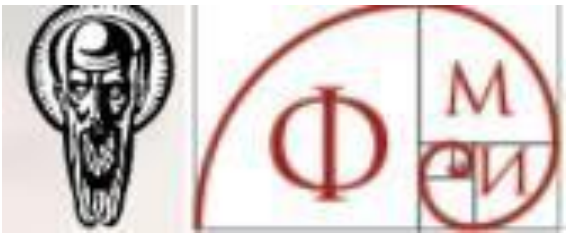

СОФИЙСКИ УНИВЕРСИТЕТ "СВ. КЛИМЕНТ ОХРИДСКИ"
ФАКУЛТЕТ ПО МАТЕМАТИКА И ИНФОРМАТИКА

Домашна работа 2

по Системи, основани на знания

НА ТЕМА: k-NN

Изготвил:
Кенан Юсеин
фак. № 71947
спец. Информационни системи , 3 курс

София,
Януари 2022г

Table of Contents

Описание

Написаната програма е посветено на приложение на метод за машинно самообучение при решаване на задача за класификация. Дадено е множество от обучаващи примери, включващи данни за пациенти, всички от които страдат от едно и също заболяване. По време на курса на лечение всеки пациент е лекуван с едно от дадените лекарства (drugA, drugB, drugC, drugX ,drugY).

Примерите включват данни на пациентите за възраст, пол, кръвно налягане и изследвани нива на холестерол, натрий, калий. Целта е лекарството, с което е лекуван всеки пациент.

	Drug	Age	Sex	BP	Cholesterol	Na_to_K
1	drugY	23	F	HIGH	HIGH	25.355
2	drugC	47	M	LOW	HIGH	13.093
3	drugC	47	M	LOW	HIGH	10.114
4	drugX	28	F	NORMAL	HIGH	7.798
5	drugY	61	F	LOW	HIGH	18.043
6	drugX	22	F	NORMAL	HIGH	8.607
7	drugY	49	F	NORMAL	HIGH	16.275
8	drugC	41	M	LOW	HIGH	11.037

Програмата получава данни за нов пациент и предвижда, кое е най-подходящото лекарство за него спрямо направените опити с предишните пациенти. За целта се трансформират данните и прилага метода на k най-близки съседи (k-NN), оценява и анализира тестовите резултати.

В дадената таблица при поява на нов индивид се изчислява разстоянията до всички вече класифицирани индивиди (чрез изчисление на евклидово разстояние), вземат се първите к елемента, които са най-близо и се прави преценка кое се среща най-често от взетите елементи. На произволен принцип се взима когато има еднакви елементи. За целта к задаваме да е 1, тоест взима най-близкия елемент и спрямо него взима решение за новия индивид.

Описание чрез псевдокод

```
public List<String> predict(List<List<Double>> X) {
    return X.stream().map(this::predictHelper).collect(Collectors.toList());
}
```

```
private String predictHelper(List<Double> x) {
    List<String> nearestN = X.stream()
            .sorted(Comparator.comparingDouble(s → euclideanDistance(s, x)))
            .limit(k)
            .map(s → y.get(X.indexOf(s)))
            .collect(Collectors.toList());

    List<String> distinct = nearestN.stream()
            .distinct()
            .sorted(Comparator.comparingInt(s → Collections.frequency(nearestN, s)))
            .collect(Collectors.toList());

    return distinct.get(distinct.size() - 1);
}
```

```java
public static Double euclideanDistance(List<Double> s1, List<Double> s2) {
    int size = s1.size();
    double s = 0;

    for (int i = 0; i < size; i++) {
        s += Math.pow(s1.get(i) - s2.get(i), 2);
    }

    return Math.sqrt(s);
}
```

Инструкции за компилация

В папката „executable program for testing“ има папка lib, където се намира jar-файла, който е build-нат и готов за пускане чрез Gradle (той съдържа псевдо кода, необходим за пускане на програмата).

За да се пусне програмата е достатъчно да имате инсталирана Java 15.

Отваряте скрипта “runProgram.bat”, който пуска програмата.

„ drug200.csv” е файлът, в който се въвежда информацията в електронна таблица и от която програмата чете входни данни.

Примерни резултати

Входни данни: drug200.csv

1.

```
"C:\Program Files\Java\jdk-15\bin\java.exe" "-javaagent:C:\Program Files
Columns - [Age,Sex,BP,Cholesterol,Na_to_K,Drug]
X - [[23, F, HIGH, HIGH, 25.355, drugY], [47, M, LOW, HIGH, 13.093, drugC],
ages - [23.0, 47.0, 47.0, 28.0, 61.0, 22.0, 49.0, 41.0, 60.0, 43.0, 47.0, 34.0,
genders - [1.0, 0.0, 0.0, 1.0, 1.0, 1.0, 1.0, 0.0, 0.0, 0.0, 1.0, 1.0, 0.0, 1.0, 1
bps - [[0.0, 0.0, 1.0], [1.0, 0.0, 0.0], [1.0, 0.0, 0.0], [0.0, 1.0, 0.0], [1.
chols - [1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 0.0, 1.0, 0.0, 1.0, 1.0, 1
ntk - [25.355, 13.093, 10.114, 7.798, 18.043, 8.607, 16.275, 11.037, 15.171, 1
Y - [drugY, drugC, drugC, drugX, drugY, drugX, drugY, drugC, drugY, drugY, d
bps - [[0.0, 0.0, 1.0, 23.0, 1.0, 1.0, 25.355], [1.0, 0.0, 0.0, 47.0, 0.0, 1.0
Predicted: [drugY, drugY, drugY, drugX, drugB, drugY, drugX, drugY, drug
Actual: [drugY, drugY, drugY, drugX, drugY, drugY, drugX, drugY, drugA,
31
39
Accuracy: 0,794872
```

2.

```
Predicted: [drugY, drugY, drugY, drugY, drugA, drugY, drugY, drugY, drugB, drugY, drugX, drugY, drugY, drugX, drugC, drugY, drugA, dr
Actual: [drugY, drugY, drugY, drugY, drugA, drugY, drugY, drugY, drugY, drugY, drugX, drugY, drugY, drugB, drugC, drugY, drugY, drugX
26
39
Accuracy: 0,666667
```

3.

```
Predicted: [drugA, drugX, drugX,
Actual: [drugA, drugX, drugX, dru
29
39
Accuracy: 0,743590
```