2CS404: PROGRAMMING FOR SCIENTIFIC COMPUTING

SEMESTER-4

INNOVATIVE ASSIGNMENT

FAKE NEWS DETECTION

Prepared By :

21BCE019: Ayushi Shah

21BCE076: Keyaba Gohil

❖ INTRODUCTION-

The advent of the World Wide Web and the rapid adoption of social media platforms such as Facebook and Twitter paved the way for information dissemination that has never been witnessed in the human history before. With the current usage of social media platforms, consumers are creating and sharing more information than ever before, some of which are misleading with no relevance to reality. Automated classification of a text article as misinformation or disinformation is a challenging task. Even an expert in a particular domain has to explore multiple aspects before giving a verdict on the truthfulness of an article.

Using this project, we propose to use machine learning ensemble approach for automated classification of news articles. Our study explores different textual properties that can be used to distinguish fake contents from real. By using those properties, we train a combination of different machine learning algorithms using various ensemble methods and evaluate their performance on real world datasets. We have used 6 Machine Learning algorithms for the same. These include:

- Logistic Regression
- Decision Tree Classifier
- XG Boost
- Support Vector Machine (SVM)
- Random Forest Classifier
- Naïve Bayes

❖ PYTHON CONCEPTS USED IN PROJECT-

- Pandas
- NUMPY array
- Matplotlib plots
- Machine Learning based algorithms

❖ LIBRARIES TO BE INSTALLED-

```
pip install pandas
pip install numpy
pip install scikit-learn
pip install matplotlib
pip install nltk
pip install wordcloud
pip install xgboost
```

