NAME OF THE PROJECT

Fake News Prediction Project

Submitted by:

P. Sai Deekshith Reddy

ACKNOWLEDGMENT

This includes mentioning of all the references, research papers, data sources, professionals and other resources that helped you and guided you in completion of the project.

INTRODUCTION

• Business Problem Framing

Fake news has become one of the biggest problems of our age. It has serious impact on our online as well as offline discourse. One can even go as far as saying that, to date, fake news poses a clear and present danger to western democracy and stability of the society.

• Motivation for the Problem Undertaken

In the era of news in our lives, it is the people's responsibility to not to share any misleading information as there are many sources available now-a-days. The fraud news such as spam messages, funding news or any false information to be fall out or reach to the people we consider it as a serious issue although it is extremely complicated to find out which is fraud and which is not a fraud profile or users in social media, they replicate the information as the original one. As the technology evolved and the machine intelligence has come into existence everyone tends to use available sources for creating and dissemination of fraud news. People who are illiterate might be new to digital media as they are inexperienced, so they are the ones who believe that fraud news easily and makes it practical in their lives. To a minimum, we have deviled a simple web application which statistically detects false information, and also real news.

Analytical Problem Framing

- **Mathematical/ Analytical Modeling of the Problem**

1. Naive Bayes
2. Logistic Regression
3. Decision tree
4. Random Forest

- **Data Sources and their formats**

Data source from google, data in csv formate

- **Data Preprocessing Done**

we are cleaning our text by steaming, lemmatization, remove stopwords, remove special symbols and numbers, etc. After cleaning the data we have to feed this text data into a vectorizer which will convert this text data into numerical features.

- **Hardware and Software Requirements and Tools**

UsedHardware -laptop**, keyboard, mouse**
Software – Jupiter Notbook
Tools - Scikit-learn,scipy.stats
library – using python library – numpy , pandas, matplotlib,sns,

machine algorithm – Naïve bayes,logistic regression, decision tree classifier, random forest classifier

Model/s Development and Evaluation

- Testing of Identified Approaches (Algorithms)

We used algorithms for training and testing:

Naïve bayes

Logistic regression

Decision tree

Random forest

Classification report

Confusion matrix

Decision tree has given the bset accuracy score

- Run and Evaluate selected models

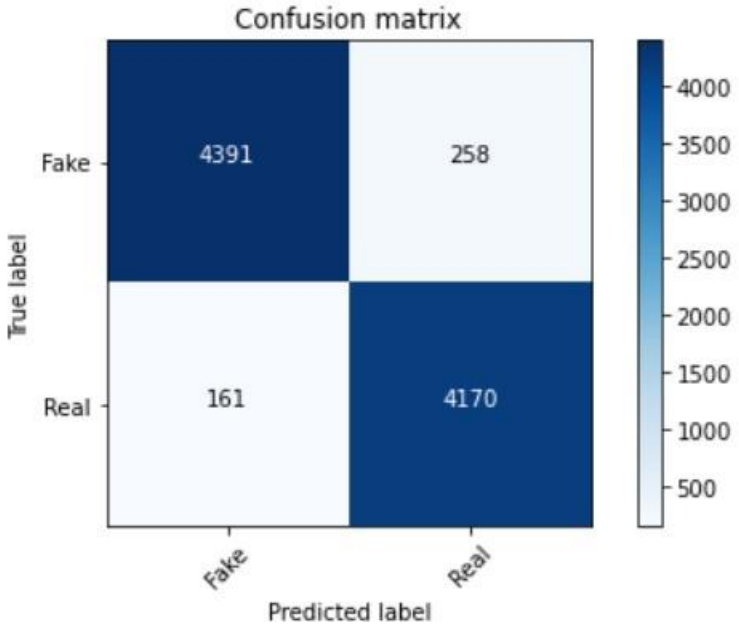
Naive Bayes

```
dct = dict()

from sklearn.naive_bayes import MultinomialNB

NB_classifier = MultinomialNB()
pipe = Pipeline([('vect', CountVectorizer()),
                 ('tfidf', TfidfTransformer()),
                 ('model', NB_classifier)])

model = pipe.fit(X_train, y_train)
prediction = model.predict(X_test)
print("accuracy: {}%".format(round(accuracy_score(y_test, prediction)*100,2)))

dct['Naive Bayes'] = round(accuracy_score(y_test, prediction)*100,2)
```

```
accuracy: 95.33%
```

