**Задание KNN1 - Решение задачи классификации методом kNN**

Выполнил студент 2 курса

группы 09-715(1)

Санамян Артак Размикович

**Текст задания:**

1. Найдите реальные данные для задачи классификации. Зафиксируйте свой выбор в приведённом ниже объекте "Wiki".

2. Перечислите факторы (и укажите единицы их измерения) и поясните смысл переменной отклика.

3. Если число факторов m <= 2, то визуализируйте данные (постройте облако точек).

4. Реализуйте метод kNN для найденных данных.

5. Оцените точность полученного решения с помощью метода кросс-валидации. (Скриншот кросс-валидационнной таблицы включите в отчёт. Поясните, как Вы обеспечили репрезентативность обучающей выборки.)

6. Задайте нескольких новых данных. Если m <= 2, то покажите соответствующие точки на графике (выделите их другим цветом).

7. Определите значение переменной отклика (номер класса) для новых данных.

8. Оформите отчёт в виде Word-файла и прикрепите его здесь.

Источник числовых данных, использованный для выполнения задания: <http://archive.ics.uci.edu/ml/datasets/Flags>

**Описание выполненной работы:**

Данные представляют собой информацию о странах на 1990 год. Необходимо определить религию страны:

0 – Католики;

1 – Остальные христиане;

2 – Мусульмане;

3 – Буддисты;

4 – Индуисты;

5 – Язычники;

6 – Марксисты;

7 – Другие.

Факторами являются следующие признаки:

1. Название страны (не числовой);
2. Материк (1-6);
3. Зона: географический квадрант, основанный на Гринвиче и экваторе(1-4)
4. Площадь: в тысячах квадратных километров;
5. Население: в миллионах;
6. Язык: 1 = английский, 2 = испанский, 3 = французский, 4 = немецкий, 5 = славянский, 6 = другой индоевропейский, 7 = китайский, 8 = арабский, 9 = японский / турецкий / финский / мадьяр, 10 = Другие;
7. Религия: 0 = католик, 1 = другой христианин, 2 = мусульманин, 3 = буддист, 4 = индус, 5 = этнический, 6 = марксист, 7 = другие;
8. Полосы: количество вертикальных полос на флаге;
9. Полосы: количество горизонтальных полос на флаге;
10. Цвета: количество различных цветов на флаге;
11. Красный: 0 – нет, 1 – цвет есть на флаге;
12. Зеленый: 0 – нет, 1 – цвет есть на флаге;
13. Синий: 0 – нет, 1 – цвет есть на флаге;
14. Золотой: 0 – нет, 1 – цвет есть на флаге (также желтый);
15. Белый: 0 – нет, 1 – цвет есть на флаге;
16. Черный: 0 – нет, 1 – цвет есть на флаге;
17. Оранжевый: 0 – нет, 1 – цвет есть на флаге (также коричневый);
18. Основной оттенок (не числовой);
19. Количество кругов;
20. Количество крестов;
21. Количество Андреевских крестов
22. Количество частей флага;
23. Количество символов солнца или звезд;
24. Полумесяц: 1, если присутствует символ полумесяца, иначе 0
25. Треугольник: 1, если присутствуют какие-либо треугольники, 0 в противном случае
26. 1, если присутствует неодушевленное изображение (например, лодка), в противном случае 0
27. 1, если присутствует одушевленное изображение (например, орел, дерево, человеческая рука), 0 в противном случае
28. Текст: 1 если какие-либо буквы или письма на флаге (например, девиз или слоган), 0 в противном случае
29. Цвет в верхнем левом углу;
30. Цвет в левом нижнем углу.

Применим нормализацию данных по формуле:

Подсчитаем частоты встречаемости классов в исходной выборке:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 20.6 | 30.9 | 18.6 | 4.1 | 2.1 | 13.9 | 7.7 | 2.1 |

Подсчитаем частоту присутствия каждого класса в обучающей выборке:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 20.0 | 24.5 | 20.0 | 5.5 | 1.8 | 17.3 | 8.2 | 2.7 |

Репрезентативность выборки обеспечена путем ее перемешивания.

