Лабораторная работа №1

Наивный Байесовский классификатор

Выполнил

студент гр. 33504/2 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Лелюхин Д. О.

Руководитель: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Селин И. А.

**Формулировка Задания**

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

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.

### Выполнение 1-ого задания:

#### Код программы на языке R:

library(e1071)

A\_raw <- read.table("Tic\_tac\_toe.txt", sep = ",", stringsAsFactors = TRUE)

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

set.seed(12345)

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

for(i in seq(0.1, 0.9, by = 0.1))

{

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

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

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

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

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

t <- table(A\_predicted, A\_test$V10)

print(t)

}

#### Результаты работы программы:

A\_predicted negative positive

negative 133 104

positive 163 463

A\_predicted negative positive

negative 119 104

positive 139 405

A\_predicted negative positive

negative 96 73

positive 132 370

A\_predicted negative positive

negative 80 63

positive 115 317

A\_predicted negative positive

negative 66 51

positive 91 271

A\_predicted negative positive

negative 50 43

positive 74 217

A\_predicted negative positive

negative 34 38

positive 53 163

A\_predicted negative positive

negative 22 20

positive 40 110

A\_predicted negative positive

negative 13 7

positive 18 58

- - - - - -

#### Код программы на языке R:

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)

}

#### Результаты работы программы:

[1] 4581

[1] 20

nonspam spam

nonspam 6 1

spam 5 8

[1] 4081

[1] 520

nonspam spam

nonspam 166 10

spam 140 204

[1] 3581

[1] 1020

nonspam spam

nonspam 333 21

spam 264 402

[1] 3081

[1] 1520

nonspam spam

nonspam 524 34

spam 396 566

[1] 2581

[1] 2020

nonspam spam

nonspam 650 46

spam 572 752

[1] 2081

[1] 2520

nonspam spam

nonspam 899 60

spam 622 939

[1] 1581

[1] 3020

nonspam spam

nonspam 1080 58

spam 755 1127

[1] 1081

[1] 3520

nonspam spam

nonspam 1174 83

spam 984 1279

[1] 581

[1] 4020

nonspam spam

nonspam 1284 73

spam 1153 1510

[1] 81

[1] 4520

nonspam spam

nonspam 960 79

spam 1778 1703

## Вывод

В результате, мы получаем что, при уменьшении объема обучающей выборки уменьшается количество угаданных результатов.

## Выполнение 2-ого задания:

#### Код программы на языке R:

library(e1071)

x1\_1 <- rnorm(50, mean = 10, sd = 4)

x1\_2 <- rnorm(50, mean = 20, sd = 3)

x2\_1 <- rnorm(50, mean = 14, sd = 4)

x2\_2 <- rnorm(50, mean = 18, sd = 4)

plot(x1\_1, x2\_1, pch=21, xlim=c(0, 30),ylim=c(0, 30))

points(x1\_2,x2\_2, pch=22)

x1 <- c(x1\_1, x1\_2)

x2 <- c(x2\_1, x2\_2)

class<-c(rep('-1',50),rep('1',50))

t<-data.frame(x1, x2, class, stringsAsFactors = TRUE)

for (i in seq(20,80,by=10))

{

idx<-sample(1:dim(t)[1],20)

train<-t[-idx, ]

test<-t[idx, ]

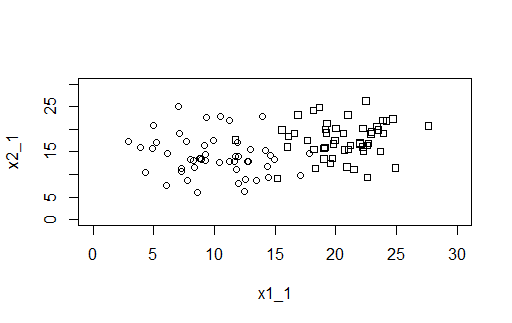
model<-naiveBayes(train[,-3],train$class)

t\_predicted<-predict(model, test)

print(table(t\_predicted,test$class))

}

#### Результаты работы программы:



t\_predicred -1 1

-1 10 0

1 0 10

t\_predicred -1 1

-1 6 2

1 0 12

t\_predicred -1 1

-1 6 2

1 2 10

t\_predicred -1 1

-1 11 1

1 0 8

t\_predicred -1 1

-1 10 1

1 1 8

t\_predicred -1 1

-1 7 0

1 2 11

t\_predicred -1 1

-1 9 2

1 1 8