Exercício 05: Classificador de Bayes aplicado a um problema multivariado

Aluno: Giovanni Martins de Sá Júnior - Matrícula: 2017001850

O classificador de Bayes é um algoritmo de Classificação baseado no Teorema de Bayes, para executar a categorização de dados em classes. Assim, quando ele é aplicado em problemas multivariados, o classificador lida com situações em que os dados de entrada possui diferentes características e atributos.

Com isso, ele pode ser bem eficaz em diferentes cenários, porém, ele se destaca quando a ocasião traz um número limitado de dados de treinamento disponíveis e quando as características são independentes ou aproximadamente independentes condicionalmente. Contudo, é necessário lembrar que a suposição da independência condicional pode não ser válida para todos os casos, e para estas situações.

Diante disso, neste exercício será avaliado a capacidade do classificador de Bayes diante de um problema multivariado. Para a execução do teste, foi escolhido o dataset $Heart$, disponibilizado em locais como Kaggle ou UCI. E com isso, aplicar o conjunto de dados ao classificador, realizando a separação dos dados em conjuntos de teste e treinamento.

Para este exercício, foi proposto inicialmente realizar a separação dos dados em 90% para treinamento e 10 % para teste, e em seguida medir a acurácia para esta separação. Após este primeiro teste, o procedimento será repetido agora para percentuais menores para os conjuntos de treinamento, sendo eles 70% e 20% respectivamente. A seguir, é mostrada toda a implementação feita para o problema.

```
# Importação de Bibliotecas
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.naive_bayes import GaussianNB
from sklearn.metrics import accuracy_score


# Carregamento e separacao dos dados
dados = pd.read_csv('heart.csv', sep=',', header=None)


dados
```

	0	1	2	3	4	5	6	7	8	9	10	11	12
0	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	ta
1	52	1	0	125	212	0	1	168	0	1	2	2	3	
2	53	1	0	140	203	1	0	155	1	3.1	0	0	3	
3	70	1	0	145	174	0	1	125	1	2.6	0	0	3	
4	61	1	0	148	203	0	1	161	0	0	2	1	3	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	
1021	59	1	1	140	221	0	1	164	1	0	2	0	2	
1022	60	1	0	125	258	0	0	141	1	2.8	1	1	3	
1023	47	1	0	110	275	0	0	118	1	1	1	1	2	
1024	50	0	0	110	254	0	0	159	0	0	2	0	2	
1025	54	1	0	120	188	0	1	113	0	1.4	1	1	3	

1026 rows × 14 columns

```
# Fatiamento dos conjuntos de dados X e Y:
x = dados.iloc[1: , : -1]
y = dados.iloc[1: , -1]


x
```

	0	1	2	3	4	5	6	7	8	9	10	11	12
1	52	1	0	125	212	0	1	168	0	1	2	2	3
2	53	1	0	140	203	1	0	155	1	3.1	0	0	3
3	70	1	0	145	174	0	1	125	1	2.6	0	0	3
4	61	1	0	148	203	0	1	161	0	0	2	1	3
5	62	0	0	138	294	1	1	106	0	1.9	1	3	2
...	...	...	...	...	...	...	...	...	...	...	...	...	...
1021	59	1	1	140	221	0	1	164	1	0	2	0	2
1022	60	1	0	125	258	0	0	141	1	2.8	1	1	3

```
y
```

```
1       0
2       0
3       0
4       0
5       0
       ..
1021    1
1022    0
1023    0
1024    1
1025    0
Name: 13, Length: 1025, dtype: object
```

```
tamanhos_amostra_treinamento = [0.9, 0.7, 0.2]


for tamanho_treinamento in tamanhos_amostra_treinamento:
    # Divisão dos daos em treinamento e teste
    x_train, x_test, y_train, y_test = train_test_split(x, y, test_size = 1 - tamanho_treinamento, random_state = 42)

    # Criação e treinamento do classificador Naive Bayes
    clf = GaussianNB()
    clf.fit(x_train, y_train)

    # Realiza previsoes no conjunto de teste
    y_pred = clf.predict(x_test)

    # Calculo da Acuracia
    accuracy = accuracy_score(y_test, y_pred)

    print(f'Tamanho da amostra de Treinamento: {tamanho_treinamento * 100}%')
    print(f'Acuracia: {accuracy * 100}%')
    print('----------------------------------------------------------------\n')
```

```
Tamanho da amostra de Treinamento: 90.0%
Acuracia: 79.6116504854369%
----------------------------------------------------------------

Tamanho da amostra de Treinamento: 70.0%
Acuracia: 81.4935064935065%
----------------------------------------------------------------

Tamanho da amostra de Treinamento: 20.0%
Acuracia: 83.04878048780488%
----------------------------------------------------------------
```

Como foi possível observar o Classificador de Bayes mostrou um resultado ligeiramente melhor de acurácia a media que o conjunto de dados de treinamento era diminuído. Apesar disso, outros fatores podem influenciar na acurácia do modelo, como poderia ser a simplicidade do modelo, a representatividade dos dados, ou até mesmo uma tendência de Overfitting.