In [1]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import re

import string

from sklearn.model_selection import train_test_split
from sklearn import feature_extraction # for vectorizer

from sklearn import pipeline
from sklearn.metrics import accuracy_score,confusion_matrix
```

In [2]:

```
df = pd.read_csv("Language Detection.csv")
df.head()
```

Out[2]:

	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

In [3]:

```
string.punctuation # this command wil give us all type of punctuations, we have to rem
```

Out[3]:

```
'!"#$%&\'()*+,-./:;<=>?@[\\]^_`{|}~'
```

In [4]:

```
def remove_pun(text):  # we are going to remove punctuations from text column
    for pun in string.punctuation:
        text = text.replace(pun,"")
    text=text.lower() # will make text in lowercase
    return(text)
```

In [5]:

```
# lets check
remove_pun("Although humans are part of nature,! human & *acti..")
# function is working
```

Out[5]:

```
'although humans are part of nature human  acti'
```

In [6]:

```
df["Text"].apply(remove_pun)
```

Out[6]:

```
0         nature in the broadest sense is the natural p...
1        nature can refer to the phenomena of the physi...
2        the study of nature is a large if not the only...
3        although humans are part of nature human activ...
4        1 the word nature is borrowed from the old fre...
                               ...                        
10332    ನಿಮ್ಮ ತಪ್ಪು ಏನು ಬಂದಿದೆಯೆಂದರೆ ಆ ದಿನದಿಂದ ನಿಮಗೆ ಒ....
10333    ನಾರ್ಸಿಸಾ ತಾನು ಮೊದಲಿಗೆ ಹೆಣಗಾಡುತ್ತಿದ್ದ ಮಾರ್ಗಗಳನ್...
10334    ಹೇಗೆ  ನಾರ್ಸಿಸಿಸಮ್ ಈಗ ಮರಿಯನ್ ಅವರಿಗೆ ಸಂಭವಿಸಿದ ಎಲ...
10335    ಅವಳು ಈಗ ಹೆಚ್ಚು ಚಿನ್ನದ ಬ್ರೆಡ್ ಬಯಸುವುದಿಲ್ಲ ಎಂದು ...
10336    ಟೆರ್ರಿ ನೀವು ನಿಜವಾಗಿಯೂ ಆ ದೇವದೂತನಂತೆ ಸ್ವಲ್ಪ ಕಾಣು...
Name: Text, Length: 10337, dtype: object
```

In [7]:

```
# lets save it
df["Text"]=df["Text"].apply(remove_pun)
```

In [8]:

```
df.head()
```

Out[8]:

	Text	Language
0	nature in the broadest sense is the natural p...	English
1	nature can refer to the phenomena of the physi...	English
2	the study of nature is a large if not the only...	English
3	although humans are part of nature human activ...	English
4	1 the word nature is borrowed from the old fre...	English

In [ ]:

```

```

now we have to divide out dataset into training and testing ,split

In [9]:

```
x=df.iloc[:,0] # selecting all rows with column 0
y=df.iloc[:,1] # selecting all rows with column 1
```

In [10]:

```
x
```

Out[10]:

```
0         nature in the broadest sense is the natural p...
1        nature can refer to the phenomena of the physi...
2        the study of nature is a large if not the only...
3        although humans are part of nature human activ...
4        1 the word nature is borrowed from the old fre...
                               ...                        
10332    ನಿಮ್ಮ ತಪ್ಪು ಏನು ಬಂದಿದೆಯೆಂದರೆ ಆ ದಿನದಿಂದ ನಿಮಗೆ ಒ....
10333    ನಾರ್ಸಿಸಾ ತಾನು ಮೊದಲಿಗೆ ಹೆಣಗಾಡುತ್ತಿದ್ದ ಮಾರ್ಗಗಳನ್...
10334    ಹೇಗೆ  ನಾರ್ಸಿಸಿಸಮ್ ಈಗ ಮರಿಯನ್ ಅವರಿಗೆ ಸಂಭವಿಸಿದ ಎಲ...
10335    ಅವಳು ಈಗ ಹೆಚ್ಚು ಚಿನ್ನದ ಬ್ರೆಡ್ ಬಯಸುವುದಿಲ್ಲ ಎಂದು ...
10336    ಟೆರ್ರಿ ನೀವು ನಿಜವಾಗಿಯೂ ಆ ದೇವದೂತನಂತೆ ಸ್ವಲ್ಪ ಕಾಣು...
Name: Text, Length: 10337, dtype: object
```

In [11]:

```
y
```

Out[11]:

```
0        English
1        English
2        English
3        English
4        English
          ...
10332    Kannada
10333    Kannada
10334    Kannada
10335    Kannada
10336    Kannada
Name: Language, Length: 10337, dtype: object
```

Training and Testing

In [12]:

```
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.2)
```

In [13]:

```
# now we have to convert the text(string) in some features or numerical value to pass
# we will use vectorizer or TF-IDF
```

In [14]:

```
vec = feature_extraction.text.TfidfVectorizer(ngram_range=(1,2),analyzer="char")
```

In [15]:

```
#pipeline to used to create a continous flow f ofunctions and select the algorithm by
```

model selection and model evaluation

In [16]:

```
from sklearn import pipeline
from sklearn import linear_model
```

In [17]:

```
model_pipe=pipeline.Pipeline([("vec",vec),("clf",linear_model.LogisticRegression())])
```

In [18]:

```
#pip install pipeline
```

In [19]:

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

Out[19]:

```
Pipeline(steps=[('vec', TfidfVectorizer(analyzer='char', ngram_range=(1, 2))),
                ('clf', LogisticRegression())])
```

In [20]:

```
model_pipe.classes_  # we can our model component in y_variable
```

Out[20]:

```
array(['Arabic', 'Danish', 'Dutch', 'English', 'French', 'German',
       'Greek', 'Hindi', 'Italian', 'Kannada', 'Malayalam', 'Portugeese',
       'Russian', 'Spanish', 'Sweedish', 'Tamil', 'Turkish'], dtype=object)
```

Testing dataset

In [21]:

```
# now lets analyze the model
```

In [22]:

```
y_pred_test = model_pipe.predict(x_test)
```

In [23]:

```
# lets calculate accuracy
```

In [24]:

```
from sklearn.metrics import accuracy_score,confusion_matrix
```

In [25]:

```
Accuracy = accuracy_score(y_test,y_pred_test)
print("accuracy",Accuracy*100)
```

```
accuracy 98.21083172147002
```

In [ ]:

In [ ]:

lets test model

In [26]:

```
model_pipe.predict(["My Name is Swapnil"])
```

Out[26]:

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

In [27]:

```
model_pipe.predict(["how are you?"])
```

Out[27]:

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

In [28]:

```
model_pipe.predict(["हमारे देश में त्यौहार का जाल बिछा हुआ है"])
```

Out[28]:

```
array(['Hindi'], dtype=object)
```

In [29]:

```
model_pipe.predict([" மற்றும் பொதுக் கட்டுரைகளின் தொகுப்பு"])
```

Out[29]:

```
array(['Tamil'], dtype=object)
```

In [30]:

```
model_pipe.predict(["ഇന്ത്യയിൽ കേരള സംസ്ഥാനത്തിലും"])
```

Out[30]:

```
array(['Malayalam'], dtype=object)
```

In [ ]:

now we have to set this model as a pickle file

In [45]:

```
import pickle
```

In [50]:

```
new_file=open("model.pckl","wb")
pickle.dump(model_pipe,new_file)
new_file.close()
```

In [ ]:

In [ ]: