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. 13345/2 И.С. Ожмегов

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

Санкт-Петербург 2019

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 как с изъятием дублирующих элементов, так и без изъятия.

На основе проделанных опытов можно сделать ряд выводов:

1) при небиективном (не взаимно однозначном) отображении наблюдается значительное падение точности;

2) при использовании классификатора, в котором отсутствует метод взвешенных ближайших соседей, наблюдается нестабильность точности при сравнении групп результатов с изъятием дубликатов из выборки.

Таблица 1 – Зависимость точности от числа ближайших соседей

	Выборка без изъятия дубликатов		Выборка с изъятием дубликатов	
k, число ближайших соседей	Точность без весовых коэффициентами, %	Точность с весовыми коэффициентами, %	Точность без весовых коэффициентов, %	Точность с весовыми коэффициентов, %
3	82.222	77.778	82.051	89.744
5	82.222	80.0	87.179	89.744
7	80.0	80.0	92.308	92.308
9	77.778	75.556	84.615	92.308
11	80.0	77.778	87.179	92.308
13	84.444	80.0	87.179	92.308
15	84.444	80.0	87.179	92.308
17	84.444	80.0	89.744	92.308
19	82.222	80.0	89.744	92.308
21	82.222	80.0	89.744	92.308
23	82.222	80.0	89.744	92.308
25	82.222	80.0	89.744	92.308
27	84.444	80.0	92.308	92.308
29	84.444	80.0	89.744	92.308