LANGUAGE DETECTION USING ML

1. Introduction:

The project aims to classify the language of a given text using a Naive Bayes classifier. Natural Language Processing (NLP) techniques are employed to preprocess the text data, and the Multinomial Naive Bayes algorithm is utilized for classification. The dataset used for training and testing the model consists of text samples labeled with their respective languages.

PROGRAM:

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

data = pd.read\_csv("https://raw.githubusercontent.com/amankharwal/Websi te-data/master/dataset.csv")

print(data.head())

OUTPUT:

Text language

0 klement gottwaldi surnukeha palsameeriti ning ... Estonian

1 sebes joseph pereira thomas på eng the jesuit... Swedish

2 ถนนเจริญกรุง อักษรโรมัน thanon charoen krung เ... Thai

3 விசாகப்பட்டினம் தமிழ்ச்சங்கத்தை இந்துப் பத்திர... Tamil

4 de spons behoort tot het geslacht haliclona en... Dutch

2. Data Collection and Preprocessing:

The dataset is fetched directly from a CSV file hosted on GitHub using pandas.

Preliminary data exploration is performed by examining the first few rows of the dataset to understand its structure.

Null value detection is conducted to ensure data integrity.

The distribution of languages in the dataset is analyzed to ascertain class balance.

PROGRAM:

data.isnull().sum()

OUTPUT:

Text 0

language 0

dtype: int64

3. Feature Engineering:

The textual data is converted into numerical features using the CountVectorizer from scikit-learn. This step transforms the text into a matrix of token counts.

The dataset is split into training and testing sets using the train\_test\_split function from scikit-learn.

PROGRAM:

data["language"].value\_counts()

OUTPUT:

language

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: count, dtype: int64

4. Model Training:

A Multinomial Naive Bayes classifier is chosen for its effectiveness in handling discrete features, making it suitable for text classification tasks.

The model is trained on the training dataset using the fit method.

PROGRAM:

x = np.array(data["Text"])

y = np.array(data["language"])

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)

5. Model Evaluation:

The trained model's performance is evaluated on the testing dataset using the score method, which computes the accuracy of the model.

Accuracy serves as the primary evaluation metric in this project.

PROGRAM:

model = MultinomialNB()

model.fit(X\_train,y\_train)

model.score(X\_test,y\_test)

OUTPUT:

0.953168044077135

6. User Interaction:

The user is prompted to input a text string for language classification.

The input text is transformed into numerical features using the same CountVectorizer instance used during training.

The trained model predicts the language of the input text.

The predicted language is displayed as output.

PROGRAM:

user = input("Enter a Text: ")

data = cv.transform([user]).toarray()

output = model.predict(data)

print(output)

OUTPUT:

Enter a Text: नमस्कार धन्यवाद सुप्रभात

['Hindi']

7. Conclusion:

The project demonstrates the application of Naive Bayes classification for language detection. By leveraging NLP techniques and machine learning algorithms, the model accurately identifies the language of a given text input. The simplicity and efficiency of the Naive Bayes approach make it a viable solution for real-world language detection tasks.

# PROJECT REPORT

# TEAM BY

MANOJKUMAR. P

SENTHIL KUMAR .K .S

SUKESHRAMAN .S

VELMURUGAN .M