

```
In [1]:
from sklearn.pipeline import Pipeline
from sklearn.model_selection import GridSearchCV, train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.feature_extraction.text import TfidfVectorizer
from nltk.stem.porter import PorterStemmer
from nltk.corpus import stopwords
import nltk
import re
from sklearn.naive_bayes import MultinomialNB, BernoulliNB, GaussianNB
import pandas as pd
import string
from sklearn import feature_extraction, linear_model, model_selection, preprocessing
from sklearn.metrics import accuracy_score
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression

In [2]:
# Opening the file
f = open("amazon_cells_labelled.txt", "r")

data =[]
# Converting it to pandas dataframe
for line in f:
    review = line[:len(line) - 2]
    sentiment = line[len(line)-2]
    row = [review, sentiment]
    data.append(row)

df = pd.DataFrame(data, columns = ['reviews', 'sentiment'])

In [3]:
# Opening the file
f = open("yelp_labelled.txt", "r")

data =[]
# Converting it to pandas dataframe
for line in f:
    review = line[:len(line) - 2]
    sentiment = line[len(line)-2]
    row = [review, sentiment]
    data.append(row)

df1 = pd.DataFrame(data, columns = ['reviews', 'sentiment'])

In [4]:
# Opening the file
f = open("imdb_labelled.txt", "r")

data =[]
# Converting it to pandas dataframe
for line in f:
    review = line[:len(line) - 2]
    sentiment = line[len(line)-2]
    row = [review, sentiment]
    data.append(row)

df2 = pd.DataFrame(data, columns = ['reviews', 'sentiment'])

In [6]:
len(df)

Out[6]: 1000

In [7]:
df = df.append(df1).append(df2)

In [8]:
len(df)

Out[8]: 3000

In [11]:
import pandas as pd
from nltk.tokenize import word_tokenize
from nltk.corpus import stopwords

#removing the stop words from the corpus
stop = stopwords.words('english')

df['reviews'] = df.reviews.str.replace("[^\w\s]", "").str.lower()
df['reviews'] = df['reviews'].apply(lambda x: ' '.join([item for item in x.split() if item not in stop]))

<ipython-input-11-ab282e7c83b5>:10: FutureWarning: The default value of regex will change from True to False in a future version
df['reviews'] = df.reviews.str.replace("[^\w\s]", "").str.lower()

In [12]:
df.head()

Out[12]:
      reviews sentiment
0  way plug us unless go converter      0
1  good case excellent value      1
2  great jawbone      1
3  tied charger conversations lasting 45 minutes...      0
4  mic great      1

In [13]:
#function to remove numeric characters

def remove_numeric_character(x):
    x = re.sub(r'\b[0-9]+\b\s+', '', x)
    return x

#function to remove punctuations
def remove_punctuations(text):
    for punctuation in string.punctuation:
        text = text.replace(punctuation, '')
    return text

#function to clean the string
def clean_string(s):
    tokens = s.split().split()
    clean_tokens = [t for t in tokens if re.match(r'[^\\Wd]*$', t)]
    clean_s = ' '.join(clean_tokens)
    return clean_s

In [14]:
#mapping the above functions

df['reviews'] = df['reviews'].apply(remove_punctuations)
df["reviews"] = df['reviews'].apply(remove_numeric_character)
