

МІНІСТЕРСТВО ОСВІТИ ТА НАУКИ УКРАЇНИ
Національний технічний університет
«Харківський політехнічний інститут»
Кафедра ГМКГ

Лабораторна работа №1

З дисципліни «Інтелектуальний аналіз даних»

Виконав:

Студент групи ІКМ-220 г.

Ульянов Кирило Юрійович

Перевірив:

Доц. Дашкевич А.О.

Харків 2023

Мета роботи: вивчення класифікації методом k-найближчих сусідів

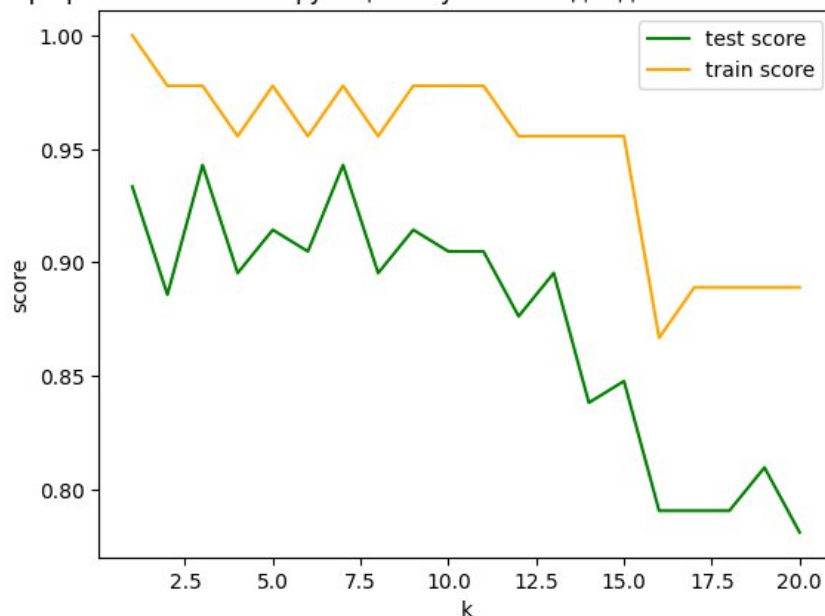
Завдання на роботу: завантаження набору даних «Іриси Фішера»,
формування навчальної та тестової вибірок, визначення функціоналу якості
класифікатору за різних значень k, візуалізація результатів.

Таблиця результатів:

Встановив параметр розподілення датасету train_size = 0.3

k	Точність train score	Точність test score
1	1.0	0.9333333333333333
2	0.9777777777777777	0.8857142857142857
3	0.9777777777777777	0.9428571428571428
4	0.9555555555555556	0.8952380952380953
5	0.9777777777777777	0.9142857142857143
6	0.9555555555555556	0.9047619047619048
7	0.9777777777777777	0.9428571428571428
8	0.9555555555555556	0.8952380952380953
9	0.9777777777777777	0.9142857142857143
10	0.9777777777777777	0.9047619047619048
11	0.9777777777777777	0.9047619047619048
12	0.9555555555555556	0.8761904761904762
13	0.9555555555555556	0.8952380952380953
14	0.9555555555555556	0.8380952380952381
15	0.9555555555555556	0.8476190476190476
16	0.8666666666666667	0.7904761904761904
17	0.8888888888888888	0.7904761904761904
18	0.8888888888888888	0.7904761904761904
19	0.8888888888888888	0.8095238095238095
20	0.8888888888888888	0.780952380952381

Об'єднаний графік залежностей функціоналу якості від k для навчальної та тестової вибірки



Код програми:

```
import matplotlib.pyplot as plt
import seaborn as sns
from matplotlib.colors import ListedColormap
from sklearn.model_selection import train_test_split
from sklearn import datasets, neighbors

# Набір даних «Іриси фішера»
iris = datasets.load_iris()

# Розробка датасету
X = iris.data # зберігає вимірні признаки ірісів
Y = iris.target # зберігає класи ірісів

x_train, x_test, y_train, y_test = train_test_split(X, Y,
train_size=0.3,random_state=0) # тут за допомогою функції
train_test_split розбиваємо датасет на тренінг та тест вибірки
випадково у розмірі 30% - вивчення, 70% - тестування.

# Навчання класифікатора
```

```

k = 1

k_values = []

test_score = [] # зберігання оцінок тестової виборки
train_score = [] # зберігання оцінок навчальної виборки

while k <= 20:

    k_values.append(k)

    knn = neighbors.KNeighborsClassifier(k)

    knn.fit(x_train, y_train)

    k += 1

    # функціонал якості на навчальній та тестовій вибірці для
всіх значень k

    test_score_1 = knn.score(x_test, y_test)
    train_score_1 = knn.score(x_train, y_train)
    test_score.append(test_score_1)
    train_score.append(train_score_1)


plt.plot(k_values, test_score, color="green", label="test
score")

plt.plot(k_values, train_score, color="orange", label="train
score")

plt.xlabel('k')

plt.title('Об'єднаний графік залежностей функціоналу якості від
k для навчальної та тестової вибірки')

plt.ylabel('score')

plt.legend()

plt.show()


print(test_score) # функціонал якості тестової вибірки
print(train_score) # функціонал якості навчальної вибірки

```