```
import numpy  as np
import pandas as pd
import matplotlib.pyplot as plt
import nltk
from nltk.corpus import stopwords
from nltk.stem.porter import PorterStemmer
```

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

```
Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/dri
```

```
df = pd.read_csv("/content/spam.csv",encoding="latin")
df.head()
```

	v1	v2	Unnamed: 2	Unnamed: 3	Unnamed: 4
0	ham	Go until jurong point, crazy.. Available only ...	NaN	NaN	NaN
1	ham	Ok lar... Joking wif u oni...	NaN	NaN	NaN
2	spam	Free entry in 2 a wkly comp to win FA Cup fina...	NaN	NaN	NaN
3	ham	U dun say so early hor... U c already then say...	NaN	NaN	NaN
4	ham	Nah I don't think he goes to usf, he lives aro...	NaN	NaN	NaN

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5572 entries, 0 to 5571
Data columns (total 5 columns):
 #   Column      Non-Null Count  Dtype 
---  ------      --------------  ----- 
 0   v1          5572 non-null   object
 1   v2          5572 non-null   object
 2   Unnamed: 2  50 non-null     object
 3   Unnamed: 3  12 non-null     object
 4   Unnamed: 4  6 non-null      object
dtypes: object(5)
memory usage: 217.8+ KB
```

```
df.isna().sum()
```

```
v1               0
v2               0
Unnamed: 2    5522
Unnamed: 3    5560
Unnamed: 4    5566
dtype: int64
```

```
df.rename({"v1":"label","v2":"text"},inplace=True,axis=1)
```

```
df.tail()
```

```
df.tail()
```

	label	text	Unnamed: 2	Unnamed: 3	Unnamed: 4
5567	spam	This is the 2nd time we have tried 2 contact u...	NaN	NaN	NaN
5568	ham	Will Ì_ b going to esplanade fr home?	NaN	NaN	NaN
5569	ham	Pity, * was in mood for that. So...any other s...	NaN	NaN	NaN
5570	ham	The guy did some bitching but I acted like i'd...	NaN	NaN	NaN
5571	ham	Rofl. Its true to its name	NaN	NaN	NaN

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

```
from sklearn.model_selection import train_test_split
X = np.random.rand(100, 10)
y = np.random.randint(0, 2, 100)
X_train, X_test, y_train, y_test=train_test_split(X,y, test_size = 0.20, random_state = 0)
```

```
print("Before OverSampling, counts of label '1': {}".format(sum(y_train == 1)))
print("Before OverSampling, counts of label '0': {}".format(sum(y_train == 0)))
```

```
Before OverSampling, counts of label '1': 33
Before OverSampling, counts of label '0': 47
```

```
from imblearn.over_sampling import SMOTE
sm = SMOTE(random_state = 2)
X_train_res, y_train_res = sm.fit_resample(X_train, y_train.ravel())
```

```
print('After OverSampling, the shape of train_X: {}'.format(X_train_res.shape))
print('After OverSampling, the shape of train_y: {}'.format(y_train_res.shape))
```

```
After OverSampling, the shape of train_X: (94, 10)
After OverSampling, the shape of train_y: (94,)
```

```
print("After OverSampling, counts of label '1': {}".format(sum(y_train_res == 1)))
print("After OverSampling, counts of label '0': {}".format(sum(y_train_res == 0)))
```

```
After OverSampling, counts of label '1': 47
After OverSampling, counts of label '0': 47
```

```
nltk.download("stopwords")
```

```
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
True
```

```
import nltk
from nltk.corpus import stopwords
from nltk.stem import PorterStemmer
```

```
import re
corpus = []
```

```
length = len(df)


for i in range(0,length):
  text = re.sub("[^a-zA-Z0-9]"," ",df["text"][i])
  text = text.lower()
  text = text.split()
  pe = PorterStemmer()
  stopword = stopwords.words("english")
  text = [pe.stem(word) for word in text if not word in set (stopword)]
  text = " ".join(text)
  corpus.append(text)


corpus
```

```
 wan come come for din c stripe skirt ,
 'xma stori peac xma msg love xma miracl jesu hav bless month ahead amp wish u merri xma',
 'number',
 'chang e one next escal',
 'yetund class run water make ok pl',
 'lot happen feel quiet beth aunt charli work lot helen mo',
 'wait 4 bu stop aft ur lect lar dun c go get car come back n pick',
 'aight thank comin',
 ...]
```

```
from sklearn.feature_extraction.text import CountVectorizer
cv = CountVectorizer(max_features=35000)
x = cv.fit_transform(corpus).toarray()
```

```
import pickle
pickle.dump(cv, open('cv.pkl', 'wb'))
```

```
df.describe()
```

	label
count	5572.000000
mean	0.134063
std	0.340751
min	0.000000
25%	0.000000
50%	0.000000
75%	0.000000
max	1.000000

```
df.shape
```

```
(5572, 5)
```

```
df["label"].value_counts().plot(kind="bar",figsize=(12,6))
plt.xticks(np.arange(2), ('Non spam', 'spam'),rotation=0);
```


```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.20, random_state = 0)
```

```
from sklearn.tree import DecisionTreeClassifier
model = DecisionTreeClassifier()
model.fit(X_train_res, y_train_res)
```

```
▾ DecisionTreeClassifier
DecisionTreeClassifier()
```

```
from sklearn.ensemble import RandomForestClassifier
model = RandomForestClassifier()
model.fit(X_train_res, y_train_res)
```

```
▾ RandomForestClassifier
RandomForestClassifier()
```

```
from sklearn.naive_bayes import MultinomialNB
model = MultinomialNB()
```

```
model.fit(X_train_res, y_train_res)
```

```
▾ MultinomialNB
MultinomialNB()
```

```
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense
```

```
model = Sequential()
```

```
X_train.shape
```

```
(80, 10)
```

```
model.add(Dense(units = X_train_res.shape[1],activation="relu" ,kernel_initializer="random_uniform"))
```

```
model.add(Dense(units=100,activation="relu",kernel_initializer="random_uniform"))
```

```
model.add(Dense(units=100,activation="relu",kernel_initializer="random_uniform"))
```

```
model.add(Dense(units=1,activation="sigmoid"))
```

```
model.compile(optimizer="adam",loss="binary_crossentropy",metrics=['accuracy'])
```

```
generator = model.fit(X_train_res,y_train_res,epochs=10,steps_per_epoch=len(X_train_res)//64)
```

```
Epoch 1/10
1/1 [==============================] - 1s 1s/step - loss: 0.6931 - accuracy: 0.5000
Epoch 2/10
1/1 [==============================] - 0s 11ms/step - loss: 0.6931 - accuracy: 0.5532
Epoch 3/10
1/1 [==============================] - 0s 11ms/step - loss: 0.6930 - accuracy: 0.5532
Epoch 4/10
1/1 [==============================] - 0s 14ms/step - loss: 0.6929 - accuracy: 0.5532
Epoch 5/10
1/1 [==============================] - 0s 10ms/step - loss: 0.6929 - accuracy: 0.5851
Epoch 6/10
1/1 [==============================] - 0s 10ms/step - loss: 0.6928 - accuracy: 0.5957
Epoch 7/10
1/1 [==============================] - 0s 13ms/step - loss: 0.6926 - accuracy: 0.6170
Epoch 8/10
1/1 [==============================] - 0s 13ms/step - loss: 0.6925 - accuracy: 0.6383
Epoch 9/10
1/1 [==============================] - 0s 12ms/step - loss: 0.6924 - accuracy: 0.6064
Epoch 10/10
1/1 [==============================] - 0s 10ms/step - loss: 0.6922 - accuracy: 0.6064
```

```
y_pred=model.predict(X_test)
y_pred
```

```
1/1 [==============================] - 0s 72ms/step
array([[0.49902344],
       [0.5003518 ],
       [0.50153124],
       [0.5015287 ],
       [0.49908033],
       [0.50143987],
       [0.5026305 ],
       [0.5001773 ],
       [0.50033206],
       [0.49913207],
       [0.50165904],
       [0.50051934],
       [0.50287753],
       [0.5005187 ],
       [0.50051713],
       [0.49911037],
       [0.50226706],
       [0.5013489 ],
       [0.49811435],
       [0.50095797]], dtype=float32)
```

```
y_pr = np.where(y_pred>0.5,1,0)
```

```
y_test
```

```
array([1, 0, 1, 1, 0, 0, 0, 0, 0, 1, 0, 1, 1, 1, 1, 1, 1, 0, 0, 0])
```

```
from sklearn.metrics import confusion_matrix,accuracy_score
cm = confusion_matrix(y_test, y_pr)
score = accuracy_score(y_test, y_pr)
print(cm)
print('Accuracy Score Is:-' ,score*100)
```

```
[[2 8]
 [3 7]]
Accuracy Score Is:- 45.0
```

```
def new_review(new_review):
  new_review = new_review
  new_review = re.sub('[^a-zA-Z]', ' ', new_review)
  new_review = new_review.lower()
  new_review = new_review.split()
  ps = PorterStemmer()
  all_stopwords = stopwords.words('english')
  all_stopwords.remove('not')
  new_review = [ps.stem(word) for word in new_review if not word in set(all_stopwords)]
  new_review = ' '.join(new_review)
  new_corpus = [new_review]
  new_X_test = cv.transform(new_corpus).toarray()
  print(new_X_review)
  new_y_pred = loaded_model.predict(new_X_test)
  print(new_y_pred)
  new_review = new_review(str(input("Enter new review...")))
```

```
from sklearn.metrics import confusion_matrix,accuracy_score, classification_report
cm = confusion_matrix(y_test, y_pr)
score = accuracy_score(y_test, y_pr)
print(cm)
print('Accuracy Score Is Naive Bayes:- ' ,score*100)
```

```
[[2 8]
 [3 7]]
Accuracy Score Is Naive Bayes:-  45.0
```

```
cm = confusion_matrix(y_test, y_pr)
score = accuracy_score(y_test, y_pr)
print(cm)
print('Accuracy Score Is:- ' ,score*100)

cm = confusion_matrix(y_test, y_pr)
score = accuracy_score(y_test, y_pr)
print(cm)
print('Accuracy Score Is;- ' ,score*100)
```

```
[[2 8]
 [3 7]]
Accuracy Score Is:-  45.0
[[2 8]
 [3 7]]
Accuracy Score Is;-  45.0
```

```
import tensorflow as tf
```

```
model = tf.keras.models.Sequential([
    tf.keras.layers.Dense(64, activation='relu', input_shape=(100,)),
    tf.keras.layers.Dense(1, activation='sigmoid')
])

model.compile(loss='binary_crossentropy', optimizer='adam', metrics=['accuracy'])



model.save('spam.h5')
```

✓ 0s completed at 10:55 AM