df["reviews"] = df['reviews'].apply(clean_string)

In [15]:
df.info()

<class 'pandas.core.frame.DataFrame'>
Int64Index: 3000 entries, 0 to 999
Data columns (total 2 columns):
 #   Column      Non-Null Count  Dtype
---  -
 0   reviews    3000 non-null    object
 1   sentiment   3000 non-null    object
dtypes: object(2)
memory usage: 70.3+ KB

In [16]:
#train test split and creating pipeline

x_train,x_test,y_train,y_test = train_test_split(df["reviews"], df.sentiment, test_size=0.33, random_state=20)
pipe_lr = Pipeline([('tfidf', TfidfVectorizer()), ('clf', LogisticRegression(random_state=42))])

In [17]:
#parameters for logistic regression

param_range_fl = [0.1, 1, 10, 100]
grid_params_lr = [{'clf_penalty': ['l1', 'l2'],
                    'clf_C': param_range_fl,
                    'clf_solver': ['newton-cg', 'liblinear']}]]

In [28]:
#Logistic regression grid CV

LR = GridSearchCV(estimator=pipe_lr,
                  param_grid=grid_params_lr,
                  scoring='roc_auc',
                  cv=10, verbose = 3 )

In [29]:
LR.fit(x_train, y_train)

Fitting 10 folds for each of 16 candidates, totalling 160 fits
[CV 1/10] END clf_C=0.1, clf_penalty=l1, clf_solver=newton-cg, score=nan total time= 0.0s
[CV 2/10] END clf_C=0.1, clf_penalty=l1, clf_solver=newton-cg, score=nan total time= 0.0s
[CV 3/10] END clf_C=0.1, clf_penalty=l1, clf_solver=newton-cg, score=nan total time= 0.0s
[CV 4/10] END clf_C=0.1, clf_penalty=l1, clf_solver=newton-cg, score=nan total time= 0.0s
[CV 5/10] END clf_C=0.1, clf_penalty=l1, clf_solver=newton-cg, score=nan total time= 0.0s
[CV 6/10] END clf_C=0.1, clf_penalty=l1, clf_solver=newton-cg, score=nan total time= 0.0s
[CV 7/10] END clf_C=0.1, clf_penalty=l1, clf_solver=newton-cg, score=nan total time= 0.0s
[CV 8/10] END clf_C=0.1, clf_penalty=l1, clf_solver=newton-cg, score=nan total time= 0.0s
[CV 9/10] END clf_C=0.1, clf_penalty=l1, clf_solver=newton-cg, score=nan total time= 0.0s
[CV 10/10] END clf_C=0.1, clf_penalty=l1, clf_solver=newton-cg, score=nan total time= 0.0s
[CV 1/10] END clf_C=0.1, clf_penalty=l1, clf_solver=liblinear, score=0.599 total time= 0.0s
[CV 2/10] END clf_C=0.1, clf_penalty=l1, clf_solver=liblinear, score=0.593 total time= 0.0s
[CV 3/10] END clf_C=0.1, clf_penalty=l1, clf_solver=liblinear, score=0.590 total time= 0.0s
[CV 4/10] END clf_C=0.1, clf_penalty=l1, clf_solver=liblinear, score=0.598 total time= 0.0s
[CV 5/10] END clf_C=0.1, clf_penalty=l1, clf_solver=liblinear, score=0.592 total time= 0.0s
[CV 6/10] END clf_C=0.1, clf_penalty=l1, clf_solver=liblinear, score=0.618 total time= 0.0s
[CV 7/10] END clf_C=0.1, clf_penalty=l1, clf_solver=liblinear, score=0.598 total time= 0.0s
[CV 8/10] END clf_C=0.1, clf_penalty=l1, clf_solver=liblinear, score=0.600 total time= 0.0s
[CV 9/10] END clf_C=0.1, clf_penalty=l1, clf_solver=liblinear, score=0.600 total time= 0.0s
[CV 10/10] END clf_C=0.1, clf_penalty=l1, clf_solver=liblinear, score=0.612 total time= 0.0s
[CV 1/10] END clf_C=0.1, clf_penalty=l2, clf_solver=newton-cg, score=0.834 total time= 0.0s
