Report

(Fake News Detection)

MACHINE LEARNING

Section: KM031

Course code: INT247

Faculty: Dr. Aditiya Khamparia

Submitted By

Roll no.	**Registration no.**	**Name**
15	11708262	KIRAN PATHRO
16	11708716	HARISH MANI

INTRODUCTION

Fake News:- A type of yellow journalism, fake news encapsulates pieces of news that may be hoaxes and is generally spread through social media and other online media. This is often done to further or impose certain ideas and is often achieved with political agendas. Such news items may contain false and/or exaggerated claims, and may end up being viralized by algorithms, and users may end up in a filter bubble.

These days' fake news is creating different issues from sarcastic articles to a fabricated news and plan government propaganda in some outlets. Fake news and lack of trust in the media are growing problems with huge ramifications in our society. Obviously, a purposely misleading story is "fake news " but lately blathering social media's discourse is changing its definition. Some of them now use the term to dismiss the facts counter to their preferred viewpoints.

The importance of disinformation within American political discourse was the subject of weighty attention, particularly following the American president election. The term 'fake news' became common parlance for the issue, particularly to describe factually incorrect and misleading articles published mostly for the purpose of making money through page views.

Facebook has been at the epicenter of much critique following media attention. They have already implemented a feature to flag fake news on the site when a user sees it ; they have also said publicly they are working on to to distinguish these articles in an automated way. Certainly, it is not an easy task. A given algorithm must be politically unbiased – since fake news exists on both ends of the spectrum – and also give equal balance to legitimate news sources on either end of the spectrum. In addition, the question of legitimacy is a difficult one. However, in order to solve this problem, it is necessary to have an understanding on what Fake News is. Later, it is needed to look into how the techniques in the fields of machine learning, natural language processing help us to detect fake news.

REQUIREMENTS

- Python
- numpy
- pandas
- itertools
- matplotlib
- sklearn

Download Dataset:

https://drive.google.com/file/d/1er9NJTLUA3qnRuyhfzuN0XUsoIC4a-_q/view

Read the data

```
In [14]: df=pd.read_csv('F:/web/news.csv')

In [15]: df.shape
Out[15]: (6335, 4)

In [16]: df.head()
```

Out[16]:

	Unnamed: 0	title	text	label
0	8476	You Can Smell Hillary's Fear	Daniel Greenfield, a Shillman Journalism Fello...	FAKE
1	10294	Watch The Exact Moment Paul Ryan Committed Pol...	Google Pinterest Digg Linkedin Reddit Stumbleu...	FAKE
2	3608	Kerry to go to Paris in gesture of sympathy	U.S. Secretary of State John F. Kerry said Mon...	REAL
3	10142	Bernie supporters on Twitter erupt in anger ag...	— Kaydee King (@KaydeeKing) November 9, 2016 T...	FAKE
4	875	The Battle of New York: Why This Primary Matters	It's primary day in New York and front-runners...	REAL

TfidfVectorizer:

TF (Term Frequency): The number of times a word appears in a document is its Term Frequency. A higher value means a term appears more often than others, and so, the document is a good match when the term is part of the search terms.

IDF (Inverse Document Frequency): Words that occur many times a document, but also occur many times in many others, may be irrelevant. IDF is a measure of how significant a term is in the entire corpus.

The TfidfVectorizer converts a collection of raw documents into a matrix of TF-IDF features.

Initialize a TfidfVectorizer

```
In [19]: tfidf_vectorizer=TfidfVectorizer(stop_words='english', max_df=0.7)

In [20]: tfidf_train=tfidf_vectorizer.fit_transform(x_train)
         tfidf_test=tfidf_vectorizer.transform(x_test)
```

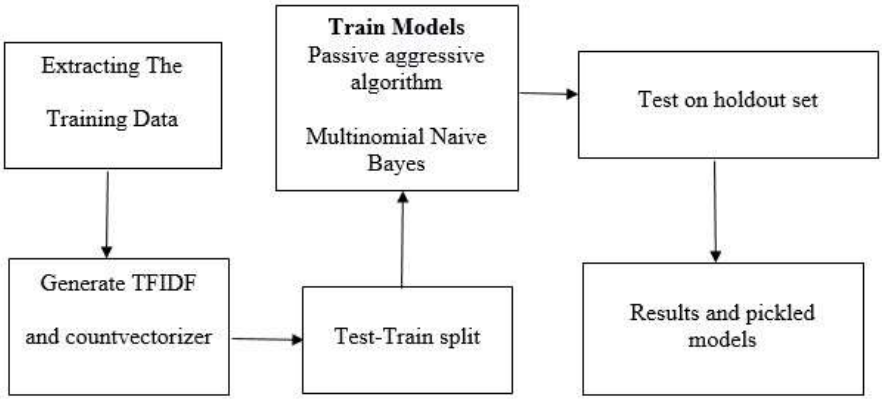
Splitting Data:

Split the dataset

```
In [18]: x_train,x_test,y_train,y_test=train_test_split(df['text'], labels, test_size=0.2, random_state=7)
```

Models

Passive Aggressive Classifier



Passive Aggressive algorithms are online learning algorithms. Such an algorithm remains passive for a correct classification outcome, and turns aggressive in the event of a miscalculation,

updating and adjusting. Unlike most other algorithms, it does not converge. Its purpose is to make updates that correct the loss, causing very little change in the norm of the weight vector.

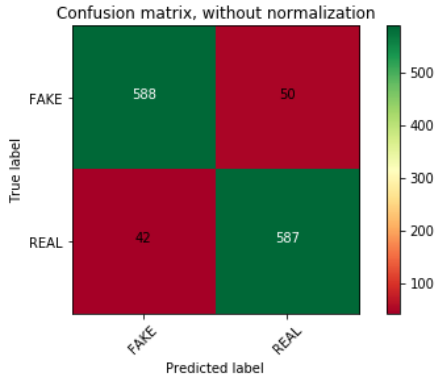
Predict on the test set and calculate accuracy

In [41]:
```
y_pred=pac.predict(tfidf_test)
score=accuracy_score(y_test,y_pred)
print(f'Accuracy: {round(score*100,2)}%')
```

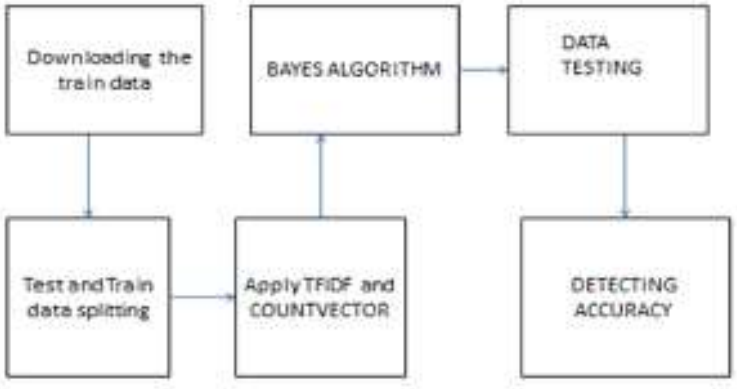
```
Accuracy: 92.74%
```

In [42]:
```
confusion = metrics.confusion_matrix(y_test,y_pred)
plt.figure()
plot_confusion_matrix(confusion, classes=['FAKE','REAL'], title='Confusion matrix, without normalization')
```

```
Confusion matrix, without normalization
```



Naive Bayes

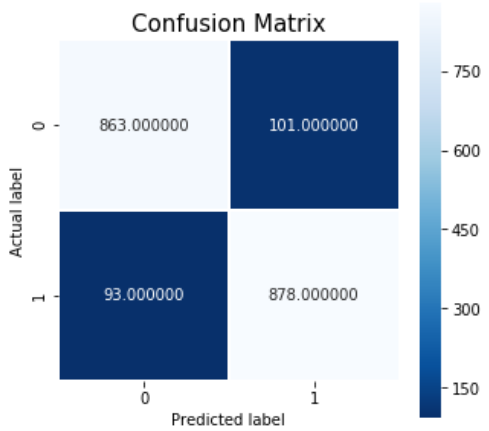


Naive Bayes classifiers are a collection of classification algorithms based on Bayes' Theorem. It is not a single algorithm but a family of algorithms where all of them share a common principle, i.e. every pair of features being classified is independent of each other.

```
[18]: print('accuracy {}'.format(accuracy_score(test_set['label'], all_predictions2)))
      print('confusion matrix\n {}'.format(confusion_matrix(test_set['label'], all_predictions2)))
      print('(row=expected, col=predicted)')
```

```
accuracy 0.8997416020671835
confusion matrix
 [[863 101]
 [ 93 878]]
(row=expected, col=predicted)
```

	precision	recall	f1-score	support
FAKE	0.90	0.90	0.90	964
REAL	0.90	0.90	0.90	971
avg / total	0.90	0.90	0.90	1935



Support Vector Machine

A support vector machine (SVM), which can be used interchangeably with a support vector network (SVN), is also considered to be a supervised learning algorithm. SVMs work by being trained with specific data already organized into two different categories. Hence, the model is constructed after it has already been trained. Furthermore, the goal of the SVM method is to distinguish which category any new data falls under, in addition, it must also maximize the margin between the two classes

Support vectors are the data points, which are closest to the hyperplane. These points will define the separating line better by calculating margins. These points are more relevant to the construction of the classifier.

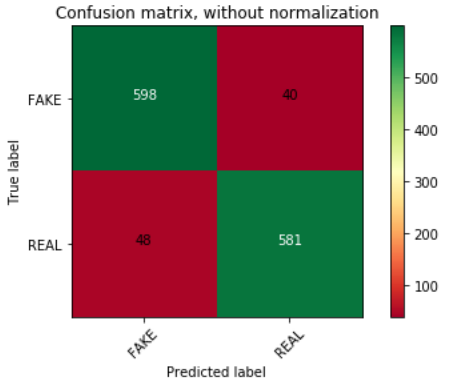
Predict on the test set and calculate accuracy

In [11]:
```
y_pred=svm.predict(tfidf_test)
score=accuracy_score(y_test,y_pred)
print(f'Accuracy: {round(score*100,2)}%')
```

```
Accuracy: 93.05%
```

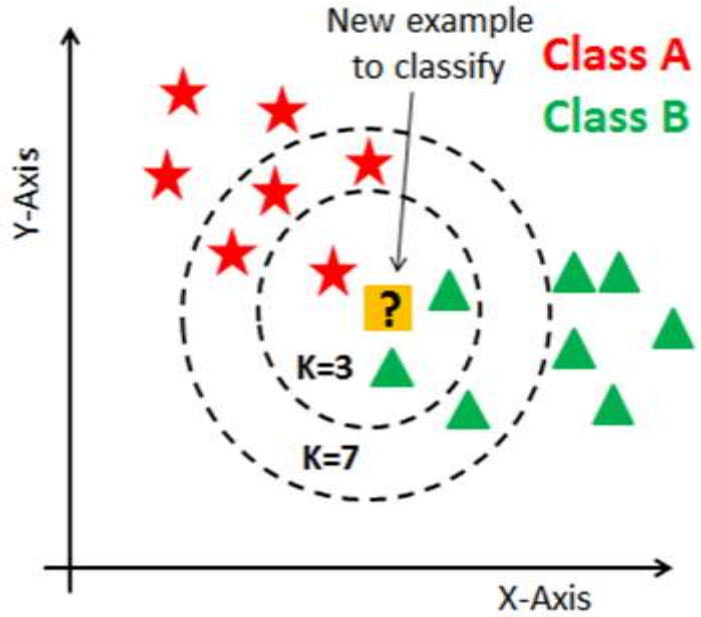
In [12]:
```
confusion = metrics.confusion_matrix(y_test,y_pred)
plt.figure()
plot_confusion_matrix(confusion, classes=['FAKE','REAL'], title='Confusion matrix, without normalization')
```

```
Confusion matrix, without normalization
```



knearest neighbors Classifier

The KNN algorithm assumes that similar things exist in close proximity.



In KNN, K is the number of nearest neighbors. The number of neighbors is the core deciding factor. K is generally an odd number if the number of classes is 2. When K=1, then the algorithm is known as the nearest neighbor algorithm. This is the simplest case.

KNN has the following basic steps:-

- Calculate distance
- Find closest neighbors
- Vote for labels

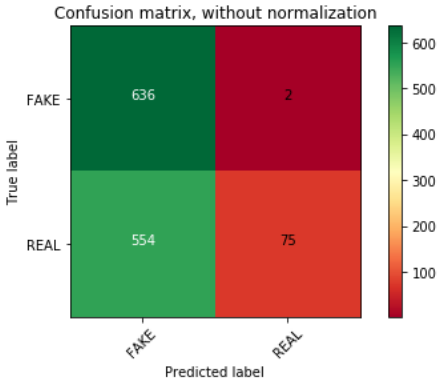
Predict on the test set and calculate accuracy

```
In [23]: y_pred=knn.predict(tfidf_test)
         score=accuracy_score(y_test,y_pred)
         print(f'Accuracy: {round(score*100,2)}%')

         Accuracy: 56.12%
```

```
In [24]: confusion = metrics.confusion_matrix(y_test,y_pred)
         plt.figure()
         plot_confusion_matrix(confusion, classes=['FAKE','REAL'], title='Confusion matrix, without normalization')

         Confusion matrix, without normalization
```



Results:

Methods	**Accuracy**
PassiveAggressiveClassifier	92.74
Naive Bayes	89
Support Vector Machine	93.05
knearest neighbors Classifier	56.12