cfbct7qez

March 30, 2024

[35]:

```
import pandas as pd
from sklearn.svm import SVC
from sklearn.model_selection import train_test_split,GridSearchCV,KFold
from sklearn.metrics import␣
 ↪accuracy_score,classification_report,confusion_matrix
from sklearn.feature_extraction.text import CountVectorizer
from imblearn.over_sampling import SMOTE
```

[3]:

```
data=pd.read_csv("spam.csv")
```

[4]:

```
data.head()
```

[4]:

```
  Label                                          EmailText
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…
```

[5]:

```
x=data["EmailText"]
```

[14]:

```
y=data["Label"]
```

1 Count Vectorizer

[8]:

```
cvec=CountVectorizer()
```

[9]:

```
cx=cvec.fit_transform(x)
```

[11]:

```
cx.toarray()
```

[11]:

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

```
[12]: cx.shape
```

```
[12]: (5572, 8679)
```

```
[15]: y.value_counts()
```

```
[15]: ham     4825
      spam     747
      Name: Label, dtype: int64
```

```
[17]: smt=SMOTE()
      x_sm,y_sm=smt.fit_resample(cx,y)
```

```
[18]: x_sm
```

```
[18]: <9650x8679 sparse matrix of type '<class 'numpy.int64'>'
              with 180863 stored elements in Compressed Sparse Row format>
```

```
[19]: y_sm
```

```
[19]: 0        ham
      1        ham
      2       spam
      3        ham
      4        ham
              …
      9645    spam
      9646    spam
      9647    spam
      9648    spam
      9649    spam
      Name: Label, Length: 9650, dtype: object
```

```
[20]: y_sm.value_counts()
```

```
[20]: spam    4825
      ham     4825
      Name: Label, dtype: int64
```

```
[22]: x_sm.shape
```

```
[22]: (9650, 8679)
```

2 SVM

[23]:
```
x_train, x_test, y_train,y_test = train_test_split(x_sm,y_sm,test_size=0.
 ↪2,random_state=0)
```

[24]:
```
params={"kernel":["rbf","linear"]}
cval=KFold(n_splits=5)
model=SVC()
```

[26]:
```
gsearch=GridSearchCV(model,params,cv=cval)
```

[27]:
```
gsearch.fit(x_train,y_train)
```

[27]:
```
GridSearchCV(cv=KFold(n_splits=5, random_state=None, shuffle=False),
             estimator=SVC(), param_grid={'kernel': ['rbf', 'linear']})
```

[28]:
```
gsearch.best_params_
```

[28]:
```
{'kernel': 'rbf'}
```

[29]:
```
bmodel=SVC(kernel="rbf")
```

[30]:
```
bmodel.fit(x_train,y_train)
```

[30]:
```
SVC()
```

[31]:
```
y_pred=bmodel.predict(x_test)
```

[32]:
```
y_pred
```

[32]:
```
array(['ham', 'spam', 'spam', …, 'spam', 'spam', 'spam'], dtype=object)
```

[33]:
```
accuracy_score(y_test,y_pred)
```

[33]:
```
0.9528497409326425
```

[37]:
```
confusion_matrix(y_test,y_pred)
```

[37]:
```
array([[874,  62],
       [ 29, 965]])
```

[38]:
```
print(classification_report(y_test,y_pred))
```

```
              precision    recall  f1-score   support

         ham       0.97      0.93      0.95       936
        spam       0.94      0.97      0.95       994
```

```
    accuracy                           0.95      1930
   macro avg       0.95      0.95      0.95      1930
weighted avg       0.95      0.95      0.95      1930
```

[40]:

```
emails=["Hey, you have won a car..!!!", "Dear Applicant, Your Cv has been recieved. Regards"]
```

[41]:

```
bmodel.predict(cvec.transform(emails))
```

[41]:

```
array(['spam', 'ham'], dtype=object)
```

[ ]:

```
```