REGRESSÃO LINEAR GHFLUSÃO

from sklearn.datasets import load\_boston # para carregar os dados

from sklearn.model\_selection import train\_test\_split

from sklearn.linear\_model import LinearRegression # importa o modelo

# carrega os dados

house\_data = load\_boston()

X = house\_data['data']

y = house\_data['target']

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.3, random\_state=42)

regr = LinearRegression() # cria o modelo

regr.fit(X\_train, y\_train) # treina o modelo

r2\_train = regr.score(X\_train, y\_train)

r2\_test = regr.score(X\_test, y\_test)

print('R2 no set de treino: %.2f' % r2\_train)

print('R2 no set de teste: %.2f' % r2\_test)

KNN GHFLUSÃO

import numpy as np

import matplotlib.pyplot as plt

import pandas as pd

from sklearn.datasets import load\_iris

from sklearn.model\_selection import train\_test\_split

from sklearn.neighbors import KNeighborsClassifier

iris\_dataset = load\_iris()

X\_train, X\_test, y\_train, y\_test = train\_test\_split(iris\_dataset['data'], iris\_dataset['target'], random\_state=0)

knn = KNeighborsClassifier(n\_neighbors=1)

knn.fit(X\_train, y\_train)

y\_pred = knn.predict(X\_test)

print("Test set predictions:\n {}".format(y\_pred))

print("Test set score: {:.2f}".format(np.mean(y\_pred == y\_test)))