```
pip install vaderSentiment
```

```
Collecting vaderSentiment
  Downloading vaderSentiment-3.3.2-py2.py3-none-any.whl (125 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 126.0/126.0 kB 1.5 MB/s eta 0:00:00
Requirement already satisfied: requests in /usr/local/lib/python3.10/dist-packages (from vaderSentiment) (2.31.0)
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.10/dist-packages (from requests->vaderSentiment) (3.3.
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.10/dist-packages (from requests->vaderSentiment) (3.6)
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.10/dist-packages (from requests->vaderSentiment) (2.0.7)
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.10/dist-packages (from requests->vaderSentiment) (2023.11.17
Installing collected packages: vaderSentiment
Successfully installed vaderSentiment-3.3.2
```

```
import matplotlib.pyplot as plt
from vaderSentiment.vaderSentiment import SentimentIntensityAnalyzer
import seaborn as sns
from PIL import Image
from wordcloud import WordCloud, STOPWORDS, ImageColorGenerator
sns.set()
%matplotlib inline
from sklearn.pipeline import Pipeline
from sklearn.svm import SVC
from sklearn.ensemble import GradientBoostingClassifier
from sklearn.metrics import precision_score, recall_score, f1_score, accuracy_score
import warnings
warnings.filterwarnings("ignore")
import sklearn
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import CountVectorizer
from nltk.tokenize.treebank import TreebankWordDetokenizer
from sklearn.linear_model import LogisticRegression
from sklearn import metrics
from sklearn.naive_bayes import MultinomialNB
import nltk
from nltk.stem import WordNetLemmatizer,PorterStemmer
from nltk.tokenize import word_tokenize
from wordcloud import WordCloud, STOPWORDS
from nltk.corpus import stopwords
from collections import  Counter
nltk.download('punkt')
nltk.download('stopwords')
nltk.download('wordnet')
import re
import numpy as np
import pandas as pd
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

```
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]   Package punkt is already up-to-date!
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
[nltk_data] Downloading package wordnet to /root/nltk_data...
[nltk_data]   Package wordnet is already up-to-date!
```

```
data = pd.read_csv('BBC News.csv')
data.head(5)
```

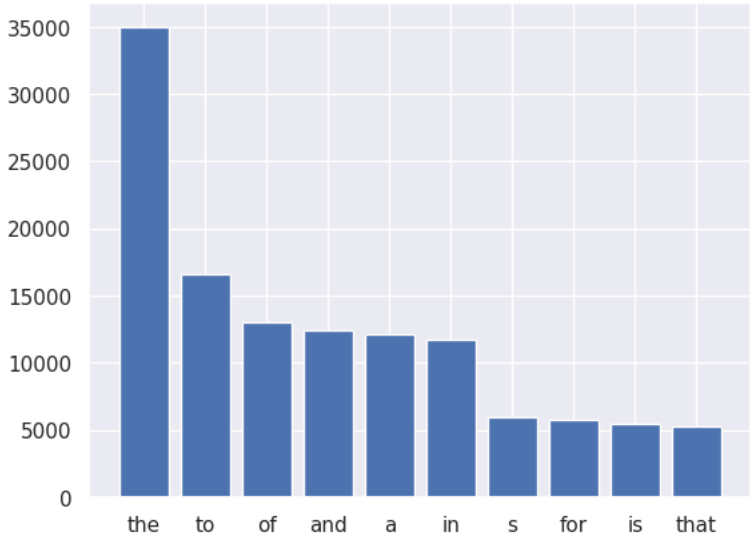
	ArticleId	Text	Category
0	1833	worldcom ex-boss launches defence lawyers defe...	business
1	154	german business confidence slides german busin...	business
2	1101	bbc poll indicates economic gloom citizens in ...	business
3	1976	lifestyle governs mobile choice faster bett...	tech
4	917	enron bosses in $168m payout eighteen former e...	business

```
data.Category.unique()
```

```
array(['business', 'tech', 'politics', 'sport', 'entertainment'],
      dtype=object)
```

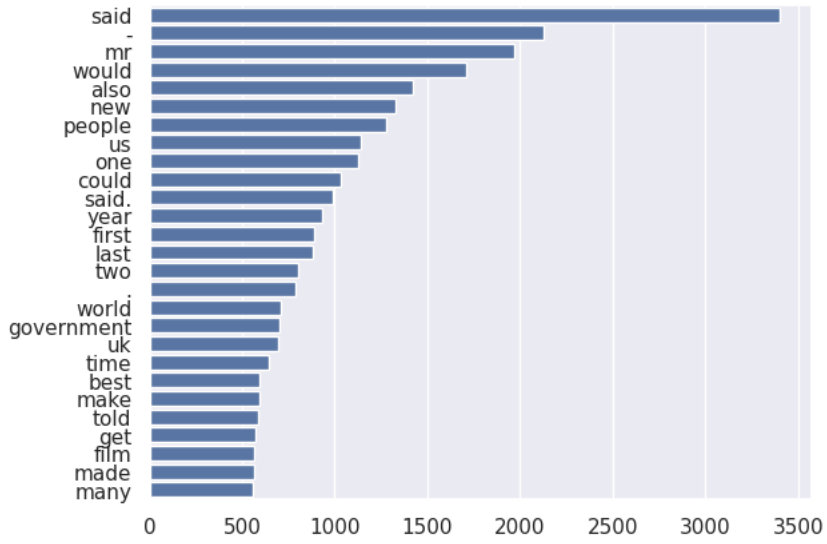
remove stop words

```
def plot_stopwords(data1):
    stop=set(stopwords.words('english'))
    data_split= data1.str.split()
    data_list=data_split.values.tolist()
    corpus=[word for i in data_list for word in i]
    from collections import defaultdict
    dictionary_stopwords=defaultdict(int)
    for word in corpus:
        if word in stop:
            dictionary_stopwords[word]+=1

    top=sorted(dictionary_stopwords.items(), key=lambda x:x[1],reverse=True)[:10]
    x,y=zip(*top)
    plt.bar(x,y)
plot_stopwords(data['Text'])
```



counting most frequesnt words

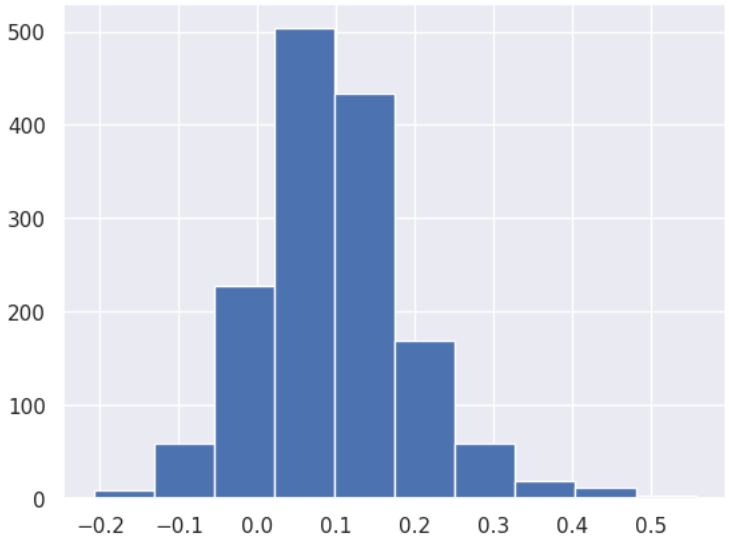
```
def top_frequent_words(data1):
    stop=set(stopwords.words('english'))
    data_split= data1.str.split()
    data_list=data_split.values.tolist()
    corpus=[word for i in data_list for word in i]
    counter=Counter(corpus)
    mostCommon=counter.most_common()
    x, y=[], []
    for word,count in mostCommon[:100]:
        if (word not in stop):
            x.append(word)
            y.append(count)
    sns.barplot(x=y,y=x)
top_frequent_words(data['Text'])
```




