Fake_News_Detection

April 24, 2022

1 Project I : Fake News Detection.

[1]:
```
#Importing libraries.
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt

import string as st
from wordcloud import WordCloud
import nltk
from nltk import PorterStemmer, WordNetLemmatizer
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score, confusion_matrix
from sklearn.ensemble import RandomForestClassifier
from sklearn.linear_model import LogisticRegression
from sklearn.linear_model import PassiveAggressiveClassifier
from sklearn.svm import SVC
from sklearn.tree import DecisionTreeClassifier
from mlxtend.plotting import plot_confusion_matrix
```

[2]:
```
#Loading data.
news_df = pd.read_csv('news.csv')
```

[3]:
```
#checking the data structure.
news_df.shape
```

[3]:
```
(6335, 4)
```

[4]:
```
#Glimpse of the data.
news_df.head()
```

[4]:
```
   Unnamed: 0                                              title  \
0        8476                       You Can Smell Hillary's Fear
1       10294  Watch The Exact Moment Paul Ryan Committed Pol…
2        3608        Kerry to go to Paris in gesture of sympathy
3       10142  Bernie supporters on Twitter erupt in anger ag…
```

```
4         875   The Battle of New York: Why This Primary Matters

                                                text label
0  Daniel Greenfield, a Shillman Journalism Fello…  FAKE
1  Google Pinterest Digg Linkedin Reddit Stumbleu…  FAKE
2  U.S. Secretary of State John F. Kerry said Mon…  REAL
3  - Kaydee King (@KaydeeKing) November 9, 2016 T…  FAKE
4  It's primary day in New York and front-runners…  REAL
```

[5]:
```
#checking for null values.
news_df.isnull().sum()
```

[5]:
```
Unnamed: 0    0
title         0
text          0
label         0
dtype: int64
```

The dataset does not have null values.

2 Exploratory Data Analysis (EDA).

[6]:
```
#Removing all punctuations.
def remove_punctuation(text):
    return ("".join([ch for ch in text if ch not in st.punctuation]))
```

[7]:
```
news_df['New_text']=news_df['text'].apply(lambda x: remove_punctuation(x))
news_df.head()
```

[7]:
```
   Unnamed: 0                                              title  \
0        8476                       You Can Smell Hillary's Fear
1       10294  Watch The Exact Moment Paul Ryan Committed Pol…
2        3608        Kerry to go to Paris in gesture of sympathy
3       10142  Bernie supporters on Twitter erupt in anger ag…
4         875   The Battle of New York: Why This Primary Matters

                                                text label  \
0  Daniel Greenfield, a Shillman Journalism Fello…  FAKE
1  Google Pinterest Digg Linkedin Reddit Stumbleu…  FAKE
2  U.S. Secretary of State John F. Kerry said Mon…  REAL
3  - Kaydee King (@KaydeeKing) November 9, 2016 T…  FAKE
4  It's primary day in New York and front-runners…  REAL

                                            New_text
0  Daniel Greenfield a Shillman Journalism Fellow…
1  Google Pinterest Digg Linkedin Reddit Stumbleu…
2  US Secretary of State John F Kerry said Monday…
```

```
3  - Kaydee King KaydeeKing November 9 2016 The l…
4  Its primary day in New York and frontrunners H…
```

[8]:
```
#Dropping the feature which we don't need
news_df = news_df.drop(['Unnamed: 0'],axis=1)
news_df.head()
```

[8]:
```
                                                title  \
0                       You Can Smell Hillary’s Fear
1  Watch The Exact Moment Paul Ryan Committed Pol…
2        Kerry to go to Paris in gesture of sympathy
3  Bernie supporters on Twitter erupt in anger ag…
4   The Battle of New York: Why This Primary Matters

                                                text label  \
0  Daniel Greenfield, a Shillman Journalism Fello…  FAKE
1  Google Pinterest Digg Linkedin Reddit Stumbleu…  FAKE
2  U.S. Secretary of State John F. Kerry said Mon…  REAL
3  - Kaydee King (@KaydeeKing) November 9, 2016 T…  FAKE
4  It's primary day in New York and front-runners…  REAL

                                            New_text
0  Daniel Greenfield a Shillman Journalism Fellow…
1  Google Pinterest Digg Linkedin Reddit Stumbleu…
2  US Secretary of State John F Kerry said Monday…
3  - Kaydee King KaydeeKing November 9 2016 The l…
4  Its primary day in New York and frontrunners H…
```

[9]:
```
#Convert text in lower case, Split() applied for white space
import re
def tokenize(text):
    text = re.split('\s+', text)
    return [x.lower() for x in text]
```

[10]:
```
news_df['New_text'] = news_df['New_text'].apply(lambda msg:tokenize(msg))
news_df.head()
```

[10]:
```
                                                title  \
0                       You Can Smell Hillary’s Fear
1  Watch The Exact Moment Paul Ryan Committed Pol…
2        Kerry to go to Paris in gesture of sympathy
3  Bernie supporters on Twitter erupt in anger ag…
4   The Battle of New York: Why This Primary Matters

                                                text label  \
0  Daniel Greenfield, a Shillman Journalism Fello…  FAKE
1  Google Pinterest Digg Linkedin Reddit Stumbleu…  FAKE
```

```
2  U.S. Secretary of State John F. Kerry said Mon…  REAL
3  - Kaydee King (@KaydeeKing) November 9, 2016 T…  FAKE
4  It's primary day in New York and front-runners…  REAL

                                            New_text
0  [daniel, greenfield, a, shillman, journalism, …
1  [google, pinterest, digg, linkedin, reddit, st…
2  [us, secretary, of, state, john, f, kerry, sai…
3  [-, kaydee, king, kaydeeking, november, 9, 201…
4  [its, primary, day, in, new, york, and, frontr…
```

[11]:
```
#Removal of tokens less than length 2.
def remove_small_words(text):
    return [x for x in text if len(x)>2]
```

[12]:
```
news_df['New_text'] = news_df['New_text'].apply(lambda x: remove_small_words(x))
news_df.head()
```

