```
import pandas as pd
import numpy as np
```

```
df=pd.read_csv("SMSSpamCollection",sep="\t",names=['Label','Message'])
print("Data for Spam or Ham is:\n",df)
```

```
Data for Spam or Ham is:
      Label                                            Message
0      ham  Go until jurong point, crazy.. Available only ...
1      ham                      Ok lar... Joking wif u oni...
2     spam  Free entry in 2 a wkly comp to win FA Cup fina...
3      ham  U dun say so early hor... U c already then say...
4      ham  Nah I don't think he goes to usf, he lives aro...
...    ...                                                ...
5567  spam  This is the 2nd time we have tried 2 contact u...
5568   ham               Will ü b going to esplanade fr home?
5569   ham  Pity, * was in mood for that. So...any other s...
5570   ham  The guy did some bitching but I acted like i'd...
5571   ham                         Rofl. Its true to its name

[5572 rows x 2 columns]
```

```
#machine learning model does not understand any string format data
#so for reading string format data we need to do text preprocessing
!pip install nltk
```

```
Requirement already satisfied: nltk in /usr/local/lib/python3.10/dist-packages (3.8.1)
Requirement already satisfied: click in /usr/local/lib/python3.10/dist-packages (from nltk) (8.1.7)
Requirement already satisfied: joblib in /usr/local/lib/python3.10/dist-packages (from nltk) (1.3.2)
Requirement already satisfied: regex>=2021.8.3 in /usr/local/lib/python3.10/dist-packages (from nltk) (2023.6.3)
Requirement already satisfied: tqdm in /usr/local/lib/python3.10/dist-packages (from nltk) (4.66.1)
```

```
import nltk
```

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

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

```
from nltk.tokenize import word_tokenize
from nltk.corpus import stopwords
from nltk.stem import PorterStemmer
from sklearn.feature_extraction.text import TfidfVectorizer
```

```
nltk.download('punkt')
```

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

```
ps=PorterStemmer()
swords=stopwords.words('english')
```

```
def Clean_Text(sentence):
  tokens=word_tokenize(sentence)
  clean=[word for word in tokens
         if word.isdigit() or word.isalpha()]
  clean=[ps.stem(word) for word in clean
         if word not in swords]
  return clean
```

```
sentence1="Hello Mayuri How are you? We will be learning Python in Machine Learning Today!!"
Clean_Text(sentence1)
```

```
['hello', 'mayuri', 'how', 'we', 'learn', 'python', 'machin', 'learn', 'today']
```

```
x=df['Message']
y=df['Label']
```

```
tfidf=TfidfVectorizer()
```

```
x_new=tfidf.fit_transform(x)
```

```
x_new
```

```
<5572x8713 sparse matrix of type '<class 'numpy.float64'>'
        with 74169 stored elements in Compressed Sparse Row format>
```

```
before=x.shape
after=x_new.shape
print("Shape Before Cleaning:",before)
print("Shape After Cleaning:",after)
```

```
Shape Before Cleaning: (5572,)
Shape After Cleaning: (5572, 8713)
```

```
from sklearn.model_selection import train_test_split
```

```
x_train,x_test,y_train,y_test=train_test_split(x_new,y,random_state=0,test_size=0.25)
```

```
x_train.shape
```

```
(4179, 8713)
```

```
y_train.shape
```

```
(4179,)
```

```
x_test.shape
```

```
(1393, 8713)
```

```
y_test.shape
```

```
(1393,)
```

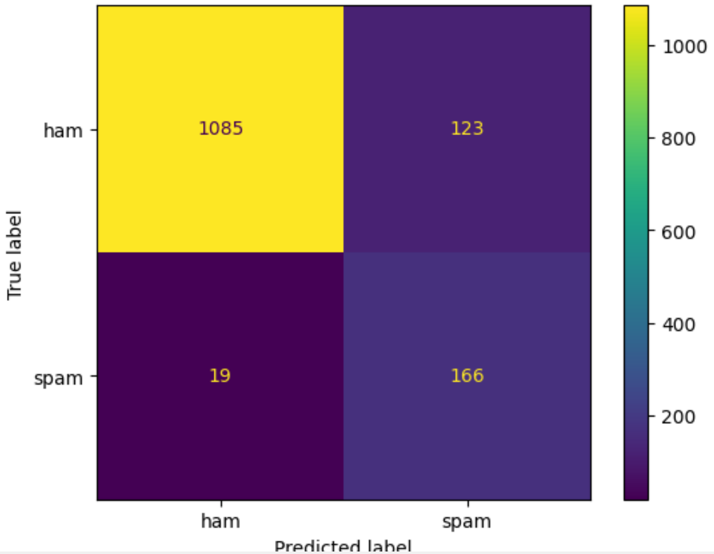
```
from sklearn.naive_bayes import GaussianNB
nb=GaussianNB()
nb.fit(x_train.toarray(),y_train)
```

```
▾ GaussianNB
GaussianNB()
```

```
y_pred=nb.predict(x_test.toarray())
```

```
from sklearn.metrics import ConfusionMatrixDisplay,accuracy_score
print("The Matrix Display is:\n",ConfusionMatrixDisplay.from_predictions(y_test,y_pred))
```

```
The Matrix Display is:
 <sklearn.metrics._plot.confusion_matrix.ConfusionMatrixDisplay object at 0x7e68c0875
```



```
from sklearn.metrics import classification_report
print("Classification Report is:\n",classification_report(y_test,y_pred))
```

```
Classification Report is:
               precision    recall  f1-score   support
```

```
             ham       0.98      0.90      0.94      1208
            spam       0.57      0.90      0.70       185

        accuracy                           0.90      1393
       macro avg       0.78      0.90      0.82      1393
    weighted avg       0.93      0.90      0.91      1393
```

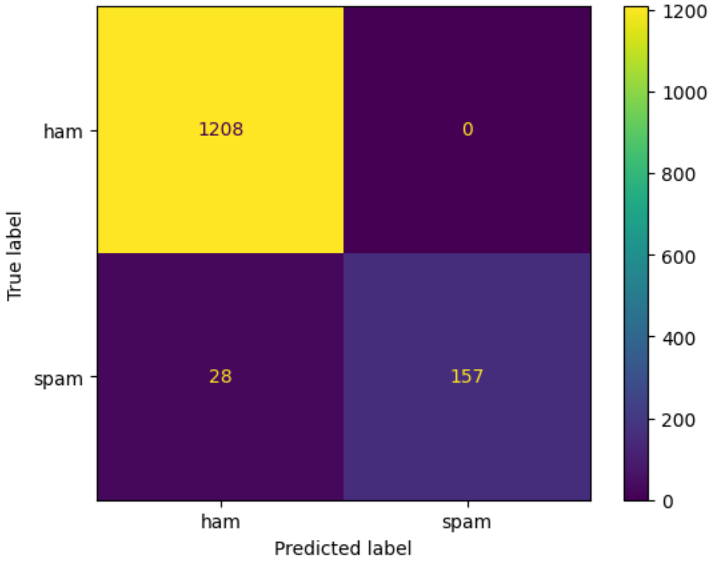
```
print("Accuracy Score:",accuracy_score(y_test,y_pred))
```

```
Accuracy Score: 0.8980617372577172
```

```
from sklearn.ensemble import RandomForestClassifier
rf=RandomForestClassifier(random_state=0)
rf.fit(x_train,y_train)
y_pred=rf.predict(x_test)
```

```
ConfusionMatrixDisplay.from_predictions(y_test,y_pred)
```

```
<sklearn.metrics._plot.confusion_matrix.ConfusionMatrixDisplay at 0x7e68be08f160>
```



```
from sklearn.metrics import classification_report
print("Classification Report is:\n",classification_report(y_test,y_pred))
```

```
Classification Report is:
               precision    recall  f1-score   support

         ham       0.98      1.00      0.99      1208
        spam       1.00      0.85      0.92       185

    accuracy                           0.98      1393
   macro avg       0.99      0.92      0.95      1393
weighted avg       0.98      0.98      0.98      1393
```

```
print("Accuracy Score:",accuracy_score(y_test,y_pred))
```

```
Accuracy Score: 0.9798994974874372
```

```
from sklearn.linear_model import LogisticRegression
log=LogisticRegression()
log.fit(x_train,y_train)
y_pred=log.predict(x_test)
print("Accuracy Score:",accuracy_score(y_test,y_pred))
```

```
Accuracy Score: 0.9612347451543432
```

```
from sklearn.model_selection import GridSearchCV
parameters={
    'criterion':['grid','entropy'],
    'max_features':['sqrt','log2'],
    'random_state':[0,1,2,3,4,5],
    'class_weight':['balanced','balanced_subsample']
}
```

```
grid=GridSearchCV(rf,param_grid=parameters,cv=5,scoring='accuracy')
```

```
grid.fit(x_train,y_train)
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/model_selection/_validation.py:378: F
120 fits failed out of a total of 240.
The score on these train-test partitions for these parameters will be set to nan.
If these failures are not expected, you can try to debug them by setting error_score=

Below are more details about the failures:
--------------------------------------------------------------------------------
120 fits failed with the following error:
Traceback (most recent call last):
  File "/usr/local/lib/python3.10/dist-packages/sklearn/model_selection/_validation.p
    estimator.fit(X_train, y_train, **fit_params)
  File "/usr/local/lib/python3.10/dist-packages/sklearn/ensemble/_forest.py", line 34
    self._validate_params()
  File "/usr/local/lib/python3.10/dist-packages/sklearn/base.py", line 600, in _valid
    validate_parameter_constraints(
  File "/usr/local/lib/python3.10/dist-packages/sklearn/utils/_param_validation.py",
    raise InvalidParameterError(
sklearn.utils._param_validation.InvalidParameterError: The 'criterion' parameter of R

  warnings.warn(some_fits_failed_message, FitFailedWarning)
/usr/local/lib/python3.10/dist-packages/sklearn/model_selection/_search.py:952: UserW
        nan        nan        nan        nan        nan        nan
 0.97128642 0.97200355 0.97248288 0.97104747 0.97128499 0.97224365
 0.96267255 0.96482623 0.96506604 0.96219351 0.96363006 0.96410767
        nan        nan        nan        nan        nan        nan
        nan        nan        nan        nan        nan        nan
 0.96985073 0.97415724 0.97200327 0.97104719 0.97176403 0.97272126
 0.9636292  0.96386901 0.96315102 0.96386901 0.96339025 0.96362891]
  warnings.warn(
▸           GridSearchCV
▸ estimator: RandomForestClassifier
      ▸ RandomForestClassifier
```

```
rf=grid.best_estimator_
y_pred=rf.predict(x_test)
print("Accuracy Score:",accuracy_score(y_test,y_pred))
```

```
Accuracy Score: 0.9791816223977028
```