In [3]:

```
import pandas as pd
```

In [4]:

```
#Read the data
df=pd.read_csv('D:/Fake-news/train/train.csv')
```

In [5]:

```
df.head()
```

Out[5]:

	id	title	author	text	label
0	0	House Dem Aide: We Didn’t Even See Comey’s Let...	Darrell Lucus	House Dem Aide: We Didn’t Even See Comey’s Let...	1
1	1	FLYNN: Hillary Clinton, Big Woman on Campus - ...	Daniel J. Flynn	Ever get the feeling your life circles the rou...	0
2	2	Why the Truth Might Get You Fired	Consortiumnews.com	Why the Truth Might Get You Fired October 29, ...	1
3	3	15 Civilians Killed In Single US Airstrike Hav...	Jessica Purkiss	Videos 15 Civilians Killed In Single US Airstr...	1
4	4	Iranian woman jailed for fictional unpublished...	Howard Portnoy	Print \nAn Iranian woman has been sentenced to...	1

In [6]:

```
## Get the Independent Features

X=df.drop('label',axis=1)
```

In [7]:

```
X.head()
```

Out[7]:

	id	title	author	text
0	0	House Dem Aide: We Didn’t Even See Comey’s Let...	Darrell Lucus	House Dem Aide: We Didn’t Even See Comey’s Let...
1	1	FLYNN: Hillary Clinton, Big Woman on Campus - ...	Daniel J. Flynn	Ever get the feeling your life circles the rou...
2	2	Why the Truth Might Get You Fired	Consortiumnews.com	Why the Truth Might Get You Fired October 29, ...
3	3	15 Civilians Killed In Single US Airstrike Hav...	Jessica Purkiss	Videos 15 Civilians Killed In Single US Airstr...
4	4	Iranian woman jailed for fictional unpublished...	Howard Portnoy	Print \nAn Iranian woman has been sentenced to...

In [8]:

```
## Get the Dependent features
y=df['label']
```

In [9]:

```
y.head()
```

Out[9]:

```
0    1
1    0
2    1
3    1
4    1
Name: label, dtype: int64
```

In [10]:

```
df.shape
```

Out[10]:

```
(20800, 5)
```

In [11]:

```
from sklearn.feature_extraction.text import CountVectorizer, TfidfVectorizer, HashingVectorizer
```

In [12]:

```
df=df.dropna()
```

In [13]:

```
df.shape
```

Out[13]:

```
(18285, 5)
```

In [14]:

```
df.head(10)
```

Out[14]:

	id	title	author	text	label
0	0	House Dem Aide: We Didn’t Even See Comey’s Let...	Darrell Lucus	House Dem Aide: We Didn’t Even See Comey’s Let...	1
1	1	FLYNN: Hillary Clinton, Big Woman on Campus - ...	Daniel J. Flynn	Ever get the feeling your life circles the rou...	0
2	2	Why the Truth Might Get You Fired	Consortiumnews.com	Why the Truth Might Get You Fired October 29, ...	1
3	3	15 Civilians Killed In Single US Airstrike Hav...	Jessica Purkiss	Videos 15 Civilians Killed In Single US Airstr...	1
4	4	Iranian woman jailed for fictional unpublished...	Howard Portnoy	Print \nAn Iranian woman has been sentenced to...	1
5	5	Jackie Mason: Hollywood Would Love Trump if He...	Daniel Nussbaum	In these trying times, Jackie Mason is the Voi...	0
7	7	Benoît Hamon Wins French Socialist Party’s Pre...	Alissa J. Rubin	PARIS — France chose an idealistic, traditi...	0
9	9	A Back-Channel Plan for Ukraine and Russia, Co...	Megan Twohey and Scott Shane	A week before Michael T. Flynn resigned as nat...	0
10	10	Obama’s Organizing for Action Partners with So...	Aaron Klein	Organizing for Action, the activist group that...	0
11	11	BBC Comedy Sketch "Real Housewives of ISIS" Ca...	Chris Tomlinson	The BBC produced spoof on the “Real Housewives...	0

In [15]:

```
messages=df.copy()
```

In [16]:

```
messages.reset_index(inplace=True)
```

In [17]:

```
messages.head(10)
```

Out[17]:

	index	id	title	author	text	label
0	0	0	House Dem Aide: We Didn’t Even See Comey’s Let...	Darrell Lucus	House Dem Aide: We Didn’t Even See Comey’s Let...	1
1	1	1	FLYNN: Hillary Clinton, Big Woman on Campus - ...	Daniel J. Flynn	Ever get the feeling your life circles the rou...	0
2	2	2	Why the Truth Might Get You Fired	Consortiumnews.com	Why the Truth Might Get You Fired October 29, ...	1
3	3	3	15 Civilians Killed In Single US Airstrike Hav...	Jessica Purkiss	Videos 15 Civilians Killed In Single US Airstr...	1
4	4	4	Iranian woman jailed for fictional unpublished...	Howard Portnoy	Print \nAn Iranian woman has been sentenced to...	1
5	5	5	Jackie Mason: Hollywood Would Love Trump if He...	Daniel Nussbaum	In these trying times, Jackie Mason is the Voi...	0
6	7	7	Benoît Hamon Wins French Socialist Party’s Pre...	Alissa J. Rubin	PARIS — France chose an idealistic, traditi...	0
7	9	9	A Back-Channel Plan for Ukraine and Russia, Co...	Megan Twohey and Scott Shane	A week before Michael T. Flynn resigned as nat...	0
8	10	10	Obama’s Organizing for Action Partners with So...	Aaron Klein	Organizing for Action, the activist group that...	0
9	11	11	BBC Comedy Sketch "Real Housewives of ISIS" Ca...	Chris Tomlinson	The BBC produced spoof on the “Real Housewives...	0

In [18]:

```
messages['title'][6]
```

Out[18]:

```
'Benoît Hamon Wins French Socialist Party’s Presidential Nomination - The New York Times'
```

In [25]:

```
from nltk.corpus import stopwords
from nltk.stem.porter import PorterStemmer
import re
ps = PorterStemmer()
corpus = []
for i in range(0, len(messages)):
    review = re.sub('[^a-zA-Z]', ' ', messages['title'][i])
    review = review.lower()
    review = review.split()

    review = [ps.stem(word) for word in review if not word in stopwords.words('english')]
    review = ' '.join(review)
    corpus.append(review)
```

In [26]:

```
corpus[3]
```

Out[26]:

```
'civilian kill singl us airstrik identifi'
```

In [28]:

```
## TFidf Vectorizer
from sklearn.feature_extraction.text import TfidfVectorizer
tfidf_v=TfidfVectorizer(max_features=5000,ngram_range=(1,3))
X=tfidf_v.fit_transform(corpus).toarray()
```

In [29]:

```
X.shape
```

Out[29]:

```
(18285, 5000)
```

In [30]:

```
y=messages['label']
```

In [31]:

```
## Divide the dataset into Train and Test
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.33, random_state=0)
```

In [32]:

```
tfidf_v.get_feature_names()[:20]
```

Out[32]:

```
['abandon',
 'abc',
 'abc news',
 'abduct',
 'abe',
 'abedin',
 'abl',
 'abort',
 'abroad',
 'absolut',
 'abstain',
 'absurd',
 'abus',
 'abus new',
 'abus new york',
 'academi',
 'accept',
 'access',
 'access pipelin',
 'access pipelin protest']
```

In [33]:

```
tfidf_v.get_params()
```

Out[33]:

```
{'analyzer': 'word',
 'binary': False,
 'decode_error': 'strict',
 'dtype': numpy.float64,
 'encoding': 'utf-8',
 'input': 'content',
 'lowercase': True,
 'max_df': 1.0,
 'max_features': 5000,
 'min_df': 1,
 'ngram_range': (1, 3),
 'norm': 'l2',
 'preprocessor': None,
 'smooth_idf': True,
 'stop_words': None,
 'strip_accents': None,
 'sublinear_tf': False,
 'token_pattern': '(?u)\\b\\w\\w+\\b',
 'tokenizer': None,
 'use_idf': True,
 'vocabulary': None}
```

In [34]:

```
count_df = pd.DataFrame(X_train, columns=tfidf_v.get_feature_names())
```

In [35]:

```
count_df.head()
```

Out[35]:

	abandon	abc	abc news	abduct	abe	abedin	abl	abort	abroad	absolut	...	zero	zika	zika viru	zionist	zone	zone new	zone new york	zoo	zu
0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.000000	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.000000	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.000000	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.000000	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.305244	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

5 rows × 5000 columns

In [36]:

```
import matplotlib.pyplot as plt
```

In [37]:

```
def plot_confusion_matrix(cm, classes,
                          normalize=False,
                          title='Confusion matrix',
                          cmap=plt.cm.Blues):
    plt.imshow(cm, interpolation='nearest', cmap=cmap)
    plt.title(title)
    plt.colorbar()
    tick_marks = np.arange(len(classes))
    plt.xticks(tick_marks, classes, rotation=45)
    plt.yticks(tick_marks, classes)

    if normalize:
        cm = cm.astype('float') / cm.sum(axis=1)[:, np.newaxis]
        print("Normalized confusion matrix")
    else:
        print('Confusion matrix, without normalization')

    thresh = cm.max() / 2.
    for i, j in itertools.product(range(cm.shape[0]), range(cm.shape[1])):
        plt.text(j, i, cm[i, j],
                 horizontalalignment="center",
                 color="white" if cm[i, j] > thresh else "black")

    plt.tight_layout()
    plt.ylabel('True label')
    plt.xlabel('Predicted label')
```

MultinomialNB Algorithm

In [38]:

```
from sklearn.naive_bayes import MultinomialNB
classifier=MultinomialNB()
```

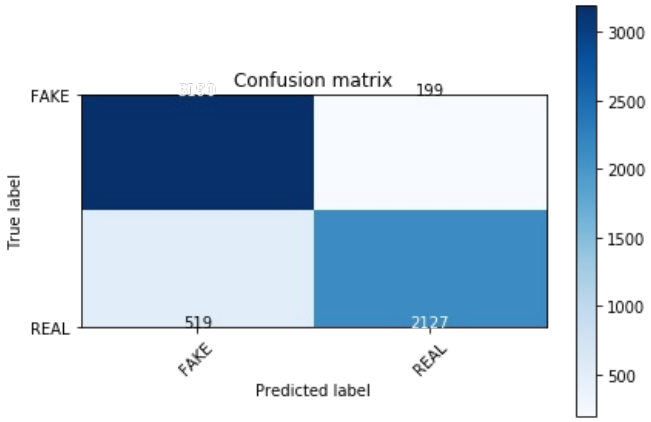
In [39]:

```
from sklearn import metrics
import numpy as np
import itertools
```

In [40]:

```
classifier.fit(X_train, y_train)
pred = classifier.predict(X_test)
score = metrics.accuracy_score(y_test, pred)
print("accuracy:   %0.3f" % score)
cm = metrics.confusion_matrix(y_test, pred)
plot_confusion_matrix(cm, classes=['FAKE', 'REAL'])
```

```
accuracy:   0.881
Confusion matrix, without normalization
```



In [41]:

```
classifier.fit(X_train, y_train)
pred = classifier.predict(X_test)
score = metrics.accuracy_score(y_test, pred)
score
```

Out[41]:

```
0.8810273405136703
```

In [42]:

```
y_train.shape
```

Out[42]:

```
(12250,)
```

In [45]:

```
from sklearn.linear_model import LogisticRegression#logistic Regression
```

In [57]:

```
logreg = LogisticRegression(C=1e5)
logreg.fit(X_train,y_train)
pred=logreg.predict(X_test)
print('Accuracy of Lasso classifier on training set: {:2f}'
     .format(logreg.score(X_train, y_train)))
print('Accuracy of Lasso classsifier on test set: {:.2f}'
     .format(logreg.score(X_test, y_test)))
cm = metrics.confusion_matrix(y_test, pred)
cm
```

```
C:\Users\sainath\Anaconda3\lib\site-packages\sklearn\linear_model\logistic.py:432: FutureWarning: De
fault solver will be changed to 'lbfgs' in 0.22. Specify a solver to silence this warning.
  FutureWarning)

Accuracy of Lasso classifier on training set: 1.000000
Accuracy of Lasso classsifier on test set: 0.88
```

Out[57]:

```
array([[1814,  247],
       [ 209, 1387]], dtype=int64)
```

Decision Tree

In [51]:

```
from sklearn.tree import DecisionTreeClassifier
classifier = DecisionTreeClassifier()
classifier.fit(X_train, y_train)
```

Out[51]:

```
DecisionTreeClassifier(class_weight=None, criterion='gini', max_depth=None,
                       max_features=None, max_leaf_nodes=None,
                       min_impurity_decrease=0.0, min_impurity_split=None,
                       min_samples_leaf=1, min_samples_split=2,
                       min_weight_fraction_leaf=0.0, presort=False,
                       random_state=None, splitter='best')
```

In [53]:

```
y_pred = classifier.predict(X_test)
```

In [54]:

```
from sklearn.metrics import classification_report, confusion_matrix
print(confusion_matrix(y_test, y_pred))
print(classification_report(y_test, y_pred))
```

```
[[1881  180]
 [ 174 1422]]
              precision    recall  f1-score   support

           0       0.92      0.91      0.91      2061
           1       0.89      0.89      0.89      1596

    accuracy                           0.90      3657
   macro avg       0.90      0.90      0.90      3657
weighted avg       0.90      0.90      0.90      3657
```

Randomforest

In [55]:

```
# Feature Scaling
from sklearn.preprocessing import StandardScaler

sc = StandardScaler()
X_train = sc.fit_transform(X_train)
X_test = sc.transform(X_test)
```

im