```
def plot_wordcloud(data):
    stop=set(stopwords.words('english'))
    def _preprocess_text(data):
        corpus=[]
        stem=PorterStemmer()
        lem=WordNetLemmatizer()
        for news in data:
            words=[w for w in word_tokenize(news) if (w not in stop)]

            words=[lem.lemmatize(w) for w in words if len(w)>2]

            corpus.append(words)
        return corpus

    corpus=_preprocess_text(data)

    wordcloud = WordCloud(

          background_color='white',
        stopwords=set(STOPWORDS),
        max_words=200,
        max_font_size=25,
        scale=3,
        random_state=1)

    wordcloud=wordcloud.generate(str(corpus))

    fig = plt.figure(1, figsize=(12, 12))
    plt.axis('off')

    plt.imshow(wordcloud)
    plt.show()
plot_wordcloud(data['Text'])
```

plotting positive, negative, neutral sentiment

```
from textblob import TextBlob
def polarity_histo(data):

    def check_semantics(data):
        return TextBlob(data).sentiment.polarity

    polarity_val =data.apply(lambda a : check_semantics(a))
    polarity_val.hist()
polarity_histo(data['Text'])
```



```
analyzer = SentimentIntensityAnalyzer()

# Define function for sentiment analysis using VADER
def analyze_sentiment_vader(text):
    """
    Perform sentiment analysis using VADER.
    """
    # Analyze sentiment
    sentiment_scores = analyzer.polarity_scores(text)
     # Determine sentiment label based on compound score
    if sentiment_scores['compound'] >= 0.05:
        sentiment = "Positive"
    elif sentiment_scores['compound'] <= -0.05:
        sentiment = "Negative"
    else:
        sentiment = "Neutral"

    return sentiment, sentiment_scores['compound']

# Apply sentiment analysis function to each row in the 'text' column
data[['Sentiment', 'Compound Score']] = data['Text'].apply(lambda x: pd.Series(analyze_sentiment_vader(x)))

# Print the DataFrame with sentiment analysis results
print(data)
```

```
      ArticleId                                               Text  Category  \
0          1833  worldcom ex-boss launches defence lawyers defe...         1
1           154  german business confidence slides german busin...         1
2          1101  bbc poll indicates economic gloom citizens in ...         1
3          1976  lifestyle  governs mobile choice  faster  bett...         0
4           917  enron bosses in $168m payout eighteen former e...         1
...         ...                                                ...       ...
1485        857  double eviction from big brother model caprice...         3
1486        325  dj double act revamp chart show dj duo jk and ...         3
1487       1590  weak dollar hits reuters revenues at media gro...         1
1488       1587  apple ipod family expands market apple has exp...         0
1489        538  santy worm makes unwelcome visit thousands of ...         0

                                             text_clean Sentiment  Polarity  \
0     worldcom ex-boss launches defence lawyers defe...  Negative  0.023320
1     german business confidence slides german busin...  Positive  0.077917
2     bbc poll indicates economic gloom citizens maj...  Negative  0.077754
3     lifestyle governs mobile choice faster better ...  Positive  0.156109
4     enron bosses payout eighteen former enron dire...  Negative  0.009685
...                                                 ...       ...       ...
1485  double eviction big brother model caprice holb...  Positive  0.107292
1486  dj double act revamp chart show dj duo jk joel...  Positive  0.221375
1487  weak dollar hits reuters revenues media group ...  Positive  0.066856
1488  apple ipod family expands market apple expande...  Positive  0.203186
1489  santy worm makes unwelcome visit thousands web...  Negative  0.015103

      Subjectivity  Compound Score
0         0.395125         -0.9609
1         0.353854          0.6967
2         0.390688         -0.9243
3         0.478536          0.9578
4         0.342117         -0.9486
...            ...             ...
1485      0.340278          0.9670
1486      0.458355          0.9928
1487      0.417992          0.9595
1488      0.427091          0.9983
1489      0.512602         -0.8510

[1490 rows x 8 columns]
```

```
analyze_sentiment_vader("Snowfall finally returns to Kashmir valley after prolonged dry spell")
```

```
('Neutral', 0.0)
```

```
analyze_sentiment_vader("Putin praises India's independent foreign policy, warns West not to 'play games'")
```

```
('Positive', 0.2415)
```

```
print(data)
```

```
      ArticleId                                               Text  Category  \
0          1833  worldcom ex-boss launches defence lawyers defe...         1
1           154  german business confidence slides german busin...         1
```

```
2         1101  bbc poll indicates economic gloom citizens in ...         1
3         1976  lifestyle  governs mobile choice  faster  bett...         0
4          917  enron bosses in $168m payout eighteen former e...         1
...        ...                                                ...       ...
1485       857  double eviction from big brother model caprice...         3
1486       325  dj double act revamp chart show dj duo jk and ...         3
1487      1590  weak dollar hits reuters revenues at media gro...         1
1488      1587  apple ipod family expands market apple has exp...         0
1489       538  santy worm makes unwelcome visit thousands of ...         0

                                          text_clean Sentiment  Polarity  \
0     worldcom ex-boss launches defence lawyers defe...  Positive  0.023320
1     german business confidence slides german busin...  Positive  0.077917
2     bbc poll indicates economic gloom citizens maj...  Positive  0.077754
3     lifestyle governs mobile choice faster better ...  Positive  0.156109
4     enron bosses payout eighteen former enron dire...  Positive  0.009685
...                                                 ...       ...       ...
1485  double eviction big brother model caprice holb...  Positive  0.107292
1486  dj double act revamp chart show dj duo jk joel...  Positive  0.221375
1487  weak dollar hits reuters revenues media group ...  Positive  0.066856
1488  apple ipod family expands market apple expande...  Positive  0.203186
1489  santy worm makes unwelcome visit thousands web...  Positive  0.015103

      Subjectivity
0         0.395125
1         0.353854
2         0.390688
3         0.478536
4         0.342117
...            ...
1485      0.340278
1486      0.458355
1487      0.417992
1488      0.427091
1489      0.512602

[1490 rows x 7 columns]
```

data cleaning

```
def cleaning(bbc_text):
  if len(bbc_text)==1:
      word_tokens = word_tokenize(bbc_text)
  else:
      print('Wait! Data is getting cleaned...')
      # Tokenize : dividing Sentences into words
      bbc_text['text_clean'] = data['Text'].apply(nltk.word_tokenize)
      print('Step 1-Tokenization Done!.')
      print(bbc_text.head(5))

  # Remove stop words
  if len(data)==1:
      stop_words = set(stopwords.words('english'))
      filtered_sentence = [w for w in word_tokens if not w in stop_words]
      filtered_sentence = []
      for w in word_tokens:
          if w not in stop_words:
             filtered_sentence.append(w)
  else:
      stop_words=set(nltk.corpus.stopwords.words("english"))
      data['text_clean'] = data['text_clean'].apply(lambda x: [item for item in x if item not in stop_words])
      print('Step 2-All stop words are removed from the list.')
      print(data.head(5))
  #Will keep words and remove numbers and special characters
  if len(data)!=1:
      regex = '[a-z]+'
      data['text_clean'] = data['text_clean'].apply(lambda x: [char for char in x if re.match(regex, char)])
      print('Step3-Numbers and Special Characters are removed.')
      print(data.head(5))

cleaning(data)
```

```
    Wait! Data is getting cleaned...
    Step 1-Tokenization Done!.
       ArticleId                                               Text  Category  \
    0       1833  worldcom ex-boss launches defence lawyers defe...  business
    1        154  german business confidence slides german busin...  business
    2       1101  bbc poll indicates economic gloom citizens in ...  business
    3       1976  lifestyle  governs mobile choice  faster  bett...      tech
```