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

```
Confusion matrix, without normalization
```



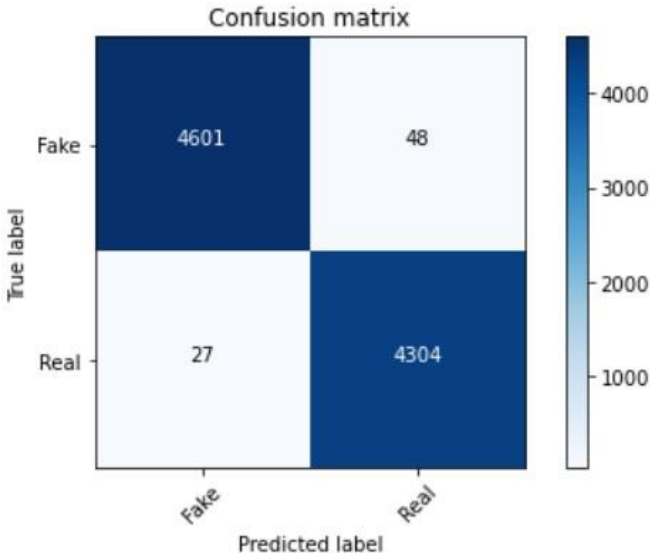
Logistic regression

In [35]:
```
# Logistic regression
from sklearn.linear_model import LogisticRegression

pipe = Pipeline([('vect', CountVectorizer()),
                 ('tfidf', TfidfTransformer()),
                 ('model', LogisticRegression())])

# Fitting the model
model = pipe.fit(X_train, y_train)

# Accuracy
prediction = model.predict(X_test)
print("accuracy: {}%".format(round(accuracy_score(y_test, prediction)*100,2)))
dct['Logistic Regression'] = round(accuracy_score(y_test, prediction)*100,2)
```

```
accuracy: 99.16%
```

In [36]:
```
cm = metrics.confusion_matrix(y_test, prediction)
plot_confusion_matrix(cm, classes=['Fake', 'Real'])
```

```
Confusion matrix, without normalization
```



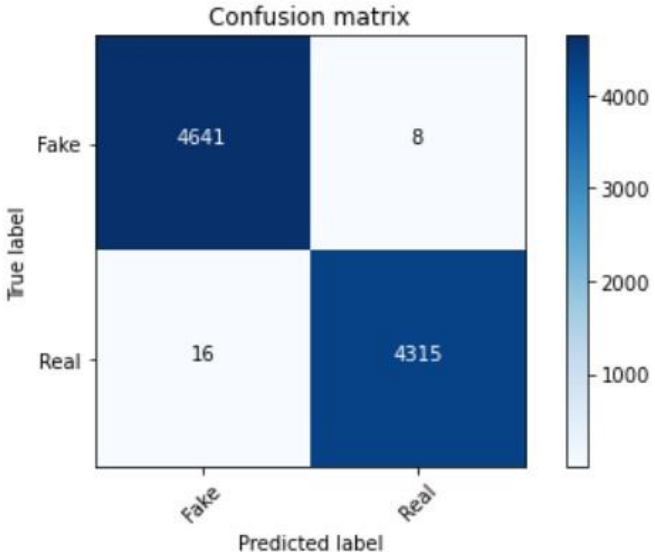
Decision Tree

In [37]:
```
from sklearn.tree import DecisionTreeClassifier

# Vectorizing and applying TF-IDF
pipe = Pipeline([('vect', CountVectorizer()),
                 ('tfidf', TfidfTransformer()),
                 ('model', DecisionTreeClassifier(criterion= 'entropy',
                                                  max_depth = 20,
                                                  splitter='best',
                                                  random_state=42))])
# Fitting the model
model = pipe.fit(X_train, y_train)

# Accuracy
prediction = model.predict(X_test)
print("accuracy: {}%".format(round(accuracy_score(y_test, prediction)*100,2)))
dct['Decision Tree'] = round(accuracy_score(y_test, prediction)*100,2)
```

```
accuracy: 99.73%
```

In [38]:
```
cm = metrics.confusion_matrix(y_test, prediction)
plot_confusion_matrix(cm, classes=['Fake', 'Real'])
```

```
Confusion matrix, without normalization
```



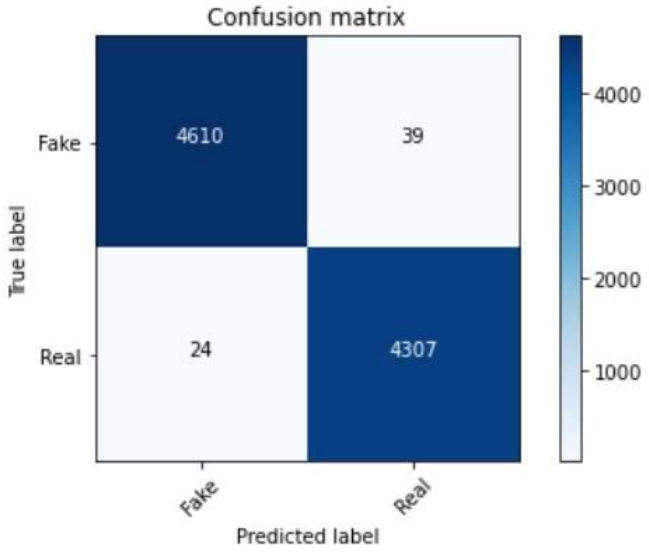
Random Forest

In [39]:
```
from sklearn.ensemble import RandomForestClassifier

pipe = Pipeline([('vect', CountVectorizer()),
                 ('tfidf', TfidfTransformer()),
                 ('model', RandomForestClassifier(n_estimators=50, criterion="entropy"))])

model = pipe.fit(X_train, y_train)
prediction = model.predict(X_test)
print("accuracy: {}%".format(round(accuracy_score(y_test, prediction)*100,2)))
dct['Random Forest'] = round(accuracy_score(y_test, prediction)*100,2)
```

```
accuracy: 99.3%
```

In [40]:
```
cm = metrics.confusion_matrix(y_test, prediction)
plot_confusion_matrix(cm, classes=['Fake', 'Real'])
```

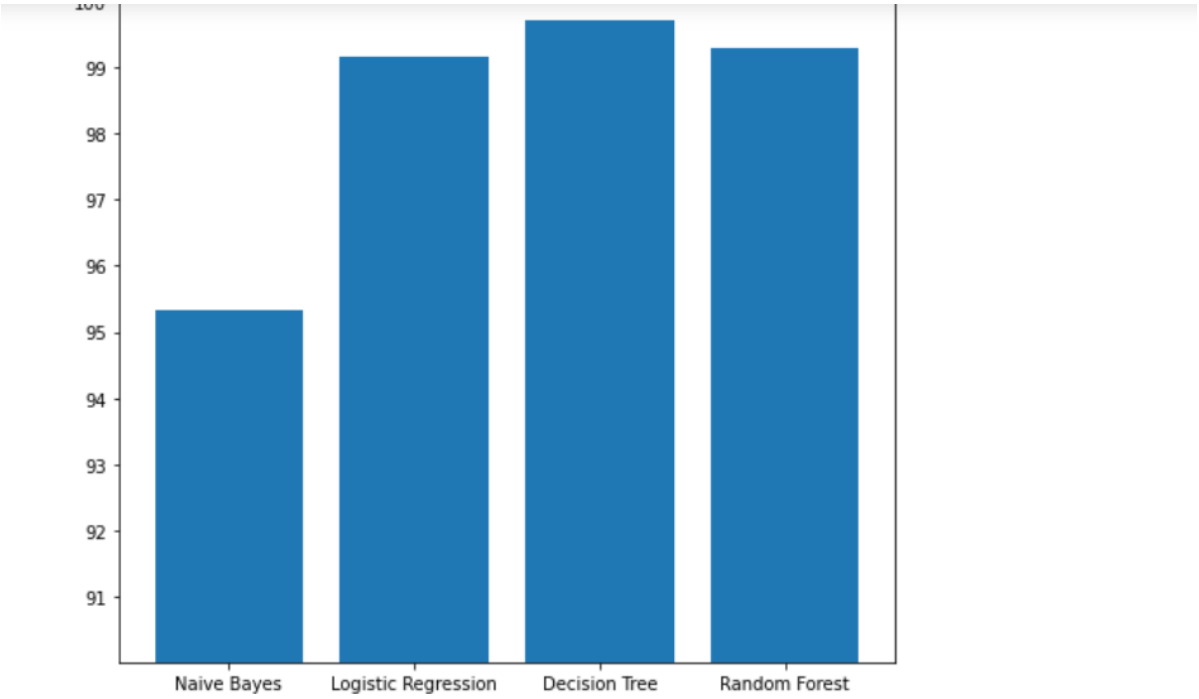
```
Confusion matrix, without normalization
```



Comparing Different Models

In [41]:
```
import matplotlib.pyplot as plt
plt.figure(figsize=(8,7))
plt.bar(list(dct.keys()),list(dct.values()))
plt.ylim(90,100)
plt.yticks((91, 92, 93, 94, 95, 96, 97, 98, 99, 100))
```

Out[41]:
```
([<matplotlib.axis.YTick at 0x2ac0a8390d0>,
  <matplotlib.axis.YTick at 0x2ac0bf14910>,
  <matplotlib.axis.YTick at 0x2ac0bf2d7c0>,
  <matplotlib.axis.YTick at 0x2ac0a902340>,
  <matplotlib.axis.YTick at 0x2ac0a902a90>,
  <matplotlib.axis.YTick at 0x2ac0a902ca0>,
  <matplotlib.axis.YTick at 0x2ac0a8f02e0>,
  <matplotlib.axis.YTick at 0x2ac0a8f0a30>,
  <matplotlib.axis.YTick at 0x2ac0a8ef1c0>,
  <matplotlib.axis.YTick at 0x2ac0a8ef910>],
 [Text(0, 0, ''),
  Text(0, 0, ''),
  Text(0, 0, ''),
  Text(0, 0, ''),
  Text(0, 0, ''),
  Text(0, 0, ''),
  Text(0, 0, ''),
  Text(0, 0, ''),
  Text(0, 0, ''),
  Text(0, 0, '')])
```

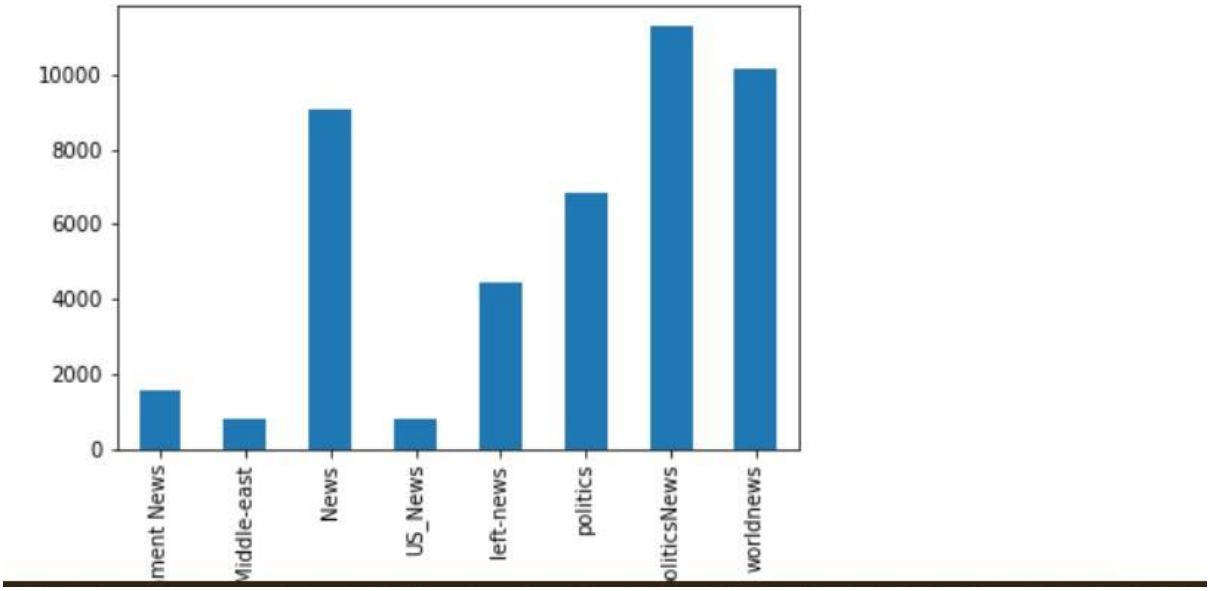


- Visualizations

Count plot

```
# How many articles per subject?
print(data.groupby(['subject'])['text'].count())
data.groupby(['subject'])['text'].count().plot(kind="bar")
plt.show()
```

```
subject
Government News     1570
Middle-east          778
News                9050
US_News              783
left-news           4459
politics            6841
politicsNews       11272
worldnews          10145
Name: text, dtype: int64
```

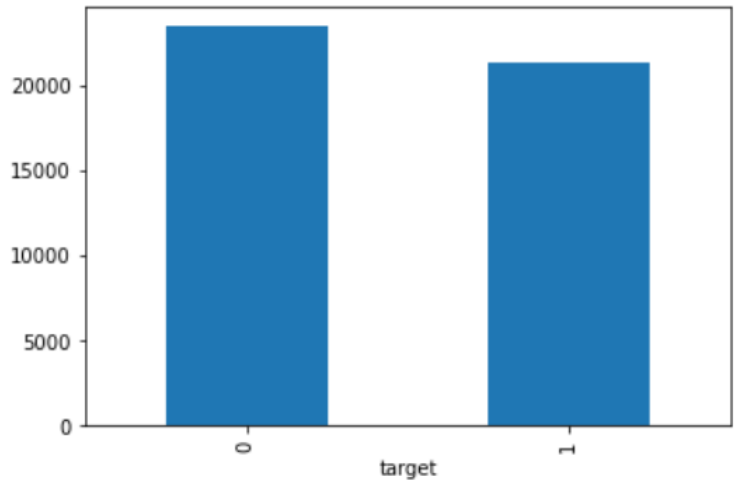


Observation: by this count plot We observed that most articles on politicsNews subject

Count plot:

[25]:
```
# How many fake and real articles?
print(data.groupby(['target'])['text'].count())
data.groupby(['target'])['text'].count().plot(kind="bar")
plt.show()
```

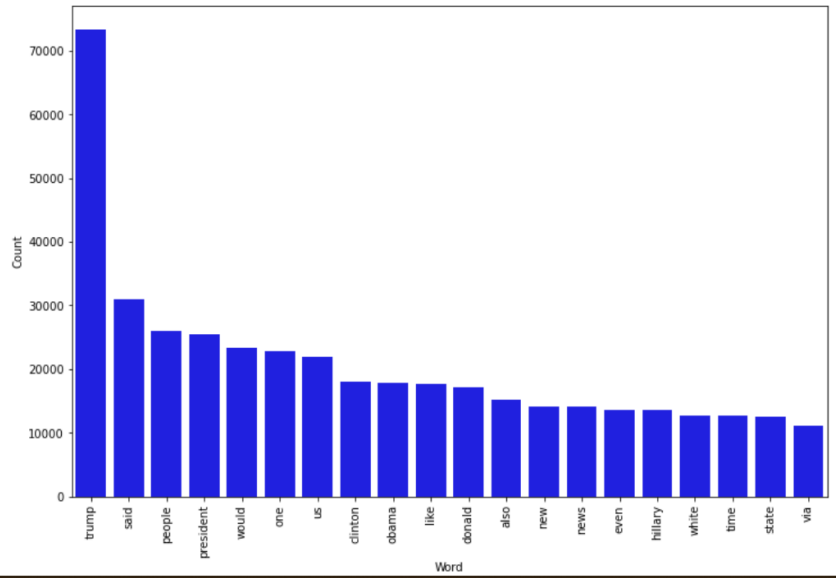
```
target
0    23481
1    21417
Name: text, dtype: int64
```



Observation : by this plot we find that how many fake and real articales in data, so as we have find there are most fake news is available as compare to real news.

Bar plot:

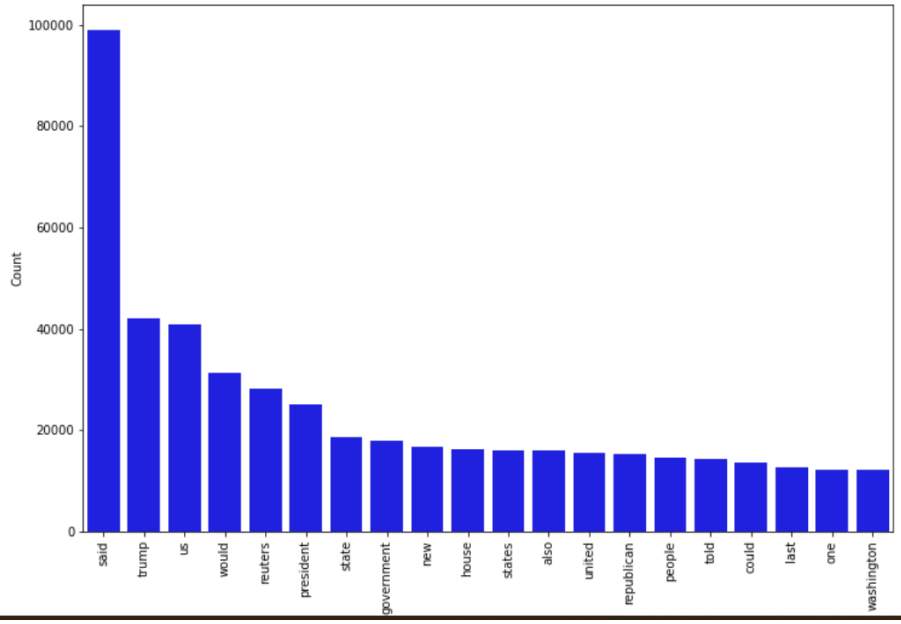
```
In [27]: # Most frequent words in fake news
counter(data[data["target"] == 0], "text", 20)
```



Observation : by this bar plot we find that which is most frequent words uses in this fake articles, so we observed that ‘trump’ word has used many times.

Bar plot:

```
In [28]: # Most frequent words in real news
counter(data[data["target"] == 1], "text", 20)
```



Observation: by this bar plot we find that which is most frequent words uses in this real news, so we observed that ‘said word has used many times.

- Interpretation of the Results

  Give a summary of what results were interpreted from the visualizations, preprocessing and modelling.

  By the visualizations : by count plot We observed that most articles on politicsNews subject,

  how many fake and real articales in data, so as we have find there are most fake news is available as compare to real news.

by this bar plot we find that which is most frequent words uses in this fake and real news, so we observed that ‘trump’ and ‘said’ word has used many times.

Preprocessing : we are cleaning our text by steaming, lemmatization, remove stopwords, remove special symbols and numbers, etc. After cleaning the data we have to feed this text data into a vectorizer which will convert this text data into numerical features.

Modelling – we have use many algorithm like Naïve Bayes, logistic regression, decision tree, svm, random forest.

But we can find the decision tree has given the best accuracy score 99 %

CONCLUSION

- **Key Findings and Conclusions of the Study**

  The model that we developed is not particularly belongs to one media and in our dataset all the data consists of news reports from various digital media that means our model understanding could be applied to any digital media to know what is fraud and what is fraud not.

- ## Learning Outcomes of the Study in respect of DataScience

we are cleaning our text by remove stopwords, remove special symbols and numbers, etc.

we have used 5 machine larning algorithm

best algorithm is decision tree classifer has given the best score around 99%

- ## Limitations of this work and Scope for Future Work

Our model's future work is to develop a dynamic model so that our users can download our app and can easily detect any news and any fake URL's and we are also thinking to develop a model so that it can detect any fake profiles present in any media such as Facebook, Instagram, Stack Overflow and also any fraud reviews for duplicate products. Not only these but there are also many unsettled controversies and topics about celebrity's fraud news and the news articles about the world is always a concern to each and everyone. By considering above matters professionals have to share out with them. For example, recently many frauds news articles about covid vaccine created an immense effect on people. My opinion people must research about news if any new news is in front of us. Now-a-days many news are being forwarded to WhatsApp application about funding and it is especially important to point out the major sources of that news and have knowledge about them and share to people so that everyone can able to understand about a particular news. I think everything is there in people's hand, if we SCRS Conference Proceedings on Intelligent Systems (2021) 67 are concern about any social cause then there will be no spread of Fake news. Proposed BERT outperforms LR and KNN models and accuracy may be improved by using machine learning ensemble methods.