Министерство образования и науки 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)

И.И. Кручинин

(к.т.н. доцент)

**ЛАБОРАТОРНАЯ РАБОТА № 1**

**по курсу «Методы машинного обучения»**

## ЛИНЕЙНЫЕ КЛАССИФИКАТОРЫ

**ПРИЛОЖЕНИЕ**

Калуга

2018

**Примеры программ на языке *R***

**Пример 1. Реализация вычислительного процесса**

rm(list = ls()) #очистка памяти

kol = function(x) { #количество эл в векторе

x = length(min(x):max(x))

return(x)

}

div = function(x, k = length(x) / 10) { # разбиение вектора на k равных интервалов

#tmp = round(seq(min(x), max(x), len = k + 1))

tmp = seq(min(x), max(x), len = k + 1)

print(tmp)

i = 1

x\_abc = list()

y\_ord = list()

while (i < length(tmp)) {

str = tmp[i]: tmp[i + 1]

print(str)

x\_abc = c(median(str), x\_abc)

y\_ord = c(kol(str) / kol(tmp), y\_ord)

text = c("количество элементов равно" , kol(str))

print(text)

i = i + 1

}

plot(x\_abc, y\_ord, type = "o")

}

#x = round(runif(20, 1, 100))

#print(x)

#div(x, 5)

x = runif(5000, 1, 100)

l = 500

func = l \* exp(-l \* x)

#print(func)

div(func, 100)

**Пример 2. Построение фрейма**

rm(list = ls()) #очистка памяти

kol = function(x) { #количество эл в векторе

x = length(min(x):max(x))

return(x)

}

div = function(x, k = length(x) / 10) { # разбиение вектора на k равных интервалов

#tmp = round(seq(min(x), max(x), len = k + 1))

tmp = seq(min(x), max(x), len = k + 1)

print(tmp)

i = 1

x\_abc = list()

y\_ord = list()

while (i < length(tmp)) {

str = tmp[i]: tmp[i + 1]

print(str)

x\_abc = c(median(str), x\_abc)

y\_ord = c(kol(str) / kol(tmp), y\_ord)

text = c("количество элементов равно" , kol(str))

print(text)

i = i + 1

}

plot(x\_abc, y\_ord, type = "o")

}

#x = round(runif(20, 1, 100))

#print(x)

#div(x, 5)

x = runif(5000, 1, 100)

l = 500

func = l \* exp(-l \* x)

#print(func)

div(func, 100)

**Данные из текстового файла:**

country population area gdp ind

China 1319498000 9640821 2630113 ind

India 1169016000 3287263 886867 ind

USA 302425000 9629091 13244550 ind

Indonesia 231627000 1904569 364239 ind

Brazil 186736000 8514877 1067706 ind

Pakistan 160757000 880254 128996 ind

Bangladesh 158665000 143998 65216 ind

Nigeria 148093000 923768 115350 ind

Russia 142499000 17098242 979048 ind

**Пример 3. Построениемашиныопорныхвекторов.**

#install.packages('e1071\_1.6-8.zip', dependencies = TRUE)

library(e1071)

data1 = data.frame(read.table("svmdata3.txt", header = TRUE, sep = ""))

test = data.frame(read.table("svmdata3test.txt", header = TRUE, sep = ""))

names(data1)[1] <- "X1"

names(test)[1] <- "X1"

model1a = svm(Colors ~ ., data = data1, kernel = "polynomial", degree=10)

model1b = svm(Colors ~ ., data = data1, kernel = "polynomial", degree = 1000)

model2 = svm(Colors ~ ., data = data1, kernel = "radial")

model3 = svm(Colors ~ ., data = data1, kernel = "sigmoid")

print(data1)

#plot(x = model1b, data = test)

#plot(x = model2, data = test)

#plot(x = model3, data = test)

**Данные из текстовых файлов**

X1 X2 Colors

1 -0.335467047065564 0.102235638179093 red

2 0.671854180794949 0.787542430583787 green

3 1.91690032289261 -0.909648739038396 green

4 -0.582359432991493 0.632418469789369 red

5 -0.329231896660673 0.927994003174767 red

6 -0.984695346615275 -0.0510385551171024 red

7 -0.520880129353124 -0.513199165056229 red

8 -0.371719714248278 -0.155616389580535 red

9 -0.289211322827175 0.246015648044419 red

10 0.165635685835116 -0.642558763025519 red

**X1 X2 Colors**

1 -2.24246004235479 0.47048871459103 green

2 -0.305601118344231 0.194386931492083 red

3 -1.03924465132247 1.32822561738792 green

4 -0.635354620495016 -1.0256581475334 green

5 -1.07819131338742 -0.881511993751145 green

6 0.321438347662475 -0.404281649168687 red

7 1.49786129222265 1.25377726498095 green

8 0.403182073193313 1.37752261137620 green

9 0.802650744279983 -0.384938550369766 red

# Пример 4. Создание табличных фреймовых структур

# #вспомогательныефункции

# rndSeq = function(characterSet, len) {

# return(paste(sample(characterSet, len, TRUE), sep = "", collapse = ""))

# }

# getRndNames = function(characterSet, count, len) {

# tmp = c(rndSeq(characterSet, len))

# for (i in 2:count) {

# tmp = c(tmp, rndSeq(characterSet, len))

# }

# return(tmp)

# }

# getRndEmployYears = function(birthYears, lastYear, employAge = 18) {

# tmp = round(runif(1, birthYears[1] + employAge, lastYear))

# for (i in 2:length(birthYears)) {

# tmp = c(tmp, round(runif(1, birthYears[i] + employAge, lastYear)))

# }

# return(tmp)

# }

# getSalaryForYear = function(birthYear, employYear, minYear, year, baseSalary) {

# if (year == employYear) {

# return (baseSalary)

# }

# if (birthYear > minYear) {

# return((log(year - employYear) + 1) \* baseSalary)

# }

# return((log(year - employYear, 2) + 1) \* baseSalary)

# }

# getCurrentSalayForEachEmployee = function(birthYears, employYears, minYear, currentYear, baseSalary) {

# tmp = getSalaryForYear(birthYears[1], employYears[1], minYear, currentYear, baseSalary)

# for (i in 2:length(birthYears)) {

# tmp = c(tmp, getSalaryForYear(birthYears[i], employYears[i], minYear, currentYear, baseSalary))

# }

# return (tmp)

# }

# getEmployeesWithSalaryMoreThan = function(frame, salary) {

# result = character()

# for (i in 1:length(frame[["Nrow"]])) {

# if (frame[["Salary"]][i] > salary) {

# result = c(result, as.character(frame[["Name"]])[i])

# }

# }

# return (result)

# }

# getIncomeTaxForYear = function(birthYear, employYear, minYear, year, baseSalary, rate) {

# return(rate \* getSalaryForYear(birthYear, employYear, minYear, year, baseSalary))

# }

# getTotalIncomeTax = function(birthYear, employYear, minYear, currentYear, baseSalary, rate) {

# result = 0

# for (i in employYear:currentYear) {

# result = result + getIncomeTaxForYear(birthYear, employYear, minYear, i, baseSalary, rate)

# }

# return (result)

# }

# getTotalIncomeTaxForEachEmployee = function(birthYears, employYears, minYear, currentYear, baseSalary, rate) {

# result = double()

# for (i in 1:length(birthYears)) {

# result = c(result, getTotalIncomeTax(birthYears[i], employYears[i], minYear, currentYear, baseSalary, rate))

# }

# return (result)

# }

# #подготовка данных для фрейма

# N = 20

# Nrow = 1:N

# Name = getRndNames(LETTERS, N, 5)

# BirthYear = round(runif(N, 1960, 1985))

# EmployYear = getRndEmployYears(BirthYear, 2006)

# Salary = getCurrentSalayForEachEmployee(BirthYear, EmployYear, 1975, 2007, 8000)

# #созданиефрейма

# frame = data.frame(Nrow, Name, BirthYear, EmployYear, Salary)

# print(frame)

# #подсчет количества сотрудников с зарплатой, большей 15000

# employeesCount = length(getEmployeesWithSalaryMoreThan(frame, 15000))

# print(employeesCount)

# #добавлениеполя "подоходныйналог"

# TotalIncomeTax = getTotalIncomeTaxForEachEmployee(BirthYear, EmployYear, 1975, 2007, 8000, 0.13)

# frame = data.frame(Nrow, Name, BirthYear, EmployYear, Salary, TotalIncomeTax)

# print(frame)

# Пример. 5 Алгоритм сортировки вставками.

***Си***

int pass, j, hold;

for (pass = 0; pass < SIZE-1; pass++){

for (j = pass+1; j < SIZE; j++){

if (n[pass]>n[j]){

hold = n[j];

n[j] = n[pass];

n[pass] = hold;

}

}

}

typedef struct List {

struct List\* next;

struct List\* prev;

uint32\_t id;

} list\_t;

void list\_insert\_prev(list\_t\* head, list\_t\* tail) {

tail->prev = head->prev;

tail->next = head;

head->prev = tail;

if (tail->prev) tail->prev->next = tail;

}

list\_t\* list\_cut(list\_t\* head) {

list\_t\* cut = head;

if (cut->next) head->next->prev = head->prev;

if (cut->prev) head->prev->next = head->next;

return cut;

}

void list\_insertion\_sort(list\_t\* head) {

while (head->next) {

if (head->next->id < head->id) {

list\_t\* cut = list\_cut(head->next);

while ((head->prev) && (cut->id < head->prev->id)) {

head = head->prev;

}

list\_insert\_prev(head, cut);

} else {

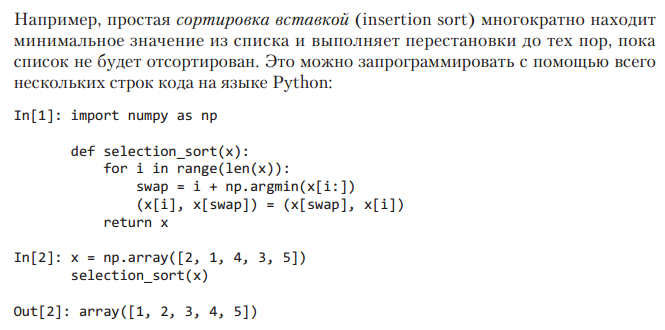
head = head->next;

}

}

}

# Python



# def insertion\_sort(a):

# for j in range(1,len(a)):

# key = a[j]

# i = j-1

# while (i > -1) and key < a[i]:

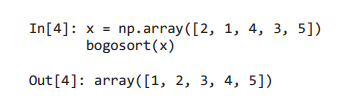
# a[i+1]=a[i]

# i=i-1

# a[i+1] = key

# return a

# 



# 

# Паскаль

# const N = 255;

# type TArray = array [1..N] of integer;

# procedure InsertSort(var x: TArray);

# var

# i, j, buf: integer;

# begin

# for i := 2 to N do

# begin

# buf := x[i];

# j := i - 1;

# while (j >= 1) and (x[j] > buf) do

# begin

# x[j + 1] := x[j];

# j := j - 1;

# end;

# x[j + 1] := buf;

# end;

# end;

**Пример 6. Метод градиентного спуска (язык Python)**

import numpy

import math

from pylab import \*

from sympy import \*

from scipy.optimize import minimize\_scalar

import matplotlib.pyplot as plt

from matplotlib import cm

from mpl\_toolkits.mplot3d import axes3d, Axes3D

z\_str = '3 \* a[0] \*\* 2 + a[1] \*\* 2 - a[0] \* a[1] - 4 \* a[0]'

exec 'z = lambda a: ' + z\_str

z\_str = z\_str.replace('a[0]', 'x')

z\_str = z\_str.replace('a[1]', 'y')

def z\_grad(a):

x = Symbol('x')

y = Symbol('y')

exec 'z\_d = ' + z\_str

yprime = z\_d.diff(y)

dif\_y=str(yprime).replace('y', str(a[1]))

dif\_y=dif\_y.replace('x', str(a[0]))

yprime = z\_d.diff(x)

dif\_x=str(yprime).replace('y', str(a[1]))

dif\_x=dif\_x.replace('x', str(a[0]))

return numpy.array([eval(dif\_y), eval(dif\_x)])

def mininize(a):

l\_min = minimize\_scalar(lambda l: z(a - l \* z\_grad(a))).x

return a - l\_min \* z\_grad(a)

def norm(a):

return math.sqrt(a[0] \*\* 2 + a[1] \*\* 2)

def grad\_step(dot):

return mininize(dot)

dot = [numpy.array([-150.0, 150.0])]

dot.append(grad\_step(dot[0]))

eps = 0.0001

while norm(dot[-2] - dot[-1]) > eps: dot.append(grad\_step(dot[-1]))

def makeData ():

x = numpy.arange (-200, 200, 1.0)

y = numpy.arange (-200, 200, 1.0)

xgrid, ygrid = numpy.meshgrid(x, y)

zgrid = z([xgrid, ygrid])

return xgrid, ygrid, zgrid

xt, yt, zt = makeData()

fig = plt.figure()

ax = plt.axes(projection='3d')

ax.plot\_surface(xt, yt, zt, cmap=cm.hot)

ax.plot([x[0] for x in dot], [x[1] for x in dot], [z(x) for x in dot], color='b')

plt.show()

**Таблица показателей анализа крови**

|  |  |  |  |
| --- | --- | --- | --- |
| № | Показатель | Диапазон значений считающихся нормальными | Примечание |
| 1 | Альбумин | 35-50 |  |
| 2 | Трансферрин | 2-4 |  |
| 3 | Ферритин | 20-250 |  |
| 4 | Билирубин | 8,6-20,5 |  |
| 5 | Глобулин | 40-60 |  |
| 6 | ГЕМОГЛОБИН | 120-160 |  |
| 7 | ЭРИТРОЦИТЫ | 3.7-5.1 |  |
| 8 | ЛЕЙКОЦИТЫ | 4-9 |  |
| 9 | лимфоциты, | 40.7 |  |
| 10 | нейтрофилы, , | 2-11 |  |
| 11 | Базофилы | 0.3 |  |
| 12 | плазмоциты | 3-11 |  |
| 13 | Эозинофилы | 2.4 |  |
| 14 | ТРОМБОЦИТЫ | 160-320 |  |
| 15 | ГЛЮКОЗА | 3.5-6.5 |  |
| 16 | ОБЩИЙ БЕЛОК | 60-80 |  |
| 17 | КРЕАТИНИН | 0.18 |  |
| 18 | Гематокрит | 36-48 |  |
| 19 | Ретикулоциты | 2-12 |  |
| 20 | Моноциты | 10 |  |

Формы заболевания болезни Паркинсона: Дрожательно-ригидная, Акинетико-ригидная, Акинетическая форма, Дрожательная форма, ригидная форма. Степень заболевания характеризуется этапами от 0 до 5 (бессимптомное течение, маловыраженные односторонние изменения. маловыраженные двусторонние изменения, двусторонние изменения с появлением первых заметных двигательных ограничений, ощутимые ограничения, пациент сам не справляется, полная потеря трудоспособности и самостоятельной жизнедеятельности).

# Примеры программного кода.

Персептрон Розенблатта.

#data(iris)

library(kohonen)

library(RSNNS)

library(class)

library(gmodels)

library(modeest)

library(nnet)

library(e1071)

ramFo = data.frame(read.table("cleverK.txt", header = TRUE, sep = ""))

#print("Исходные данные")

#print(ramFo)

alg <- matrix(0:0, nrow=150, ncol=4)

for (i in 1:50) {

for (j in 1:4) {

alg[i,j] = sample(5:50,1)

}

}

for (i in 51:100) {

for (j in 1:4) {

alg[i,j] = sample(51:95,1)

}

}

for (i in 101:150) {

for (j in 1:4) {

alg[i,j] = sample(1:4,1)

}

}

#alg

write.table(alg, file="GMB1.txt")

ramFo2 = data.frame(read.table("GMB1.txt", header = TRUE, sep = ""))

#C1 <- c("Выгодная")

ramFoT = data.frame(read.table("org.txt", header = TRUE, sep = ""))

ramFo2 <- cbind(ramFo2, ramFoT$V1)

#ramFo2

trainPerceptron <- function(dm, dc) {

result <- list()

oldW <- c()

w <- rep(0, ncol(dm))

while (!identical(w, oldW)) {

oldW <- w

for (i in 1:nrow(dm)) {

pred <- sign(dm[i,] %% w)

w = w+ (dc[i] - pred) \* dm[i, ]

}

result <- c(result, list(w))

}

result

}

plotPerceptronSteps <- function(pcResult, dm, dc, steps) {

for (s in steps) {

w <- pcResult[[s]]

plot(dm[,1:2], main=paste("step", s),

pch=ifelse(dc > 0, 1, 2), sub=paste(w, collapse=''))

abline(-w[3] / w[2], -w[1] / w[2])

}

}

dd <- ramFo2[ramFoT$V1 != "Мини-Скидка", -(3:4)]

dm <- data.matrix(dd[, 1:2])

dm <- cbind(dm, 1)

dc <- rep(1, nrow(dm))

dc[ramFoT$V1 == "Выгодная"] <- -1

pc <- trainPerceptron(dm, dc)

plotPerceptronSteps(pc, dm, dc, length(pc))

#par(mfrow=c(2,2))

#plotPerceptronSteps(pc, dm, dc, c(20, 50, 250,length(pc)))

Машина опорных векторов.

#data(iris)

library(kohonen)

library(RSNNS)

library(class)

library(gmodels)

library(modeest)

library(nnet)

library(e1071)

ramFo = data.frame(read.table("cleverK.txt", header = TRUE, sep = ""))

#print("Исходные данные")

#print(ramFo)

alg <- matrix(0:0, nrow=150, ncol=4)

for (i in 1:50) {

for (j in 1:4) {

alg[i,j] = sample(5:50,1)

}

}

for (i in 51:100) {

for (j in 1:4) {

alg[i,j] = sample(51:95,1)

}

}

for (i in 101:150) {

for (j in 1:4) {

alg[i,j] = sample(1:4,1)

}

}

#alg

write.table(alg, file="GMB1.txt")

ramFo2 = data.frame(read.table("GMB1.txt", header = TRUE, sep = ""))

#C1 <- c("Выгодная")

ramFoT = data.frame(read.table("org.txt", header = TRUE, sep = ""))

ramFo2 <- cbind(ramFo2, ramFoT$V1)

#ramFo2

## classification mode

# default with factor response:

model <- svm(ramFoT$V1 ~ ., data = ramFo2, type = "C-classification", kernel="linear")

# alternatively the traditional interface:

x <- subset(ramFo2)

y <- ramFoT$V1

model <- svm(x, y)

print(model)

summary(model)

# test with train data

pred <- predict(model, x)

# (same as:)

pred <- fitted(model)

# Check accuracy:

table(pred, y)

# compute decision values and probabilities:

pred <- predict(model, x, decision.values = TRUE)

attr(pred, "decision.values")[1:4,]

# visualize (classes by color, SV by crosses):

plot(cmdscale(dist(ramFo2[,-5])),

col = as.integer(ramFo2[,5]),

pch = c("o","+")[1:150 %in% model$index + 1])

# Заданияклабораторнойработе

**Вариант 1**

1. Создайте фрейм данных из *N* = 25 записей со следующими полями: *Nrow* – номер записи, *Name* – имя сотрудника, *BirthYear* – год рождения, *EmployYear* – год приема на работу, *Salary* – зарплата. *EyEColor*–цвет глаз, *SkinColor*–цвет кожи, *BloodType*–группа крови, *HairColor*–цвет волос на голове. Заполните данный фрейм данными так, что *Nrow* изменяется от 1 до *N*,

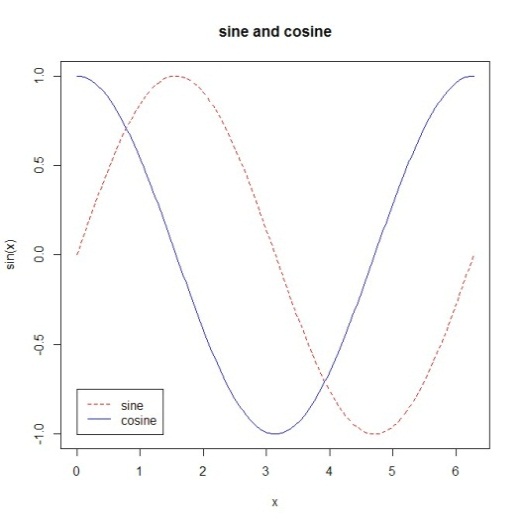


Рис. 1:

Name задается произвольно, *BithYear* распределен равномерно (случайно) на отрезке [1978*,*1995], *EmployYear* распределен равномерно на отрезке [*BirthY ear*+18*,*2015], *Salary* для работников младше 1990 г.р. определяется по формуле *Salary* = (*ln*(2016 − *EmployY ear*) +1) ∗ 9000, для остальных *Salary* = (*log*2(2016 − *EmployY ear*) + 1) ∗ 9000.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ***Nrow*** | ***Name*** | ***BirthYear*** | ***EmployYear*** | ***Salary*** | ***EyEColor*** | ***SkinColor*** | ***BloodType*** | ***HairColor*** | **Прим.** |
| 1 | Ярослав | 1978 | 2014 |  | 01 | 09 | 2 | 15 | Укр |
| 2 | Осип | 1979 | 2013 |  | 04 | 09 | 2 | 17 | Укр |
| 3 | Олесь | 1981 | 2015 |  | 06 | 09 | 2 | 16 | Укр |
| 4 | Микита | 1988 | 2007 |  | 01 | 09 | 2 | 17 | Укр |
| 5 | Данило | 1991 | 2011 |  | 04 | 09 | 2 | 16 | Укр |
| 6 | Рати | 1992 | 2010 |  | 01 | 14 | 1 | 01 | Инд |
| 7 | Амита | 1980 | 2015 |  | 01 | 14 | 1 | 15 | Инд |
| 8 | Аванти | 1984 | 2002 |  | 05 | 14 | 1 | 15 | Инд |
| 9 | Амала | 1985 | 2003 |  | 01 | 14 | 1 | 15 | Инд |
| 10 | Вазанта | 1986 | 2004 |  | 05 | 14 | 1 | 15 | Инд |
| 11 | Бхарат | 1987 | 2005 |  | 01 | 14 | 1 | 15 | Инд |
| 12 | Тришна | 1988 | 2008 |  | 05 | 14 | 1 | 15 | Инд |
| 13 | Йенс | 1992 | 2012 |  | 05 | 09 | 2 | 16 | Дат |
| 14 | Ларс | 1991 | 2014 |  | 04 | 09 | 2 | 16 | Дат |
| 15 | Бент | 1993 | 2015 |  | 04 | 09 | 2 | 16 | Дат |
| 16 | Хенрик | 1994 | 2014 |  | 04 | 09 | 2 | 16 | Дат |
| 17 | Браин | 1995 | 2015 |  | 05 | 09 | 2 | 16 | Дат |
| 18 | Джеспер | 1978 | 2008 |  | 04 | 09 | 2 | 16 | Дат |
| 19 | Вилфред | 1989 | 2007 |  | 05 | 09 | 2 | 16 | Дат |
| 20 | Лин | 1990 | 2010 |  | 01 | 14 | 3 | 15 | Бирм |
| 21 | Кхи | 1992 | 2012 |  | 06 | 09 | 3 | 15 | Бирм |
| 22 | Аун | 1991 | 2011 |  | 05 | 14 | 3 | 15 | Бирм |
| 23 | Чжи | 1993 | 2013 |  | 01 | 09 | 3 | 15 | Бирм |
| 24 | Док Ха | 1994 | 2014 |  | 05 | 14 | 3 | 15 | Бирм |
| 25 | Зе Бе | 1995 | 2015 |  | 06 | 14 | 3 | 15 | Бирм |

**Вспомогательная таблица цветов**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № | Цвет | Код цвета глаз | Код цвета кожи | Код цветаволос | Примечание |
| 1 | Черный | 01 | 08 | 15 |  |
| 2 | Белый | 02 | 09 | 16 |  |
| 3 | Рыжий | 03 | 10 | 17 |  |
| 4 | Голубой | 04 | 11 | 18 |  |
| 5 | Карий | 05 | 12 | 19 |  |
| 6 | Серый | 06 | 13 | 21 |  |
| 7 | Желтый | 07 | 14 | 22 |  |

В Excel применим полезные функции =СЛУЧМЕЖДУ(1978;1995), =СЛУЧМЕЖДУ(C2+18;2015), =ВПР(K2;$M$11:$O$25;3;0), =(LOG10(2016-D2)+1)\*9000, =(LOG(2016-D22;2)+1)\*9000

Подсчитайте число сотрудников с зарплатой, большей 17000 (отсортировать сотрудников по размеру зарплаты по возрастающей использованием известных стандартных средств сортировки). Добавьте в таблицу поле, соответствующее суммарному подоходному налогу (ставка 13%), выплаченному сотрудником за время работы в организации, если его зарплата за каждый год начислялась согласно формулам для *Salary*, где вместо 2016 следует последовательно подставить каждый год работы сотрудника в организации.

import numpy as np

import pandas as pd

sals = pd.read\_csv('clever.csv')

sals

df = pd.DataFrame(rng.rand(1000, 3), columns=['A', 'B', 'C'])

df.head()

df.eval('D = (A + B) / C', inplace=True)

df.head()

1. Постройте линейный классификатор для классификации сотрудников данной международной организации (признаки классификации: группа крови, цвет волос, глаз и цвет кожи). Использовать машину опорных векторов и алгоритм персептрона. Полученные результаты сравнить.

Таблица соответствия групп крови национальностям

|  |  |  |  |
| --- | --- | --- | --- |
| **№** | **Национальность** | **Преобладающая группа крови** | **Примечание** |
| 1 | Исландцы | 1 |  |
| 2 | Англичане | 1 |  |
| 3 | Итальянцы | 1 |  |
| 4 | Датчане | 2 |  |
| 5 | Немцы | 2 |  |
| 6 | Русские | 2 |  |
| 7 | Испанцы | 2 |  |
| 8 | Китайцы | 1 |  |
| 9 | Индусы | 1 |  |
| 10 | Бирманцы | 3 |  |
| 11 | Японцы | 2 |  |
| 12 | Татары | 2 |  |
| 13 | Китайцы Пекина | 3 |  |
| 14 | Украинцы | 2 |  |
| 15 | Португальцы | 2 |  |

Цвет глаз, кожи или волос можно закодировать определенным числом. В данном варианте использовать национальности: **Украинец, Индус, Датчанин, Бирманец**.

Таким образом, в задании 4 группы (или класса) для классификации:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **№** | **Группы** | **Группа крови** | **Цвет глаз** | **Цвет кожи** | **Цвет волос** |
| 1 | Украинец | 2 | Голубой, черный, карий | белый | Черный, белый, рыжий |
| 2 | Индус | 1 | Черный, карий, серый | желтый | Черный, серый |
| 3 | Датчанин | 2 | Голубой, карий, серый | белый | Белый, черный, серый |
| 4 | Бирманец | 3 | Черный, карий | Черный и желтый | Черный |

Для машины опорных векторов типа "C-classification" с линейным ядром, добейтесь нулевой ошибки сначала на обучающей выборке, а затем на тестовой, путем изменения параметра *C*.

1. Допустим, что решающая функция линейного классификатора в упрощенном виде выглядит так:



Найти координаты и значение функции в точке минимума методом градиентного спуска.

**Вариант 2**

1. Создайте фрейм данных из *N* = 30 записей со следующими полями: *Nrow* – номер записи, *Name* – имя сотрудника, *BirthYear* – год рождения, *EmployYear* – год приема на работу, *Salary* – зарплата. EyEColor–цвет глаз, SkinColor–цвет кожи, BloodType–группа крови, HairColor–цвет волос на голове. Заполните данный фрейм данными так, что *Nrow* изменяется от 1 до *N*,

Name задается произвольно, *BithYear* распределен равномерно (случайно) на отрезке [1974*,*1993], *EmployYear* распределен равномерно на отрезке [*BirthY ear*+17*,*2014], *Salary* для работников младше 1989 г.р. определяется по формуле *Salary* = (*ln*(2015 − *EmployY ear*) +1) ∗ 7000, для остальных *Salary* = (*log*2(2015 − *EmployY ear*) + 1) ∗ 7000.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ***Nrow*** | ***Name*** | ***BirthYear*** | ***EmployYear*** | ***Salary*** | ***EyEColor*** | ***SkinColor*** | ***BloodType*** | ***HairColor*** | **Прим.** |
| 1 | Стивен | 1974 | 2013 |  | 01 | 09 | 1 | 15 | Анг |
| 2 | Чарльз | 1978 | 2012 |  | 04 | 09 | 1 | 17 | Анг |
| 3 | Джон | 1980 | 2014 |  | 06 | 09 | 1 | 16 | Анг |
| 4 | Пол | 1987 | 2006 |  | 01 | 09 | 1 | 17 | Анг |
| 5 | Уэйн | 1990 | 2010 |  | 04 | 09 | 1 | 16 | Анг |
| 6 | Дональд | 1991 | 2009 |  | 01 | 09 | 1 | 01 | Анг |
| 7 | Бенджамин | 1979 | 2014 |  | 01 | 09 | 1 | 15 | Анг |
| 8 | Майкл | 1983 | 2001 |  | 05 | 09 | 1 | 15 | Анг |
| 9 | Николас | 1984 | 2002 |  | 01 | 09 | 1 | 15 | Анг |
| 10 | Джеральд | 1985 | 2003 |  | 05 | 09 | 1 | 15 | Анг |
| 11 | Чжен | 1986 | 2004 |  | 01 | 14 | 3 | 15 | Кит |
| 12 | Шеньчао | 1987 | 2007 |  | 05 | 14 | 3 | 15 | Кит |
| 13 | Ханьвэнь | 1991 | 2011 |  | 05 | 14 | 3 | 16 | Кит |
| 14 | Ли Сюэпин | 1990 | 2013 |  | 04 | 14 | 3 | 16 | Кит |
| 15 | Лю Имин | 1992 | 2014 |  | 01 | 09 | 3 | 16 | Кит |
| 16 | Фань Сяодун | 1993 | 2013 |  | 04 | 14 | 3 | 16 | Кит |
| 17 | Хэ Гуань | 1994 | 2014 |  | 05 | 09 | 3 | 16 | Кит |
| 18 | Цзян Чжи | 1977 | 2007 |  | 04 | 14 | 3 | 16 | Кит |
| 19 | Линьпэн | 1988 | 2006 |  | 05 | 09 | 3 | 16 | Кит |
| 20 | Томас | 1989 | 2009 |  | 01 | 09 | 2 | 15 | Нем |
| 21 | Франц | 1991 | 2011 |  | 06 | 09 | 2 | 15 | Нем |
| 22 | Марио | 1990 | 2010 |  | 05 | 09 | 2 | 16 | Нем |
| 23 | Гюнтер | 1992 | 2012 |  | 01 | 09 | 2 | 15 | Нем |
| 24 | Карл | 1994 | 2013 |  | 05 | 09 | 2 | 16 | Нем |
| 25 | Пьер | 1979 | 2007 |  | 06 | 09 | 2 | 17 | Нем |
| 26 | Оливер | 1992 | 2014 |  | 06 | 09 | 2 | 15 | Нем |
| 27 | Гердт | 1981 | 2001 |  | 01 | 09 | 2 | 16 | Нем |
| 28 | Филипп | 1982 | 2002 |  | 04 | 09 | 2 | 17 | Нем |
| 29 | Мануэль | 1974 | 2001 |  | 06 | 09 | 2 | 15 | Нем |
| 30 | Генри | 1990 | 2000 |  | 04 | 09 | 2 | 16 | Нем |

**Вспомогательная таблица цветов**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № | Цвет | Код цвета глаз | Код цвета кожи | Код цвета волос | Примечание |
| 1 | Черный | 01 | 08 | 15 |  |
| 2 | Белый | 02 | 09 | 16 |  |
| 3 | Рыжий | 03 | 10 | 17 |  |
| 4 | Голубой | 04 | 11 | 18 |  |
| 5 | Карий | 05 | 12 | 19 |  |
| 6 | Серый | 06 | 13 | 21 |  |
| 7 | Желтый | 07 | 14 | 22 |  |

Подсчитайте число сотрудников с зарплатой, большей 16000(отсортировать сотрудников по размеру зарплаты по возрастающей использованием известных стандартных средств сортировки). Добавьте в таблицу поле, соответствующее суммарному подоходному налогу (ставка 13%), выплаченному сотрудником за время работы в организации, если его зарплата за каждый год начислялась согласно формулам для *Salary*, где вместо 2015 следует последовательно подставить каждый год работы сотрудника в организации.

1. Постройте линейный классификатор для классификации сотрудников данной международной организации (признаки классификации: группа крови, цвет волос, глаз и цвет кожи). Использовать машину опорных векторов и алгоритм персептрона. Полученные результаты сравнить.

Цвет глаз, кожи или волос можно закодировать определенным числом. В данном варианте использовать национальности: **Англичанин, Китаец Пекина, Немец**.

Таким образом, в задании 3 группы (или класса) для классификации:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **№** | **Группы** | **Группа крови** | **Цвет глаз** | **Цвет кожи** | **Цвет волос** |
| 1 | Англичанин | 1 | Голубой, черный, карий | белый | Черный, белый, рыжий |
| 2 | Китаец Пекина | 1 | Черный, карий, серый | Желтый, белый | Черный, серый |
| 3 | Немец | 2 | Голубой, карий, серый | белый | Белый, черный, серый, рыжий |

Для машины опорных векторов типа "C-classification" с сигмоидальным ядром, добейтесь нулевой ошибки сначала на обучающей выборке, а затем на тестовой, путем изменения параметра *C*.

1. Допустим, что решающая функция линейного классификатора в упрощенном виде выглядит так:



Найти координаты и значение функции в точке минимума методом наискорейшего градиентного спуска.

**Вариант 3**

1. Создайте фрейм данных из *N* = 27 записей со следующими полями: *Nrow* – номер записи, *Name* – имя сотрудника, *BirthYear* – год рождения, *EmployYear* – год приема на работу, *Salary* – зарплата. EyEColor–цвет глаз, SkinColor–цвет кожи, BloodType–группа крови, HairColor–цвет волос на голове. Заполните данный фрейм данными так, что *Nrow* изменяется от 1 до *N*,

Name задается произвольно, *BithYear* распределен равномерно (случайно) на отрезке [1972*,*1994], *EmployYear* распределен равномерно на отрезке [*BirthY ear*+19*,*2013], *Salary* для работников младше 1980 г.р. определяется по формуле *Salary* = (*ln*(2014 − *EmployY ear*) +1) ∗ 8000, для остальных *Salary* = (*log*2(2014 − *EmployY ear*) + 1) ∗ 8000.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ***Nrow*** | ***Name*** | ***BirthYear*** | ***EmployYear*** | ***Salary*** | ***EyEColor*** | ***SkinColor*** | ***BloodType*** | ***HairColor*** | **Прим.** |
| 1 | Степан | 1972 | 2011 |  | 01 | 09 | 2 | 15 | Рус |
| 2 | Федор | 1976 | 2010 |  | 04 | 09 | 2 | 17 | Рус |
| 3 | Иван | 1978 | 2012 |  | 06 | 09 | 2 | 16 | Рус |
| 4 | Максим | 1985 | 2004 |  | 05 | 09 | 2 | 17 | Рус |
| 5 | Евгений | 1988 | 2008 |  | 04 | 09 | 2 | 16 | Рус |
| 6 | Курбан | 1991 | 2009 |  | 01 | 14 | 2 | 15 | Тат |
| 7 | Малик | 1977 | 2012 |  | 01 | 14 | 2 | 15 | Тат |
| 8 | Эльмир | 1983 | 2001 |  | 06 | 14 | 2 | 15 | Тат |
| 9 | Далер | 1984 | 2002 |  | 05 | 14 | 2 | 15 | Тат |
| 10 | Ильзат | 1985 | 2003 |  | 01 | 14 | 2 | 15 | Тат |
| 11 | Мигель | 1986 | 2004 |  | 01 | 14 | 2 | 15 | Пор |
| 12 | Густаво | 1987 | 2007 |  | 05 | 09 | 2 | 15 | Пор |
| 13 | Энцо | 1989 | 2009 |  | 05 | 14 | 2 | 16 | Пор |
| 14 | Педро | 1988 | 2011 |  | 04 | 14 | 2 | 16 | Пор |
| 15 | Мануэлла | 1992 | 2014 |  | 01 | 09 | 2 | 16 | Пор |
| 16 | Азуми | 1993 | 2013 |  | 01 | 14 | 2 | 16 | Япон |
| 17 | Акира | 1994 | 2014 |  | 05 | 14 | 2 | 16 | Япон |
| 18 | Киоко | 1977 | 2007 |  | 01 | 09 | 2 | 16 | Япон |
| 19 | Кэори | 1988 | 2006 |  | 05 | 14 | 2 | 16 | Япон |
| 20 | Мидори | 1989 | 2009 |  | 01 | 09 | 2 | 15 | Япон |
| 21 | Иошико | 1991 | 2011 |  | 06 | 14 | 2 | 15 | Япон |
| 22 | Марко | 1990 | 2010 |  | 04 | 09 | 1 | 16 | Итал |
| 23 | Лука | 1992 | 2012 |  | 01 | 09 | 1 | 15 | Итал |
| 24 | Луиджи | 1994 | 2013 |  | 05 | 09 | 1 | 16 | Итал |
| 25 | Фабио | 1979 | 2007 |  | 06 | 09 | 1 | 17 | Итал |
| 26 | Паоло | 1992 | 2014 |  | 04 | 09 | 1 | 15 | Итал |
| 27 | Роберто | 1994 | 2001 |  | 01 | 09 | 1 | 16 | Итал |

**Вспомогательная таблица цветов**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № | Цвет | Код цвета глаз | Код цвета кожи | Код цвета волос | Примечание |
| 1 | Черный | 01 | 08 | 15 |  |
| 2 | Белый | 02 | 09 | 16 |  |
| 3 | Рыжий | 03 | 10 | 17 |  |
| 4 | Голубой | 04 | 11 | 18 |  |
| 5 | Карий | 05 | 12 | 19 |  |
| 6 | Серый | 06 | 13 | 21 |  |
| 7 | Желтый | 07 | 14 | 22 |  |

Подсчитайте число сотрудников с зарплатой, большей 14000(отсортировать сотрудников по размеру зарплаты по возрастающей использованием известных стандартных средств сортировки). Добавьте в таблицу поле, соответствующее суммарному подоходному налогу (ставка 13%), выплаченному сотрудником за время работы в организации, если его зарплата за каждый год начислялась согласно формулам для *Salary*, где вместо 2014 следует последовательно подставить каждый год работы сотрудника в организации.

1. Постройте линейный классификатор для классификации сотрудников данной международной организации (признаки классификации: группа крови, цвет волос, глаз и цвет кожи). Использовать машину опорных векторов и алгоритм персептрона. Полученные результаты сравнить.

Цвет глаз, кожи или волос можно закодировать определенным числом. В данном варианте использовать национальности: **Русский, Татарин, Португалец, Японец, Итальянец**.

Таким образом, в задании 5 групп (или класса) для классификации:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **№** | **Группы** | **Группа крови** | **Цвет глаз** | **Цвет кожи** | **Цвет волос** |
| 1 | Русский | 2 | Голубой, черный, карий | белый | Черный, белый, рыжий |
| 2 | Татарин | 2 | Черный, карий, серый | желтый | Черный, с |
| 3 | Португалец | 2 | Голубой, карий, серый | Белый, желтый | черный, серый |
| 4 | Японец | 2 | Черный, карий | Желтый, белый | Черный |
| 5 | Итальянец | 1 | Голубой, черный | белый | Черный, белый |

Для машины опорных векторов типа "C-classification" с радиальным ядром, добейтесь нулевой ошибки сначала на обучающей выборке, а затем на тестовой, путем изменения параметра *C*.

1. Допустим, что решающая функция линейного классификатора в упрощенном виде выглядит так:



Найти координаты и значение функции в точке минимума методом градиентного спуска.