[12]:
```
                                          title  \
0                      You Can Smell Hillary’s Fear
1  Watch The Exact Moment Paul Ryan Committed Pol…
2        Kerry to go to Paris in gesture of sympathy
3  Bernie supporters on Twitter erupt in anger ag…
4   The Battle of New York: Why This Primary Matters

                                           text label  \
0  Daniel Greenfield, a Shillman Journalism Fello…  FAKE
1  Google Pinterest Digg Linkedin Reddit Stumbleu…  FAKE
2  U.S. Secretary of State John F. Kerry said Mon…  REAL
3  - Kaydee King (@KaydeeKing) November 9, 2016 T…  FAKE
4  It's primary day in New York and front-runners…  REAL

                                            New_text
0  [daniel, greenfield, shillman, journalism, fel…
1  [google, pinterest, digg, linkedin, reddit, st…
2  [secretary, state, john, kerry, said, monday, …
3  [kaydee, king, kaydeeking, november, 2016, the…
4  [its, primary, day, new, york, and, frontrunne…
```

[13]:
```
#Remove stopwords.
def remove_stopword(text):
    return[word for word in text if word not in nltk.corpus.stopwords.
 ↪words('english')]
```

[14]:
```
news_df['New_text'] = news_df['New_text'].apply(lambda x: remove_stopword(x))
news_df.head()
```

[14]:
```
                                                title  \
0                       You Can Smell Hillary’s Fear
1  Watch The Exact Moment Paul Ryan Committed Pol…
2        Kerry to go to Paris in gesture of sympathy
3  Bernie supporters on Twitter erupt in anger ag…
4   The Battle of New York: Why This Primary Matters

                                                text label  \
0  Daniel Greenfield, a Shillman Journalism Fello…  FAKE
1  Google Pinterest Digg Linkedin Reddit Stumbleu…  FAKE
2  U.S. Secretary of State John F. Kerry said Mon…  REAL
3  - Kaydee King (@KaydeeKing) November 9, 2016 T…  FAKE
4  It's primary day in New York and front-runners…  REAL

                                              New_text
0  [daniel, greenfield, shillman, journalism, fel…
1  [google, pinterest, digg, linkedin, reddit, st…
2  [secretary, state, john, kerry, said, monday, …
3  [kaydee, king, kaydeeking, november, 2016, les…
4  [primary, day, new, york, frontrunners, hillar…
```

[15]:
```
#Lemmetization
def lemmatizer(text):
    word_net = WordNetLemmatizer()
    return [word_net.lemmatize(word) for word in text]
```

[16]:
```
news_df['New_text'] = news_df['New_text'].apply(lambda x: lemmatizer(x))
news_df.head(10)
```

[16]:
```
                                                title  \
0                       You Can Smell Hillary’s Fear
1  Watch The Exact Moment Paul Ryan Committed Pol…
2        Kerry to go to Paris in gesture of sympathy
3  Bernie supporters on Twitter erupt in anger ag…
4   The Battle of New York: Why This Primary Matters
5                                         Tehran, USA
6  Girl Horrified At What She Watches Boyfriend D…
7                  ‘Britain’s Schindler’ Dies at 106
8  Fact check: Trump and Clinton at the 'commande…
9  Iran reportedly makes new push for uranium con…

                                                text label  \
0  Daniel Greenfield, a Shillman Journalism Fello…  FAKE
1  Google Pinterest Digg Linkedin Reddit Stumbleu…  FAKE
2  U.S. Secretary of State John F. Kerry said Mon…  REAL
3  - Kaydee King (@KaydeeKing) November 9, 2016 T…  FAKE
4  It's primary day in New York and front-runners…  REAL
```

```
5    \nI'm not an immigrant, but my grandparents …  FAKE
6  Share This Baylee Luciani (left), Screenshot o…  FAKE
7  A Czech stockbroker who saved more than 650 Je…  REAL
8  Hillary Clinton and Donald Trump made some ina…  REAL
9  Iranian negotiators reportedly have made a las…  REAL

                                            New_text
0  [daniel, greenfield, shillman, journalism, fel…
1  [google, pinterest, digg, linkedin, reddit, st…
2  [secretary, state, john, kerry, said, monday, …
3  [kaydee, king, kaydeeking, november, 2016, les…
4  [primary, day, new, york, frontrunners, hillar…
5  [i'm, immigrant, grandparent, year, ago, arriv…
6  [share, baylee, luciani, left, screenshot, bay…
7  [czech, stockbroker, saved, 650, jewish, child…
8  [hillary, clinton, donald, trump, made, inaccu…
9  [iranian, negotiator, reportedly, made, lastdi…
```

[17]:
```
# Create sentences to get clean text as input for vectors
def return_sentances(tokens):
    return " ".join([word for word in tokens])
```

[18]:
```
news_df['New_text'] = news_df['New_text'].apply(lambda x: return_sentances(x))
news_df.head()
```

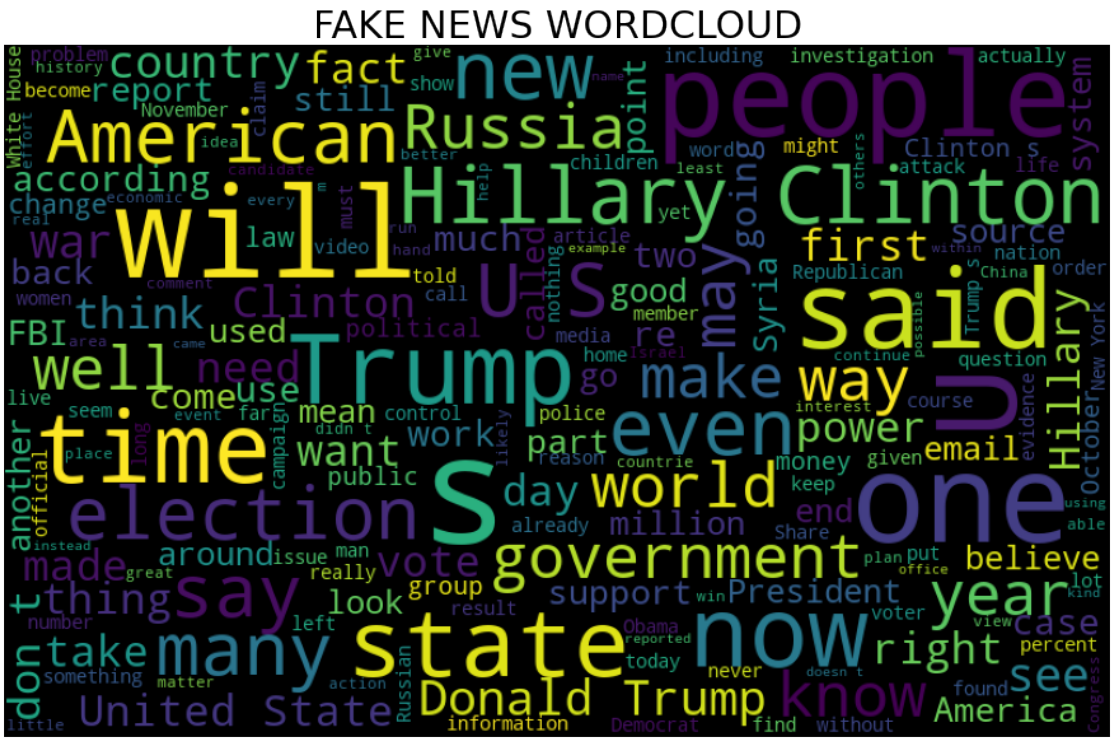
[18]:
```
                                          title  \
0                      You Can Smell Hillary's Fear
1  Watch The Exact Moment Paul Ryan Committed Pol…
2       Kerry to go to Paris in gesture of sympathy
3  Bernie supporters on Twitter erupt in anger ag…
4   The Battle of New York: Why This Primary Matters

                                           text label  \
0  Daniel Greenfield, a Shillman Journalism Fello…  FAKE
1  Google Pinterest Digg Linkedin Reddit Stumbleu…  FAKE
2  U.S. Secretary of State John F. Kerry said Mon…  REAL
3  - Kaydee King (@KaydeeKing) November 9, 2016 T…  FAKE
4  It's primary day in New York and front-runners…  REAL

                                            New_text
0  daniel greenfield shillman journalism fellow f…
1  google pinterest digg linkedin reddit stumbleu…
2  secretary state john kerry said monday stop pa…
3  kaydee king kaydeeking november 2016 lesson to…
4  primary day new york frontrunners hillary clin…
```

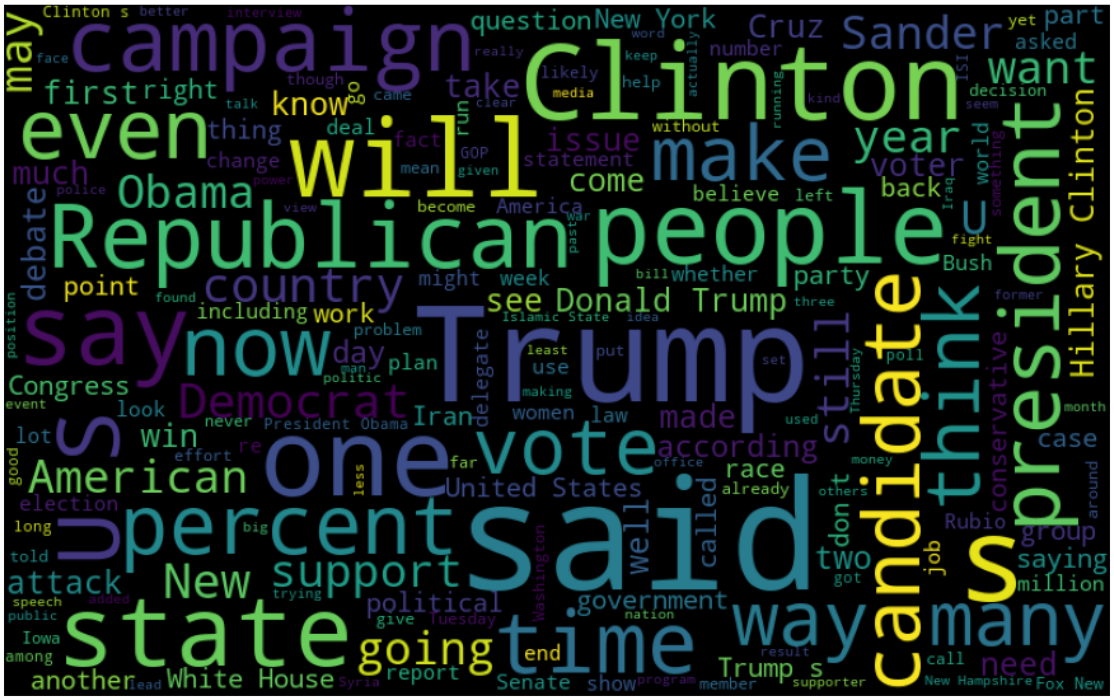
2.0.1 Word Cloud

[19]:
```
# Create and generate a word cloud image FAKE_NEWS
fake_data = news_df[news_df["label"] == "FAKE"]
fake_text = ' '.join([text for text in fake_data.text])
wordcloud = WordCloud(width=800, height=500, random_state=21,␣
 ↪max_font_size=110).generate(fake_text)
plt.figure(figsize= [20,10])
plt.imshow(wordcloud)
plt.axis("off")
plt.title('FAKE NEWS WORDCLOUD',fontsize= 30)
plt.show()
```



[20]:
```
# Create and generate a word cloud image REAL_NEWS
real_data = news_df[news_df["label"] == "REAL"]
real_text = ' '.join([text for text in real_data.text])
wordcloud = WordCloud(width=800, height=500, random_state=21,␣
 ↪max_font_size=110).generate(real_text)
plt.figure(figsize= [20,10])
plt.imshow(wordcloud)
plt.axis("off")
plt.title('REAL NEWS WORDCLOUD',fontsize= 30)
plt.show()
```

REAL NEWS WORDCLOUD

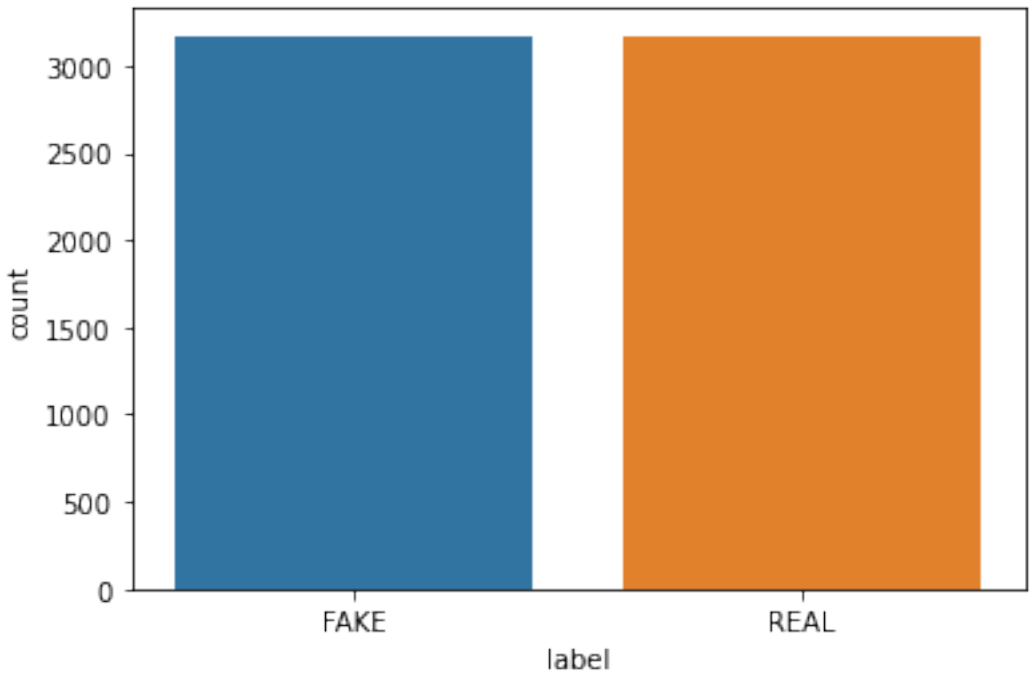


[21]:
```
#Comparing the frequency of Fake and Real News.
print(news_df.groupby(['label'])['text'].count())
sns.countplot(news_df['label'])
```

```
label
FAKE    3164
REAL    3171
Name: text, dtype: int64
```

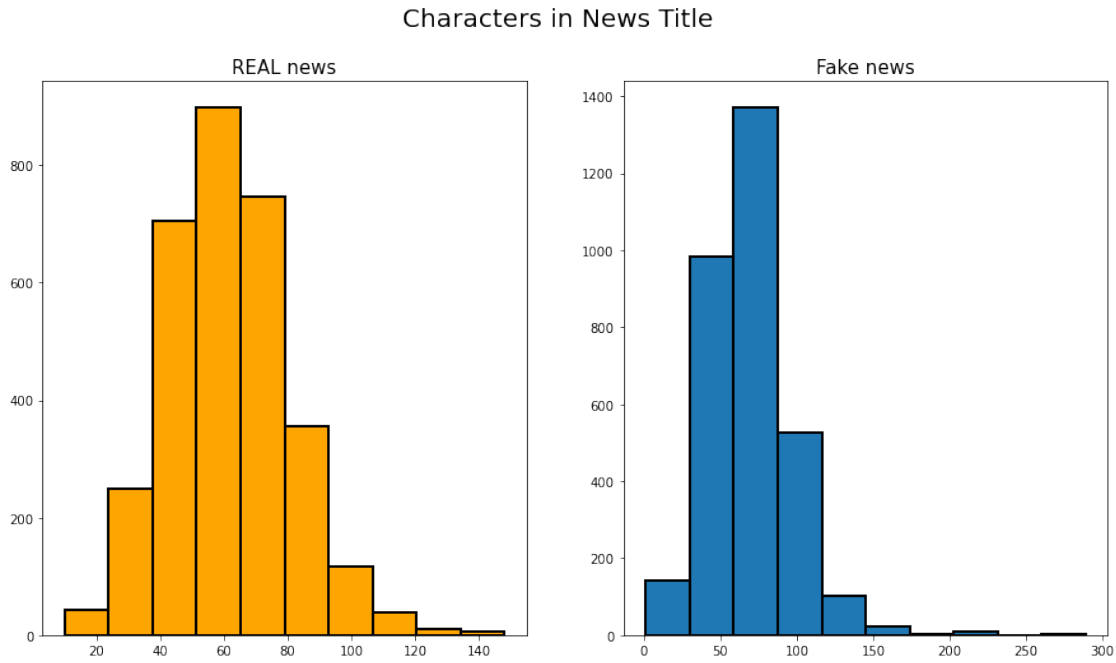
```
C:\Users\dpend\Anaconda3\lib\site-packages\seaborn\_decorators.py:36:
FutureWarning: Pass the following variable as a keyword arg: x. From version
0.12, the only valid positional argument will be `data`, and passing other
arguments without an explicit keyword will result in an error or
misinterpretation.
  warnings.warn(
```

[21]: <AxesSubplot:xlabel='label', ylabel='count'>



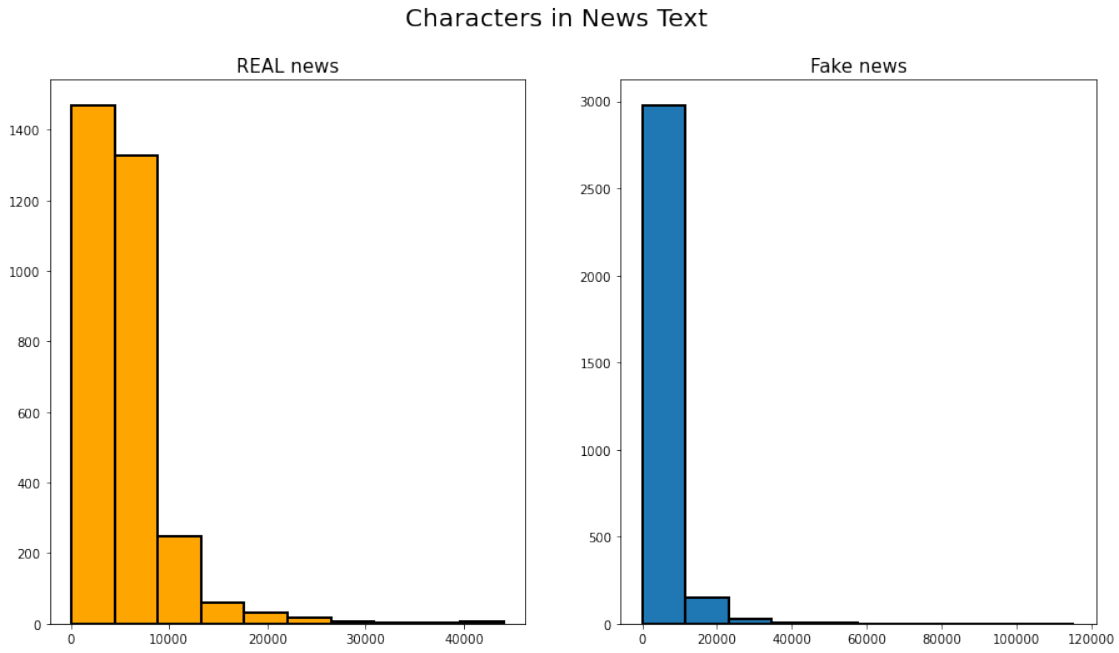
[22]:
```
#Comparing the Total numbers of Characters in the Feature Title.
fig, (ax1,ax2)=plt.subplots(1,2,figsize=(15,8))
fig.suptitle('Characters in News Title',fontsize=20)
news_len=news_df[news_df['label']=='REAL']['title'].str.len()
ax1.hist(news_len,color='orange',linewidth=2,edgecolor='black')
ax1.set_title('REAL news',fontsize=15)
news_len=news_df[news_df['label']=='FAKE']['title'].str.len()
ax2.hist(news_len,linewidth=2,edgecolor='black')
ax2.set_title('Fake news',fontsize=15)
```

[22]:
```
Text(0.5, 1.0, 'Fake news')
```



[23]:
```
#Comparing the Total numbers of Characters in the Feature Text.
fig, (ax1,ax2)=plt.subplots(1,2,figsize=(15,8))
fig.suptitle('Characters in News Text',fontsize=20)
news_len=news_df[news_df['label']=='REAL']['text'].str.len()
ax1.hist(news_len,color='orange',linewidth=2,edgecolor='black')
ax1.set_title('REAL news',fontsize=15)
news_len=news_df[news_df['label']=='FAKE']['text'].str.len()
ax2.hist(news_len,linewidth=2,edgecolor='black')
ax2.set_title('Fake news',fontsize=15)
```

[23]:
```
Text(0.5, 1.0, 'Fake news')
```



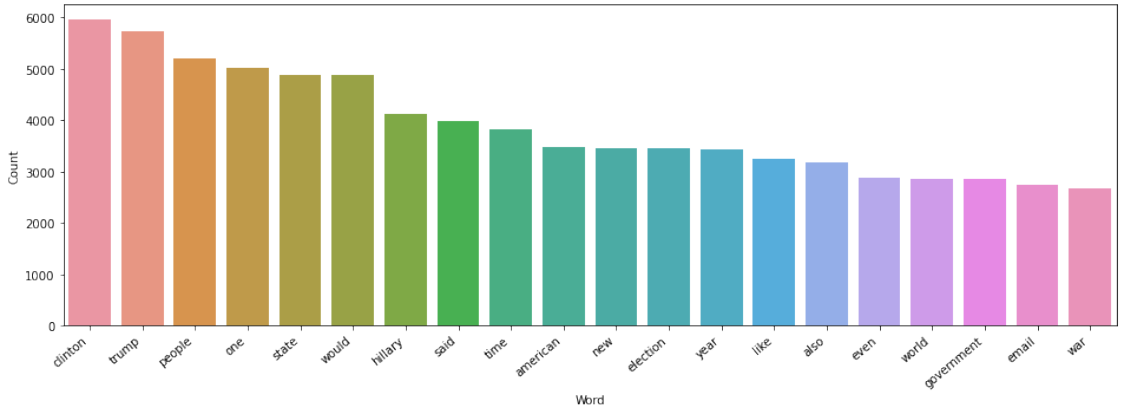
[24]:
```
#creating a bag of words with the consecutive frequency for fake text.
fake_text_vis =' '.join([str(x) for x in␣
 ↪news_df[news_df['label']=='FAKE']['New_text']])
a = nltk.FreqDist(fake_text_vis.split())
d = pd.DataFrame({'Word': list(a.keys()),
                  'Count': list(a.values())})
d.sample(10)
```

[24]:

	Word	Count
21971	‘justify’	3
72431	endorsement39	1
42102	rasmussenpoll	1
28956	speechless	5
6351	exploding	11
3984	starting	110
39822	dhabi	3
3591	dummy	8
18586	gutted	8
44918	"hillary’s	5

[25]:
```
# selecting top 20 most frequent hashtags.
d = d.nlargest(columns="Count", n = 20)
plt.figure(figsize=(16,5))
ax = sns.barplot(data=d, x= "Word", y = "Count")
ax.set_xticklabels(d["Word"], rotation=40, ha="right")
ax.set(ylabel = 'Count')
```

```
plt.show()
```

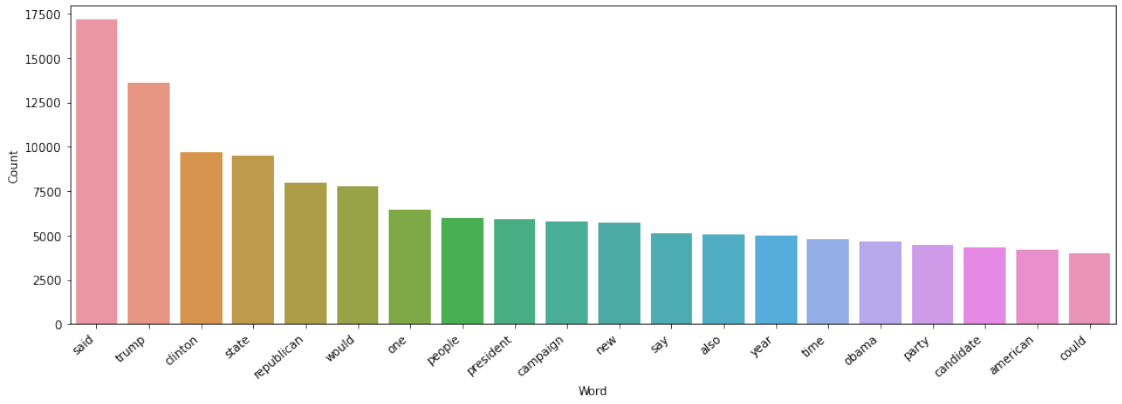


[26]:
```
#creating a bag of words with the consecutive frequency for Real text.
real_text_vis =' '.join([str(x) for x in␣
 ↪news_df[news_df['label']=='REAL']['New_text']])
a = nltk.FreqDist(real_text_vis.split())
d = pd.DataFrame({'Word': list(a.keys()),
                  'Count': list(a.values())})
d.sample(10)
```

[26]:

	Word	Count
43218	diffused	2
876	"not	95
42157	pagan	2
12090	"nightmare	1
32458	westfield	3
45867	dalaf	1
11787	sang	12
26445	1914	4
48127	retreating"	1
40306	nearrecord	2

[27]:
```
d = d.nlargest(columns="Count", n = 20)
plt.figure(figsize=(16,5))
ax = sns.barplot(data=d, x= "Word", y = "Count")
ax.set_xticklabels(d["Word"], rotation=40, ha="right")
ax.set(ylabel = 'Count')
plt.show()
```



[28]:
```
#Label Encoding.
news_df["label"]=news_df["label"].replace(["FAKE","REAL"],value=[1,0])
```

[29]:
```
news_df.head()
```

[29]:
```
                                               title  \
0                       You Can Smell Hillary's Fear
1  Watch The Exact Moment Paul Ryan Committed Pol…
2        Kerry to go to Paris in gesture of sympathy
3  Bernie supporters on Twitter erupt in anger ag…
4   The Battle of New York: Why This Primary Matters

                                                text  label  \
0  Daniel Greenfield, a Shillman Journalism Fello…      1
1  Google Pinterest Digg Linkedin Reddit Stumbleu…      1
2  U.S. Secretary of State John F. Kerry said Mon…      0
3  - Kaydee King (@KaydeeKing) November 9, 2016 T…      1
4  It's primary day in New York and front-runners…      0

                                            New_text
0  daniel greenfield shillman journalism fellow f…
1  google pinterest digg linkedin reddit stumbleu…
2  secretary state john kerry said monday stop pa…
3  kaydee king kaydeeking november 2016 lesson to…
4  primary day new york frontrunners hillary clin…
```

[30]:
```
#Splitting data into test and train.
X_train,X_test,y_train,y_test =␣
↪train_test_split(news_df['New_text'],news_df['label'],test_size=0.2,␣
↪random_state = 10)
```

[31]:
```
#TF-IDF : Term Frequency - Inverse Document Frequency
tfidf = TfidfVectorizer()
X_train = tfidf.fit_transform(X_train)
X_test = tfidf.transform(X_test)
print(X_train.shape)
print(X_test.shape)
```

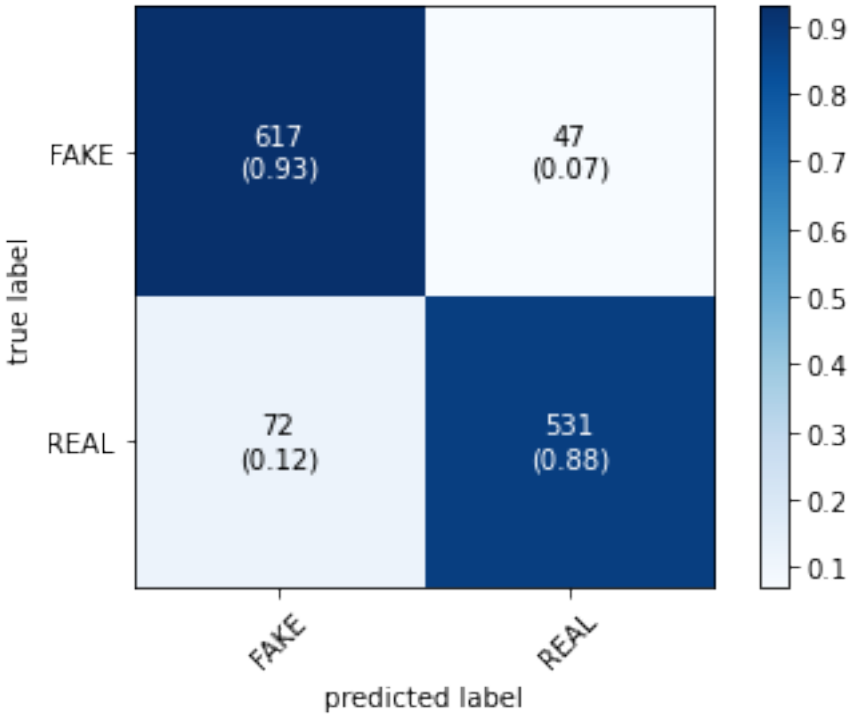
```
(5068, 70803)
(1267, 70803)
```

3 Radndom Forest Classifier.

[32]:
```
model1= RandomForestClassifier()
model1.fit(X_train,y_train)
pred1 = model1.predict(X_test)
accuracy1 = accuracy_score(y_test,pred1)
cm1 = confusion_matrix(y_test,pred1)
print("Accuracy score : {}".format(accuracy1))
plot_confusion_matrix(conf_mat=cm1,show_absolute=True,
                                  show_normed=True,
                                  colorbar=True,class_names=['FAKE','REAL'])
```

```
Accuracy score : 0.9060773480662984
```

[32]:
```
(<Figure size 432x288 with 2 Axes>,
 <AxesSubplot:xlabel='predicted label', ylabel='true label'>)
```

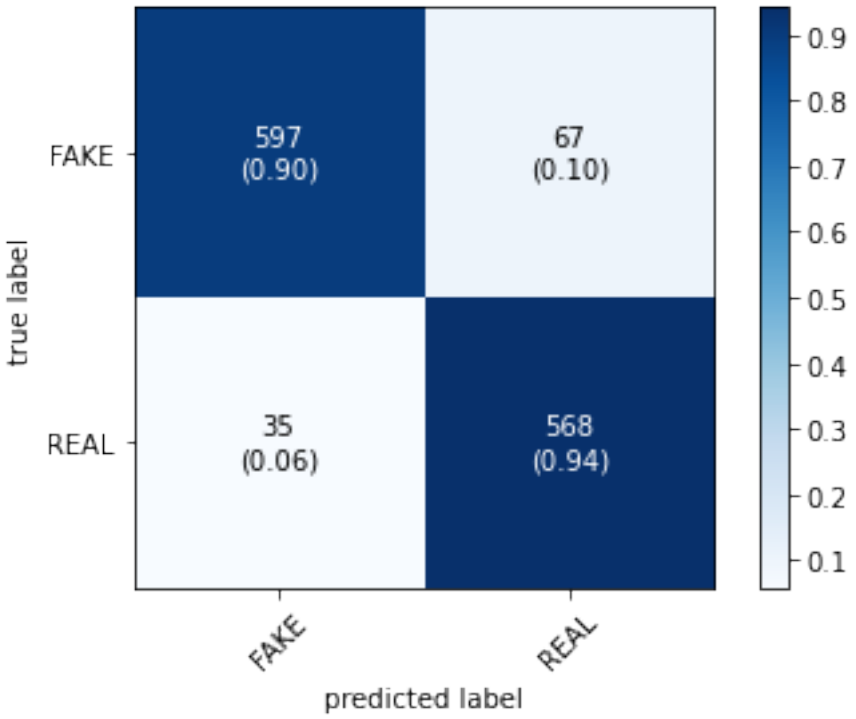


4 Logistic Regression.

[33]:
```
model2 = LogisticRegression(max_iter = 500)
model2.fit(X_train, y_train)
pred2 = model2.predict(X_test)
accuracy2 = accuracy_score(y_test,pred2)
cm2 = confusion_matrix(y_test,pred2)
print("Accuracy score : {}".format(accuracy2))
plot_confusion_matrix(conf_mat=cm2,show_absolute=True,
                                  show_normed=True,
                                  colorbar=True,class_names=['FAKE','REAL'])
```

```
Accuracy score : 0.9194948697711128
```

[33]:
```
(<Figure size 432x288 with 2 Axes>,
 <AxesSubplot:xlabel='predicted label', ylabel='true label'>)
```

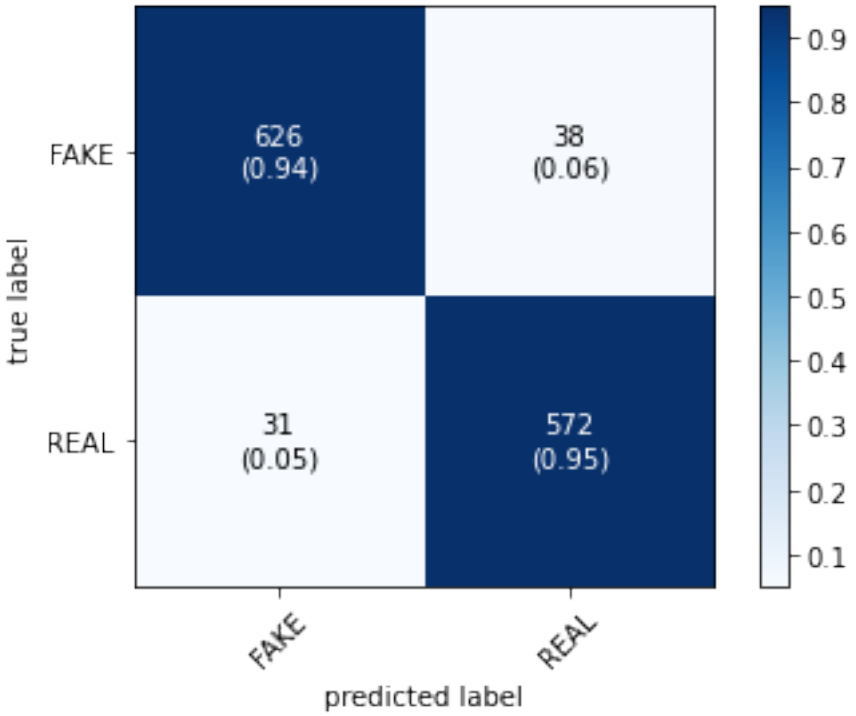


5 Passive Aggressive Classifier

[34]:
```
model3 = PassiveAggressiveClassifier(max_iter=50)
model3.fit(X_train,y_train)
pred3 = model3.predict(X_test)
accuracy3 = accuracy_score(y_test,pred3)
cm3 = confusion_matrix(y_test,pred3)
print("Accuracy score : {}".format(accuracy3))
plot_confusion_matrix(conf_mat=cm3,show_absolute=True,
                                  show_normed=True,
                                  colorbar=True,class_names=['FAKE','REAL'])
```

```
Accuracy score : 0.9455406471981057
```

[34]:
```
(<Figure size 432x288 with 2 Axes>,
 <AxesSubplot:xlabel='predicted label', ylabel='true label'>)
```



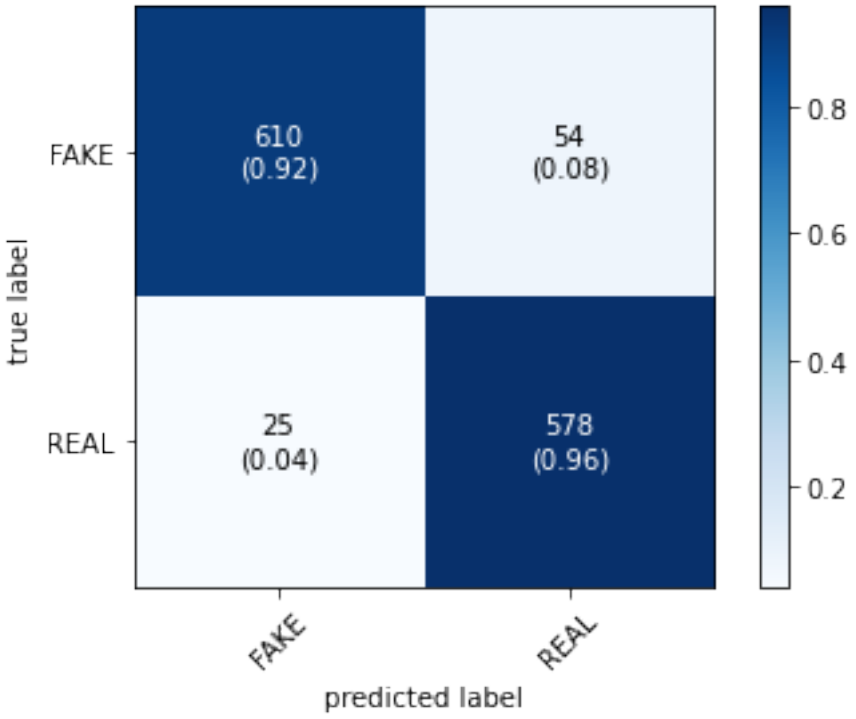
6 Support Vector Classification.

[35]:
```
model4=SVC()
model4.fit(X_train,y_train)
pred4 = model4.predict(X_test)
accuracy4 = accuracy_score(y_test,pred4)
```

```
cm4 = confusion_matrix(y_test,pred4)
print("Accuracy score : {}".format(accuracy4))
plot_confusion_matrix(conf_mat=cm4,show_absolute=True,
                                  show_normed=True,
                                  colorbar=True,class_names=['FAKE','REAL'])
```

```
Accuracy score : 0.9376479873717443
```

[35]:
```
(<Figure size 432x288 with 2 Axes>,
 <AxesSubplot:xlabel='predicted label', ylabel='true label'>)
```

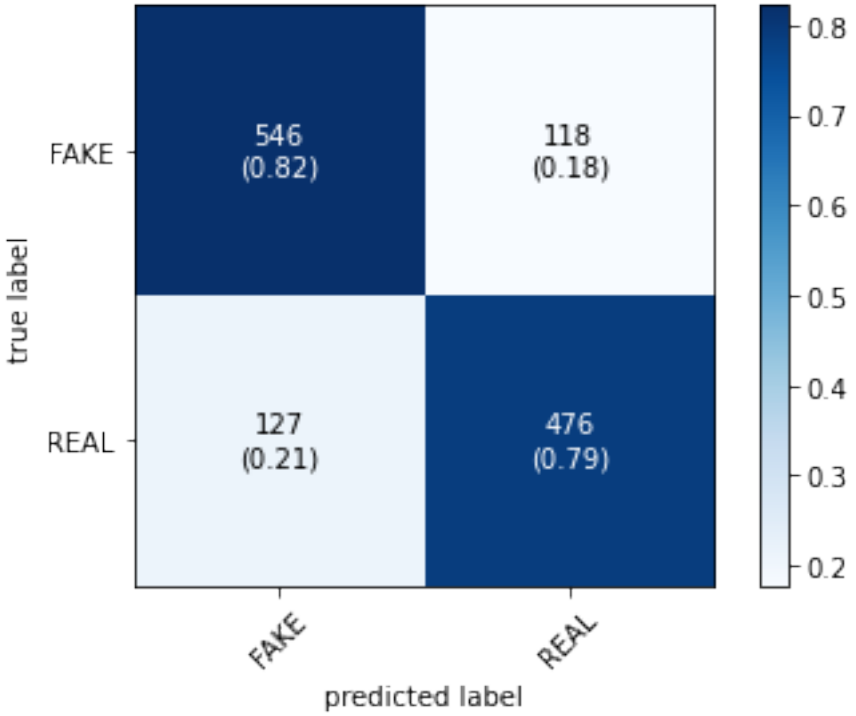


7 Decision Tree Classifier

[36]:
```
model5=DecisionTreeClassifier()
model5.fit(X_train,y_train)
pred5 = model5.predict(X_test)
accuracy5 = accuracy_score(y_test,pred5)
cm5 = confusion_matrix(y_test,pred5)
print("Accuracy score : {}".format(accuracy5))
plot_confusion_matrix(conf_mat=cm5,show_absolute=True,
                                  show_normed=True,
                                  colorbar=True,class_names=['FAKE','REAL'])
```

```
Accuracy score : 0.8066298342541437
```

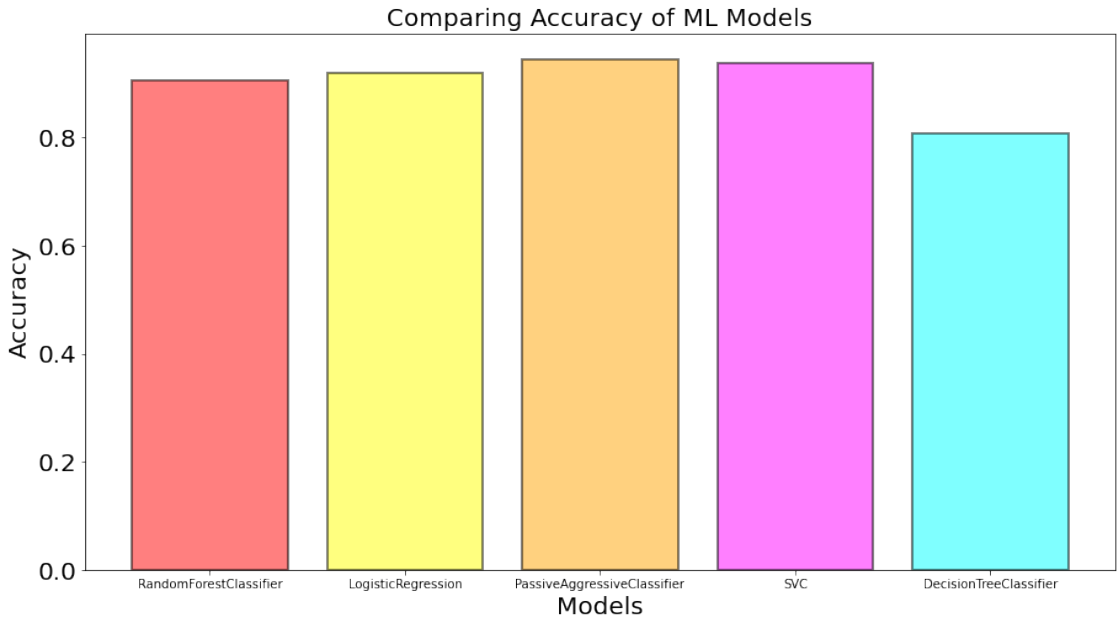
[36]:
```
(<Figure size 432x288 with 2 Axes>,
 <AxesSubplot:xlabel='predicted label', ylabel='true label'>)
```



[37]:
```
#Creating the Dictionary with model name as key and accuracy as key-value
labels={'RandomForestClassifier':accuracy1,'LogisticRegression':
 ↪accuracy2,'PassiveAggressiveClassifier':accuracy3,
        'SVC':accuracy4,'DecisionTreeClassifier':accuracy5}
```

[38]:
```
#Plotting accuracy of all the models with Bar-Graphs
plt.figure(figsize=(15,8))
plt.title('Comparing Accuracy of ML Models',fontsize=20)
colors=['red','yellow','orange','magenta','cyan']
plt.xticks(fontsize=10,color='black')
plt.yticks(fontsize=20,color='black')
plt.ylabel('Accuracy',fontsize=20)
plt.xlabel('Models',fontsize=20)
plt.bar(labels.keys(),labels.values(),edgecolor='black',color=colors,␣
 ↪linewidth=2,alpha=0.5)
```

[38]:
```
<BarContainer object of 5 artists>
```


Comparing Accuracy of ML Models
Accuracy
0.0
0.2
0.4
0.6
0.8
RandomForestClassifier
LogisticRegression
PassiveAggressiveClassifier
SVC
DecisionTreeClassifier
Models

[ ]: