Instituto de Ciências Matemáticas e de Computação
Universidade de São Paulo

Ciência de Dados:
Projeto: Classificação

Exercício 1 Considere a base de dados sobre doenças cardíacas:
`https://www.kaggle.com/ronitf/heart-disease-uci`
Faça o pré-processamento dos dados e classifique os pacientes de acordo com a variável "target". Considere os classificadores: Bayesiano paramétrico, Bayesiano não-paramétrico e Naive Bayes.

Exercício 2 No classificar não-paramétrico, verifique o efeito do hiperparâmetro h na classificação dos dados de diabetes, encontrando seu melhor valor:
`https://www.kaggle.com/uciml/pima-indians-diabetes-database`

Exercício 3 Considere o código abaixo para gerar dados artificialmente.

```
from sklearn import datasets
import matplotlib.pyplot as plt
plt.figure(figsize=(6,4))
n_samples = 1000
data  = datasets.make_moons(n_samples=n_samples, noise=.05)
X = data[0]
y = data[1]
plt.scatter(X[:,0], X[:,1], c=y, cmap='viridis', s=50, alpha=0.7)
plt.show(True)
```

Compare os resultados para os métodos Naive Bayes, Classificador Bayesiano paramétrico e o classificador Bayesiano não-paramétrico, variando o ruído (noise).

Exercício 4 Considerando os dados artificiais do exercício anterior, mostre as regiões de separação para os métodos Naive Bayes, k-vizinhos e regressão logística.

Exercício 5 Gere dois conjuntos de pontos em duas dimensões usando o código a seguir:

```
from sklearn.datasets import make_blobs
import numpy as np
import matplotlib.pyplot as plt
n = 500
c = [(1,1), (10,10)] #center of the points
std = [5.0, 2] # standard deviation
nc = [400,50] #number of points in each class
X, y = make_blobs(n_samples=n, n_features=2, cluster_std=std,centers= c)
plt.scatter(X[:,0],X[:,1], c=y)
plt.show(True)
```

Compare os classificadores Naive Bayes e Bayesiano Paramétrico variando a separação entre as nuvens de pontos – mantenha a posição de uma classe fixa e mude a posição do centro da outra classe, calculando a distância entre os centros.

Exercício 6 Gere dois conjuntos de pontos em duas dimensões usando o código a seguir:

```
from sklearn import datasets
import matplotlib.pyplot as plt
plt.figure(figsize=(6,4))
n_samples = 1000
data  = datasets.make_moons(n_samples=n_samples, noise=.05)
X = data[0]
y = data[1]
plt.scatter(X[:,0], X[:,1], c=y, cmap='viridis', s=50, alpha=0.7)
plt.show(True)
```

Compare os classificadores Naive Bayes, k-vizinhos mais próximos e regressão logística variando o nível de ruído (noise) no intervalo [0,1]. No caso do algoritmo k-vizinho, use o método selection.GridSearchCV da biblioteca sciki-learn para determinar a melhor medida de distância e o valor de k.

Exercício 7 Considerando os dados do código anterior, compare os algoritmos: árvores de decisão, forestas aleatórias e bagging. Use o método selection.GridSearchCV da biblioteca sciki-learn para determinar os melhores parâmetros dos modelos.

Exercício 8 Considere as bases: Vehicle, winequality-red e vertebralcolumn-3C. Compare os classificadores: (a) Naive Bayes, (b) Florestas aleatórias, (c) k-vizinhos, (d) regressão logística. Considere as medidas: (i) AUC (área sob a curva ROC), (ii) precisão, (iii) medida F1 e (iv) acurácia. Ou seja, faça uma tabela para cada base, onde as linhas representam os classificadores e as colunas, as medidas de avaliação.

Exercício 9 Considere os código abaixo. Avalie como o desbalanceamento influencia nos resultados usando as medidas i) AUC (área sob a curva ROC), (ii) precisão, (iii) medida F1 e (iv) acurácia. Elabore um estudo e proceda com o desenvolvimento dos códigos. Dica: use o método de validação cruzada estratificado na classificação e discuta a comparação com o caso sem o uso de estratificação.

```
# Generate and plot a synthetic imbalanced classification dataset
from collections import Counter
from sklearn.datasets import make_classification
from matplotlib import pyplot
from numpy import where
# define dataset
X, y = make_classification(n_samples=10000, n_features=2, n_redundant=0,
n_clusters_per_class=1, weights=[0.99], flip_y=0, random_state=1)
# summarize class distribution
counter = Counter(y)
print(counter)
# scatter plot of examples by class label
for label, _ in counter.items():
row_ix = where(y == label)[0]
pyplot.scatter(X[row_ix, 0], X[row_ix, 1], label=str(label))
pyplot.legend()
pyplot.show()
```

Exercício 10 Realize a classificação da base Titanic. Use os métodos de seleção de modelos e determine o melhor modelo e seus hiperparâmetros. Use também seleção de atributos e tente melhorar os resultados.