Project - by Harsha Teja Bolla

Help Twitter Combat Hate Speech Using NLP and Machine Learning Project 2

DESCRIPTION

Using NLP and ML, make a model to identify hate speech (racist or sexist tweets) in Twitter.

Problem Statement:

Twitter is the biggest platform where anybody and everybody can have their views heard. Some of these voices spread hate and negativity. Twitter is wary of its platform being used as a medium to spread hate.

You are a data scientist at Twitter, and you will help Twitter in identifying the tweets with hate speech and removing them from the platform. You will use NLP techniques, perform specific cleanup for tweets data, and make a robust model.

Domain: Social Media

Analysis to be done: Clean up tweets and build a classification model by using NLP techniques, cleanup specific for tweets data, regularization and hyperparameter tuning using stratified k-fold and cross validation to get the best model.

Content:

id: identifier number of the tweet

Label: 0 (non-hate) /1 (hate)

Tweet: the text in the tweet

Tasks:

Load the tweets file using read_csv function from Pandas package.

Get the tweets into a list for easy text cleanup and manipulation.

To cleanup:

Normalize the casing.

Using regular expressions, remove user handles. These begin with '@'.

Using regular expressions, remove URLs.

Using TweetTokenizer from NLTK, tokenize the tweets into individual terms.

Remove stop words.

Remove redundant terms like 'amp', 'rt', etc.

Remove '#' symbols from the tweet while retaining the term.

Extra cleanup by removing terms with a length of 1.

Check out the top terms in the tweets:

First, get all the tokenized terms into one large list.

Use the counter and find the 10 most common terms.

Data formatting for predictive modeling:

Join the tokens back to form strings. This will be required for the vectorizers.

Assign x and y.

Perform train_test_split using sklearn.

We'll use TF-IDF values for the terms as a feature to get into a vector space model.

Import TF-IDF vectorizer from sklearn.

Instantiate with a maximum of 5000 terms in your vocabulary.

Fit and apply on the train set.

Apply on the test set.

Model building: Ordinary Logistic Regression

Instantiate Logistic Regression from sklearn with default parameters.

Fit into the train data.

Make predictions for the train and the test set.

Model evaluation: Accuracy, recall, and f_1 score.

Report the accuracy on the train set.

Report the recall on the train set: decent, high, or low.

Get the f1 score on the train set.

Looks like you need to adjust the class imbalance, as the model seems to focus on the 0s.

Adjust the appropriate class in the LogisticRegression model.

Train again with the adjustment and evaluate.

Train the model on the train set.

Evaluate the predictions on the train set: accuracy, recall, and f_1 score.

Regularization and Hyperparameter tuning:

Import GridSearch and StratifiedKFold because of class imbalance.

Provide the parameter grid to choose for 'C' and 'penalty' parameters.

Use a balanced class weight while instantiating the logistic regression.

Find the parameters with the best recall in cross validation.

Choose 'recall' as the metric for scoring.

Choose stratified 4 fold cross validation scheme.

Fit into the train set.

What are the best parameters?

Predict and evaluate using the best estimator.

Use the best estimator from the grid search to make predictions on the test set.

What is the recall on the test set for the toxic comments?

What is the f_1 score?

In [1]:
```
cd C:\Users\harsha.teja\Desktop\myg\NLP\major r=project\project 2
```

```
C:\Users\harsha.teja\Desktop\myg\NLP\major r=project\project 2
```

In [133…
```
import pandas as pd
import numpy as np
import pandas as pd
import re
import nltk
import spacy
import string
```

In [158…
```
df = pd .read_csv("TwitterHate.csv")
```

In [159…
```
df.head()
```

Out[159…

	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

In [160…
```
df['tweet']  = df['tweet'].str.lower()
df.head()
```

Out[160…

	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

In [161…
```
#removing punctuation, creating a new column called 'text_punct]'
df['text_punct'] = df['tweet'].str.replace('[^\w\s]','')
df.head()
```

