AI BASED SOLUTION FOR FLAGGING OF FALSE INFORMATION ON ONLINE PLATFORMS

In [1]:

```
import pandas as pd
import numpy as np
data = pd.read_csv("E:/file2/Desktop/new_newsdesk.csv")
```

In [2]:

```
data = data.dropna(how = 'any', axis = 0)
```

In [3]:

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

Out[3]:

```
label    0
text     0
dtype: int64
```

In [4]:

```
data.label.value_counts()
```

Out[4]:

```
FAKE    1871
REAL    1850
Name: label, dtype: int64
```

In [5]:

```
from nltk.stem.porter import PorterStemmer
import re
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.preprocessing import LabelEncoder
from nltk.stem import WordNetLemmatizer
```

In [6]:

```
from nltk.corpus import stopwords
import nltk
```

In [7]:

```
stemming = PorterStemmer()
lemmatizer = WordNetLemmatizer()
```

In [8]:

```
from sklearn.model_selection import train_test_split,RandomizedSearchCV,KFold
```

In [9]:

```
X=data[['text']]
Y=data['label']
```

In [10]:

```
X
```

Out[10]:

	text
0	Payal has accused filmmaker Anurag Kashyap of ...
1	A four-minute-long video of a woman criticisin...
2	Republic Poll, a fake Twitter account imitatin...
3	Delhi teen finds place on UN green list, turns...
4	Delhi: A high-level meeting underway at reside...
...	...
3724	19:17 (IST) Sep 20\n\nThe second round of coun...
3725	19:17 (IST) Sep 20\n\nThe second round of coun...
3726	The Bengaluru City Police’s official Twitter h...
3727	Sep 20, 2020, 08:00AM IST\n\nSource: TOI.in\n\...
3728	Read Also\n\nRead Also\n\nAdvocate Ishkaran Bh...

3721 rows × 1 columns

In [11]:

```
p=data['text']
print(p)
```

```
0       Payal has accused filmmaker Anurag Kashyap of ...
1       A four-minute-long video of a woman criticisin...
2       Republic Poll, a fake Twitter account imitatin...
3       Delhi teen finds place on UN green list, turns...
4       Delhi: A high-level meeting underway at reside...
                              ...
3724    19:17 (IST) Sep 20\n\nThe second round of coun...
3725    19:17 (IST) Sep 20\n\nThe second round of coun...
3726    The Bengaluru City Police’s official Twitter h...
3727    Sep 20, 2020, 08:00AM IST\n\nSource: TOI.in\n\...
3728    Read Also\n\nRead Also\n\nAdvocate Ishkaran Bh...
Name: text, Length: 3721, dtype: object
```

In [12]:

```
x_train,x_test,y_train,y_test = train_test_split(X,Y,test_size=0.2,random_state=42)
```

In [13]:

```
print('x_train:',x_train.shape)
print('y_train:',y_train.shape)
print('x_test:',x_test.shape)
print('y_test:',y_test.shape)
```

```
x_train: (2976, 1)
y_train: (2976,)
x_test: (745, 1)
y_test: (745,)
```

In [14]:

```
X_train = x_train
```

In [15]:

```
x_train.head()
```

Out[15]:

	text
209	Several mainstream news outlets such as the Ti...
3374	NEW DELHI: The Drugs Controller General of Ind...
3540	A screenshot purporting to be a newspaper clip...
2472	A viral video of a woman with infant traveling...
2510	Read Also\n\nBe it winning hearts or winning t...

In [16]:

```
X_test = x_test
```

In [17]:

```
y_train.head()
```

Out[17]:

```
209     FAKE
3374    REAL
3540    FAKE
2472    FAKE
2510    REAL
Name: label, dtype: object
```

In [18]:

```
X_train.head()
```

Out[18]:

	text
209	Several mainstream news outlets such as the Ti...
3374	NEW DELHI: The Drugs Controller General of Ind...
3540	A screenshot purporting to be a newspaper clip...
2472	A viral video of a woman with infant traveling...
2510	Read Also\n\nBe it winning hearts or winning t...

In [19]:

```
X_test.head(10)
```

Out[19]:

	text
908	NEW DELHI: A final decision on Pakistan's stat...
3454	NEW DELHI: Seven of the top 10 most valued dom...
1790	Kareena Kapoor Khan, who is all set to ring in...
1167	A photo purporting to show a television news g...
1605	A disturbing video of a woman being flogged by...
184	A graphic photo of a human skeleton found insi...
2960	Delhi: A high-level meeting underway at reside...
1067	Social media has been rife with reports of the...
2348	A disturbing CCTV footage showing a Tamil Nadu...
3417	A disturbing video of a mentally ill woman hec...

