18CSC305J - Artificial Intelligence Course Project

Project Title: Language Detection using Naive Bayes algorithm

Github upload:**https://github.com/naveen-parth/Language-detection-using-Naive-Bayes-Classifier**

PENDYALA NAGA KASI SAI RAM - RA191100301696
KISHORE REDDY - RA1911003010700
NAVEEN PARTHASARATHY - RA1911003010718

Problem statement: A language detection program to analyse and predict the language of the text being entered using the Naive Bayes predictive classifier algorithm. The main agenda of this project is to squash the language barrier while communicating with individuals.

Contents of the project manual

1. Problem Statement:

The basic idea behind this project is to make language detection easier using deep learning techniques. A custom text being entered into this compiler, accurately predicts the language being used in the input text.

This is project is made using Python and the libraries used are
Numpy: Basic mathematical operations library of python
Pandas: Data analysis and manipulation
Matplotlib: For data visualization
Seaborn: For 3 dimensional, interactive data visualization charts
Warnings: To accurately predict the issues popping up while running the script

2. Objectives:

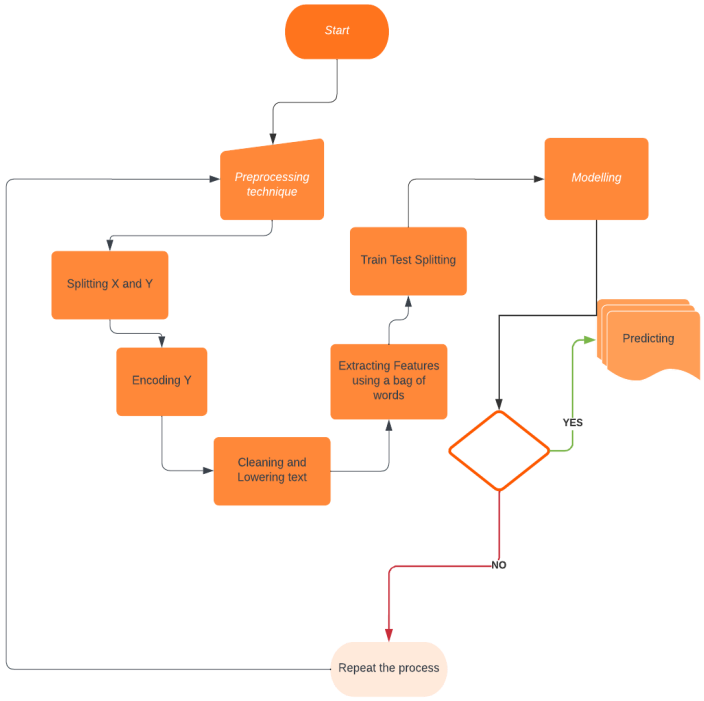
Data Preprocessing

a) Splitting X and Y
b) Encoding Y
c) Cleaning and Lowering text
d) Extracting features using a bag of words
e) Train Test Splitting

Modelling

Predicting

3. Flow Diagram:



Algorithm:

- Start the predicting by categorizing the information that is to be used for training the model
- Enter the Preprocessing phase, by first establishing the data types of the dataset items being used for easier classification
- The First Pre-processing step is to Split the data into X and Y variables, each representing a group of values in the specified axis
- Followed by Encoding Y into numerical values for feeding the data into the machine learning model
- For easier classification, 22 languages are considered and the languages are given a numerical value from 0 to 21

- Cleaning and Lowering texts involve separation of the text entered into alphabetical and non-alphabetical components and all the non-alphabetical components are considered as spaces
- Extracting features using a bag of words is categorising the text into vectors where each feature will be a word
- The processed table of vectors is further sent for Splitting Train and Testing Data phase where the table is used to train the machine learning model
- The modelling phase specifies the type of algorithm to be used i.e. Naive Bayes Classifier algorithm and applies the same for predicting the language detection accuracy. Depending on the results, the model is trained further to provide the max accurate results
- Now the model asks for an input which is then converted to a bag of words
- This vector bag predicts the language depending on the dataset used for training it and then finds the predicted value
- The predicted value corresponds to the assigned numerical value during encoding and thus returns the numerical value which is then decoded to give the language used
- For a better understanding, the actual language must be specified to the results can be compared to the predicted value given by the model implemented which would maximise the prediction accuracy levels

