AIM: PROGRAM TO IMPLEMENT TEXT CLASSIFICATION USING SVM

IMPORTING APPROPRIATE LIBRARIES

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

READING DATASET

```
df=pd.read_csv('/content/SMSSpamCollection',sep='\t',names=['label','message'])
```

```
df.head(3)
```

	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...

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5572 entries, 0 to 5571
Data columns (total 2 columns):
 #   Column   Non-Null Count  Dtype 
---  ------   --------------  ----- 
 0   label    5572 non-null   object
 1   message  5572 non-null   object
dtypes: object(2)
memory usage: 87.2+ KB
```

IMPORTING LIBRARIES FOR PREPRCESSING AS WELL AS THE MODEL(SVM)

```
import string
from nltk.corpus import stopwords
from sklearn.feature_extraction.text import CountVectorizer,TfidfTransformer
from sklearn import svm
```

DOWNLOADING STOPWORDS FROM NLTK

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

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

```
True
```

DEFINING A FUNCTION TO REMOVE PUNCTUATIONS & STOPWORDS

```
def converter(mess):
  nopunc=[char for char in mess if char not in string.punctuation]
  nopunc=''.join(nopunc)
  return[word for word in nopunc.split() if word.lower() not in stopwords.words('english')
```

SPLITTING DATA AS TRAIN AND TEST DATA

```
from sklearn.model_selection import train_test_split
xtrain,xtest,ytrain,ytest=train_test_split(df['message'],df['label'],test_size=0.2,random_
```

CREATION OF PIPELINE

```
from sklearn.pipeline import Pipeline
pipe=Pipeline([ ('bow',CountVectorizer(analyzer=converter)),
                ('tfidf',TfidfTransformer()),
                ('classifier',svm.SVC(C=1.0,kernel='linear',degree=3,gamma='auto'))
              ])
```

FITTING DATA INTO MODEL

```
pipe.fit(xtrain,ytrain)
```

```
Pipeline(steps=[('bow',
                 CountVectorizer(analyzer=<function converter at 0x7f118b21ea70>)),
                ('tfidf', TfidfTransformer()),
                ('classifier', SVC(gamma='auto', kernel='linear'))])
```

PREDICTING FOR TEST DATA

```
predictions=pipe.predict(xtest)
```

MODEL EVALUATION USING LIBRARIES FROM SKLEARN'S METRICS LIBRARIES

```
from sklearn.metrics import classification_report,accuracy_score,confusion_matrix
print("CLASSIFICATION REPORT:\n"+classification_report(ytest,predictions))
print("ACCURACY SCORE:")
print(+accuracy_score(ytest,predictions))
```

```
CLASSIFICATION REPORT:
              precision    recall  f1-score   support
```

```
         ham       0.99      1.00      0.99       966
        spam       0.99      0.92      0.95       149

    accuracy                           0.99      1115
   macro avg       0.99      0.96      0.97      1115
weighted avg       0.99      0.99      0.99      1115

ACCURACY SCORE:
0.9874439461883409
```

✓ 0s completed at 10:55 AM ● ×