Language_Detection

June 8, 2023

1 Problem Statement: Langauge Detection Uisng Machine Learning

1.1 Description:

- **Language detection is a natural language processing (NLP) task that involves identifying the language of a text or document. It is a challenging task, as there are over 7,000 known languages in the world, and many of them are very similar.**
- **There are a number of different machine learning techniques that can be used for language detection. One common approach is to use a statistical model that is trained on a large corpus of text in multiple languages. The model learns to identify the features that are characteristic of each language, and then uses these features to predict the language of an unknown text.**
- **Another approach to language detection is to use a neural network. Neural networks are a type of machine learning algorithm that can learn complex relationships between features. Neural networks have been shown to be very effective for language detection, and they are often used in commercial products such as Google Translate.**
- **In this notbook we are going to use simple machine learning algorithm *MutinomialNB* let's go.**

2 Importing Libraries

[ ]:
```
import pandas as pd
import numpy as np
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.model_selection import train_test_split
from sklearn.naive_bayes import MultinomialNB
```

3 Reading Datasets

[ ]:
```
data=pd.read_csv("https://raw.githubusercontent.com/amankharwal/Website-data/
  ↪master/dataset.csv")
data.head()
```

[ ]:
```
                                                  Text  language
0  klement gottwaldi surnukeha palsameeriti ning …  Estonian
1  sebes joseph pereira thomas  på eng the jesuit…   Swedish
2              thanon charoen krung …       Thai
3                         …     Tamil
4  de spons behoort tot het geslacht haliclona en…     Dutch
```

3.1 Saving Datasets

[ ]:
```
data.to_csv("Language.csv",index=False)
```

4 Exploring Datasets

[ ]:
```
data.shape
```

[ ]:
```
(22000, 2)
```

[ ]:
```
data['language'].value_counts()
```

[ ]:
```
Estonian      1000
Swedish       1000
English       1000
Russian       1000
Romanian      1000
Persian       1000
Pushto        1000
Spanish       1000
Hindi         1000
Korean        1000
Chinese       1000
French        1000
Portugese     1000
Indonesian    1000
Urdu          1000
Latin         1000
Turkish       1000
Japanese      1000
Dutch         1000
Tamil         1000
Thai          1000
Arabic        1000
Name: language, dtype: int64
```

[ ]:
```
data['language'].value_counts().unique()
```

[ ]:
```
array([1000])
```

- **This dataset contains 22 languages with 1000 sentences from each language. This is a very balanced dataset with no missing values, so we can say this dataset is completely ready to be used to train a machine learning model.**

5 Dependent and Independent

[ ]:
```
x=np.array(data['Text']) #independent
y = np.array(data["language"]) # dependent
```

[ ]:
```
x
```

[ ]:
```
array(['klement gottwaldi surnukeha palsameeriti ning paigutati mausoleumi
surnukeha oli aga liiga hilja ja oskamatult palsameeritud ning hakkas ilmutama
lagunemise tundemärke  aastal viidi ta surnukeha mausoleumist ära ja kremeeriti
zlíni linn kandis aastatel - nime gottwaldov ukrainas harkivi oblastis kandis
zmiivi linn aastatel - nime gotvald',
       'sebes joseph pereira thomas  på eng the jesuits and the sino-russian
treaty of nerchinsk  the diary of thomas pereira bibliotheca instituti historici
s i --   rome libris ',
       '           thanon charoen krung


                                  ',
       …,
       'con motivo de la celebración del septuagésimoquinto ° aniversario de la
fundación del departamento en  guillermo ceballos espinosa presentó a la
gobernación de caldas por encargo de su titular dilia estrada de gómez el himno
que fue adoptado para solemnizar dicha efemérides y que siguieron interpretando
las bandas de música y los planteles de educación de esta sección del país en
retretas y actos oficiales con gran aceptación[]\u200b',
       '          mai-k     baby i
like     bip·record                          love
day after
tomorrow                                    ',
       ' aprilie sonda spațială messenger a nasa și-a încheiat misiunea de
studiu de  ani prăbușindu-se pe suprafața planetei mercur sonda a rămas fără
combustibil fiind împinsă de gravitația solară din ce în ce mai aproape de
mercur'],
      dtype=object)
```

[ ]:
```
y
```

[ ]:
```
array(['Estonian', 'Swedish', 'Thai', …, 'Spanish', 'Chinese',
       'Romanian'], dtype=object)
```

6 Model Training

- CountVectorizer is a class in the scikit-learn library that is used to convert a collection of text documents into a matrix of token counts. This matrix is called a bag-of-words (BoW) representation.

[ ]:
```
cv = CountVectorizer()
X = cv.fit_transform(x)
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.33,␣
 ↪random_state=42)
```

- **As this is a problem of multiclass classification, so I will be using the Multinomial Naïve Bayes algorithm to train the language detection model as this algorithm always performs very well on the problems based on multiclass classification:**

[ ]:
```
model = MultinomialNB()
model.fit(X_train,y_train)
model.score(X_test,y_test)
```

[ ]:
```
0.953168044077135
```

7 Testing

[ ]:
```
user = input("Enter a Text: ")
data = cv.transform([user]).toarray()
output = model.predict(data)
print(user)
print(output)
```

```
Hellow World!
['English']
```

[ ]:
```
user = input("Enter a Text: ")
data = cv.transform([user]).toarray()
output = model.predict(data)
print(user) #
print(output)
```

```
['Hindi']
```

[ ]:
```
user = input("Enter a Text: ")
data = cv.transform([user]).toarray()
output = model.predict(data)
print(user) #
print(output)
```

```
['Chinese']
```

- **So as you can see that the model performs well. One thing to note here is that this model can only detect the languages mentioned in the dataset.**

8 Converting Model into Pickle File

9 Saving Model as Pickle

[ ]:
```
import pickle

# Create a model
model = model

# Save the model to a file
with open('model.pikle', 'wb') as f:
    pickle.dump(model, f)
```

10 Thank You !