In [20]:

```
y_test
```

Out[20]:

```
908     REAL
3454    REAL
1790    REAL
1167    FAKE
1605    FAKE
        ...
1239    FAKE
2409    FAKE
1958    FAKE
2680    FAKE
955     FAKE
Name: label, Length: 745, dtype: object
```

Data Preprocessing

In [21]:

```
def preprocess(pro):
    process = re.sub('[^a-zA-Z]'," ",pro)
    lowe = process.lower()
    tokens = lowe.split()

    stop = [lemmatizer.lemmatize(i) for i in tokens if i not in stopwords.words('English'
    lemmas =pd.Series([ " ".join(stop),len(stop)])
    return lemmas
```

In [22]:

```
px_train = X_train['text'].apply(preprocess)
```

In [23]:

```
px_train.head()
```

Out[23]:

	0	1
209	several mainstream news outlet time india hind...	396
3374	new delhi drug controller general india approv...	257
3540	screenshot purporting newspaper clipping claim...	289
2472	viral video woman infant traveling precariousl...	355
2510	read also winning heart winning trophy easy bi...	123

In [24]:

```
type(px_train)
```

Out[24]:

```
pandas.core.frame.DataFrame
```

Test data preprocessing

In [25]:

```
px_test = X_test['text'].apply(preprocess)
```

In [26]:

```
px_test.head()
```

Out[26]:

	0	1
908	new delhi final decision pakistan status finan...	150
3454	new delhi seven top valued domestic company sa...	187
1790	kareena kapoor khan set ring birthday tomorrow...	105
1167	photo purporting show television news graphic ...	170
1605	disturbing video woman flogged law husband all...	180

In [27]:

```
px_test.columns = ['clean_text','text_length']
px_test.head()
```

Out[27]:

	clean_text	text_length
908	new delhi final decision pakistan status finan...	150
3454	new delhi seven top valued domestic company sa...	187
1790	kareena kapoor khan set ring birthday tomorrow...	105
1167	photo purporting show television news graphic ...	170
1605	disturbing video woman flogged law husband all...	180

In [28]:

```
px_train.columns = ['clean_text','text_length']
px_train.head()
```

Out[28]:

	clean_text	text_length
209	several mainstream news outlet time india hind...	396
3374	new delhi drug controller general india approv...	257
3540	screenshot purporting newspaper clipping claim...	289
2472	viral video woman infant traveling precariousl...	355
2510	read also winning heart winning trophy easy bi...	123

In [29]:

```
X_train = pd.concat([X_train,px_train],axis=1)
X_train.head()
```

Out[29]:

	text	clean_text	text_length
209	Several mainstream news outlets such as the Ti...	several mainstream news outlet time india hind...	396
3374	NEW DELHI: The Drugs Controller General of Ind...	new delhi drug controller general india approv...	257
3540	A screenshot purporting to be a newspaper clip...	screenshot purporting newspaper clipping claim...	289
2472	A viral video of a woman with infant traveling...	viral video woman infant traveling precariousl...	355
2510	Read Also\n\nBe it winning hearts or winning t...	read also winning heart winning trophy easy bi...	123

In [30]:

```
X_test = pd.concat([X_test,px_test],axis=1)
```

In [31]:

```
X_test.head()
```

Out[31]:

	text	clean_text	text_length
908	NEW DELHI: A final decision on Pakistan's stat...	new delhi final decision pakistan status finan...	150
3454	NEW DELHI: Seven of the top 10 most valued dom...	new delhi seven top valued domestic company sa...	187
1790	Kareena Kapoor Khan, who is all set to ring in...	kareena kapoor khan set ring birthday tomorrow...	105
1167	A photo purporting to show a television news g...	photo purporting show television news graphic ...	170
1605	A disturbing video of a woman being flogged by...	disturbing video woman flogged law husband all...	180

In [32]:

```
from wordcloud import WordCloud
```

In [33]:

```
y_train
```

Out[33]:

```
209     FAKE
3374    REAL
3540    FAKE
2472    FAKE
2510    REAL
        ...
1133    REAL
1297    REAL
863     FAKE
3515    REAL
3182    FAKE
Name: label, Length: 2976, dtype: object
```

In [34]:

```
y_test
```

Out[34]:

```
908     REAL
3454    REAL
1790    REAL
1167    FAKE
1605    FAKE
        ...
1239    FAKE
2409    FAKE
1958    FAKE
2680    FAKE
955     FAKE
Name: label, Length: 745, dtype: object
```

In [35]:

```
real_n = X_train.loc[y_train=='REAL', :]
real_n.head()
```

Out[35]:

	text	clean_text	text_length
3374	NEW DELHI: The Drugs Controller General of Ind...	new delhi drug controller general india approv...	257
2510	Read Also\n\nBe it winning hearts or winning t...	read also winning heart winning trophy easy bi...	123
599	WASHINGTON: Enter Journey's Crossing Church in...	washington enter journey crossing church washi...	215
1707	NEW DELHI: The finance ministry on Saturday in...	new delhi finance ministry saturday informed l...	266
3676	PANAJI: The second phase of reviving the cocon...	panaji second phase reviving coconut tree line...	98

In [36]:

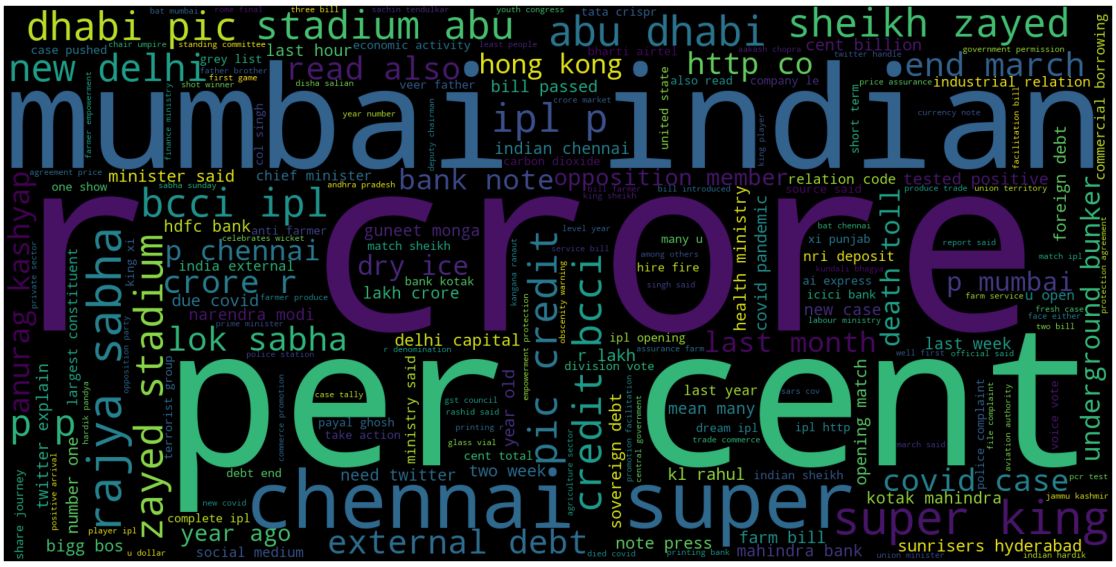
```
words = ' '.join(real_n['clean_text'])
clean_word = " ".join([word for word in words.split()])
```

In [37]:

```
real_word = WordCloud(stopwords=stopwords.words("english"),
                      background_color='black',
                      width=1600,
                      height=800).generate(clean_word)
```

In [38]:

```
plt.figure(1,figsize=(30,20))
plt.imshow(real_word)
plt.axis('off')
plt.show()
```



In [39]:

```
fake_n = X_train.loc[y_train=='FAKE', :]
fake_n.head()
```

Out[39]:

	text	clean_text	text_length
209	Several mainstream news outlets such as the Ti...	several mainstream news outlet time india hind...	396
3540	A screenshot purporting to be a newspaper clip...	screenshot purporting newspaper clipping claim...	289
2472	A viral video of a woman with infant traveling...	viral video woman infant traveling precariousl...	355
2704	A press release detailing restrictions imposed...	press release detailing restriction imposed mo...	233
1224	A video of a customer losing his cool at a bak...	video customer losing cool bakery manager kara...	238

In [40]:

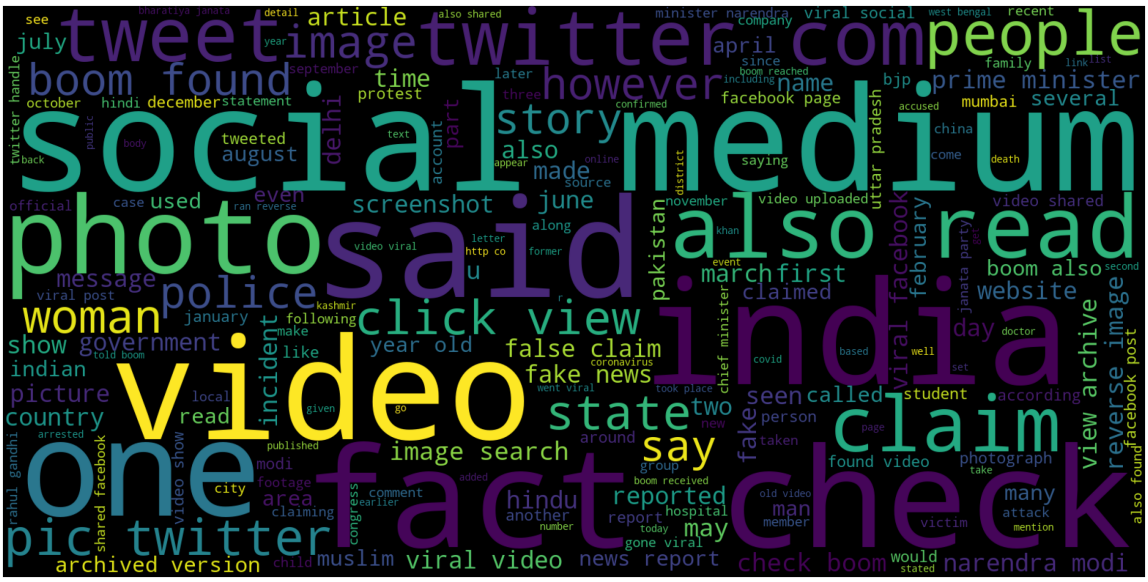
```
words_f = ' '.join(fake_n['clean_text'])
clean_word_f = " ".join([word for word in words_f.split()])
```

In [41]:

```
real_word_f = WordCloud(stopwords=stopwords.words("english"),
                        background_color='black',
                        width=1600,
                        height=800).generate(clean_word_f)
```

In [42]:

```
plt.figure(1,figsize=(30,20))
plt.imshow(real_word_f)
plt.axis('off')
plt.show()
```



Tfidf Vectorizer

In [43]:

```
from sklearn.feature_extraction.text import TfidfVectorizer
```

In [44]:

```
tf_vector = TfidfVectorizer()
```

In [45]:

```
X_train_t = tf_vector.fit_transform(X_train['clean_text'])
```

In [46]:

```
(X_train_t)
```

Out[46]:

```
<2976x28314 sparse matrix of type '<class 'numpy.float64'>'
        with 442232 stored elements in Compressed Sparse Row format>
```