```
4       917  enron bosses in $168m payout eighteen former e...  business

                                        text_clean
0  [worldcom, ex-boss, launches, defence, lawyers...
1  [german, business, confidence, slides, german,...
2  [bbc, poll, indicates, economic, gloom, citize...
3  [lifestyle, governs, mobile, choice, faster, b...
4  [enron, bosses, in, $, 168m, payout, eighteen,...
Step 2-All stop words are removed from the list.
   ArticleId                                               Text  Category  \
0       1833  worldcom ex-boss launches defence lawyers defe...  business
1        154  german business confidence slides german busin...  business
2       1101  bbc poll indicates economic gloom citizens in ...  business
3       1976  lifestyle  governs mobile choice  faster  bett...      tech
4        917  enron bosses in $168m payout eighteen former e...  business

                                        text_clean
0  [worldcom, ex-boss, launches, defence, lawyers...
1  [german, business, confidence, slides, german,...
2  [bbc, poll, indicates, economic, gloom, citize...
3  [lifestyle, governs, mobile, choice, faster, b...
4  [enron, bosses, $, 168m, payout, eighteen, for...
Step3-Numbers and Special Characters are removed.
   ArticleId                                               Text  Category  \
0       1833  worldcom ex-boss launches defence lawyers defe...  business
1        154  german business confidence slides german busin...  business
2       1101  bbc poll indicates economic gloom citizens in ...  business
3       1976  lifestyle  governs mobile choice  faster  bett...      tech
4        917  enron bosses in $168m payout eighteen former e...  business

                                        text_clean
0  [worldcom, ex-boss, launches, defence, lawyers...
1  [german, business, confidence, slides, german,...
2  [bbc, poll, indicates, economic, gloom, citize...
3  [lifestyle, governs, mobile, choice, faster, b...
4  [enron, bosses, payout, eighteen, former, enro...
```

```
print(len(data))
def detokenize(data):
  for i in range(len(data)):
    data_w = data['text_clean'][i]
    a=TreebankWordDetokenizer().detokenize(data_w)
    data.at[i,'text_clean']=a
detokenize(data)
```

```
1490
```

```
print(data['text_clean'])
```

```
0       worldcom ex-boss launches defence lawyers defe...
1       german business confidence slides german busin...
2       bbc poll indicates economic gloom citizens maj...
3       lifestyle governs mobile choice faster better ...
4       enron bosses payout eighteen former enron dire...
                              ...
1485    double eviction big brother model caprice holb...
1486    dj double act revamp chart show dj duo jk joel...
1487    weak dollar hits reuters revenues media group ...
1488    apple ipod family expands market apple expande...
1489    santy worm makes unwelcome visit thousands web...
Name: text_clean, Length: 1490, dtype: object
```

```
data.Category = data.Category.map({'tech':0, 'business':1, 'sport':2, 'entertainment':3, 'politics':4})
data.Category.unique()
```

```
array([1, 0, 4, 2, 3])
```

```
data.isnull().sum()
```

```
ArticleId     0
Text          0
Category      0
text_clean    0
dtype: int64
```

split data in testing and training sets

```
X = data.text_clean
y = data.Category
#split data
X_train, X_test, y_train, y_test = train_test_split(X, y, train_size = 0.6, random_state = 1)
```

count vectorization

```
vector = CountVectorizer(stop_words = 'english',lowercase=False)
# fit the vectorizer on the training data
vector.fit(X_train)
# print(len(vector.get_feature_names()))
vector.vocabulary_
X_transformed = vector.transform(X_train)
# print(X_transformed.toarray())
X_transformed.toarray()
# for test data
X_test_transformed = vector.transform(X_test)
print(X_test_transformed.toarray)
```

```
<bound method _cs_matrix.toarray of <596x18593 sparse matrix of type '<class 'numpy.int64'>'
        with 75478 stored elements in Compressed Sparse Row format>>
```

Logistic Regression

```
logistic_reg = LogisticRegression()
logistic_reg.fit(X_transformed, y_train)
```

