language detection using SVM nlp

May 11, 2023

[1]:
```
import pandas as pd
```

[4]:
```
df = pd.read_csv(r'C:\Users\Rakesh\Downloads\Language Detection.csv')
df.head()
```

[4]:
```
                                                Text Language
0   Nature, in the broadest sense, is the natural…  English
1  "Nature" can refer to the phenomena of the phy…  English
2  The study of nature is a large, if not the onl…  English
3  Although humans are part of nature, human acti…  English
4  [1] The word nature is borrowed from the Old F…  English
```

[6]:
```
df.Language.value_counts()
```

[6]:
```
English       1385
French        1014
Spanish        819
Portugeese     739
Italian        698
Russian        692
Sweedish       676
Malayalam      594
Dutch          546
Arabic         536
Turkish        474
German         470
Tamil          469
Danish         428
Kannada        369
Greek          365
Hindi           63
Name: Language, dtype: int64
```

[7]:
```
df.isnull().sum()
```

[7]:
```
Text        0
Language    0
dtype: int64
```

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

```
[8]: Text        0
     Language    0
     dtype: int64
```

```
[10]: df.dtypes
```

```
[10]: Text        object
      Language    object
      dtype: object
```

```
[11]: x = df['Text']
      y = df['Language']
```

```
[13]: import re
      data = []
      for text in x:
          text = re.sub(r'[!@#$(),n"%^*?:;~`0-9]','',text)
          text = re.sub(r'[[]]', '',text)
          text = text.lower()
          data.append(text)
```

```
C:\Users\Rakesh\AppData\Local\Temp\ipykernel_7200\1037316791.py:5:
FutureWarning: Possible nested set at position 1
  text = re.sub(r'[[]]', '',text)
```

```
[14]: x = data
```

```
[15]: len(x)
```

```
[15]: 10337
```

```
[17]: from sklearn.model_selection import train_test_split
      x_train,x_test,y_train,y_test = train_test_split(x,y, test_size=0.
       ↪3,random_state=40)
```

```
[18]: print(len(x_train))
      print(len(x_test))
```

```
7235
3102
```

```
[23]: from sklearn.pipeline import Pipeline
      from sklearn.feature_extraction.text import TfidfVectorizer


      from sklearn.svm import LinearSVC
      lang = Pipeline([('tfid' , TfidfVectorizer()),('clf',LinearSVC())])
```

```
lang.fit(x_train,y_train)
```

```
[23]: Pipeline(steps=[('tfid', TfidfVectorizer()), ('clf', LinearSVC())])
```

```
[26]: lang.predict(['hi how are you'])
```

```
[26]: array(['English'], dtype=object)
```

```
[35]: lang.predict(['    '])
```

```
[35]: array(['Kannada'], dtype=object)
```

```
[29]: predict = lang.predict(x_test)
```

```
[30]: from sklearn.metrics import confusion_matrix, classification_report,␣
       ↪accuracy_score
      print(confusion_matrix(y_test,predict))
```

```
[[152   0   0   0   0   0   0   0   0   0   0   0   8   0   0   0   0]
 [  0 122   2   0   0   1   0   0   2   0   0   0   3   0   5   0   0]
 [  0   0 162   0   0   0   0   0   0   0   0   0   4   0   0   0   0]
 [  0   0   0 381   0   0   0   0   2   0   1   0   2   0   0   0   0]
 [  0   0   0   2 290   0   0   0   1   0   0   0   4   0   0   0   0]
 [  0   0   0   0   0 135   0   0   0   0   0   0   1   0   1   0   0]
 [  0   0   0   0   0   0 113   0   0   0   0   0   3   0   0   0   0]
 [  0   0   0   0   0   0   0  17   0   0   0   0   0   0   0   0   0]
 [  0   0   0   0   0   0   0   0 206   0   0   0   5   4   0   0   0]
 [  0   0   0   0   0   0   0   0   0 117   0   0   5   0   0   0   0]
 [  0   0   0   0   0   0   0   0   0   0 167   0   6   0   0   0   0]
 [  0   0   0   0   1   0   0   0   0   0   0 186   2   5   0   0   0]
 [  0   0   0   0   0   0   0   0   0   0   0   0 214   0   0   0   0]
 [  0   0   0   1   0   0   0   0   2   0   0   1   0 248   0   0   0]
 [  0   3   0   0   0   1   0   0   0   0   0   0   3   0 204   0   1]
 [  0   0   0   0   0   0   0   0   0   0   0   0   1   0   0 155   0]
 [  0   0   0   0   1   0   0   0   1   0   0   0   9   0   0   0 139]]
```

```
[31]: print(classification_report(y_test,predict))
```

```
              precision    recall  f1-score   support

      Arabic       1.00      0.95      0.97       160
      Danish       0.98      0.90      0.94       135
       Dutch       0.99      0.98      0.98       166
     English       0.99      0.99      0.99       386
      French       0.99      0.98      0.98       297
      German       0.99      0.99      0.99       137
       Greek       1.00      0.97      0.99       116
       Hindi       1.00      1.00      1.00        17
     Italian       0.96      0.96      0.96       215
     Kannada       1.00      0.96      0.98       122
   Malayalam       0.99      0.97      0.98       173
  Portugeese       0.99      0.96      0.98       194
     Russian       0.79      1.00      0.88       214
     Spanish       0.96      0.98      0.97       252
    Sweedish       0.97      0.96      0.97       212
       Tamil       1.00      0.99      1.00       156
     Turkish       0.99      0.93      0.96       150

    accuracy                           0.97      3102
   macro avg       0.98      0.97      0.97      3102
weighted avg       0.97      0.97      0.97      3102
```

[33]:
```
print(accuracy_score(y_test,predict))
```

```
0.9696969696969697
```

[ ]:
```
```