

МИНЕСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное бюджетное образовательное учреждение высшего образования
«УЛЬЯНОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»
Кафедра «Измерительно-вычислительные комплексы»

«Технология обработки информации»

Отчет по лабораторной работе №4
Метод k – ближайших соседей (k -NN)

Выполнил:

Студент группы ИСТбд-41

Калашников М. А.

Проверил:

Шишкин В.В.

Ульяновск

2022

Лабораторная работа № 4. Метод k – ближайших соседей (k-NN)

1. Создать симулированный набор данных и записать его на диск в виде csv файла с параметрами.

```
# Задание классификаций для продуктов
Params = ['Продукт', 'Сладость', 'Хруст', 'Класс']
# Создание списка с продуктами и их параметрами
Eda1 = [['Яблоко', '7', '7', '0'],
        ['Салат', '2', '5', '1'],
        ['Бекон', '1', '2', '2'],
        ['Банан', '9', '1', '0'],
        ['Орехи', '1', '5', '2'],
        ['Рыба', '1', '1', '2'],
        ['Сыр', '1', '1', '2'],
        ['Виноград', '8', '1', '0'],
        ['Морковь', '2', '8', '1'],
        ['Апельсин', '6', '1', '0'],
        ['Мандарин', '9', '1', '0'],
        ['Арбуз', '3', '7', '1'],
        ['Тыква', '3', '1', '2'],
        ['Репка', '5', '3', '0'],
        ['Подсолнух', '1', '7', '1']]
#Создание разметки
```

```
# запись данных в файл
with open('File_s_datas.csv', 'w', encoding='utf8') as file:
    writer = csv.writer(file, lineterminator="\n")
    writer.writerow(Params)
    for row in Eda1:
        writer.writerow(row)
```

2. Запрограммировать метрический классификатор по методу k-NN. Для проверки решить ту же задачу методом k-NN библиотеки sklearn.

```
Eda2 = [['Яблоко', '7', '7', '0'],
        ['Салат', '2', '5', '1'],
        ['Арбуз', '10', '8', '3'],
        ['Яйца', '1', '2', '2'],
        ['Банан', '9', '1', '0'],
        ['Фундук', '10', '9', '3'],
        ['Орехи', '1', '5', '2'],
        ['Рыба', '1', '1', '2'],
        ['Сыр', '1', '1', '2'],
        ['Виноград', '8', '1', '0'],
        ['Шипучка', '8', '10', '3'],
        ['Морковь', '2', '8', '1'],
        ['Апельсин', '6', '1', '0'],
        ['конфета', '6', '9', '3'],
        ['Малина', '9', '1', '0'],
        ['Капуста', '3', '7', '1'],
        ['Халва', '6', '7', '3'],
        ['Говядина', '3', '1', '2'],
        ['Свиника', '5', '3', '0'],
        ['Свекла', '1', '7', '1'],
        ['Мармелад', '7', '8', '3']]
```

Реализация классификатора по методу k-NN:

```
# Реализация классификатора по методу k-NN
def knnKLASS(products, OriginalSIZE, pars_window, Resultats):
    # Индексация переменных
    Materials = np.array(products)
    Size1 = len(Materials) - OriginalSIZE
    Distances = np.zeros((Size1, OriginalSIZE))
    All_class = [0] * Size1
    # Реализация классификатора №1
    for i in range(Size1):
        for j in range(OriginalSIZE):
            distans = Razmetka(int(Materials[OriginalSIZE + i][1]),
                                int(Materials[OriginalSIZE + i][2]), int(Materials[j + 1][1]),
                                int(Materials[j + 1][2]))
            Distances[i][j] = distans if distans < pars_window else 1000
    # Реализация классификатора №2
    for i in range(Size1):
        print(str(i) + ') ' + Materials[OriginalSIZE + i][0])
        Massa = [0] * products.iloc[:, 'Класс'].nunique()
        neighbor = np.sum(Distances[i] != 1000)
        for j in range(neighbor + 1):
            min = Distances[i].argmin()
            Massa[int(Materials[min + 1][3])] += ((neighbor - j + 1) / neighbor)
            Distances[i][min] = 1000
        All_class[i] = np.array(Massa).argmax()
    # Проверка классов (Предположительный и Оригинальный)
    # 'Должны все совпадать!'
    # Сообщает о правильности проведения эксперимента!
    print('Предположительный класс: ', All_class[i], 'Оригинальный класс: ', Materials[OriginalSIZE + i][3])
    if int(All_class[i]) != int(Materials[OriginalSIZE + i][3]):
        print('нет совпадения')
    else:
        print('совпадение')
        Resultats += 1
    print(All_class)
    print('Совпадений ручного классификатора: ', str(Resultats))
    return All_class
# Реализация классификатора методом k-NN библиотеки sklearn
```

Реализация классификатора методом k-NN библиотеки sklearn:

```
Реализация классификатора методом k-NN библиотеки sklearn
def sklearnKNN(values, k, y):
    x_train, x_test, y_train, y_test = train_test_split(values, y, test_size=0.3, shuffle=False, stratify=None)
    scaler = StandardScaler()
    scaler.fit(x_train)
    x_train = scaler.transform(x_train)
    x_test = scaler.transform(x_test)
    model = KNeighborsClassifier(n_neighbors=k)
    model.fit(x_train, y_train)
    predictions = model.predict(x_test)
    return x_train, x_test, y_train, y_test, predictions
```

3. Результат работы ручного классификатора с тестовой выборкой:

```
D:\lab-4\venv\Scripts\python.exe D:\lab-4\main.py
0) Мандарин
Предположительный класс: 0 Оригинальный класс: 0
совпадение
1) Арбуз
Предположительный класс: 1 Оригинальный класс: 1
совпадение
2) Тыква
Предположительный класс: 2 Оригинальный класс: 2
совпадение
3) Репка
Предположительный класс: 0 Оригинальный класс: 0
совпадение
4) Подсолнух
Предположительный класс: 1 Оригинальный класс: 1
совпадение
[0, 1, 2, 0, 1]
Совпадений ручного классификатора: 5
```

Результат работы метода k-NN библиотеки sklearn подтверждает правильность проведенного эксперимента при использовании ручного классификатора по методу k-NN. Все классы продуктов были определены верно.

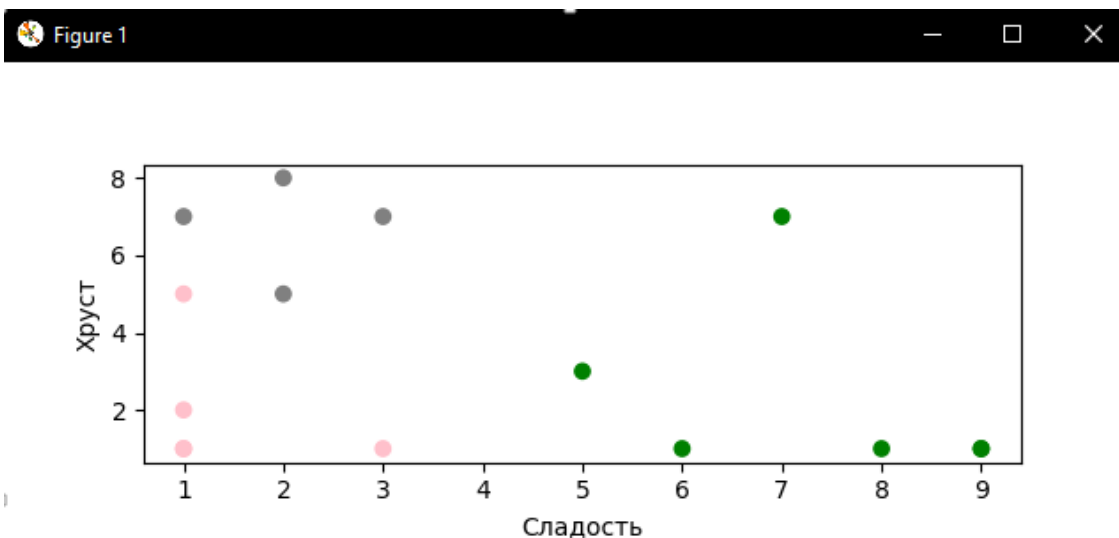
```
# Проверка классов (Предположительный и Оригинальный)
# !Должны все совпадать!
# Сообщает о правильности проведения эксперимента!
```

```
Статистика качества прогнозирования
с использованием sklearn
```

			precision	recall	f1-score	support
0	1.00	1.00	1.00	2		
1	1.00	1.00	1.00	2		
2	1.00	1.00	1.00	1		
accuracy			1.00	5		
macro avg	1.00	1.00	1.00	5		
weighted avg	1.00	1.00	1.00	5		

```
[[2 0 0]
 [0 2 0]
 [0 0 1]]
```

4. Визуализация результатов классификации для набора данных с 3 классами:



5. Проведение эксперимента с новым набором данных с дополнительным классом:

```
0) Малина
Предположительный класс: 0 Оригинальный класс: 0
совпадение
1) Капуста
Предположительный класс: 1 Оригинальный класс: 1
совпадение
2) Халва
Предположительный класс: 3 Оригинальный класс: 3
совпадение
3) Говядина
Предположительный класс: 2 Оригинальный класс: 2
совпадение
4) Свиника
Предположительный класс: 0 Оригинальный класс: 0
совпадение
5) Свекла
Предположительный класс: 1 Оригинальный класс: 1
совпадение
6) Мармелад
Предположительный класс: 3 Оригинальный класс: 3
совпадение
[0, 1, 3, 2, 0, 1, 3]
Совпадений ручного классификатора: 7
```

Классификация новых данных при помощи библиотеки – использованы матрицы статистики и матрицы классификации:

```
Статистика качества прогнозирования
с использованием scikit-learn
precision    recall  f1-score   support

      0       1.00      1.00      1.00         2
      1       1.00      1.00      1.00         2
      2       1.00      1.00      1.00         1
      3       1.00      1.00      1.00         2

 accuracy          1.00         7
 macro avg          1.00         7
weighted avg          1.00         7

[[2 0 0 0]
 [0 2 0 0]
 [0 0 1 0]
 [0 0 0 2]]

Process finished with exit code 0
```

Результат эксперимента при использовании библиотеки также подтверждает корректность результатов ручного классификатора. Классы тестовой выборки состоящей из семи элементов были определены верно.

6. Визуализация результатов классификации для набора данных с 4 классами:

