

Experiment :7 (Use Google colab)**Problem statement:**

Implement Naïve Bayes theorem to classify the English text.

Aim: to Implement Naïve Bayes theorem to classify the English text.

ALGORITHM:

Step 1: Start

Step 2: The dataset is divided into two parts, namely, **feature matrix** and the **response/target vector**.

Step 3: The **Feature matrix** (X) contains all the vectors(rows) of the dataset in which each vector consists of the value of **dependent features**. The number of features is **d** i.e. **X = (x1,x2,x2, xd)**.

Step 4: The **Response/target vector** (y) contains the value of **class/group variable** for **each row of feature matrix**.

Step 5: Now the "naïve" conditional independence assumptions are calculated.

Step 6: Stop

PROGRAM:

```
#importing libraries
import prettytable
print("\n *-----* Classification using Naïve bayes *-----* \n')
total_documents = int(input("Enter the Total Number of documents: "))
doc_class = []
i = 0
keywords = []
while not i == total_documents:
    doc_class.append([])
    text = input(f"\nEnter the text of Doc-{i+1} : ").lower()
    clas = input(f"Enter the class of Doc-{i+1} : ")
    doc_class[i].append(text.split())
    doc_class[i].append(clas)
    keywords.extend(text.split())
    i = i+1
keywords = set(keywords)
keywords = list(keywords)
keywords.sort()
to_find = input(
    "\nEnter the Text to classify using Naive Bayes: ").lower().split()

probability_table = []
for i in range(total_documents):
    probability_table.append([])
    for j in keywords:
        probability_table[i].append(0)
```

```

doc_id = 1
for i in range(total_documents):
    for k in range(len(keywords)):
        if keywords[k] in doc_class[i][0]:
            probability_table[i][k] += doc_class[i][0].count(keywords[k])
print("\n")
import prettytable
keywords.insert(0, 'Document ID')
keywords.append("Class")
Prob_Table = prettytable.PrettyTable()
Prob_Table.field_names = keywords
Prob_Table.title = 'Probability of Documents'
x = 0
for i in probability_table:
    i.insert(0, x+1)
    i.append(doc_class[x][1])
    Prob_Table.add_row(i)
    x = x+1
print(Prob_Table)
print("\n")
for i in probability_table:
    i.pop(0)
totalpluswords = 0
totalnegwords = 0
totalplus = 0
totalneg = 0
vocabulary = len(keywords)-2
for i in probability_table:
    if i[len(i)-1] == "+":
        totalplus += 1
        totalpluswords += sum(i[0:len(i)-1])
    else:
        totalneg += 1
        totalnegwords += sum(i[0:len(i)-1])
keywords.pop(0)
keywords.pop(len(keywords)-1)

```

OUTPUT:

Enter the Total Number of documents: 3

Enter the text of Doc-1 : I watched the movie

Enter the class of Doc-1 : +

Enter the text of Doc-2 : I hated the movie

Enter the class of Doc-2 : -

Enter the text of Doc-3 : poor acting

Enter the class of Doc-3 : +

Enter the Text to classify using Naive Bayes: I hated the acting

Probability of Documents									
Document ID	acting	hated	i	movie	poor	the	watched	Class	
1	0	0	1	1	0	1	1	+	
2	0	1	1	1	0	1	0	-	
3	1	0	0	0	1	0	0	+	

Result: The program has been executed successfully and implemented the naïve Bayesian algorithm for English text classification.