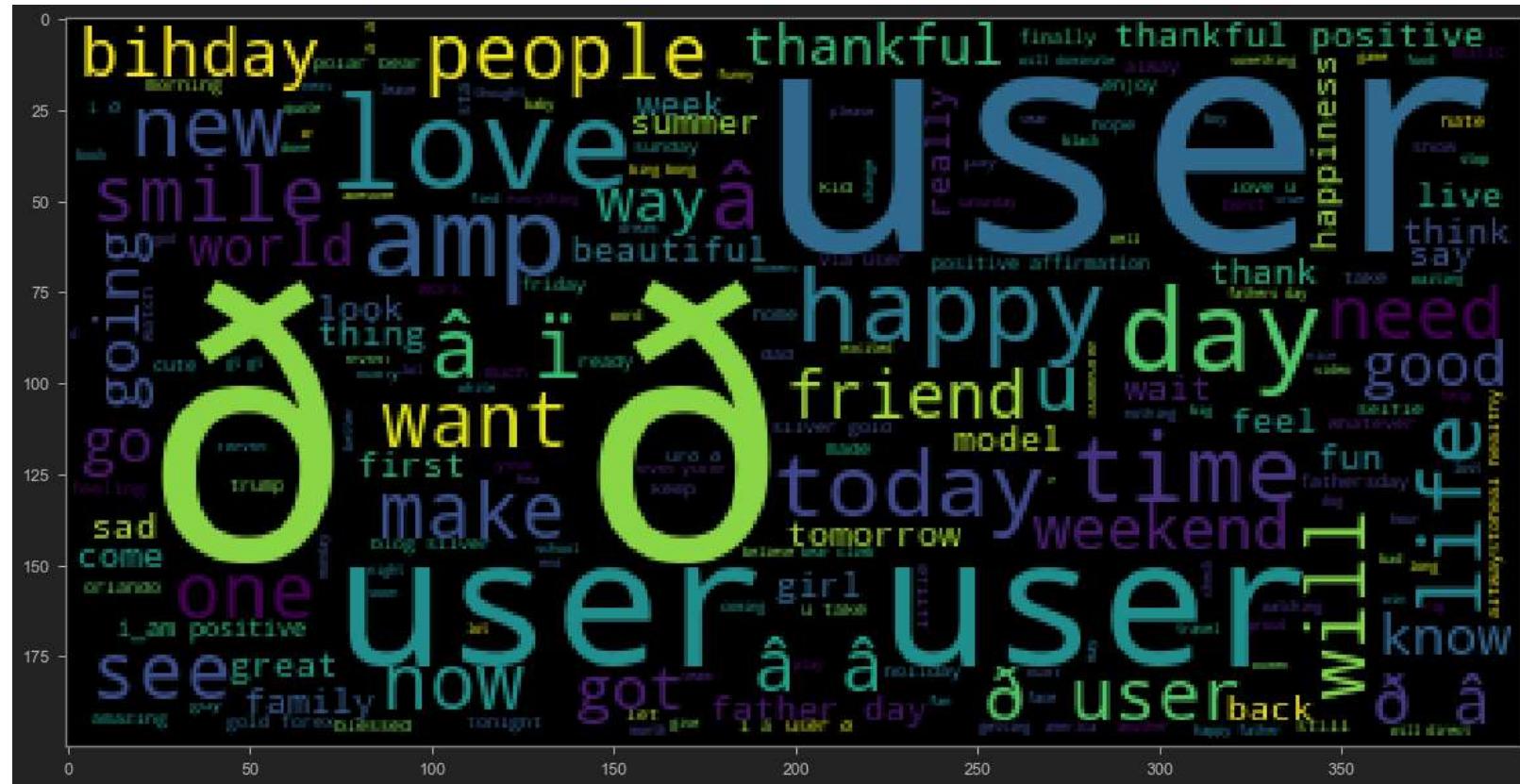
```
In [22]: !pip install wordcloud
```

```
Requirement already satisfied: wordcloud in c:\users\administrator\anaconda31\lib\site-packages (1.8.1)
Requirement already satisfied: pillow in c:\users\administrator\anaconda31\lib\site-packages (from wordcloud) (7.0.0)
Requirement already satisfied: matplotlib in c:\users\administrator\anaconda31\lib\site-packages (from wordcloud) (3.1.3)
Requirement already satisfied: numpy>=1.6.1 in c:\users\administrator\anaconda31\lib\site-packages (from wordcloud) (1.18.1)
Requirement already satisfied: cycler>=0.10 in c:\users\administrator\anaconda31\lib\site-packages (from matplotlib->wordcloud) (0.10.0)
Requirement already satisfied: kiwisolver>=1.0.1 in c:\users\administrator\anaconda31\lib\site-packages (from matplotlib->wordcloud) (1.1.0)
Requirement already satisfied: pyparsing!=2.0.4,!=2.1.2,!=2.1.6,>=2.0.1 in c:\users\administrator\anaconda31\lib\site-packages (from matplotlib->wordcloud) (2.4.6)
Requirement already satisfied: python-dateutil>=2.1 in c:\users\administrator\anaconda31\lib\site-packages (from matplotlib->wordcloud) (2.8.1)
Requirement already satisfied: six in c:\users\administrator\anaconda31\lib\site-packages (from cycler>=0.10->matplotlib->wordcloud) (1.14.0)
Requirement already satisfied: setuptools in c:\users\administrator\anaconda31\lib\site-packages (from kiwisolver>=1.0.1->matplotlib->wordcloud) (45.2.0.post20200210)
```

```
In [23]: from wordcloud import WordCloud
```

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

```
<matplotlib.image.AxesImage at 0x13f92dc6cc8>
```