[CV 2/10] END clf_C=0.1, clf_penalty=l2, clf_solver=newton-cg, score=0.865 total time= 0.0s
[CV 3/10] END clf_C=0.1, clf_penalty=l2, clf_solver=newton-cg, score=0.846 total time= 0.0s
[CV 4/10] END clf_C=0.1, clf_penalty=l2, clf_solver=newton-cg, score=0.813 total time= 0.0s
[CV 5/10] END clf_C=0.1, clf_penalty=l2, clf_solver=newton-cg, score=0.859 total time= 0.0s
[CV 6/10] END clf_C=0.1, clf_penalty=l2, clf_solver=newton-cg, score=0.846 total time= 0.0s
[CV 7/10] END clf_C=0.1, clf_penalty=l2, clf_solver=newton-cg, score=0.882 total time= 0.0s
[CV 8/10] END clf_C=0.1, clf_penalty=l2, clf_solver=newton-cg, score=0.844 total time= 0.0s
[CV 9/10] END clf_C=0.1, clf_penalty=l2, clf_solver=newton-cg, score=0.887 total time= 0.0s
[CV 10/10] END clf_C=0.1, clf_penalty=l2, clf_solver=liblinear, score=0.875 total time= 0.0s
[CV 1/10] END clf_C=0.1, clf_penalty=l2, clf_solver=liblinear, score=0.834 total time= 0.0s
[CV 2/10] END clf_C=0.1, clf_penalty=l2, clf_solver=liblinear, score=0.814 total time= 0.0s
[CV 3/10] END clf_C=0.1, clf_penalty=l2, clf_solver=liblinear, score=0.852 total time= 0.0s
[CV 4/10] END clf_C=0.1, clf_penalty=l2, clf_solver=liblinear, score=0.855 total time= 0.0s
[CV 5/10] END clf_C=0.1, clf_penalty=l2, clf_solver=liblinear, score=0.875 total time= 0.0s
[CV 6/10] END clf_C=0.1, clf_penalty=l2, clf_solver=liblinear, score=0.882 total time= 0.0s
[CV 7/10] END clf_C=0.1, clf_penalty=l2, clf_solver=liblinear, score=0.882 total time= 0.0s
[CV 8/10] END clf_C=0.1, clf_penalty=l2, clf_solver=liblinear, score=0.844 total time= 0.0s
[CV 9/10] END clf_C=0.1, clf_penalty=l2, clf_solver=liblinear, score=0.876 total time= 0.0s
[CV 10/10] END clf_C=0.1, clf_penalty=l2, clf_solver=liblinear, score=0.887 total time= 0.0s
[CV 1/10] END clf_C=1, clf_penalty=l1, clf_solver=newton-cg, score=nan total time= 0.0s
[CV 2/10] END clf_C=1, clf_penalty=l1, clf_solver=newton-cg, score=nan total time= 0.0s
[CV 3/10] END clf_C=1, clf_penalty=l1, clf_solver=newton-cg, score=nan total time= 0.0s
[CV 4/10] END clf_C=1, clf_penalty=l1, clf_solver=newton-cg, score=nan total time= 0.0s
[CV 5/10] END clf_C=1, clf_penalty=l1, clf_solver=newton-cg, score=nan total time= 0.0s
[CV 6/10] END clf_C=1, clf_penalty=l1, clf_solver=newton-cg, score=nan total time= 0.0s
[CV 7/10] END clf_C=1, clf_penalty=l1, clf_solver=newton-cg, score=nan total time= 0.0s
[CV 8/10] END clf_C=1, clf_penalty=l1, clf_solver=newton-cg, score=nan total time= 0.0s
[CV 9/10] END clf_C=1, clf_penalty=l1, clf_solver=newton-cg, score=nan total time= 0.0s
[CV 10/10] END clf_C=1, clf_penalty=l1, clf_solver=newton-cg, score=nan total time= 0.0s
[CV 1/10] END clf_C=1, clf_penalty=l2, clf_solver=newton-cg, score=0.855 total time= 0.0s
[CV 2/10] END clf_C=1, clf_penalty=l2, clf_solver=newton-cg, score=0.886 total time= 0.0s
[CV 3/10] END clf_C=1, clf_penalty=l2, clf_solver=newton-cg, score=0.831 total time= 0.0s
[CV 4/10] END clf_C=1, clf_penalty=l2, clf_solver=newton-cg, score=0.869 total time= 0.0s
[CV 5/10] END clf_C=1, clf_penalty=l2, clf_solver=newton-cg, score=0.877 total time= 0.0s