4. Execution:

Basics: Importing the required libraries from python

Importing Required Libraries

```
import pandas as pd
import numpy as np
import re
import seaborn as sns
import matplotlib.pyplot as plt
import warnings
warnings.simplefilter("ignore")
```

Numpy: Basic mathematical operations library of python
Pandas: Data analysis and manipulation
Matplotlib: For data visualization
Seaborn: For 3 dimensional, interactive data visualization charts
Warnings: To accurately predict the issues popping up while running the script

Importing information in a dataset
After data selection and preprocessing, 22 selective languages from the original dataset Which Includes the following Languages

- English
- Arabic
- French
- Hindi
- Urdu
- Portuguese
- Persian
- Pushto
- Spanish
- Korean
- Tamil

- Turkish
- Estonian
- Russian
- Romanian
- Chinese
- Swedish
- Latin
- Indonesian
- Dutch
- Japanese
- Thai

```
data = pd.read_csv("dataset.csv")
data.head(10)
```

	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
5	エノが行きがかりでバスに乗ってしまい、気分が悪くなった際に助けるが、今すぐバスを降りたいと運...	Japanese
6	tsutinalar İngilizce tsuutina kanadada albert...	Turkish
7	müller mox figura centralis circulorum doctoru...	Latin
8	برقی بار electric charge کی ذرات زیرجوہری تمام...	Urdu
9	シャーリー・フィールドは、サン・ベルナルド・アベニュー沿い市民センターとrtマーティン高校に...	Japanese

Reading the dataset to know the data types and memory allocation of the same

Info

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 22000 entries, 0 to 21999
Data columns (total 2 columns):
 #   Column    Non-Null Count  Dtype
---  ------    --------------  -----
 0   Text      22000 non-null  object
 1   language  22000 non-null  object
dtypes: object(2)
memory usage: 343.9+ KB
```

Data Preprocessing

1. **Splitting X and Y:** Specifying the parameters for easier representation and classification of the objects used

Splitting X and Y

```
X = data["Text"]
y = data["language"]
```

2. **Encoding Y**: Machine learning models can only work with numerical values. For this reason, it is necessary to transform the categorical values of the relevant features into numerical ones. This process is called feature encoding.

Here, we are encoding all languages(y) to numerical values to feed into our ML model.

```
from sklearn.preprocessing import LabelEncoder
le = LabelEncoder()
y = le.fit_transform(y)
```

```
print(np.unique(y))
```

```
[ 0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21]
```

```
print(len(np.unique(y)))
```

```
22
```

Transforming the Y component and we're selecting 22 languages out of all possible combinations of languages used to train the model and thus we're assigning each language a unique numerical value from 0-to 21

3. **Cleaning and Lowering Text**: In this phase, the text received is cleaned by eliminating all the values other than alphabetical values and replacing those replaced values with a space

Cleaning and Lowering Text

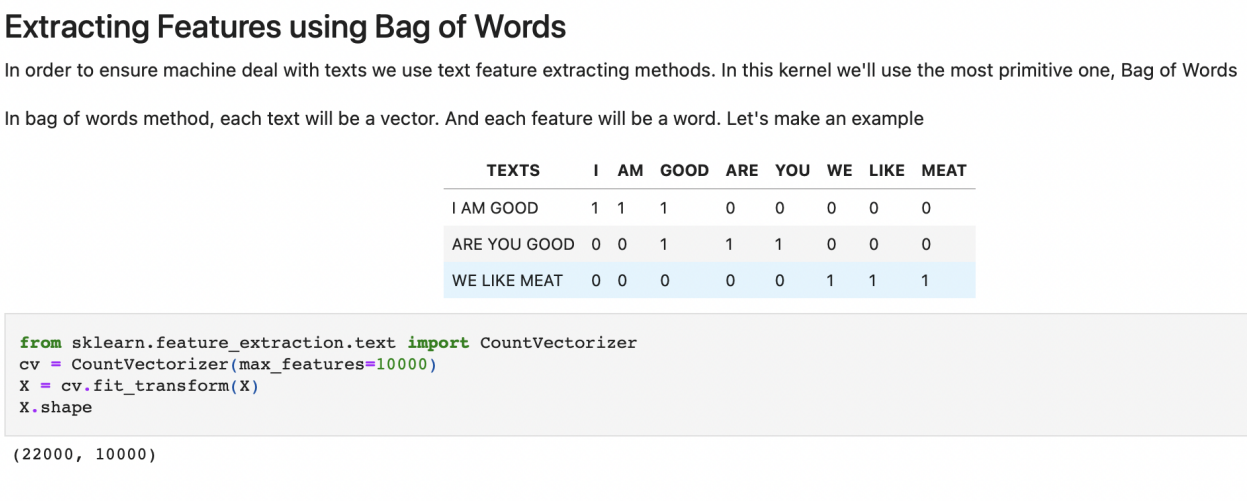
```
def cleanLower(texts):
    pattern = "^[a-zA-Z]"
    cleanText = []
    for text in texts:
        # re.sub(pattern) means replace everything with a space except alphabetical characters
        cleanText.append(re.sub(pattern," ",text).lower())