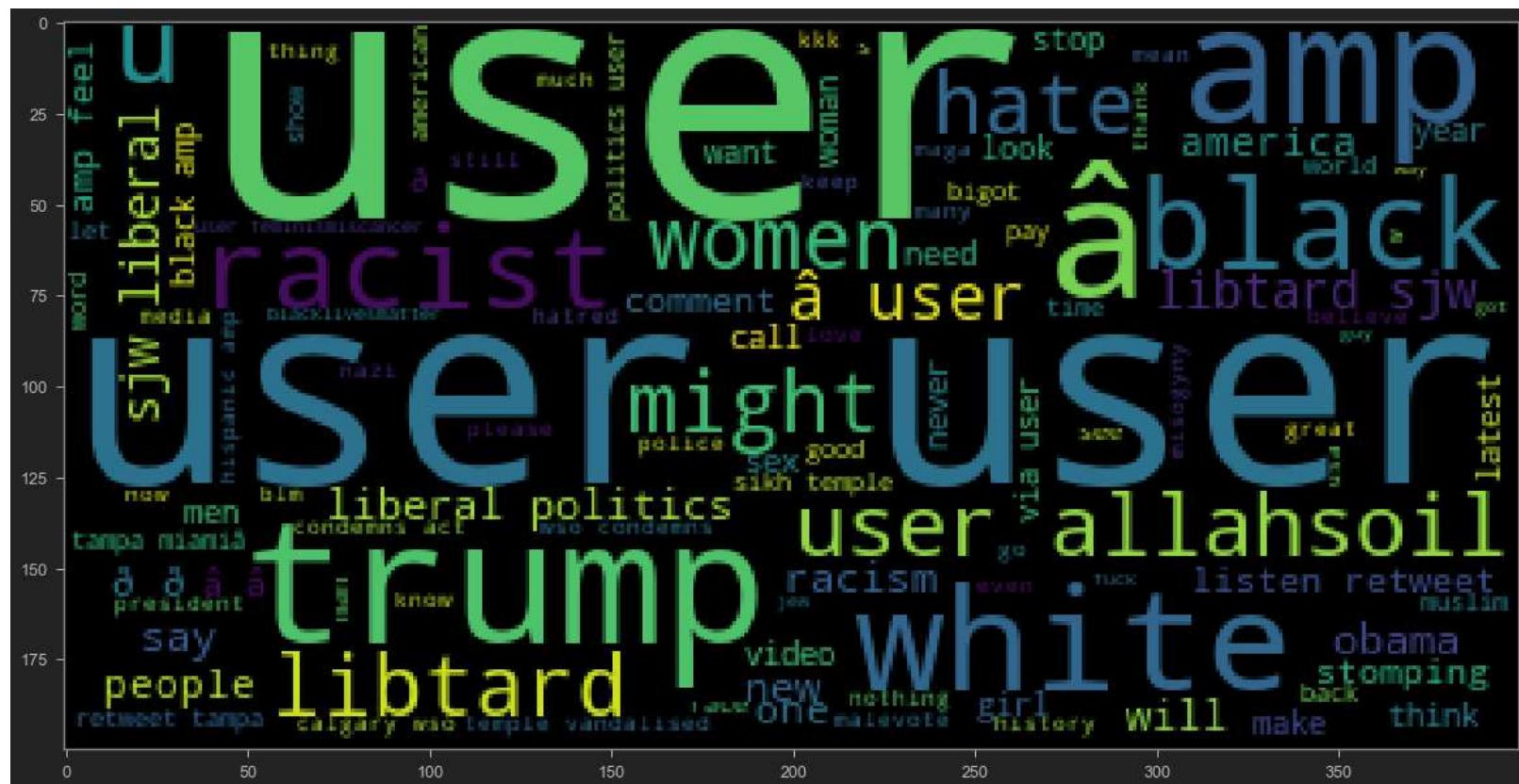


PRACTICE OPPORTUNITY #3:

PRACTICE OPPORTUNITY

1. Using the tweets_df DataFrame, plot the wordcloud of the "negative" dataframe.
2. What do you notice? Does the data make sense?

```
In [24]: 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))
```



TASK #5: PERFORM DATA CLEANING - REMOVE PUNCTUATION FROM TEXT

```
In [25]: import string  
string.punctuation
```

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

```
In [26]: 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
```

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

```
In [27]: Test = 'Good morning beautiful people :)... I am having fun learning Machine learning and AI!!'
```

```
In [28]: Test_punc_removed = [char for char in Test if char not in string.punctuation]
Test_punc_removed
```

```
[ '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 [29]: # Join the characters again to form the string.
```

```
Test_punc_removed_join = ''.join(Test_punc_removed)
```

```
Test_punc_removed_join
```

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

TASK 6: PERFORM DATA CLEANING - REMOVE STOPWORDS

```
In [30]: 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
[nltk_data]     C:\Users\Administrator\AppData\Roaming\nltk_data...
[nltk_data]     Unzipping corpora\stopwords.zip.

['i',
 'me',
 'my',
 'myself',
 'we',
 'our',
 'ours',
 'ourselves',
 'you',
 "you're",
 "you've",
 "you'll",
 "you'd",
 'your',
 'yours',
 'yourself',
 'yourselves',
 'he',
 'him',
```

```
In [31]: 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.lower() not in stopw
```

```
In [32]: Test_punc_removed_join_clean # Only important (no so common) words are left  
  
['enjoy', 'coding,', 'programming', 'Artificial', 'intelligence']
```

```
In [33]: Test_punc_removed_join  
  
'I enjoy coding, programming and Artificial intelligence'
```

PRACTICE OPPORTUNITY #4:

PRACTICE OPPORTUNITY

For the following text, create a pipeline to remove punctuations followed by removing stopwords

mini_challenge = 'Here is a mini challenge, that will teach you how to remove stopwords and punctuations!'

```
In [34]: mini_challenge = 'Here is a mini challenge, that will teach you how to remove stopwords and punctuations!'
```

```
In [35]: # Remove punctuations
challenge = [ char for char in mini_challenge if char not in string.punctuation ]
challenge = ''.join(challenge)
challenge
```

```
'Here is a mini challenge that will teach you how to remove stopwords and punctuations'
```

```
In [36]: challenge = [ word for word in challenge.split() if word.lower() not in stopwords.words('english') ]
challenge
['mini', 'challenge', 'teach', 'remove', 'stopwords', 'punctuations']
```

TASK 7: PERFORM COUNT VECTORIZATION (TOKENIZATION)

TOKENIZATION (COUNT VECTORIZER)

This is the first paper.
This paper is the second paper.
And this is the third one.
Is this the first paper?



```
[[0 1 1 1 0 0 1 0 1]
 [0 2 0 1 0 1 1 0 1]
 [1 0 0 1 1 0 1 1 1]
 [0 1 1 1 0 0 1 0 1]]
```

	'and'	'paper'	'first'	'is'	'one'	'second'	'the'	'third'	'this'
Training Sample #1	0	1	1	1	0	0	1	0	1
Training Sample #2	0	2	0	1	0	1	1	0	1
Training Sample #3	1	0	0	1	1	0	1	1	1
Training Sample #4	0	1	1	1	0	0	1	0	1

```
In [37]: from sklearn.feature_extraction.text import CountVectorizer  
sample_data = ['This is the first paper.', 'This document is the second paper.', 'And this is the third one.'  
  
vectorizer = CountVectorizer()  
X = vectorizer.fit_transform(sample_data)
```

```
In [38]: print(vectorizer.get_feature_names())
```

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

```
In [39]: 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]]
```

PRACTICE OPPORTUNITY #5:

PRACTICE OPPORTUNITY

1. Without doing any code, perform count vectorization for the following list:

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

2. - Confirm your answer with code

```
In [40]: mini_challenge = ['Hello World','Hello Hello World','Hello World world world']

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

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

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

TASK #8: CREATE A PIPELINE TO REMOVE PUNCTUATIONS, STOPWORDS AND PERFORM COUNT VECTORIZATION

```
In [41]: # 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.lower() not in :
        return Test_punc_removed_join_clean
```

```
In [42]: # Let's test the newly added function
```

```
tweets_df_clean = tweets_df['tweet'].apply(message_cleaning)
```

```
In [43]: print(tweets_df_clean[5]) # show the cleaned up version
```

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

```
In [44]: print(tweets_df['tweet'][5]) # show the original version
```

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

```
In [46]: 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 [47]:

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

```
['0', '0000001', '00027', '001', '0035', '00h30', '01', '0115', '0161', '019', '01926889917', '02', '0265', '0266808099', '02900', '03', '030916', '03111880779', '032', '033', '0345', '039', '04', '045', '04k', '05', '0506823156', '06', '06052016', '0606', '060616', '0608', '0608wed', '0609', '0610', '061116', '0612', '0613', '0616', '0617', '0618', '0618saturday7months couple', '0618â\x99j', '0620', '062016', '0622', '0624', '06Â', '07', '07000', '07040', '07044', '07150', '07190', '07400', '07468', '07500', '076', '07788427999', '07800', '07840', '07850', '07870', '07900', '07930', '07950', '08', '0806', '080616', '088b', '08â\x80!', '09', '09062016', '0933m', '09600', '0k', '0shares', '0tolerancemovie', '0ð\x9f\x98¥ð\x98¹ï.\x8f', '1', '10', '100', '1000', '10000', '10003', '10007', '1000gifts', '1000th', '1000x', '1000yr', '1000â\x82~', '1001', '1001000s', '10014', '10021', '10025', '10040', '100616', '10064', '100d', '100daysofcode', '100daysofpigpaintings', '100daysoftea', '100faces', '100happydays', '100happydaysâ\x80!', '100happysongs', '100juiceð\x9f\x8d\x8dð\x9f\x8d\x93ð\x9f\x8d\x8d\x87ð\x9f\x8d\x92ð\x9f\x8d\x8d\x91ð\x9f\x8d\x8b', '100k', '100ml', '100pm', '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\x92ð\x9f\x92¥', '1027', '102816', '102pm', '1030', '10353', '104', '1044', '10450', '10480', '10550', '1059am', '105kg', '106', '10650', '10670', '1070', '10700', '1080', '10830', '1096', '10a', '10alltypespos', '10am', '10days', '10hrs', '10k', '10kday', '10kms', '10m', '10meses', '10miler', '10millionmiler', '10minmile', '10minutes', '10minutestogo', '10month', '10moreyrs', '10pm', '10ppm', '10s', '10th', '10x', '10year', '10yr', '11', '110', '11000', '1106', '110616', '1109', '1111', '1117', '11180', '11192016', '112000', '11230', '11240', '11250', '1126', '11270', '1127tn', '1130', '11300', '1130am', '1130pm', '11350', '113days', '114', '11400', '1145', '115', '1155', '115pm215pm4pm', '116', '116000', '1162', '11740', '118', '119', '11911', '11amp12', '11daystogo', '11inchlatcock', '11j', '11june2016', '11juneâ\x9c\x88ï.\x8f', '11millionsubs', '11pm', '11pt', '11th', '11thmonthsary', '12', '120', '1200', '12000', '1200custom', '1200d', '1206', '12062016', '121', '1213', '123', '12313', '1235', '123rd', '124', '125', '12500', '126', '1260', '12700', '12720', '1280', '12840', '12850', '129', '12900', '12952', '1299', '12daystoolong', '12june', '12k', '12mill', '12minm', '12months', '12pm', '12pt', '12th', '12weeksofexerciseâ\x80!', '12x12', '12x12best', '12yearold', '12yrs', '12Â', '13', '1300', '1301', '1306', '130616', '130pm', '1314', '1316', '1320', '13479', '1355', '136', '13615', '13615onedirection', '13671370', '138', '138k', '139', '13cabs', '13days togo', '13hrs', '13percent', '13th', '13thdocumentary', '13yr', '14', '140', '1400', '14000', '1406', '14140', '1418now', '14200', '142017', '1426', '1428', '1430', '14300', '14330', '144000', '14440', '14509', '146', '146540383315', '146548350793', '146549118088', '1465518550221', '14655211716', '146552066096', '1465520416764', '146561572275', '146562489121', '146566206798', '146566278800', '14657200021']
```

In [48]:

```
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 [49]: tweets_countvectorizer.shape
```

```
(31962, 47386)
```

```
In [50]: X = pd.DataFrame(tweets_countvectorizer.toarray())
```

```
In [51]: X
```

	0	1	2	3	4	5	6	7	8	9	...	47376	47377	47378	47379	47380	47381	47382	47383	47384	47385
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
2	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
4	0	0	0	0	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	0	0	0	0
31958	0	0	0	0	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	0	0	0	0
31960	0	0	0	0	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	0	0	0	0

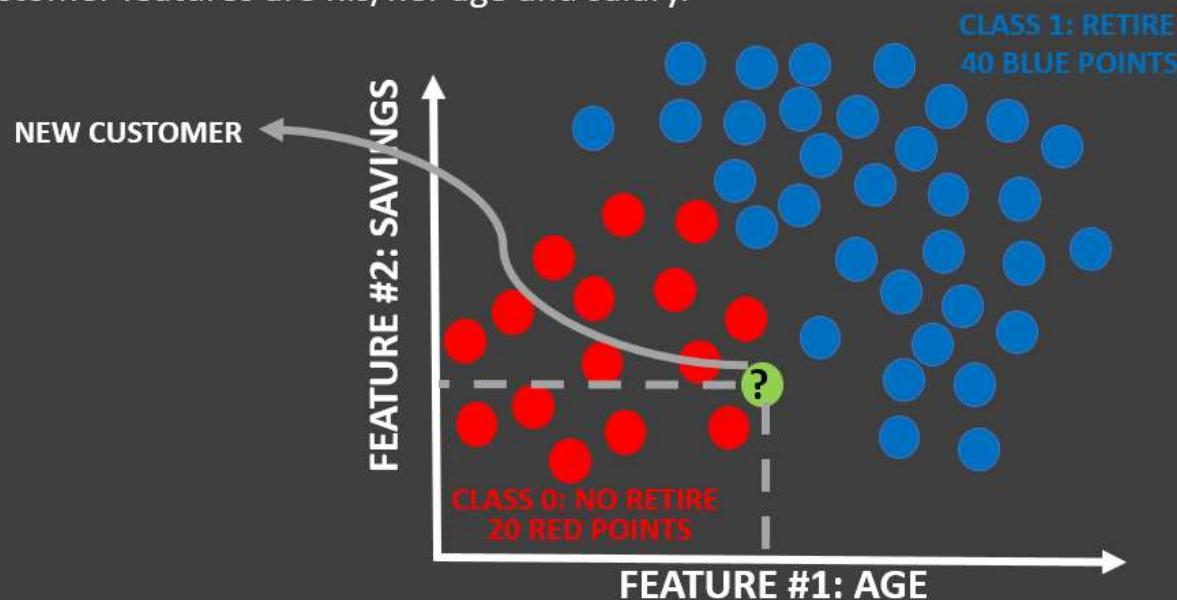
```
31962 rows × 47386 columns
```

```
In [52]: y = tweets_df['label']
```

TASK #9: UNDERSTAND THE THEORY AND

NAÏVE BAYES: INTUITION

- Naïve Bayes is a classification technique based on Bayes' Theorem.
- Let's assume that you are data scientist working major bank in NYC and you want to classify a new client as eligible to retire or not.
- Customer features are his/her age and salary.



NAÏVE BAYES: 1. PRIOR PROBABILITY

- Points can be classified as RED or BLUE and our task is to classify a new point to RED or BLUE.
- Prior Probability: Since we have more BLUE compared to RED, we can assume that our new point is twice as likely to be BLUE than RED.

