Importing the Dependencies

In [26]:
```
import numpy as np
import pandas as pd
```

In [27]:
```
full_data = pd.read_csv("data.csv")
full_data.head()
```

Out[27]:

	tweet_id	tweet	sentiment
0	1701	#sxswnui #sxsw #apple defining language of tou...	1
1	1851	Learning ab Google doodles! All doodles should...	1
2	2689	one of the most in-your-face ex. of stealing t...	2
3	4525	This iPhone #SXSW app would b pretty awesome i...	0
4	3604	Line outside the Apple store in Austin waiting...	1

Exploratory Data Analysis

In [28]:
```
full_data.shape
```

Out[28]:
```
(7274, 3)
```

In [29]:
```
full_data.size
```

Out[29]:
```
21822
```

In [30]:
```
full_data.describe()
```

Out[30]:

	tweet_id	sentiment
count	7274.000000	7274.000000
mean	4521.736071	1.299148
std	2617.858745	0.607829
min	2.000000	0.000000
25%	2261.000000	1.000000
50%	4530.500000	1.000000
75%	6796.750000	2.000000
max	9092.000000	3.000000

In [31]:
```
# getting some information about the dataset

full_data.info()
```

Out[31]:
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 7274 entries, 0 to 7273
Data columns (total 3 columns):
 #   Column       Non-Null Count  Dtype 
---  ------       --------------  ----- 
 0   tweet_id     7274 non-null   int64 
 1   tweet        7273 non-null   object
 2   sentiment    7274 non-null   int64 
dtypes: int64(2), object(1)
memory usage: 170.6+ KB
```

In [32]:
```
# remove serial number column as it is not adding any value.

full_data = full_data.drop(['tweet_id'],axis = 1)
full_data.head()
```

Out[32]:

	tweet	sentiment
0	#sxswnui #sxsw #apple defining language of tou...	1
1	Learning ab Google doodles! All doodles should...	1
2	one of the most in-your-face ex. of stealing t...	2
3	This iPhone #SXSW app would b pretty awesome i...	0
4	Line outside the Apple store in Austin waiting...	1

In [33]:
```
full_data['sentiment'].value_counts()
```

Out[33]:
```
1    4311
2    2382
0     456
3     125
Name: sentiment, dtype: int64
```

In [34]:
```
full_data.isnull().sum()
```

Out[34]:
```
tweet        1
sentiment    0
dtype: int64
```

In [35]:
```
# null values in %

full_data.isnull().mean()*100
```

Out[35]:
```
tweet        0.013748
sentiment    0.000000
dtype: float64
```

In [36]:
```
full_data.isnull().sum()
```

Out[36]:
```
tweet        1
sentiment    0
dtype: int64
```

In [37]:
```
full_data['tweet'].fillna(full_data['tweet'].mean,inplace = True)
```

In [38]:
```
full_data.isnull().sum()
```

Out[38]:
```
tweet        0
sentiment    0
dtype: int64
```

In [39]:
```
# subset on data
data = full_data[['tweet','sentiment']]
data.columns = ['X','y']
data.head()
```

Out[39]:

	X	y
0	#sxswnui #sxsw #apple defining language of tou...	1
1	Learning ab Google doodles! All doodles should...	1
2	one of the most in-your-face ex. of stealing t...	2
3	This iPhone #SXSW app would b pretty awesome i...	0
4	Line outside the Apple store in Austin waiting...	1

In [40]:
```
all_text = data['X']
all_text['X']= all_text['X'].str.lower()
```

```
C:\Users\aksha\AppData\Local\Temp\ipykernel_9976\1885637827.py:2: SettingWithCopy
Warning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stabl
e/user_guide/indexing.html#returning-a-view-versus-a-copy
  all_text['X']= all_text['X'].str.lower()
```

In [41]:
```
# first complaint
first_tweet = data.iloc[0][0]
first_tweet
```

Out[41]:
```
'#sxswnui #sxsw #apple defining language of touch with different dialects becomin
g smaller'
```

In [42]:
```
# apply word tokenizer
from nltk.tokenize import word_tokenize
word_token = word_tokenize(first_tweet)
print(word_token)
```

```
['#', 'sxswnui', '#', 'sxsw', '#', 'apple', 'defining', 'language', 'of', 'touc
h', 'with', 'different', 'dialects', 'becoming', 'smaller']
```

Lemmatizer

In [43]:
```
import nltk
nltk.download('wordnet')
```

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

Out[43]:
```
True
```

In [44]:
```
>>> nltk.download('omw-1.4')
```

```
[nltk_data] Downloading package omw-1.4 to
[nltk_data]     C:\Users\aksha\AppData\Roaming\nltk_data...
[nltk_data]   Package omw-1.4 is already up-to-date!
```

Out[44]:
```
True
```

In [45]:
```
from nltk.stem import WordNetLemmatizer
text = r"Natural language processing is really fun and i want to study it more"'
tokens = word_tokenize(text)
lemma = WordNetLemmatizer()
lemma_word = [lemma.lemmatize(i) for i in tokens]
print(lemma_word)
```

```
['``', 'Natural', 'language', 'processing', 'is', 'really', 'fun', 'and', 'i', 'w
ant', 'to', 'study', 'it', 'more', "''"]
```

Count vectorizer

In [46]:
```
from collections import Counter
count_vectorizer = Counter(word_token)
print(count_vectorizer)
```

```
Counter({'#': 3, 'sxswnui': 1, 'sxsw': 1, 'apple': 1, 'defining': 1, 'language':
1, 'of': 1, 'touch': 1, 'with': 1, 'different': 1, 'dialects': 1, 'becoming': 1,
'smaller': 1})
```

Stopwords

In [47]:
```
import nltk
nltk.download("stopwords")
from nltk.corpus import stopwords
print(set(stopwords.words('english')))
```

```
{"hasn", "which", "same", "isn't", "be", "it", "doesn't", "haven't", "nor", "aren
', "themselves", "that", "was", "to", "couldn't", "has", "for", "shouldn", "before
", "on", "doing", "all", "are", "you'd", "because", "with", "yourself", "d", "didn
", "there", "o", "what", "being", "mightn't", "down", "a", "only", "been", "do", "di
d", "or", "an", "wasn", "she", "too", "yourselves", "will", "other", "over", "y",
"weren't", "had", "should've", "as", "against", "you've", "his", "about", "your",
"ll", "i", "its", "mightn't", "when", "while", "so", "why", "wasn't", "ain", "hav
e", "can', "my", "it', "their", "shouldn't", "this', "our', "from', "they', "her
e", "mustn't", "you'll", "up", "in', "ma', "is', "more', "ve', "am', "you', "wo
n", "she's", "down', "hers', "not', "where', "off', "once', "very', "m', "such',
"most', "that'll", "s', "shan', "yours', "didn't", "into', "no', "further', "coul
dn', "above', "each', "shan't", "any', "below', "ourselves', "mustn't", "theirs',
"you're", "these', "herself', "needn', "hasn't", "wouldn't", "than', "aren't", "h
aving', "wouldn', "her', "himself', "isn't", "again', "now', "don', "hadn't", "th
rough', "don't", "myself', "but', "both', "does', "he', "them', "some', "the', "u
ntil', "it's", "won't", "ours', "own', "itself', "whom', "out', "under', "shoul
d', "those', "who', "we', "of', "needn't", "if', "during', "him', "re', "how', "f
ew', "by", "after', "between', "then', "hadn', "and', "haven', "weren', "just',
"we', "were', "at"}
[nltk_data] Downloading package stopwords to
[nltk_data]     C:\Users\aksha\AppData\Roaming\nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
```

Stopword removal from sample data

In [48]:
```
stop_words = set(stopwords.words('english'))
print("first_tweet : ", first_tweet)
print("================================================================")
bow = word_tokenize(first_tweet)
print("Words before performing stopword removal",bow)
print("Lenght of words before performing stopword removal : --------->",len(bow))
print("================================================================")
bow_stop_word_removed = [x for x in bow if x not in stop_words ]
print("Words after performing stopword removal",bow_stop_word_removed)
print("Lenght of words after performing stopword removal : --------->",len(bow_sto
p_word_removed))
```

```
first_tweet :  #sxswnui #sxsw #apple defining language of touch with different di
alects becoming smaller
================================================================
Words before performing stopword removal ['#', 'sxswnui', '#', 'sxsw', '#', 'appl
e', 'defining', 'language', 'of', 'touch', 'with', 'different', 'dialects', 'beco
ming', 'smaller']
Lenght of words before performing stopword removal : ---------> 15
================================================================
Words after performing stopword removal ['#', 'sxswnui', '#', 'sxsw', '#', 'appl
e', 'defining', 'language', 'touch', 'different', 'dialects', 'becoming', 'smalle
r']
Lenght of words after performing stopword removal : ---------> 13
```

Out[48]:
```
1
```

TF - IDF

In [49]:
```
from sklearn.metrics import accuracy_score,roc_auc_score
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.preprocessing import LabelEncoder
from sklearn.feature_extraction.text import TfidfVectorizer
```

Model Evaluation¶

In [51]:
```
# Replace missing values with empty string
all_text['X'].fillna('', inplace=True)
```

```
C:\Users\aksha\AppData\Local\Temp\ipykernel_9976\3400486798.py:2: SettingWithCopy
Warning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stabl
e/user_guide/indexing.html#returning-a-view-versus-a-copy
  all_text['X'].fillna('', inplace=True)
```

In [52]:
```
tfidf = TfidfVectorizer(stop_words = 'english')
vector = tfidf.fit_transform(all_text['X'])
X_tfidf = vector.toarray()
labels = data[['y']]
le = LabelEncoder()
labels['y'] = le.fit_transform(labels['y'])
X_train, X_test, y_train, y_test = train_test_split(X_tfidf,labels['y'],test_size=
log_reg_tfidf = LogisticRegression(random_state=42)
log_reg_tfidf.fit(X_train,y_train)
acc_tfidf = log_reg_tfidf.score(X_test,y_test)
print(acc_tfidf)
```

```
C:\Users\aksha\AppData\Local\Temp\ipykernel_9976\2839386268.py:6: SettingWithCopy
Warning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stabl
e/user_guide/indexing.html#returning-a-view-versus-a-copy
  labels['y'] = le.fit_transform(labels['y'])
0.6619331195602383
```

In [ ]:

In [ ]: