RAMAIAH
Institute of Technology

INTERNSHIP PROJECT REPORT
FAKE NEWS DETECTION

Submitted in partial fulfillment of requirements for the degree of Bachelor of Engineering in Information Science

Submitted by:
Aston Glen Noronha
1MS18IS021

Under the guidance of
Shri. Siddhartha Banerjee, Sc. 'F'
Centre for Artificial Intelligence & Robotics (CAIR)
DRDO Complex, C.V. Raman Nagar, Bangalore-560 093

Department of Information Science
M.S. Ramaiah Institute of Technology

2021-2022

ACKNOWLEDGEMENT

I would like to take this opportunity to extend my gratitude to everyone who has helped me in the completion of this report.

I would like to thank my mentor, Mr. Siddhartha Banerjee, an outstanding scientist from CAIR-DRDO, for mentoring me throughout the completion of this project and guiding me with valuable information.

I am certain this knowledge that I have come to gain will surely help me in advancing my career goals and lastly, I would like to thank my friends and family for helping me finish and submit all assignments and this report on time.

Place: Bangalore
Date: 25/03/2022

TABLE OF CONTENTS

1. ABSTRACT

The fake news on social media and various other media is wide spreading and is a matter of serious concern due to its ability to cause a lot of social and national damage with destructive impacts. A lot of research is already focused on detecting it. This paper makes an analysis of the research related to fake news detection and explores the traditional machine learning models to choose the best, in order to create a model of a product with supervised machine learning algorithm, that can classify fake news as true or false, by using tools like python scikit-learn, NLP for textual analysis. This process will result in feature extraction and vectorization; we propose using Python scikit-learn library to perform tokenization and feature extraction of text data, because this library contains useful tools like Count Vectorizer and Tiff Vectorizer. Then, we will perform feature selection methods, to experiment and choose the best fit features to obtain the highest precision, according to confusion matrix results.

2. INTRODUCTION

The introduction of the World Wide Web and the quick adoption of social media platforms (such as Facebook and Twitter) prepared the door for unprecedented levels of knowledge distribution in human history. Among other things, news organizations benefited from the widespread usage of social media platforms by providing subscribers with updated news in near real time. Newspapers, tabloids, and magazines have given way to digital news platforms, blogs, social media feeds, and other digital media formats. Consumers now have access to the most up-to-date information at their fingertips. 70% of traffic to news websites comes from Facebook referrals. These social media platforms are highly strong and beneficial in their current state since they allow users to discuss and exchange ideas as well as debate problems such as democracy, education, and health. However, certain individuals exploit such platforms for undesirable purposes, such as establishing skewed ideas, influencing mindsets, and disseminating satire or ridiculousness.

In the recent decade, the propagation of fake news has accelerated dramatically. Such a boom of posting items online that do not match to reality has resulted in a slew of issues, not just in politics, but in sports, health, and science as well. The

financial markets are one area affected by fake news, where a rumor can have severe implications and even bring the market to a halt.

Our ability to make decisions is largely determined by the information we absorb. Our worldview is influenced by the information we consume. There is mounting evidence that people have reacted irrationally to news that afterwards turned out to be false. One recent example is the propagation of the new corona virus, which saw bogus stories regarding the virus's origin, nature, and behavior spread throughout the Internet. As more individuals learned about the bogus content online, the situation escalated. Finding such news on the internet is a difficult effort.

Fortunately, a number of computational algorithms exist that can be used to identify bogus articles based on their linguistic content. Fact-checking websites like "PolitiFact" and "Snopes" are used in the majority of these tactics. Researchers maintain a variety of archives that provide listings of websites that have been identified as unclear or false. The difficulty with these services is that they require human knowledge to recognize bogus articles/websites. More crucially, fact-checking websites only feature stories from specific domains, such as politics, and are not designed to detect false news from a variety of domains, such as entertainment,

sports, or technology.

Data in many types, such as documents, films, and audios, can be found on the World Wide Web. It's challenging to discover and classify news released online in an unstructured style because it necessitates human skill. Computational techniques such as natural language processing (NLP) can, however, be used to discover anomalies that distinguish a deceptive text piece from one that is based on facts. Other strategies include examining the spread of fake news in comparison to true news. More specifically, the method examines how a fake news article differs from a genuine piece in terms of how it spreads over a network. The response to an article can be theoretically separated to identify the article as real or phoney.

3. LITERATURE SURVEY

3.1 Related Work

3.1.1 *Social Media and Fake News*

Social media includes websites and programs that are devoted to forums, social websites, microblogging, social bookmarking and wikis [1][2]. On the other side, some researchers consider the fake news as a result of accidental issues such as educational shock or unwitting actions like what happened in Nepal Earthquake case [3][4]. In 2020, there was widespread fake news concerning health that had exposed global health at risk. The WHO released a warning during early February 2020 that the COVID-19 outbreak has caused massive 'infodemic', or a spurt of real and fake news—which included lots of misinformation.

3.1.2 *Natural Language Processing*

The main reason for utilizing Natural Language Processing is to consider one or more specializations of system or an algorithm. The Natural Language Processing (NLP) rating of an algorithmic system enables the combination of speech understanding and speech generation. In addition, it could be utilized to detect actions with various languages.[5] suggested a new ideal system for extraction actions from languages of English, Italian and Dutch

speeches through utilizing various pipelines of various languages such as Emotion Analyzer and Detection, Named Entity Recognition (NER), Parts of Speech (POS) Taggers and so on.

3.1.3 *Data Mining*

Data mining techniques are categorized into two main methods, which is; supervised and unsupervised. The supervised method utilizes the training information in order to foresee the hidden activities. Unsupervised Data Mining is a try to recognize hidden data models provided without providing training data for example, pairs of input labels and categories.

3.1.4 *Machine Learning Classification*

Machine Learning (ML) is a class of algorithms that help software systems achieve more accurate results without having to reprogram them directly. Data scientists characterize changes or characteristics that the model needs to analyze and utilize to develop predictions. When the training is completed, the algorithm splits the learned levels into new data [6]. There are six algorithms that are adopted in this paper for classifying the fake news.

