import numpy as np

import matplotlib.pyplot as plt

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

from sklearn.linear\_model import LinearRegression

from sklearn.metrics import mean\_squared\_error

# Згенерувати випадковий набір даних та нормалізувати значення

n = 1000  # Кількість значень

X = np.random.rand(n)  # Основна змінна

noise = 0.1 \* np.random.randn(n)  # Випадковий шум

Y = 2 \* X + 1 + noise

X = (X - X.mean()) / X.std()

Y = (Y - Y.mean()) / Y.std()

# Розділити дані на навчальну та тестову вибірки

X\_train, X\_test, Y\_train, Y\_test = train\_test\_split(X, Y, test\_size=0.2, random\_state=42)

# Навчити К-регресор з різними значеннями К та знайти MSE для кожного K

K\_values = [1, 2, 3, 4, 5]

mse\_values = []

for K in K\_values:

    model = LinearRegression()

    model.fit(X\_train.reshape(-1, 1), Y\_train)

    Y\_pred = model.predict(X\_test.reshape(-1, 1))

    mse = mean\_squared\_error(Y\_test, Y\_pred)

    mse\_values.append(mse)

    print(f"К = {K}, MSE = {mse}")

# Вибрати найкраще значення K

best\_K = K\_values[np.argmin(mse\_values)]

# Візуалізація результатів

plt.figure(figsize=(10, 6))

plt.scatter(X\_test, Y\_test, color='blue', label='Справжні значення')

model = LinearRegression()

model.fit(X\_train.reshape(-1, 1), Y\_train)

Y\_pred = model.predict(X\_test.reshape(-1, 1))

plt.plot(X\_test, Y\_pred, color='red', label=f'Прогноз (K = {best\_K})')

plt.legend()

plt.xlabel('X')

plt.ylabel('Y')

plt.title('Регресія з оптимальним значенням K')

plt.show()

RESULT

Зображення, що містить текст, Шрифт, знімок екрана, білий

Автоматично згенерований опис

Зображення, що містить текст, знімок екрана, Графік, ряд

Автоматично згенерований опис