Министерство образования и науки Российской Федерации
Санкт-Петербургский политехнический университет Петра Великого
Институт металлургии, машиностроения и транспорта
Кафедра «Мехатроника и Роботостроение» при ЦНИИ РТК

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №1

«Классификация: метод k-ближайших соседей (k-nearest neighbors)» Дисциплина:
Математические методы интеллектуальных технологий

Выполнил студент гр. 43328/1 И.С. Ожмегов

Руководитель,
к.т.н, доцент кафедры МиР А.В. Бахшиев

Санкт-Петербург 2018

Задача 1

Установите различные значения числа соседей k, а также задайте веса, чтобы получить классификатор, использующий метод взвешенных ближайших соседей.

Для того чтобы получить классификатор, который использует метод взвешенных ближайших соседей необходимо установить значение параметра **weights** у классификатора **KNeighborsClassifier** значение **distance**. Таким образом получится следующее:

```
knn = neighbors.KNeighborsClassifier(weights='distance')
```

Задача 2

Допишите код, отвечающий за расчет точности по заданным примерам. В качестве заданных примеров сначала используйте обучающую базу.

Для подсчета точности использовалась следующая формула:

$$A = \frac{l - \sum_{i=1}^{l} [a(x_i, X^l, k) \neq y_i]}{l}.$$

Созданная функция представлена ниже.

```
def get_accuracy(method, x_test, y_test):
    accuracy = 0
    n = len(y_test)

    y_new = method.predict(x_test)

    for i in range(n):
        if y_new[i] != y_test[i]:
            accuracy += 1.0

    return 1 - accuracy/n
```

Задача 3

Разделите базу на обучающую и тестовую выборку СЛУЧАЙНЫМ образом и задайте соответствующие выборки для «обучения» и оценки точности алгоритма.

В данной работе разрешено использовать пакет **sklearn**, поэтому воспользуемся одной из его утилит **train_test_split**, которая разбивает массивы случайным образом на

обучающую и тестовую выборки. В программе такое разбиение выглядит следующим образом:

```
X_train, X_test, Y_train, Y_test =
                    train_test_split(X, Y, test_size=0.3, random_state=15)
```

В данном случае для того, чтобы разделение при каждой компиляции оставалось одинаковым используем параметр **random_state**, который является зерном для генератора псевдо случайных чисел.

Задача 4

Найдите параметры, при которых точность будет максимальна. Составьте таблицу результатов, показывающую точность для различных значений параметров. Графики выводить не нужно.

При решении поставленной задачи было обнаружено, что в выборке существуют дублирующиеся элементы, которые вносят значительные ошибки как при обучении, так и при оценки точности, поэтому в таблице 1 представлены результаты без изъятия дубликатов, а в таблице 2 изъятием.

Таблица 1 – Зависимость точности от числа ближайших соседей в выборке без изъятия дубликатов

k, число ближайших соседей	Точность, %
3	77.778
5	80.0
7	80.0
9	75.556
11	77.778
13	80.0
15	80.0
17	80.0
19	80.0
21	80.0
23	80.0
25	80.0
27	80.0
29	80.0

Таблица 2 – Зависимость точности от числа ближайших соседей в выборке с изъятием дубликатов

k, число ближайших соседей	Точность, %
3	89.744
5	89.744
7	92.308
9	92.308
11	92.308
13	92.308
15	92.308
17	92.308
19	92.308
21	92.308
23	92.308
25	92.308
27	92.308
29	92.308