Out[161…

	id	label	tweet	text_punct
0	1	0	@user when a father is dysfunctional and is s...	user when a father is dysfunctional and is so...
1	2	0	@user @user thanks for #lyft credit i can't us...	user user thanks for lyft credit i cant use ca...
2	3	0	bihday your majesty	bihday your majesty
3	4	0	#model i love u take with u all the time in ...	model i love u take with u all the time in u...
4	5	0	factsguide: society now #motivation	factsguide society now motivation

In [162…
```
def clean_url(review_text):
    return re.sub(r'http\S+', ' ',review_text)
```

In [163…
```
df['text_punct'] = df['text_punct'].apply(clean_url)
```

In [164…
```
def removing_nonalhanumeric(review_text):
    return re.sub('[^a-zA-Z]', ' ',review_text)
```

In [165…
```
df['text_punct'] = df['text_punct'].apply(removing_nonalhanumeric)
```

In [168…
```
df['text_punct'] = df['text_punct'].replace('((25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)(\.
```

In [169…
```
#remove email adress
df['text_punct'] = df['text_punct'].replace('[a-zA-Z0-9-_.]+@[a-zA-Z0-9-_.]+', '', rege
```

In [170…
```
#remove punctaitions and special chracters
df['text_punct'] = df['text_punct'].str.replace('[^\w\s]','')
```

In [171…
```
from nltk.tokenize import TweetTokenizer
tweet_tokenizer = TweetTokenizer()
def tweet_token(sent):
    return tweet_tokenizer.tokenize(sent)
df['text_tokens'] = df['text_punct'].apply(tweet_token)
df.head()
```

Out[171…

	id	label	tweet	text_punct	text_tokens
0	1	0	@user when a father is dysfunctional and is s...	user when a father is dysfunctional and is so...	[user, when, a, father, is, dysfunctional, and...
1	2	0	@user @user thanks for #lyft credit i can't us...	user user thanks for lyft credit i cant use ca...	[user, user, thanks, for, lyft, credit, i, can...
2	3	0	bihday your majesty	bihday your majesty	[bihday, your, majesty]
3	4	0	#model i love u take with u all the time in ...	model i love u take with u all the time in u...	[model, i, love, u, take, with, u, all, the, t...
4	5	0	factsguide: society now #motivation	factsguide society now motivation	[factsguide, society, now, motivation]

In [180…
```
#Importing stopwords from nltk library
from nltk.corpus import stopwords
from string import punctuation

stop_nltk = stopwords.words('english')
stop_punct = list(punctuation)
```

In [181…
```
stop_punct.extend(['...',"'",".."])
```

In [182…
```
stop_context = ['rt','amp']
```

In [183…
```
stop_final = stop_nltk+stop_punct+stop_context
```

In [184…
```
# Function to remove the stopwords
def del_stop(sent):
    return [re.sub("#", '',term) for term in sent if((term not in stop_final) &(len(ter
```

In [185…
```
df["text_stop"] = df["text_tokens"].apply(del_stop)
df["text_stop"].head()
```

Out[185…
```
0    [user, father, dysfunctional, selfish, drags, ...
1    [user, user, thanks, lyft, credit, cant, use, ...
2                                    [bihday, majesty]
3                       [model, love, take, time, ur]
4                    [factsguide, society, motivation]
Name: text_stop, dtype: object
```

In [186…
```
# Checking the first 10 most frequent words
from collections import Counter
```

```
cnt = Counter()
for text in df["text_stop"].values:
    for word in text:
        cnt[word] += 1

cnt.most_common(15)
```

Out[186…
```
[('user', 17511),
 ('love', 2742),
 ('day', 2301),
 ('happy', 1698),
 ('like', 1157),
 ('im', 1145),
 ('life', 1139),
 ('time', 1129),
 ('today', 1013),
 ('new', 989),
 ('positive', 934),
 ('thankful', 925),
 ('get', 920),
 ('people', 867),
 ('bihday', 858)]
```