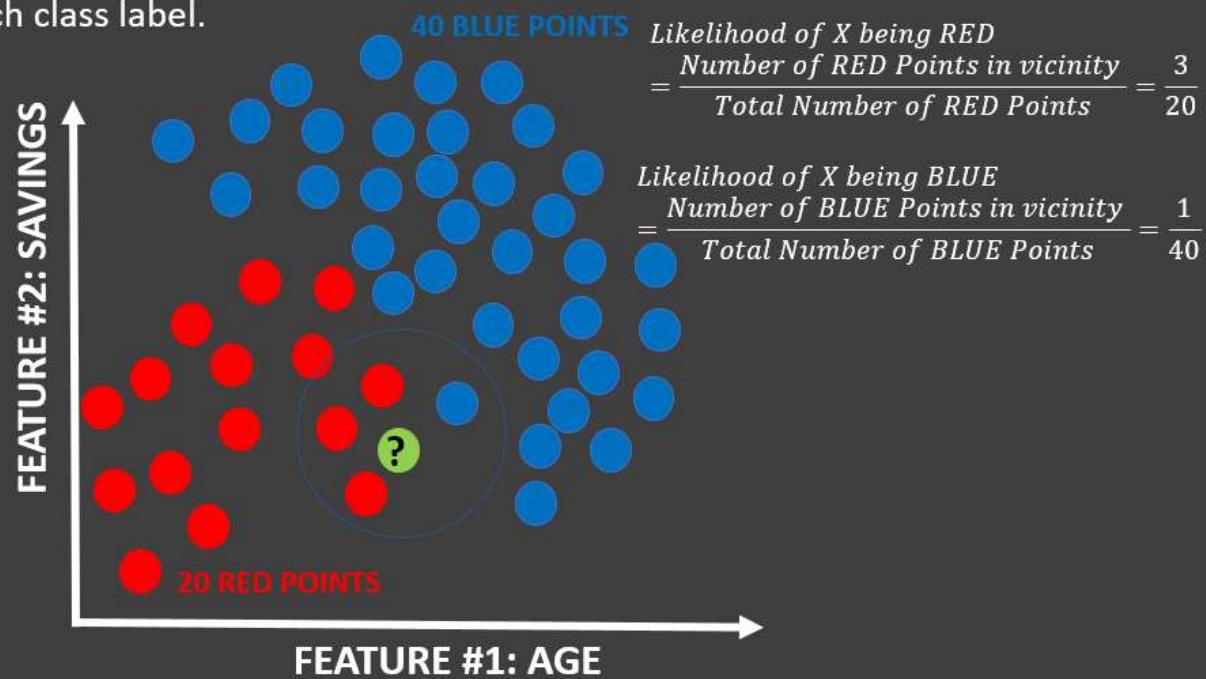


$$\text{Prior Probability for RED} = \frac{\text{Number of RED Points}}{\text{Total Number of Points}} = \frac{20}{60}$$

$$\text{Prior Probability for BLUE} = \frac{\text{Number of BLUE Points}}{\text{Total Number of Points}} = \frac{40}{60}$$

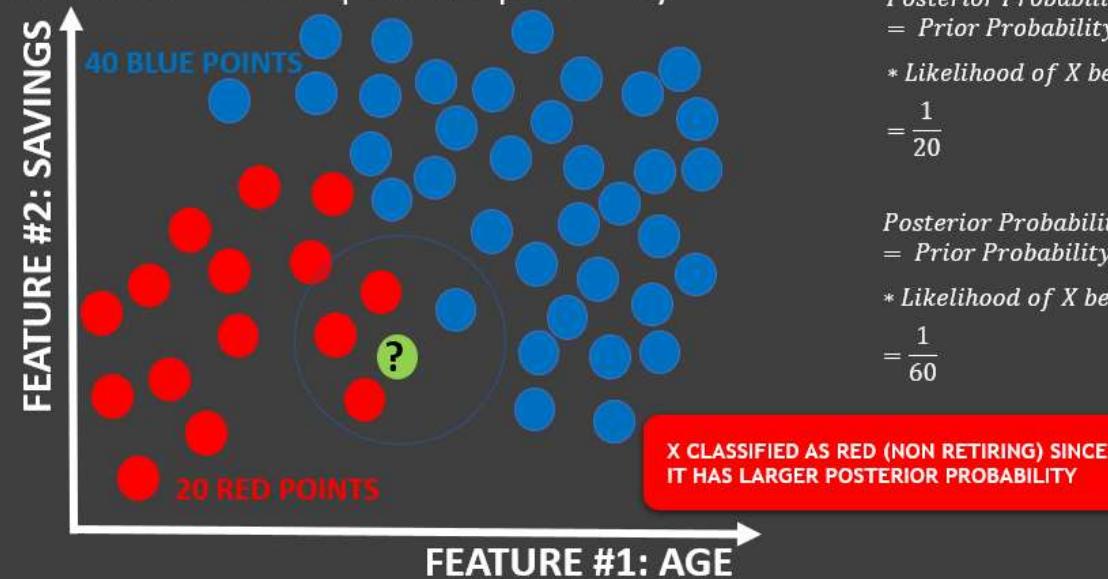
NAÏVE BAYES: 2. LIKELIHOOD

- For the new point, if there are more BLUE points in its vicinity, it is more likely that the new point will be classified as BLUE.
- So we draw a circle around the point, then we calculate the number of points in the circle belonging to each class label.



NAÏVE BAYES: 3. POSTERIOR PROBABILITY

- Let's combine prior probability and likelihood to create a posterior probability.
- Prior probabilities: suggests that X may be classified as BLUE Because there are 2x as much blue points.
- Likelihood: suggests that X is RED because there are more RED points in the vicinity of X.
- Bayes' Rule combines both to form a posterior probability.



$$\begin{aligned} \text{Posterior Probability of } X \text{ being RED} &= \text{Prior Probability of RED} \\ * \text{Likelihood of } X \text{ being RED} &= \frac{20}{60} * \frac{3}{20} \\ &= \frac{1}{20} \end{aligned}$$

$$\begin{aligned} \text{Posterior Probability of } X \text{ being BLUE} &= \text{Prior Probability of BLUE} \\ * \text{Likelihood of } X \text{ being BLUE} &= \frac{40}{60} * \frac{1}{40} \\ &= \frac{1}{60} \end{aligned}$$

NAÏVE BAYES: MATH (DON'T PANIC!)

LIKELIHOOD

PRIOR PROBABILITY
OF RETIRING

MARGINAL LIKELIHOOD

$$P(\text{Retire}|X) = \frac{P(X|\text{Retire}) * P(\text{Retire})}{P(X)}$$

- Naïve Bayes is a classification technique based on Bayes' Theorem.
- X : New Customer's features; age and savings
- $P(\text{Retire}|X)$: probability of customer retiring given his/her features, such as age and savings
- $P(\text{Retire})$: Prior probability of retiring, without any prior knowledge
- $P(X|\text{Retire})$: likelihood
- $P(X)$: Marginal likelihood, the probability of any point added lies into the circle

NAÏVE BAYES: MATH (DON'T PANIC!)

$$P(\text{Retire}|X) = \frac{P(X|\text{Retire}) * P(\text{Retire})}{P(X)}$$

Diagram illustrating the components of the Naïve Bayes formula:

- PRIOR PROBABILITY OF RETIRING**: $P(\text{Retire})$
- LIKELIHOOD**: $P(X|\text{Retire})$
- MARGINAL LIKELIHOOD**: $P(X)$