Рассмотрим кросс-валидационную таблицу:

Cell Contents

|-------------------------|

| N |

| N / Row Total |

| N / Col Total |

| N / Table Total |

|-------------------------|

Total Observations in Table: 84

| data\_test\_pred

test\_data\_labels | 0 | 1 | 2 | 3 | 5 | Row Total |

-----------------|-----------|-----------|-----------|-----------|-----------|-----------|

0 | 10 | 3 | 2 | 0 | 3 | 18 |

| 0.556 | 0.167 | 0.111 | 0.000 | 0.167 | 0.214 |

| 0.385 | 0.136 | 0.100 | 0.000 | 0.214 | |

| 0.119 | 0.036 | 0.024 | 0.000 | 0.036 | |

-----------------|-----------|-----------|-----------|-----------|-----------|-----------|

1 | 11 | 16 | 3 | 0 | 3 | 33 |

| 0.333 | 0.485 | 0.091 | 0.000 | 0.091 | 0.393 |

| 0.423 | 0.727 | 0.150 | 0.000 | 0.214 | |

| 0.131 | 0.190 | 0.036 | 0.000 | 0.036 | |

-----------------|-----------|-----------|-----------|-----------|-----------|-----------|

2 | 0 | 1 | 9 | 1 | 3 | 14 |

| 0.000 | 0.071 | 0.643 | 0.071 | 0.214 | 0.167 |

| 0.000 | 0.045 | 0.450 | 0.500 | 0.214 | |

| 0.000 | 0.012 | 0.107 | 0.012 | 0.036 | |

-----------------|-----------|-----------|-----------|-----------|-----------|-----------|

3 | 0 | 0 | 2 | 0 | 0 | 2 |

| 0.000 | 0.000 | 1.000 | 0.000 | 0.000 | 0.024 |

| 0.000 | 0.000 | 0.100 | 0.000 | 0.000 | |

| 0.000 | 0.000 | 0.024 | 0.000 | 0.000 | |

-----------------|-----------|-----------|-----------|-----------|-----------|-----------|

4 | 0 | 1 | 0 | 0 | 1 | 2 |

| 0.000 | 0.500 | 0.000 | 0.000 | 0.500 | 0.024 |

| 0.000 | 0.045 | 0.000 | 0.000 | 0.071 | |

| 0.000 | 0.012 | 0.000 | 0.000 | 0.012 | |

-----------------|-----------|-----------|-----------|-----------|-----------|-----------|

5 | 3 | 0 | 2 | 0 | 3 | 8 |

| 0.375 | 0.000 | 0.250 | 0.000 | 0.375 | 0.095 |

| 0.115 | 0.000 | 0.100 | 0.000 | 0.214 | |

| 0.036 | 0.000 | 0.024 | 0.000 | 0.036 | |

-----------------|-----------|-----------|-----------|-----------|-----------|-----------|

6 | 2 | 1 | 2 | 1 | 0 | 6 |

| 0.333 | 0.167 | 0.333 | 0.167 | 0.000 | 0.071 |

| 0.077 | 0.045 | 0.100 | 0.500 | 0.000 | |

| 0.024 | 0.012 | 0.024 | 0.012 | 0.000 | |

-----------------|-----------|-----------|-----------|-----------|-----------|-----------|

7 | 0 | 0 | 0 | 0 | 1 | 1 |

| 0.000 | 0.000 | 0.000 | 0.000 | 1.000 | 0.012 |

| 0.000 | 0.000 | 0.000 | 0.000 | 0.071 | |

| 0.000 | 0.000 | 0.000 | 0.000 | 0.012 | |

-----------------|-----------|-----------|-----------|-----------|-----------|-----------|

Column Total | 26 | 22 | 20 | 2 | 14 | 84 |

| 0.310 | 0.262 | 0.238 | 0.024 | 0.167 | |

-----------------|-----------|-----------|-----------|-----------|-----------|-----------|

Зададим несколько новых данных:

1. Australia,6,2,7690,15,1,1,0,0,3,1,0,1,0,1,0,0,blue,0,1,1,1,6,0,0,0,0,0,white,blue
2. Austria,3,1,84,8,4,0,0,3,2,1,0,0,0,1,0,0,red,0,0,0,0,0,0,0,0,0,0,red,red
3. Bahamas,1,4,19,0,1,1,0,3,3,0,0,1,1,0,1,0,blue,0,0,0,0,0,0,1,0,0,0,blue,blue
4. Bahrain,5,1,1,0,8,2,0,0,2,1,0,0,0,1,0,0,red,0,0,0,0,0,0,0,0,0,0,white,red
5. Bangladesh,5,1,143,90,6,2,0,0,2,1,1,0,0,0,0,0,green,1,0,0,0,0,0,0,0,0,0,green,green
6. Barbados,1,4,0,0,1,1,3,0,3,0,0,1,1,0,1,0,blue,0,0,0,0,0,0,0,1,0,0,blue,blue
7. Belgium,3,1,31,10,6,0,3,0,3,1,0,0,1,0,1,0,gold,0,0,0,0,0,0,0,0,0,0,black,red
8. Belize,1,4,23,0,1,1,0,2,8,1,1,1,1,1,1,1,blue,1,0,0,0,0,0,0,1,1,1,red,red
9. Benin,4,1,113,3,3,5,0,0,2,1,1,0,0,0,0,0,green,0,0,0,0,1,0,0,0,0,0,green,green
10. Bermuda,1,4,0,0,1,1,0,0,6,1,1,1,1,1,1,0,red,1,1,1,1,0,0,0,1,1,0,white,red
11. Bhutan,5,1,47,1,10,3,0,0,4,1,0,0,0,1,1,1,orange,4,0,0,0,0,0,0,0,1,0,orange,red
12. Bolivia,2,3,1099,6,2,0,0,3,3,1,1,0,1,0,0,0,red,0,0,0,0,0,0,0,0,0,0,red,green

Определим значение переменной отклика (номер класса) для новых данных:

1. 2 (1)
2. 2 (0)
3. 0 (1)
4. 1 (2)
5. 2 (2)
6. 1 (1)
7. 6 (0)
8. 1 (1)
9. 2 (5)
10. 1 (1)
11. 6 (3)
12. 2 (0)

**Выводы:**

Как видно из кросс - валидационной таблицы, метод KNN допустил 44 ошибки из 84. Процент ошибок = 52, 38.Таким образом, точность метода составляет 47, 62 %.

**Код программы:**

setwd("C://Users/pc/Documents/Магистратура/2 семестр/R/task3 kNN")

getwd()

# Прочитаем файл с данными в переменную flagdata

flagdata <- read.csv("flagData.txt", head = FALSE, sep=",") #, sep="\t"

flagdata

# Формула нормализации данных

normalize <- function(x) {return ((x - min(x)) / (max(x) - min(x)))}

# Нормализуем данные

# исключим не численные факторы и переменную отклика

flag <- as.data.frame(lapply(flagdata[c(2:6, 8:17, 19:28)], normalize))

# добавим перменную отклика в конец для удобства

flag[26] = flagdata[7]

flag

# Прочитали 194 записей, в каждой - 26 значений, где

# последняя (26-я) переменная - номер класса (переменная отклика)

# Часть выборки, например, 110 записей, будем использовать как

# обучающую

# Сохраним объём всей выборки и объём обучающей выборки в переменных

n\_data = 194

n\_train = 110

# Обучающая выборка должна быть репрезентативной -

# т.е. каждый класс должен быть представлен в ней в той же пропорции,

# что и во всей выборке.

# Подсчитаем частоты встречаемости классов в исходной выборке

# (выразим эти частоты в %, округлив результат до десятых)

round(prop.table(table(flag[26]))\*100, digits = 1)

# Чтобы обеспечить репрезентативность выборки, перемешаем её

set.seed(12345)

flagdata\_mixed=flag[order(runif(n\_data)),]

# Выберем обучающую выборку

train\_data = flagdata\_mixed[1:n\_train,]

# Сохраним номера классов для строк обучающей выборки

train\_data\_labels = train\_data[,26]

# Перед вызовом метода knn удалим номера классов из train\_data

# Подсчитаем частоту присутствия каждого класса в обучающей выборке

# и сравним с соответствующими частотами в исходной выборке

round(prop.table(table(train\_data[26]))\*100, digits = 1)

# Оставшуюся часть "перемешанной выборки" будем использовать

# как тестовую выборку

test\_data = flagdata\_mixed[(n\_train+1):n\_data, ]

test\_data\_labels = test\_data[,26]

# Удалим из обучающей и тестовой выборок столбец с номерами классов

train\_data = train\_data[-26]

test\_data = test\_data[-26]

# Подключим пакет class

library("class")

data\_test\_pred <- knn(train = train\_data,

test = test\_data,

cl = train\_data\_labels,

k = round(sqrt(n\_train)))

# Вызовем метод knn для тестовых данных

# Результат (номера классов) запишем в вектор data\_test\_pred

# Для оценки качества прогноза

# подключим библиотеку gmodels

library("gmodels")

# Построим кросс-валидационную таблицу:

CrossTable(x = test\_data\_labels, y = data\_test\_pred, prop.chisq=FALSE)

# Зададим новые данные

predict <- read.csv("predict.txt", sep=",", head=FALSE)

predict = predict[c(2:6, 8:17, 19:28)]

# Определим значение переменной отклика (номер класса) для новых данных

new\_predict <- knn(train = train\_data,

test = predict,

cl = train\_data\_labels,

k = round(sqrt(n\_train)))

new\_predict