    return cleanText

X = cleanLower(X)
X[:10]
```

```
[' lement 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',
 ' ebes 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 ',
 'ถนนเจริญกรุง อักษรโรมัน thanon charoen krung เริ่มตั้งแต่ถนนสนามไชยถึงแม่น้ำเจ้าพระยาที่ถนนตก กรุงเทพมหานคร เป็นถนนรุ่นแรกที่ใช้เทคนิคการสร้างแบบตะวันตก ปัจจุบันผ่านพื้นที่เขตพระนคร เขตป้อมปราบศัตรูพ่าย เขตสัมพันธวงศ์ เขตบางรัก เขตสาทร และเขตบางคอแหลม',
 'விசாகப்பட்டினம் தமிழ்ச்சங்கத்தை இந்துப் பத்திரிகை-விசாகப்பட்டின ஆசிரியர் சம்பத்துடன் இணைந்து விரிவுபடுத்தினார்  ஆண்டுகள் தொடர்ந்து செயலராக இருந்து தமிழ்மன்றத்தை நடத்திச் சென்றார்  கோவை செம்மொழி மாநாட்டில் "தமிழ்ச்சங்கங்களும் தமிழும்" எனும் தலைப்பில் பிற மாநிலங்களில் தமிழ்வளர்ச்சி பற்றி கட்டுரை வாசித்தார்',
 ' e spons behoort tot het geslacht haliclona en behoort tot de familie chalinidae de wetenschappelijke naam van de soort werd voor het eerst geldig gepubliceerd in  door kudelin',
 'エノが行きがかりでバスに乗ってしまい、気分が悪くなった際に助けるが、今すぐバスを降りたいと運転手に頼む際、本当のことを言ってしまうと彼女が恥ずかしい思いをすると察して「僕ウンコしたいんです」と言ってバスを降りた。エノは内心「私もしたいみたいじゃないの」と思うも、別れ際にエノの髪を「ふわふわのお菓子みたい」と言い、この台詞に憧れていたエノに強い衝撃を与えた。この話を聞いたリコは、以後彼のことを『ウンコ王子』または『ウンコ』というあだ名で呼ぶようになったが、エノは普通に「戸田くん」と呼んでいる。',
 ' sutinalar İngilizce tsuutina kanadada alberta bölgesinde calgaryde yaşarlar tek başına grup oluştururlar ve pasifik ve güney atabaskları ile antik yakınlıklar göstermiştir',
 ' üller mox figura centralis circulorum doctorum vindobonesium fiebat quibus intererant petrus altenberg albertus ehrenstein egon friedell anscharius kokoschka adolphus loos egon schiele et alii',
 'برقی بار electric charge تمام زیرجوہری ذرات کی ا یک بنیادی محفوظ conserved اس با خصوصیت ہے جو انکے برقناطیسی تفاعلات کا تعین کرتی ہے۔
ت یوں بھی کہ سکتے ہیں کہ برقی بار کا حامل ذرہ یا مادہ  ایک برقناطیسی میدان کی وجہ سے متاثر ہوتا ہے یا اسکے زیر اثر ہوتا ہے جبکہ خود اسی
سے ا interaction کی وجہ سے برقناطیسی میدان پیدا بھی ہوتا ہے۔ برقناطیسی میدان اور اس میں حرکت کرتے ہوئے ایک بار دار ذرے کے آپس میں تفاعل
یک قسم کی قوت یا توانائی نمودار ہوتی ہے جس کو برقناطیسی قوت کہا جاتا ہے۔ اور یہ برقناطیسی قوت  دراصل اس کائنات میں پائی جانے والی چار ب
نیادی قوتوں میں سے ایک ہے۔',
 'シャーリー・フィールドは、サン・ベルナルド・アベニュー沿い市民センターとrtマーティン高校に隣接してある。年にマーティン高校と共に建設された。ラレド独立学区のアウトドア・アスレチックがあり、毎年ボーダー・オリンピックが開催される。通常席だが、エンドゾーンに追加席を加えることで席まで拡大できる。メキシコのプロ・サッカークラブがここで様々な練習試合を行い、天然芝が「良い」サッカー試合を作るとしている。フットボール、サッカーおよび陸上競技が行われる。この歴史ある競技場に大きな改修工事が計画されている。']
```

Using seaborn.countplot, we're comparing the number of samples encoded with the language. In the graph, the language is represented using numerical values ranging from 0-to 21 on the x-axis and the samples encoded on the y-axis. The slopes represent the instances created of each language through modelling and training the data.

4. **Extracting features using a bag of words**: To ensure machines deal with texts we use text feature extracting methods. In this kernel, we'll use the most primitive one, Bag of Words

In the bag of words method, each text will be a vector. And each feature will be a word. Let's take an example



Extracting Features using Bag of Words

In order to ensure machine deal with texts we use text feature extracting methods. In this kernel we'll use the most primitive one, Bag of Words

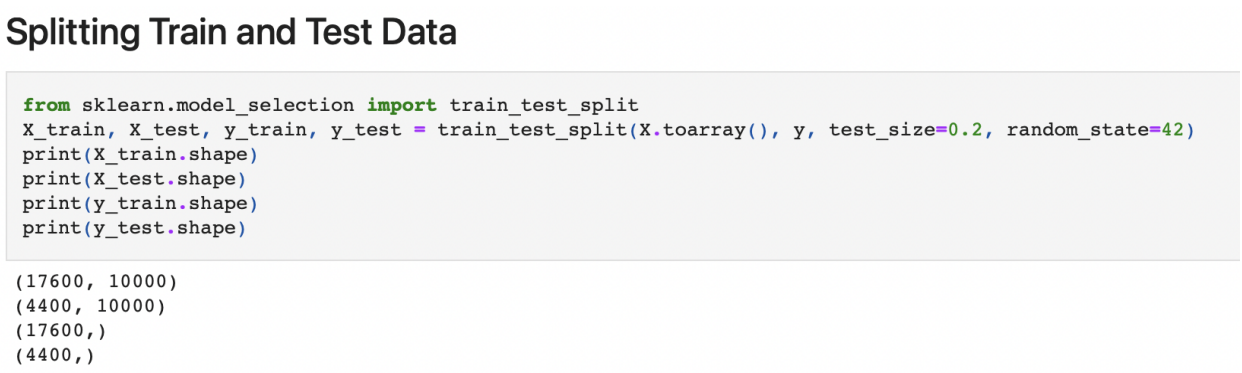
In bag of words method, each text will be a vector. And each feature will be a word. Let's make an example

TEXTS	I	AM	GOOD	ARE	YOU	WE	LIKE	MEAT
I AM GOOD	1	1	1	0	0	0	0	0
ARE YOU GOOD	0	0	1	1	1	0	0	0
WE LIKE MEAT	0	0	0	0	0	1	1	1

```
from sklearn.feature_extraction.text import CountVectorizer
cv = CountVectorizer(max_features=10000)
X = cv.fit_transform(X)
X.shape
```

```
(22000, 10000)
```

5. **Splitting and Training Data**:



Splitting Train and Test Data

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X.toarray(), y, test_size=0.2, random_state=42)
print(X_train.shape)
print(X_test.shape)
print(y_train.shape)
print(y_test.shape)
```

```
(17600, 10000)
(4400, 10000)
(17600,)
(4400,)
```

Modelling

Naive Bayes algorithm has been used for training the model.
Naive stands for each feature being independent of each other and the weight of the features is of equal importance
Bayes theorem: Bayes' Theorem finds the probability of an event occurring given the probability of another event that has already occurred. Bayes' theorem is stated mathematically as the following equation:

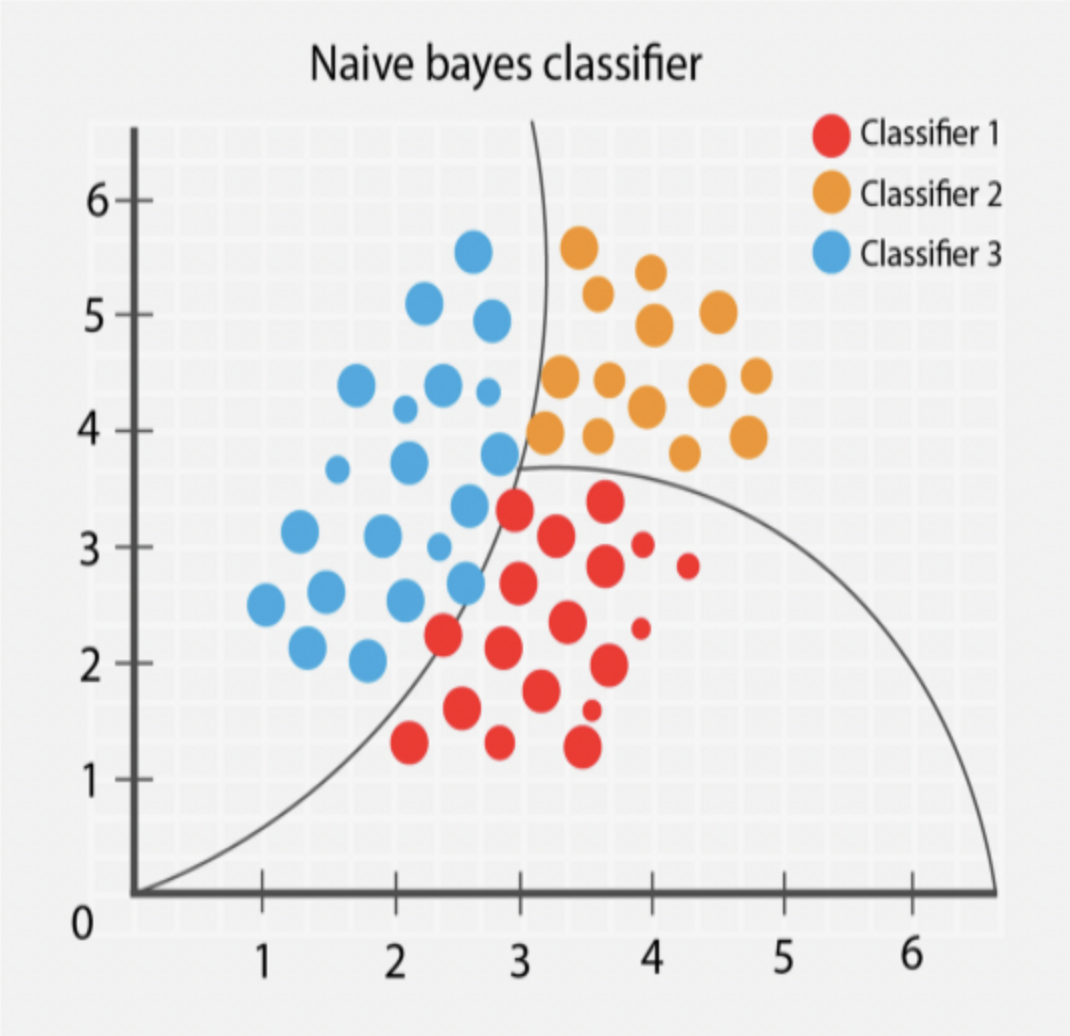
where A and B are events and $P(B) \neq 0$.

- We are trying to find the probability of event A, given that event B is true. Event B is also termed as **evidence**.
- P(A) is the **priori** of A (the prior probability, i.e. Probability of event before evidence is seen). The evidence is an attribute value of an unknown instance(here, it is event B).
- P(A|B) is a **posteriori** probability of B, i.e. probability of event after evidence is seen

$$P(A|B) = \frac{P(B|A)\ P(A)}{P(B)}$$

using Bayesian probability terminology, the above equation can be written as

$$\text{Posterior} = \frac{\text{prior x likelihood}}{\text{evidence}}$$



The multinomial Naive Bayes model is used here, that's because the data is multinomial distributed. It is suitable for classification with discrete features (e.g., word counts for text classification).

```
from sklearn.naive_bayes import MultinomialNB
model = MultinomialNB()
model.fit(X_train, y_train)
```

```
MultinomialNB()
```

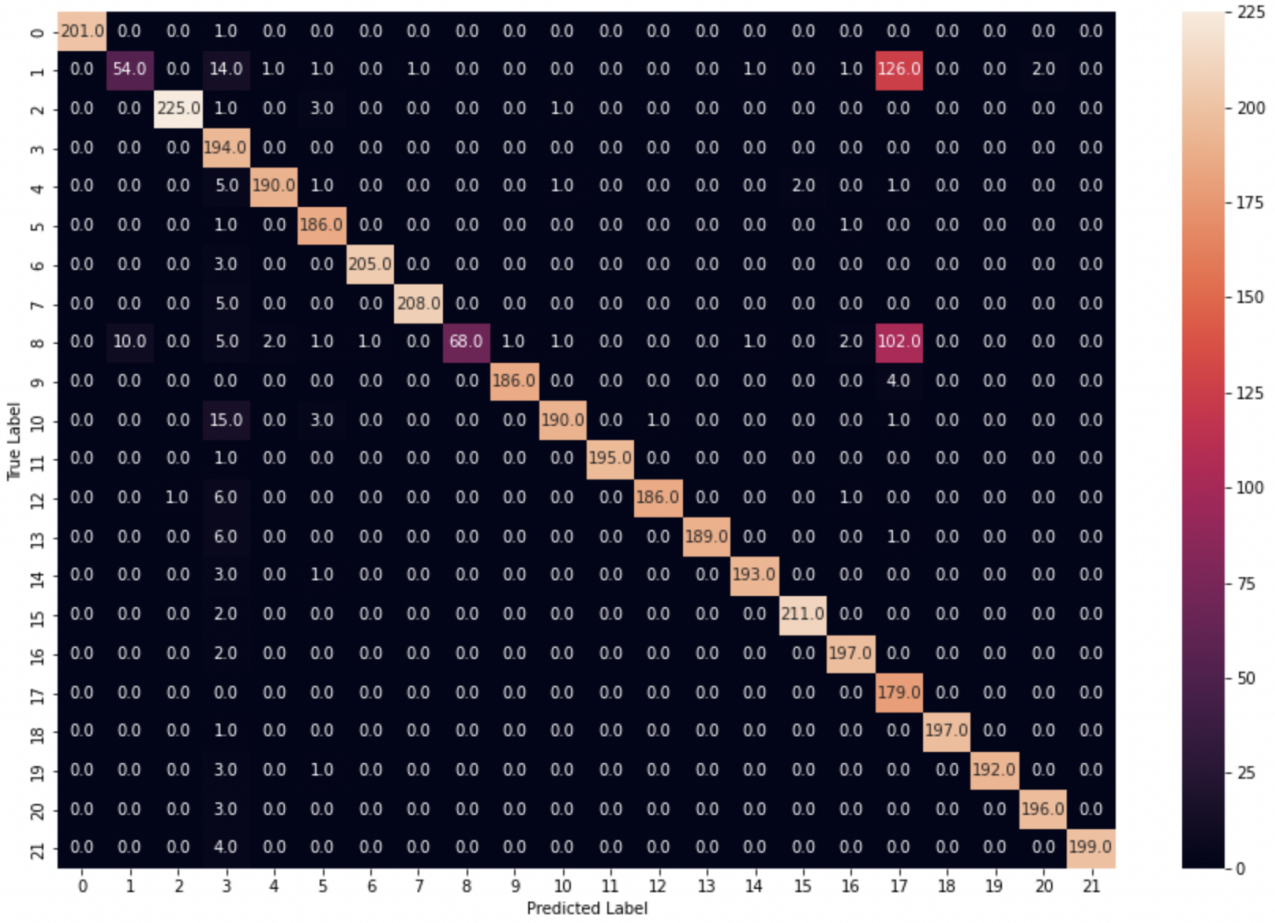
```
y_pred = model.predict(X_test)
```

```
from sklearn.metrics import accuracy_score, confusion_matrix
ac = accuracy_score(y_test, y_pred)
cm = confusion_matrix(y_test, y_pred)

print("Accuracy is {}%:".format(round(ac*100,2)))
```

```
Accuracy is 91.84%:
```

