MSSV: 21014091 Họ và Tên Đặng Bảo Thông ca TH: t7\_t1012

1. Creating features

import pandas as pd

sales\_df = pd.read\_csv(".//advertising\_and\_sales\_clean.csv")

sales\_df.head()

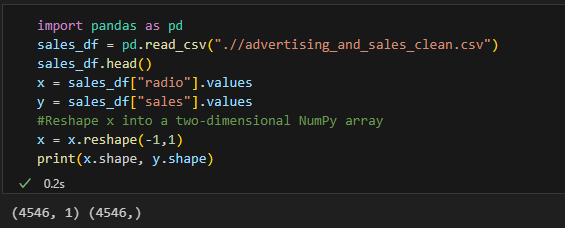
x = sales\_df["radio"].values

y = sales\_df["sales"].values

#Reshape x into a two-dimensional NumPy array

x = x.reshape(-1,1)

print(x.shape, y.shape)



1. Building a linear regression model

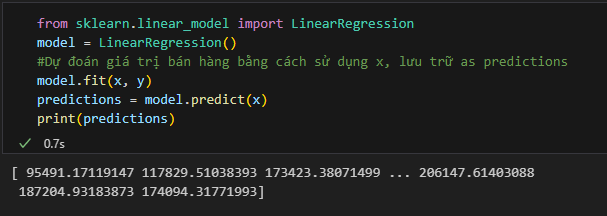
from sklearn.linear\_model import LinearRegression

model = LinearRegression()

model.fit(x, y)

predictions = model.predict(x)

print(predictions)



1. Visualizing a linear regression model

import matplotlib.pyplot as plt

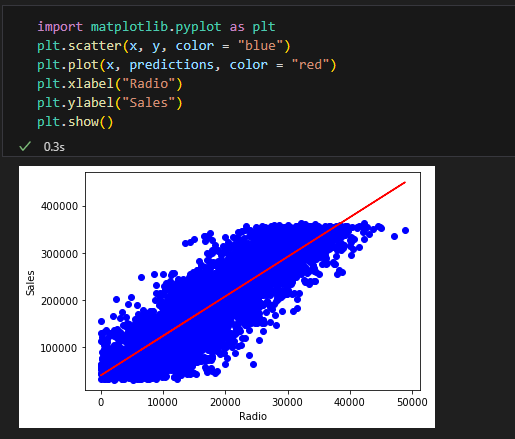
plt.scatter(x, y, color = "blue")

plt.plot(x, predictions, color = "red")

plt.xlabel("Radio")

plt.ylabel("Sales")

plt.show()



1. Fit and predict for regression

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

from sklearn.linear\_model import LinearRegression

import numpy as np

#Tạo x, một mảng chứa các giá trị của tất cả features trong sales\_df và y, chứa tất cả các giá trị từ cột "sales".

x = pd.get\_dummies(sales\_df.drop(columns=['sales'])).values

y = sales\_df['sales'].values

#Chia dữ liệu thành tập huấn luyện và tập kiểm tra

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

#Khởi tạo mô hình hồi quy tuyến tính.

model = LinearRegression()

model.fit(x\_train, y\_train)

#Dự đoán

y\_pred = model.predict(x\_test)

print(y\_pred)

Ảnh có chứa văn bản, ảnh chụp màn hình, phần mềm, Phần mềm đa phương tiện

Mô tả được tạo tự động

1. Regression performance

from sklearn.metrics import mean\_squared\_error

import numpy as np

#Tính R-squared score

r\_squared = model.score(x\_test, y\_test)

#Tính RMSE

rmse = np.sqrt(mean\_squared\_error(y\_test, y\_pred))

print('R-squared:', r\_squared)

print('Root Mean Square Error:', rmse)

Ảnh có chứa văn bản, ảnh chụp màn hình, phần mềm, màn hình

Mô tả được tạo tự động

1. Cross-validation for R-squared

from sklearn.model\_selection import KFold, cross\_val\_score

from sklearn.linear\_model import LinearRegression

model = LinearRegression()

kf = KFold(n\_splits=6, shuffle=True, random\_state=5)

cv\_scores = cross\_val\_score(model, x, y, cv=kf)

print(cv\_scores)

Ảnh có chứa văn bản, ảnh chụp màn hình, màn hình, phần mềm

Mô tả được tạo tự động

1. Analyzing cross-validation metrics

mean\_cv\_scores = np.mean(cv\_scores)

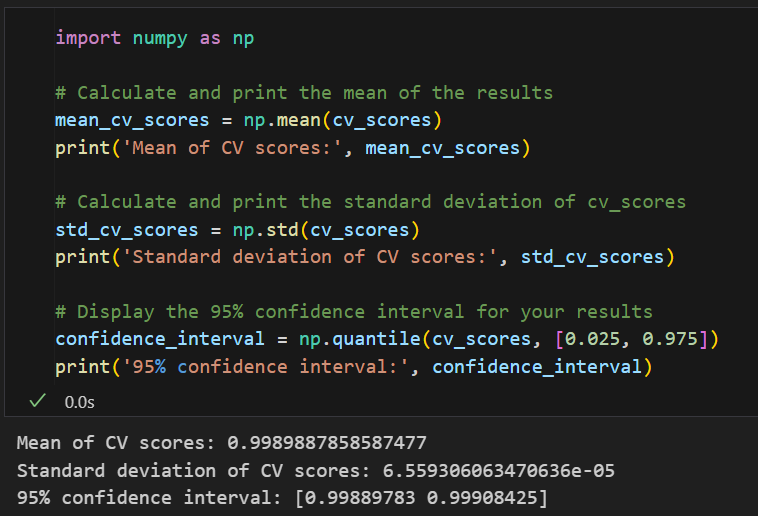
print('mean of cv\_scores: ', mean\_cv\_scores)

std\_cv\_scores = np.std(cv\_scores)

print('standard deviation of cv\_scores:',std\_cv\_scores)

confidene\_interval = np.quantile(cv\_scores, [0.025, 0.975])

print('95% confidence interval:', confidene\_interval)



1. Regularized regression: Ridge

from sklearn.linear\_model import Ridge

from sklearn.metrics import r2\_score

#Define alphas

alphas = [0.1, 1.0, 10.0, 100.0, 1000.0, 10000.0]

#Interate over alphas

for alpha in alphas:

#Khởi tạo và điều chỉnh mô hình hồi quy Ridge

model = Ridge(alpha=alpha)

model.fit(x\_train, y\_train)

#Dự đoán doanh số bán hàng bằng test features

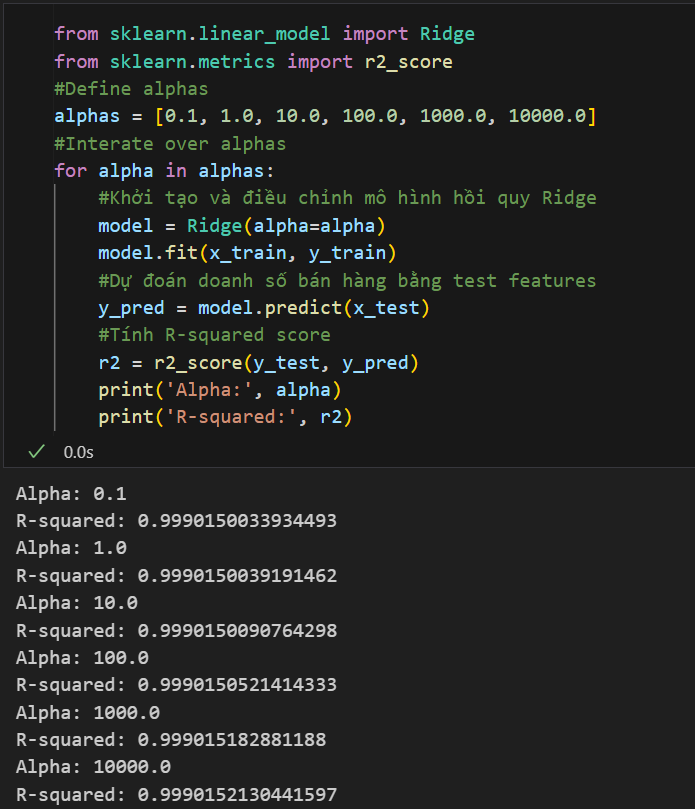
y\_pred = model.predict(x\_test)

#Tính R-squared score

r2 = r2\_score(y\_test, y\_pred)

print('Alpha:', alpha)

print('R-squared:', r2)



1. Lasso regression for feature importance

from sklearn.linear\_model import Lasso

#Khởi tạo một bộ hồi quy Lasso với hệ số alpha là 0,3.

model = Lasso(alpha=0.3)

#Điều chỉnh mô hình cho phù hợp với dữ liệu.

model.fit(x\_train, y\_train)

#Tính toán các hệ số của mô hình, lưu dưới dạng lasso\_coef.

lasso\_coef = model.coef\_

print(lasso\_coef)

Ảnh có chứa văn bản, ảnh chụp màn hình, phần mềm, màn hình

Mô tả được tạo tự động