Completed Coursera Guided Project by Suhaimi William Chan

Instructor: Ryan Ahmed, Ph.D.

Project Structure

The hands on project on ***Twitter Sentiment Analysis*** is divided into following tasks:

Task #1: Understand the Problem Statement and business case

Task #2: Import libraries and datasets

Task #3: Perform Exploratory Data Analysis

Task #4: Plot the word cloud

Task #5: Perform data cleaning - removing punctuation

Task #6: Perform data cleaning - remove stop words

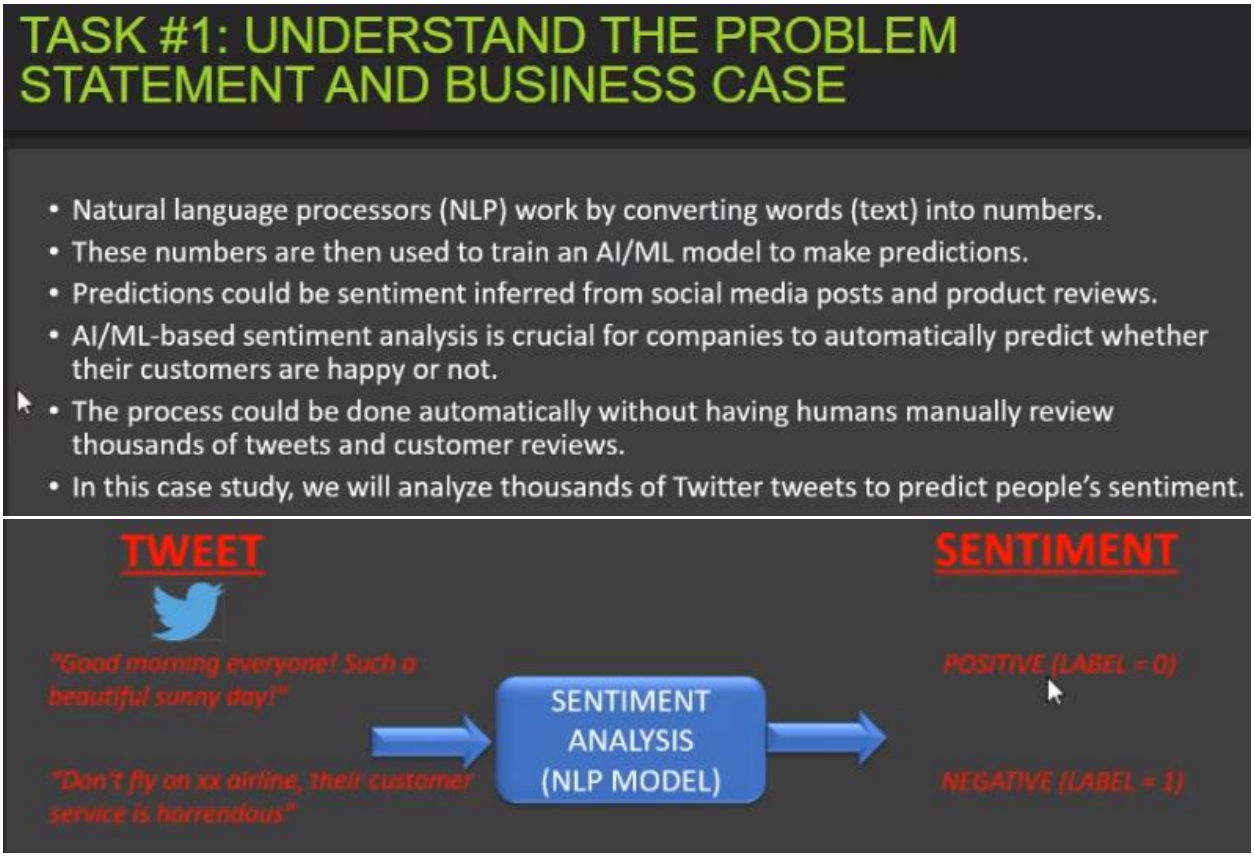
Task #7: Perform Count Vectorization (Tokenization)

Task #8: Create a pipeline to remove stop-words, punctuation, and perform tokenization

Task #9: Understand the theory and intuition behind Naive Bayes classifiers

Task #10: Train a Naive Bayes Classifier

Task #11: Assess trained model performance


TASK #1: UNDERSTAND THE PROBLEM STATEMENT AND BUSINESS CASE
• 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
SENTIMENT
"Good morning everyone! Such a beautiful sunny day!"
"Don't fly on xx airline, their customer service is horrendous"
SENTIMENT ANALYSIS (NLP MODEL)
POSITIVE (LABEL = 0)
NEGATIVE (LABEL = 1)

data source: https://www.kaggle.com/arkhoshghalb/twitter-sentiment-analysis-hatred-speech

TASK #2: IMPORT LIBRARIES AND DATASETS

In [2]:
```
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 [3]:
```
# Load the data
tweets_df = pd.read_csv('twitter.csv')
```

In [4]:
```
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

MINI CHALLENGE #1:

- Drop the 'id' column from the DataFrame.
- Ensure that the column has been succesfully dropped.

In [8]:
```
tweets_df = tweets_df.drop(['id'], axis = 1)
```

In [9]:
```
tweets_df
```

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

31962 rows × 2 columns

In [5]:
```
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 [6]:
```
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 [7]:
```
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˜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
```

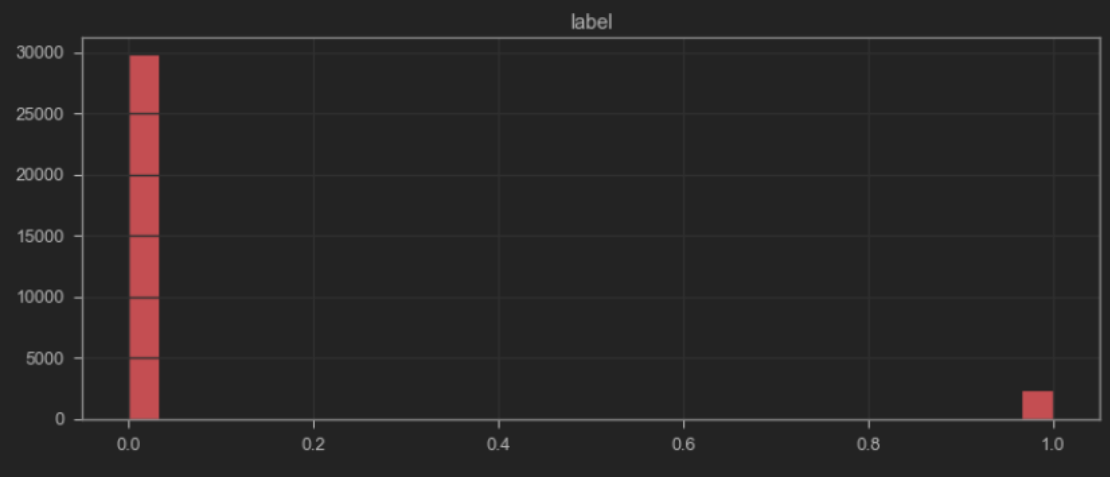
TASK #3: EXPLORE DATASET

In [10]:
```
sns.heatmap(tweets_df.isnull(), yticklabels = False, cbar = False, cmap="Blues")
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x1a27412e5c8>
```

In [11]:
```
tweets_df.hist(bins = 30, figsize = (13,5), color = 'r')
```

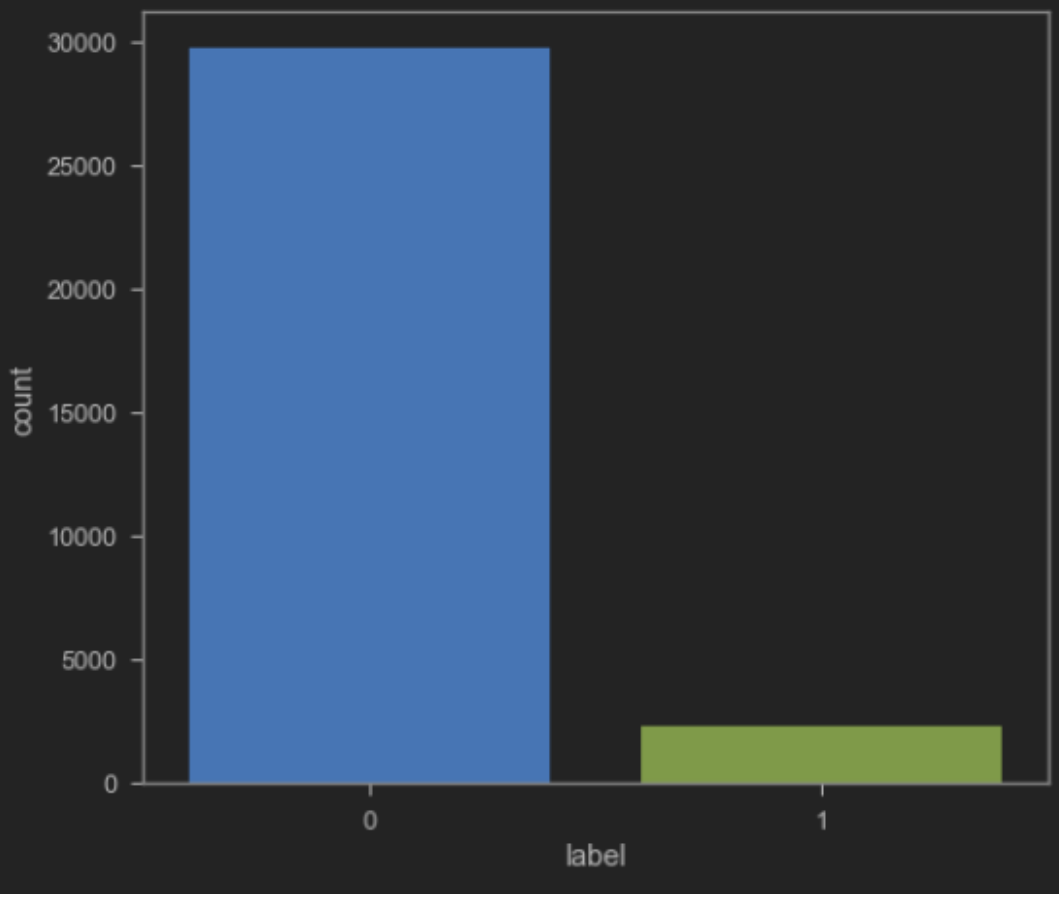
```
array([[<matplotlib.axes._subplots.AxesSubplot object at 0x000001A2740FBAC8>]],
      dtype=object)
```



- Plot similar figure using seaborn countplot

```
In [12]: sns.countplot(tweets_df['label'], label = 'Count')
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x1a274ac2d08>
```



In [13]:
```
# Let's get the length of the messages
tweets_df['length'] = tweets_df['tweet'].apply(len)
```

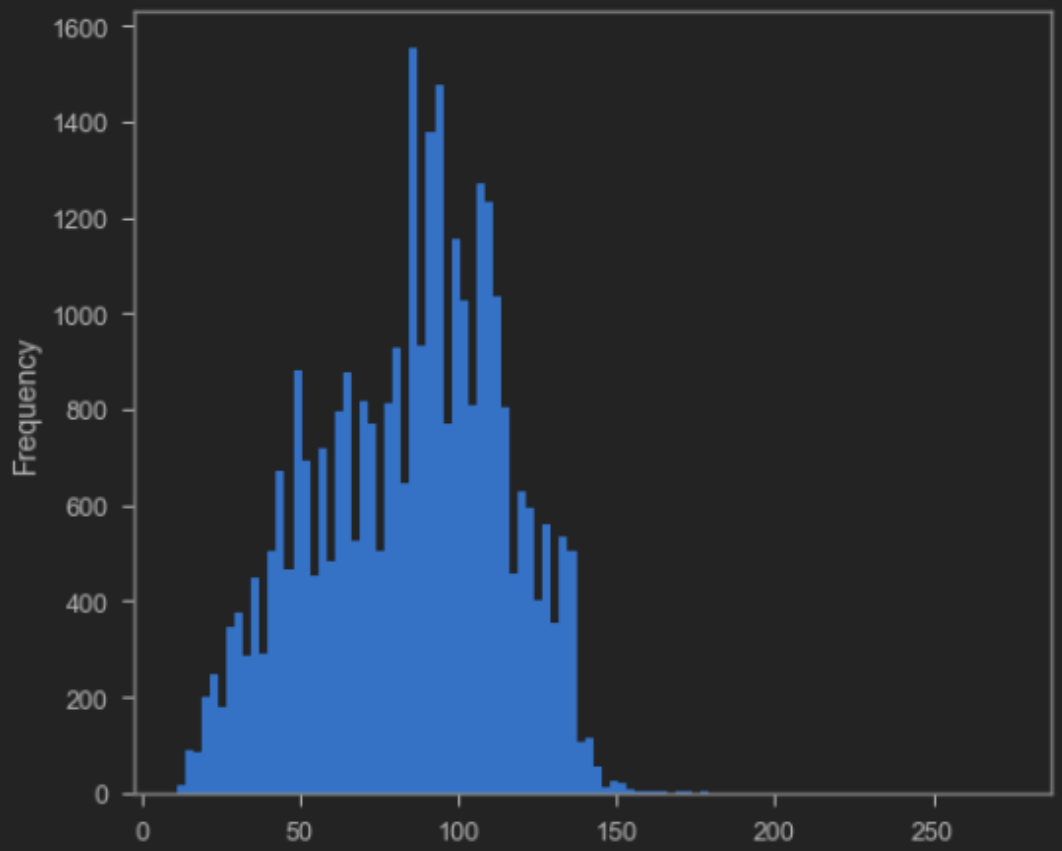
In [14]:
```
tweets_df
```

	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˜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 [15]:
```
tweets_df['length'].plot(bins=100, kind='hist')
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x1a275100b08>
```



In [16]:
```
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 [17]:
```
# Let's see the shortest message
tweets_df[tweets_df['length'] == 11]['tweet'].iloc[0]
```

```
'i love you '
```

MINI CHALLENGE #3

- View the message with the average length

In [18]:
```
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 [19]:
```
positive = tweets_df[tweets_df['label']==0]
```

In [20]:
```
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˜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 [21]:
```
negative = tweets_df[tweets_df['label']==1]
```

In [22]:
```
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 #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

TASK #4: PLOT THE WORDCLOUD

In [23]:
```
sentences = tweets_df['tweet'].tolist()
```

In [24]:
```
sentences
```

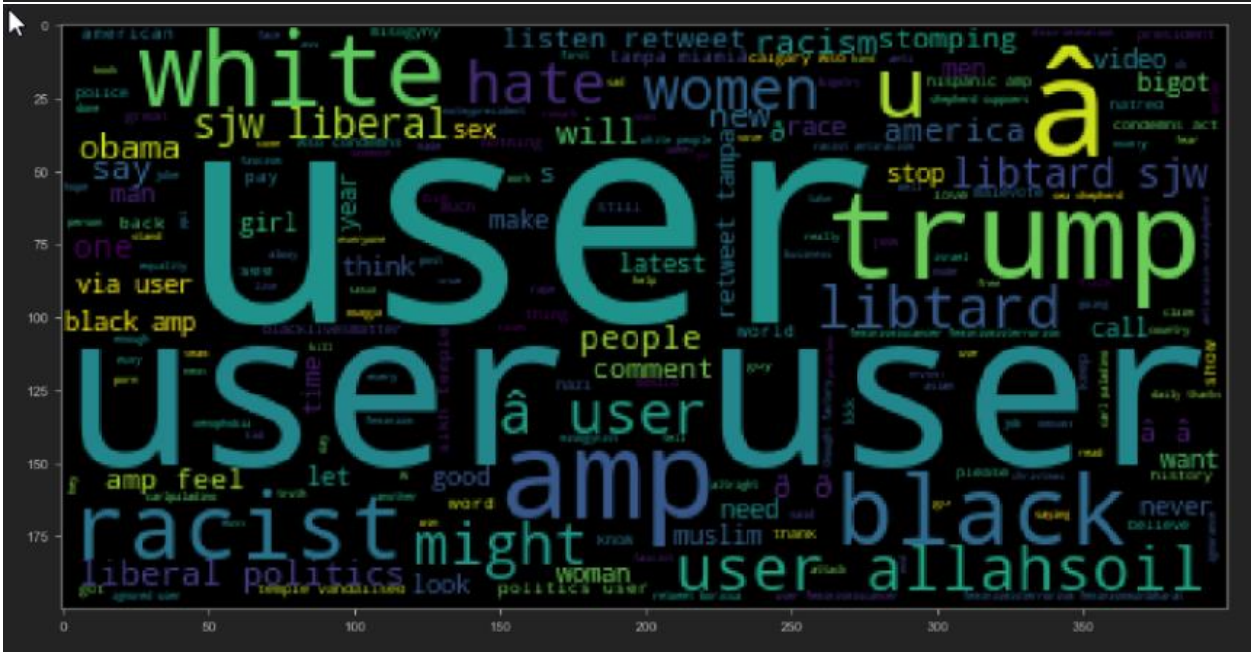
```
[' @user when a father is dysfunctional and is so selfish he drags his kids into his dysfunction.   #run',
 "@user @user thanks for #lyft credit i can't use cause they don't offer wheelchair vans in pdx.    #disapointed #getthanked",
 '  bihday your majesty',
 '#model   i love u take with u all the 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¦  ',
 ' factsguide: society now    #motivation',
 '[2/2] huge fan fare and big talking before they leave. chaos and pay disputes when 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 land!!! #allin #cavs #champions #cleveland #clevelandcavaliers  â\x80¦ ',
 " @user @user welcome here !  i'm   it's so #gr8 ! ",
 ' â\x86\x9d #ireland consumer price index (mom) climbed from previous 0.2% to 0.5% in may   #blog #silver #gold #forex',
 'we are so selfish. #orlando #standwithorlando #pulseshooting #orlandoshooting #biggerproblems #selfish #heabreaking   #values #love #',
 'i get to see my daddy today!!   #80days #gettingfed',
 '@user #cnn calls #michigan middle school \'build the wall\' chant \'\' #tcot  ',
 'no comment!  in #australia   #opkillingbay #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. ',
```

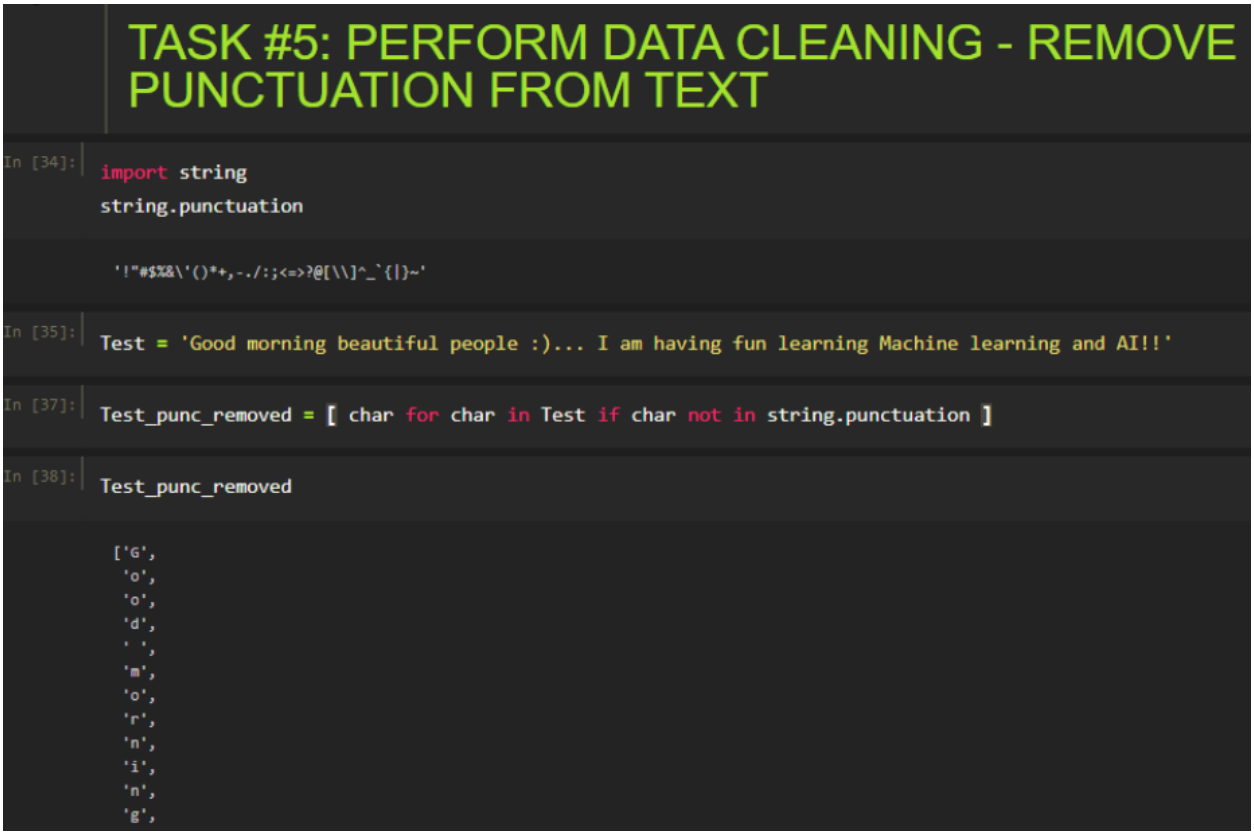
```
In [25]: len(sentences)

         31962

In [26]: sentences_as_one_string = " ".join(sentences)

In [27]: !pip install WordCloud
         from wordcloud import WordCloud

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

```
MINI CHALLENGE #4:
- Plot the wordcloud of the "negative" dataframe.
- What do you notice? Does the data make sense?
```

```
In [33]: negative_list = negative['tweet'].tolist()
         negative_sentences_as_one_string = " ".join(negative_list)
         plt.figure(figsize=(20,20))
         plt.imshow(WordCloud().generate(negative_sentences_as_one_string))
```

```
<matplotlib.image.AxesImage at 0x1a275971ec8>
```




TASK #5: PERFORM DATA CLEANING - REMOVE PUNCTUATION FROM TEXT
In [34]:
import string
string.punctuation
'!"#$%&\'()*+,-./:;<=>?@[\\]^_`{|}~'
In [35]:
Test = 'Good morning beautiful people :)... I am having fun learning Machine learning and AI!!'
In [37]:
Test_punc_removed = [char for char in Test if char not in string.punctuation]
In [38]:
Test_punc_removed
['G',
'o',
'o',
'd',
' ',
'm',
'o',
'r',
'n',
'i',
'n',
'g',

In [45]:
```
# 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'
```

```
MINI CHALLENGE #5:
- Remove punctuations using a different method
```

In [46]:
```
Test_punc_removed = []
for char in Test:
    if char not in string.punctuation:
        Test_punc_removed.append(char)

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 [41]:
```
import nltk # Natural Language tool kit

nltk.download('stopwords')
```

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

True
```

In [42]:
```
# You have to download stopwords Package to execute this command
from nltk.corpus import stopwords
stopwords.words('english')
```

```
['i',
 'me',
 'my',
 'myself',
 'we',
 'our',
 'ours',
```

In [47]:
```
_clean = [ word for word in Test_punc_removed_join.split() if word.lower() not in stopwords.words('english')]
```

In [48]:
```
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',
```

In [49]:
```
# Only important (no so common) words are left
Test_punc_removed_join_clean
```

```
['Good',
 'morning',
 'beautiful',
 'people',
 'fun',
 'learning',
 'Machine',
 'learning',
 'AI']
```

MINI CHALLENGE #6:

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

In [50]:
```
mini_challenge = 'Here is a mini challenge, that will teach you how to remove stopwords and punctuations!'
```

In [51]:
```
challenge = [ char for char in mini_challenge if char not in string.punctuation ]
challenge = ''.join(challenge)
challenge = [ word for word in challenge.split() if word.lower() not in stopwords.words('english')]
```

In [52]:
```
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 [53]:
```
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 [54]:
```
print(vectorizer.get_feature_names())
```

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

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

MINI CHALLENGE #7:

- Without doing any code, perform count vectorization for the following list:
  - mini_challenge = ['Hello World','Hello Hello World','Hello World world world']
- Confirm your answer with code

In [56]:
```
mini_challenge = ['Hello World','Hello Hello World','Hello 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 [57]:
```
# 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 s
    return Test_punc_removed_join_clean
```

