import pandas as pd

import numpy as np

from sklearn.datasets import load\_iris

from sklearn.neighbors import KNeighborsClassifier

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

from sklearn.metrics import accuracy\_score

from sklearn.datasets import make\_blobs

import matplotlib.pyplot as plt

from matplotlib.colors import ListedColormap

# Завантаження датасету Iris

iris = load\_iris()

X, y, labels, feature\_names = iris.data, iris.target, iris.target\_names, iris['feature\_names']

# Створення DataFrame для датасету Iris

df\_iris = pd.DataFrame(X, columns=feature\_names)

df\_iris['label'] = y

features\_dict = {k: v for k, v in enumerate(labels)}

df\_iris['label\_names'] = df\_iris.label.apply(lambda x: features\_dict[x])

# Перемішування записів у датасеті Iris

df\_iris = df\_iris.sample(frac=1, random\_state=2021).reset\_index(drop=True)

# Нормалізація параметрів датасету Iris

from sklearn.preprocessing import MinMaxScaler

scaler = MinMaxScaler()

normalized\_data = scaler.fit\_transform(df\_iris.iloc[:, :-3])

# Розділення датасету Iris на навчальну та тестову вибірки

X\_train, X\_test, y\_train, y\_test = train\_test\_split(normalized\_data, df\_iris['label'], test\_size=0.2, random\_state=42)

# Виведення розмірів навчальної та тестової вибірок

print("Розмір навчальної вибірки:", X\_train.shape)

print("Розмір тестової вибірки:", X\_test.shape)

# Створення KNN-класифікатора для датасету Iris

k\_values = range(1, 21)

best\_k\_iris = None

score\_best\_iris = 0

for k in k\_values:

    knn = KNeighborsClassifier(n\_neighbors=k)

    knn.fit(X\_train, y\_train)

    y\_pred = knn.predict(X\_test)

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

    if accuracy > score\_best\_iris:

        score\_best\_iris = accuracy

        best\_k\_iris = k

print("Найкраще значення K для датасету Iris: K = {}, Точність = {:.2f}".format(best\_k\_iris, score\_best\_iris))

# Створення штучного датасету для бінарної класифікації

np.random.seed = 2021

X\_D2, y\_D2 = make\_blobs(n\_samples=300, n\_features=2, centers=8, cluster\_std=1.3, random\_state=4)

y\_D2 = y\_D2 % 2

# Виведення графіку для штучного датасету

plt.figure()

plt.title('Sample binary classification problem with non-linearly separable classes')

plt.scatter(X\_D2[:, 0], X\_D2[:, 1], c=y\_D2, marker='o', s=30, cmap=cmap\_bold)

# Розділення штучного датасету на навчальну та тестову вибірки

X\_train\_D2, X\_test\_D2, y\_train\_D2, y\_test\_D2 = train\_test\_split(X\_D2, y\_D2, test\_size=0.2, random\_state=42)

# Створення KNN-класифікатора для штучного датасету

best\_k\_D2 = None

score\_best\_D2 = 0

for k in k\_values:

    knn = KNeighborsClassifier(n\_neighbors=k)

    knn.fit(X\_train\_D2, y\_train\_D2)

    y\_pred\_D2 = knn.predict(X\_test\_D2)

    accuracy\_D2 = accuracy\_score(y\_test\_D2, y\_pred\_D2)

    if accuracy\_D2 > score\_best\_D2:

        score\_best\_D2 = accuracy\_D2

        best\_k\_D2 = k

print("Найкраще значення K для штучного датасету: K = {}, Точність = {:.2f}".format(best\_k\_D2, score\_best\_D2))

Зображення, що містить текст, програмне забезпечення, Комп’ютерна піктограма, Веб-сторінка

Автоматично згенерований опис

import seaborn as sns

import matplotlib.pyplot as plt

# Графік розподілу параметрів квітів Iris

plt.figure(figsize=(12, 8))

sns.pairplot(df\_iris, hue='label\_names', palette='viridis', markers=["o", "s", "D"])

plt.suptitle('Pairplot for Iris Dataset', y=1.02)

plt.show()

# Графік розподілу класів у штучному датасеті

plt.figure(figsize=(8, 6))

plt.title('Distribution of Classes in Artificial Dataset')

plt.scatter(X\_D2[:, 0], X\_D2[:, 1], c=y\_D2, marker='o', s=30, cmap=cmap\_bold)

plt.xlabel('Feature 1')

plt.ylabel('Feature 2')

plt.show()