|  |  |
| --- | --- |
| Gerb-BMSTU_01 | **Министерство науки и высшего образования Российской Федерации**  Калужский филиал  федерального государственного бюджетного  образовательного учреждения высшего образования  ***«Московский государственный технический университет имени Н.Э. Баумана (национальный исследовательский университет)»***  ***(КФ МГТУ им. Н.Э. Баумана)*** |

**ФАКУЛЬТЕТ** ***ИУК «Информатика и управление»***

**КАФЕДРА** \_\_***ИУК4 «Программное обеспечение ЭВМ, информационные технологии»***

**ЛАБОРАТОРНАЯ РАБОТА №4**

**«Логические методы классификации многомерных объектов пересекающихся классов»**

**ДИСЦИПЛИНА: «Методы машинного обучения»**

|  |  |  |
| --- | --- | --- |
| Выполнил: студент гр. ИУК4-62Б | | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ ( Калашников А.С. )  (Подпись) (Ф.И.О.) |
| Проверил: | | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ (      Кручинин И.И. )  (Подпись) (Ф.И.О.) |
| Дата сдачи (защиты):  Результаты сдачи (защиты): | | |
|  | - Балльная оценка:  - Оценка: | |

Калуга, 2023

**Цель работы:** изучение основных процедур классификации многомерных объектов пересекающихся классов.

**Вариант 2**

Разработать логический классификатор с использованием алгоритмов «Кора», «ID3», «CART», «Бэггинг», «Бустинг», генетического алгоритма для классификации товаров супермаркета по категориям «Скидки - Нет» 0-4 %, «Скидка-мини» 5-25 %, «Выгодная Скидка» 26-40%, «Супер Скидка» 50-70% за пять дней.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Код | Кп | Кв | Кс | Кч | Кп | Ксб | Квс | Кз | Квн | Кл | Ко | Кт |  |
| 011 | 1 |  |  |  |  |  |  | 0.7 | 0.54 | 1 | 1 | 0.45 |  |
| 012 | 0.77 |  |  |  |  |  |  |  |  |  |  | 1 |  |
| 013 |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 014 |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 015 |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 016 |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 017 |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 018 |  |  |  |  |  |  |  |  |  |  |  | 0.15 |  |
| 019 | 1 |  | 1 |  |  |  | 1 |  |  | 1 |  | 0.23 |  |
| 020 | 0.58 |  |  |  |  |  |  |  |  |  |  |  |  |

1). Для алгоритма Кора: выбрать частоту встречаемости конъюнкций MinNum= 4

2). Для генетического алгоритма выбрать: генную бинарную комбинацию **1101101**

**3). Для алгоритма CART в функции rpart выбрать параметр** method = "anova"

**4). Для алгоритма Bagging в функции RandomForest выбрать параметр N.trees= 400, в функции train выбрать параметр** method = "Adabag"

5). Для алгоритма Boosting в функции gbm выбрать параметр N.trees = 300, параметр distribution = “bernoully”, параметр bag.Fraction = 0.47

6). Результаты визуализировать и сравнить.

**Листинг:**

DATA <- read.csv2(file = "file.csv", header = TRUE, row.names = 1)

head(DATA)

number2binchar <- function(number, nBits) {

paste(tail(rev(as.numeric(intToBits(number))), nBits),collapse = "")

}

# Поиск конъюнкций по набору битовых масок

MaskCompare <- function(Nclass, KSize, BitMask,vec\_pos, vec\_neg, ColCom) {

nK <- sapply(BitMask, function(x) {

if (sum(x == vec\_neg) > 0) return (0)

if (minNum > (countK = sum(x == vec\_pos))) return(0)

# Cохранение конъюнкции в трех объектах list

Value.list[[length(Value.list) + 1]] <<-

list(Nclass = Nclass, KSize = KSize, countK = countK, Bits = x)

ColCom.list[[length(ColCom.list) + 1]] <<- list(ColCom)

RowList.list[[length(RowList.list) + 1]] <<-

list(which(vec\_pos %in% x))

return(countK) } )

}

DFace <- read.csv2(file = "file.csv", header = TRUE, row.names = 1)

maxKSize <- 4

minNum <- 4

# Списки для хранения результатов

Value.list <- list() # Nclass, KSize, BitMask, countK

ColCom.list <- list() # Наименования переменных ColCom

RowList.list <- list() # Номера индексов строк RowList

# Перебор конъюнкций разной длины

for (KSize in 2:maxKSize) {

BitMask <- sapply(0:(2^KSize - 1),function(x) number2binchar(x, KSize))

cols <- combn(colnames(DFace[, -17]), KSize)

for (i in 1:ncol(cols)) {

SubArr <- DFace[, (names(DFace) %in% cols[, i])]

vec1 <- apply(SubArr[DFace$Class == 1, ],1,function(x) paste(x, collapse = ""))

vec2 <- apply(SubArr[DFace$Class == 2,], 1,function(x) paste(x, collapse = ""))

MaskCompare(1, KSize, BitMask, vec1, vec2, cols[, i])

MaskCompare(2, KSize, BitMask, vec2, vec1, cols[, i])

}

}

# Создание результирующей таблицы

DFval = do.call(rbind.data.frame, Value.list)

nrow = length(Value.list)

DFvar <- as.data.frame(matrix(NA, ncol = maxKSize + 1, nrow = nrow,

dimnames = list(1:nrow, c(paste("L", 1:maxKSize, sep = ""),"Объекты:"))))

for (i in 1:nrow) {

Varl <- unlist(ColCom.list[[i]])

DFvar[i, 1:length( Varl)] <- Varl

Objl <- unlist(RowList.list[[i]])

DFvar[i, maxKSize + 1] <- paste(Objl, collapse = " ")

}

DFout <- cbind(DFval, DFvar)

print("Конъюнкции класса 1")

DFout[DFout$Nclass == 1, ]

print("Конъюнкции класса 2")

DFout[DFout$Nclass == 2, ]

library(genalg)

library(ggplot2)

dataset <- read.csv2(file = "file.csv", header = TRUE, row.names = 1)

weightlimit <- 20

chromosome = c(1, 1, 0, 1, 1, 0, 1)

dataset[chromosome == 1, ]

evalFunc <- function(x) {

current\_solution\_survivalpoints <- x %\*% dataset$survivalpoints

current\_solution\_weight <- x %\*% dataset$weight

if (current\_solution\_weight > weightlimit)

return(0) else return(-current\_solution\_survivalpoints)

}

Dataset <- read.csv2(file = "file.csv", header = TRUE, row.names = 1)

library(rpart)

# grow tree

fit <- rpart(Dataset$Наименование ~ Dataset$коэффициент.понедельника + Dataset$Эталонная.цена,method="anova", data=Dataset)

printcp(fit) # display the results

plotcp(fit) # visualize cross-validation results

summary(fit) # detailed summary of splits

# plot tree

plot(fit, uniform=TRUE, main="Classification Tree")

text(fit, use.n=TRUE, all=TRUE, cex=.8)

post(fit, title = "Classification Tree")

library(randomForest)

library(caret)

x <- read.csv2(file = "file.csv", header = TRUE, row.names = 1)

set.seed(101)

model <- randomForest(x$Наименование ~ x$Эталонная.цена, data=x, ntree=400, mtry=9)

set.seed(101)

bag.a1 <- train(x, preProc = c('center', 'scale'),

method = 'Adabag', tuneGrid = expand.grid(.mtry = ncol(x)))

plot(ranfor.a1$finalModel, col = "blue", lwd = 2)

plot(bag.a1$finalModel, col = "green", lwd = 2, add = TRUE)

legend("topright", c("Bagging", "RandomForrest"),col = c("green","blue"), lwd = 2)

library(gbm)

x <- read.csv2(file = "file.csv", header = TRUE, row.names = 1)

set.seed(1)

xd <- cbind(a1 = x$Эталонная.цена, x)

boost.a1 = gbm(a1 ~ ., data = xd, distribution = "bernoully", n.trees = 300, interaction.depth = 3)

summary(boost.a1, plotit = FALSE)

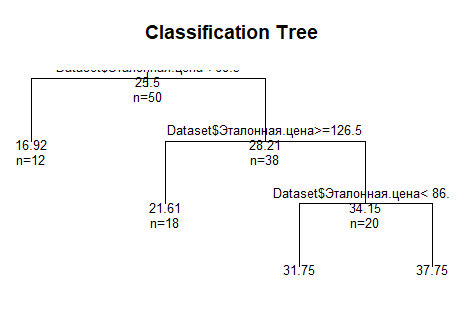
library(bst)

(boostFit.a1 <- train(a1 ~ ., data = xd, method = 'bstTree',

trControl = trainControl(method = "cv"), preProc = c('center', 'scale')))

plot(boostFit.a1)

**Результаты работы:**



**Вывод:** были изучены основные процедуры классификации многомерных объектов пересекающихся классов.