- $P(\text{Retire}) = \frac{\# \text{ of Retiring}}{\text{Total points}} = 40/60$
- $P(X|\text{Retire}) = \frac{\# \text{ of similar observations for retiring}}{\text{Total # retiring}} = 1/40$
- $P(X) = \frac{\# \text{ of Similar observations}}{\text{Total # Points}} = 4/60$
- $P(\text{Retire}|X) = \frac{\frac{40}{60} * \frac{1}{40}}{\frac{4}{60}} = \frac{1/60}{4/60} = 0.25$

PRACTICE OPPORTUNITY #6:

PRACTICE OPPORTUNITY

Calculate the probability of non-retiring (red class)

$$P(\text{No Retire}|X) = ?$$

PRACTICE OPPORTUNITY SOLUTION

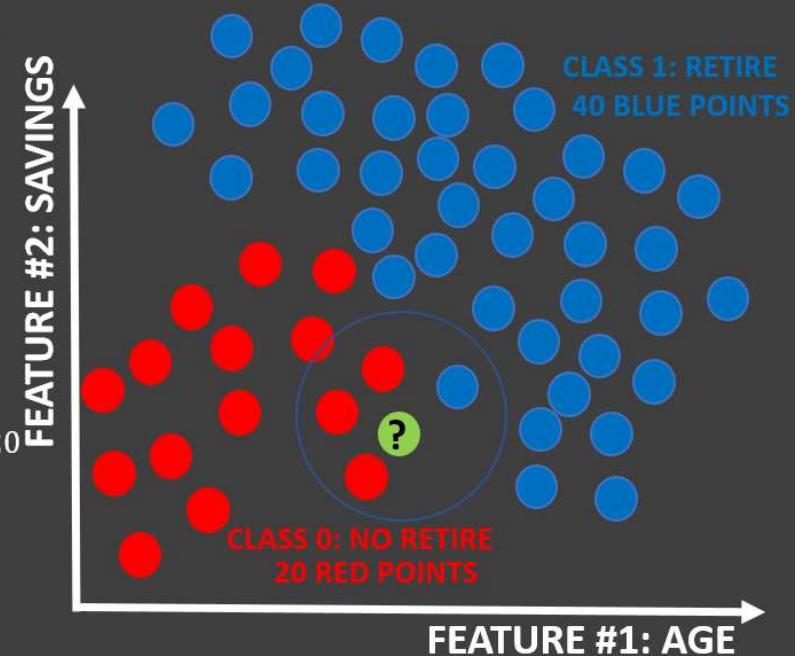
$$P(\text{No Retire}|X) = \frac{P(X|\text{No Retire}) * P(\text{No Retire})}{P(X)}$$

LIKELIHOOD ← PRIOR PROBABILITY OF NO RETIRING →

↓ ↓

MARGINAL LIKELIHOOD

- $P(\text{No Retire}) = \frac{\# \text{ of No Retiring}}{\text{Total points}} = 20/60$
- $P(X|\text{No Retire}) = \frac{\# \text{ of similar observations for No retiring}}{\text{Total # no retiring}} = 3/20$
- $P(X) = \frac{\# \text{ of Similar observations}}{\text{Total # Points}} = 4/60$
- $P(\text{No Retire}|X) = \frac{\frac{20}{60} * \frac{3}{20}}{\frac{4}{60}} = \frac{3/60}{4/60} = 0.75$



NOTE: $P(\text{Non Retire}|X) = 1 - 0.75 = 0.25$

TASK #10: TRAIN AND EVALUATE A NAIVE BAYES CLASSIFIER MODEL

```
In [53]: X.shape
```

(31962, 47386)

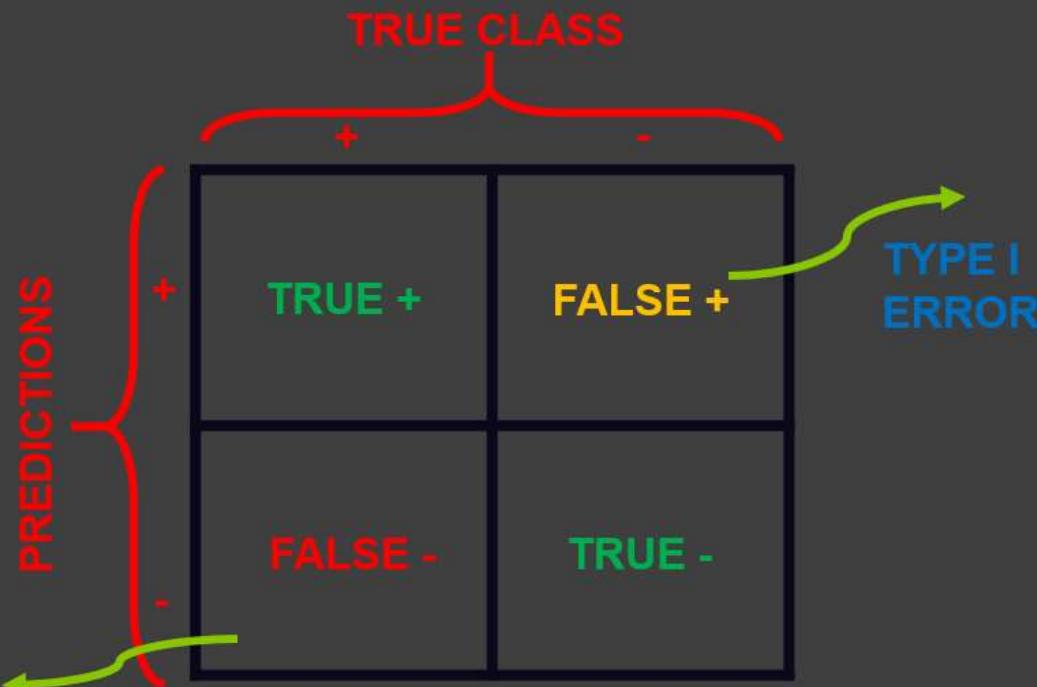
```
In [54]: y.shape
```

```
(31962,)
```

```
In [55]: 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)
```

```
In [56]: from sklearn.naive_bayes import MultinomialNB  
  
NB_classifier = MultinomialNB()  
NB_classifier.fit(X_train, y_train)  
  
MultinomialNB(alpha=1.0, class_prior=None, fit_prior=True)
```

