```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings("ignore")
```

```
data = pd.read_csv("news.csv")
data.head()
```

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

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 44898 entries, 0 to 44897
Data columns (total 3 columns):
 #   Column  Non-Null Count  Dtype 
---  ------  --------------  ----- 
 0   title   44898 non-null  object
 1   text    44898 non-null  object
 2   label   44898 non-null  object
dtypes: object(3)
memory usage: 1.0+ MB
```

```
data["label"].value_counts()
```

```
fake    23481
true    21417
Name: label, dtype: int64
```

```
# Checking for Null values

data.isnull().sum()
```

```
title    0
text     0
label    0
dtype: int64
```

```
# Removing Blank Spaces

empty_idx = []
for indx,tlt,txt,lbl in data.itertuples():
  if type(tlt)==str:
    if tlt.isspace():
      empty_idx.append(indx)

print(empty_idx)
```

```
[]
```

```
import nltk
from nltk.tokenize import word_tokenize
nltk.download("punkt")

from nltk.corpus import stopwords
nltk.download("stopwords")

from nltk.stem import PorterStemmer,WordNetLemmatizer
nltk.download("wordnet")
```

```
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]   Unzipping tokenizers/punkt.zip.
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Unzipping corpora/stopwords.zip.
[nltk_data] Downloading package wordnet to /root/nltk_data...
True
```

```
# Text Preprocessing

def clean_text(text):
  token = word_tokenize(text.lower())

  ftoken = [i for i in token if i.isalpha()]

  stpwd = stopwords.words("english")
  stoken = [i for i in ftoken if i not in stpwd]

  lemma = WordNetLemmatizer()
  ltoken = [lemma.lemmatize(i) for i in stoken]

  return " ".join(ltoken)


import nltk
nltk.download('omw-1.4')
```

```
[nltk_data] Downloading package omw-1.4 to /root/nltk_data...
True
```

```
data["Title_clean"]=data["title"].apply(clean_text)


data.head()
```

	title	text	label	Title_clean
0	Donald Trump Sends Out Embarrassing New Year'...	Donald Trump just couldn t wish all Americans ...	fake	donald trump sends embarrassing new year eve m...
1	Drunk Bragging Trump Staffer Started Russian ...	House Intelligence Committee Chairman Devin Nu...	fake	drunk bragging trump staffer started russian c...
2	Sheriff David Clarke Becomes An Internet Joke...	On Friday, it was revealed that former Milwauk...	fake	sheriff david clarke becomes internet joke thr...
3	Trump Is So Obsessed He Even Has Obama's Name...	On Christmas day, Donald Trump announced that ...	fake	trump obsessed even obama name coded website i...
4	Pope Francis Just Called Out Donald Trump Dur...	Pope Francis used his annual Christmas Day mes...	fake	pope francis called donald trump christmas speech

```
x = data["Title_clean"]
x
```

```
0        donald trump sends embarrassing new year eve m...
1        drunk bragging trump staffer started russian c...
2        sheriff david clarke becomes internet joke thr...
3        trump obsessed even obama name coded website i...
4        pope francis called donald trump christmas speech
                              ...
44893          committed nato back new approach afghanistan
44894       lexisnexis withdrew two product chinese market
44895                  minsk cultural hub becomes authority
44896    vatican upbeat possibility pope francis visiti...
44897               indonesia buy billion worth russian jet
Name: Title_clean, Length: 44898, dtype: object
```

```
# Encoding label column

from sklearn.preprocessing import LabelEncoder
le = LabelEncoder()
data['label'] = le.fit_transform(data['label'])


y = data["label"]
y
```

```
0        0
1        0
2        0
3        0
4        0
        ..
44893    1
44894    1
44895    1
44896    1
44897    1
Name: label, Length: 44898, dtype: int64
```

```
 # Vectorization

from sklearn.feature_extraction.text import TfidfVectorizer
vec = TfidfVectorizer(min_df=0.01)
vec.fit(x)
x = vec.transform(x).toarray()


x.shape
```

```
     (44898, 102)
```

```
# Splitting the dataset into training and testing

from sklearn.model_selection import train_test_split

xtrain,xtest,ytrain,ytest = train_test_split(x,y,test_size=0.30,random_state=1)



from sklearn.naive_bayes import GaussianNB,MultinomialNB,BernoulliNB



def mymodel(model):
  model.fit(xtrain,ytrain)
  ypred = model.predict(xtest)
  print(classification_report(ytest,ypred))


from sklearn.metrics import classification_report


gnb = GaussianNB()
mymodel(gnb)
```

```
                   precision    recall  f1-score   support

               0       0.86      0.77      0.81      7053
               1       0.77      0.86      0.81      6417

         accuracy                           0.81     13470
        macro avg       0.81      0.81      0.81     13470
     weighted avg       0.82      0.81      0.81     13470
```

```
mnb = MultinomialNB()
mymodel(mnb)
```

```
                   precision    recall  f1-score   support

               0       0.75      0.87      0.81      7053
               1       0.82      0.69      0.75      6417

         accuracy                           0.78     13470
        macro avg       0.79      0.78      0.78     13470
     weighted avg       0.79      0.78      0.78     13470
```

```
bnb = BernoulliNB()
mymodel(bnb)
```

```
                   precision    recall  f1-score   support

               0       0.88      0.77      0.82      7053
               1       0.77      0.88      0.82      6417

         accuracy                           0.82     13470
        macro avg       0.82      0.82      0.82     13470
     weighted avg       0.83      0.82      0.82     13470
```

```
from sklearn.linear_model import LogisticRegression
lr = LogisticRegression()
mymodel(lr)
```

```
                   precision    recall  f1-score   support

               0       0.89      0.76      0.82      7053
               1       0.78      0.90      0.83      6417

         accuracy                           0.83     13470
        macro avg       0.83      0.83      0.83     13470
```

```
weighted avg       0.84      0.83      0.83     13470
```