In [58]:
```
# Let's test the newly added function
tweets_df_clean = tweets_df['tweet'].apply(message_cleaning)
```

In [59]:
```
print(tweets_df_clean[5]) # show the cleaned up version
```

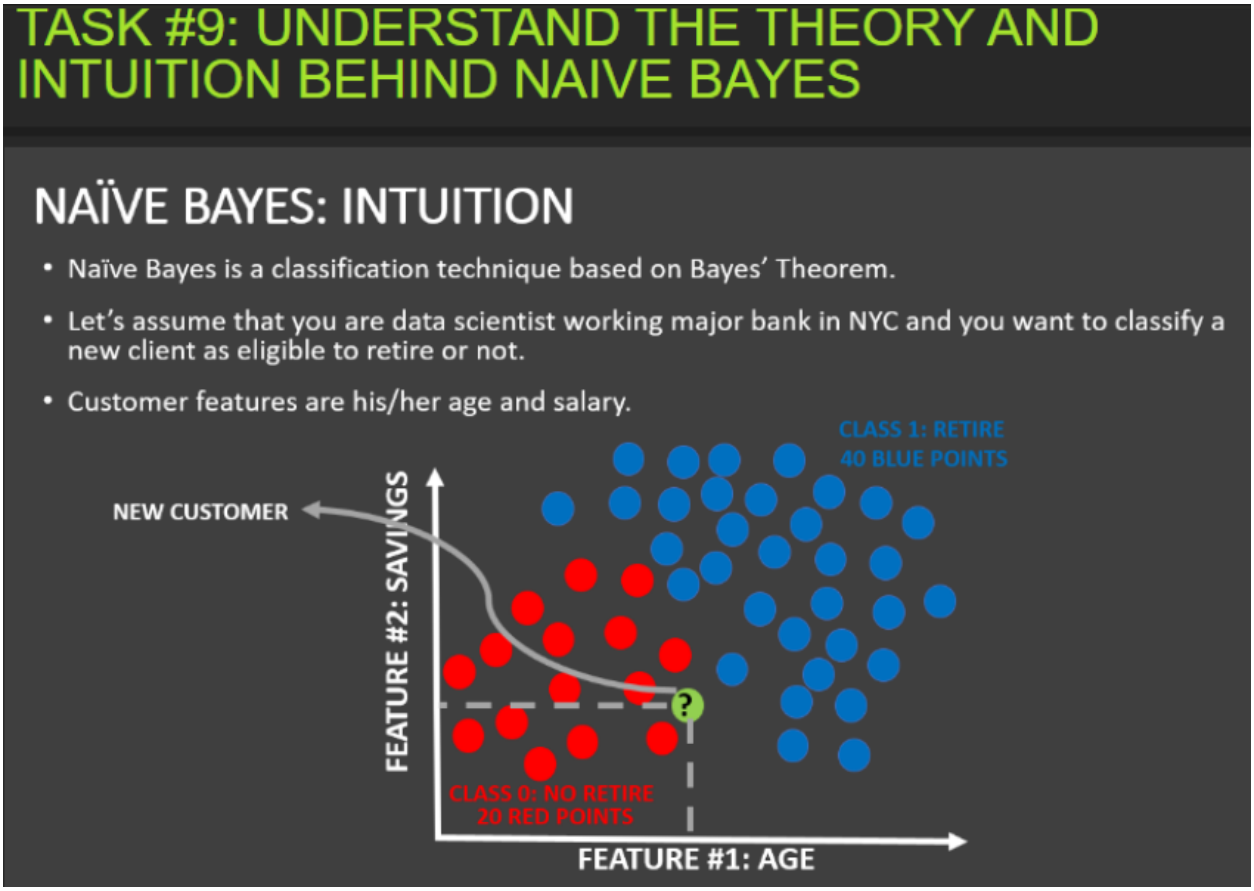
```
['22', 'huge', 'fan', 'fare', 'big', 'talking', 'leave', 'chaos', 'pay', 'disputes', 'get', 'allshowandnogo']
```

In [60]:
```
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 [68]:
```
from sklearn.feature_extraction.text import CountVectorizer
# Define the cleaning pipeline we defined earlier
vectorizer = CountVectorizer(analyzer = message_cleaning)
tweets_countvectorizer = CountVectorizer(analyzer = message_cleaning,
                                         dtype = 'uint8').fit_transform(tweets_df['tweet']).toarray()
```

```
In [70]: tweets_countvectorizer.shape

         (31962, 47386)

In [ ]:

In [71]: X = tweets_countvectorizer

In [ ]:

In [72]: y = tweets_df['label']
```



NAÏVE BAYES: 1. PRIOR PROBABILITY

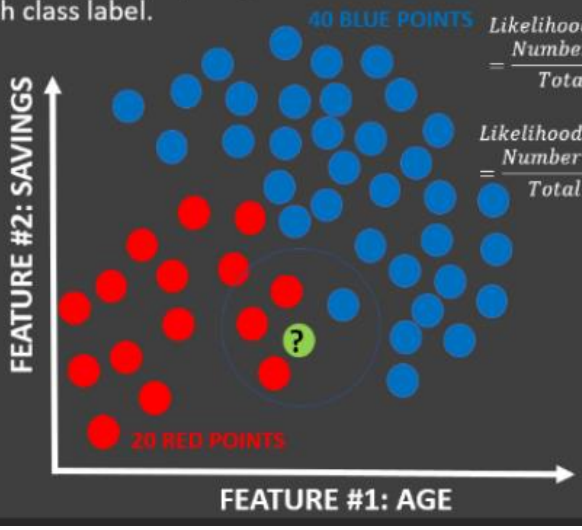
- Points can be classified as RED or BLUE and our task is to classify a new point to RED or BLUE.
- Prior Probability: Since we have more BLUE compared to RED, we can assume that our new point is twice as likely to be BLUE than RED.

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

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

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.

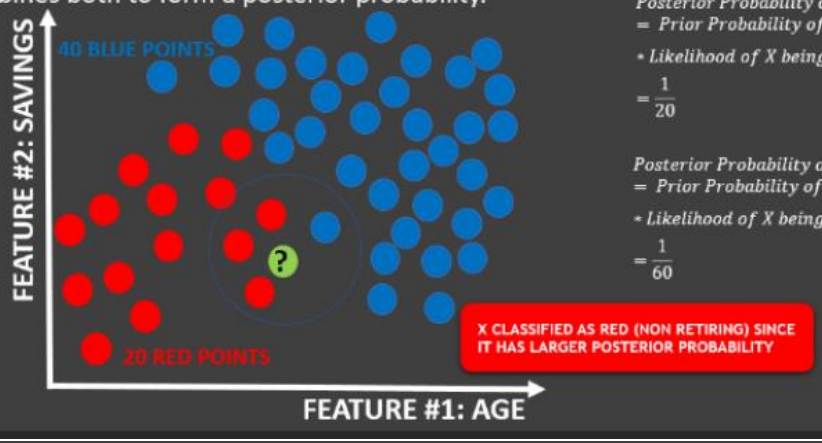


$$\text{Likelihood of X being RED} = \frac{\text{Number of RED Points in vicinity}}{\text{Total Number of RED Points}} = \frac{3}{20}$$

$$\text{Likelihood of X being BLUE} = \frac{\text{Number of BLUE Points in vicinity}}{\text{Total Number of BLUE Points}} = \frac{1}{40}$$

NAÏVE BAYES: 3. POSTERIOR PROBABILITY

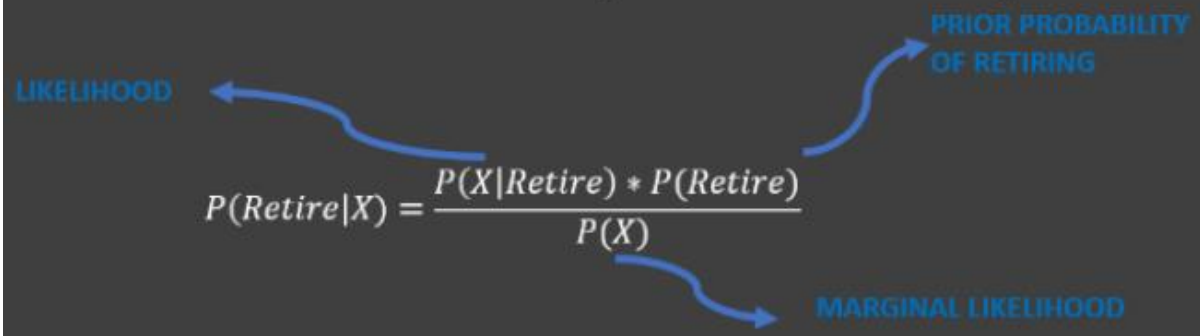
- Let's combine prior probability and likelihood to create a posterior probability.
- Prior probabilities: suggests that X may be classified as BLUE Because there are 2x as much blue points.
- Likelihood: suggests that X is RED because there are more RED points in the vicinity of X.
- Bayes' Rule combines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



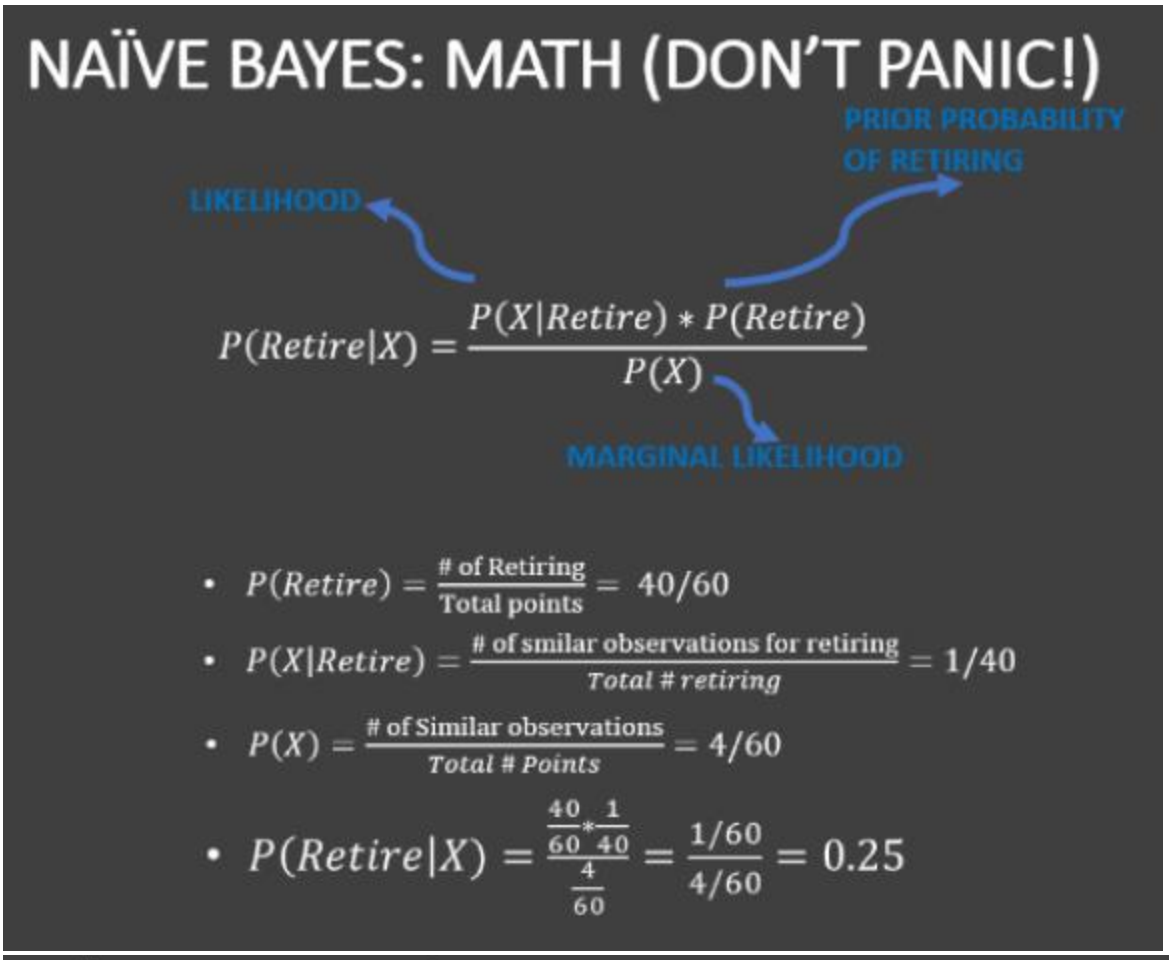
$$\textit{Posterior Probability of X being RED} = \textit{Prior Probability of RED} * \textit{Likelihood of X being RED} = \frac{20}{60} * \frac{3}{20} = \frac{1}{20}$$

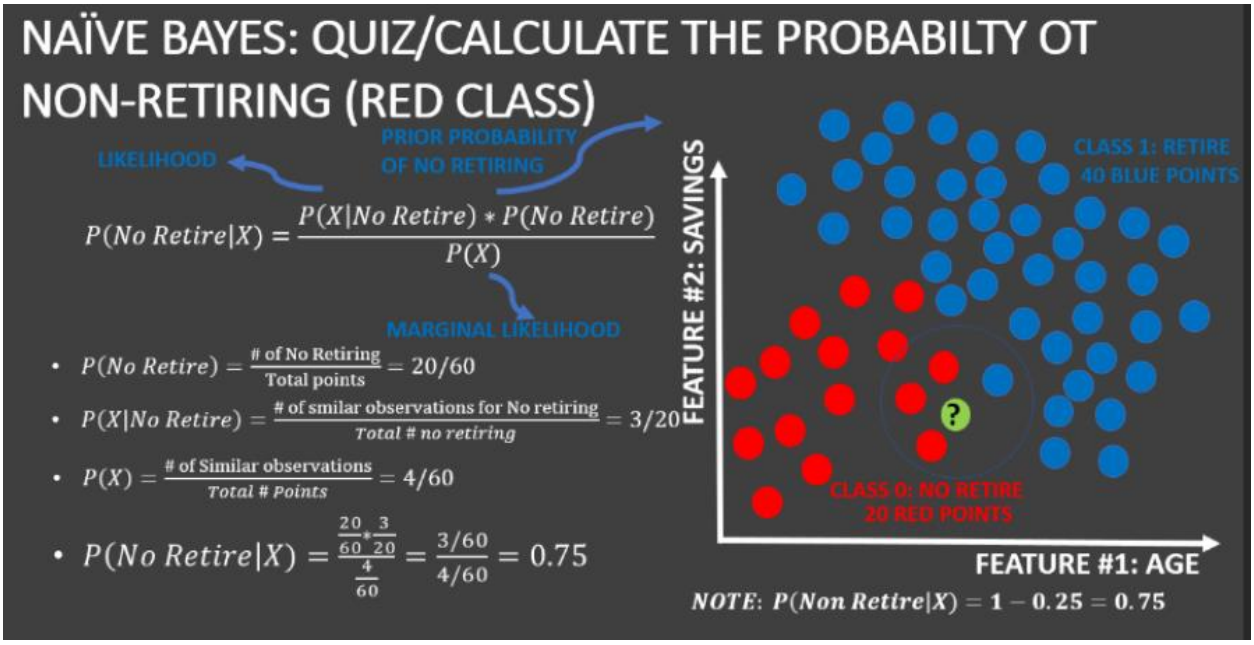
