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 № 4
по курсу «Методы машинного обучения»

Логические методы классификации многомерных объектов пересекающихся классов

ПРИЛОЖЕНИЕ

Калуга
2018

Варианты заданий

Таблица с перечнем из номенклатурного списка **супермаркета «Новая Эра»**

Код	Наименование	Категория	Эталонная цена	
001	Кока-Кола	Напитки	60	
002	Спрайт	Напитки	70	
003	Добрый Палпи	Напитки	50	
004	Сила Фруктов	Напитки	45	
005	Кофе Лебо Голд		250	
006	Чай Гринфилд		110	
007	Чай Липтон		115	
008	Пиво Хейнекен		80	
009	Пиво Холсен		90	
010	Пиво Балтика		60	
011	Чипсы Лэйс		70	
012	Чипсы Эстрелла		83	
013	Чипсы Принглс		160	
	Пиво Бад		75	
014	Лимонад Аквалайн		44	
015	Лимонад Лайман		47	
016	Сок Вико		70	
017	Сок Джей 7		110	
018	Пюре Агуша		60	
019	Грудинка Столичная		140	
020	Балык Дарницкий		154	
021	Колбаса Краковская		52	
022	Сервелат Старорузский		200	
023	Колбаса Док-		350	

	торская			
024	Ветчина Мясная		165	
025	Сосиски Папа Может		125	
026	Сосиски Премиум		140	
027	Йогурт Вкуснотеево		85	
028	Ряженка Домик в Деревне		84	
030	Сливки Простоквашина		126	
031	Йогурт Чудо Детки		42	
032	Йогурт Данон		35	
033	Творожок Чудо		83	
034	Пломбир Ванильный		45	
035	Мороженое 48 копеек		81	
036	Сыр Брест Литовск		176	
037	Сыр Чеддер		186	
038	Майонез Провансаль Оливковый	Продукты	67	
039	Майонез Провансаль Ряба	Продукты	73	
040	Пицца Ресторанте	Продукты	264	
041	Пельмени Мяснушки	Продукты	127	
042	Кофе Черная Карта		300	
043	Чай Акбар		76	
044	Кофе Амбассадор		231	
045	Кофе Милагро		240	
046	Чай Ахмад	Деликатесы	280	

047	Кофе Якобс	Деликатесы	386	
048	Чай Тесс	Деликатесы	88	
049	Конфеты Вдохновение	Деликатесы	250	
050	Сайра Тихоокеанская	Морепродукты	98	

Каждый пункт номенклатурного списка характеризуется коэффициентом дня недели (от понедельника до воскресенья), сезонным коэффициентом времени года и температурным коэффициентом. Кп- коэф. понедельника, Кв – коэф. вторника, Кс – к-т среды, Кч – к-т четверга, Кп- к-т пятницы, Ксб- к-т субботы, Квс – к-т воскресенья, Кз – зимний коэфф., Кл – летний коэф., Ко- к-т осени, Кт – температурный коэффициент. Эталонная цена умножается на характеристические коэффициенты и так рассчитывается значение скидки на товар в выбранный день недели. В задании лабораторной работы надо классифицировать полученную скидку выбранных товаров в определенный день по категориям: «Скидки - Нет» 0-4 %, «Скидка-мини» 5-25 %, «Выгодная Скидка» 26-40%, «Супер Скидка» 50-70%.

Вариант 1

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	
001	1	0.11	0.21	0.88	0.32	0.85	1	0.95	0.67	0.73	0.98	1	
002	0.4	1	0.25	0.89	0.42	0.94	0.43	1	0.63	0.21	0.97	1	
003	0.7	0.34	1	0.76	0.51	0.67	1	0.85	1	0.18	0.96	0.1	
004	0.5	0.45	0.38	1	0.74	1	0.49	0.74	0.59	1	0.95	0.2	
005	0.6	0.29	0.37	0.43	1	0.63	0.51	1	0.35	0.95	1	0.3	
006	0.2	0.78	0.71	1	0.81	1	0.56	0.28	1	0.63	0.89	1	
007	0.1	0.94	1	0.65	0.97	0.35	1	0.19	0.26	1	0.87	0.9	
008	0.9	1	0.17	0.68	0.99	0.26	0.64	1	0.17	0.29	0.58	1	
009	1	0.21	0.54	1	0.11	1	0.77	0.21	1	0.39	1	0.8	
010	0.8	0.86	0.33	0.94	1	0.17	0.89	0.73	0.76	1	0.48	0.7	

Фактическая стоимость каждого товара рассчитывается по формуле: Стоимость Факт = Стоимость Эталон * Коэфф Дня * Коэфф Время Года * Коэфф Температура

Итак, пять дней в году: 02.01.2017 (понедельник - зима), 14.02.2017 (вторник - зима), 08.03.2017 (среда - весна), 21.07.2017 (пятница - лето), 29.10.2017 (воскресенье - осень).

Код	02.01.2017	14.02.2017	08.03.2017	21.07.2017	29.10.2017	Эталон
001	57,00	6,27	8,44	14,02	58,80	60
002	28,00	70,00	11,03	6,17	29,20	70
003	2,98	1,45	5,00	0,46	4,80	50
004	3,33	3,00	2,02	6,66	4,19	45
005	45,00	21,75	9,71	71,25	38,25	250
006	6,16	24,02	78,10	56,13	54,82	110
007	1,97	18,49	26,91	100,40	90,05	115
008	72,00	80,00	2,31	22,97	29,70	80
009	15,12	3,18	38,88	3,09	55,44	90
010	24,53	26,37	10,53	42,00	17,94	60

Код	Кп	Кз	Кт	Эталон	02.01.2017
1	1,000	0,950	1,000	60,000	57,00
2	0,400	1,000	1,000	70,000	28,00
3	0,700	0,850	0,100	50,000	2,98
4	0,500	0,740	0,200	45,000	3,33
5	0,600	1,000	0,300	250,000	45,00
6	0,200	0,280	1,000	110,000	6,16
7	0,100	0,190	0,900	115,000	1,97
8	0,900	1,000	1,000	80,000	72,00
9	1,000	0,210	0,800	90,000	15,12

10	0,800	0,730	0,700	60,000	24,53

Код	Кв	Кз	Кт	Эталон	14.02.2017
1	0,11	0,950	1,000	60,000	6,27
2	1	1,000	1,000	70,000	70,00
3	0,34	0,850	0,100	50,000	1,45
4	0,45	0,740	0,200	45,000	3,00
5	0,29	1,000	0,300	250,000	21,75
6	0,78	0,280	1,000	110,000	24,02
7	0,94	0,190	0,900	115,000	18,49
8	1	1,000	1,000	80,000	80,00
9	0,21	0,210	0,800	90,000	3,18
10	0,86	0,730	0,700	60,000	26,37

Код	Кс	Квн	Кт	Эталон	08.03.2017
1	0,21	0,67	1,000	60,000	8,44
2	0,25	0,63	1,000	70,000	11,03
3	1	1	0,100	50,000	5,00
4	0,38	0,59	0,200	45,000	2,02
5	0,37	0,35	0,300	250,000	9,71
6	0,71	1	1,000	110,000	78,10
7	1	0,26	0,900	115,000	26,91
8	0,17	0,17	1,000	80,000	2,31
9	0,54	1	0,800	90,000	38,88
10	0,33	0,76	0,700	60,000	10,53

Код	Кпт	Кл	Кт	Эталон	21.07.2017
1	0,32	0,73	1,000	60,000	14,02
2	0,42	0,21	1,000	70,000	6,17
3	0,51	0,18	0,100	50,000	0,46
4	0,74	1	0,200	45,000	6,66
5	1	0,95	0,300	250,000	71,25
6	0,81	0,63	1,000	110,000	56,13
7	0,97	1	0,900	115,000	100,40
8	0,99	0,29	1,000	80,000	22,97
9	0,11	0,39	0,800	90,000	3,09
10	1	1	0,700	60,000	42,00

Код	Квс	Ко	Кт	Эталон	29.10.2017
1	1	0,98	1,000	60,000	58,80
2	0,43	0,97	1,000	70,000	29,20
3	1	0,96	0,100	50,000	4,80
4	0,49	0,95	0,200	45,000	4,19
5	0,51	1	0,300	250,000	38,25
6	0,56	0,89	1,000	110,000	54,82
7	1	0,87	0,900	115,000	90,05
8	0,64	0,58	1,000	80,000	29,70
9	0,77	1	0,800	90,000	55,44
10	0,89	0,48	0,700	60,000	17,94

по категориям «Скидки - Нет» 0-4 %, «Скидка-мини» 5-25 %, «Выгодная Скидка» 26-40%, «Супер Скидка» 50-99% за пять дней.

