Модульна робота 1

Студента КНм-11

Ділай Р.І.

Завдання 1.

Імпортуєм бібліотеки

import numpy as np

import pandas as pd

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

from sklearn.preprocessing import MinMaxScaler

from sklearn.neighbors import KNeighborsClassifier

from sklearn.metrics import accuracy\_score

import matplotlib.pyplot as plt

1. Створення KNN – класифікатора у Python.

* 1. Завантажуєм базу параметрів fruit dataset.

fruit\_data = pd.read\_csv("fruit\_data\_with\_colors.txt", sep='\t')

X\_classification = fruit\_data[['mass', 'width', 'height', 'color\_score']]

y\_classification = fruit\_data['fruit\_label']

* Перемішуєм записи у базі.

X\_classification, y\_classification = np.array(X\_classification), np.array(y\_classification)

permutation = np.random.permutation(len(X\_classification))

X\_classification = X\_classification[permutation]

y\_classification = y\_classification[permutation]

* Нормалізуєм параметри

scaler\_classification = MinMaxScaler()

X\_classification = scaler\_classification.fit\_transform(X\_classification)

* Розділяєм на навчальну та тестову вибірки

X\_train\_class, X\_test\_class, y\_train\_class, y\_test\_class = train\_test\_split(

X\_classification, y\_classification, test\_size=0.2, random\_state=42

)

# Test K from 1 to 20

k\_values = range(1, 21)

accuracy\_scores = []

for k in k\_values:

knn\_classifier = KNeighborsClassifier(n\_neighbors=k)

knn\_classifier.fit(X\_train\_class, y\_train\_class)

y\_pred\_class = knn\_classifier.predict(X\_test\_class)

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

accuracy\_scores.append(accuracy)

* Вибрати величину K для найкращих показників якості класифікацій у тестовій вибірці

best\_k\_class = k\_values[np.argmax(accuracy\_scores)]

best\_accuracy = np.max(accuracy\_scores)

print("KNN Класифікатор:")

print(f"Найкраще K: {best\_k\_class}, Найвища точність: {best\_accuracy:.4f}")

plt.figure(figsize=(10, 5))

plt.plot(k\_values, accuracy\_scores, marker='o')

plt.title('Точність класифікації KNN для різних K')

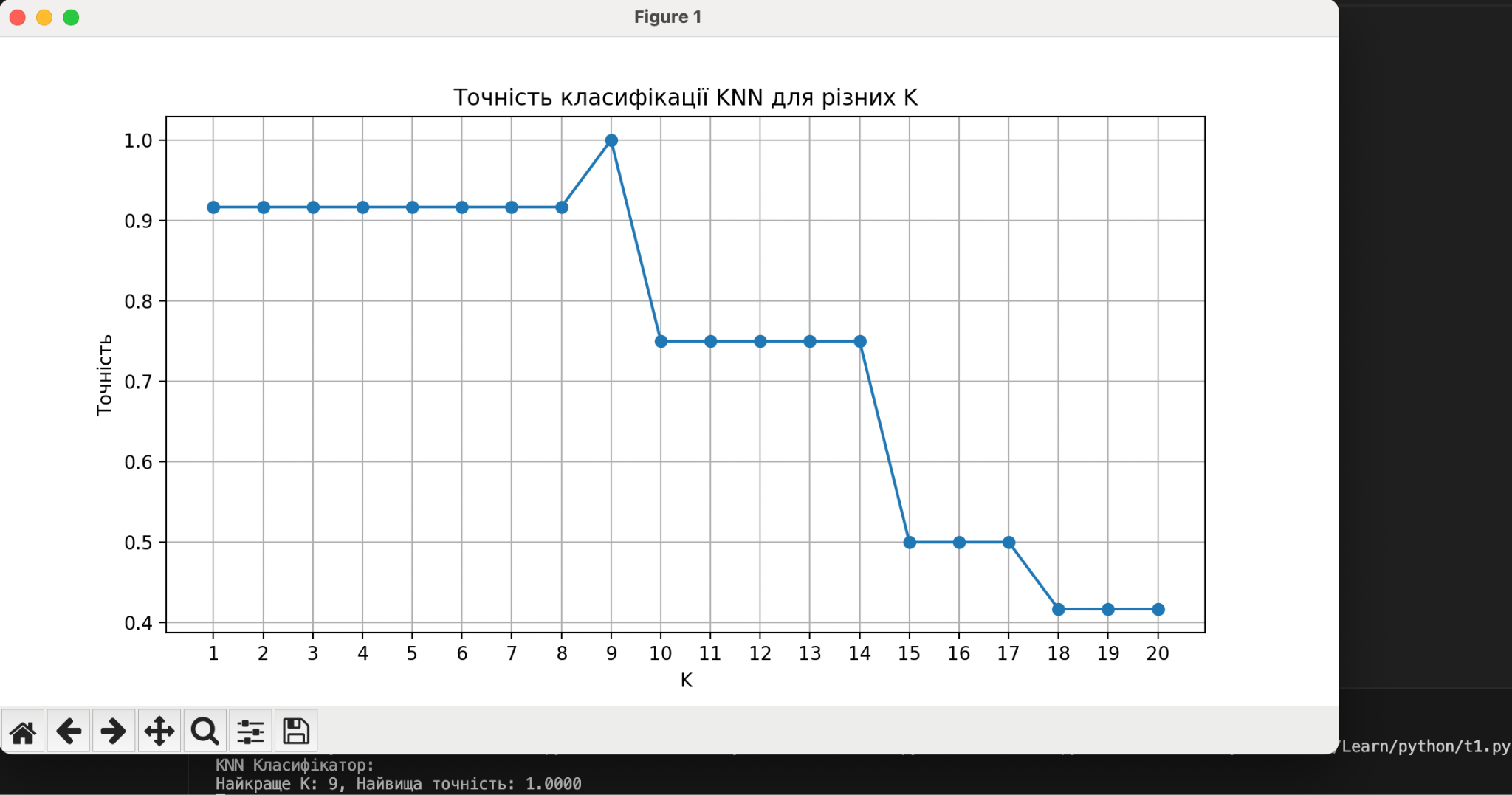
plt.xlabel('K')

plt.ylabel('Точність')

plt.xticks(k\_values)

plt.grid(True)

plt.show()



Завдання 2.

Створення KNN – регресора у Python.

import numpy as np

import pandas as pd

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

from sklearn.preprocessing import MinMaxScaler

from sklearn.neighbors import KNeighborsRegressor

from sklearn.metrics import mean\_squared\_error, r2\_score

import matplotlib.pyplot as plt

np.random.seed(42)

X\_regression = np.random.rand(1000, 1) \* 10

y\_regression = 2 \* X\_regression.squeeze() + np.random.randn(1000) \* 2

scaler\_regression = MinMaxScaler()

X\_regression = scaler\_regression.fit\_transform(X\_regression)

X\_train\_reg, X\_test\_reg, y\_train\_reg, y\_test\_reg = train\_test\_split(

X\_regression, y\_regression, test\_size=0.2, random\_state=42

)

k\_values\_reg = range(1, 101, 5)

mse\_scores = []

r2\_scores = []

for k in k\_values\_reg:

knn\_regressor = KNeighborsRegressor(n\_neighbors=k)

knn\_regressor.fit(X\_train\_reg, y\_train\_reg)

y\_pred\_reg = knn\_regressor.predict(X\_test\_reg)

mse = mean\_squared\_error(y\_test\_reg, y\_pred\_reg)

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

mse\_scores.append(mse)

r2\_scores.append(r2)

best\_k\_reg\_mse = k\_values\_reg[np.argmin(mse\_scores)]

best\_mse = np.min(mse\_scores)

best\_k\_reg\_r2 = k\_values\_reg[np.argmax(r2\_scores)]

best\_r2 = np.max(r2\_scores)

print("\nKNN Регресор:")

print(f"Найкраще K (MSE): {best\_k\_reg\_mse}, Найменша MSE: {best\_mse:.4f}")

print(f"Найкраще K (R^2): {best\_k\_reg\_r2}, Найвищий R^2: {best\_r2:.4f}")

plt.figure(figsize=(10, 5))

plt.plot(k\_values\_reg, mse\_scores, marker='o')

plt.title('MSE регресії KNN для різних K')

plt.xlabel('K')

plt.ylabel('MSE')

plt.xticks(k\_values\_reg)

plt.grid(True)

plt.show()

plt.figure(figsize=(10, 5))

plt.scatter(X\_test\_reg, y\_test\_reg, label='Фактичні значення', alpha=0.6)

plt.scatter(X\_test\_reg, knn\_regressor.predict(X\_test\_reg), label='Прогнозовані значення', alpha=0.6)

plt.title(f'Порівняння фактичних та прогнозованих значень (K={best\_k\_reg\_mse})')

plt.xlabel('X')

plt.ylabel('Y')

plt.legend()

plt.show()