$$\textit{Posterior Probability of X being BLUE} = \textit{Prior Probability of BLUE} * \textit{Likelihood of X being BLUE} = \frac{40}{60} * \frac{1}{40} = \frac{1}{60}$$

NAÏVE BAYES: MATH (DON'T PANIC!)



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


NAÏVE BAYES: MATH (DON'T PANIC!)
PRIOR PROBABILITY OF RETIRING
LIKELIHOOD
$P(Retire|X) = \frac{P(X|Retire) * P(Retire)}{P(X)}$
MARGINAL LIKELIHOOD
• $P(Retire) = \frac{\#\text{ of Retiring}}{\text{Total points}} = 40/60$
• $P(X|Retire) = \frac{\#\text{ of smilar observations for retiring}}{Total\ \#\ retiring} = 1/40$
• $P(X) = \frac{\#\text{ of Similar observations}}{Total\ \#\ Points} = 4/60$
• $P(Retire|X) = \frac{\frac{40}{60} * \frac{1}{40}}{\frac{4}{60}} = \frac{1/60}{4/60} = 0.25$


NAÏVE BAYES: QUIZ/CALCULATE THE PROBABILTY OT NON-RETIRING (RED CLASS)
LIKELIHOOD
PRIOR PROBABILITY OF NO RETIRING
$P(No\ Retire|X) = \frac{P(X|No\ Retire) * P(No\ Retire)}{P(X)}$
MARGINAL LIKELIHOOD
• $P(No\ Retire) = \frac{\#\text{ of No Retiring}}{\text{Total points}} = 20/60$
• $P(X|No\ Retire) = \frac{\#\text{ of smilar observations for No retiring}}{Total\ \#\ no\ retiring} = 3/20$
• $P(X) = \frac{\#\text{ of Similar observations}}{Total\ \#\ Points} = 4/60$
• $P(No\ Retire|X) = \frac{\frac{20}{60} * \frac{3}{20}}{\frac{4}{60}} = \frac{3/60}{4/60} = 0.75$
FEATURE #2: SAVINGS
CLASS 1: RETIRE
40 BLUE POINTS
?
CLASS 0: NO RETIRE
20 RED POINTS
FEATURE #1: AGE
NOTE: $P(Non\ Retire|X) = 1 - 0.25 = 0.75$

TASK #10: TRAIN A NAIVE BAYES CLASSIFIER MODEL

In [73]:
```
X.shape
```