CONFUSION MATRIX

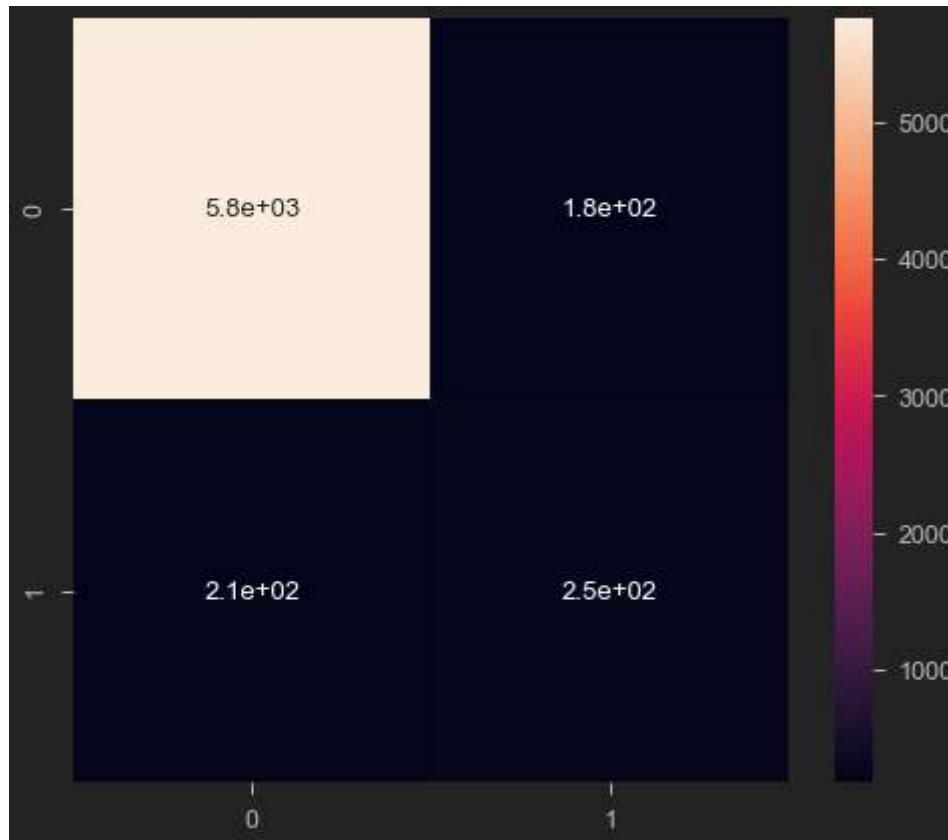


```
In [57]: from sklearn.metrics import classification_report, confusion_matrix
```

```
In [58]: # 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)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x13f95382bc8>
```



```
In [59]: print(classification_report(y_test, y_predict_test))
```

	precision	recall	f1-score	support
0	0.97	0.97	0.97	5933
1	0.59	0.55	0.57	460
accuracy			0.94	6393
macro avg	0.78	0.76	0.77	6393
weighted avg	0.94	0.94	0.94	6393

FINAL PROJECT

FINAL PROJECT CARD

GOAL:

- *Build, train, test and deploy an Artificial intelligence (AI) model to predict sentiment from real Amazon Echo customer reviews.*

TOOL:

- *Anaconda, Python, Scikit-Learn, Matplotlib, Seaborn*

PRACTICAL REAL-WORLD APPLICATION:

- *AI/ML-based sentiment analysis is crucial for companies to automatically predict whether their customers are happy or not. The process could be done automatically without having humans manually review thousands of customer reviews.*

DATA:

- **INPUTS:**
 - *Amazon Reviews*
- **OUTPUT:**
 - *Sentiment (0 or 1)*



Source: [https://commons.wikimedia.org/wiki/File:Amazon_Echo_Dot_\(black\)_on_a_wood_surface.jpg](https://commons.wikimedia.org/wiki/File:Amazon_Echo_Dot_(black)_on_a_wood_surface.jpg)

Data Source: <https://www.kaggle.com/sid321axn/amazon-alex-audio-reviews>

FINAL PROJECT TASKS

1. Instead of using Twitter dataset, use the “Amazon reviews” dataset provided in the course package.
2. Perform Exploratory data analysis.
3. Plot the wordcloud and list the top 3 popular words.
4. Train multiple models using Scikit Learn.
5. Assess trained models performance.

28

FINAL PROJECT SOLUTION TASK #1: IMPORT DATA AND PERFORM EXPLORATORY DATA ANALYSIS

In [60]:

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

from jupyterthemes import jtplot
jtplot.style(theme='monokai', context='notebook', ticks=True, grid=False)
# setting the style of the notebook to be monokai theme
# this line of code is important to ensure that we are able to see the x and y axes clearly
# If you don't run this code line, you will notice that the xlabel and ylabel on any plot is black on black
```

```
In [61]: # Load the data
```

```
reviews_df = pd.read_csv('amazon_reviews.csv')  
reviews_df
```

	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 [62]: # View the DataFrame Information  
reviews_df.info()
```

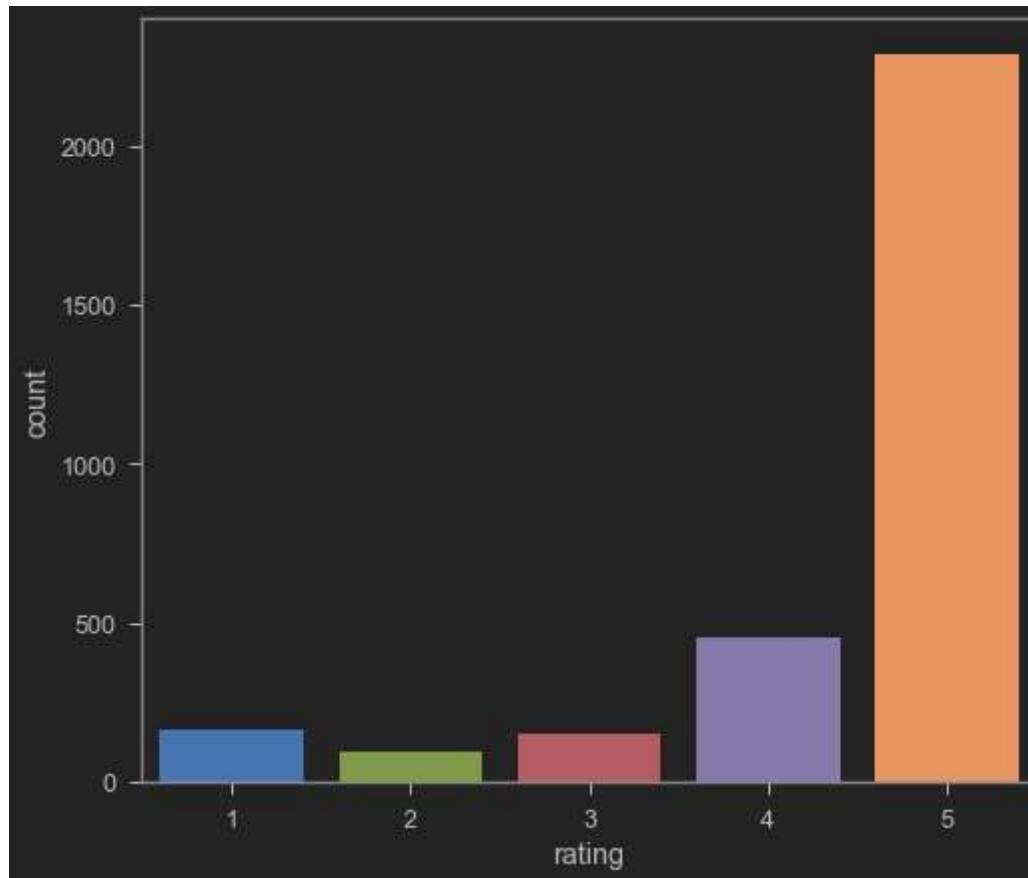
```
<class 'pandas.core.frame.DataFrame'>  
RangeIndex: 3150 entries, 0 to 3149  
Data columns (total 5 columns):  
 #   Column           Non-Null Count  Dtype     
---  --     
 0   rating          3150 non-null    int64    
 1   date            3150 non-null    object    
 2   variation       3150 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 [63]: # View DataFrame Statistical Summary  
reviews_df.describe()
```

