optimizing spam filtering - Colab

colab.research.google.com/drive/18MLbGNspuPS1MpogBeQmn20f7t7h6rQ#scrollTo=i5l0dlLa2TTn

optimizing spam filtering

File Edit View Insert Runtime Tools Help All changes saved

Comment Share

Files

sample_data

spam.csv

Disk 82.28 GB available

+ Code + Text

```
[40] 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()
```

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

```
[42]
     df.info()
```

```
[15]
     df.isna().sum()
```

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

```
[27] # bootom 5 rows of the dataframe
     df.tail()
```

```
[28] from sklearn.preprocessing import LabelEncoder
```

0s completed at 1:41 PM

33°C Partly sunny ENG IN 1:42 PM 9/23/2023

optimizing spam filtering

File Edit View Insert Runtime Tools Help All changes saved

Files

- ..
- sample_data
- spam.csv

```
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!
```

```
import nltk
from nltk.corpus import stopwords
from nltk.stem import PorterStemmer
```

```
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)
```

```
from sklearn.feature_extraction.text import CountVectorizer
cv = CountVectorizer(max_features=35000)
X = cv.fit_transfrom(corpus).toarray()
```

0s completed at 1:41 PM

optimizing spam filtering

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

[28]
```
from sklearn.preprocessing import LabelEncoder
le = LabelEncoder()
df['label'] = le.fit_transform(df['label'])
```

[29]
```
#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%
```

[30]
```
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(random_state = 2)
X_train_res,y_train_res = sm.fit_resample(x_train,y_train.ravel())

 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)))
```

[ ]
```
nltk.dowload("stopwords")
[nltk_data]Dowloading package stopwords to
```

optimizing spam filtering - Colab

colab.research.google.com/drive/

optimizing spam filtering

File Edit View Insert Runtime Tools Help Last saved at 1:42 PM

Comment Share

Files

Connecting to a runtime to enable file browsing.

+ Code + Text Allocating

```
from sklearn.feature_extraction.text import CountVectorizer
cv = CountVectorizer(max_features=35000)
X = cv.fit_transfrom(corpus).toarray()
```

```
df.describe()
```

```
df.shape
```

```
df["label"].value_counts().plot(kind="bar",figsize=(12,6))
plt.xticks(np.arange(2),('Non spam','spam'),rotation=0);
```

```
# perfroming feature Scaling op[eration using standard scaller on x part of the dataset because]
# there different type of values in the colums
sc=StandardScaler()
x_bal=sc.fit_transform(x_bal)
x_bal = pd.DataFrame(x_bal,columns=names)
```

```
#splitting 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%
```

```
from sklearn.tree import DecisionTreeClassifier
model = DecisionTreeClassifier()
model.fit(X_train_res,Y_train_res)
```

Waiting for clients6.google.com...

33°C Partly sunny ENG 1:56 PM