The accuracy of the predicted value is retrieved from the execution phase of the model that is implemented using the Multinomial Naive Bayes Theorem.

```
plt.figure(figsize=(15,10))
sns.heatmap(cm, annot = True, fmt=".1f")
plt.xlabel("Predicted Label")
plt.ylabel("True Label")
plt.show()
```




```
print(le.inverse_transform([1, 17]))
```

```
['Chinese' 'Swedish']
```

```
print(le.inverse_transform([8, 17]))
```

```
['Japanese' 'Swedish']
```

The above graph shows the number of times the true label has been predicted while considering the predicted label and to make our model more efficient, modelling has to be done to refine the output thus reducing errors.

Case 1: The language that has the numerical value 1 is Chinese and the model confuses the Swedish alphabet with the Chinese alphabet
Case 2: The language that has the numerical value 1 is Japanese and the model confuses Swedish alphabets with Japanese alphabets

Predicting:

- The text entered as input is first converted into a bag of words model consisting of vectors
- Assign the predicted value a numerical value according to the classification of languages
- Finding the language corresponding to the predicted value

Predicting

```
def predict(text):
     x = cv.transform([text]).toarray() # converting text to bag of words model (Vector)
     lang = model.predict(x) # predicting the language
     lang = le.inverse_transform(lang) # finding the language corresponding the the predicted value
     return lang[0]
```

```
data = []
with open("inputs.txt") as f:
    for line in f:
        line = line.split('-')
        lang = line[0].strip()
        text = line[1].strip()
        data.append([text, lang, predict(text)])

df = pd.DataFrame(data, columns=['text', 'Actual', 'Predicted'])
df
```

Output :

	text	Actual	Predicted
0	When I use a word, it means just what I choose...	English	English
1	يونكود في النظم القائمة وفيما يخص التطبيقات ال...	Arabic	Arabic
2	Vous êtes le Phénix des hôtes de ces bois	French	French
3	इनके उपयोग से आनलाइन/आफलाइन कहीं भी हिन्दी में...	Hindi	Hindi
4	...فعہ ا۔ تمام انسان آزاد اور حقوق و عزت کے اعتبا	Urdu	Urdu
5	Toda a pessoa tem direito à educação. A educaç...	Portuguese	Portugese
6	... هر کس حق دارد آزادانه در زندگی فرهنگی اجتماعی	Persian	Persian
7	پښتو د هند	Pushto	Pushto
8	史密斯是王明　的　朋友	Chinese	Swedish
9	Jag heter Simon och jag är fjorton år. Jag bor...	Swedish	Swedish
10	質ネ読53代ねえお負消エフ為38発さつ製訃挙つーみな状積定求ッイぶむ犯戻ラえつぱ製都ぱろゆ	Japanese	Swedish

Test case 1:

```
# บรรทัดฐานภาษาไทย เล่ม๑, หนังสืออุเทศภาษาไทย, ชุด บรรทัดฐาน" สถาบันภาษาไทย, กรมวิชาการ, กระทรวงศึกษาธิการ
txt = input("Enter a sentence: ")
print(predict(txt))
```

```
Thai
```

Test case 2:

```
# भारतीय लिपियों का परस्पर परिवर्तन करता है और इन परिवर्तनों को करते समय फॉर्मटिंग को भी बनाये रखता है
txt = input("Enter a sentence: ")
print(predict(txt))
```

```
Hindi
```

Societal Importance: This model when deployed helps in identifying the language accurately and returns an ideal result. Further, this model offers room for improvement with a website showcasing the functioning.