❖ LIBRARIES TO BE IMPORTED-

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.feature_extraction.text import TfidfTransformer
from sklearn import feature_extraction, linear_model, model_selection, preprocessing
from sklearn.metrics import accuracy_score
from sklearn.model_selection import train_test_split
from sklearn.pipeline import Pipeline
```

❖ LINK TO INSTALL DATASETS-

https://drive.google.com/drive/folders/1ByadNwMrPyds53cA6SDCHLelTAvIdoF_

❖ ABOUT MACHINE LEARNING BASED ALGORITHMS USED IN PROJECT-

1. Logistic Regression-
   Logistic Regression is used to estimate discrete values (usually binary values like 0/1) from a set of independent variables. It helps predict the probability of an event by fitting data to a logit function. It is also called logit regression.

2. Decision Tree Classifier-
   Decision Tree algorithm in machine learning is one of the most popular algorithm in use today; this is a supervised learning algorithm that is used for classifying

problems. It works well in classifying both categorical and continuous dependent variables. This algorithm divides the population into two or more homogeneous sets based on the most significant attributes/ independent variables.

3. XG Boost-

   XGBoost, which stands for Extreme Gradient Boosting, is a scalable, distributed gradient-boosted decision tree (GBDT) machine learning library. It provides parallel tree boosting and is the leading machine learning library for regression, classification, and ranking problems. A Gradient Boosting Decision Trees (GBDT) is a decision tree ensemble learning algorithm similar to random forest, for classification and regression. Ensemble learning algorithms combine multiple machine learning algorithms to obtain a better model.Both random forest and GBDT build a model consisting of multiple decision trees. The difference is in how the trees are built and combined.

4. Support Vector Machine(SVM)-

   SVM algorithm is a method of a classification algorithm in which you plot raw data as points in an n-dimensional space (where n is the number of features you have). The value of each feature is then tied to a particular coordinate, making it easy to classify the data. Lines called classifiers can be used to split the data and plot them on a graph.

5. Random Forest Classifier-

   A collective of decision trees is called a Random Forest. To classify a new object based on its attributes, each tree is classified, and the tree “votes” for that class. The forest chooses the classification having the most votes (over all the trees in the forest).Each tree is planted & grown as follows: If the number of cases in the training set is N, then a sample of N cases is taken at random. This sample will be the training set for growing the tree, If there are M input variables, a number m<<M is specified such that at each node, m variables are selected at random out of the M, and the best split on this m is used to split the node. The value of m is held constant during this process, Each tree is grown to the most substantial extent possible. There is no pruning.

6. Naïve Bayes-

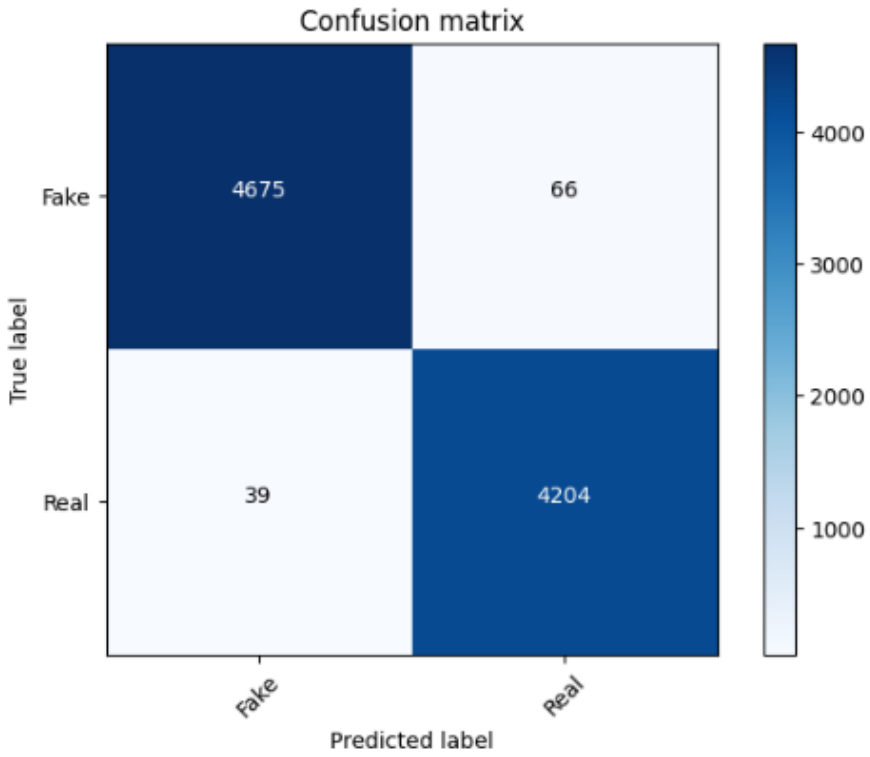
   Naïve Bayes algorithm is a supervised learning algorithm, which is based on Bayes theorem and used for solving classification problems. It is mainly used in *text*

classification that includes a high-dimensional training dataset. Naïve Bayes Classifier is one of the simple and most effective Classification algorithms which helps in building the fast machine learning models that can make quick predictions. It is a probabilistic classifier, which means it predicts on the basis of the probability of an object.

❖ OUTPUT OF WORKING MODEL-

In [29]:
```
#Logistic Regression

from sklearn.linear_model import LogisticRegression

lr=LogisticRegression()
lr.fit(xv_train,y_train)
pred_lr=lr.predict(xv_test)
#print(pred_lr)
lr_score=lr.score(xv_test,y_test)
print(lr_score)
dct['Logistic Regression'] = round(accuracy_score(y_test, prediction)*100,2)
print("accuracy: {}%".format(round(accuracy_score(y_test, prediction)*100,2)))
#print(classification_report(y_test,pred_lr))

cm = metrics.confusion_matrix(y_test, pred_lr)
plot_confusion_matrix(cm, classes=['Fake', 'Real'])
```

```
0.9883125556544969
accuracy: 99.57%
Confusion matrix, without normalization
```

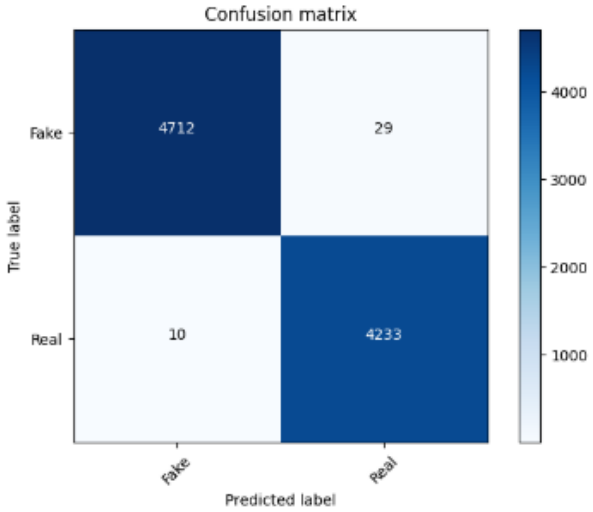


In [24]:
```
from sklearn.tree import DecisionTreeClassifier

dt=DecisionTreeClassifier(max_features=None)
dt.fit(xv_train,y_train)
# Accuracy
prediction = dt.predict(xv_test)
#print(prediction)
dt_score=dt.score(xv_test,y_test)
print(dt_score)
print("accuracy: {}%".format(round(accuracy_score(y_test, prediction)*100,2)))
```

```
0.9956589492430988
accuracy: 99.57%
```

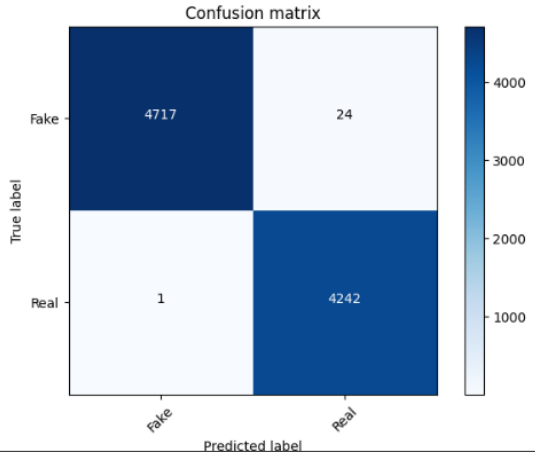
In [27]:
```
cm = metrics.confusion_matrix(y_test, prediction)
plot_confusion_matrix(cm, classes=['Fake', 'Real'])
```

```
Confusion matrix, without normalization
```



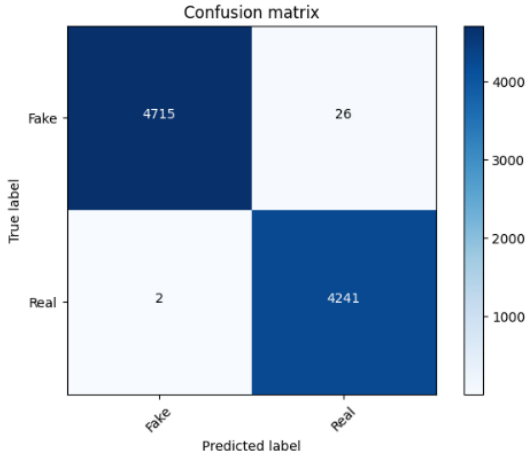
In [31]:
```
#XGBoost

import xgboost as xgb
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder, OneHotEncoder

# y_train=temp
# np.where(temp=='fake',0,1)
# print(temp)
# Fitting XGBoost to the training data
my_model = xgb.XGBClassifier()
my_model.fit(xv_train,temp)
temp2=y_test
temp2=np.where(temp2=="fake",0,1)
my_model_score=my_model.score(xv_test,temp2)
# Predicting the Test set results
y_pred = my_model.predict(xv_test)
print(my_model_score)
dct['XG Boost'] = round(accuracy_score(temp2, y_pred)*100,2)
print("accuracy: {}%".format(round(accuracy_score(temp2, y_pred)*100,2)))

cm = metrics.confusion_matrix(temp2, y_pred)
plot_confusion_matrix(cm, classes=['Fake', 'Real'])
```

```
0.9972172751558326
accuracy: 99.72%
Confusion matrix, without normalization
```



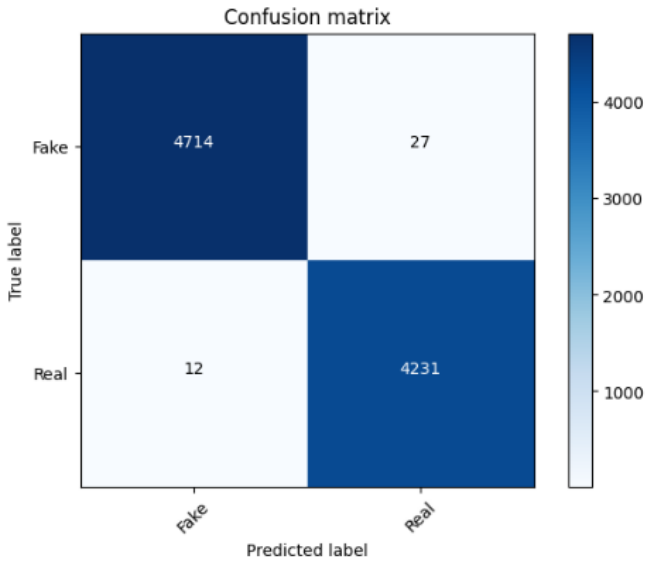
In [35]:
```
#random forest classifier
from sklearn.ensemble import RandomForestClassifier

temp=y_train
temp=np.where(temp=="fake",0,1)

rfc = RandomForestClassifier(max_features=None)
rfc.fit(xv_train,temp)
prediction_rfc =rfc.predict(xv_test)
temp2=y_test
temp2=np.where(temp2=="fake",0,1)
rfc_score=rfc.score(xv_test,temp2)
#print(prediction)
print(rfc_score)
print("accuracy: {}%".format(round(accuracy_score(temp2, prediction_rfc)*100,2)))
dct['Random Forest'] = round(accuracy_score(temp2, prediction_rfc)*100,2)

cm = metrics.confusion_matrix(temp2, prediction_rfc)
plot_confusion_matrix(cm, classes=['Fake', 'Real'])
```

```
0.9968833481745325
accuracy: 99.69%
Confusion matrix, without normalization
```



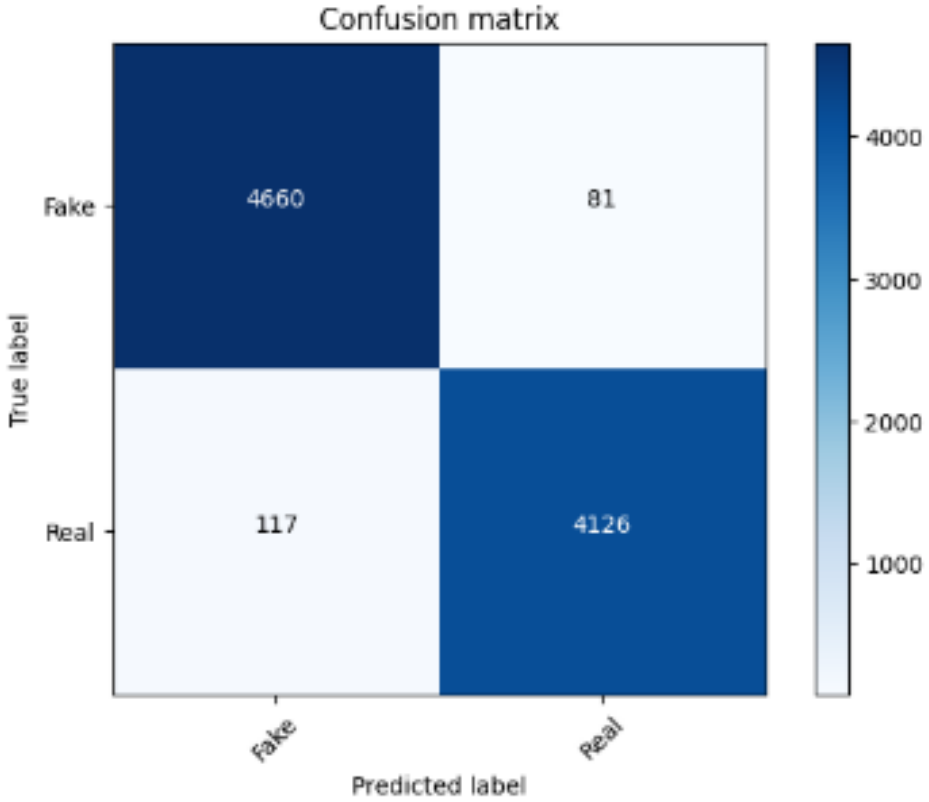
:
```
#svm

from sklearn import svm

temp=y_train
temp=np.where(temp=="fake",0,1)

temp2=y_test
temp2=np.where(temp2=="fake",0,1)

sv=svm.SVC(kernel='linear')
sv.fit(xv_train,temp)
prediction_svm=sv.predict(xv_test)
sv_score=sv.score(xv_test,temp2)
print(sv_score)
print("accuracy: {}%".format(round(accuracy_score(temp2, prediction_svm)*100,2)))
dct['SVM'] = round(accuracy_score(temp2, prediction_svm)*100,2)

cm = metrics.confusion_matrix(temp2, prediction_svm)
plot_confusion_matrix(cm, classes=['Fake', 'Real'])
```

```
0.9956589492430988
accuracy: 99.57%
Confusion matrix, without normalization
```



```
#naive bayes
from sklearn.naive_bayes import MultinomialNB

temp=y_train
temp=np.where(temp=="fake",0,1)

nb=MultinomialNB(alpha=0.0)
nb.fit(xv_train,temp)
prediction_nb=nb.predict(xv_test)
#print(prediction_nb)

# print(y_test)
# print(temp)
nb_score=nb.score(xv_test,temp2)
print(nb.score)
print("accuracy: {}%".format(round(accuracy_score(temp2, prediction_nb)*100,2)))
dct['Naive Bayes'] = round(accuracy_score(temp2, prediction_nb)*100,2)

cm = metrics.confusion_matrix(temp2, prediction_nb)
plot_confusion_matrix(cm, classes=['Fake', 'Real'])
```

```
<bound method ClassifierMixin.score of MultinomialNB(alpha=0.0)>
accuracy: 97.8%
Confusion matrix, without normalization
```



```
def fake_news(news):
    news=remove_punct(news)
    input_data=[news]
    ser = pd.Series(input_data)
    ser=ser.apply(lambda x: ' '.join([word for word in x.split() if word not in (stop)]))
    v_inputdata=v.transform(ser)
    #v_inputdata=v.transform(input_data)
    prediction=lr.predict(v_inputdata)
    if(prediction[0]=="true"):
        print("Using Logisitic Regression : {} ".format("TRUE NEWS"))
    else:
        print("Using Logisitic Regression : {} ".format("FAKE NEWS"))

    prediction2=dt.predict(v_inputdata)
    if(prediction2[0]=="true"):
        print("Using Decison Tree Classifier : {} ".format("TRUE NEWS"))
    else:
        print("Using Decison Tree Classifier : {} ".format("FAKE NEWS"))


    prediction3=my_model.predict(v_inputdata)
    if(prediction3==0):
        print("Using Xgboost : {}".format("FAKE NEWS"))
    else:
        print("Using Xgboost : {}".format("TRUE NEWS"))


    prediction4=sv.predict(v_inputdata)
    if(prediction4==0):
        print("Using SVM : {}".format("FAKE NEWS"))
    else:
        print("Using SVM : {}".format("TRUE NEWS"))


    prediction5=nb.predict(v_inputdata)
    if(prediction5==0):
        print("Using Naive Bayes : {}".format("FAKE NEWS"))
    else:
        print("Using Naive Bayes : {}".format("TRUE NEWS"))



    prediction6=rfc.predict(v_inputdata)
    if(prediction6==0):
        print("Using Random Forest : {}".format("FAKE NEWS"))
    else:
        print("Using Random Forest : {}".format("TRUE NEWS"))




news=input("Enter news: ")
print(" ")
fake_news(news)
```

```
Enter news: Donald Trump just couldn t wish all Americans a Happy New Year and leave it at that. Instead, he had to give a shou
t out to his enemies, haters and  the very dishonest fake news media.  The former reality show star had just one job to do and
he couldn t do it. As our Country rapidly grows stronger and smarter, I want to wish all of my friends, supporters, enemies, ha
ters, and even the very dishonest Fake News Media, a Happy and Healthy New Year,  President Angry Pants tweeted.  2018 will be
a great year for America! As our Country rapidly grows stronger and smarter, I want to wish all of my friends, supporters, enem
ies, haters, and even the very dishonest Fake News Media, a Happy and Healthy New Year. 2018 will be a great year for America!
Donald J. Trump (@realDonaldTrump) December 31, 2017Trump s tweet went down about as welll as you d expect.What kind of preside
nt sends a New Year s greeting like this despicable, petty, infantile gibberish? Only Trump! His lack of decency won t even all
ow him to rise above the gutter long enough to wish the American citizens a happy new year!  Bishop Talbert Swan (@TalbertSwan)
December 31, 2017no one likes you  Calvin (@calvinstowell) December 31, 2017Your impeachment would make 2018 a great year for A
merica, but I ll also accept regaining control of Congress.  Miranda Yaver (@mirandayaver) December 31, 2017Do you hear yoursel
f talk? When you have to include that many people that hate you you have to wonder? Why do the they all hate me?  Alan Sandoval
(@AlanSandoval13) December 31, 2017Who uses the word Haters in a New Years wish??  Marlene (@marlene399) December 31, 2017You c
an t just say happy new year?  Koren pollitt (@Korencarpenter) December 31, 2017Here s Trump s New Year s Eve tweet from 2016.H
appy New Year to all, including to my many enemies and those who have fought me and lost so badly they just don t know what to
do. Love!  Donald J. Trump (@realDonaldTrump) December 31, 2016This is nothing new for Trump. He s been doing this for years.Tr
ump has directed messages to his  enemies  and  haters  for New Year s, Easter, Thanksgiving, and the anniversary of 9/11. pic.
twitter.com/4FPAe2KypA  Daniel Dale (@ddale8) December 31, 2017Trump s holiday tweets are clearly not presidential.How long did
he work at Hallmark before becoming President?  Steven Goodine (@SGoodine) December 31, 2017He s always been like this . . . th
e only difference is that in the last few years, his filter has been breaking down.  Roy Schulze (@thbthttt) December 31, 2017W
ho, apart from a teenager uses the term haters?  Wendy (@WendyWhistles) December 31, 2017he s a fucking 5 year old  Who Knows
(@rainyday80) December 31, 2017So, to all the people who voted for this a hole thinking he would change once he got into power,
you were wrong! 70-year-old men don t change and now he s a year older.Photo by Andrew Burton/Getty Images.

Using Logisitic Regression : FAKE NEWS
Using Decison Tree Classifier : FAKE NEWS
Using Xgboost : FAKE NEWS
Using SVM : FAKE NEWS
Using Naive Bayes : FAKE NEWS
Using Random Forest : FAKE NEWS
```