In [47]:

```
print('unique words:',len(tf_vector.vocabulary_))
print('Shape of input data:',X_train_t.shape)
```

```
unique words: 28314
Shape of input data: (2976, 28314)
```

Test data

In [48]:

```
X_test_tf = tf_vector.transform(X_test['clean_text'])
```

In [49]:

```
X_test_tf
```

Out[49]:

```
<745x28314 sparse matrix of type '<class 'numpy.float64'>'
        with 107305 stored elements in Compressed Sparse Row format>
```

Label Encoding

In [50]:

```
label = LabelEncoder()
```

In [51]:

```
y_train = label.fit_transform(y_train)
```

In [52]:

```
y_train
```

Out[52]:

```
array([0, 1, 0, ..., 0, 1, 0])
```

In [53]:

```
Y_test = label.transform(y_test)
```

In [54]:

```
Y_test
```

Out[54]:

```
array([1, 1, 1, 0, 0, 0, 1, 0, 0, 0, 1, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 1,
       0, 1, 1, 1, 0, 1, 1, 0, 0, 1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 0, 1,
       0, 0, 1, 0, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 0, 1, 0, 1, 1, 0, 1,
       0, 1, 0, 1, 1, 0, 1, 1, 0, 0, 1, 0, 1, 0, 0, 0, 1, 1, 0, 1, 0, 0,
       0, 0, 0, 1, 1, 0, 0, 1, 1, 0, 0, 0, 1, 1, 0, 1, 0, 1, 1, 1, 0, 1,
       1, 0, 0, 1, 1, 0, 0, 0, 1, 0, 0, 1, 0, 1, 0, 1, 0, 1, 1, 1, 1, 0,
       1, 1, 1, 1, 1, 1, 1, 0, 0, 1, 1, 0, 1, 1, 0, 0, 0, 0, 1, 0, 1, 0,
       1, 0, 1, 1, 0, 0, 0, 0, 0, 1, 1, 0, 0, 1, 0, 1, 0, 1, 0, 0, 1, 0,
       1, 0, 1, 1, 0, 1, 1, 1, 1, 0, 0, 1, 0, 0, 0, 0, 1, 1, 1, 1, 0, 1,
       0, 0, 0, 1, 0, 1, 1, 0, 1, 0, 1, 0, 1, 0, 0, 1, 0, 0, 1, 0, 1, 1,
       1, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 1,
       1, 1, 0, 1, 0, 1, 1, 1, 0, 1, 1, 1, 0, 1, 1, 1, 1, 0, 0, 1, 1, 0,
       0, 1, 0, 1, 1, 0, 1, 0, 1, 1, 0, 1, 1, 1, 1, 0, 1, 0, 0, 0, 0, 1,
       0, 0, 1, 1, 0, 0, 1, 0, 1, 0, 0, 1, 0, 1, 0, 1, 1, 1, 1, 0, 1, 1,
       0, 0, 0, 1, 1, 0, 0, 0, 0, 1, 0, 1, 1, 0, 0, 1, 1, 0, 0, 1, 1, 0,
       1, 1, 1, 1, 0, 0, 0, 0, 1, 0, 1, 1, 0, 1, 0, 1, 1, 0, 0, 0, 1, 0,
       1, 0, 1, 0, 1, 1, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1,
       1, 0, 1, 0, 0, 0, 1, 1, 0, 1, 1, 0, 0, 0, 0, 1, 0, 1, 0, 0, 1, 1,
       0, 1, 1, 1, 1, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0,
       1, 0, 1, 0, 1, 1, 0, 1, 0, 0, 0, 0, 1, 1, 0, 1, 1, 1, 0, 0, 1, 0,
       1, 0, 0, 1, 1, 0, 1, 0, 0, 1, 0, 0, 1, 0, 0, 1, 1, 0, 0, 1, 0, 1,
       0, 1, 1, 0, 0, 0, 0, 1, 1, 0, 1, 1, 0, 0, 1, 0, 0, 1, 0, 1, 1, 0,
       0, 0, 0, 0, 1, 0, 1, 0, 1, 0, 1, 0, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1,
       1, 1, 0, 0, 1, 1, 0, 1, 0, 1, 0, 0, 1, 1, 0, 1, 0, 0, 0, 1, 1, 0,
       0, 0, 1, 0, 1, 0, 1, 0, 1, 0, 0, 0, 1, 1, 1, 1, 1, 1, 0, 0, 1, 1,
       0, 1, 0, 1, 1, 0, 1, 0, 1, 1, 0, 1, 0, 1, 1, 1, 1, 0, 0, 0, 1, 1,
       0, 1, 0, 1, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 0, 0, 1, 1, 1, 0, 1, 1,
       0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 0, 1, 1, 0, 0, 0, 0, 1,
       1, 0, 1, 1, 1, 1, 1, 1, 0, 0, 1, 1, 0, 0, 0, 0, 0, 1, 1, 1, 0, 0,
       0, 0, 1, 1, 1, 1, 0, 0, 0, 0, 0, 1, 1, 0, 1, 1, 1, 1, 1, 0, 1, 1,
       0, 0, 0, 1, 1, 0, 1, 0, 1, 1, 0, 0, 1, 0, 1, 1, 1, 1, 0, 1, 1, 0,
       0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 1, 0, 0, 1, 0, 1, 1, 0,
       0, 1, 1, 0, 1, 0, 1, 1, 1, 1, 1, 1, 0, 1, 1, 0, 0, 0, 1, 1, 1, 0,
       1, 1, 0, 0, 1, 1, 0, 1, 0, 1, 1, 0, 0, 1, 0, 0, 0, 0, 0])
```

