CODE:

# Assignment 5: Write a program to implement the naive Bayesian classifier for a sample training data set stored as a  
# .CSV file. Compute the accuracy of the classifier, considering few test data sets.  
  
  
from csv import reader  
  
  
with open('dataset.csv') as csv\_file:  
 data = list(reader(csv\_file))  
rec\_count = len(data)  
print('\nTotal records:'**,** rec\_count)  
  
  
# Fetching Classes:  
cls1 = data[**0**][-**1**]  
for rec in data:  
 if rec[-**1**] != cls1:  
 cls2 = rec[-**1**]  
 break  
else:  
 cls2 = None  
 exit('All the data belong to a single class: ' + cls1)  
  
print(f'Classes: "{cls1}" and "{cls2}"')  
  
  
split\_ratio = **0.9** # b/w 0 and 1 (both not included)  
print('Split ratio (training data:testing data):'**,** split\_ratio)  
split\_index = int(rec\_count\*split\_ratio)  
  
  
# Training:  
cls1\_data**,** cls2\_data = []**,** []  
cls1\_wrd\_cnt**,** cls2\_wrd\_cnt = **0, 0**for rec in data[:split\_index]:  
 cls = rec[-**1**]  
 if cls == cls1:  
 for word in rec:  
 cls1\_data.append(word)  
 cls1\_wrd\_cnt += **1** else: # if cls = cls2  
 for word in rec:  
 cls2\_data.append(word)  
 cls2\_wrd\_cnt += **1**unq\_wrd\_cnt = len(set(cls1\_data+cls2\_data))  
print(f'Words in class "{cls1}": {cls1\_wrd\_cnt}')  
print(f'Words in class "{cls2}": {cls2\_wrd\_cnt}')  
print(f'Total unique words: {unq\_wrd\_cnt} \n')  
  
  
# Testing:  
correct**,** total = **0,** rec\_count-split\_index  
for rec in data[split\_index:]:  
 cls1\_prob**,** cls2\_prob = **1, 1** for word in rec[:-**1**]:  
 # Using Laplace Smoothing:  
 cls1\_prob = (cls1\_data.count(word)+**1**) / (cls1\_wrd\_cnt + unq\_wrd\_cnt)  
 cls2\_prob = (cls2\_data.count(word)+**1**) / (cls2\_wrd\_cnt + unq\_wrd\_cnt)  
 # print('Record:', rec) # debugging  
 # print(f'Probabilities of classes => "{cls1}": {cls1\_prob}, "{cls2}": {cls2\_prob}') # debugging  
 if cls1\_prob > cls2\_prob:  
 pred\_cls = cls1  
 elif cls2\_prob > cls1\_prob:  
 pred\_cls = cls2  
 else: # if cls1\_prob = cls2\_prob  
 # print('Equal probability, skipping \n') # debugging  
 total -= **1** # removing this rec's count  
 continue  
 act\_cls = rec[-**1**]  
 # print(f'Predicted Class: "{pred\_cls}", Actual Class: "{act\_cls}" \n') # debugging  
 if pred\_cls == act\_cls:  
 correct += **1**print('Accuracy in %:'**,** correct\***100**/total)

OUTPUT:

Total records: 768

Classes: "1" and "0"

Split ratio (training data:testing data): 0.9

Words in class "1": 2133

Words in class "0": 4086

Total unique words: 959

Accuracy in %: 66.23376623376623