

*Métodos Matemáticos em Engenharia Química*

*Programa de Pós-Graduação em Engenharia Química - PPGEQ*

*Trabalho 02: kNN*

*Prof. Éliton Fontana*

Utilize o seguinte comando para gerar um conjunto contendo 100 elementos de dados com 3 atributos e 2 classes:

```
from sklearn.datasets import make_gaussian_quantiles
```

```
X, y = make_gaussian_quantiles(n_features=3, n_classes=2, n_samples = 100, random_state = a)
```

Para o parâmetro  $a$ , utilize um valor aleatório com 4 ou 5 dígitos. O conjunto gerado será utilizado como conjunto de treinamento para um algoritmo de classificação usando kNN.

a) Considere que seja necessário classificar um novo ponto, por exemplo,  $z = (0, 1, -1)$ . Crie um vetor que armazene a distância deste ponto com relação a cada um dos 100 elementos do conjunto de treinamento;

b) Crie uma lista contendo a distância de cada elemento e a classe a qual este elemento pertence. Na sequência, ordene esta lista em ordem crescente de distância <sup>1</sup>;

c) Utilizando uma classificação 1-NN, atribua ao ponto avaliado a mesma classe que o primeiro elemento da lista obtida no item b (já em ordem crescente);

d) Para realizar uma classificação k-NN, defina primeiramente um valor para  $k$ . Faça uma média dos  $k$  primeiros valores de classe presentes na lista obtida no item b. Como as classes são atribuídas por padrão como 0 ou 1, caso a média for menor que 0.5 o ponto pertence à classe 0, caso contrário à classe 1. Obtenha a classificação do ponto utilizando  $k = 3$  e  $k = 5$ .

**Obs:** Para entrega do trabalho, é suficiente entregar o arquivo .py com a classificação para  $k = 5$ .

---

<sup>1</sup>Isto pode ser feito, por exemplo, utilizando a função `sorted()`. Para uma lista `list`, o comando `list2 = sorted(list)` irá criar uma nova lista `list2` com os elementos da lista `list` em ordem crescente, baseado no primeiro elemento desta lista. Observe que isto é diferente do comando `list.sort`, que irá alterar todos os elementos da lista.

**Obs:** Caso vocês queiram visualizar o conjunto de pontos em um espaço 3D, pode-se utilizar os seguintes comandos:

```
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D

fig = plt.figure()
ax = fig.add_subplot(111, projection='3d')
# #Pontos azuis se y == 0
ax.scatter(X[y == 0 ,0], X[y == 0 ,1], X[y == 0 ,2], c='b', marker='o')
# #Pontos vermelhos se y == 1
ax.scatter(X[y == 1 ,0], X[y == 1 ,1], X[y == 1 ,2], c='r', marker='o')
ax.set_xlabel('X1')
ax.set_ylabel('X2')
ax.set_zlabel('X3')
```