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

```
<matplotlib.axes._subplots.AxesSubplot at 0x13f94f5dc08>
```



```
In [65]: # Let's get the length of the verified_reviews column  
reviews_df['length'] = reviews_df['verified_reviews'].apply(len)
```

```
In [66]: reviews_df
```

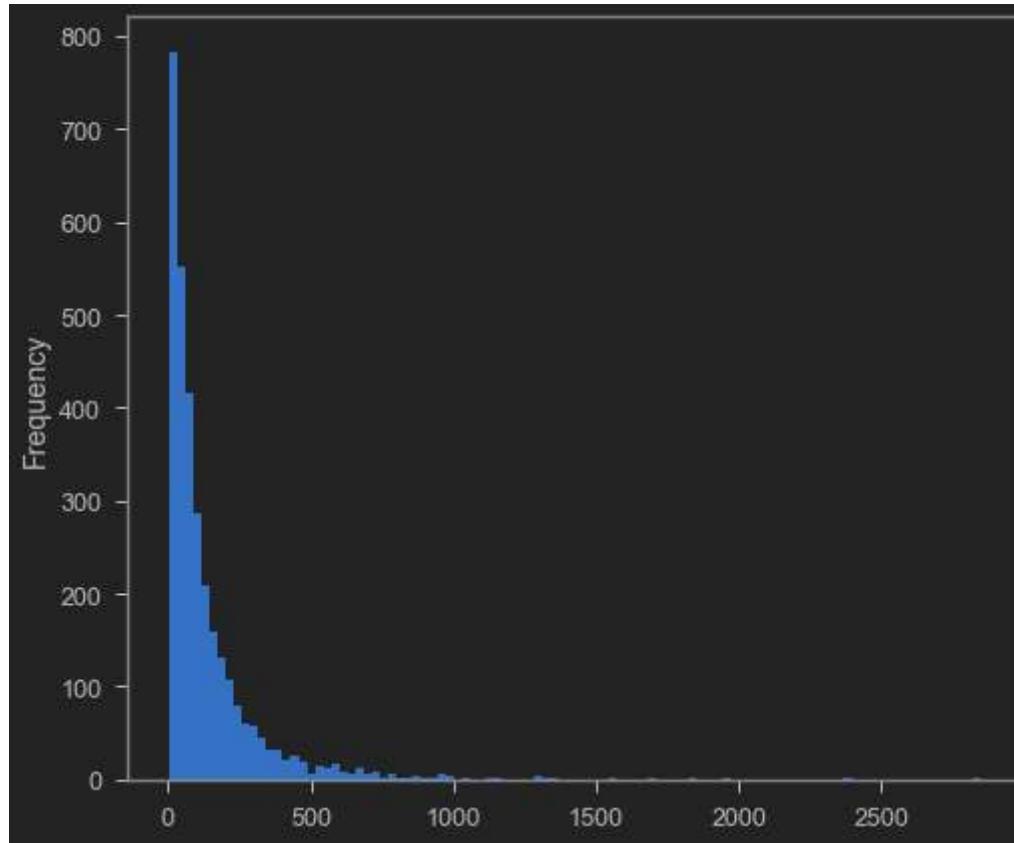
	rating	date	variation		verified_reviews	feedback	length
0	5	31-Jul-18	Charcoal Fabric	Love my Echo!	1	13	
1	5	31-Jul-18	Charcoal Fabric	Loved it!	1	9	
2	4	31-Jul-18	Walnut Finish	Sometimes while playing a game, you can answer...	1	195	
3	5	31-Jul-18	Charcoal Fabric	I have had a lot of fun with this thing. 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 [67]: # Plot the histogram for the Length
```

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

```
<matplotlib.axes._subplots.AxesSubplot at 0x13f9536a9c8>
```



```
In [68]: # Apply the describe method to get statistical summary  
reviews_df.describe()
```

	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

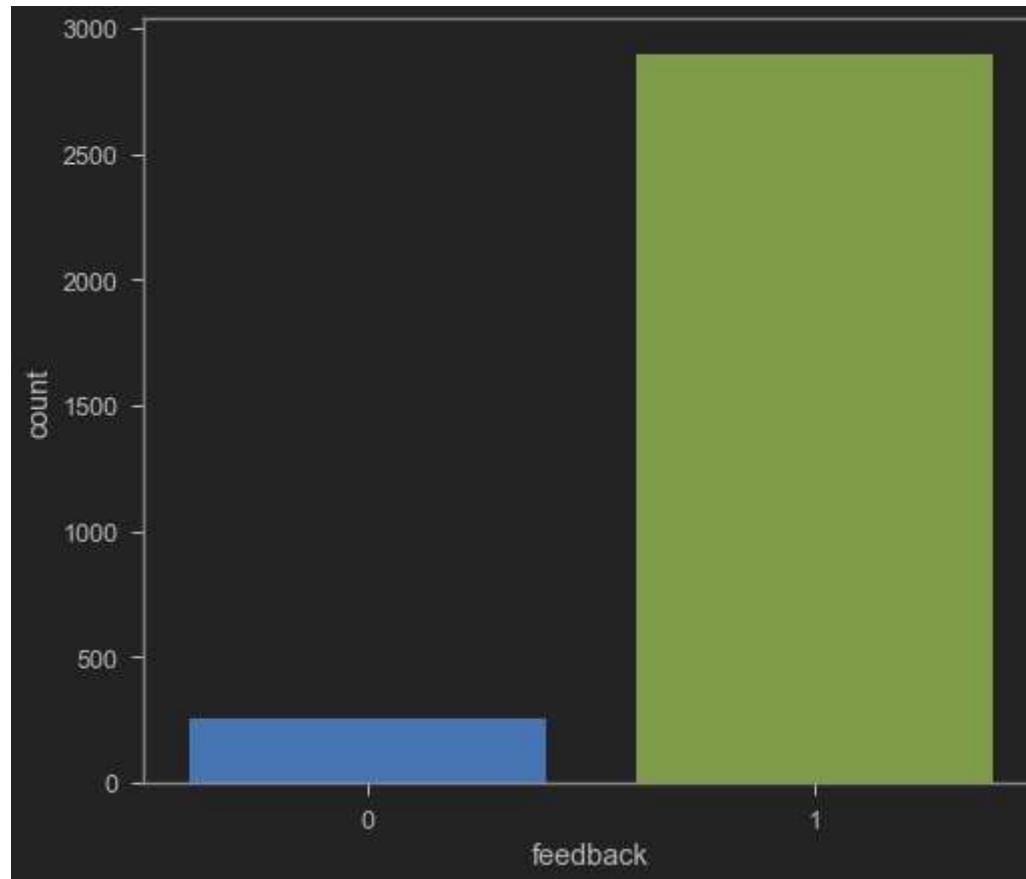
```
In [69]: # Plot the countplot for feedback
```

```
# Positive ~2800
```

```
# Negative ~250
```

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

```
<matplotlib.axes._subplots.AxesSubplot at 0x13f9576f848>
```



FINAL PROJECT SOLUTION TASK #2: PLOT WORDCLOUD

```
In [70]: # Obtain only the positive reviews
```

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

	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 [71]: # Obtain the negative reviews only
```

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

	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 [72]: # Convert to List format
```

```
sentences = positive['verified_reviews'].tolist()  
len(sentences)
```

2893

```
In [73]: # Join all reviews into one large string
```

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

In [74]:

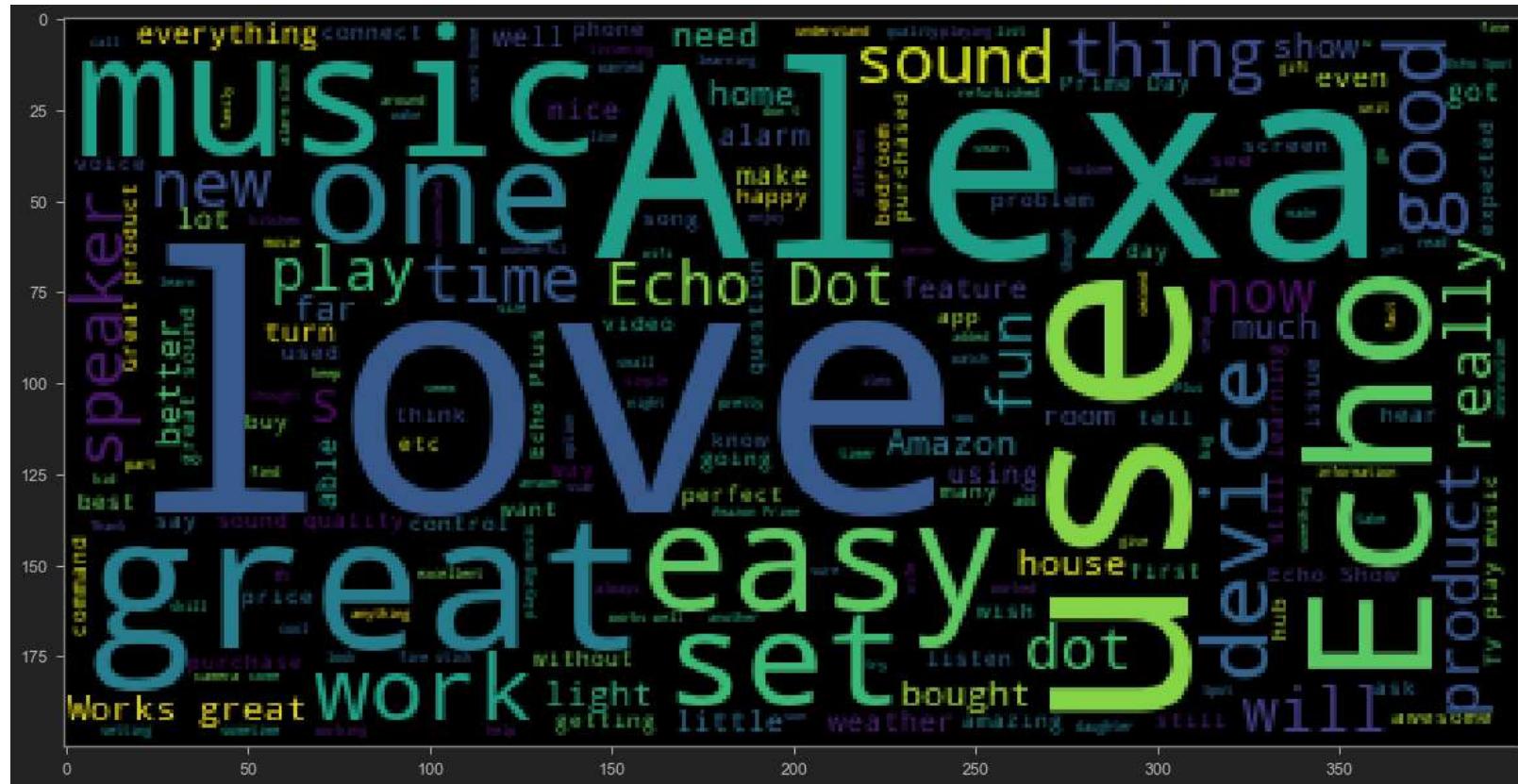
```
sentences_as_one_string
```

'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 playing music as well. Music I received the echo as a gift. I needed another Bluetooth or something to play music easily accessible, and found this smart speaker. Can't wait to see what else it can do. Without having a cellphone, I cannot use many of her features. I have an iPad but do not see that 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 does 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 specifilay 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, weather, information! It's great! I sent it to my 85 year old Dad, and he talks to it constantly. I love it! Learning knew things with it eveyday! Still figuring out how everything works but so far it's been easy to use and understand. She does make me laugh at times I purchased this for my mother who is having knee problems now, to give her something to do while trying to over come not getting around so fast like she did. She enjoys all the little and big things it can do...Alexa play this song, What time is it and where, and how to cook this and that! Love, Love, Love!! Just what I expected.... I love it, wife hates it. Really happy with this purchase. Great speaker and easy to set up. We have only been using Alexa for a couple of days and are having a lot of fun with our new toy. It like having a new household member! We are trying to learn all the different feautes and benefits that come with it. We love the size of the 2nd generation echo. Still needs a little improvement on sound I liked the original Echo. This is the same but shorter and with greater fabric/color choices. I miss the volume ring on top, now it's just the plus/minus buttons. Not a big deal but the ring was comforting. :) Other than that, well I do like the use of a standard USB charger /port instead of the previous round pin. Other than that, I guess it sounds the same, seems to work the same, still answers to Alexa/Echo/Computer. So what's not to like? :) Love the Echo and how good the music sounds playing off it. Alexa understands most commands but it is difficult at times for her to find specific playlists or songs on Spotify. She is good with Amazon Music but is lacking in other major programs. We love Alexa! We use her to play music, play radio through iTunes, play podcasts through Anypod, and set reminders. We listen to our flash briefing of news and weather every morning. We rely on our custom lists. We like being able to voice control the volume. We're sure we'll continue to find new uses for it sometimes.'

```
In [75]: from wordcloud import WordCloud
```

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

```
<matplotlib.image.AxesImage at 0x13f953a3408>
```



```
In [ ]: 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))
```

FINAL PROJECT SOLUTION TASK #3: PERFORM DATA CLEANING

```
In [ ]: # 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.lower() not in :
        return Test_punc_removed_join_clean
```

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

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

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

```
In [ ]: 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 [ ]: print(vectorizer.get_feature_names())
```

```
In [ ]: print(reviews_countvectorizer.toarray())
```

```
In [ ]: reviews_countvectorizer.shape
```

```
In [ ]: reviews = pd.DataFrame(reviews_countvectorizer.toarray())
```

```
In [ ]: X = reviews
```

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

FINAL PROJECT SOLUTION TASK #4: TRAIN AND TEST AI/ML MODELS

```
In [ ]: 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)
```

```
In [ ]: from sklearn.naive_bayes import MultinomialNB  
  
NB_classifier = MultinomialNB()  
NB_classifier.fit(X_train, y_train)
```

```
In [ ]: from sklearn.metrics import classification_report, confusion_matrix
```

```
In [ ]: # 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)
```

```
In [ ]: print(classification_report(y_test, y_predict_test))
```

```
In [ ]: 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)

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

```
In [ ]: 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)

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

**EXCELLENT JOB! YOU SHOULD BE PROUD OF
YOUR NEWLY ACQUIRED SKILLS**

In []: