CHANDAN KUMAR

ID: GO_STP_13267

```python
import pandas as pd
import numpy as np
import sklearn
import re
import string
import nltk
from sklearn.feature_extraction.text import CountVectorizer
nltk.download('stopwords')
from nltk.corpus import stopwords
```

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

```python
df=pd.read_csv(r"./spam.csv")
df.drop_duplicates(inplace=True)
df.head()
```

	Category	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...

```python
df['Category']=df['Category'].map({'ham':0, 'spam':1})
df.head()
```

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

```python
def clean_data(message):
    message_without_punc=[char for char in message if char not in string.punctuation]
    message_without_punc="".join(message_without_punc)
    separator=' '
    return separator.join([word for word in message_without_punc.split() if word.lower

df['Message']=df['Message'].apply(func=clean_data)
```

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

```
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix
```

```
cv=CountVectorizer()
x=cv.fit_transform(x)
```

```
x_train, x_test, y_train, y_test=train_test_split(x, y, test_size=0.2, random_state=0)
```

```
from sklearn.naive_bayes import MultinomialNB
model=MultinomialNB()
model.fit(x_train, y_train)
```

```
MultinomialNB(alpha=1.0, class_prior=None, fit_prior=True)
```

```
predictions=model.predict(x_test)
print("Accuracy:",accuracy_score(y_test, predictions))
print("Confusion matrix:\n",confusion_matrix(y_test, predictions))
print("Classification report:\n",classification_report(y_test, predictions))
```

```
Accuracy: 0.9738372093023255
Confusion matrix:
 [[880  17]
 [ 10 125]]
Classification report:
               precision    recall  f1-score   support

           0       0.99      0.98      0.98       897
           1       0.88      0.93      0.90       135

    accuracy                           0.97      1032
   macro avg       0.93      0.95      0.94      1032
weighted avg       0.97      0.97      0.97      1032
```

```
def predict(text):
    labels=['Not Spam','Spam']
    x=cv.transform(text).toarray()
    p=model.predict(x)
    return labels[p[0]]

print(predict(['Congratuations you won a lottery of $4000']))
```

```
Spam
```

✓ 0s completed at 4:05 PM