3.1.5 *Naïve Bayes*

This algorithm works on Bayes theory under the assuming that its free from predictors and is used in multiple machine learning problems [7]. Simply put, Naive Bayes assumes that one function in the category has nothing to do with another. For example, the fruit will be classified as an apple when its of red color, swirls, and the diameter is close to 3 inches. Regardless of whether these functions depend on each other or on different functions, and even if these functions depend on each other or on other functions, Naive Bayes assumes that all these functions share a separate proof of the apples.

Naive Bayes Equation:

$$P(c \mid x) = \frac{P(x \mid c)P(c)}{P(x)}$$

3.2 Related Work on Fake News Detection

MykhailoGranik and VolodymyrMesyura. [8] “Fake news detection using naive Bayes classifier.” - pointed out various sources of media and made the suitable studies whether the submitted article is reliable or fake. The paper utilizes models based on speech characteristics and predictive models that do not fit with the other current models.

Gilda, S. [9] “Evaluating machine learning algorithms for fake news detection.” - used naïve Bayes classifier to detect fake news by Naive Bayes. This method was performed as a software framework and experimented it with various records from the Facebook, etc., resulting in an accuracy of 74%. The paper neglected the punctuation errors, resulting in poor accuracy.

Prabhjot Kaur et al. [10] “Hybrid Text Classification Method for Fake News Detection.” - aimed to utilize machine learning methods to detect fake news. Three common methods are utilized through their researches: Naïve Bayes, Neural Network and Support Vector Machine (SVM). Normalization technique is an essential stage in data cleansing prior machine learning is used to categorizing the data. The output proved that that Naïve Bayes has an accuracy of 96.08% for detecting fake messages. Two more advanced methods, the neural network and the machine vector (SVM) reached an accuracy of 99.90%.

4. METHODOLOGY

This section presents the methodology used for the classification. Using this model, a tool is implemented for detecting the fake articles. In this method supervised machine learning is used for classifying the dataset. The first step in this classification problem is dataset collection phase, followed by preprocessing, implementing features selection, then perform the training and testing of dataset and finally running the classifiers. Figure (1) describes the proposed system methodology.

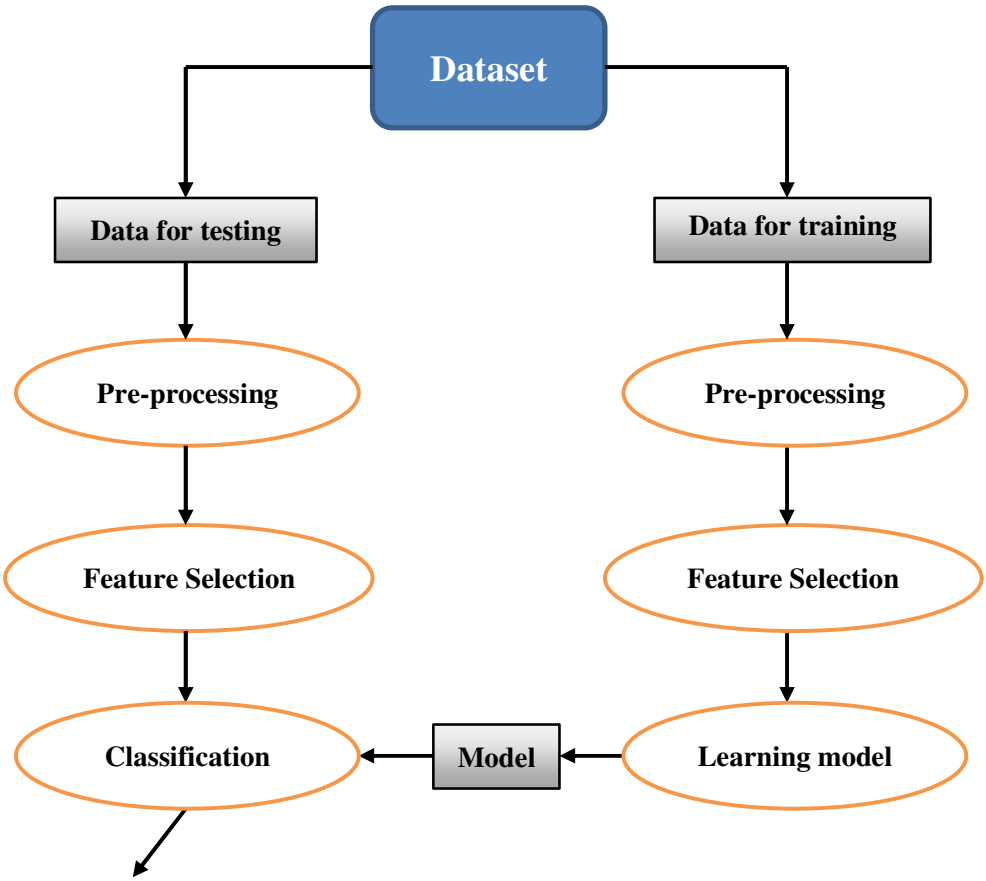


Figure (1): Describes the proposed System Methodology

5. EXPERIMENT

5.1 Data Preprocessing

Data processing is an essential step in building machine learning model and depending on how well the data has been processed; the results are seen. When it comes to textual data, there are a lot of inconsistencies, unnecessary words, punctuations and other special characters. To work with textual data, several pre-processing steps have to be applied to the data to transform words into numerical features that work with machine learning algorithms. The following techniques have been used in data pre-processing:

- *Tokenization*: it is the technique used to split the sentences into words. It helps in interpreting the meaning of the text by analyzing the sequence of the word.
- *Lowercasing*: a word in uppercase means the same in lowercase, hence this transforms all uppercase letters in the dataset to lowercase.
- *Stop words removal*: there are some common words in the dataset (such as “the”, “a”, “an”, “in”) that add no significant importance in classification or prediction. It uses the Bag of Words model which is used to extract features from text for use in modeling, such as ML algorithms. Therefore, we would

not want these words to take up the space in our database, or take up valuable processing time. By using this technique, stopwords are removed from the dataset.

- *Stemming*: Stemming is the process of producing morphological variants of a root/base word. Stemming programs are commonly referred to as stemming algorithms or stemmers. A stemming algorithm reduces the words “chocolates”, “chocolatey”, “choco” to the root word, “chocolate” and “retrieval”, “retrieved”, “retrieves” reduce to the stem “retrieve”. Here, PorterStemmer is the stemming algorithm used to perform this task.

5.2 Feature Extraction and Feature Selection

In order to use textual data for predictive modeling, the text must be parsed to remove certain words – this process is called tokenization. These words need to then be encoded as integers, or floating-point values, for use as inputs in machine learning algorithms. This process is called feature extraction or vectorization.

The two types of vectorizers used are Count Vectorizer and TF-IDF Vectorizer.

5.2.1 Count Vectorizer

CountVectorizer is a great tool provided by the scikit-learn library in Python. It is used to transform a given text into a vector on the basis of the frequency (count) of each word that occurs in the entire text. This is helpful when we have multiple such texts, and we wish to convert each word in each text into vectors.

CountVectorizer creates a matrix in which each unique word is represented by a column of the matrix, and each text sample from the document is a row in the matrix. The value of each cell is nothing but the count of the word in that particular text sample.

Inside CountVectorizer, these words are not stored as strings. Rather, they are given a particular index value. This way of representation is known as a Sparse Matrix.

5.2.2 TF-IDF Vectorizer

TF-IDF stands for term frequency-inverse document frequency and it is a measure, used in the fields

of information retrieval (IR) and machine learning, that can quantify the importance or relevance of string representations (words, phrases, lemmas, etc) in a document amongst a collection of documents.

TF (Term Frequency): works by looking at the frequency of a particular term you are concerned with relative to the document.

IDF (Inverse Document Frequency): Inverse document frequency

looks at how common (or uncommon) a word is amongst the corpus. IDF is calculated as follows

where t is the term (word) we are looking to measure the commonness of and N is the number of documents (d) in the corpus (D). The denominator is simply the number of documents in which the term, t, appears in.

$$idf(t, D) = log\left(\frac{N}{count\ (d \in D : t \in d)}\right)$$

To summarize the key intuition motivating TF-IDF is the importance of a term is inversely related to its frequency across documents.TF gives us information on how often a term appears in a document and IDF gives us information about the relative rarity of a term in the collection of documents. By multiplying these values together, we can get our final TF-IDF value.

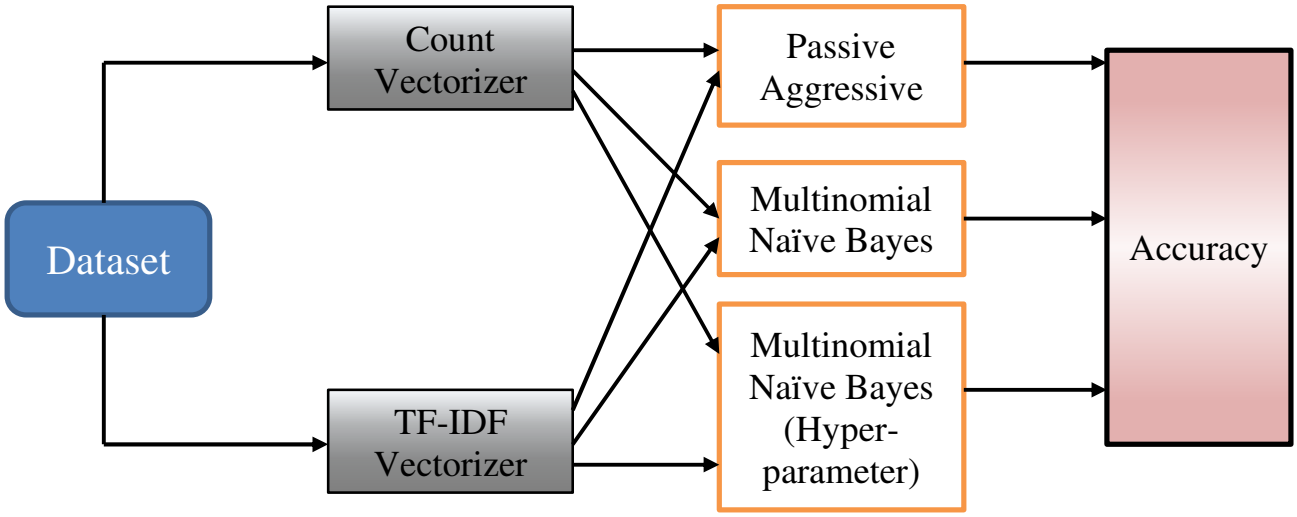


Figure (2): The Classification Algorithm

To determine whether the news is real or fake, we implement statistical models which predict based on the vectors. The two models that are implemented are:

5.3 Multinomial Naïve Bayes

Multinomial Naive Bayes is one of the most popular supervised learning classifications that is used for the analysis of the categorical text data.

This algorithm is a probabilistic learning method that is mostly used in Natural Language Processing (NLP). The algorithm is based on the Bayes theorem and predicts the tag of a text such as a piece of email or newspaper article.

It calculates the probability of each tag for a given sample and then gives the tag with the highest probability as output.

Naive Bayes classifier is a collection of many algorithms where all the algorithms share one common principle, and that is each feature being classified is not related to any other feature. The presence or absence of a feature does not affect the presence or absence of the other feature.

This algorithm works mainly on the Bayes theorem, which is:

$$P(A|B) = P(A) * P(B|A)/P(B)$$

Where we are calculating the probability of class A when predictor B is already provided.

P(B) = prior probability of B

P(A) = prior probability of class A

P(B|A) = occurrence of predictor B given class A probability

This formula helps in calculating the probability of the tags in the text.

5.4 Multinomial Naïve Bayes with Hyperparameter

Hyperparameters are parameters whose values control the learning process and determine the values of model parameters that a learning algorithm ends up learning. The prefix 'hyper_' suggests that they are 'top-level' parameters that control the learning process and the model parameters that result from it.

Hyperparameters are used by the learning algorithm when it is learning but they are not part of the resulting model. At the end of the learning process, the model parameters are trained which effectively is what we refer to as the model. The hyperparameters that were used during training are not part of this model. One

cannot for instance know what hyperparameter values were used to train a model from the model itself, we only know the model parameters that were learned.

Here, a hyper-parameter for the Multinomial Naïve Bayes is used to enhance the learning process of the model. The parameter used is called the alpha parameter. It is used to control the form of the model. It is also used to get rid of Laplace smoothing. So, what is basically done here is that a prediction is made for each and every value of alpha between 0 to 1 with an increment of 0.1. The best accuracy for a specific alpha value will be chosen for the respective problem.

5.5 Passive Aggressive Classifier

It is a family of ML online-learning algorithms. Passive-Aggressive algorithms are generally used for large-scale learning. It is one of the few ‘online-learning algorithms‘. In online machine learning algorithms, the input data comes in sequential order and the machine learning model is updated step-by-step, as opposed to batch learning, where the entire training dataset is used at once. This is very useful in situations where there is a huge amount of data and it is computationally infeasible to train the entire dataset because of the sheer size of the data. We can simply say that an online-learning algorithm will get a training example, update the

classifier, and then throw away the example.

A very good example of this would be to detect fake news on a social media website like Twitter, where new data is being added every second. To dynamically read data from Twitter continuously, the data would be huge, and using an online-learning algorithm would be ideal.

Passive-Aggressive algorithms are somewhat similar to a Perceptron model, in the sense that they do not require a learning rate. However, they do include a regularization parameter.

How it works is:

Passive: If the prediction is correct, keep the model and do not make any changes. i.e., the data in the example is not enough to cause any changes in the model.

Aggressive: If the prediction is incorrect, make changes to the model. i.e., some change to the model may correct it.

6.CODE

Fake News Classifier (COUNT VECTORIZER)

In [1]:

```
import pandas as pd
```

In [2]:

```
df=pd.read_csv('train.csv')
```

In [3]:

```
df.head()
```

Out[3]:

	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 [4]:

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

In [5]:

```
X.head()
```

Out[5]:

	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 [6]:

```
y=df['label']
```

In [7]:

```
y.head()
```

Out[7]:

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

In [8]:

```
df.shape
```

Out[8]:

```
(20800, 5)
```

In [9]:

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

In [10]:

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

In [11]:

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

Out[11]:

	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 [12]:

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

In [13]:

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

In [14]:

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

Out[14]:

	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 [15]:

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

Out[15]:

```
'Benoît Hamon Wins French Socialist Party's Presidential Nomination - The Ne
w York Times'
```

In [16]:

```
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 [17]:

```
corpus[3]
```

Out[17]:

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

In [18]:

```
from sklearn.feature_extraction.text import CountVectorizer
cv = CountVectorizer(max_features=5000,ngram_range=(1,3))
X = cv.fit_transform(corpus).toarray()
```

In [19]:

```
X.shape
```

Out[19]:

```
(18285, 5000)
```

In [20]:

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

In [21]:

```
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 [22]:

```
cv.get_feature_names()[:5]
```

Out[22]:

```
['abandon', 'abc', 'abc news', 'abduct', 'abe']
```

In [23]:

```
cv.get_params()
```

Out[23]:

```
{'analyzer': 'word',
 'binary': False,
 'decode_error': 'strict',
 'dtype': numpy.int64,
 'encoding': 'utf-8',
 'input': 'content',
 'lowercase': True,
 'max_df': 1.0,
 'max_features': 5000,
 'min_df': 1,
 'ngram_range': (1, 3),
 'preprocessor': None,
 'stop_words': None,
 'strip_accents': None,
 'token_pattern': '(?u)\\b\\w\\w+\\b',
 'tokenizer': None,
 'vocabulary': None}
```

In [24]:

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

In [25]:

```
count_df.head()
```

Out[25]:

	abandon	abc	abc news	abduct	abe	abedin	abl	abort	abroad	absolut	...	zero	zika	zika viru
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
2	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
4	0	0	0	0	0	0	0	0	0	1	...	0	0	0

5 rows × 5000 columns

In [26]:

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

In [27]:

```
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 [28]:

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

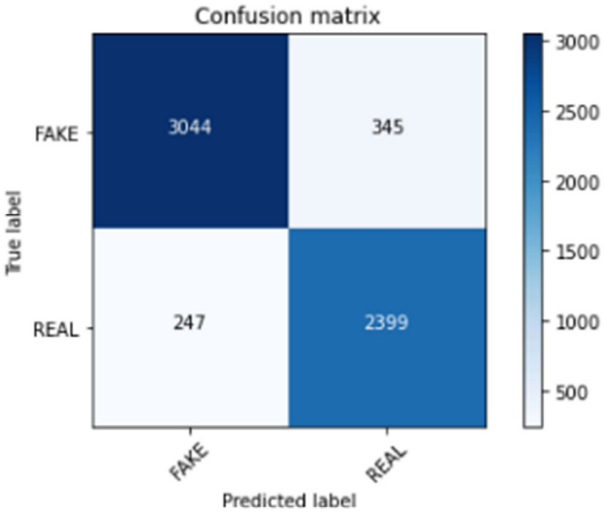
In [29]:

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

In [30]:

```
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:   0.902
Confusion matrix, without normalization
```



Passive Aggressive Classifier Algorithm

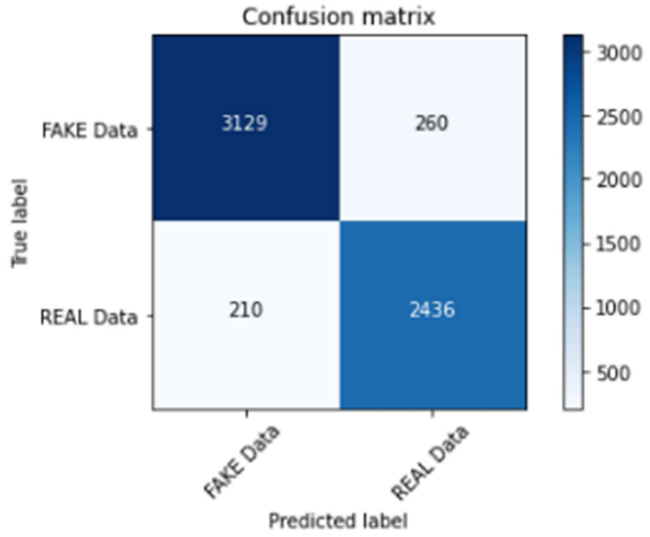
In [42]:

```
from sklearn.linear_model import PassiveAggressiveClassifier
linear_clf = PassiveAggressiveClassifier()
```

In [51]:

```
linear_clf.fit(X_train, y_train)
pred = linear_clf.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 Data', 'REAL Data'])
```

```
accuracy:   0.922
Confusion matrix, without normalization
```



Multinomial Classifier with Hyperparameter

In [45]:

```
classifier=MultinomialNB(alpha=0.1)
```

In [46]:

```
previous_score=0
for alpha in np.arange(0,1,0.1):
    sub_classifier=MultinomialNB(alpha=alpha)
    sub_classifier.fit(X_train,y_train)
    y_pred=sub_classifier.predict(X_test)
    score = metrics.accuracy_score(y_test, y_pred)
    if score>previous_score:
        classifier=sub_classifier
    print("Alpha: {}, Score : {}".format(alpha,score))
```

```
C:\Users\Aston Glen Noronha\anaconda3\lib\site-packages\sklearn\naive_bayes.
py:511: UserWarning: alpha too small will result in numeric errors, setting
alpha = 1.0e-10
  warnings.warn('alpha too small will result in numeric errors, '

Alpha: 0.0, Score : 0.8903065451532726
Alpha: 0.1, Score : 0.9020712510356255
Alpha: 0.2, Score : 0.9025683512841757
Alpha: 0.30000000000000004, Score : 0.9024026512013256
Alpha: 0.4, Score : 0.9017398508699255
Alpha: 0.5, Score : 0.9015741507870754
Alpha: 0.6000000000000001, Score : 0.9022369511184756
Alpha: 0.7000000000000001, Score : 0.9025683512841757
Alpha: 0.8, Score : 0.9015741507870754
Alpha: 0.9, Score : 0.9017398508699255
```

In [47]:

```
feature_names = cv.get_feature_names()
```

In [48]:

```
classifier.coef_[0]
```

Out[48]:

```
array([ -9.10038883,  -8.62276128,  -9.10038883, ..., -10.79498456,
        -8.91467169,  -9.32864749])
```

In [49]:

```
sorted(zip(classifier.coef_[0], feature_names), reverse=True)[:20]
```

Out[49]:

```
[(-4.000149156604985, 'trump'),
 (-4.287872694443541, 'hillari'),
 (-4.396389621061519, 'clinton'),
 (-4.899969726208735, 'elect'),
 (-5.176598600897756, 'new'),
 (-5.234730366348767, 'comment'),
 (-5.273968180973631, 'video'),
 (-5.3868167681180115, 'war'),
 (-5.396821854078974, 'us'),
 (-5.412019714988405, 'hillari clinton'),
 (-5.417137433425386, 'fbi'),
 (-5.48068448454208, 'vote'),
 (-5.566255475855405, 'email'),
 (-5.578238842742501, 'world'),
 (-5.634015380199913, 'obama'),
 (-5.734501455772904, 'donald'),
 (-5.763095255139644, 'donald trump'),
 (-5.785090276725191, 'russia'),
 (-5.846224665218559, 'day'),
 (-5.862110622807369, 'america')]
```

In [50]:

```
sorted(zip(classifier.coef_[0], feature_names))[:5000]
```

Out[50]:

```
[(-10.794984555596727, 'abe'),
 (-10.794984555596727, 'abroad'),
 (-10.794984555596727, 'abus new'),
 (-10.794984555596727, 'abus new york'),
 (-10.794984555596727, 'act new'),
 (-10.794984555596727, 'act new york'),
 (-10.794984555596727, 'advic'),
 (-10.794984555596727, 'advis new'),
 (-10.794984555596727, 'advis new york'),
 (-10.794984555596727, 'age new'),
 (-10.794984555596727, 'age new york'),
 (-10.794984555596727, 'agenda breitbart'),
 (-10.794984555596727, 'ail'),
 (-10.794984555596727, 'aleppo new'),
 (-10.794984555596727, 'aleppo new york'),
 (-10.794984555596727, 'ali'),
 (-10.794984555596727, 'america breitbart'),
 (-10.794984555596727, 'america new york'),
```

Fake News Classifier (TF-IDF VECTORIZER)

In [4]:

```
import pandas as pd
```

In [5]:

```
df=pd.read_csv('train.csv')
```

In [6]:

```
df.head()
```

Out[6]:

	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 [7]:

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

In [8]:

```
X.head()
```

Out[8]:

	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 [9]:

```
y=df['label']
```

In [10]:

```
y.head()
```

Out[10]:

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

In [11]:

```
df.shape
```

Out[11]:

```
(20800, 5)
```

In [12]:

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

In [13]:

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

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['text'][6]
```

Out[18]:

'PARIS — France chose an idealistic, traditional candidate in Sunday's primary to represent the Socialist and parties in the presidential electio n this spring. The candidate, Benoît Hamon, 49, who ran on the slogan that h e would "make France's heart beat," bested Manuel Valls, the former prime mi nister, whose campaign has promoted more policies and who has a strong background. Mr. Hamon appeared to have won by a wide margin, with incomplete returns showing him with an estimated 58 percent of the vote to Mr. Valls's 41 percent. "Tonight the left holds its head up high again it is looking to the future," Mr. Hamon said, addressing his supporters. "Our country needs t he left, but a modern, innovative left," he said. Mr. Hamon's victory was th e clearest sign yet that voters on the left want a break with the policies o f President François Hollande, who in December announced that he would not s eek . However, Mr. Hamon's strong showing is unlikely to change widespread assessments that candidates have little chance of making it into the secon d round of voting in the general election. The first round of the general el ection is set for April 23 and the runoff for May 7. The Socialist Party is deeply divided, and one measure of its lack of popular enthusiasm was the re latively low number of people voting. About two million people voted in the second round of the primary on Sunday, in contrast with about 2. 9 million i n the second round of the last presidential primary on the left, in 2011. Ho wever, much of the conventional wisdom over how the elections will go has be en thrown into question over the past week, because the leading candidate, F rançois Fillon, who represents the main party, the Republicans, was accuse d of paying his wife large sums of money to work as his parliamentary aide. While nepotism is legal in the French political system, it is not clear that she actually did any work. Prosecutors who specialize in financial malfeasan ce are reviewing the case. France's electoral system allows multiple candida tes to run for president in the first round of voting, but only the top two go on to a second round. Mr. Hamon is entering a race that is already crowde d on the left, with candidates who include Mélenchon on the far left, and Emmanuel Macron, an independent who served as economy minister in Mr. Hollan de's government and who embraces more policies. Unless he decides to withd raw, Mr. Fillon, the mainstream right candidate, will also run, as will the extreme right candidate Marine Le Pen. The two have been expected to go to t he runoff. Mr. Hamon's victory can be attributed at least in part to his ima ge as an idealist and traditional leftist candidate who appeals to union vot ers as well as more environmentally concerned and socially liberal young peo ple. Unlike Mr. Valls, he also clearly distanced himself from some of Mr. Ho llande's more unpopular policies, especially the economic ones. Thomas Keken bosch, 22, a student and one of the leaders of the group the Youth With Beno ît Hamon, said Mr. Hamon embodied a new hope for those on the left. "We have a perspective we have something to do, to build," Mr. Kekenbosch said. Mr. H ollande had disappointed many young people because under him the party aband oned ideals, such as support for workers, that many voters believe in, acc ording to Mr. Kekenbosch. Mr. Hollande's government, under pressure from the European Union to meet budget restraints, struggled to pass labor code refor ms to make the market more attractive to foreign investors and also to encou rage French businesses to expand in France. The measures ultimately passed a fter weeks of strikes, but they were watered down and generated little concr ete progress in improving France's roughly 10 percent unemployment rate and its nearly 25 percent youth joblessness rate. Mr. Hamon strongly endorses a stimulus approach to improving the economy and has promised to phase in a un iversal income, which would especially help young people looking for work, b ut would also supplement the livelihood of French workers. The end goal wo

```
omeone that trusts us," Mr. Kekenbosch said, "who says: 'I give you enough t
o pay for your studies. You can have a scholarship which spares you from wor
king at McDonald's on provisional contracts for 4 years. " Mr. Hamon advocat
es phasing out diesel fuel and encouraging drivers to replace vehicles that
use petroleum products with electrical ones. His leftist pedigree began earl
y. His father worked at an arsenal in Brest, a city in the far west of Britt
any, and his mother worked off and on as a secretary. He was an early member
of the Movement of Young Socialists, and he has continued to work closely wi
th them through his political life. He also worked for Martine Aubry, now th
e mayor of Lille and a former Socialist Party leader.'
```

In [ ]:

```
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['text'][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 [ ]:

```
corpus[3]
```

In [ ]:

```
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 [ ]:

```
X.shape
```

In [ ]:

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

In [ ]:

```
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 [21]:

```
tfidf_v.get_feature_names()[:8]
```

Out[21]:

```
['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 [22]:

```
tfidf_v.get_params()
```

Out[22]:

```
{'analyzer': 'word',
 'binary': False,
 'decode_error': 'strict',
 'dtype': numpy.int64,
 '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 [23]:

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

In [27]:

```
count_df.head()
```

Out[27]:

	abandon	abc	abc news	abduct	abe	abedin	abl	abort	abroad	absolut	...	zero	zika	zik vir
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.
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.
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.
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.
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.

5 rows × 5000 columns

In [28]:

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

In [29]:

```
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 [30]:

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

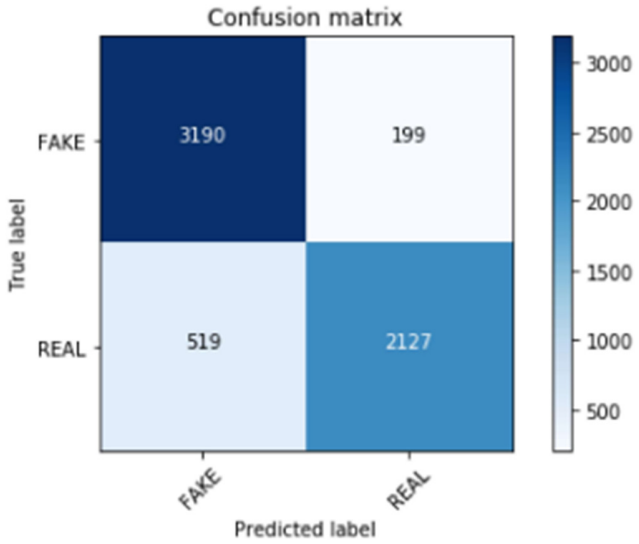
In [31]:

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

In [32]:

```
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 [33]:

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

Out[33]:

```
0.8810273405136703
```

In [34]:

```
y_train.shape
```

Out[34]:

```
(12250,)
```

Passive Aggressive Classifier Algorithm

In [35]:

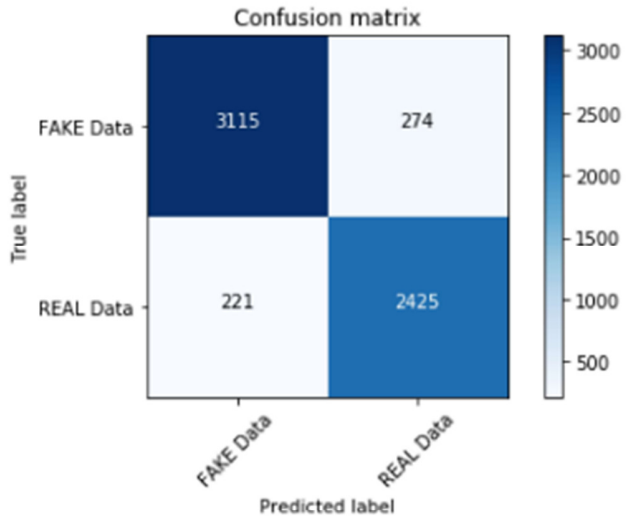
```
from sklearn.linear_model import PassiveAggressiveClassifier
linear_clf = PassiveAggressiveClassifier(n_iter=50)
```

In [36]:

```
linear_clf.fit(X_train, y_train)
pred = linear_clf.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 Data', 'REAL Data'])
```

```
c:\users\krish.naik\appdata\local\continuum\anaconda3\envs\nlp\lib\site-pack
ages\sklearn\linear_model\stochastic_gradient.py:117: DeprecationWarning: n_
iter parameter is deprecated in 0.19 and will be removed in 0.21. Use max_it
er and tol instead.
  DeprecationWarning)

accuracy:   0.918
Confusion matrix, without normalization
```



Multinomial Classifier with Hyperparameter

In [37]:

```
classifier=MultinomialNB(alpha=0.1)
```

In [38]:

```
previous_score=0
for alpha in np.arange(0,1,0.1):
    sub_classifier=MultinomialNB(alpha=alpha)
    sub_classifier.fit(X_train,y_train)
    y_pred=sub_classifier.predict(X_test)
    score = metrics.accuracy_score(y_test, y_pred)
    if score>previous_score:
        classifier=sub_classifier
    print("Alpha: {}, Score : {}".format(alpha,score))
```

```
c:\users\krish.naik\appdata\local\continuum\anaconda3\envs\nlp\lib\site-pack
ages\sklearn\naive_bayes.py:472: UserWarning: alpha too small will result in
numeric errors, setting alpha = 1.0e-10
  'setting alpha = %.1e' % _ALPHA_MIN)

Alpha: 0.0, Score : 0.8662800331400166
Alpha: 0.1, Score : 0.8777133388566695
Alpha: 0.2, Score : 0.8801988400994201
Alpha: 0.30000000000000004, Score : 0.87986743993372
Alpha: 0.4, Score : 0.8808616404308203
Alpha: 0.5, Score : 0.8806959403479702
Alpha: 0.6000000000000001, Score : 0.8815244407622204
Alpha: 0.7000000000000001, Score : 0.8813587406793704
Alpha: 0.8, Score : 0.8816901408450705
Alpha: 0.9, Score : 0.8816901408450705
```

In [106]:

```
## Get Features names
feature_names = cv.get_feature_names()
```

In [109]:

```
classifier.coef_[0]
```

Out[109]:

```
array([ -9.10038883,  -8.62276128,  -9.10038883, ..., -10.79498456,
        -8.91467169,  -9.32864749])
```

In [107]:

```
### Most real
sorted(zip(classifier.coef_[0], feature_names), reverse=True)[:20]
```

Out[107]:

```
[(-4.000149156604985, 'trump'),
 (-4.287872694443541, 'hillari'),
 (-4.396389621061519, 'clinton'),
 (-4.899969726208735, 'elect'),
 (-5.176598600897756, 'new'),
 (-5.234730366348767, 'comment'),
 (-5.273968180973631, 'video'),
 (-5.3868167681180115, 'war'),
 (-5.396821854078974, 'us'),
 (-5.412019714988405, 'hillari clinton'),
 (-5.417137433425386, 'fbi'),
 (-5.48068448454208, 'vote'),
 (-5.566255475855405, 'email'),
 (-5.578238842742501, 'world'),
 (-5.634015380199913, 'obama'),
 (-5.734501455772904, 'donald'),
 (-5.763095255139644, 'donald trump'),
 (-5.785090276725191, 'russia'),
 (-5.846224665218559, 'day'),
 (-5.862110622807369, 'america')]
```

In [135]:

```
### Most fake
sorted(zip(classifier.coef_[0], feature_names))[:5000]
```

Out[135]:

```
[(-10.794984555596727, 'abe'),
 (-10.794984555596727, 'abroad'),
 (-10.794984555596727, 'abus new'),
 (-10.794984555596727, 'abus new york'),
 (-10.794984555596727, 'act new'),
 (-10.794984555596727, 'act new york'),
 (-10.794984555596727, 'advic'),
 (-10.794984555596727, 'advis new'),
 (-10.794984555596727, 'advis new york'),
 (-10.794984555596727, 'age new'),
 (-10.794984555596727, 'age new york'),
 (-10.794984555596727, 'agenda breitbart'),
 (-10.794984555596727, 'ail'),
 (-10.794984555596727, 'aleppo new'),
 (-10.794984555596727, 'aleppo new york'),
 (-10.794984555596727, 'ali'),
 (-10.794984555596727, 'america breitbart'),
 (-10.794984555596727, 'america new york'),
```

7.RESULT

The scope of this project is to classify fake and true news. After performing an analysis on the dataset using two different vectorizers and two machine learning algorithms, the results are conveyed in the form of accuracy score and confusion matrices.

Accuracy Table

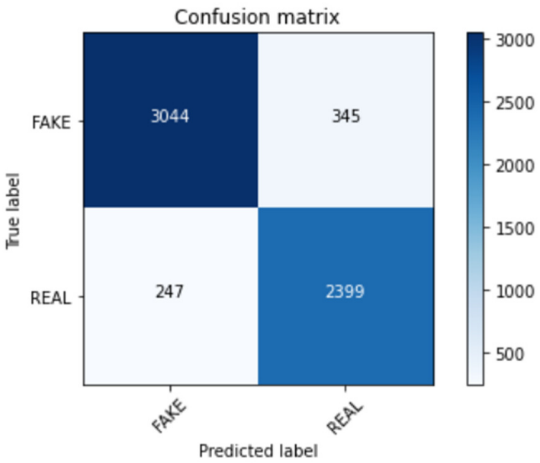
Vectorizer type/ ML Model	Multinomial Naïve Bayes	Multinomial Naïve Bayes (with hyperparameter)	Passive Aggressive Classifier
CountVectorizer	90.2%	90.25% (at Alpha=0.7)	92.2%
TFIDF Vectorizer	88.1%	88.169% (at Alpha=0.9)	91.8%

Using Count Vectorizer:

Figure (3) expresses the accuracy score and confusion matrix of the three statistical algorithms used. As shown, the Passive

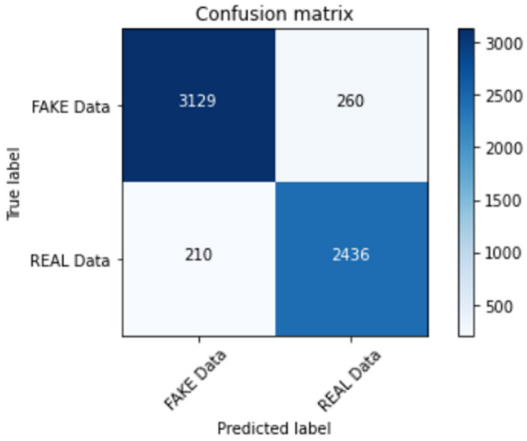
Aggressive algorithm is depicting the highest accuracy of 92.2%.

```
accuracy:   0.902
Confusion matrix, without normalization
```



Multinomial Naïve Bayes

```
accuracy:   0.922
Confusion matrix, without normalization
```



Passive Aggressive

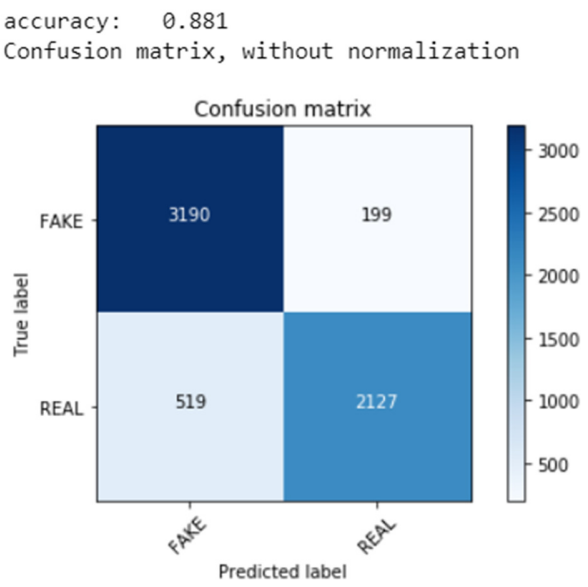
```
Alpha: 0.0, Score : 0.8903065451532726
Alpha: 0.1, Score : 0.9020712510356255
Alpha: 0.2, Score : 0.9025683512841757
Alpha: 0.30000000000000004, Score : 0.9024026512013256
Alpha: 0.4, Score : 0.9017398508699255
Alpha: 0.5, Score : 0.9015741507870754
Alpha: 0.6000000000000001, Score : 0.9022369511184756
Alpha: 0.7000000000000001, Score : 0.9025683512841757
Alpha: 0.8, Score : 0.9015741507870754
Alpha: 0.9, Score : 0.9017398508699255
```

Multinomial Naïve Bayes (with hyperparameter)

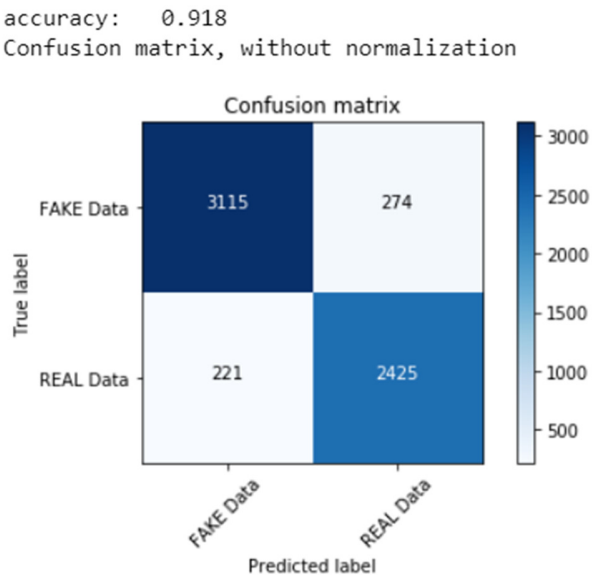
Figure (3): Accuracy score and confusion matrix of Count Vectorizer

Using TF-IDF Vectorizer:

Figure (4) expresses the accuracy score and confusion matrix of the three statistical algorithms used. As shown, the Passive Aggressive algorithm is depicting the highest accuracy of 92.2%.



Multinomial Naïve Bayes



Passive Aggressive

```
Alpha: 0.0, Score : 0.8662800331400166
Alpha: 0.1, Score : 0.8777133388566695
Alpha: 0.2, Score : 0.8801988400994201
Alpha: 0.30000000000000004, Score : 0.87986743993372
Alpha: 0.4, Score : 0.8808616404308203
Alpha: 0.5, Score : 0.8806959403479702
Alpha: 0.6000000000000001, Score : 0.8815244407622204
Alpha: 0.7000000000000001, Score : 0.8813587406793704
Alpha: 0.8, Score : 0.8816901408450705
Alpha: 0.9, Score : 0.8816901408450705
```

Multinomial Naïve Bayes (with hyperparameter)

Figure (4): Accuracy score and confusion matrix of TF-IDF Vectorizer

8. REFERENCES

1. Economic and Social Research Council. Using Social Media. Available at: https://esrc.ukri.org/research/impact-toolkit/social-media/using-social-media
2. Gil, P. Available at: https://www.lifewire.com/what-exactly-is-twitter-2483331. 2019, April 22.

3. E. C. Tandoc Jr et al. “Defining fake news a typology of scholarly definitions”. Digital Journalism , 1–17. 2017.

4. J. Radianti et al. “An Overview of Public Concerns During the Recovery Period after a Major Earthquake: Nepal Twitter Analysis.” HICSS '16 Proceedings of the 2016 49th Hawaii International Conference on System Sciences (HICSS) (pp. 136-145). Washington, DC, USA: IEEE. 2016.

5. Jeonghee Yi et al. “Sentiment analyzer: Extracting sentiments about a given topic using natural language processing techniques.” In Data Mining, 2003. ICDM 2003. Third IEEE International Conference (pp. 427-434). http://citeseerx.ist.psu.edu. 200).2003

6. Sumeet Dua, Xian Du. “Data Mining and Machine Learning in

Cybersecurity". New York: Auerbach Publications.19 April 2016.

7. Jasmin Kevric et el. "An effective combining classifier approach using tree algorithms for network intrusion detection." Neural Computing and Applications, 1051–1058. 2017.

8. MykhailoGranik and VolodymyrMesyura. "Fake news detection using naive Bayes classifier." First Ukraine Conference on Electrical and Computer Engineering (UKRCON). Ukraine: IEEE. 2017.

9. Gilda, S. "Evaluating machine learning algorithms for fake news detection." 15th Student Conference on Research and Development (SCOReD) (pp. 110-115). IEEE. 2017.

10. Prabhjot Kaur et al. "Hybrid Text Classification Method for Fake News Detection." International Journal of Engineering and Advanced Technology (IJEAT), 2388-2392. 2019.

Vectorizers: https://scikit-learn.org/

NLP Techniques: https://www.udemy.com/course/python- for-data-science-and-machine-learning-bootcamp/learn/lecture/5733574#overview