[CV 6/10] END clf_C=1, clf_penalty=l2, clf_solver=newton-cg, score=0.867 total time= 0.0s
[CV 7/10] END clf_C=1, clf_penalty=l2, clf_solver=newton-cg, score=0.884 total time= 0.0s
[CV 8/10] END clf_C=1, clf_penalty=l2, clf_solver=newton-cg, score=0.858 total time= 0.0s
[CV 9/10] END clf_C=1, clf_penalty=l2, clf_solver=newton-cg, score=0.895 total time= 0.0s
[CV 10/10] END clf_C=1, clf_penalty=l2, clf_solver=newton-cg, score=0.911 total time= 0.0s
[CV 1/10] END clf_C=1, clf_penalty=l2, clf_solver=liblinear, score=0.855 total time= 0.0s
[CV 2/10] END clf_C=1, clf_penalty=l2, clf_solver=liblinear, score=0.886 total time= 0.0s
[CV 3/10] END clf_C=1, clf_penalty=l2, clf_solver=liblinear, score=0.831 total time= 0.0s
[CV 4/10] END clf_C=1, clf_penalty=l2, clf_solver=liblinear, score=0.869 total time= 0.0s
[CV 5/10] END clf_C=1, clf_penalty=l2, clf_solver=liblinear, score=0.877 total time= 0.0s
[CV 6/10] END clf_C=1, clf_penalty=l2, clf_solver=liblinear, score=0.867 total time= 0.0s
[CV 7/10] END clf_C=1, clf_penalty=l2, clf_solver=liblinear, score=0.873 total time= 0.0s
[CV 8/10] END clf_C=1, clf_penalty=l2, clf_solver=liblinear, score=0.858 total time= 0.0s
[CV 9/10] END clf_C=1, clf_penalty=l2, clf_solver=liblinear, score=0.895 total time= 0.0s
[CV 10/10] END clf_C=1, clf_penalty=l2, clf_solver=liblinear, score=0.911 total time= 0.0s
[CV 1/10] END clf_C=10, clf_penalty=l1, clf_solver=newton-cg, score=nan total time= 0.0s
[CV 2/10] END clf_C=10, clf_penalty=l1, clf_solver=newton-cg, score=nan total time= 0.0s
[CV 3/10] END clf_C=10, clf_penalty=l1, clf_solver=newton-cg, score=nan total time= 0.0s
[CV 4/10] END clf_C=10, clf_penalty=l1, clf_solver=newton-cg, score=nan total time= 0.0s
[CV 5/10] END clf_C=10, clf_penalty=l1, clf_solver=newton-cg, score=nan total time= 0.0s
[CV 6/10] END clf_C=10, clf_penalty=l1, clf_solver=newton-cg, score=nan total time= 0.0s
[CV 7/10] END clf_C=10, clf_penalty=l1, clf_solver=newton-cg, score=nan total time= 0.0s
[CV 8/10] END clf_C=10, clf_penalty=l1, clf_solver=newton-cg, score=nan total time= 0.0s
[CV 9/10] END clf_C=10, clf_penalty=l1, clf_solver=newton-cg, score=nan total time= 0.0s
[CV 10/10] END clf_C=10, clf_penalty=l1, clf_solver=newton-cg, score=nan total time= 0.0s
[CV 1/10] END clf_C=10, clf_penalty=l2, clf_solver=newton-cg, score=0.806 total time= 0.0s
[CV 2/10] END clf_C=10, clf_penalty=l2, clf_solver=newton-cg, score=0.846 total time= 0.0s
[CV 3/10] END clf_C=10, clf_penalty=l2, clf_solver=newton-cg, score=0.875 total time= 0.0s
[CV 4/10] END clf_C=10, clf_penalty=l2, clf_solver=newton-cg, score=0.859 total time= 0.0s
[CV 5/10] END clf_C=10, clf_penalty=l2, clf_solver=newton-cg, score=0.875 total time= 0.0s
[CV 6/10] END clf_C=10, clf_penalty=l2, clf_solver=newton-cg, score=0.884 total time= 0.0s
[CV 7/10] END clf_C=10, clf_penalty=l2, clf_solver=newton-cg, score=0.895 total time= 0.0s
[CV 8/10] END clf_C=10, clf_penalty=l2, clf_solver=newton-cg, score=0.829 total time= 0.0s
[CV 9/10] END clf_C=10, clf_penalty=l2, clf_solver=newton-cg, score=0.884 total time= 0.0s
