optimizing spam filtering

File Edit View Insert Runtime Tools Help All changes saved

Comment Share

+ Code + Text

RAM Disk

```
[ ] import numpy as np # scientific computation
    import pandas as pd # loading dataset file
    import matplotlib.pyplot as plt # visulization
    import nltk # preprocessing our text
    from nltk.corpus import stopwords # removing all the stop words
    from nltk.stem.porter import PorterStemmer # stemming of words
```

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

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

```
[ ]
    df.info()
```

```
[ ]
    df.isna().sum()
```

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

```
[ ] # bootom 5 rows of the dataframe
```

optimizing spam filtering

File Edit View Insert Runtime Tools Help

Comment Share

+ Code + Text

RAM Disk

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

```
[ ] #Spliting data into train and validation sets using train_test_split

    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)

    ##train size 80% and test size 20%
```

```
[ ] print("Before OverSampling,counts of label'1':{}".format(sum(y_train==1)))
    print("Before OverSampling,counts of label'0':{}\n".format(sum(y_train==0)))
    #import SMOTE module from imblearn library
    #pip install imblearn(if you don't have imblearn in your system)
    from imblearn.over_sampling import SMOTE
    sm=SMOTE(randam_state = 2)
    X_train_res,y_train_res = sm.fit_resample(x_train,y_train.revel())

     print('After OverSampling, the shape of train_x:{}'.format(X_train_res.shape))
     print('After OverSampling, the shape of train_y:{} \n'.format(y_train_res.shape))

     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)))
```

✓ 0s completed at 7:51 PM

+ Code + Text

RAM
Disk

```
nltk.dowload("stopwords")
[nltk_data] Dowloading package stopwords to
[nltk_data]    c:\Users\smart\AppData\Roaming\nltk_data...
[nltk_data]  package stopwords is already up-to-date!
```

```
[12] import nltk
     from nltk.corpus import stopwords
     from nltk.stem import PorterStemmer
```

```
[4]  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)
```

```
[6]  from sklearn.feature_extraction.text import CountVectorizer
```

```
[ ]  print('Accuracy Score Is:-',score*100)

     cm1 = confusion_matrix(y_test,y_pred1)
     score1 = accuracy_score(y_test,y_pred1)
     print(cm1)
     print('Accuracy Score Is:-',score1*100)
```

```
[ ]  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)
```

```
[18] corpus

     []
```

```
[20] from sklearn.feature_extraction.text import CountVectorizer
     cv = CountVectorizer(max_features=35000)
     x = cv.fit_transform(carpus).toarray()
```

```
import pickle
pickle.dump(cv,open('cv1.pk1','wb'))
```