База знаний. Условия достижения результата. Конъюнкции

1 .Пф =0
2. Пф > 0

3. Пф < 4
4. Пф = 4
5. Пф =5
6. Пф > 5
7. Пф = 50
8. Пф < 50
9. Пф =51
10. Пф < 51
11. Пф = 95
12. Пф < 95
13. Пф =96
14. Пф < 96
15. Пф = 100
16. Пф > 4
17. Пф < 5
18. Пф > 50
19. Пф > 96
20. Пф > 95
21. Пф > 51
22. Пф = 100

Переведем фактическую стоимость в проценты от Эталонной стоимости:
Пф = Сф * 100 / Сэп

Процент1	Процент2	Процент3	Процент4	Процент5
95,00	10,45	14,07	23,37	98,00
40,00	100,00	15,76	8,81	41,71
5,96	2,90	10,00	0,92	9,60
7,40	6,67	4,49	14,80	9,31
18,00	8,70	3,88	28,50	15,30
5,60	21,84	71,00	51,03	49,84
1,71	16,08	23,40	87,30	78,30
90,00	100,00	2,89	28,71	37,13
16,80	3,53	43,20	3,43	61,60
40,88	43,95	17,55	70,00	29,90

КЛАСС 1 – “Скидки – нет» (96% - 100%) (13,14,15,19,22)
КЛАСС 2 – «Мини - Скидка» (51% - 95 %) (=51, >51, <95, = 95) (9,10,11,12,20,21)
КЛАСС 3 – «Выгодная» (5% - 50%) (=5, >5, <50, =50) (5,6,7,8,17,18)
КЛАСС 4 – «Супер» (0 % - 4 %) (=0, > 0, <4, =4) (1,2,3,4)

Пример программного кода. Генетический анализ конъюнкций.

```
rm(list = ls())

#library(genalg)
#library(ggplot2)

source = data.frame(read.table("logic0401.txt", header = TRUE, sep = ""))
names(source)[1] <- "code"
print("Исходные данные")
print(source)

bz = data.frame(read.table("bz.txt", header = TRUE, sep = ""))
names(bz)[1] <- "NN"
print("База конъюнкций")
print(bz)

rw = data.frame(read.table("rows.txt", header = TRUE, sep = ""))
names(rw)[1] <- "NN"
print("Список ранжирования")
print(rw)


resday1 <- matrix(0:0, nrow=24, ncol=10)

ranj <- matrix(0:0, nrow=8, ncol=3)
for (i in 1:length(source$code)) {
   for (j in 1:length(rw$NN))          {

   if (source$PR1[i] < rw$DT[j])  {
       ranj[j,1] = 1  }
  }
```

```
  m <- 1
  for (n in 1:length(rw$NN)) {
       resday1[m,i] = ranj[n,1]
            m = m +3
                }
  ranj <- matrix(0:0, nrow=8, ncol=3)
}

for (i in 1:length(source$code)) {
   for (j in 1:length(rw$NN))          {

   if (source$PR1[i] == rw$DT[j])  {
       ranj[j,2] = 1  }
  }

  m <- 2
  for (n in 1:length(rw$NN)) {
       resday1[m,i] = ranj[n,2]
            m = m +3
                }
   ranj <- matrix(0:0, nrow=8, ncol=3)
}

for (i in 1:length(source$code)) {
   for (j in 1:length(rw$NN))          {

   if (source$PR1[i] > rw$DT[j])  {
       ranj[j,3] = 1  }
  }

  m <- 3
  for (n in 1:length(rw$NN)) {
       resday1[m,i] = ranj[n,3]
            m = m +3
                }
   ranj <- matrix(0:0, nrow=8, ncol=3)
}

resday1
```

```
resday2 <- matrix(0:0, nrow=24, ncol=10)

ranj <- matrix(0:0, nrow=8, ncol=3)
for (i in 1:length(source$code)) {
   for (j in 1:length(rw$NN))          {

   if (source$PR2[i] < rw$DT[j])  {
       ranj[j,1] = 1  }
  }

  m <- 1
  for (n in 1:length(rw$NN)) {
       resday2[m,i] = ranj[n,1]
           m = m +3
               }
  ranj <- matrix(0:0, nrow=8, ncol=3)
}

for (i in 1:length(source$code)) {
   for (j in 1:length(rw$NN))          {

   if (source$PR2[i] == rw$DT[j])  {
       ranj[j,2] = 1  }
  }

  m <- 2
  for (n in 1:length(rw$NN)) {
       resday2[m,i] = ranj[n,2]
           m = m +3
               }
   ranj <- matrix(0:0, nrow=8, ncol=3)
}

for (i in 1:length(source$code)) {
   for (j in 1:length(rw$NN))          {

   if (source$PR2[i] > rw$DT[j])  {
       ranj[j,3] = 1  }
  }

  m <- 3
```

```
   for (n in 1:length(rw$NN)) {
        resday2[m,i] = ranj[n,3]
            m = m +3
                }
    ranj <- matrix(0:0, nrow=8, ncol=3)
}

resday2



resday3 <- matrix(0:0, nrow=24, ncol=10)

ranj <- matrix(0:0, nrow=8, ncol=3)
for (i in 1:length(source$code)) {
    for (j in 1:length(rw$NN))          {

    if (source$PR3[i] < rw$DT[j])  {
        ranj[j,1] = 1  }
   }

   m <- 1
   for (n in 1:length(rw$NN)) {
        resday3[m,i] = ranj[n,1]
            m = m +3
                }
   ranj <- matrix(0:0, nrow=8, ncol=3)
}

for (i in 1:length(source$code)) {
    for (j in 1:length(rw$NN))          {

    if (source$PR3[i] == rw$DT[j])  {
        ranj[j,2] = 1  }
   }

   m <- 2
   for (n in 1:length(rw$NN)) {
        resday3[m,i] = ranj[n,2]
            m = m +3
                }
```

```
   ranj <- matrix(0:0, nrow=8, ncol=3)
}

for (i in 1:length(source$code)) {
   for (j in 1:length(rw$NN))          {

   if (source$PR3[i] > rw$DT[j])  {
       ranj[j,3] = 1  }
  }

  m <- 3
  for (n in 1:length(rw$NN)) {
       resday3[m,i] = ranj[n,3]
           m = m +3
               }
   ranj <- matrix(0:0, nrow=8, ncol=3)
}

resday3



resday4 <- matrix(0:0, nrow=24, ncol=10)

ranj <- matrix(0:0, nrow=8, ncol=3)
for (i in 1:length(source$code)) {
   for (j in 1:length(rw$NN))          {

   if (source$PR4[i] < rw$DT[j])  {
       ranj[j,1] = 1  }
  }

  m <- 1
  for (n in 1:length(rw$NN)) {
       resday4[m,i] = ranj[n,1]
           m = m +3
               }
  ranj <- matrix(0:0, nrow=8, ncol=3)
}

for (i in 1:length(source$code)) {
```

```
   for (j in 1:length(rw$NN))          {

   if (source$PR4[i] == rw$DT[j])  {
       ranj[j,2] = 1  }
  }

  m <- 2
  for (n in 1:length(rw$NN)) {
       resday4[m,i] = ranj[n,2]
           m = m +3
               }
   ranj <- matrix(0:0, nrow=8, ncol=3)
}

for (i in 1:length(source$code)) {
   for (j in 1:length(rw$NN))          {

   if (source$PR4[i] > rw$DT[j])  {
       ranj[j,3] = 1  }
  }

  m <- 3
  for (n in 1:length(rw$NN)) {
       resday4[m,i] = ranj[n,3]
           m = m +3
               }
   ranj <- matrix(0:0, nrow=8, ncol=3)
}

resday4



resday5 <- matrix(0:0, nrow=24, ncol=10)

ranj <- matrix(0:0, nrow=8, ncol=3)
for (i in 1:length(source$code)) {
   for (j in 1:length(rw$NN))          {

   if (source$PR5[i] < rw$DT[j])  {
       ranj[j,1] = 1  }
```

```
  }

  m <- 1
  for (n in 1:length(rw$NN)) {
       resday5[m,i] = ranj[n,1]
            m = m +3
                 }
  ranj <- matrix(0:0, nrow=8, ncol=3)
}

for (i in 1:length(source$code)) {
   for (j in 1:length(rw$NN))          {

   if (source$PR5[i] == rw$DT[j])  {
       ranj[j,2] = 1  }
  }

  m <- 2
  for (n in 1:length(rw$NN)) {
       resday5[m,i] = ranj[n,2]
            m = m +3
                 }
   ranj <- matrix(0:0, nrow=8, ncol=3)
}

for (i in 1:length(source$code)) {
   for (j in 1:length(rw$NN))          {

   if (source$PR5[i] > rw$DT[j])  {
       ranj[j,3] = 1  }
  }

  m <- 3
  for (n in 1:length(rw$NN)) {
       resday5[m,i] = ranj[n,3]
            m = m +3
                 }
   ranj <- matrix(0:0, nrow=8, ncol=3)
}

resday5
```

```
total_w <- matrix(0:0, nrow=24, ncol=1)

for (i in (1:24)) {
  for (j in (1:10))  {
      if (resday1[i,j] == 1) {
            total_w[i] = total_w[i] +1
           }
      if (resday2[i,j] == 1) {
            total_w[i] = total_w[i] +1
           }
      if (resday3[i,j] == 1) {
            total_w[i] = total_w[i] +1
           }
      if (resday4[i,j] == 1) {
            total_w[i] = total_w[i] +1
           }
      if (resday5[i,j] == 1) {
            total_w[i] = total_w[i] +1
           }

    }

}


write.table(total_w, file="summs.txt")
total_w

ga_chrom <- matrix(0:0, nrow=25, ncol=10)

for (i in 1:10) {
  for (j in 1:24) {
      ga_chrom[j,i] = sample(0:1,1)
   }
       }

ga_chrom


for (i in 1:10) {
```

```
  for (j in 1:24) {
     if (ga_chrom[j,i] == 1) {
         for (n in 1:10) {
              ga_chrom[25,i] = ga_chrom[25,i]+resday1[j,n]
              ga_chrom[25,i] = ga_chrom[25,i]+resday2[j,n]
              ga_chrom[25,i] = ga_chrom[25,i]+resday3[j,n]
              ga_chrom[25,i] = ga_chrom[25,i]+resday4[j,n]
              ga_chrom[25,i] = ga_chrom[25,i]+resday5[j,n]
                    }
                     }
  }
      }
ga_chrom


gamin1                                                                    <-
c(ga_chrom[25,1],ga_chrom[25,2],ga_chrom[25,3],ga_chrom[25,4],ga_chrom
[25,5],ga_chrom[25,6],ga_chrom[25,7],ga_chrom[25,8],ga_chrom[25,9],ga_ch
rom[25,10])
gamax1                                                                    <-
c(ga_chrom[25,1],ga_chrom[25,2],ga_chrom[25,3],ga_chrom[25,4],ga_chrom
[25,5],ga_chrom[25,6],ga_chrom[25,7],ga_chrom[25,8],ga_chrom[25,9],ga_ch
rom[25,10])

gamin <- min(gamin1)
gamax <- max(gamax1)

gamin
gamax

# GA standart
library(genalg)

#layout (matrix (c(1,2,3,4,5,6,7,8),2,4))
layout (matrix (c(1,2,3,4,5,6,7,8,9,10),2,5))

#evaluate <- function(string=c()) {
#returnVal = 1 / sum(string);
#returnVal
#}
```

```
#rbga.results = rbga.bin(size=10, mutationChance=0.01, zeroToOneRatio=0.5,
#evalFunc=evaluate)
#plot(rbga.results)
#plot(rbga.results, type="hist")

# optimize two values to match pi and sqrt(50)
#evaluate <- function(string=c()) {
#returnVal = NA;
#if (length(string) == 2) {
#returnVal = abs(string[1]-pi) + abs(string[2]-sqrt(50));
#} else {
#stop("Expecting a chromosome of length 2!");
#}
#returnVal
#}
#monitor <- function(obj) {
# plot the population
#xlim = c(obj$stringMin[1], obj$stringMax[1]);
#ylim = c(obj$stringMin[2], obj$stringMax[2]);
#plot(obj$population, xlim=xlim, ylim=ylim,
#xlab="pi", ylab="sqrt(50)");
#}
#rbga.results = rbga(c(1, 1), c(5, 10), monitorFunc=monitor,
#evalFunc=evaluate, verbose=TRUE, mutationChance=0.01)
#plot(rbga.results)
#plot(rbga.results, type="hist")
#plot(rbga.results, type="vars")


evaluate <- function(string=c()) {
returnVal = NA;
if (length(string) == 2) {
returnVal = abs(string[1]) + abs(string[2]);
} else {
stop("Expecting a chromosome of length 2!");
}
returnVal
}
monitor <- function(obj) {
# plot the population
```

```
xlim = c(obj$stringMin[1], obj$stringMax[1]);
ylim = c(obj$stringMin[2], obj$stringMax[2]);
plot(obj$population, xlim=xlim, ylim=ylim,
xlab="X", ylab="Y");
}
rbga.results = rbga(c(1,gamin), c(1, gamax), iters = 5,  monitorFunc=monitor,
evalFunc=evaluate, verbose=TRUE, mutationChance=0.01)
plot(rbga.results)

#plot(rbga.results, type="hist")
#plot(rbga.results, type="vars")
```

1).Для алгоритма Кора: выбрать частоту встречаемости конъюнкций MinNum= 3

Представленный ниже скрипт выполняет перебор всех возможных комбинаций столбцов xi , i=1,…,m=16 по 2, 3 и maxKSize = 4 с использованием функции combn(). Для каждой комбинации столбцов перебираются все возможные варианты событий (x_i = 0 или x_i = 1). Для сокращения объема вычислений подмножества исходной матрицы трансформировались в векторы символьных бинарных переменных, а комбинации значений xi выражались наборами “битовых масок” (например, "000", "001", "010", "011", "100", "101", "110", "111").
В ходе перебора конъюнкции, отобранные по совокупности условий, будем сохранять в трех глобальных объектах класса list, для чего используем оператор глобального присваивания <<-. Определим предварительно несколько функций:

```
# Преобразование числа в набор битов (5 -> "0101")
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)
```

```
      #  Сохранение 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) } )
}
```

Зададим минимальную частоту встречаемости конъюнкций minNum = 4 (т.е. $\tau = 50\%$) и выполним формирование всех логических правил для рассматриваемого примера:

```
DFace <- read.delim(file = "data/Faces.txt",
                 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 = " ")
}

DFvar[is.na(DFvar)] <- " "
DFout <- cbind(DFval, DFvar)

#  Вывод результатов
print("Конъюнкции класса 1")
## [1] "Конъюнкции класса 1"
DFout[DFout$Nclass == 1, ]
print("Конъюнкции класса 2")
## [1] "Конъюнкции класса 2"
DFout[DFout$Nclass == 2, ]
```

Результат, содержащий логические высказывания для каждого класса (Nclass) выглядит стандартным образом, где KSize - длина коньюнкции, Bits - ее битовая маска, L1-L4 - наименования исходных переменных, countK - встречаемость конъюнкции на объектах своего класса.