```
▾ LogisticRegression
LogisticRegression()
```

```
LogisticRegression()
```

```
▾ LogisticRegression
LogisticRegression()
```

```
# fit
logistic_reg.fit(X_transformed,y_train)

# predict class
y_predicted = logistic_reg.predict(X_test_transformed)

# predict probabilities
y_pred_probability = logistic_reg.predict_proba(X_test_transformed)
```

```
metrics.accuracy_score(y_test, y_predicted)
```

```
0.9681208053691275
```

confusion matrix

```
confusion_mat = metrics.confusion_matrix(y_test, y_predicted)
print(confusion_mat)
TrueNeg = confusion_mat[0, 0]
TruePos = confusion_mat[1, 1]
FalseNeg = confusion_mat[1, 0]
FalsePos = confusion_mat[0, 1]
sensitivity = TruePos / float(FalseNeg + TruePos)
print("sensitivity",sensitivity)

specificity = TrueNeg / float(TrueNeg + FalsePos)
print("specificity",specificity)
```

```
[[107   1   2   0   0]
 [  3 116   0   0   3]
 [  0   0 137   0   0]
 [  0   0   2 111   2]
 [  0   3   3   0 106]]
sensitivity 0.9747899159663865
specificity 0.9907407407407407
```

```
PRECISION_SCORE = metrics.precision_score(y_test, y_predicted, average = 'micro')
RECALL_SCORE = metrics.recall_score(y_test, y_predicted, average = 'micro')
F1_SCORE =metrics.f1_score(y_test, y_predicted, average = 'micro')

print("PRECISION SCORE :",PRECISION_SCORE)
print("RECALL SCORE :",RECALL_SCORE)
print("F1 SCORE :",F1_SCORE)
```

```
PRECISION SCORE : 0.9681208053691275
RECALL SCORE : 0.9681208053691275
F1 SCORE : 0.9681208053691275
```

using naive bayes classifier to classify text

```
naivebayes = MultinomialNB()
naivebayes.fit(X_transformed, y_train)
```

```
▾ MultinomialNB
MultinomialNB()
```

```
MultinomialNB()
```

```
▾ MultinomialNB
MultinomialNB()
```

```
naivebayes.fit(X_transformed,y_train)
# predict class
y_predict = naivebayes.predict(X_test_transformed)
# predict probabilities
y_pred_probability = naivebayes.predict_proba(X_test_transformed)
```

```
metrics.accuracy_score(y_test, y_predict)
```

```
0.9731543624161074
```

```
metrics.confusion_matrix(y_test, y_predict)
```

```
array([[109,   0,   1,   0,   0],
       [  4, 114,   0,   0,   4],
       [  0,   0, 137,   0,   0],
       [  4,   0,   0, 110,   1],
       [  0,   2,   0,   0, 110]])
```

```
confusion = metrics.confusion_matrix(y_test, y_predict)
print(confusion)
TrueNeg = confusion_mat[0, 0]
TruePos = confusion_mat[1, 1]
FalseNeg = confusion_mat[1, 0]
FalsePos = confusion_mat[0, 1]
sensitivity = TruePos / float(FalseNeg + TruePos)
print("sensitivity",sensitivity)

specificity = TrueNeg / float(TrueNeg + FalsePos)
print("specificity",specificity)
```

```
[[109   0   1   0   0]
 [  4 114   0   0   4]
 [  0   0 137   0   0]
 [  4   0   0 110   1]
 [  0   2   0   0 110]]
sensitivity 0.9747899159663865
specificity 0.9907407407407407
```

```
PRECISION_SCORE = metrics.precision_score(y_test, y_predicted, average = 'micro')
RECALL_SCORE = metrics.recall_score(y_test, y_predicted, average = 'micro')
F1_SCORE =metrics.f1_score(y_test, y_predicted, average = 'micro')

print("PRECISION SCORE :",PRECISION_SCORE)
print("RECALL SCORE :",RECALL_SCORE)
```