In [187…
```
df.head()
```

Out[187…

	id	label	tweet	text_punct	text_tokens	text_stop
0	1	0	@user when a father is dysfunctional and is s...	user when a father is dysfunctional and is so...	[user, when, a, father, is, dysfunctional, and...	[user, father, dysfunctional, selfish, drags, ...
1	2	0	@user @user thanks for #lyft credit i can't us...	user user thanks for lyft credit i cant use ca...	[user, user, thanks, for, lyft, credit, i, can...	[user, user, thanks, lyft, credit, cant, use, ...
2	3	0	bihday your majesty	bihday your majesty	[bihday, your, majesty]	[bihday, majesty]
3	4	0	#model i love u take with u all the time in ...	model i love u take with u all the time in u...	[model, i, love, u, take, with, u, all, the, t...	[model, love, take, time, ur]
4	5	0	factsguide: society now #motivation	factsguide society now motivation	[factsguide, society, now, motivation]	[factsguide, society, motivation]

In [188…
```
def to_string(listreview):
    return ' '.join(listreview)
```

In [189…
```
df['processed_text'] = df['text_stop'].apply(to_string)
```

In [190…
```
df.head()
```

Out[190…

	id	label	tweet	text_punct	text_tokens	text_stop	processed_text

	id	label	tweet	text_punct	text_tokens	text_stop	processed_text
0	1	0	@user when a father is dysfunctional and is s...	user when a father is dysfunctional and is so...	[user, when, a, father, is, dysfunctional, and...	[user, father, dysfunctional, selfish, drags, ...	user father dysfunctional selfish drags kids d...
1	2	0	@user @user thanks for #lyft credit i can't us...	user user thanks for lyft credit i cant use ca...	[user, user, thanks, for, lyft, credit, i, can...	[user, user, thanks, lyft, credit, cant, use, ...	user user thanks lyft credit cant use cause do...
2	3	0	bihday your majesty	bihday your majesty	[bihday, your, majesty]	[bihday, majesty]	bihday majesty
3	4	0	#model i love u take with u all the time in ...	model i love u take with u all the time in u...	[model, i, love, u, take, with, u, all, the, t...	[model, love, take, time, ur]	model love take time ur
4	5	0	factsguide: society now #motivation	factsguide society now motivation	[factsguide, society, now, motivation]	[factsguide, society, motivation]	factsguide society motivation

In [191…
```
X= df['processed_text']
y = df['label']
```

In [192…
```
X.head()
```

Out[192…
```
0    user father dysfunctional selfish drags kids d...
1    user user thanks lyft credit cant use cause do...
2                                      bihday majesty
3                              model love take time ur
4                        factsguide society motivation
Name: processed_text, dtype: object
```

In [218…
```
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,random_state = 4
```

In [219…
```
X_train.shape,y_train.shape
```

Out[219…
```
((25569,), (25569,))
```

In [220…
```
X_test.shape,y_test.shape
```

Out[220…
```
((6393,), (6393,))
```

In [221…
```
from sklearn.feature_extraction.text import TfidfVectorizer
tfid = TfidfVectorizer(  input='content', encoding='utf-8',
                         decode_error='strict',
                         strip_accents=None, lowercase=True,
                         preprocessor=None, tokenizer=None,
                         analyzer='word', stop_words=None,
                         token_pattern='(?u)\\b\\w\\w+\\b',
                         ngram_range=(1, 2), max_df=0.5, min_df=1,
```

```
                                max_features=5000, vocabulary=None,
                                binary=False,
                                 norm='l2',
                                use_idf=True, smooth_idf=True, sublinear_tf=False)
```

In [222…
```
X_train = tfid.fit_transform(X_train)
```

In [223…
```
TfidfVectorizer
```

Out[223… sklearn.feature_extraction.text.TfidfVectorizer

In [224…
```
from sklearn.linear_model import LogisticRegression
```

In [225…
```
li =  LogisticRegression()
```

In [226…
```
li.fit(X_train,y_train)
```

Out[226… LogisticRegression()