Существует возможность использовать сгенерированные конъюнкции для экзамена тестируемых примеров по принципу голосования. Чтобы классифицировать новое наблюдение x , подсчитывается число отобранных конъюнкций Lk , характерных для каждого k -го класса, которые верны для тестируемого бинарного вектора. Если Lk является максимальным из всех, то принимается решение о принадлежности объекта k -му классу.

```
rm(list = ls())

source = data.frame(read.table("logic0401.txt", header = TRUE, sep = ""))
names(source)[1] <- "code"
print("Исходные данные")
print(source)
```

```
bz = data.frame(read.table("bz.txt", header = TRUE, sep = ""))
names(bz)[1] <- "NN"
print("База конъюнкций")
print(bz)

rw = data.frame(read.table("rows.txt", header = TRUE, sep = ""))
names(rw)[1] <- "NN"
print("Список ранжирования")
print(rw)



resday1 <- matrix(0:0, nrow=24, ncol=10)
resday1

ranj <- matrix(0:0, nrow=8, ncol=3)
for (i in 1:length(source$code)) {
   for (j in 1:length(rw$NN))          {

   if (source$PR1[i] < rw$DT[j])  {
       ranj[j,1] = 1  }
  }

  m <- 1
  for (n in 1:length(rw$NN)) {
       resday1[m,i] = ranj[n,1]
           m = m +3
               }
  ranj <- matrix(0:0, nrow=8, ncol=3)
}

for (i in 1:length(source$code)) {
   for (j in 1:length(rw$NN))          {

   if (source$PR1[i] == rw$DT[j])  {
       ranj[j,2] = 1  }
  }

  m <- 2
  for (n in 1:length(rw$NN)) {
       resday1[m,i] = ranj[n,2]
```

```
            m = m +3
                }
    ranj <- matrix(0:0, nrow=8, ncol=3)
}

for (i in 1:length(source$code)) {
    for (j in 1:length(rw$NN))          {

    if (source$PR1[i] > rw$DT[j])  {
        ranj[j,3] = 1  }
   }

   m <- 3
   for (n in 1:length(rw$NN)) {
        resday1[m,i] = ranj[n,3]
            m = m +3
                }
    ranj <- matrix(0:0, nrow=8, ncol=3)
}

resday1

# for 10 items (products) of 1 day

krosklas1 <- matrix(0:0, nrow=4, ncol=6)
j <- 1
for (i in 1:length(bz$NN)) {
 if (bz$KS[i] == 1)  {
        krosklas1[1,j] = bz$NN[i]
   j=j+1  }
}
j <- 1
for (i in 1:length(bz$NN)) {
 if (bz$KS[i] == 2)  {
        krosklas1[2,j] = bz$NN[i]
   j=j+1  }
}
j <- 1
for (i in 1:length(bz$NN)) {
 if (bz$KS[i] == 3)  {
        krosklas1[3,j] = bz$NN[i]
```

```
  j=j+1  }
}
j <- 1
for (i in 1:length(bz$NN)) {
 if (bz$KS[i] == 4)  {
        krosklas1[4,j] = bz$NN[i]
  j=j+1  }
}

krosklas1[1,5] = 1
krosklas1[1,6] = 24

krosklas1

klasday1 <- matrix(0:0, nrow=4, ncol=10)


# for all products of day number 1
# product 1

resklas1 <- matrix(0:0, nrow=4, ncol=6)
for (i in 1:6) {
  if (resday1[krosklas1[1,i],1] == 1) {
    resklas1[1,i] = 1       }
}
for (i in 1:6) {
  if (resday1[krosklas1[2,i],1] == 1) {
    resklas1[2,i] = 1       }
}
for (i in 1:6) {
  if (resday1[krosklas1[3,i],1] == 1) {
    resklas1[3,i] = 1       }
}
for (i in 1:6) {
  if (resday1[krosklas1[4,i],1] == 1) {
    resklas1[4,i] = 1       }
}

resklas1
```

```
M11=0
M12=0
M13=0
M14=0
M15=0
M16=0
DD1=0
DD2=0
KK1=0

for (j in 1:4) {
   if ( resklas1[j,1] == 1) {  M11 = 1
            }
   if ( resklas1[j,2] == 1) {  M12 = 1
            }
   if ( resklas1[j,3] == 1) {  M13 = 1
            }
   if ( resklas1[j,4] == 1) {  M14 = 1
            }
   if ( resklas1[j,5] == 1) {  M15 = 1
            }
   if ( resklas1[j,6] == 1) {  M16 = 1
            }

   if ((M11 == 1) || (M12 == 1)) {
               DD1=1
                        }
   if ((M13 == 1) || (M14 == 1)) {
               DD2=1
                        }
   if ((M15 == 1) && (M16 == 1)) {
               KK1=1
                        }

    if (KK1 == 1) {
      klasday1[j,1] = source$PR1[1]
        }
    if ((DD1 == 1) && (DD2 == 1)) {
      klasday1[j,1] = source$PR1[1]
     }
```

```
    M11=0
    M12=0
    M13=0
    M14=0
    M15=0
    M16=0
    DD1=0
    DD2=0
    KK1=0

  }

# product 2
resklas1 <- matrix(0:0, nrow=4, ncol=6)
for (i in 1:6) {
 if (resday1[krosklas1[1,i],2] == 1) {
   resklas1[1,i] = 1       }
}
for (i in 1:6) {
 if (resday1[krosklas1[2,i],2] == 1) {
   resklas1[2,i] = 1       }
}
for (i in 1:6) {
 if (resday1[krosklas1[3,i],2] == 1) {
   resklas1[3,i] = 1       }
}
for (i in 1:6) {
 if (resday1[krosklas1[4,i],2] == 1) {
   resklas1[4,i] = 1       }
}

resklas1



 M11=0
 M12=0
 M13=0
 M14=0
 M15=0
```

```
M16=0
DD1=0
DD2=0
KK1=0

for (j in 1:4) {
   if ( resklas1[j,1] == 1) {  M11 = 1
            }
   if ( resklas1[j,2] == 1) {  M12 = 1
            }
   if ( resklas1[j,3] == 1) {  M13 = 1
            }
   if ( resklas1[j,4] == 1) {  M14 = 1
            }
   if ( resklas1[j,5] == 1) {  M15 = 1
            }
   if ( resklas1[j,6] == 1) {  M16 = 1
            }

   if ((M11 == 1) || (M12 == 1)) {
               DD1=1
                     }
   if ((M13 == 1) || (M14 == 1)) {
               DD2=1
                     }
   if ((M15 == 1) && (M16 == 1)) {
               KK1=1
                     }

    if (KK1 == 1) {
      klasday1[j,2] = source$PR1[2]
       }
    if ((DD1 == 1) && (DD2 == 1)) {
      klasday1[j,2] = source$PR1[2]
     }

    M11=0
    M12=0
    M13=0
    M14=0
    M15=0
```

```
    M16=0
    DD1=0
    DD2=0
    KK1=0

  }

 # product 3 of day 1

for (i7 in 3:length(source$code)) {

resklas1 <- matrix(0:0, nrow=4, ncol=6)
for (i in 1:6) {
  if (resday1[krosklas1[1,i],i7] == 1) {
    resklas1[1,i] = 1       }
}
for (i in 1:6) {
  if (resday1[krosklas1[2,i],i7] == 1) {
    resklas1[2,i] = 1       }
}
for (i in 1:6) {
  if (resday1[krosklas1[3,i],i7] == 1) {
    resklas1[3,i] = 1       }
}
for (i in 1:6) {
  if (resday1[krosklas1[4,i],i7] == 1) {
    resklas1[4,i] = 1       }
}

resklas1



 M11=0
 M12=0
 M13=0
 M14=0
 M15=0
 M16=0
 DD1=0
 DD2=0
```

```
KK1=0

for (j in 1:4) {
   if ( resklas1[j,1] == 1) {  M11 = 1
            }
   if ( resklas1[j,2] == 1) {  M12 = 1
            }
   if ( resklas1[j,3] == 1) {  M13 = 1
            }
   if ( resklas1[j,4] == 1) {  M14 = 1
            }
   if ( resklas1[j,5] == 1) {  M15 = 1
            }
   if ( resklas1[j,6] == 1) {  M16 = 1
            }

   if ((M11 == 1) || (M12 == 1)) {
               DD1=1
                        }
   if ((M13 == 1) || (M14 == 1)) {
               DD2=1
                        }
   if ((M15 == 1) && (M16 == 1)) {
               KK1=1
                        }

    if (KK1 == 1) {
      klasday1[j,i7] = source$PR1[i7]
        }
    if ((DD1 == 1) && (DD2 == 1)) {
      klasday1[j,i7] = source$PR1[i7]
     }

    M11=0
    M12=0
    M13=0
    M14=0
    M15=0
    M16=0
    DD1=0
    DD2=0
```

```
    KK1=0

  }

}

klasday1

layout (matrix (c(1,2,3,4,5,6,7,8),2,4))

#layout.show(1)
attach(source)
plot(code,PR1)
abline(lm(code~PR1))
detach(source)


class3 <- c(klasday1[3,9], klasday1[3,2], klasday1[3,3], klasday1[3,4], klas-
day1[3,5], klasday1[3,6], klasday1[3,10])
class2 <- c(klasday1[2,1], klasday1[2,2], klasday1[2,3], klasday1[2,4], klas-
day1[2,5], klasday1[2,6], klasday1[2,8])
class4 <- c(klasday1[4,1], klasday1[4,2], klasday1[4,3], klasday1[4,4], klas-
day1[4,5], klasday1[4,6], klasday1[4,7])
product <- c("Prod.1","Prod.2","Prod.3","Prod.4","Prod.5","Prod.6","Prod.7")


#layout.show(2)
plot(class3,type="b")
#layout.show(3)
plot(class2,class3,type="b")
#layout.show(4)
plot(product,class3,type="b", col="red")
lines(product,class2,type="b", col="blue")
lines(product,class4,type="b", col="green")
#layout.show(5)
pie(class3, labels = product, main="KLASS 3")

conto1=0
conto2=0
conto3=0
conto4=0
```

```
for (i in 1:10) {
  if (klasday1[1,i] > 0) {
       conto1=conto1+1
       }
  if (klasday1[2,i] > 0) {
       conto2=conto2+1
       }
  if (klasday1[3,i] > 0) {
       conto3=conto1+3
       }
  if (klasday1[4,i] > 0) {
       conto4=conto4+1
       }
}

resX <- c(conto1, conto2, conto3, conto4)
resY <- c("KLas 1","KLas 2","KLas 3","KLas 4")
#layout.show(6)
pie(resX, labels = resY, main=" TOTAL DAY N1")

pct <- round(resX/sum(resX)*100)
resY2 <- paste(resY, " ", pct, "%", sep="")
#layout.show(7)
pie(resX, labels=resY2, col=rainbow(4), main="RING DIAGRAM with %% of DAY 1")


hist(resX, col="green")
```

2).Для генетического алгоритма выбрать: генную бинарную комбинацию **1001101**

```
rm(list = ls())

#library(genalg)
#library(ggplot2)

# GA standart
```

```
library(genalg)

ramFo = data.frame(read.table("GA4pr.txt", header = TRUE, sep = ""))
print("Исходные данные")
print(ramFo)

#ga_chrom = sample(0:1,1)

alg <- matrix(0:0, nrow=150, ncol=4)
#resday1 <- matrix(0:0, nrow=24, ncol=10)

for (i in 1:150) {
  for (j in 1:4)    {
    alg[i,j] = sample(1:99,1)
}
}

write.table(alg, file="ga55.txt")
ramFo2 = data.frame(read.table("ga55.txt", header = TRUE, sep = ""))
ramFo3=cbind(ramFo2, ramFo$V5)
ramFo3


library(MASS)
X <- cbind(scale(ramFo3[,1:4]), matrix(rnorm(36*150), 150, 36))
Y <- ramFo3[,5]
ramFo3.evaluate <- function(indices) {
result = 1
if (sum(indices) > 2) {
huhn <- lda(X[,indices==1], Y, CV=TRUE)$posterior
result = sum(Y != dimnames(huhn)[[2]][apply(huhn, 1,
function(x)
which(x == max(x)))]) / length(Y)
}
result
}
monitor <- function(obj) {
minEval = min(obj$evaluations);
plot(obj, type="hist");
}
woppa <- rbga.bin(size=40, mutationChance=0.05, zeroToOneRatio=10,
```

evalFunc=ramFo3.evaluate, verbose=TRUE, monitorFunc=monitor, iter = 11)

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

```
# algorithm CART - package RPART
library(rpart)


frst = data.frame(read.table("IDCarttreesK.txt", header = TRUE, sep = ""))
print("Список Данных для CART")
print(frst)

    }

layout (matrix (c(1,2,3,4,5,6,7,8,9,10),2,5))


plot(frst$V2, frst$V14,
xlab = "Price", ylab = " <> 50")
mlowess <- function(x, y, ...) {
keep <- !(is.na(x) | is.na(y))
lowess(x[keep], y[keep], ...)
}
with(frst, lines(mlowess(V2, V14, f = 0.5)))

#z.auto <- rpart(V4 ~ V2+V29+V30, frst)
#summary(z.auto)

#fit <- rpart(V4 ~ V2 + V7 + V11, frst)
#par(xpd = TRUE)
#plot(fit, compress = TRUE)
```

```
#text(fit, use.n = TRUE)


fit <- rpart(V4 ~ V2+V29+V30, data = frst)
fit2 <- rpart(V4 ~ V2 + V29 + V30+V11+V12+V7+V8, data = frst)
#parms = list(prior = c(0.65, 0.35), split = "information"))
#fit3 <- rpart(V4 ~ V2 + V29 + V30, data = frst,
#control = rpart.control(cp = 0.05))
#par(mfrow = c(1,2), xpd = TRUE)
plot(fit)
text(fit, use.n = TRUE)
plot(fit2)
text(fit2, use.n = TRUE)

#z.auto <- rpart(V4 ~ V2, frst)
#meanvar(z.auto, log = 'xy')

fit <- rpart(V4 ~ V2 + V29 + V30,
data = frst, method = "anova")
plot(fit)
text(fit, use.n = TRUE)
summary(residuals(fit))
predict(fit)


#z.auto <- rpart(V4 ~ V2+V29+V30, frst)
#zp <- prune(z.auto, cp = 0.1)
#plot(zp) #plot smaller rpart object
```

4).Для алгоритма Bagging в функции RandomForest выбрать параметр N.trees= 300, в функции train выбрать параметр method = "bagFDA"

```
rm(list = ls())

library(randomForest)
#library(ggplot2)
```

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

print("Исходные данные")
print(ramFo)


rf1 <- randomForest(V4 ~ ., ramFo, ntree=30, norm.votes=FALSE)
print(rf1)

getTree(randomForest(ramFo[,-4], ramFo[,4], ntree=17), 7, labelVar=TRUE)

rf2 <- randomForest(ramFo[,-4], ramFo[,4], ntree=12)
print(rf2)


ramFo.rf <- randomForest(ramFo[,-4], ramFo[,4], prox=TRUE)
ramFo.p <- classCenter(ramFo[,-4], ramFo[,4], ramFo.rf$prox)
plot(ramFo[,2], ramFo[,5], pch=21, xlab=names(ramFo)[2],
ylab=names(ramFo)[5],
bg=c("red", "blue", "green")[as.numeric(factor(ramFo$V4))],
main="Products Data with %%%")
points(ramFo.p[,2], ramFo.p[,5], pch=21, cex=2, bg=c("red", "blue", "green"))
```

5).Для алгоритма Boosting в функции gbm выбрать параметр N.trees = 400, параметр distribution = “laplace”, параметр bag.Fraction = 0.5

```
rm(list = ls())

library(gbm)
#library(ggplot2)

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


layout (matrix (c(1,2,3,4,5,6,7,8,9,10),2,5))


library(MASS)
X <- cbind(scale(ramFo2[,1:4]), matrix(rnorm(36*150), 150, 36))
Y <- ramFo2[,5]
ramFo2.evaluate <- function(indices) {
result = 1
if (sum(indices) > 2) {
```

```
huhn <- lda(X[,indices==1], Y, CV=TRUE)$posterior
result = sum(Y != dimnames(huhn)[[2]][apply(huhn, 1,
function(x)
which(x == max(x)))]) / length(Y)
}
result
}
monitor <- function(obj) {
minEval = min(obj$evaluations);
plot(obj, type="hist");
}
woppa <- rbga.bin(size=40, mutationChance=0.05, zeroToOneRatio=10,
evalFunc=ramFo2.evaluate, verbose=TRUE, monitorFunc=monitor, iter = 11)

ramFo2.mod <- gbm(ramFoT$V1 ~ ., distribution="multinomial",
data=ramFo2,
n.trees=2000, shrinkage=0.01, cv.folds=5,
verbose=FALSE, n.cores=1)
ramFo2.mod

gbm1 <- gbm(ramFoT$V1 ~ ., data = ramFo2,
distribution = "gaussian", n.trees = 100, shrinkage = 0.1,
 bag.fraction = 0.5, train.fraction = 0.5,
n.minobsinnode = 10, cv.folds = 5,
verbose = FALSE, n.cores = 1)
# Check performance using the out-of-bag (OOB) error; the OOB error typi-
cally
# underestimates the optimal number of iterations
best.iter <- gbm.perf(gbm1, method = "OOB")
print(best.iter)

# Check performance using the 50% heldout test set
best.iter <- gbm.perf(gbm1, method = "test")
print(best.iter)
# Check performance using 5-fold cross-validation
best.iter <- gbm.perf(gbm1, method = "cv")
print(best.iter)
# Plot relative influence of each variable
par(mfrow = c(1, 2))
summary(gbm1, n.trees = 1)          # using first tree
summary(gbm1, n.trees = best.iter)  # using estimated best number of trees
```

```
# Compactly print the first and last trees for curiosity
print(pretty.gbm.tree(gbm1, i.tree = 1))
print(pretty.gbm.tree(gbm1, i.tree = gbm1$n.trees))
plot(gbm1, i.var = 2:4, n.trees = best.iter)
```

Алгоритм RPART

```
rm(list = ls())

#library(genalg)
#library(ggplot2)

source = data.frame(read.table("logic0401.txt", header = TRUE, sep = ""))
names(source)[1] <- "code"
print("Исходные данные")
print(source)

bz = data.frame(read.table("bz.txt", header = TRUE, sep = ""))
names(bz)[1] <- "NN"
print("База конъюнкций")
print(bz)

trs = data.frame(read.table("trees.txt", header = TRUE, sep = ""))
print("Список Товаров")
print(trs)

tps = data.frame(read.table("topics.txt", header = TRUE, sep = ""))
print("Список Классов Классификации")
print(tps)



d1 = data.frame(read.table("rday1.txt", header = TRUE, sep = ""))
print("Список реализаций 1")
print(d1)

d2 = data.frame(read.table("rday2.txt", header = TRUE, sep = ""))
print("Список реализаций 2")
print(d2)
```

```
d3 = data.frame(read.table("rday3.txt", header = TRUE, sep = ""))
print("Список реализаций 3")
print(d3)

d4 = data.frame(read.table("rday4.txt", header = TRUE, sep = ""))
print("Список реализаций 4")
print(d4)


d5 = data.frame(read.table("rday5.txt", header = TRUE, sep = ""))
print("Список реализаций 5")
print(d5)


km1 = data.frame(read.table("rklasday1.txt", header = TRUE, sep = ""))
print("Разделение классов 1")
print(km1)

km2 = data.frame(read.table("rklasday2.txt", header = TRUE, sep = ""))
print("Разделение классов 2")
print(km2)

km3 = data.frame(read.table("rklasday3.txt", header = TRUE, sep = ""))
print("Разделение классов 3")
print(km3)

km4 = data.frame(read.table("rklasday4.txt", header = TRUE, sep = ""))
print("Разделение классов 4")
print(km4)

km5 = data.frame(read.table("rklasday5.txt", header = TRUE, sep = ""))
print("Разделение классов 5")
print(km5)


ranj <- matrix(0:0, nrow=50, ncol=30)

M=1
for (i in 1:5) {
  for (j in 1:10) {
   ranj[M,1] =trs$NN[j]
```

```
   M=M+1
  }
}


M=1
  for (j in 1:10) {
   ranj[M,2] =source$PR1[j]
   ranj[M,3] =1
    M=M+1
  }
M=11
  for (j in 1:10) {
   ranj[M,2] =source$PR2[j]
   ranj[M,3] =2
    M=M+1
  }
M=21
  for (j in 1:10) {
   ranj[M,2] =source$PR3[j]
   ranj[M,3] =3
    M=M+1
  }

M=31
  for (j in 1:10) {
   ranj[M,2] =source$PR4[j]
   ranj[M,3] =4
    M=M+1
  }

M=41
  for (j in 1:10) {
   ranj[M,2] =source$PR5[j]
   ranj[M,3] =5
    M=M+1
  }



for (K in 1:4) {
```

```
if (km1$V1[K] > 0) {
    ranj[1,4] = K
    ranj[1,29]= tps$ST[K]
    ranj[1,30]= tps$ED[K]
          }
 if (km1$V2[K] > 0)  {
    ranj[2,29]= tps$ST[K]
    ranj[2,30]= tps$ED[K]
    ranj[2,4] = K  }
 if (km1$V3[K] > 0)  {
    ranj[3,29]= tps$ST[K]
    ranj[3,30]= tps$ED[K]

     ranj[3,4] = K  }
 if (km1$V4[K] > 0)  {
    ranj[4,29]= tps$ST[K]
    ranj[4,30]= tps$ED[K]

     ranj[4,4] = K  }
 if (km1$V5[K] > 0)  {
    ranj[5,29]= tps$ST[K]
    ranj[5,30]= tps$ED[K]

     ranj[5,4] = K  }
 if (km1$V6[K] > 0)  {
    ranj[6,29]= tps$ST[K]
    ranj[6,30]= tps$ED[K]

     ranj[6,4] = K  }
 if (km1$V7[K] > 0)  {
    ranj[7,29]= tps$ST[K]
    ranj[7,30]= tps$ED[K]

     ranj[7,4] = K  }
 if (km1$V8[K] > 0)  {
    ranj[8,29]= tps$ST[K]
    ranj[8,30]= tps$ED[K]

     ranj[8,4] = K  }
 if (km1$V9[K] > 0)  {
    ranj[9,29]= tps$ST[K]
```

```
        ranj[9,30]= tps$ED[K]

         ranj[9,4] = K  }
    if (km1$V10[K] > 0)  {
        ranj[10,29]= tps$ST[K]
        ranj[10,30]= tps$ED[K]

         ranj[10,4] = K  }
     }

for (K in 1:4) {
  if (km2$V1[K] > 0) {
        ranj[11,29]= tps$ST[K]
        ranj[11,30]= tps$ED[K]

        ranj[11,4] = K   }
    if (km2$V2[K] > 0)  {
        ranj[12,29]= tps$ST[K]
        ranj[12,30]= tps$ED[K]

         ranj[12,4] = K  }
    if (km2$V3[K] > 0)  {
        ranj[13,29]= tps$ST[K]
        ranj[13,30]= tps$ED[K]

         ranj[13,4] = K  }
    if (km2$V4[K] > 0)  {
        ranj[14,29]= tps$ST[K]
        ranj[14,30]= tps$ED[K]

         ranj[14,4] = K  }
    if (km2$V5[K] > 0)  {
        ranj[15,29]= tps$ST[K]
        ranj[15,30]= tps$ED[K]

         ranj[15,4] = K  }
    if (km2$V6[K] > 0)  {
        ranj[16,29]= tps$ST[K]
        ranj[16,30]= tps$ED[K]

         ranj[16,4] = K  }
```

```
    if (km2$V7[K] > 0)  {
       ranj[17,29]= tps$ST[K]
       ranj[17,30]= tps$ED[K]

        ranj[17,4] = K  }
    if (km2$V8[K] > 0)  {
       ranj[18,29]= tps$ST[K]
       ranj[18,30]= tps$ED[K]

        ranj[18,4] = K  }
    if (km2$V9[K] > 0)  {
       ranj[19,29]= tps$ST[K]
       ranj[19,30]= tps$ED[K]

        ranj[19,4] = K  }
    if (km2$V10[K] > 0)  {
       ranj[20,29]= tps$ST[K]
       ranj[20,30]= tps$ED[K]

        ranj[20,4] = K  }
     }

for (K in 1:4) {
  if (km3$V1[K] > 0) {
       ranj[21,29]= tps$ST[K]
       ranj[21,30]= tps$ED[K]

       ranj[21,4] = K   }
    if (km3$V2[K] > 0)  {
       ranj[22,29]= tps$ST[K]
       ranj[22,30]= tps$ED[K]

        ranj[22,4] = K  }
    if (km3$V3[K] > 0)  {
       ranj[23,29]= tps$ST[K]
       ranj[23,30]= tps$ED[K]

        ranj[23,4] = K  }
    if (km3$V4[K] > 0)  {
       ranj[24,29]= tps$ST[K]
       ranj[24,30]= tps$ED[K]
```

```
        ranj[24,4] = K  }
    if (km3$V5[K] > 0)  {
       ranj[25,29]= tps$ST[K]
       ranj[25,30]= tps$ED[K]

        ranj[25,4] = K  }
    if (km3$V6[K] > 0)  {
       ranj[26,29]= tps$ST[K]
       ranj[26,30]= tps$ED[K]

        ranj[26,4] = K  }
    if (km3$V7[K] > 0)  {
       ranj[27,29]= tps$ST[K]
       ranj[27,30]= tps$ED[K]

        ranj[27,4] = K  }
    if (km3$V8[K] > 0)  {
       ranj[28,29]= tps$ST[K]
       ranj[28,30]= tps$ED[K]

        ranj[28,4] = K  }
    if (km3$V9[K] > 0)  {
       ranj[29,29]= tps$ST[K]
       ranj[29,30]= tps$ED[K]

        ranj[29,4] = K  }
    if (km3$V10[K] > 0)  {
       ranj[30,29]= tps$ST[K]
       ranj[30,30]= tps$ED[K]

        ranj[30,4] = K  }
     }


for (K in 1:4) {
  if (km4$V1[K] > 0) {
       ranj[31,29]= tps$ST[K]
       ranj[31,30]= tps$ED[K]

       ranj[31,4] = K   }
```

```
if (km4$V2[K] > 0)  {
   ranj[32,29]= tps$ST[K]
   ranj[32,30]= tps$ED[K]

    ranj[32,4] = K  }
if (km4$V3[K] > 0)  {
   ranj[33,29]= tps$ST[K]
   ranj[33,30]= tps$ED[K]

    ranj[33,4] = K  }
if (km4$V4[K] > 0)  {
   ranj[34,29]= tps$ST[K]
   ranj[34,30]= tps$ED[K]

    ranj[34,4] = K  }
if (km4$V5[K] > 0)  {
   ranj[35,29]= tps$ST[K]
   ranj[35,30]= tps$ED[K]

    ranj[35,4] = K  }
if (km4$V6[K] > 0)  {
   ranj[36,29]= tps$ST[K]
   ranj[36,30]= tps$ED[K]

    ranj[36,4] = K  }
if (km4$V7[K] > 0)  {
   ranj[37,29]= tps$ST[K]
   ranj[37,30]= tps$ED[K]

    ranj[37,4] = K  }
if (km4$V8[K] > 0)  {
   ranj[38,29]= tps$ST[K]
   ranj[38,30]= tps$ED[K]

    ranj[38,4] = K  }
if (km4$V9[K] > 0)  {
   ranj[39,29]= tps$ST[K]
   ranj[39,30]= tps$ED[K]

    ranj[39,4] = K  }
if (km4$V10[K] > 0)  {
```

```
        ranj[40,29]= tps$ST[K]
        ranj[40,30]= tps$ED[K]

         ranj[40,4] = K  }
      }

for (K in 1:4) {
   if (km5$V1[K] > 0) {
        ranj[41,29]= tps$ST[K]
        ranj[41,30]= tps$ED[K]

        ranj[41,4] = K   }
    if (km5$V2[K] > 0)  {
        ranj[42,29]= tps$ST[K]
        ranj[42,30]= tps$ED[K]

         ranj[42,4] = K  }
    if (km5$V3[K] > 0)  {
        ranj[43,29]= tps$ST[K]
        ranj[43,30]= tps$ED[K]

         ranj[43,4] = K  }
    if (km5$V4[K] > 0)  {
        ranj[44,29]= tps$ST[K]
        ranj[44,30]= tps$ED[K]

         ranj[44,4] = K  }
    if (km5$V5[K] > 0)  {
        ranj[45,29]= tps$ST[K]
        ranj[45,30]= tps$ED[K]

         ranj[45,4] = K  }
    if (km5$V6[K] > 0)  {
        ranj[46,29]= tps$ST[K]
        ranj[46,30]= tps$ED[K]

         ranj[46,4] = K  }
    if (km5$V7[K] > 0)  {
        ranj[47,29]= tps$ST[K]
        ranj[47,30]= tps$ED[K]
```

```
        ranj[47,4] = K  }
    if (km5$V8[K] > 0)  {
       ranj[48,29]= tps$ST[K]
       ranj[48,30]= tps$ED[K]

        ranj[48,4] = K  }
    if (km5$V9[K] > 0)  {
       ranj[49,29]= tps$ST[K]
       ranj[49,30]= tps$ED[K]

        ranj[49,4] = K  }
    if (km5$V10[K] > 0)  {
       ranj[50,29]= tps$ST[K]
       ranj[50,30]= tps$ED[K]

        ranj[50,4] = K  }
     }

N=5
for (i in 1:24) {
   ranj[1,N] = d1$V1[i]
   ranj[2,N] = d1$V2[i]
   ranj[3,N] = d1$V3[i]
   ranj[4,N] = d1$V4[i]
   ranj[5,N] = d1$V5[i]
   ranj[6,N] = d1$V6[i]
   ranj[7,N] = d1$V7[i]
   ranj[8,N] = d1$V8[i]
   ranj[9,N] = d1$V9[i]
   ranj[10,N] = d1$V10[i]
    N=N+1
}

N=5
for (i in 1:24) {
   ranj[11,N] = d2$V1[i]
   ranj[12,N] = d2$V2[i]
   ranj[13,N] = d2$V3[i]
   ranj[14,N] = d2$V4[i]
   ranj[15,N] = d2$V5[i]
   ranj[16,N] = d2$V6[i]
```

```
  ranj[17,N] = d2$V7[i]
  ranj[18,N] = d2$V8[i]
  ranj[19,N] = d2$V9[i]
  ranj[20,N] = d2$V10[i]
   N=N+1
}

N=5
for (i in 1:24) {
  ranj[21,N] = d3$V1[i]
  ranj[22,N] = d3$V2[i]
  ranj[23,N] = d3$V3[i]
  ranj[24,N] = d3$V4[i]
  ranj[25,N] = d3$V5[i]
  ranj[26,N] = d3$V6[i]
  ranj[27,N] = d3$V7[i]
  ranj[28,N] = d3$V8[i]
  ranj[29,N] = d3$V9[i]
  ranj[30,N] = d3$V10[i]
   N=N+1
}

N=5
for (i in 1:24) {
  ranj[31,N] = d4$V1[i]
  ranj[32,N] = d4$V2[i]
  ranj[33,N] = d4$V3[i]
  ranj[34,N] = d4$V4[i]
  ranj[35,N] = d4$V5[i]
  ranj[36,N] = d4$V6[i]
  ranj[37,N] = d4$V7[i]
  ranj[38,N] = d4$V8[i]
  ranj[39,N] = d4$V9[i]
  ranj[40,N] = d4$V10[i]
   N=N+1
}

N=5
for (i in 1:24) {
  ranj[41,N] = d5$V1[i]
  ranj[42,N] = d5$V2[i]
```

```
   ranj[43,N] = d5$V3[i]
   ranj[44,N] = d5$V4[i]
   ranj[45,N] = d5$V5[i]
   ranj[46,N] = d5$V6[i]
   ranj[47,N] = d5$V7[i]
   ranj[48,N] = d5$V8[i]
   ranj[49,N] = d5$V9[i]
   ranj[50,N] = d5$V10[i]
    N=N+1
}

ranj

write.table(ranj, file="IDCarttrees.txt")


sm = data.frame(read.table("summs.txt", header = TRUE, sep = ""))
print("Качество Конъюнкций:")
print(sm)

mxi <- matrix(0:0, nrow=4, ncol=1)
k1=1
for (i in 1:24) {
  if (sm$V1[i] > 45) {
      mxi[k1] = i+4
      k1=k1+1
        }
}
mxi

frst = data.frame(read.table("IDCarttrees.txt", header = TRUE, sep = ""))

n1=0
n2=0
n3=0
n4=0
n5=0
 gu <- matrix(0:0, nrow=47, ncol=6)


for (i in 1:50) {
```

```
  if (frst$V7[i] == 1) {
     n2=1
       }
  if (frst$V20[i] == 1) {
      n3=1
       }
  if (frst$V23[i] == 1) {
      n4=1
       }
  if (frst$V26[i] == 1) {
      n5=1
       }