```
(31962, 47386)
```

In [74]:
```
y.shape
```

```
(31962,)
```

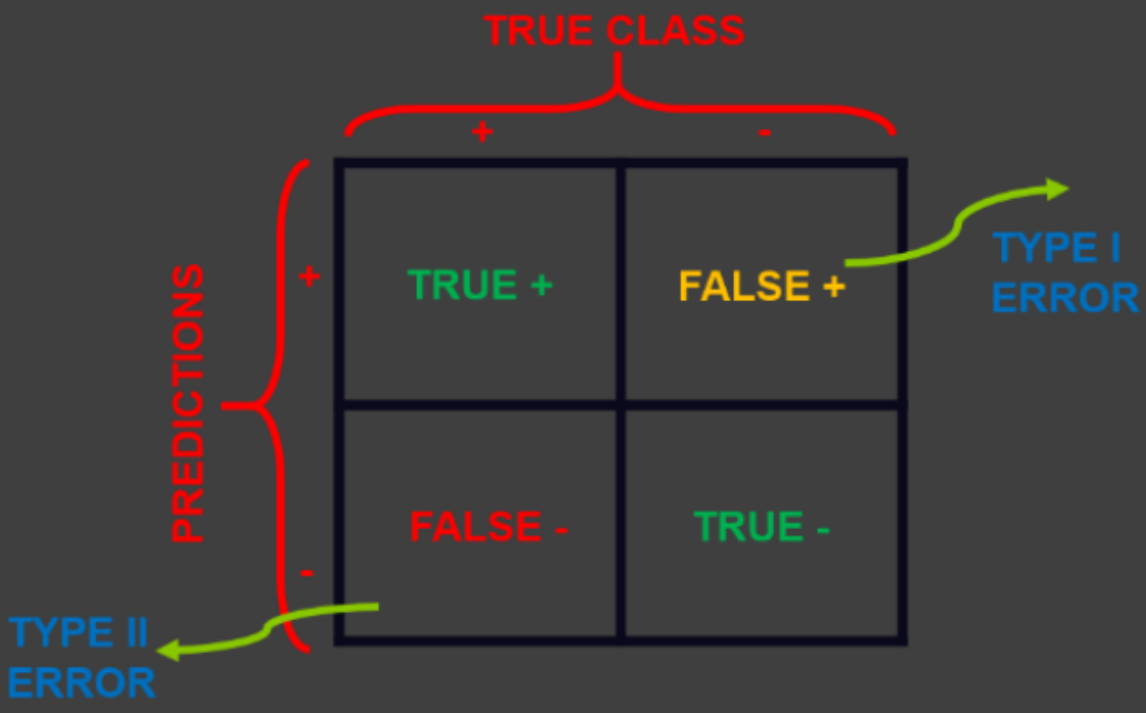
In [75]:
```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2)
```

In [76]:
```
from sklearn.naive_bayes import MultinomialNB

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

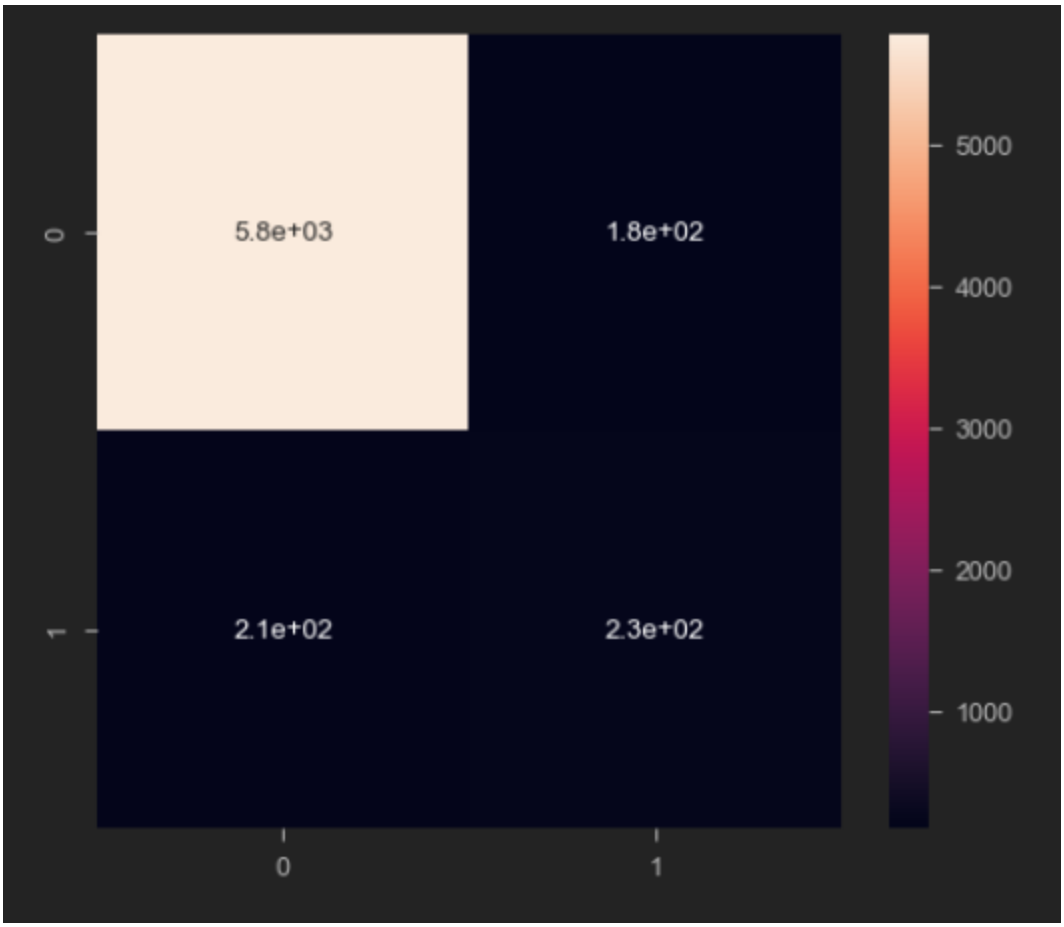
```
MultinomialNB(alpha=1.0, class_prior=None, fit_prior=True)
```

TASK #11: ASSESS TRAINED MODEL PERFORMANCE

CONFUSION MATRIX




```
In [77]: from sklearn.metrics import classification_report, confusion_matrix

In [78]: # 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 0x1a27a5f4bc8>
```




```
In [79]: print(classification_report(y_test, y_predict_test))
```

	precision	recall	f1-score	support
0	0.96	0.97	0.97	5954
1	0.56	0.51	0.54	439
accuracy			0.94	6393
macro avg	0.76	0.74	0.75	6393
weighted avg	0.94	0.94	0.94	6393