In [227…
```
X_test = tfid.transform(X_test)
```

In [228…
```
Pred = li.predict(X_test)
```

In [229…
```
from sklearn.metrics import accuracy_score
```

In [230…
```
 accuracy_score(y_test,Pred)
```

Out[230… 0.9513530423901142

In [231…
```
from sklearn.metrics import classification_report
```

In [232…
```
print(classification_report(y_test,Pred))
```

```
              precision    recall  f1-score   support

           0       0.95      1.00      0.97      5945
           1       0.92      0.33      0.49       448

    accuracy                           0.95      6393
   macro avg       0.94      0.67      0.73      6393
weighted avg       0.95      0.95      0.94      6393
```

In [233…
```
df.head()
```

Out[233...

	id	label	tweet	text_punct	text_tokens	text_stop	processed_text
0	1	0	@user when a father is dysfunctional and is s...	user when a father is dysfunctional and is so...	[user, when, a, father, is, dysfunctional, and...	[user, father, dysfunctional, selfish, drags, ...	user father dysfunctional selfish drags kids d...
1	2	0	@user @user thanks for #lyft credit i can't us...	user user thanks for lyft credit i cant use ca...	[user, user, thanks, for, lyft, credit, i, can...	[user, user, thanks, lyft, credit, cant, use, ...	user user thanks lyft credit cant use cause do...
2	3	0	bihday your majesty	bihday your majesty	[bihday, your, majesty]	[bihday, majesty]	bihday majesty
3	4	0	#model i love u take with u all the time in ...	model i love u take with u all the time in u...	[model, i, love, u, take, with, u, all, the, t...	[model, love, take, time, ur]	model love take time ur
4	5	0	factsguide: society now #motivation	factsguide society now motivation	[factsguide, society, now, motivation]	[factsguide, society, motivation]	factsguide society motivation

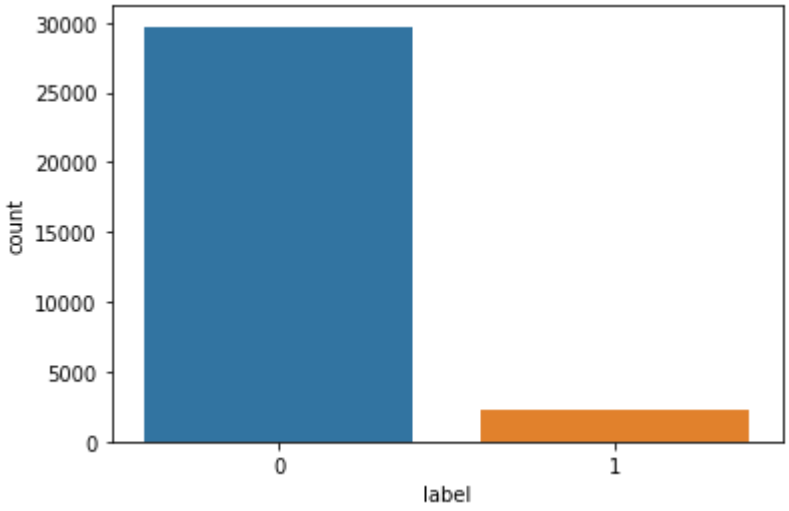
In [236...

```
import seaborn as sns
import matplotlib.pyplot as plt
```

In [237...

```
sns.countplot(df['label'], data = df)
plt.show()
```

```
C:\Users\harsha.teja\Anaconda3\envs\NLP\lib\site-packages\seaborn\_decorators.py:43: Fut
ureWarning: Pass the following variable as a keyword arg: x. From version 0.12, the only
valid positional argument will be `data`, and passing other arguments without an explici
t keyword will result in an error or misinterpretation.
  FutureWarning
```



In [239...

```
from imblearn.under_sampling import RandomUnderSampler
uos =  RandomUnderSampler(sampling_strategy=1)
X_train_uos,y_train_uos =uos.fit_sample(X_train,y_train)
X_train_uos.shape,y_train_uos.shape
print("orginal values {} ".format(Counter(y)))
print("underresampled data {} ".format(Counter(y_train_uos)))
```

```
orginal values Counter({0: 29720, 1: 2242})
underresampled data Counter({0: 1794, 1: 1794})
```

In [240…
```
from imblearn.over_sampling import RandomOverSampler
os =  RandomOverSampler(sampling_strategy='minority')
X_train_os,y_train_os = os.fit_sample(X_train,y_train)
print("orginal values {} ".format(Counter(y)))
print("resampled data {} ".format(Counter(y_train_os)))
```

```
orginal values Counter({0: 29720, 1: 2242})
resampled data Counter({0: 23775, 1: 23775})
```

In [241…
```
from sklearn.model_selection import RepeatedStratifiedKFold
from sklearn.tree import DecisionTreeClassifier
from sklearn.model_selection import cross_val_score
from numpy import mean
```

In [242…
```
de = DecisionTreeClassifier()
```

In [245…
```
X_train, X_test, y_train, y_test = train_test_split(X_train_os,y_train_os, test_size=0.
```

In [246…
```
lr = LogisticRegression(C = 0.001, penalty=  'l2',class_weight="balanced")
```

In [247…
```
lr_model = lr.fit(X_train,y_train)
```

In [248…
```
lrpred1 = lr.predict(X_test)
```

In [249…
```
accuracy_score(y_test,lrpred1)*100
```

Out[249…
```
77.33964248159832
```

In [250…
```
print(classification_report(y_test,lrpred1))
```

```
              precision    recall  f1-score   support

           0       0.73      0.85      0.79      4711
           1       0.83      0.70      0.76      4799

    accuracy                           0.77      9510
   macro avg       0.78      0.77      0.77      9510
weighted avg       0.78      0.77      0.77      9510
```

In [251…
```
from sklearn.model_selection import GridSearchCV
```

In [252…
```
clf =  LogisticRegression()
```

In [253…
```
grid_values  = {'penalty':['L1','l2'],
                'C': np.logspace(-3,3,7)}
```

In [255…
```
from sklearn.model_selection import GridSearchCV, StratifiedKFold
```

In [256…
```
grid_acc = GridSearchCV(clf,param_grid = grid_values, scoring='accuracy',verbose=0, pre
```

In [257…
```
grid_acc.fit(X_train,y_train)
```

```
C:\Users\harsha.teja\Anaconda3\envs\NLP\lib\site-packages\sklearn\model_selection\_valid
ation.py:552: FitFailedWarning: Estimator fit failed. The score on this train-test parti
tion for these parameters will be set to nan. Details:
Traceback (most recent call last):
  File "C:\Users\harsha.teja\Anaconda3\envs\NLP\lib\site-packages\sklearn\model_selectio
n\_validation.py", line 531, in _fit_and_score
    estimator.fit(X_train, y_train, **fit_params)
  File "C:\Users\harsha.teja\Anaconda3\envs\NLP\lib\site-packages\sklearn\linear_model\_
logistic.py", line 1304, in fit
    solver = _check_solver(self.solver, self.penalty, self.dual)
  File "C:\Users\harsha.teja\Anaconda3\envs\NLP\lib\site-packages\sklearn\linear_model\_
logistic.py", line 439, in _check_solver
    " got %s." % (all_penalties, penalty))
ValueError: Logistic Regression supports only penalties in ['l1', 'l2', 'elasticnet', 'n
one'], got L1.

  FitFailedWarning)
C:\Users\harsha.teja\Anaconda3\envs\NLP\lib\site-packages\sklearn\model_selection\_valid
ation.py:552: FitFailedWarning: Estimator fit failed. The score on this train-test parti
tion for these parameters will be set to nan. Details:
Traceback (most recent call last):
  File "C:\Users\harsha.teja\Anaconda3\envs\NLP\lib\site-packages\sklearn\model_selectio
n\_validation.py", line 531, in _fit_and_score
    estimator.fit(X_train, y_train, **fit_params)
  File "C:\Users\harsha.teja\Anaconda3\envs\NLP\lib\site-packages\sklearn\linear_model\_
logistic.py", line 1304, in fit
    solver = _check_solver(self.solver, self.penalty, self.dual)
  File "C:\Users\harsha.teja\Anaconda3\envs\NLP\lib\site-packages\sklearn\linear_model\_
logistic.py", line 439, in _check_solver
    " got %s." % (all_penalties, penalty))
ValueError: Logistic Regression supports only penalties in ['l1', 'l2', 'elasticnet', 'n
one'], got L1.

  FitFailedWarning)
C:\Users\harsha.teja\Anaconda3\envs\NLP\lib\site-packages\sklearn\model_selection\_valid
ation.py:552: FitFailedWarning: Estimator fit failed. The score on this train-test parti
tion for these parameters will be set to nan. Details:
Traceback (most recent call last):
  File "C:\Users\harsha.teja\Anaconda3\envs\NLP\lib\site-packages\sklearn\model_selectio
n\_validation.py", line 531, in _fit_and_score
    estimator.fit(X_train, y_train, **fit_params)
  File "C:\Users\harsha.teja\Anaconda3\envs\NLP\lib\site-packages\sklearn\linear_model\_
logistic.py", line 1304, in fit
    solver = _check_solver(self.solver, self.penalty, self.dual)
  File "C:\Users\harsha.teja\Anaconda3\envs\NLP\lib\site-packages\sklearn\linear_model\_
logistic.py", line 439, in _check_solver
    " got %s." % (all_penalties, penalty))
ValueError: Logistic Regression supports only penalties in ['l1', 'l2', 'elasticnet', 'n
one'], got L1.
```

```
  FitFailedWarning)
C:\Users\harsha.teja\Anaconda3\envs\NLP\lib\site-packages\sklearn\model_selection\_valid
ation.py:552: FitFailedWarning: Estimator fit failed. The score on this train-test parti
tion for these parameters will be set to nan. Details:
Traceback (most recent call last):
  File "C:\Users\harsha.teja\Anaconda3\envs\NLP\lib\site-packages\sklearn\model_selectio
n\_validation.py", line 531, in _fit_and_score
    estimator.fit(X_train, y_train, **fit_params)
  File "C:\Users\harsha.teja\Anaconda3\envs\NLP\lib\site-packages\sklearn\linear_model\_
logistic.py", line 1304, in fit
    solver = _check_solver(self.solver, self.penalty, self.dual)
  File "C:\Users\harsha.teja\Anaconda3\envs\NLP\lib\site-packages\sklearn\linear_model\_
logistic.py", line 439, in _check_solver
    " got %s." % (all_penalties, penalty))
ValueError: Logistic Regression supports only penalties in ['l1', 'l2', 'elasticnet', 'n
one'], got L1.

  FitFailedWarning)
C:\Users\harsha.teja\Anaconda3\envs\NLP\lib\site-packages\sklearn\model_selection\_valid
ation.py:552: FitFailedWarning: Estimator fit failed. The score on this train-test parti
tion for these parameters will be set to nan. Details:
Traceback (most recent call last):
  File "C:\Users\harsha.teja\Anaconda3\envs\NLP\lib\site-packages\sklearn\model_selectio
n\_validation.py", line 531, in _fit_and_score
    estimator.fit(X_train, y_train, **fit_params)
  File "C:\Users\harsha.teja\Anaconda3\envs\NLP\lib\site-packages\sklearn\linear_model\_
logistic.py", line 1304, in fit
    solver = _check_solver(self.solver, self.penalty, self.dual)
  File "C:\Users\harsha.teja\Anaconda3\envs\NLP\lib\site-packages\sklearn\linear_model\_
logistic.py", line 439, in _check_solver
    " got %s." % (all_penalties, penalty))
ValueError: Logistic Regression supports only penalties in ['l1', 'l2', 'elasticnet', 'n
one'], got L1.

  FitFailedWarning)
C:\Users\harsha.teja\Anaconda3\envs\NLP\lib\site-packages\sklearn\linear_model\_logisti
c.py:764: ConvergenceWarning: lbfgs failed to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression
  extra_warning_msg=_LOGISTIC_SOLVER_CONVERGENCE_MSG)
C:\Users\harsha.teja\Anaconda3\envs\NLP\lib\site-packages\sklearn\linear_model\_logisti
c.py:764: ConvergenceWarning: lbfgs failed to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression
  extra_warning_msg=_LOGISTIC_SOLVER_CONVERGENCE_MSG)
C:\Users\harsha.teja\Anaconda3\envs\NLP\lib\site-packages\sklearn\linear_model\_logisti
c.py:764: ConvergenceWarning: lbfgs failed to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression
  extra_warning_msg=_LOGISTIC_SOLVER_CONVERGENCE_MSG)
C:\Users\harsha.teja\Anaconda3\envs\NLP\lib\site-packages\sklearn\linear_model\_logisti
c.py:764: ConvergenceWarning: lbfgs failed to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.
```

```
Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression
  extra_warning_msg=_LOGISTIC_SOLVER_CONVERGENCE_MSG)
C:\Users\harsha.teja\Anaconda3\envs\NLP\lib\site-packages\sklearn\model_selection\_valid
ation.py:552: FitFailedWarning: Estimator fit failed. The score on this train-test parti
tion for these parameters will be set to nan. Details:
Traceback (most recent call last):
  File "C:\Users\harsha.teja\Anaconda3\envs\NLP\lib\site-packages\sklearn\model_selectio
n\_validation.py", line 531, in _fit_and_score
    estimator.fit(X_train, y_train, **fit_params)
  File "C:\Users\harsha.teja\Anaconda3\envs\NLP\lib\site-packages\sklearn\linear_model\_
logistic.py", line 1304, in fit
    solver = _check_solver(self.solver, self.penalty, self.dual)
  File "C:\Users\harsha.teja\Anaconda3\envs\NLP\lib\site-packages\sklearn\linear_model\_
logistic.py", line 439, in _check_solver
    " got %s." % (all_penalties, penalty))
ValueError: Logistic Regression supports only penalties in ['l1', 'l2', 'elasticnet', 'n
one'], got L1.

  FitFailedWarning)
C:\Users\harsha.teja\Anaconda3\envs\NLP\lib\site-packages\sklearn\linear_model\_logisti
c.py:764: ConvergenceWarning: lbfgs failed to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression
  extra_warning_msg=_LOGISTIC_SOLVER_CONVERGENCE_MSG)
C:\Users\harsha.teja\Anaconda3\envs\NLP\lib\site-packages\sklearn\linear_model\_logisti
c.py:764: ConvergenceWarning: lbfgs failed to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression
  extra_warning_msg=_LOGISTIC_SOLVER_CONVERGENCE_MSG)
C:\Users\harsha.teja\Anaconda3\envs\NLP\lib\site-packages\sklearn\linear_model\_logisti
c.py:764: ConvergenceWarning: lbfgs failed to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression
  extra_warning_msg=_LOGISTIC_SOLVER_CONVERGENCE_MSG)
C:\Users\harsha.teja\Anaconda3\envs\NLP\lib\site-packages\sklearn\linear_model\_logisti
c.py:764: ConvergenceWarning: lbfgs failed to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression
  extra_warning_msg=_LOGISTIC_SOLVER_CONVERGENCE_MSG)
C:\Users\harsha.teja\Anaconda3\envs\NLP\lib\site-packages\sklearn\model_selection\_valid
ation.py:552: FitFailedWarning: Estimator fit failed. The score on this train-test parti
tion for these parameters will be set to nan. Details:
Traceback (most recent call last):
  File "C:\Users\harsha.teja\Anaconda3\envs\NLP\lib\site-packages\sklearn\model_selectio
n\_validation.py", line 531, in _fit_and_score
```

```
    estimator.fit(X_train, y_train, **fit_params)
  File "C:\Users\harsha.teja\Anaconda3\envs\NLP\lib\site-packages\sklearn\linear_model\_
logistic.py", line 1304, in fit
    solver = _check_solver(self.solver, self.penalty, self.dual)
  File "C:\Users\harsha.teja\Anaconda3\envs\NLP\lib\site-packages\sklearn\linear_model\_
logistic.py", line 439, in _check_solver
    " got %s." % (all_penalties, penalty))
ValueError: Logistic Regression supports only penalties in ['l1', 'l2', 'elasticnet', 'n
one'], got L1.

  FitFailedWarning)
C:\Users\harsha.teja\Anaconda3\envs\NLP\lib\site-packages\sklearn\linear_model\_logisti
c.py:764: ConvergenceWarning: lbfgs failed to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression
  extra_warning_msg=_LOGISTIC_SOLVER_CONVERGENCE_MSG)
C:\Users\harsha.teja\Anaconda3\envs\NLP\lib\site-packages\sklearn\linear_model\_logisti
c.py:764: ConvergenceWarning: lbfgs failed to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression
  extra_warning_msg=_LOGISTIC_SOLVER_CONVERGENCE_MSG)
C:\Users\harsha.teja\Anaconda3\envs\NLP\lib\site-packages\sklearn\linear_model\_logisti
c.py:764: ConvergenceWarning: lbfgs failed to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression
  extra_warning_msg=_LOGISTIC_SOLVER_CONVERGENCE_MSG)
C:\Users\harsha.teja\Anaconda3\envs\NLP\lib\site-packages\sklearn\linear_model\_logisti
c.py:764: ConvergenceWarning: lbfgs failed to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression
  extra_warning_msg=_LOGISTIC_SOLVER_CONVERGENCE_MSG)
C:\Users\harsha.teja\Anaconda3\envs\NLP\lib\site-packages\sklearn\linear_model\_logisti
c.py:764: ConvergenceWarning: lbfgs failed to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression
  extra_warning_msg=_LOGISTIC_SOLVER_CONVERGENCE_MSG)
```

Out[257…

```
GridSearchCV(cv=StratifiedKFold(n_splits=4, random_state=None, shuffle=False),
             estimator=LogisticRegression(),
             param_grid={'C': array([1.e-03, 1.e-02, 1.e-01, 1.e+00, 1.e+01, 1.e+02, 1.e
+03]),
                         'penalty': ['L1', 'l2']},
             scoring='accuracy')
```

In [258…

```
grid_acc2 = grid_acc.predict(X_test)
```

In [259…
```
accuracy_score(y_test,grid_acc2)*100
```

Out[259… 97.35015772870662

In [260…
```
grid_acc.best_params_
```

Out[260… {'C': 1000.0, 'penalty': 'l2'}

In [261…
```
grid_acc.best_score_
```

Out[261… 0.9673238696109359

In [262…
```
grid_acc.best_estimator_
```

Out[262… LogisticRegression(C=1000.0)

In [265…
```
# evaluate pipeline
cv = RepeatedStratifiedKFold(n_splits=10, n_repeats=3, random_state=1)
scores = cross_val_score(lr, X_train, y_train, scoring='f1_micro', cv=cv, n_jobs=-1)
score = mean(scores)
print('F1 Score: %.3f' % score)
```

```
F1 Score: 0.776
```

In [266…
```
# evaluate pipeline
cv = RepeatedStratifiedKFold(n_splits=10, n_repeats=3, random_state=1)
scores = cross_val_score(lr, X_train, y_train, scoring='recall', cv=cv, n_jobs=-1)
score = mean(scores)
print('recall: %.3f' % score)
```

```
recall: 0.697
```

In [267…
```
print(classification_report(y_test,grid_acc2 ))
```

```
              precision    recall  f1-score   support

           0       1.00      0.95      0.97      4711
           1       0.95      1.00      0.97      4799

    accuracy                           0.97      9510
   macro avg       0.97      0.97      0.97      9510
weighted avg       0.97      0.97      0.97      9510
```

BY HARSHA TEJA BOLLA