Untitled1.ipynb

File Edit View Insert Runtime Tools Help All changes saved

Comment Share

+ Code + Text

RAM Disk

```
[54] import numpy as np
     import pandas as pd
     import string
     from sklearn.model_selection import train_test_split, GridSearchCV
     from sklearn.metrics import classification_report, confusion_matrix, accuracy_score
     import nltk
     from nltk.corpus import stopwords
     from sklearn.feature_extraction.text import CountVectorizer, TfidfTransformer
     from sklearn.pipeline import Pipeline
     from sklearn.ensemble import RandomForestClassifier
     from sklearn.svm import SVC
     from sklearn.linear_model import LogisticRegression
```

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

```
Drive already mounted at /content/drive/; to attempt to forcibly remount, call drive.mount("/content/drive/", force_remount=True).
```

```
import pandas as pd
dataframe = pd.read_csv('/content/sample_data/fake review dataset - fake review dataset.csv')
dataframe.head()
```

	Unnamed: 0	category	rating	label	text_

✓ 0s completed at 11:36 AM

32°C Haze

Search

ENG IN

11:38 28-05-2023

Untitled1.ipynb

File Edit View Insert Runtime Tools Help All changes saved

Comment Share

+ Code + Text

RAM Disk

```
import pandas as pd
dataframe = pd.read_csv('/content/sample_data/fake review dataset - fake review dataset.csv')
dataframe.head()
```

	Unnamed: 0	category	rating	label	text_
0	0	Home_and_Kitchen_5	5	CG	love well made sturdi comfort i love veri pretti
1	1	Home_and_Kitchen_5	5	CG	love great upgrad origin i 've mine coupl year
2	2	Home_and_Kitchen_5	5	CG	thi pillow save back i love look feel pillow
3	3	Home_and_Kitchen_5	1	CG	miss inform use great product price i
4	4	Home_and_Kitchen_5	5	CG	veri nice set good qualiti we set two month

```
[13] dataframe.dropna(inplace=True)
dataframe['length'] = dataframe['text_'].apply(len)
dataframe[dataframe['label']=='OR'][['text_','length']].sort_values(by='length',ascending=False).head().iloc[0].text_
```

```
'weak on current scienc after see twice i agre much posit five star review out respect read review i 'll repeat everyth i like present i found goofi over ear hairdo facial hair
rrang daniel vitali describ `` wild food expert '' distract ugh ditto david wolf extrem goofi wild hairdo on hand jon gabriel describ `` author weight loss expert '' nice groom
ood present hi stori person transform fellow pound whew becom jock normal weight inspir christian northrup preserv rank one america 's cutest doctor a realli nice look woman pre
ent dr. mercola jason vale kri carr alejandro junger fine it disappoint jami oliv popular uk give babi cow growth fluid pas unscientif popular idea milk none present anyth zilch
say work doctor t. colin campbel milk bodi bad it good see present take stand sugar they agre evil sugar refin carbohydr with respect dr. northrup `` it 's fat make fat 's sugar
'' statement pas muster commun expert recogn evil sugar not mutual exclus recogn proven danger fat particularli...'
```

Untitled1.ipynb

File Edit View Insert Runtime Tools Help All changes saved

Comment Share

+ Code + Text

RAM Disk

[16]
```
def convertmyTxt(rv):
        np = [c for c in rv if c not in string.punctuation]
        np = ''.join(np)
        return [w for w in np.split() if w.lower() not in stopwords.words('english')]
```

[18]
```
from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test = train_test_split(dataframe['text_'],dataframe['label'],test_size=0.25)
```

[55]
```
from sklearn.pipeline import Pipeline
from sklearn.feature_extraction.text import CountVectorizer, TfidfTransformer
from sklearn.ensemble import RandomForestClassifier
pip = Pipeline([
    ('bow',CountVectorizer(analyzer=convertmyTxt)),
    ('tfidf',TfidfTransformer()),
    ('classifier',RandomForestClassifier())
])
```

[41]
```
import nltk
```

[42]
```
nltk.download('stopwords')
```

Untitled1.ipynb

File Edit View Insert Runtime Tools Help All changes saved

Comment Share

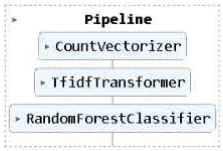
+ Code + Text

RAM Disk

```
[42] nltk.download('stopwords')
```

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

```
[29] pip.fit(x_train,y_train)
```




```
[30] randomForestClassifier = pip.predict(x_test)
     randomForestClassifier
```

```
array(['CG', 'CG', 'CG', ..., 'CG', 'CG', 'CG'], dtype=object)
```

```
[32] print('Accuracy of the model: ',str(np.round(accuracy_score(y_test,randomForestClassifier)*100,2)) + '%')
```

```
Accuracy of the model:  84.24%
```

0s completed at 11:36 AM

32°C Haze

Search

ENG IN

11:39 28-05-2023

Untitled1.ipynb

File Edit View Insert Runtime Tools Help All changes saved

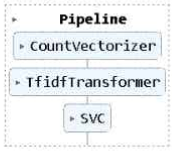
Comment Share

+ Code + Text

RAM Disk

```
[33] pip = Pipeline([
         ('bow',CountVectorizer(analyzer=convertmyTxt)),
         ('tfidf',TfidfTransformer()),
         ('classifier',SVC())
     ])
```

```
[34] pip.fit(x_train,y_train)
```




```
[36] supportVectorClassifier = pip.predict(x_test)
     supportVectorClassifier
```

```
array(['CG', 'CG', 'CG', ..., 'CG', 'OR', 'CG'], dtype=object)
```

```
[37] print('accuracy of the model:',str(np.round(accuracy_score(y_test,supportVectorClassifier)*100,2)) + '%')
```

```
accuracy of the model: 88.11%
```

0s completed at 11:36 AM

32°C Haze

Search

ENG IN

11:39 28-05-2023

Untitled1.ipynb

File Edit View Insert Runtime Tools Help All changes saved

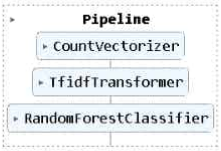
Comment Share

+ Code + Text

RAM Disk

```
[51] def text_process(text):
       processed_text = ...
       return processed_text
     pip = Pipeline([
         ('bow',CountVectorizer(analyzer=text_process)),
         ('tfidf',TfidfTransformer()),
         ('classifier',LogisticRegression())
     ])
```

```
[56] pip.fit(x_train,y_train)
```




```
[57] logisticRegression = pip.predict(x_test)
     logisticRegression
```

```
array(['CG', 'CG', 'CG', ..., 'CG', 'OR', 'CG'], dtype=object)
```

```
[58] print('accuracy of the model:',str(np.round(accuracy_score(y_test,logisticRegression)*100,2)) + '%')
```

```
accuracy of the model: 84.25%
```

0s completed at 11:36 AM

32°C Haze

Search

ENG IN

11:40 28-05-2023