Logistic Regression Model

In [55]:

```
from sklearn.linear_model import LogisticRegression
```

In [56]:

```
models = LogisticRegression()
```

In [57]:

```
max_iter = range(100, 500)
solver = ['lbfgs', 'newton-cg', 'liblinear']
warm_start = [True, False]
C = np.arange(0, 1, 0.01)
random_grid ={
    'max_iter' : max_iter,
    'warm_start' : warm_start,
    'solver' : solver,
    'C' : C,
}
```

In [58]:

```
kf1 = KFold(n_splits=5, shuffle=True)
```

In [59]:

```
#random_search = RandomizedSearchCV(models,parameter, n_iter=10, cv=kf1, n_jobs=-1)
```

In [60]:

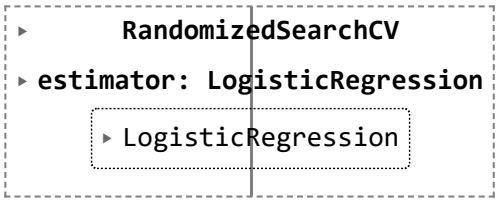
```
random_search = RandomizedSearchCV(estimator =models, param_distributions = random_grid,r
                                   n_jobs = -1,verbose = 1,random_state = 1,cv=kf1)
```

In [61]:

```
random_search.fit(X_train_t,y_train)
```

```
Fitting 5 folds for each of 10 candidates, totalling 50 fits
```

Out[61]:



In [62]:

```
n_model = LogisticRegression(warm_start = random_search.best_params_['warm_start'],solver
```

In [63]:

```
n_model.fit(X_train_t,y_train)
```

Out[63]:

```
LogisticRegression
LogisticRegression(C=0.9500000000000001, max_iter=368, solver='newton-c
g')
```

In [64]:

```
from sklearn.metrics import accuracy_score
```

In [65]:

```
new_logi =n_model.predict(X_train_t)
```

In [66]:

```
new_logi_train_accuracy = accuracy_score(new_logi,y_train)
```

In [67]:

```
print('accuracy_score',new_logi_train_accuracy)
```

```
accuracy_score 0.998991935483871
```

In [68]:

```
l_train_score = random_search.predict(X_train_t)
l_train_accuracy = accuracy_score(l_train_score,y_train)
```

In [69]:

```
print('train_accuracy:',l_train_accuracy)
```

```
train_accuracy: 0.998991935483871
```

In [70]:

```
l_test_score = random_search.predict(X_test_tf)
```

In [71]:

```
l_test_accuracy = accuracy_score(l_test_score,Y_test)
```

In [72]:

```
print('test_acccuracy:',l_test_accuracy)
```

```
test_acccuracy: 0.9919463087248322
```

In [73]:

```
from sklearn.metrics import confusion_matrix, accuracy_score, precision_score, recall_scc
```

In [74]:

```
confusion_matrix = metrics.confusion_matrix(Y_test, l_test_score)
print(confusion_matrix)
```

```
[[360   1]
 [  5 379]]
```

In [75]:

```
precision = precision_score(Y_test, l_test_score)
precision
```

Out[75]:

```
0.9973684210526316
```

In [76]:

```
recall = recall_score(Y_test, l_test_score)
recall
```

Out[76]:

```
0.9869791666666666
```

In [127]:

```
cmx_1=confusion_matrix(Y_test,l_test_score)
print("\nNo. of test samples : ",len(X_test))
print("\n Confustion Matrix : \n",cmx_1)
print("\nPerfomance measures are: \n",classification_report(Y_test, l_test_score))
```

```
No. of test samples :  745

 Confustion Matrix :
 [[360   1]
 [  5 379]]

Perfomance measures are:
               precision    recall  f1-score   support

           0       0.99      1.00      0.99       361
           1       1.00      0.99      0.99       384

    accuracy                           0.99       745
   macro avg       0.99      0.99      0.99       745
weighted avg       0.99      0.99      0.99       745
```

In [77]:

```
f1 = f1_score(Y_test, l_test_score)
print(f1)
```

```
0.9921465968586387
```

In [78]:

```
data1 = {'news':["ROME: Novak Djokovic knows it isnt model behavior when he loses his coc
```

In [79]:

```
data=pd.DataFrame(data1)
data
```

Out[79]:

	news
0	ROME: Novak Djokovic knows it isnt model behav...

