Importing the Dependencies

```
import numpy as np
import pandas as pd
import re
from nltk.corpus import stopwords
from nltk.stem.porter import PorterStemmer
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score
import matplotlib.pyplot as plt
```

```
from google.colab import drive
drive.mount('/content/drive')
```

```
    Mounted at /content/drive
```

Data Pre-processing

```
# loading the dataset to a pandas DataFrame
news_dataset = pd.read_csv("/content/drive/My Drive/dataset_fake_news/train.csv")
```

```
news_dataset.shape
```

```
    (20800, 5)
```

```
# print the first 5 rows of the dataframe
news_dataset.head()
```

	id	title	author	text
0	0	House Dem Aide: We Didn’t Even See Comey’s Let...	Darrell Lucus	House Dem Aide: We Didn’t Even See Comey’s Let...
1	1	FLYNN: Hillary Clinton, Big Woman on Campus - ...	Daniel J. Flynn	Ever get the feeling your life circles the rou...
2	2	Why the Truth Might Get You Fired	Consortiumnews.com	Why the Truth Might Get You Fired October 29, ...
		15 Civilians Killed In Single US		Videos 15 Civilians Killed In

```
# counting the number of missing values in the dataset
news_dataset.isnull().sum()
```

```
    id           0
    title      558
    author    1957
    text        39
```

```
label       0
dtype: int64
```

```
# replacing the null values with empty string
news_dataset = news_dataset.fillna('')
```

```
# merging the author name and news title
news_dataset['content'] = news_dataset['author']+' '+news_dataset['title']
```

```
print(news_dataset['content'])
```

```
0        Darrell Lucus House Dem Aide: We Didn’t Even S...
1        Daniel J. Flynn FLYNN: Hillary Clinton, Big Wo...
2        Consortiumnews.com Why the Truth Might Get You...
3        Jessica Purkiss 15 Civilians Killed In Single ...
4        Howard Portnoy Iranian woman jailed for fictio...
                               ...
20795    Jerome Hudson Rapper T.I.: Trump a ’Poster Chi...
20796    Benjamin Hoffman N.F.L. Playoffs: Schedule, Ma...
20797    Michael J. de la Merced and Rachel Abrams Macy...
20798    Alex Ansary NATO, Russia To Hold Parallel Exer...
20799              David Swanson What Keeps the F-35 Alive
Name: content, Length: 20800, dtype: object
```

```
# separating the data & label
X = news_dataset.drop(columns='label', axis=1)
Y = news_dataset['label']
```

```
print(X)
print(Y)
```

```
          id                                              title  \
0          0  House Dem Aide: We Didn’t Even See Comey’s Let...
1          1  FLYNN: Hillary Clinton, Big Woman on Campus - ...
2          2                  Why the Truth Might Get You Fired
3          3  15 Civilians Killed In Single US Airstrike Hav...
4          4  Iranian woman jailed for fictional unpublished...
...      ...                                                ...
20795  20795  Rapper T.I.: Trump a ’Poster Child For White S...
20796  20796  N.F.L. Playoffs: Schedule, Matchups and Odds -...
20797  20797  Macy’s Is Said to Receive Takeover Approach by...
20798  20798  NATO, Russia To Hold Parallel Exercises In Bal...
20799  20799                           What Keeps the F-35 Alive

                                          author  \
0                                  Darrell Lucus
1                                Daniel J. Flynn
2                             Consortiumnews.com
3                                Jessica Purkiss
4                                 Howard Portnoy
...                                          ...
20795                              Jerome Hudson
20796                           Benjamin Hoffman
20797  Michael J. de la Merced and Rachel Abrams
20798                                Alex Ansary
20799                              David Swanson
```

```
                                                      text  \
0      House Dem Aide: We Didn’t Even See Comey’s Let...
1      Ever get the feeling your life circles the rou...
2      Why the Truth Might Get You Fired October 29, ...
3      Videos 15 Civilians Killed In Single US Airstr...
4      Print \nAn Iranian woman has been sentenced to...
...                                                  ...
20795  Rapper T. I. unloaded on black celebrities who...
20796  When the Green Bay Packers lost to the Washing...
20797  The Macy’s of today grew from the union of sev...
20798  NATO, Russia To Hold Parallel Exercises In Bal...
20799    David Swanson is an author, activist, journa...

                                                 content
0      Darrell Lucus House Dem Aide: We Didn’t Even S...
1      Daniel J. Flynn FLYNN: Hillary Clinton, Big Wo...
2      Consortiumnews.com Why the Truth Might Get You...
3      Jessica Purkiss 15 Civilians Killed In Single ...
4      Howard Portnoy Iranian woman jailed for fictio...
...                                                  ...
20795  Jerome Hudson Rapper T.I.: Trump a ’Poster Chi...
20796  Benjamin Hoffman N.F.L. Playoffs: Schedule, Ma...
20797  Michael J. de la Merced and Rachel Abrams Macy...
20798  Alex Ansary NATO, Russia To Hold Parallel Exer...
20799            David Swanson What Keeps the F-35 Alive

[20800 rows x 5 columns]
0        1
1        0
2        1
3        1
4        1
```

```
import nltk
nltk.download('stopwords')
```

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

```
# printing the stopwords in English
print(stopwords.words('english'))
```

```
['i', 'me', 'my', 'myself', 'we', 'our', 'ours', 'ourselves', 'you', "you're", "you'
```

Stemming:

Stemming is the process of reducing a word to its Root word

example: actor, actress, acting --> act

```
port_stem = PorterStemmer()
```

```
def stemming(content):
    stemmed_content = re.sub('[^a-zA-Z]',' ',content)
    stemmed_content = stemmed_content.lower()
    stemmed_content = stemmed_content.split()
    stemmed_content = [port_stem.stem(word) for word in stemmed_content if not word in sto
    stemmed_content = ' '.join(stemmed_content)
    return stemmed_content
```

```
news_dataset['content'] = news_dataset['content'].apply(stemming)
```

```
print(news_dataset['content'])
```

```
0        darrel lucu hous dem aid even see comey letter...
1        daniel j flynn flynn hillari clinton big woman...
2                   consortiumnew com truth might get fire
3        jessica purkiss civilian kill singl us airstri...
4        howard portnoy iranian woman jail fiction unpu...
                               ...
20795    jerom hudson rapper trump poster child white s...
20796    benjamin hoffman n f l playoff schedul matchup...
20797    michael j de la merc rachel abram maci said re...
20798    alex ansari nato russia hold parallel exercis ...
20799                             david swanson keep f aliv
Name: content, Length: 20800, dtype: object
```

```
#separating the data and label
X = news_dataset['content'].values
Y = news_dataset['label'].values
```

```
print(X)
```

```
['darrel lucu hous dem aid even see comey letter jason chaffetz tweet'
 'daniel j flynn flynn hillari clinton big woman campu breitbart'
 'consortiumnew com truth might get fire' ...
 'michael j de la merc rachel abram maci said receiv takeov approach hudson bay new y
 'alex ansari nato russia hold parallel exercis balkan'
 'david swanson keep f aliv']
```

```
print(Y)
```

```
[1 0 1 ... 0 1 1]
```

```
Y.shape
```

```
(20800,)
```

```
news_dataset.head()
```

	id	title	author	text	label	c
0	0	House Dem Aide: We Didn’t Even See Comey’s Let...	Darrell Lucus	House Dem Aide: We Didn’t Even See Comey’s Let...	1	darrel lu dem se
1	1	FLYNN: Hillary Clinton, Big Woman on Campus - ...	Daniel J. Flynn	Ever get the feeling your life circles the rou...	0	dani fly cl v

```
X_train, X_test, y_train, y_test = train_test_split(news_dataset['content'],
                                                    news_dataset['label'],
                                                    test_size=0.2,
                                                    random_state=8)
```

```
ngram_range = (1,2)
min_df = 10
max_df = 1.
max_features = 300
```

```
tfidf = TfidfVectorizer(encoding='utf-8',
                        ngram_range=ngram_range,
                        stop_words=None,
                        lowercase=False,
                        max_df=max_df,
                        min_df=min_df,
                        max_features=max_features,
                        norm='l2',
                        sublinear_tf=True)

features_train = tfidf.fit_transform(X_train).toarray()
labels_train = y_train
print(features_train)

features_test = tfidf.transform(X_test).toarray()
labels_test = y_test
print(features_test.shape)
```

```
[[0.         0.         0.         ... 0.         0.         0.        ]
 [0.         0.         0.         ... 0.         0.28784757 0.28839608]
 [0.         0.         0.         ... 0.         0.         0.        ]
 ...
 [0.         0.         0.         ... 0.         0.         0.        ]
 [0.         0.         0.         ... 0.         0.45177427 0.45263517]
 [0.         0.         0.         ... 0.         0.         0.        ]]
(4160, 300)
```

```
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score, classification_report
from sklearn.linear_model import LogisticRegression
from sklearn.tree import DecisionTreeClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.discriminant_analysis import LinearDiscriminantAnalysis
from sklearn.naive_bayes import GaussianNB
from sklearn.svm import SVC
from nltk.tokenize import word_tokenize

import re
import warnings
warnings.filterwarnings("ignore")
```

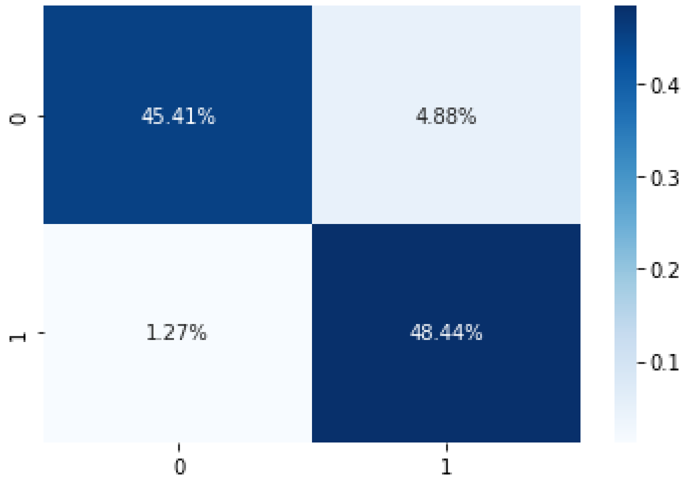
Naive Bayes

```
model = GaussianNB()
model.fit(features_train, labels_train)
model_predictions = model.predict(features_test)
print('Accuracy: ', accuracy_score(labels_test, model_predictions))
print(classification_report(labels_test, model_predictions))
```

```
Accuracy:  0.9384615384615385
              precision    recall  f1-score   support

           0       0.97      0.90      0.94      2092
           1       0.91      0.97      0.94      2068

    accuracy                           0.94      4160
   macro avg       0.94      0.94      0.94      4160
weighted avg       0.94      0.94      0.94      4160
```

```
from sklearn.metrics import confusion_matrix
import seaborn as sns
cf_matrix = confusion_matrix(labels_test, model_predictions)
print(cf_matrix)
sns.heatmap(cf_matrix/np.sum(cf_matrix), annot=True,
            fmt='.2%', cmap='Blues')
```

```
[[1889  203]
 [  53 2015]]
<matplotlib.axes._subplots.AxesSubplot at 0x7fe2f3284fd0>
```



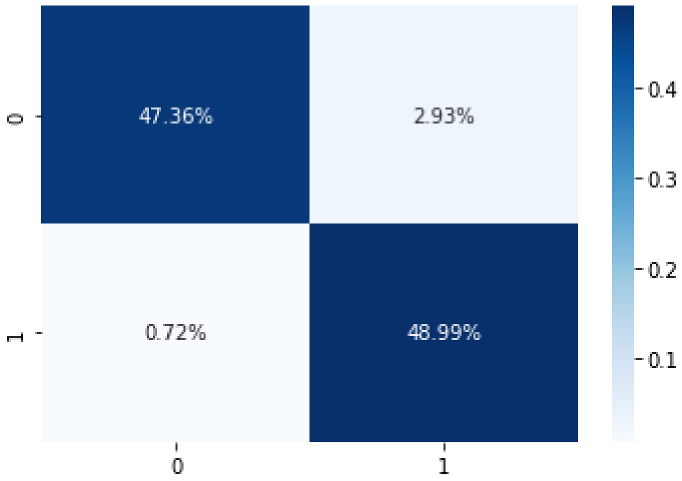
Random Forest

```
from sklearn.ensemble import RandomForestClassifier
model  = RandomForestClassifier(random_state=1)
model.fit(features_train, labels_train)
model_predictions = model.predict(features_test)
print('Accuracy: ', accuracy_score(labels_test, model_predictions))
print(classification_report(labels_test, model_predictions))
```

```
Accuracy:  0.9634615384615385
              precision    recall  f1-score   support

           0       0.98      0.94      0.96      2092
           1       0.94      0.99      0.96      2068

    accuracy                           0.96      4160
   macro avg       0.96      0.96      0.96      4160
weighted avg       0.96      0.96      0.96      4160
```

```
cf_matrix = confusion_matrix(labels_test, model_predictions)
print(cf_matrix)
sns.heatmap(cf_matrix/np.sum(cf_matrix), annot=True,
            fmt='.2%', cmap='Blues')
```

```
[[1970  122]
 [  30 2038]]
<matplotlib.axes._subplots.AxesSubplot at 0x7fe2f3163f90>
```



Logistic regression

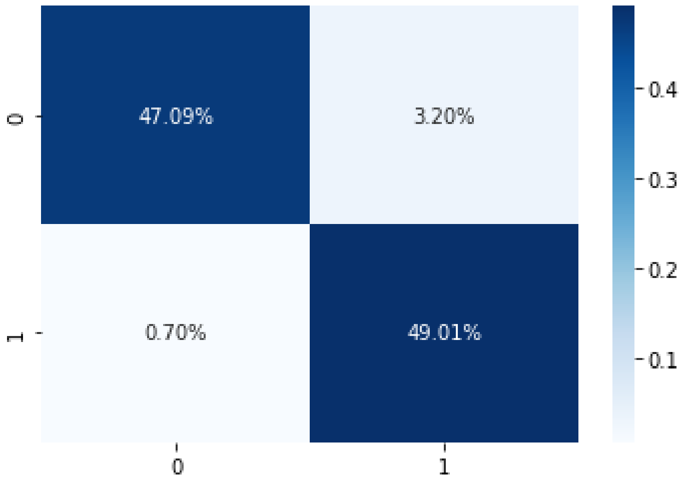
```
model = LogisticRegression()
print(model.get_params())
model.fit(features_train, labels_train)
model_predictions = model.predict(features_test)
print('Accuracy: ', accuracy_score(labels_test, model_predictions))
print(classification_report(labels_test, model_predictions))
```

```
{'C': 1.0, 'class_weight': None, 'dual': False, 'fit_intercept': True, 'intercept_sca
Accuracy:  0.9610576923076923
```

```
              precision    recall  f1-score   support

           0       0.99      0.94      0.96      2092
           1       0.94      0.99      0.96      2068

    accuracy                           0.96      4160
   macro avg       0.96      0.96      0.96      4160
weighted avg       0.96      0.96      0.96      4160
```

```
cf_matrix = confusion_matrix(labels_test, model_predictions)
print(cf_matrix)
sns.heatmap(cf_matrix/np.sum(cf_matrix), annot=True,
            fmt='.2%', cmap='Blues')
```

```
[[1959  133]
 [  29 2039]]
<matplotlib.axes._subplots.AxesSubplot at 0x7fe2f3010a90>
```



KNeighborsClassifier

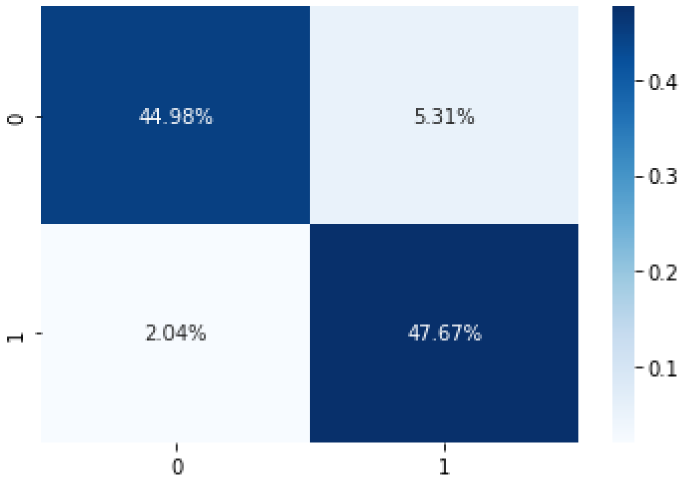
```
model = KNeighborsClassifier()
model.fit(features_train, labels_train)
model_predictions = model.predict(features_test)
print('Accuracy: ', accuracy_score(labels_test, model_predictions))
print(classification_report(labels_test, model_predictions))
```

```
Accuracy:  0.9264423076923077
              precision    recall  f1-score   support

           0       0.96      0.89      0.92      2092
           1       0.90      0.96      0.93      2068

    accuracy                           0.93      4160
   macro avg       0.93      0.93      0.93      4160
weighted avg       0.93      0.93      0.93      4160
```

```
cf_matrix = confusion_matrix(labels_test, model_predictions)
```

```
print(cf_matrix)
sns.heatmap(cf_matrix/np.sum(cf_matrix), annot=True,
            fmt='.2%', cmap='Blues')
```

```
[[1871  221]
 [  85 1983]]
<matplotlib.axes._subplots.AxesSubplot at 0x7fe2f2f434d0>
```


0
1
44.98%
5.31%
2.04%
47.67%
0
1
0.4
0.3
0.2
0.1