[CV 10/10] END clf_C=10, clf_penalty=l2, clf_solver=newton-cg, score=0.857 total time= 0.0s
[CV 1/10] END clf_C=100, clf_penalty=l1, clf_solver=liblinear, score=0.871 total time= 0.0s
[CV 2/10] END clf_C=100, clf_penalty=l1, clf_solver=liblinear, score=0.811 total time= 0.0s
[CV 3/10] END clf_C=100, clf_penalty=l1, clf_solver=liblinear, score=0.855 total time= 0.0s
[CV 4/10] END clf_C=100, clf_penalty=l1, clf_solver=liblinear, score=0.873 total time= 0.0s
[CV 5/10] END clf_C=100, clf_penalty=l1, clf_solver=liblinear, score=0.876 total time= 0.0s
[CV 6/10] END clf_C=100, clf_penalty=l1, clf_solver=liblinear, score=0.875 total time= 0.0s
[CV 7/10] END clf_C=100, clf_penalty=l1, clf_solver=liblinear, score=0.875 total time= 0.0s
[CV 8/10] END clf_C=100, clf_penalty=l1, clf_solver=liblinear, score=0.855 total time= 0.0s
[CV 9/10] END clf_C=100, clf_penalty=l1, clf_solver=liblinear, score=0.895 total time= 0.0s
[CV 10/10] END clf_C=100, clf_penalty=l1, clf_solver=liblinear, score=0.902 total time= 0.0s
[CV 1/10] END clf_C=100, clf_penalty=l2, clf_solver=newton-cg, score=nan total time= 0.0s
[CV 2/10] END clf_C=100, clf_penalty=l2, clf_solver=newton-cg, score=nan total time= 0.0s
[CV 3/10] END clf_C=100, clf_penalty=l2, clf_solver=newton-cg, score=nan total time= 0.0s
[CV 4/10] END clf_C=100, clf_penalty=l2, clf_solver=newton-cg, score=nan total time= 0.0s
[CV 5/10] END clf_C=100, clf_penalty=l2, clf_solver=newton-cg, score=nan total time= 0.0s
[CV 6/10] END clf_C=100, clf_penalty=l2, clf_solver=newton-cg, score=nan total time= 0.0s
[CV 7/10] END clf_C=100, clf_penalty=l2, clf_solver=newton-cg, score=nan total time= 0.0s
[CV 8/10] END clf_C=100, clf_penalty=l2, clf_solver=newton-cg, score=nan total time= 0.0s
[CV 9/10] END clf_C=100, clf_penalty=l2, clf_solver=newton-cg, score=nan total time= 0.0s
[CV 10/10] END clf_C=100, clf_penalty=l2, clf_solver=newton-cg, score=nan total time= 0.0s
[CV 1/10] END clf_C=100, clf_penalty=l2, clf_solver=liblinear, score=0.862 total time= 0.0s
[CV 2/10] END clf_C=100, clf_penalty=l2, clf_solver=liblinear, score=0.871 total time= 0.0s
[CV 3/10] END clf_C=100, clf_penalty=l2, clf_solver=liblinear, score=0.831 total time= 0.0s
[CV 4/10] END clf_C=100, clf_penalty=l2, clf_solver=liblinear, score=0.873 total time= 0.0s
[CV 5/10] END clf_C=100, clf_penalty=l2, clf_solver=liblinear, score=0.876 total time= 0.0s
[CV 6/10] END clf_C=100, clf_penalty=l2, clf_solver=liblinear, score=0.875 total time= 0.0s
[CV 7/10] END clf_C=100, clf_penalty=l2, clf_solver=liblinear, score=0.875 total time= 0.0s
[CV 8/10] END clf_C=100, clf_penalty=l2, clf_solver=liblinear, score=0.855 total time= 0.0s
[CV 9/10] END clf_C=100, clf_penalty=l2, clf_solver=liblinear, score=0.884 total time= 0.0s
[CV 10/10] END clf_C=100, clf_penalty=l2, clf_solver=liblinear, score=0.884 total time= 0.0s
/Users/sharanbasavasumbad/python-env/env/lib/python3.9/site-packages/sklearn/svm/_base.py:1206: ConvergenceWarn
ing: Liblinear failed to converge, increase the number of iterations.