In [80]:

```
print(data)
```

```
                                                news
0  ROME: Novak Djokovic knows it isnt model behav...
```

new prediction

In [81]:

```
news2 ={'news':['''Whether or not Christians should celebrate Halloween has been a contro
This is a time of year filled with debate, but not necessarily politics. Many Christians
Halloween is the holiday that links the seasons of fall and winter. Reportedly, it origin
Despite having at least partial roots from a Christian tradition, the relationship betwee
In the eighth century, Pope Gregory III dedicated November 1 as a time to honor all saint
Over time, Halloween advanced into a secular, community-based holiday branded by child-fr
In multiple countries around the world, as the days grow shorter and the nights get colde
Due to the efforts of community leaders and parents, Halloween has lost most of its illog
By Cherese Jackson (Virginia)Sources:History: Halloween
Kidsville News: Around the World â€“ October 2015
Grace to You: Christians and Halloween
Photo Credits:
Top Image Courtesy of Billy Wilson â€“ Flickr License
Inline Image (1) Courtesy of Richard Vignola â€“ Flickr License
Inline Image (2) Courtesy of The Forum News â€“ Flickr License
Featured Image Courtesy of John Nakamura â€“ Flickr License christianity , halloween''']}
```

In [82]:

```
data_news1 = pd.DataFrame(news2)
```

In [83]:

```
data_news1.head()
```

Out[83]:

	news
0	Whether or not Christians should celebrate Hal...

In [84]:

```
tnews1 = data_news1[['news']]
```

In [85]:

```
tnews1
```

Out[85]:

	news
0	Whether or not Christians should celebrate Hal...

In [86]:

```
def preprocess(pro):
    process = re.sub('[^a-zA-Z]'," ",pro)
    lowe = process.lower()
    tokens = lowe.split()

    stop = [lemmatizer.lemmatize(i) for i in tokens if i not in stopwords.words('English'
    lemmas =pd.Series([ " ".join(stop)])
    return lemmas
```

In [87]:

```
gnews1 = tnews1['news'].apply(preprocess)
```

In [88]:

```
gnews1.columns=['news']
gnews1
```

Out[88]:

	news
0	whether christian celebrate halloween controve...

In [89]:

```
newtf1 = tf_vector.transform(gnews1['news'])
```

In [90]:

```
print(newtf1)
```

```
  (0, 27990)    0.04745329965021367
  (0, 27989)    0.012496798739957453
  (0, 27760)    0.01663377025021477
  (0, 27724)    0.0585729509632132
  (0, 27708)    0.04584407427139434
  (0, 27627)    0.0727795518444617
  (0, 27609)    0.04888631673909161
  (0, 27574)    0.025441966164691302
  (0, 27573)    0.03138919900977684
  (0, 27523)    0.026730663660723183
  (0, 27471)    0.01709424383134715
  (0, 27424)    0.03823725627090886
  (0, 27394)    0.039350857406984144
  (0, 27043)    0.04888631673909161
  (0, 26936)    0.033309929658450914
  (0, 26713)    0.02174179495864285
  (0, 26712)    0.0327959504112792
  (0, 26459)    0.025525914439973724
  (0, 25831)    0.04745329965021367
  (0, 25758)    0.04131705138635638
  (0, 25722)    0.06176654404864191
  (0, 25625)    0.039983601433182504
  (0, 25536)    0.037938506981552066
  (0, 25533)    0.02072511642571237
  (0, 25379)    0.0332849011439085
  :     :
  (0, 3858)     0.020528108596219077
  (0, 3437)     0.04888631673909161
  (0, 3293)     0.05333767439204617
  (0, 3021)     0.05073379708776962
  (0, 3019)     0.02847700882299683
  (0, 2978)     0.0215115928349492
  (0, 2581)     0.04880907131910628
  (0, 2578)     0.05248040017313995
  (0, 2452)     0.020508736178478405
  (0, 1996)     0.026937841356822277
  (0, 1979)     0.027602774586765005
  (0, 1854)     0.037379724128905945
  (0, 1523)     0.03476975894522243
  (0, 1115)     0.041831081781860506
  (0, 1007)     0.10421617410977518
  (0, 1006)     0.03394065192814199
  (0, 898)      0.020060511172665663
  (0, 882)      0.031603283887623956
  (0, 881)      0.029977272092928286
  (0, 560)      0.027840651841111327
  (0, 458)      0.03363478690219996
  (0, 448)      0.08600388399451823
  (0, 321)      0.05315910388993769
  (0, 237)      0.04367856213053851
  (0, 80)       0.05073379708776962
```

In [91]:

```
import pickle
```

In [92]:

```
file = 'logisticm1.sav'
pickle.dump(models,open(file,'wb'))
```

In [93]:

```
model_2 = pickle.load(open('logisticm1.sav','rb'))
```

In [94]:

```
file2 = 'tfidf1.sav'
pickle.dump(tf_vector,open(file2,'wb'))
```

In [95]:

```
model_3 = pickle.load(open('tfidf1.sav','rb'))
```

In [98]:

```
ansr1 =n_model.predict(newtf1)
ansr1
```

Out[98]:

```
array([0])
```

In [100]:

```
confusion = metrics.confusion_matrix(Y_test, l_test_score)
```

SVM

In [128]:

```
from sklearn.svm import SVC
```

In [129]:

```
support = svm.SVC()
```

In [130]:

```
#C =[.01, .1, 1, 5, 10, 100]

#gamma= [0, .01, .1, 1, 5, 10, 100],
#kernel= ["rbf",'linear','poly']
#random_state=[1]
C = range(0, 10)
gamma = ['scale', 'auto']
svm_param = {
    "C":C ,
    "gamma":gamma

}
```

In [131]:

```
kf2 = KFold(n_splits=5, shuffle=True)
```

In [132]:

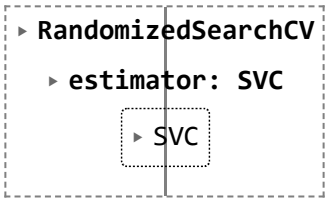
```
random_search1 = RandomizedSearchCV(estimator =support, param_distributions = svm_param,r
                                    n_jobs = -1,verbose = 1,random_state = 1,cv=kf2)
```

In [133]:

```
random_search1.fit(X_train_t,y_train)
```

```
Fitting 5 folds for each of 5 candidates, totalling 25 fits
```

Out[133]:



In [134]:

```
#best_params = random_search.best_estimator_.get_params()
#print(best_params)
print("Best hyperparameters: ", random_search1.best_params_)
```

```
Best hyperparameters:  {'gamma': 'scale', 'C': 8}
```

In [135]:

```
support_1 = svm.SVC(gamma=random_search1.best_params_['gamma'],C=random_search1.best_para
```

In [136]:

```
support_1.fit(X_train_t,y_train)
```

Out[136]:

SVC
SVC(C=8)

In [137]:

```
train_score_svm2  = support_1.predict(X_train_t)
```

In [138]:

```
train_accuracy_svm2 = accuracy_score(train_score_svm2,y_train)
```

In [139]:

```
print('train_accuracy:',train_accuracy_svm2)
```

```
train_accuracy: 1.0
```

In [140]:

```
support
```

Out[140]:

SVC
SVC()

In [141]:

```
support.fit(X_train_t,y_train)
```

Out[141]:

SVC
SVC()

In [142]:

```
from sklearn.metrics import accuracy_score
```

In [143]:

```
train_score_1 = support.predict(X_train_t)
train_accuracy_1 = accuracy_score(train_score_1,y_train)
```

In [144]:

```
print('train_accuracy:',train_accuracy_1)
```

```
train_accuracy: 1.0
```

In [145]:

```
test_score_1 = support.predict(X_test_tf)
```

In [146]:

```
test_accuracy_1 = accuracy_score(test_score_1,Y_test)
```

In [147]:

```
print('test_acccuracy:',test_accuracy_1)
```

```
test_acccuracy: 0.9892617449664429
```

In [148]:

```
news_1=X_train_t[1]
```

In [152]:

```
prediction_1 = support.predict(news_1)
print(prediction_1)

if (prediction_1[0]==0):
    print('The news is real')
else:
    print('The news is fake')
```

```
[1]
The news is fake
```

In [111]:

```
from sklearn.metrics import classification_report, confusion_matrix
```

In [150]:

```
confusion = metrics.confusion_matrix(Y_test, test_score_1)
```

In [151]:

```
cmx=confusion_matrix(Y_test,test_score_1)
print("\nNo. of test samples : ",len(X_test))
print("\n Confustion Matrix : \n",cmx)
print("\nPerfomance measures are: \n",classification_report(Y_test, test_score_1))
```

```
No. of test samples :  745

 Confustion Matrix :
 [[361   0]
 [  8 376]]

Perfomance measures are:
               precision    recall  f1-score   support

           0       0.98      1.00      0.99       361
           1       1.00      0.98      0.99       384

    accuracy                           0.99       745
   macro avg       0.99      0.99      0.99       745
weighted avg       0.99      0.99      0.99       745
```

KNN

In [102]:

```
from sklearn.neighbors import KNeighborsClassifier
```

In [103]:

```
knn_model = KNeighborsClassifier(n_neighbors=5)
```

In [104]:

```
knn_model.fit(X_train_t,y_train)
```

Out[104]:

```
KNeighborsClassifier
KNeighborsClassifier()
```

In [105]:

```
knn_1_train_score = knn_model.predict(X_train_t)
knn_train_accuracy = accuracy_score(knn_1_train_score,y_train)
```

In [106]:

```
print('train_accuracy:',knn_train_accuracy)
```

```
train_accuracy: 0.9684139784946236
```

In [107]:

```
knn_test_score = knn_model.predict(X_test_tf)
```

In [108]:

```
knn_test_accuracy = accuracy_score(knn_test_score,Y_test)
```

In [109]:

```
print('test_acccuracy:',knn_test_accuracy)
```

```
test_acccuracy: 0.9409395973154362
```

In [112]:

```
cmx_2=confusion_matrix(Y_test,knn_test_score)
print("\nNo. of test samples : ",len(X_test))
print("\n Confustion Matrix : \n",cmx_2)
print("\nPerfomance measures are: \n",classification_report(Y_test, knn_test_score))
```

```
No. of test samples :  745

 Confustion Matrix :
 [[352   9]
 [ 35 349]]

Perfomance measures are:
              precision    recall  f1-score   support

           0       0.91      0.98      0.94       361
           1       0.97      0.91      0.94       384

    accuracy                           0.94       745
   macro avg       0.94      0.94      0.94       745
weighted avg       0.94      0.94      0.94       745
```

SBERT

In [113]:

```
! pip install -U sentence-transformers
```

```
Requirement already satisfied: sentence-transformers in d:\users\safuvan\a
naconda3\lib\site-packages (2.2.2)
Requirement already satisfied: torch>=1.6.0 in d:\users\safuvan\anaconda3
\lib\site-packages (from sentence-transformers) (2.0.0)
Requirement already satisfied: nltk in d:\users\safuvan\anaconda3\lib\site
-packages (from sentence-transformers) (3.7)
Requirement already satisfied: tqdm in d:\users\safuvan\anaconda3\lib\site
-packages (from sentence-transformers) (4.64.0)
Requirement already satisfied: scikit-learn in d:\users\safuvan\anaconda3
\lib\site-packages (from sentence-transformers) (1.1.1)
Requirement already satisfied: sentencepiece in d:\users\safuvan\anaconda3
\lib\site-packages (from sentence-transformers) (0.1.98)
Requirement already satisfied: huggingface-hub>=0.4.0 in d:\users\safuvan
\anaconda3\lib\site-packages (from sentence-transformers) (0.13.4)
Requirement already satisfied: transformers<5.0.0,>=4.6.0 in d:\users\safu
van\anaconda3\lib\site-packages (from sentence-transformers) (4.28.1)
Requirement already satisfied: numpy in d:\users\safuvan\anaconda3\lib\sit
e-packages (from sentence-transformers) (1.21.5)
Requirement already satisfied: torchvision in d:\users\safuvan\anaconda3\l
ib\site-packages (from sentence-transformers) (0.15.1)
Requirement already satisfied: scipy in d:\users\safuvan\anaconda3\lib\sit
e-packages (from sentence-transformers) (1.9.1)
Requirement already satisfied: typing-extensions>=3.7.4.3 in d:\users\safu
van\anaconda3\lib\site-packages (from huggingface-hub>=0.4.0->sentence-tra
nsformers) (4.3.0)
Requirement already satisfied: requests in d:\users\safuvan\anaconda3\lib
\site-packages (from huggingface-hub>=0.4.0->sentence-transformers) (2.28.
1)
Requirement already satisfied: pyyaml>=5.1 in d:\users\safuvan\anaconda3\l
ib\site-packages (from huggingface-hub>=0.4.0->sentence-transformers) (6.
0)
Requirement already satisfied: packaging>=20.9 in d:\users\safuvan\anacond
a3\lib\site-packages (from huggingface-hub>=0.4.0->sentence-transformers)
(21.3)
Requirement already satisfied: filelock in d:\users\safuvan\anaconda3\lib
\site-packages (from huggingface-hub>=0.4.0->sentence-transformers) (3.6.
0)
Requirement already satisfied: sympy in d:\users\safuvan\anaconda3\lib\sit
e-packages (from torch>=1.6.0->sentence-transformers) (1.10.1)
Requirement already satisfied: jinja2 in d:\users\safuvan\anaconda3\lib\si
te-packages (from torch>=1.6.0->sentence-transformers) (2.11.3)
Requirement already satisfied: networkx in d:\users\safuvan\anaconda3\lib
\site-packages (from torch>=1.6.0->sentence-transformers) (2.8.4)
Requirement already satisfied: colorama in d:\users\safuvan\anaconda3\lib
\site-packages (from tqdm->sentence-transformers) (0.4.5)
Requirement already satisfied: regex!=2019.12.17 in d:\users\safuvan\anaco
nda3\lib\site-packages (from transformers<5.0.0,>=4.6.0->sentence-transfor
mers) (2022.7.9)
Requirement already satisfied: tokenizers!=0.11.3,<0.14,>=0.11.1 in d:\use
rs\safuvan\anaconda3\lib\site-packages (from transformers<5.0.0,>=4.6.0->s
entence-transformers) (0.13.2)
Requirement already satisfied: click in d:\users\safuvan\anaconda3\lib\sit
e-packages (from nltk->sentence-transformers) (8.0.4)
Requirement already satisfied: joblib in d:\users\safuvan\anaconda3\lib\si
te-packages (from nltk->sentence-transformers) (1.2.0)
Requirement already satisfied: threadpoolctl>=2.0.0 in d:\users\safuvan\an
aconda3\lib\site-packages (from scikit-learn->sentence-transformers) (2.2.
0)
Requirement already satisfied: pillow!=8.3.*,>=5.3.0 in d:\users\safuvan\a
naconda3\lib\site-packages (from torchvision->sentence-transformers) (9.2.
0)
```

KNN-Model

In [114]:

```
from sentence_transformers import SentenceTransformer, util
```

In [115]:

```
model = SentenceTransformer('all-MiniLM-L6-v2')
```

In [116]:

```
X_train = model.encode(px_train['clean_text'].tolist())
```

In [117]:

```
x_test = model.encode(px_test['clean_text'].tolist())
```

In [118]:
```
knn_model.fit(X_train_b,y_train)
```

Out[118]:
```
KNeighborsClassifier()
```

In [119]:
```
knn_bert_train_score = knn_model.predict(X_train_b)
knn_train_accuracy_bert = accuracy_score(knn_bert_train_score,y_train)
```

In [120]:
```
knn_train_accuracy_bert
```

Out[120]:
```
0.9771505376344086
```

Logistic Regression - Model

In [123]:
```
n_model.fit(X_train_b,y_train)
```

Out[123]:
```
LogisticRegression(C=0.9500000000000001, max_iter=368, solver='newton-cg')
```

In [124]:
```
logi_bert_train_score = n_model.predict(X_train_b)
logi_train_accuracy_bert = accuracy_score(logi_bert_train_score,y_train)
```

In [125]:
```
logi_train_accuracy_bert
```

Out[125]:
```
0.9848790322580645
```

In [ ]:
```

```