**Вариант 4**

1. Создайте фрейм данных из *N* = 24 записей со следующими полями: *Nrow* – номер записи, *Name* – имя пациента, *BirthYear* – год рождения, *Employ* – место работы, *Salary* – зарплата. *Cost*–стоимость лечения, *Albumin*–содержание альбумина в крови, *TransFerrin*–содержание трансферрина в крови, *Ferritin*–содержание ферритина в крови. Заполните данный фрейм данными так, что *Nrow* изменяется от 1 до *N*,

Name задается произвольно, *BithYear* распределен равномерно (случайно) на отрезке [1971*,*1997], , Costдля пациентов младше 1991 г.р. определяется по формуле Cost= (*ln*(2013 − Birth*Year*) +1) ∗11000, для остальных Cost= (*log*2(2013 − Birth*Y ear*) + 1) ∗10000.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ***Nrow*** | ***Name*** | ***BirthYear*** | ***Employ*** | ***Salary*** | ***Cost*** | ***Albumin*** | ***TransFerrin*** | ***Ferritin*** | **Прим.** |
| 1 | Степан | 1971 | АО АЛМАЗ | 33000 |  | 0.233 | 0.957 | 6.528 |  |
| 2 | Федор | 1975 | НП ПРОГРЕСС | 28000 |  | 0.245 | 0.147 | 6.53 |  |
| 3 | Иван | 1978 | АО ВОСТОК | 41000 |  | 0.238 | 0.145 | 5.21 |  |
| 4 | Максим | 1984 | ОА СЕВЕР | 53000 |  | 0.32 | 0.21 | 4.79 |  |
| 5 | Евгений | 1989 | НФ ЛУЧ | 62000 |  | 0.459 | 0.27 | 1.411 |  |
| 6 | Курбан | 1990 | АО АЛМАЗ | 18000 |  | 0.741 | 0.318 | 1.47 |  |
| 7 | Малик | 1976 | НП ПРОГРЕСС | 23000 |  | 0.886 | 0.407 | 2.089 |  |
| 8 | Эльмир | 1983 | АО ВОСТОК | 29000 |  | 0.614 | 0.581 | 3.143 |  |
| 9 | Далер | 1984 | ОА СЕВЕР | 31000 |  | 0.548 | 0.645 | 4.35 |  |
| 10 | Ильзат | 1985 | НФ ЛУЧ | 19500 |  | 0.231 | 0.714 | 3.076 |  |
| 11 | Михаил | 1986 | АО АЛМАЗ | 42000 |  | 0.89 | 0.803 | 3.48 |  |
| 12 | Павел | 1987 | НП ПРОГРЕСС | 61000 |  | 0.437 | 0.91 | 3.591 |  |
| 13 | Роберт | 1989 | АО ВОСТОК | 28000 |  | 0.362 | 0.923 | 2.57 |  |
| 14 | Петр | 1988 | ОА СЕВЕР | 26490 |  | 0.258 | 0.947 | 6.24 |  |
| 15 | Фарид | 1992 | НФ ЛУЧ | 19800 |  | 0.339 | 0.891 | 1.78 |  |
| 16 | Гумар | 1993 | АО НЕВА | 34300 |  | 0.515 | 0.87 | 1.86 |  |
| 17 | Анна | 1995 | АО ОКА | 37340 |  | 0.657 | 0.862 | 1.44 |  |
| 18 | Ольга | 1977 | НП ВЕГА | 53000 |  | 0.76 | 0.673 | 2.33 |  |
| 19 | Ксения | 1988 | НП КОЛОС | 59000 |  | 0.83 | 0.651 | 2.709 |  |
| 20 | Оксана | 1989 | АО НЕВА | 64000 |  | 0.841 | 0.704 | 2.21 |  |
| 21 | Марина | 1991 | АО ОКА | 60200 |  | 0.738 | 0.96 | 5.81 |  |
| 22 | Леонид | 1990 | НП ВЕГА | 18900 |  | 0.746 | 0.495 | 5.08 |  |
| 23 | Дамир | 1996 | НП КОЛОС | 31786 |  | 0.238 | 0.838 | 4.743 |  |
| 24 | Антон | 1997 | ФК МАРС | 50221 |  | 0.883 | 0.163 | 4.09 |  |

Ранжируйте пациентов по стоимости лечения, начиная с минимальной суммы(отсортировать пациентов с использованием известных стандартных средств сортировки). Добавьте в таблицу поле, соответствующее общему социальному вычету за лечение (ставка 13%), выплаченному пациенту, если стоимость лечения за каждый год начислялась согласно формулам для Cost, где вместо 2013 следует последовательно подставить каждый год нахождения пациента под наблюдением.

1. Постройте линейный классификатор для классификации пациентов по степени развития болезни Паркинсона (*3 степени - бессимптомное течение, маловыраженные односторонние изменения, маловыраженные двусторонние изменения*) на основе трех признаков – содержания в крови **альбумина, трансферрина, ферритина**.

Таким образом, в задании 3 группы (или класса) для классификации:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **№** | **Группы** | **Альбумин** | **Трасферрин** | **Ферритин** | **Примечание** |
| 1 | *бессимптомное течение* | 0.23-0.44 | 0.14-0.32 | 1.41-2.08 |  |
| 2 | *маловыраженные односторонние изменения* | 0.45-0.67 | 0.33-0.72 | 2.09-4.43 |  |
| 3 | *маловыраженные двусторонние изменения* | 0.68-0.89 | 0.73-0.96 | 4.44-6.53 |  |

Использовать *машину опорных векторов и алгоритм персептрона*. Полученные результаты сравнить с использованием функций языка R - Predict и Table.

Для машины опорных векторов типа "C-classification" с линейным ядром, добейтесь нулевой ошибки сначала на обучающей выборке, а затем на тестовой, путем изменения параметра *C*.

1. Допустим, что решающая функция линейного классификатора в упрощенном виде выглядит так:



Найти координаты и значение функции в точке минимума методом градиентного спуска.

**Вариант 5**

1. Создайте фрейм данных из *N* = 30 записей со следующими полями: *Nrow* – номер записи, *Name* – имя пациента, *BirthYear* – год рождения, *Employ* – место работы, *Salary* – зарплата. Cost–стоимость лечения, *Monocit*–содержание моноцитов в крови, *Retikullocit*–содержание ретилкуллоцитов в крови, *Gematokrit*–содержание гематокрита в крови, *Creatin* – содержание креатина в крови. Заполните данный фрейм данными так, что *Nrow* изменяется от 1 до *N*,

Name задается произвольно, *BithYear* распределен равномерно (случайно) на отрезке [1975*,*1995], , Costдля пациентов младше 1988 г.р. определяется по формуле Cost= (*ln*(2012 − Birth*Y ear*) +1) ∗ 6000, для остальных Cost= (*log*2(2012 − Birth*Y ear*) + 1) ∗ 7000.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ***Nrow*** | ***Name*** | ***BirthYear*** | ***Employ*** | ***Salary*** | ***Cost*** | ***Monocit*** | ***Retikullocit*** | ***Gematokrit*** | ***Creatin*** |
| 1 | Степан | 1976 | АО АЛМАЗ | 33000 |  | 0.37 | 0.78 | 2.33 | 1.47 |
| 2 | Федор | 1975 | НП ПРОГРЕСС | 28000 |  | 0.56 | 0.77 | 2.88 | 3.34 |
| 3 | Иван | 1978 | АО ВОСТОК | 41000 |  | 0.81 | 0.778 | 2.92 | 3.36 |
| 4 | Максим | 1984 | ОА СЕВЕР | 53000 |  | 0.95 | 1.46 | 8.33 | 9.21 |
| 5 | Евгений | 1989 | НФ ЛУЧ | 62000 |  | 1.31 | 1.41 | 7.64 | 9.203 |
| 6 | Курбан | 1990 | АО АЛМАЗ | 18000 |  | 0.43 | 1.47 | 6.39 | 8.05 |
| 7 | Малик | 1976 | НП ПРОГРЕСС | 23000 |  | 0.42 | 1.74 | 3.11 | 5.62 |
| 8 | Эльмир | 1983 | АО ВОСТОК | 29000 |  | 0.51 | 2.23 | 4.12 | 5.47 |
| 9 | Далер | 1984 | ОА СЕВЕР | 31000 |  | 0.62 | 2.92 | 5.18 | 3.54 |
| 10 | Ильзат | 1985 | НФ ЛУЧ | 19500 |  | 0.63 | 3.56 | 2.77 | 2.58 |
| 11 | Михаил | 1986 | АО АЛМАЗ | 42000 |  | 0.64 | 4.21 | 6.1 | 3.09 |
| 12 | Павел | 1987 | НП ПРОГРЕСС | 61000 |  | 0.49 | 4.097 | 6.23 | 4.108 |
| 13 | Роберт | 1989 | АО ВОСТОК | 28000 |  | 0.498 | 5.78 | 7.34 | 7.17 |
| 14 | Петр | 1988 | ОА СЕВЕР | 26490 |  | 1.112 | 5.32 | 6.7 | 6.22 |
| 15 | Фарид | 1992 | НФ ЛУЧ | 19800 |  | 1.212 | 5.31 | 6.08 | 6.189 |
| 16 | Гумар | 1993 | АО НЕВА | 34300 |  | 1.132 | 4.56 | 4.08 | 9.17 |
| 17 | Анна | 1995 | АО ОКА | 37340 |  | 1.304 | 2.09 | 2.409 | 8.45 |
| 18 | Ольга | 1977 | НП ВЕГА | 53000 |  | 0.907 | 1.98 | 2.363 | 7.62 |
| 19 | Ксения | 1988 | НП КОЛОС | 59000 |  | 0.91 | 1.913 | 7.86 | 6.74 |
| 20 | Оксана | 1989 | АО НЕВА | 64000 |  | 0.88 | 1.55 | 4.58 | 4.63 |
| 21 | Марина | 1991 | АО ОКА | 60200 |  | 0.79 | 3.45 | 6.32 | 9.03 |
| 22 | Леонид | 1990 | НП ВЕГА | 18900 |  | 0.76 | 5.8 | 7.31 | 7.224 |
| 23 | Дамир | 1994 | НП КОЛОС | 31786 |  | 0.681 | 4.4 | 3.42 | 1.63 |
| 24 | Антон | 1995 | ФК МАРС | 50221 |  | 0.973 | 4.3 | 5.154 | 1.863 |
| 25 | Федор | 1975 | АО ВОСТОК | 38022 |  | 0.397 | 1.5 | 8.17 | 3.331 |
| 26 | Матвей | 1977 | ОА СЕВЕР | 26871 |  | 1.045 | 0.79 | 5.446 | 1.481 |
| 27 | Андрей | 1989 | НФ ЛУЧ | 43000 |  | 1.064 | 0.71 | 7.341 | 3.43 |
| 28 | Полина | 1986 | АО НЕВА | 47900 |  | 0.52 | 0.88 |  |  |
| 29 | Семен | 1985 | АО ОКА | 59800 |  | 0.605 | 1.47 | 8.072 | 1.78 |
| 30 | Гурам | 1991 | АО ТОПАЗ | 63000 |  | 0.752 | 2.124 | 8.26 | 3.37 |

Ранжируйте пациентов по стоимости лечения, начиная с максимальной суммы (с использованием известных алгоритмов сортировки). Добавьте в таблицу поле, соответствующее общему социальному вычету за лечение (ставка 13%), выплаченному пациенту, если стоимость лечения за каждый год начислялась согласно формулам для Cost, где вместо 2012 следует последовательно подставить каждый год нахождения пациента под наблюдением.

1. Постройте линейный классификатор для классификации пациентов по степени развития болезни Паркинсона (*4 степени - двусторонние изменения с появлением первых заметных двигательных ограничений, маловыраженные односторонние изменения, маловыраженные двусторонние изменения, полная потеря трудоспособности и самостоятельной жизнедеятельности*) на основе четырех признаков – содержания в крови **креатина, моноцитов, гематокрита, ретилкуллоцитов**.

Таким образом, в задании 4 группы (или класса) для классификации:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **№** | **Группы** | **Креатин** | **Моноциты** | **Гематокрит** | **Ретилкуллоциты** |
| 1 | *двусторонние изменения с появлением первых заметных двигательных ограничений* | 1.47-3.34 | 0.34-0.51 | 2.33-2.89 | 0.77-1.5 |
| 2 | *маловыраженные односторонние изменения* | 3.35-5.61 | 0.52-0.73 | 2.9-3.41 | 1.6-2.1 |
| 3 | *маловыраженные двусторонние изменения* | 5.62-7.16 | 0.74-0.93 | 3.42-5.1 | 2.2-4.3 |
| 4 | *полная потеря трудоспособности и самостоятельной жизнедеятельности* | 7.17-9.21 | 0.94-1.32 | 5.2-8.4 | 4.4-5.8 |

Использовать машину опорных векторов и алгоритм персептрона. Полученные результаты сравнить с использованием функций языка R - Predict и Table.

Для машины опорных векторов типа "C-classification" с полиномиальным ядром, добейтесь нулевой ошибки сначала на обучающей выборке, а затем на тестовой, путем изменения параметра *C*.

1. Допустим, что решающая функция линейного классификатора в упрощенном виде выглядит так:



Найти координаты и значение функции в точке минимума методом градиентного спуска.

**Вариант 6**

1. Создайте фрейм данных из *N* = 28 записей со следующими полями: *Nrow* – номер записи, *Name* – имя пациента, *BirthYear* – год рождения, *Employ* – место работы, *Salary* – зарплата. Cost–стоимость лечения, Glukosa–содержание глюкозы в крови, Trombocit–содержание тромбоцитов в крови, Belok–содержание белков в крови, Eozinofil – содержание эозинофила в крови, Plazmocit - содержание плазмоцитов в крови. Заполните данный фрейм данными так, что *Nrow* изменяется от 1 до *N*,

Name задается произвольно, *BithYear* распределен равномерно (случайно) на отрезке [1974*,*1994], , Costдля пациентов младше 1988 г.р. определяется по формуле Cost= (*ln*(2011 − Birth*Y ear*) +1) ∗ 9000, для остальных Cost= (*log*2(2011 − Birth*Y ear*) + 1) ∗ 8000.

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ***Nrow*** | ***Name*** | ***BirthYear*** | ***Employ*** | ***Salary*** | ***Cost*** | **Glukosa** | **Trombocit** | **Belok** | **Eozinofil** | **Plazmocit** |
| 1 | Степан | 1974 | АО АЛМАЗ | 33000 |  | 0.37 | 0.78 | 2.33 | 1.47 |  |
| 2 | Федор | 1994 | НП ПРОГРЕСС | 28000 |  | 0.56 | 0.77 | 2.88 | 3.34 |  |
| 3 | Иван | 1978 | АО ВОСТОК | 41000 |  | 0.81 | 0.778 | 2.92 | 3.36 |  |
| 4 | Максим | 1984 | ОА СЕВЕР | 53000 |  | 0.95 | 1.46 | 8.33 | 9.21 |  |
| 5 | Евгений | 1989 | НФ ЛУЧ | 62000 |  | 1.31 | 1.41 | 7.64 | 9.203 |  |
| 6 | Курбан | 1990 | АО АЛМАЗ | 18000 |  | 0.43 | 1.47 | 6.39 | 8.05 |  |
| 7 | Малик | 1976 | НП ПРОГРЕСС | 23000 |  | 0.42 | 1.74 | 3.11 | 5.62 |  |
| 8 | Эльмир | 1983 | АО ВОСТОК | 29000 |  | 0.51 | 2.23 | 4.12 | 5.47 |  |
| 9 | Далер | 1984 | ОА СЕВЕР | 31000 |  | 0.62 | 2.92 | 5.18 | 3.54 |  |
| 10 | Ильзат | 1985 | НФ ЛУЧ | 19500 |  | 0.63 | 3.56 | 2.77 | 2.58 |  |
| 11 | Михаил | 1986 | АО АЛМАЗ | 42000 |  | 0.64 | 4.21 | 6.1 | 3.09 |  |
| 12 | Павел | 1987 | НП ПРОГРЕСС | 61000 |  | 0.49 | 4.097 | 6.23 | 4.108 |  |
| 13 | Роберт | 1989 | АО ВОСТОК | 28000 |  | 0.498 | 5.78 | 7.34 | 7.17 |  |
| 14 | Петр | 1988 | ОА СЕВЕР | 26490 |  | 1.112 | 5.32 | 6.7 | 6.22 |  |
| 15 | Фарид | 1992 | НФ ЛУЧ | 19800 |  | 1.212 | 5.31 | 6.08 | 6.189 |  |
| 16 | Гумар | 1993 | АО НЕВА | 34300 |  | 1.132 | 4.56 | 4.08 | 9.17 |  |
| 17 | Анна | 1995 | АО ОКА | 37340 |  | 1.304 | 2.09 | 2.409 | 8.45 |  |
| 18 | Ольга | 1977 | НП ВЕГА | 53000 |  | 0.907 | 1.98 | 2.363 | 7.62 |  |
| 19 | Ксения | 1988 | НП КОЛОС | 59000 |  | 0.91 | 1.913 | 7.86 | 6.74 |  |
| 20 | Оксана | 1989 | АО НЕВА | 64000 |  | 0.88 | 1.55 | 4.58 | 4.63 |  |
| 21 | Марина | 1991 | АО ОКА | 60200 |  | 0.79 | 3.45 | 6.32 | 9.03 |  |
| 22 | Леонид | 1990 | НП ВЕГА | 18900 |  | 0.76 | 5.8 | 7.31 | 7.224 |  |
| 23 | Дамир | 1994 | НП КОЛОС | 31786 |  | 0.681 | 4.4 | 3.42 | 1.63 |  |
| 24 | Антон | 1995 | ФК МАРС | 50221 |  | 0.973 | 4.3 | 5.154 | 1.863 |  |
| 25 | Федор | 1975 | АО ВОСТОК | 38022 |  | 0.397 | 1.5 | 8.17 | 3.331 |  |
| 26 | Матвей | 1977 | ОА СЕВЕР | 26871 |  | 1.045 | 0.79 | 5.446 | 1.481 |  |
| 27 | Андрей | 1989 | НФ ЛУЧ | 43000 |  | 1.064 | 0.71 | 7.341 | 3.43 |  |
| 28 | Полина | 1986 | АО НЕВА | 47900 |  | 0.52 | 0.88 |  |  |  |
| 29 | Семен | 1985 | АО ОКА | 59800 |  | 0.605 | 1.47 | 8.072 | 1.78 |  |
| 30 | Гурам | 1991 | АО ТОПАЗ | 63000 |  | 0.752 | 2.124 | 8.26 | 3.37 |  |

Ранжируйте пациентов по стоимости лечения, начиная с максимальной суммы. Добавьте в таблицу поле, соответствующее общему социальному вычету за лечение (ставка 13%), выплаченному пациенту, если стоимость лечения за каждый год начислялась согласно формулам для Cost, где вместо 2011 следует последовательно подставить каждый год нахождения пациента под наблюдением.

1. Постройте линейный классификатор для классификации пациентов по формам болезни Паркинсона (5 форм - Дрожательно-ригидная, Акинетико-ригидная, Акинетическая форма, Дрожательная форма, ригидная форма) на основе пяти признаков – содержания в крови глюкозы, тромбоцитов, белков, эозинофилов, плазмоцитов. Использовать машину опорных векторов и алгоритм персептрона. Полученные результаты сравнить с использованием функций языка R - Predict и Table.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **№** | **Группы** | **Glukosa** | **Trombocit** | **Belok** | **Eozinofil** | **Plazmocit** |
| 1 | Дрожательно-ригидная | 4.47-8.34 | 2.34-5.51 | 5.33-9.89 | 1.77-3.5 | 9.2-11.7 |
| 2 | Акинетико-ригидная | 8.35-11.61 | 5.52-8.73 | 9.9-14.41 | 3.6-7.1 | 11.8-16.93 |
| 3 | Дрожательная форма | 11.62-17.16 | 8.74-10.93 | 14.42-18.1 | 7.2-12.3 | 16.94-21.33 |
| 4 | ригидная форма | 17.17-19.21 | 10.94-14.32 | 18.2-20.4 | 12.4-17.8 | 21.34-26.57 |
| 5 | Акинетическая форма | 19.22-23.43 | 14.33-18.57 | 20.41-27.3 | 17.9-22.89 | 26.58-31.08 |

Для машины опорных векторов среди ядер "polynomial", "radial" выберите оптимальное в плане количества ошибок на тестовой выборке, для ядра типа polynomial можно изменять значение параметра degree

1. Допустим, что решающая функция линейного классификатора в упрощенном виде выглядит так:



Найти координаты и значение функции в точке минимума методом градиентного спуска.

**Вариант 7**

1. Создайте фрейм данных из *N* = 32 записей со следующими полями: *Nrow* – номер записи, *Name* – имя пациента, *BirthYear* – год рождения, *Employ* – место работы, *Salary* – зарплата. Cost–стоимость лечения, Basifil–содержание базифилов в крови, Neitrofil–содержание нейтрофилов в крови, Limfocit–содержание лимфоцитов в крови, Leikocit – содержание лейкоцитов в крови. Заполните данный фрейм данными так, что *Nrow* изменяется от 1 до *N*,

Name задается произвольно, *BithYear* распределен равномерно (случайно) на отрезке [1972*,*1992], , Costдля пациентов младше 1988 г.р. определяется по формуле Cost= (*ln*(2013 − Birth*Y ear*) +1) ∗ 11000, для остальных Cost= (*log*2(2013 − Birth*Y ear*) + 1) ∗ 9000.

Ранжируйте пациентов по стоимости лечения, начиная с минимальной суммы. Добавьте в таблицу поле, соответствующее общему социальному вычету за лечение (ставка 13%), выплаченному пациенту, если стоимость лечения за каждый год начислялась согласно формулам для Cost, где вместо 2013 следует последовательно подставить каждый год нахождения пациента под наблюдением.

1. Постройте линейный классификатор для классификации пациентов по формам болезни Паркинсона (4 формы - Дрожательно-ригидная, Акинетико-ригидная, Акинетическая форма, Дрожательная форма) на основе четырех признаков – содержания в крови базифилов, нейтрофилов, лимфоцитов, лейкоцитов. Использовать машину опорных векторов и алгоритм персептрона. Полученные результаты сравнить с использованием функций языка R - Predict и Table.

Для машины опорных векторов среди ядер линейное и сигмоидальное выберите оптимальное в плане количества ошибок на тестовой и обучающей выборке

1. Допустим, что решающая функция линейного классификатора в упрощенном виде выглядит так:



Найти координаты и значение функции в точке минимума методом градиентного спуска.

**Вариант 8**

1. Создайте фрейм данных из *N* = 27 записей со следующими полями: *Nrow* – номер записи, *Name* – имя пациента, *BirthYear* – год рождения, *Employ* – место работы, *Salary* – зарплата. Cost–стоимость лечения, Gemoglobin–содержание гемоглобина в крови, Limfocit–содержание лимфоцитов в крови, Bilirubin–содержание билирубина в крови. Заполните данный фрейм данными так, что *Nrow* изменяется от 1 до *N*,

Name задается произвольно, *BithYear* распределен равномерно (случайно) на отрезке [1971*,*1991], , Costдля пациентов младше 1988 г.р. определяется по формуле Cost= (*ln*(2010 − Birth*Y ear*) +1) ∗ 10000, для остальных Cost= (*log*2(2010 − Birth*Y ear*) + 1) ∗7000.

Ранжируйте пациентов по стоимости лечения, начиная с минимальной суммы. Добавьте в таблицу поле, соответствующее общему социальному вычету за лечение (ставка 13%), выплаченному пациенту, если стоимость лечения за каждый год начислялась согласно формулам для Cost, где вместо 2010 следует последовательно подставить каждый год нахождения пациента под наблюдением.

1. Постройте линейный классификатор для классификации пациентов по формам болезни Паркинсона (3 формы - Дрожательно-ригидная, Акинетико-ригидная, Дрожательная форма) на основе трех признаков – содержания в крови гемоглобина, лимфоцитов, билирубина. Использовать машину опорных векторов и алгоритм персептрона. Полученные результаты сравнить с использованием функций языка R - Predict и Table.

Для машины опорных векторов среди ядер полиномиальное и сигмоидальное выберите оптимальное в плане количества ошибок на тестовой и обучающей выборке (для ядра типа polynomial можно изменять значение параметра degree)

1. Допустим, что решающая функция линейного классификатора в упрощенном виде выглядит так:



Найти координаты и значение функции в точке минимума методом градиентного спуска.

**Вариант 9**

1. Создайте фрейм данных из *N* = 28 записей со следующими полями: *Nrow* – номер записи, *Name* – имя сотрудника, *BirthYear* – год рождения, *EmployYear* – год приема на работу, *Salary* – зарплата. EyEColor–цвет глаз, SkinColor–цвет кожи, BloodType–группа крови, HairColor–цвет волос на голове. Заполните данный фрейм данными так, что *Nrow* изменяется от 1 до *N*,

Name задается произвольно, *BithYear* распределен равномерно (случайно) на отрезке [1974*,*1994], *EmployYear* распределен равномерно на отрезке [*BirthY ear*+17*,*2013], *Salary* для работников младше 1990 г.р. определяется по формуле *Salary* = (*ln*(2014 − *EmployY ear*) +1) ∗ 9000, для остальных *Salary* = (*log*2(2014− *EmployY ear*) + 1) ∗ 9000.

Подсчитайте число сотрудников с зарплатой, большей 18000. Добавьте в таблицу поле, соответствующее суммарному подоходному налогу (ставка 13%), выплаченному сотрудником за время работы в организации, если его зарплата за каждый год начислялась согласно формулам для *Salary*, где вместо 2014 следует последовательно подставить каждый год работы сотрудника в организации.

1. Постройте линейный классификатор для классификации сотрудников данной международной организации (признаки классификации: группа крови, цвет волос, глаз и цвет кожи). Использовать машину опорных векторов и алгоритм персептрона. Полученные результаты сравнить.

Цвет глаз, кожи или волос можно закодировать определенным числом. В данном варианте использовать национальности: Англичанин, Китаец, Датчанин, Китаец из Пекина, Исландец.

Для машины опорных векторов типа "C-classification" с полиномиальным ядром, добейтесь нулевой ошибки сначала на обучающей выборке, а затем на тестовой, путем изменения параметра *C* (для ядра типа polynomial можно изменять значение параметра degree).

1. Допустим, что решающая функция линейного классификатора в упрощенном виде выглядит так:



Найти координаты и значение функции в точке минимума методом наискорейшего градиентного спуска.

**Вариант 10**

1. Создайте фрейм данных из *N* = 32 записей со следующими полями: *Nrow* – номер записи, *Name* – имя пациента, *BirthYear* – год рождения, *Employ* – место работы, *Salary* – зарплата. Cost–стоимость лечения, Plazmocitt–содержание плазмоцитов в крови, Gemoglobin–содержание гемоглобина в крови, Neitrifil–содержание нейтрофилов в крови, Creatin – содержание креатина в крови, Gematokrit – содержание гематокрита в крови, Globulin – содержание глобулина в крови. Заполните данный фрейм данными так, что *Nrow* изменяется от 1 до *N*,

Name задается произвольно, *BithYear* распределен равномерно (случайно) на отрезке [1972*,*1994], , Costдля пациентов младше 1986 г.р. определяется по формуле Cost= (*ln*(2010 − Birth*Y ear*) +1) ∗ 7000, для остальных Cost= (*log*2(2010 − Birth*Y ear*) + 1) ∗ 9000.

Ранжируйте пациентов по стоимости лечения, начиная с максимальной суммы. Добавьте в таблицу поле, соответствующее общему социальному вычету за лечение (ставка 13%), выплаченному пациенту, если стоимость лечения за каждый год начислялась согласно формулам для Cost, где вместо 2010 следует последовательно подставить каждый год нахождения пациента под наблюдением.

1. Постройте линейный классификатор для классификации пациентов по степени развития болезни Паркинсона (3 степени - маловыраженные односторонние изменения, маловыраженные двусторонние изменения, полная потеря трудоспособности и самостоятельной жизнедеятельности) на основе шести признаков – содержания в крови плазмоцитов, гемоглобина, нейтрофилов, креатина, гематокрита, глобулина. Использовать машину опорных векторов и алгоритм персептрона. Полученные результаты сравнить с использованием функций языка R - Predict и Table.

Для машины опорных векторов типа "C-classification" с радиальным ядром, добейтесь нулевой ошибки сначала на обучающей выборке, а затем на тестовой, путем изменения параметра *C*.

1. Допустим, что решающая функция линейного классификатора в упрощенном виде выглядит так:



Найти координаты и значение функции в точке минимума методом градиентного спуска.

Таблица характеристик полезных ископаемых – цветных металлов

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Полезные ископаемые | Энергия ионизации атома, ЭВ | Относительная электроотрицательность | Радиус атома, НМ |
| 1 | Церий (редкоземельные) | 5.6 | 1.12 | 0.181 |
| 2 | Лантан (редкоземельные) | 5.61 | 1.1 | 0.187 |
| 3 | Самарий (редкоземельные) | 5.4 | 1.17 | 0.181 |
| 4 | Европий (редкоземельные) | 5.47 | 1.2 | 0.185 |
| 5 | Неодим (редкоземельные) | 5.32 | 1.14 | 0.182 |
| 6 | Тербий (редкоземельные) | 5.69 | 1.2 | 0.18 |
| 7 | Гадолиний (редкоземельные) | 5.94 | 1.2 | 0.179 |
| 8 | Диспрозий (редкоземельные) | 5.67 | 1.2 | 0.18 |
| 9 | Эрбий (редкоземельные) | 5.84 | 1.24 | 0.178 |
| 10 | Гольмий (редкоземельные) | 5.74 | 1.23 | 0.179 |
| 11 | Лютеций (редкоземельные) | 5.13 | 1.27 | 0.175 |
| 12 | Иттрий (редкоземельные) | 6.15 | 1.22 | 0.178 |
| 13 | Тулий (редкоземельные) | 5.89 | 1.25 | 0.177 |
| 14 | Тербий (редкоземельные) | 5.69 | 1.2 | 0.18 |
| 15 | Бериллий (легкие цветные металлы) | 8.98 | 1.67 | 0.112 |
| 16 | Барий (легкие цветные металлы) | 5.21 | 0.89 | 0.222 |
| 17 | Цинк (тяжелые) | 9.06 | 1.65 | 0.138 |
| 18 | Олово (тяжелые) | 7.34 | 1.96 | 0.162 |
| 19 | Титан (легкие) | 6.58 | 1.54 | 0.147 |
| 20 | Литий (легкие) | 5.39 | 0.98 | 0.155 |
| 21 | Рубидий (легкие) | 4.17 | 0.82 | 0.248 |
| 22 | Цезий (легкие цветные металлы) | 3.89 | 0.79 | 0.267 |

**Вариант 11**

1. Создайте фрейм данных из *N* = 29 записей со следующими полями: *Nrow* – номер записи, *Name* – место нахождение цветного металла, *BirthYear* – год обнаружения, Count–прогнозируемое количество, *Salary* – стоимость добычи. Cost–стоимость перевозки, PowerIOn–значение энергии ионизации, Electro–относительная электроотрицательности, Radius – радиус атома. Заполните данный фрейм данными так, что *Nrow* изменяется от 1 до *N*,

Name задается произвольно, *BithYear* распределен равномерно (случайно) на отрезке [1971*,*1997], , Costдля ископаемых найденных до 1989 г.р. определяется по формуле Cost= (*ln*(2006 − Birth*Y ear*) +1) ∗ 45000, для остальных Cost= (*log*2(2006 − Birth*Year*) + 1) ∗ 65000. Стоимость добычи *Salary =* (*log*2(2013 + Birth*Year*)+1)\*73000

Ранжируйте ископаемые по стоимости добычи, начиная с максимальной суммы. Добавьте в таблицу поле, соответствующее общему социальному вычету за перевозки грузов (ставка 7%), выплаченному, если стоимость перевозок за каждый год начислялась согласно формулам для Cost, где вместо 2006 следует последовательно подставить каждый год добычи полезных ископаемых.

1. Постройте линейный классификатор для классификации 5 редкоземельных цветных металлов церия, лантана, самария, европия и неодима на основе трех признаков – Энергия ионизации атома, радиус атома, Относительная электроотрицательность. Использовать машину опорных векторов и алгоритм персептрона. Полученные результаты сравнить с использованием функций языка R - Predict и Table.

Для машины опорных векторов типа "C-classification" с линейным ядром, добейтесь нулевой ошибки сначала на обучающей выборке, а затем на тестовой, путем изменения параметра *C*.

1. Допустим, что решающая функция линейного классификатора в упрощенном виде выглядит так:



Найти координаты и значение функции в точке минимума методом градиентного спуска.

**Вариант 12**

1. Создайте фрейм данных из *N* = 33 записей со следующими полями: *Nrow* – номер записи, *Name* – место нахождение цветного металла, *BirthYear* – год обнаружения, Count–прогнозируемое количество, *Salary* – стоимость добычи. Cost–стоимость перевозки, PowerIOn–значение энергии ионизации, Electro–относительная электроотрицательности, Radius – радиус атома. Заполните данный фрейм данными так, что *Nrow* изменяется от 1 до *N*,

Name задается произвольно, *BithYear* распределен равномерно (случайно) на отрезке [1972*,*1996], , Costдля ископаемых найденных до 1990 г.р. определяется по формуле Cost= (*ln*(2012 − Birth*Y ear*) +1) ∗ 55000, для остальных Cost= (*log*2(2012 − Birth*Year*) + 1) ∗ 63000. Стоимость добычи *Salary =* (*log*2(2007 + Birth*Year*)+1)\*74000

Ранжируйте ископаемые по стоимости добычи, начиная с максимальной суммы. Добавьте в таблицу поле, соответствующее общему социальному вычету за перевозки грузов (ставка 8%), выплаченному, если стоимость перевозок за каждый год начислялась согласно формулам для Cost, где вместо 2012 следует последовательно подставить каждый год добычи полезных ископаемых.

1. Постройте линейный классификатор для классификации 4 редкоземельных цветных металлов тербия, гадолиния, диспрозия, эрбия на основе трех признаков – Энергия ионизации атома, радиус атома, Относительная электроотрицательность. Использовать машину опорных векторов и алгоритм персептрона. Полученные результаты сравнить с использованием функций языка R - Predict и Table.

Для машины опорных векторов типа "C-classification" с радиальным ядром, добейтесь нулевой ошибки сначала на обучающей выборке, а затем на тестовой, путем изменения параметра *C*.

1. Допустим, что решающая функция линейного классификатора в упрощенном виде выглядит так:



Найти координаты и значение функции в точке минимума методом градиентного спуска.

**Вариант 13**

1. Создайте фрейм данных из *N* = 30 записей со следующими полями: *Nrow* – номер записи, *Name* – место нахождение цветного металла, *BirthYear* – год обнаружения, Count–прогнозируемое количество, *Salary* – стоимость добычи. Cost–стоимость перевозки, PowerIOn–значение энергии ионизации, Electro–относительная электроотрицательности, Radius – радиус атома. Заполните данный фрейм данными так, что *Nrow* изменяется от 1 до *N*,

Name задается произвольно, *BithYear* распределен равномерно (случайно) на отрезке [1973*,*1999], , Costдля ископаемых найденных до 1990 г.р. определяется по формуле Cost= (*ln*(2008 − Birth*Y ear*) +1) ∗ 57000, для остальных Cost= (*log*2(2008 − Birth*Year*) + 1) ∗ 67000. Стоимость добычи *Salary =* (*log*2(2005 + Birth*Year*)+1)\*77000

Ранжируйте ископаемые по стоимости добычи, начиная с максимальной суммы. Добавьте в таблицу поле, соответствующее общему социальному вычету за перевозки грузов (ставка 11%), выплаченному, если стоимость перевозок за каждый год начислялась согласно формулам для Cost, где вместо 2008 следует последовательно подставить каждый год добычи полезных ископаемых.

1. Постройте линейный классификатор для классификации 5 редкоземельных цветных металлов гольмия, лютеция, иттрия, тулия, тербия на основе трех признаков – Энергия ионизации атома, радиус атома, Относительная электроотрицательность. Использовать машину опорных векторов и алгоритм персептрона. Полученные результаты сравнить с использованием функций языка R - Predict и Table.

Для машины опорных векторов типа "C-classification" с полиномиальным ядром, добейтесь нулевой ошибки сначала на обучающей выборке, а затем на тестовой, путем изменения параметра *C*.Изменяя значение параметра gamma, продемонстрируйте эффект переобучения, выполните при этом визуализацию разбиения пространства признаков на области

1. Допустим, что решающая функция линейного классификатора в упрощенном виде выглядит так:



Найти координаты и значение функции в точке минимума методом градиентного спуска.

**Вариант 14**

1. Создайте фрейм данных из *N* = 27 записей со следующими полями: *Nrow* – номер записи, *Name* – место нахождение цветного металла, *BirthYear* – год обнаружения, Count–прогнозируемое количество, *Salary* – стоимость добычи. Cost–стоимость перевозки, PowerIOn–значение энергии ионизации, Electro–относительная электроотрицательности, Radius – радиус атома. Заполните данный фрейм данными так, что *Nrow* изменяется от 1 до *N*,

Name задается произвольно, *BithYear* распределен равномерно (случайно) на отрезке [1970*,*1994], , Costдля ископаемых найденных до 1977 г.р. определяется по формуле Cost= (*ln*(2009 − Birth*Y ear*) +1) ∗ 59000, для остальных Cost= (*log*2(2009 − Birth*Year*) + 1) ∗ 69000. Стоимость добычи *Salary =* (*log*2(2007 + Birth*Year*)+1)\*79000

Ранжируйте ископаемые по стоимости добычи, начиная с минимальной суммы. Добавьте в таблицу поле, соответствующее общему социальному вычету за перевозки грузов (ставка 10%), выплаченному, если стоимость перевозок за каждый год начислялась согласно формулам для Cost, где вместо 2009 следует последовательно подставить каждый год добычи полезных ископаемых.

1. Постройте линейный классификатор для классификации тяжелых и легких цветных металлов цинка, олова, бериллия, титана, лития, рубидия на основе трех признаков – Энергия ионизации атома, радиус атома, Относительная электроотрицательность. Использовать машину опорных векторов и алгоритм персептрона. Полученные результаты сравнить с использованием функций языка R - Predict и Table.

Для машины опорных векторов типа "C-classification" с сигмоидальным ядром, добейтесь нулевой ошибки сначала на обучающей выборке, а затем на тестовой, путем изменения параметра *C*.Изменяя значение параметра gamma, продемонстрируйте эффект переобучения, выполните при этом визуализацию разбиения пространства признаков на области

1. Допустим, что решающая функция линейного классификатора в упрощенном виде выглядит так:



Найти координаты и значение функции в точке минимума методом градиентного спуска.

**Вариант 15**

1. Создайте фрейм данных из *N* = 31 записей со следующими полями: *Nrow* – номер записи, *Name* – место нахождение цветного металла, *BirthYear* – год обнаружения, Count–прогнозируемое количество, *Salary* – стоимость добычи. Cost–стоимость перевозки, PowerIOn–значение энергии ионизации, Electro–относительная электроотрицательности, Radius – радиус атома. Заполните данный фрейм данными так, что *Nrow* изменяется от 1 до *N*,

Name задается произвольно, *BithYear* распределен равномерно (случайно) на отрезке [1974*,*1999], , Costдля ископаемых найденных до 1977 г.р. определяется по формуле Cost= (*ln*(2010 − Birth*Y ear*) +1) ∗ 54000, для остальных Cost= (*log*2(2010 − Birth*Year*) + 1) ∗ 64000. Стоимость добычи *Salary =* (*log*2(2009 + Birth*Year*)+1)\*74000

Ранжируйте ископаемые по стоимости добычи, начиная с минимальной суммы. Добавьте в таблицу поле, соответствующее общему социальному вычету за перевозки грузов (ставка 8%), выплаченному, если стоимость перевозок за каждый год начислялась согласно формулам для Cost, где вместо 2010 следует последовательно подставить каждый год добычи полезных ископаемых.

1. Постройте линейный классификатор для классификации тяжелых, легких и редкоземельных цветных металлов цинка, олова, бериллия, титана, лития, рубидия, лантана, самария, европия, эрбия на основе трех признаков – Энергия ионизации атома, радиус атома, Относительная электроотрицательность. Использовать машину опорных векторов и алгоритм персептрона. Полученные результаты сравнить с использованием функций языка R - Predict и Table.

Для машины опорных векторов типа "C-classification" с полиномиальным ядром, добейтесь нулевой ошибки сначала на обучающей выборке, а затем на тестовой, путем изменения параметра *C*.Изменяя значение параметра gamma, продемонстрируйте эффект переобучения, выполните при этом визуализацию разбиения пространства признаков на области

1. Допустим, что решающая функция линейного классификатора в упрощенном виде выглядит так:



Найти координаты и значение функции в точке минимума методом градиентного спуска.