
# Fake News Prediction Project

Submitted by:

Mekhala Misra

# ACKNOWLEDGMENT

I took help from following websites:

1)Geek for geeks

2)Pandas documentation

3)researchgate.net

# 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. It is a NLP problem.

# Analytical Problem Framing

- ## Mathematical/ Analytical Modelling of the Problem

  This is a NLP problem as we have to analyse the given fake news dataset and predict whether the given news is real or fake.

- ## Data Sources and their formats

  There are two datasets are provided, one for fake news and other for true news so we just concatenate the two datasets into one dataset with 44939 rows and 5 columns.

```
In [14]:  1 df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 44940 entries, 0 to 21416
Data columns (total 5 columns):
 #   Column   Non-Null Count  Dtype 
---  ------   --------------  ----- 
 0   title    44919 non-null  object
 1   text     44919 non-null  object
 2   subject  44898 non-null  object
 3   date     44898 non-null  object
 4   label    44940 non-null  int64 
dtypes: int64(1), object(4)
memory usage: 2.1+ MB
```

Following are the columns present and their respective data types:

```
In [15]:  1 df.columns

Out[15]: Index(['title', 'text', 'subject', 'date', 'label'], dtype='object')
```

Here is the glimpse of data:

| | title | text | subject | date | label |
|---|---|---|---|---|---|
| 0 | Donald Trump Sends Out Embarrassing New Year'... | Donald Trump just couldn t wish all Americans ... | News | December 31, 2017 | 0 |
| 1 | Drunk Bragging Trump Staffer Started Russian ... | House Intelligence Committee Chairman Devin Nu... | News | December 31, 2017 | 0 |
| 2 | Sheriff David Clarke Becomes An Internet Joke... | On Friday, it was revealed that former Milwauk... | News | December 30, 2017 | 0 |
| 3 | Trump Is So Obsessed He Even Has Obama's Name... | On Christmas day, Donald Trump announced that ... | News | December 29, 2017 | 0 |
| 4 | Pope Francis Just Called Out Donald Trump Dur... | Pope Francis used his annual Christmas Day mes... | News | December 25, 2017 | 0 |

# • Data Pre-processing

- The dataset has  null values so we dropped those rows with null values after checking the data loss percent.
- Since this is a NLP problem so we have to analyse the text in text column and depending on that we have to predict whether the news is real or fake, so we dropped other three columns i.e. date, subject and title.
- Feature Engineering-We have to fetch important words from the text so in order to do that we first converted entire text feature to lower case.

```
In [28]:  1 #converting all the texts to lower case so thats its easy to analyse them.
          2 df['text']=df['text'].str.lower()
```

- Further we removed all kinds of punctuations from the feature text.

```
In [31]:  1 #Removing punctuations
          2 df['text']=df['text'].apply(lambda x: ' '.join(
          3     term for term in x.split() if term not in string.punctuation))
          4
```

- In order to focus on the words that could differentiate between fake and true news we also have to remove all kind of stop words.

```
In [32]:  1 # Removing stop words
          2 sw = set(stopwords.words('english') + ['u', 'ur', '4', '2', 'im', 'dont', 'doin', 'ure'])
          3 df['text']=df['text'].apply(lambda x: ' '.join(
          4     term for term in x.split() if term not in sw))
```

- In order to further analyse the text using morphological analysis of the words called lemmatization.

In [33]:
```
#word lemmatizer
lm=WordNetLemmatizer()
df['text']=df['text'].apply(lambda x: ' '.join(
 lm.lemmatize(t) for t in x.split()))
```

- Then finally we converted the text into its vector form Term Frequency and Inverse Document Frequency method

## Applying TF-IDF vectorizer

In [38]:
```
#Initialize the `tfidf_vectorizer`
tfidf_vectorizer = TfidfVectorizer()
#Fit and transform the training data
tfidf_train = tfidf_vectorizer.fit_transform(X_train)
#Transform the test set
tfidf_test = tfidf_vectorizer.transform(X_test)
```

- We split the data into training set and testing set using train test split method.
- On this train and test data we applied various models: logistic regression and naïve bayes classifier.
- The best performing model is logistic regression with approximately 99% of accuracy.
- So, we saved our logistic regression model.

# Hardware and Software Requirements and Tools Used

We imported following packages:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn import metrics
from sklearn.metrics import roc_curve,auc,classification_report,confusion_matrix
from sklearn.feature_extraction.text import TfidfVectorizer
import warnings
warnings.filterwarnings('ignore')
from nltk.stem import WordNetLemmatizer
import nltk
from nltk.corpus import  stopwords
import string
from sklearn.naive_bayes import MultinomialNB
from sklearn.metrics import accuracy_score
```

```
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.linear_model import LogisticRegression
```

# Model/s Development and Evaluation

- Identification of possible problem-solving approaches (methods)

- Since this is a NLP classification problem so I used two nlp classification approaches naïve bayes classifier and logistic regressor.

- Testing of Identified Approaches (Algorithms)

- We applied Logistic regression and naïve bayes classifier on the clean data
- Depending on the model accuracy, confusion matrix, auc-roc curve and classification report we opted logistic regression.
- Since logistic regressor is giving best accuracy already so I did not apply hyper parameter tuning on dataset. So, we saved our logistic regressor model.

- Run and Evaluate selected models

1. Naïve Bayes Classifier:

   We applied naïve bayes classifier on the vectorized data and found that this model is predicting correct labels 94% times.

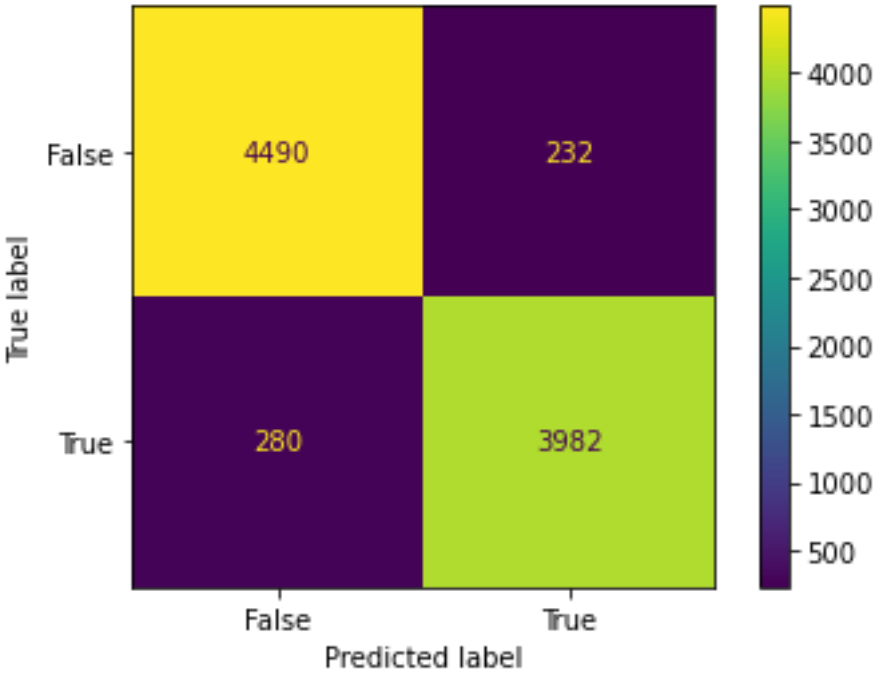
**Naive Bayes Classifier**

In [39]:
```
clf=MultinomialNB()
clf.fit(tfidf_train,y_train)
y_pred=clf.predict(tfidf_test)
print(accuracy_score(y_pred,y_test))
```

```
0.9430097951914514
```

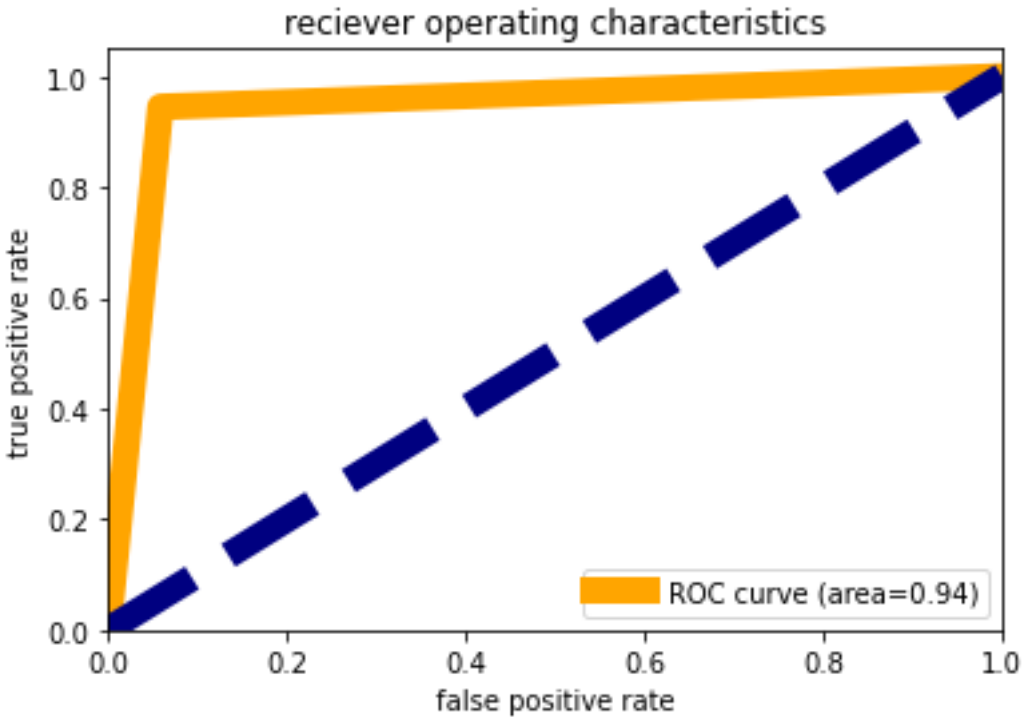
We also checked the performance of this model on various other metrics such as confusion metrics, AUC-ROC curve and classification report.

i) Confusion Matrix




Here we could observe that although the model accuracy is quite good but the false positive and false negative predicted values are quite high so in order to reduce this we applied logistic regressor.

ii) AUC-ROC curve:




Here we could observe that the area present under the curve is 0.94 which is quite a nice coverage and it means that 0.94 is the probability of giving correct prediction.

iii) Classification Report:

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.94 | 0.95 | 0.95 | 4722 |
| 1 | 0.94 | 0.93 | 0.94 | 4262 |
| accuracy | | | 0.94 | 8984 |
| macro avg | 0.94 | 0.94 | 0.94 | 8984 |
| weighted avg | 0.94 | 0.94 | 0.94 | 8984 |

Here we could observe that the model is performing better in predicting label 0 than label 1 as its f1-score is better than that of label 1.

## 2. Logistic Regressor:

We applied logistic regressor classifier on the vectorized data and found that this model is predicting correct labels 99% times.

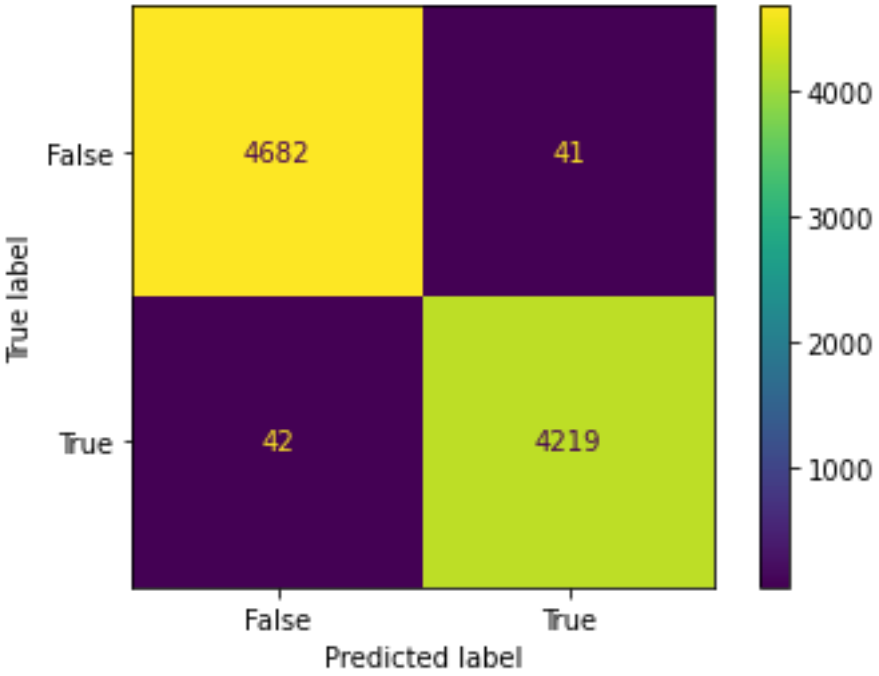
In [26]:
```
lr = LogisticRegression()
lr.fit(tfidf_train, y_train)
pred_tfidf = lr.predict(tfidf_test)

print(accuracy_score(y_test,pred_tfidf))
```

```
0.9907613535173642
```

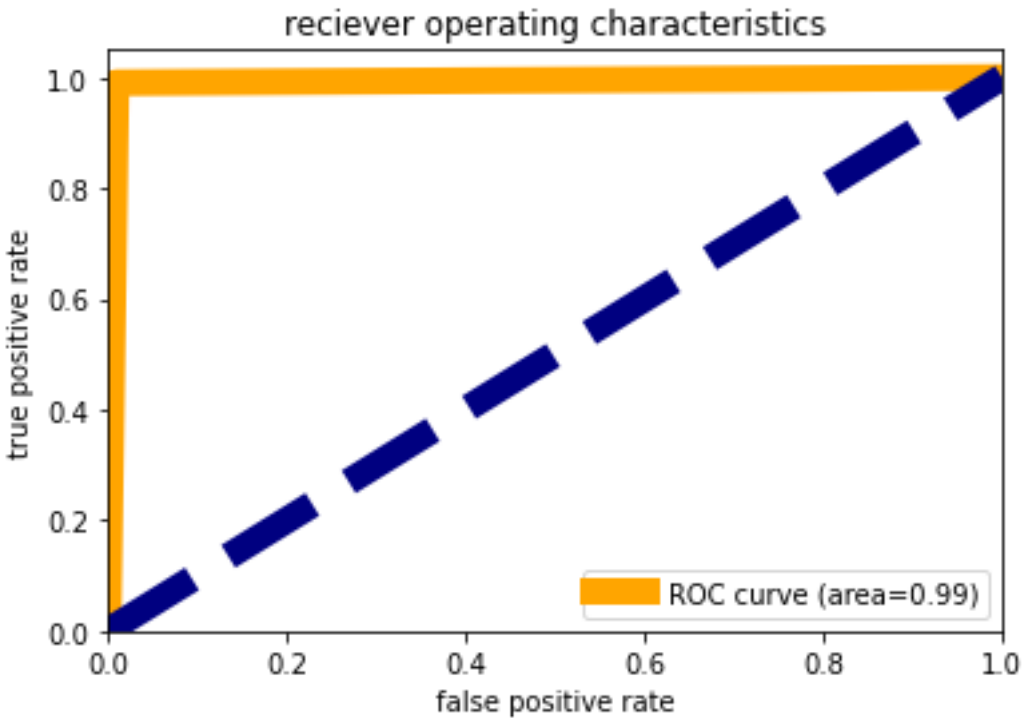
We also checked the performance of this model on various other metrics such as confusion metrics, AUC-ROC curve and classification report.

i) Confusion Matrix:




Here we could observe that the model accuracy is quite good as the false positive and false negative predicted values are very less as compared to true positive and true negative.

ii) AUC-ROC Curve:




Here we could observe that the area present under the curve is 0.99 which is quite a nice coverage and it means that 0.99 is the probability of giving correct prediction.

iii) Classification Report:

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.99 | 0.99 | 0.99 | 4723 |
| 1 | 0.99 | 0.99 | 0.99 | 4261 |
| accuracy | | | 0.99 | 8984 |
| macro avg | 0.99 | 0.99 | 0.99 | 8984 |
| weighted avg | 0.99 | 0.99 | 0.99 | 8984 |

Here we could observe that the model is performing equally well in predicting label 0 than label 1 as its f1-score is same for both.

# Conclusion

In today's time, the majority of the tasks are done online. Newspapers that were earlier preferred as hard- copies are now being substituted by applications like Facebook, Twitter, and news articles to be read online. WhatsApp's forwards are also a major source. The growing problem of fake news only makes things more complicated and tries to change or hamper the opinion and attitude of people towards use of digital technology. When a person is deceived by the real news two possible things happen- People start believing that their perceptions about a particular topic are true as assumed. Thus, in order to curb the phenomenon, we have developed our Fake news Prediction system that takes input from the user and classify it to be true or fake. To implement this, various NLP and Machine Learning Techniques have been used. The model is trained using an appropriate dataset and performance evaluation is also done using various performance measures. The best model, i.e. the model with highest accuracy is used to classify the news and our best performing model is Logistic regressor with accuracy 99%.