**Лабораторная работа №1**

**“Наивный Байесовский классификатор”**

по дисциплине “Статическое моделирование случайных процессов и систем”

|  |  |
| --- | --- |
| Выполнил  студент гр. 33504/4 | Череповский Д.К. |
| Руководитель | Селин И.А. |

**Оглавление**

[**Задание 3**](#_Toc507859789)

[**Анализ 3**](#_Toc507859790)

[**Реализация 5**](#_Toc507859791)

# **Задание**

1. Исследуйте, как объем обучающей выборки и количество тестовых данных, влияет на точность классификации или на вероятность ошибочной классификации в примере крестики-нолики.

2. Сгенерируйте 100 точек с двумя признаками X1 и X2 в соответствии с нормальным распределением так, что первые 50 точек (class -1) имеют параметры: мат. ожидание X1 равно 10, мат. ожидание X2 равно 14, среднеквадратические отклонения для обеих переменных равны 4. Вторые 50 точек (class +1) имеют параметры: мат. ожидание X1 равно 20, мат. ожидание X2 равно 18, среднеквадратические отклонения для обеих переменных равны 3. Построить соответствующие диаграммы, иллюстрирующие данные. Построить байесовский классификатор и оценить качество классификации.

# **Анализ**

1. Варьируя значение объема обучающей выборки в примере с крестиками-ноликами (Tic\_tac\_toe), была построена следующая таблица:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Объём обучающей выборки | Верно | Неверно | Верных % | Неверных % |
| 0,1 | 463 | 163 | 0,6570104 | 0,3429896 |
| 0,2 | 405 | 139 | 0,6636245 | 0,3363755 |
| 0,3 | 370 | 132 | 0,6602086 | 0,3397914 |
| 0,4 | 315 | 115 | 0,6608696 | 0,3577023 |
| 0,5 | 271 | 91 | 0,6722338 | 0,3277662 |
| 0,6 | 217 | 74 | 0,6770833 | 0,3229167 |
| 0,7 | 163 | 53 | 0,6979167 | 0,3020833 |
| 0,8 | 110 | 40 | 0,6770833 | 0,3229167 |
| 0,9 | 58 | 18 | 0.6670833 | 0,3229167 |

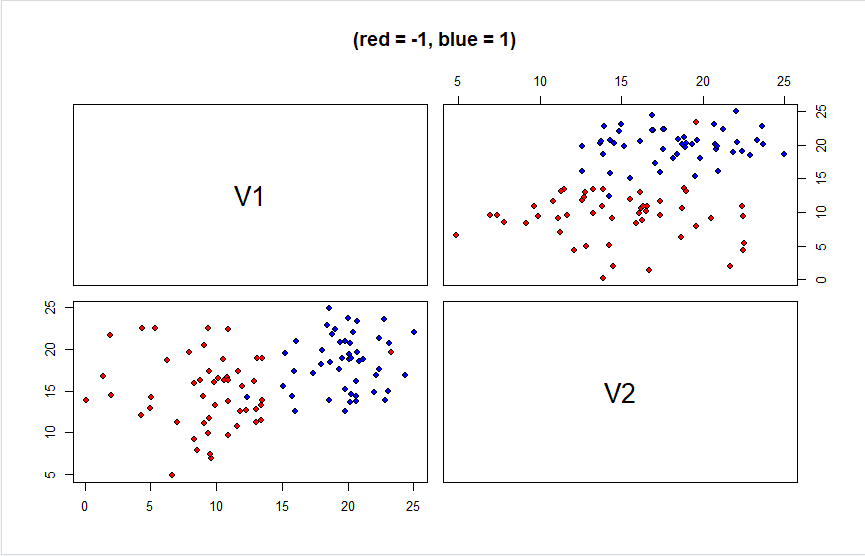
Из таблицы видно, что объем выборки почти не влияет на вероятность ошибочной классификации в данном примере.

1. Варьируя значение объёма тестовой выборки в примере о спаме, была построена следующая таблица

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Объем тестовой выборки | Верно | Неверно | Верных% | Неверных% |
| 20 | 14 | 6 | 0.7 | 0.3 |
| 50 | 29 | 21 | 0.58 | 0.42 |
| 100 | 75 | 25 | 0.75 | 0.25 |
| 500 | 343 | 151 | 0.694 | 0.306 |
| 1000 | 716 | 284 | 0.716 | 0.284 |

Из таблицы видно, что объем выборки почти не оказывает влияние на вероятность ошибочной классификации.

1. После генерации точек в соответствии с заданием получаем диаграмму:



После построения байесовского метода, варьируя значение объёма обучающей выборки получили:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Объём обучающей выборки | Верно | Неверно | Верных % | Неверных % |
| 0,1 | 19 | 1 | 0,95 | 0,05 |
| 0,2 | 18 | 2 | 0,9 | 0,1 |
| 0,3 | 27 | 3 | 0,9 | 0,1 |
| 0,4 | 38 | 2 | 0,95 | 0,05 |
| 0,5 | 46 | 4 | 0,92 | 0,08 |
| 0,6 | 57 | 3 | 0,95 | 0,05 |
| 0,7 | 66 | 4 | 0,94 | 0,057 |
| 0,8 | 78 | 2 | 0,975 | 0,025 |
| 0,9 | 76 | 14 | 0,84 | 0,156 |

Точность падает только при большом объёме тестовой выборки.

# **Реализация**

1.

#install.packages("e1071")

library(e1071)

### Naive Bayesian (данные категориальные) #########################

# 1 #############################################################

# импортируем данные в R

# установить параметр stringsAsFactors = TRUE,

# так как все данные - категориальные

A\_raw <- read.table("C:/Users/gdk17/OneDrive/Рабочий стол/универ/machine\_learning/Lab1\_NaiveBayesian/Tic\_tac\_toe.txt", sep = ",", stringsAsFactors = TRUE)

# число строк в базе

n <- dim(A\_raw)[1]

# Создан фрейм, который можно просмотреть, используя str(A\_raw).

# Имеется 9 столбцов признаков V1-V9 и V10 (класс) и

# все имеют один и тот же тип Factor.

# 2 #############################################################

# Создание обучающей и тестирующей выборки

# Скажем, имеем n примеров в исходной выборке,

# используем 80% для обучения и оставшиеся - для тестирования.

# Устанавливаем базу генерации случайных чисел и рандомизируем выборку

set.seed(12345)

A\_rand <- A\_raw[ order(runif(n)), ]

# разделим данные на обучающие и тестирующие

nt <- as.integer(n\*0.9)

A\_train <- A\_rand[1:nt, ]

A\_test <- A\_rand[(nt+1):n, ]

# Можно убедиться, какой имеется процент каждого

# класса V2 в обучающей и тестирующей выборке

prop.table(table(A\_train$V10))

prop.table(table(A\_test$V10))

# 3 ############################################################

# Используем Наивный Байесовский классификатор из пакета e1071

# A\_classifier <- naiveBayes(A\_train[,-10], A\_train$V10)

# Другой вариант классификатора

A\_classifier <- naiveBayes(V10 ~ ., data = A\_train)

# 4 ############################################################

# Теперь оценим полученную модель:

A\_predicted <- predict(A\_classifier, A\_test)

# Используем table для сравнения прогнозируемых значений с тем, что есть

table(A\_predicted, A\_test$V10)

pairs(A\_train, col=A\_train$V10)

library(kernlab)

library(e1071)

data(spam)

for(size in seq(20, 4581, by = 500))

{

idx <- sample(1:dim(spam)[1], size)

spamtrain <- spam[-idx, ]

spamtest <- spam[idx, ]

model <- naiveBayes(type ~ ., data = spamtrain)

t <- table(predict(model, spamtest), spamtest$type)

print(t)

}

array1 <- rnorm(50, 10, 4) #признак X1

array2 <- rnorm(50, 14, 4) #признак x2

array3 <- rnorm(50, 20, 3) #признак X1

array4 <- rnorm(50, 18, 3) #признак x2

A\_raw = data.frame(c(array1, array3), c(array2, array4), factor(c(rep(-1, 50), rep(1, 50))))

colnames(A\_raw) <- c("V1", "V2", "class")

pairs(A\_raw[1:2], main="(red = -1, blue = 1)", pch = 21, bg = c("red", "blue")[unclass(A\_raw$class)])

A\_rand <- A\_raw[ order(runif(n)), ]

n <- dim(A\_raw)[1]

nt <- as.integer(n\*0.1)

A\_train <- A\_rand[1:nt, ]

A\_test <- A\_rand[(nt+1):n, ]

prop.table(table(A\_train$class))

prop.table(table(A\_test$class))

model <- naiveBayes(class ~ ., data = A\_train)

print(table(predict(model, A\_test), A\_test$class))