[CV 1/10] END clf_C=10, clf_penalty=l1, clf_solver=liblinear, score=0.818 total time= 2.3s
/Users/sharanbasavasumbad/python-env/env/lib/python3.9/site-packages/sklearn/svm/_base.py:1206: ConvergenceWarn
ing: Liblinear failed to converge, increase the number of iterations.
[CV 2/10] END clf_C=10, clf_penalty=l1, clf_solver=liblinear, score=0.865 total time= 2.4s
[CV 3/10] END clf_C=10, clf_penalty=l1, clf_solver=liblinear, score=0.788 total time= 0.0s
/Users/sharanbasavasumbad/python-env/env/lib/python3.9/site-packages/sklearn/svm/_base.py:1206: ConvergenceWarn
ing: Liblinear failed to converge, increase the number of iterations.
[CV 4/10] END clf_C=10, clf_penalty=l1, clf_solver=liblinear, score=0.875 total time= 2.5s
[CV 5/10] END clf_C=10, clf_penalty=l1, clf_solver=liblinear, score=0.865 total time= 0.0s
/Users/sharanbasavasumbad/python-env/env/lib/python3.9/site-packages/sklearn/svm/_base.py:1206: ConvergenceWarn
ing: Liblinear failed to converge, increase the number of iterations.
[CV 6/10] END clf_C=10, clf_penalty=l1, clf_solver=liblinear, score=0.846 total time= 2.3s
[CV 7/10] END clf_C=10, clf_penalty=l1, clf_solver=liblinear, score=0.850 total time= 0.0s
/Users/sharanbasavasumbad/python-env/env/lib/python3.9/site-packages/sklearn/svm/_base.py:1206: ConvergenceWarn
ing: Liblinear failed to converge, increase the number of iterations.
[CV 8/10] END clf_C=10, clf_penalty=l1, clf_solver=liblinear, score=0.827 total time= 2.2s
[CV 9/10] END clf_C=10, clf_penalty=l1, clf_solver=liblinear, score=0.884 total time= 0.0s
/Users/sharanbasavasumbad/python-env/env/lib/python3.9/site-packages/sklearn/svm/_base.py:1206: ConvergenceWarn
ing: Liblinear failed to converge, increase the number of iterations.
[CV 10/10] END clf_C=10, clf_penalty=l1, clf_solver=liblinear, score=0.888 total time= 2.4s
[CV 1/10] END clf_C=10, clf_penalty=l2, clf_solver=newton-cg, score=0.861 total time= 0.0s
[CV 2/10] END clf_C=10, clf_penalty=l2, clf_solver=newton-cg, score=0.885 total time= 0.0s
[CV 3/10] END clf_C=10, clf_penalty=l2, clf_solver=newton-cg, score=0.831 total time= 0.0s
[CV 4/10] END clf_C=10, clf_penalty=l2, clf_solver=newton-cg, score=0.873 total time= 0.0s
[CV 5/10] END clf_C=10, clf_penalty=l2, clf_solver=newton-cg, score=0.876 total time= 0.0s
[CV 6/10] END clf_C=10, clf_penalty=l2, clf_solver=newton-cg, score=0.876 total time= 0.0s
[CV 7/10] END clf_C=10, clf_penalty=l2, clf_solver=newton-cg, score=0.875 total time= 0.0s
[CV 8/10] END clf_C=10, clf_penalty=l2, clf_solver=newton-cg, score=0.855 total time= 0.0s
[CV 9/10] END clf_C=10, clf_penalty=l2, clf_solver=newton-cg, score=0.897 total time= 0.0s
[CV 10/10] END clf_C=10, clf_penalty=l2, clf_solver=newton-cg, score=0.902 total time= 0.0s
[CV 1/10] END clf_C=10, clf_penalty=l2, clf_solver=liblinear, score=0.862 total time= 0.0s
[CV 2/10] END clf_C=10, clf_penalty=l2, clf_solver=liblinear, score=0.871 total time= 0.0s
[CV 3/10] END clf_C=10, clf_penalty=l2, clf_solver=liblinear, score=0.831 total time= 0.0s
[CV 4/10] END clf_C=10, clf_penalty=l2, clf_solver=liblinear, score=0.873 total time= 0.0s
[CV 5/10] END clf_C=10, clf_penalty=l2, clf_solver=liblinear, score=0.876 total time= 0.0s
[CV 6/10] END clf_C=10, clf_penalty=l2, clf_solver=liblinear, score=0.875 total time= 0.0s
[CV 7/10] END clf_C=10, clf_penalty=l2, clf_solver=liblinear, score=0.875 total time= 0.0s
[CV 8/10] END clf_C=10, clf_penalty=l2, clf_solver=liblinear, score=0.855 total time= 0.0s
[CV 9/10] END clf_C=10, clf_penalty=l2, clf_solver=liblinear, score=0.884 total time= 0.0s
[CV 10/10] END clf_C=10, clf_penalty=l2, clf_solver=liblinear, score=0.884 total time= 0.0s
/Users/sharanbasavasumbad/python-env/env/lib/python3.9/site-packages/sklearn/model_selection/_validation.py:37
2: FitFailedWarning:
40 fits failed out of a total of 160.
The score on these train-test partitions for these parameters will be set to nan.
If these failures are not expected, you can try to debug them by setting error_score='raise'.

Below are more details about the failures:
-----
40 fits failed with the following error:
Traceback (most recent call last):
  File "/Users/sharanbasavasumbad/python-env/env/lib/python3.9/site-packages/sklearn/model_selection/_validation.py", line 680, in fit_and_score
    estimator.fit(X_train, y_train, **fit_params)
  File "/Users/sharanbasavasumbad/python-env/env/lib/python3.9/site-packages/sklearn/pipeline.py", line 394, in fit
    self._final_estimator.fit(Xt, y, **fit_params_last_step)
  File "/Users/sharanbasavasumbad/python-env/env/lib/python3.9/site-packages/sklearn/linear_model/_logistic.py", line 1461, in fit
    solver = check_solver(self.solver, self.penalty, self.dual)
  File "/Users/sharanbasavasumbad/python-env/env/lib/python3.9/site-packages/sklearn/linear_model/_logistic.py", line 447, in check_solver
    raise ValueError(
ValueError: Solver newton-cg supports only 'l2' or 'none' penalties, got l1 penalty.

warnings.warn(some fits failed message, FitFailedWarning)
/Users/sharanbasavasumbad/python-env/env/lib/python3.9/site-packages/sklearn/model_selection/_search.py:969: UserWarning: One or more of the test scores are non-finite: [          nan  0.59994243  0.85572191  0.85569222          nan
0.83108869
0.87343898  0.87342906          nan  0.8505382  0.87267647  0.87268638
          nan  0.83006262  0.85563438  0.8556443]
warnings.warn(

GridSearchCV(cv=10,
              estimator=Pipeline(steps=[('tfidf', TfidfVectorizer()),
                                       ('clf',
                                        LogisticRegression(random_state=42))]),
              param_grid=[{'clf_C': [0.1, 1, 10, 100],
                           'clf_penalty': ['l1', 'l2'],
                           'clf_solver': ['newton-cg', 'liblinear']}],
              scoring='roc_auc', verbose=3)

In [30]:
print("Best: %f using %s" % (LR.best_score_, LR.best_params_))

Best: 0.873439 using {'clf_C': 1, 'clf_penalty': 'l2', 'clf_solver': 'newton-cg'})

In [31]:
print("Logistic Regression best grid score: " + str(LR.best_score_))
print("Logistic Regression best test score: " + str(LR.score(x_test, y_test)))

LR_best_params = LR.best_params_
print("Logistic Regression best params: " + str(LR_best_params))

Logistic Regression best grid score: 0.8734389967509751
Logistic Regression grid test score: 0.8798686543209374
Logistic Regression best params: {'clf_C': 1, 'clf_penalty': 'l2', 'clf_solver': 'newton-cg'})

In [32]:
!jupyter-nbconvert --to PDFViaHTML Assignment9_sharanbasav.ipynb

[NbConvertApp] Converting notebook Assignment9_sharanbasav.ipynb to PDFViaHTML
[NbConvertApp] Writing 174514 bytes to Assignment9_sharanbasav.pdf
```