  if (n2+n3+n4+n5 >= 3 )  {
      n1=n1+1
      gu[n1,1] = frst$V1[i]
      gu[n1,2] = frst$V2[i]
      gu[n1,3] = frst$V3[i]
      gu[n1,4] = frst$V4[i]
      gu[n1,5] = frst$V29[i]
      gu[n1,6] = frst$V30[i]
          }

     n2=0
     n3=0
     n4=0
     n5=0
   }

gu

write.table(gu, file="clever.txt")

car47 = data.frame(read.table("cleverK.txt", header = TRUE, sep = ""))
car47

#fit <- rpart(V4 ~ V2 + V5 + V6, car47, method =  "poisson")
fit <- rpart(V4 ~ V2 + V5 + V6, car47)

par(xpd = TRUE)
plot(fit, compress = TRUE)
```

text(fit, use.n = TRUE)

6). Результаты визуализировать и сравнить.

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 Min-Num= 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.

Вариант 3

Разработать логический классификатор с использованием алгоритмов **«Кора», «ID3», «CART», «Бэггинг», «Бустинг», генетического алго-**

ритма 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	
021	1							0.7	0.54	1	1	0.45	
022	0.77											1	
023													
024													
025													
026													
027													
028												0.15	
029	1		1				1			1		0.23	
030	0.58												

1).Для алгоритма Кора: выбрать частоту встречаемости конъюнкций Min-Num= 5

2).Для генетического алгоритма выбрать: генную бинарную комбинацию **1101000**

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

4).Для алгоритма Bagging в функции RandomForest выбрать параметр N.trees= 600, в функции train выбрать параметр method = "Treebag"

5).Для алгоритма Boosting в функции gbm выбрать параметр N.trees = 460, параметр distribution = “gussian”, параметр bag.Fraction = 0.65

6). Результаты визуализировать и сравнить.

Вариант 4

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	
031	1							0.7	0.54	1	1	0.45	
032	0.77											1	
033													
034													
035													
036													

037													
038												0.15	
039	1		1				1			1		0.23	
040	0.58												

1).Для алгоритма Кора: выбрать частоту встречаемости конъюнкций MinNum= 6

2).Для генетического алгоритма выбрать: генную бинарную комбинацию **1011001**

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

4).Для алгоритма Bagging в функции RandomForest выбрать параметр N.trees= 280, в функции train выбрать параметр method = "Logicbag"

5).Для алгоритма Boosting в функции gbm выбрать параметр N.trees = 380, параметр distribution = “adaBoost”, параметр bag.Fraction = 0.39

6). Результаты визуализировать и сравнить.

Вариант 5

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	
041	1							0.7	0.54	1	1	0.45	
042	0.77											1	
043													
044													
045													
046													
047													
048												0.15	
049	1		1				1			1		0.23	
050	0.58												

1).Для алгоритма Кора: выбрать частоту встречаемости конъюнкций MinNum= 7

2).Для генетического алгоритма выбрать: генную бинарную комбинацию **1011101**

3).Для алгоритма CART в функции rpart выбрать параметр method = "poisson"
4).Для алгоритма Bagging в функции RandomForest выбрать параметр N.trees= 580, в функции train выбрать параметр method = "bagEarth"
5).Для алгоритма Boosting в функции gbm выбрать параметр N.trees = 630, параметр distribution = “coxPH”, параметр bag.Fraction = 0.88
6). Результаты визуализировать и сравнить.

Вариант 6

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	
013	0.77											1	
015													
017													
019													
021													
023													
025												0.15	
027	1		1				1			1		0.23	
029	0.58												

1).Для алгоритма Кора: выбрать частоту встречаемости конъюнкций Min-Num= 5
2).Для генетического алгоритма выбрать: генную бинарную комбинацию 0**011100**
3).Для алгоритма CART в функции rpart выбрать параметр method = "anova"
4).Для алгоритма Bagging в функции RandomForest выбрать параметр N.trees= 430, в функции train выбрать параметр method = "Adabag"
5).Для алгоритма Boosting в функции gbm выбрать параметр N.trees = 720, параметр distribution = “poisson”, параметр bag.Fraction = 0.72
6). Результаты визуализировать и сравнить.

Вариант 7

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	
020	1							0.7	0.54	1	1	0.45	
025	0.77											1	
030													
035													
040													
045													
050													
001												0.15	
003	1		1				1			1		0.23	
011	0.58												

1).Для алгоритма Кора: выбрать частоту встречаемости конъюнкций MinNum= 8

2).Для генетического алгоритма выбрать: генную бинарную комбинацию 1**010101**

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

4).Для алгоритма Bagging в функции RandomForest выбрать параметр N.trees= 700, в функции train выбрать параметр method = "Treebag"

5).Для алгоритма Boosting в функции gbm выбрать параметр N.trees = 340, параметр distribution = “bernaully”, параметр bag.Fraction = 0.81

6). Результаты визуализировать и сравнить.

Вариант 8

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	
020	1							0.7	0.54	1	1	0.45	
025	0.77											1	
030													
035													
040													
045													
050													
001												0.15	
003	1		1				1			1		0.23	
011	0.58												

1).Для алгоритма Кора: выбрать частоту встречаемости конъюнкций Min-Num= 5

2).Для генетического алгоритма выбрать: генную бинарную комбинацию **11**10111

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

4).Для алгоритма Bagging в функции RandomForest выбрать параметр N.trees= 730, в функции train выбрать параметр method = "adabag"

5).Для алгоритма Boosting в функции gbm выбрать параметр N.trees = 590, параметр distribution = “laplace”, параметр bag.Fraction = 0.48

6). Результаты визуализировать и сравнить.

Вариант 9

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	
020	1							0.7	0.54	1	1	0.45	
025	0.77											1	
030													
035													
040													
045													

050													
001												0.15	
003	1		1				1			1		0.23	
011	0.58												

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

2).Для генетического алгоритма выбрать: генную бинарную комбинацию 0**011101**

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

4).Для алгоритма Bagging в функции RandomForest выбрать параметр N.trees= 620, в функции train выбрать параметр method = "Treebag"

5).Для алгоритма Boosting в функции gbm выбрать параметр N.trees = 540, параметр distribution = “adaboost”, параметр bag.Fraction = 0.79

6). Результаты визуализировать и сравнить.

Вариант 10

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	
020	1							0.7	0.54	1	1	0.45	
025	0.77											1	
030													
035													
040													
045													
050													
001												0.15	
003	1		1				1			1		0.23	
011	0.58												

1).Для алгоритма Кора: выбрать частоту встречаемости конъюнкций MinNum= 8

2).Для генетического алгоритма выбрать: генную бинарную комбинацию 11**10101**

3).Для алгоритма CART в функции rpart выбрать параметр method = "poisson"
4).Для алгоритма Bagging в функции RandomForest выбрать параметр N.trees= 300, в функции train выбрать параметр method = "adabag"
5).Для алгоритма Boosting в функции gbm выбрать параметр N.trees = 380, параметр distribution = “bernaully”, параметр bag.Fraction = 0.55
6). Результаты визуализировать и сравнить.

Вариант 11

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	
020	1							0.7	0.54	1	1	0.45	
025	0.77											1	
030													
035													
040													
045													
050													
001												0.15	
003	1		1				1			1		0.23	
011	0.58												

1).Для алгоритма Кора: выбрать частоту встречаемости конъюнкций Min-Num= 8
2).Для генетического алгоритма выбрать: генную бинарную комбинацию 0**010100**
3).Для алгоритма CART в функции rpart выбрать параметр method = "anova"
4).Для алгоритма Bagging в функции RandomForest выбрать параметр N.trees= 780, в функции train выбрать параметр method = "Treebag"
5).Для алгоритма Boosting в функции gbm выбрать параметр N.trees = 340, параметр distribution = “bernaully”, параметр bag.Fraction = 0.81
6). Результаты визуализировать и сравнить.

Вариант 12

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	
020	1							0.7	0.54	1	1	0.45	
025	0.77											1	
030													
035	1												
040													
045													
050	1	0.73											
001												0.15	
003	1		1				1			1		0.23	
011	0.58												

1).Для алгоритма Кора: выбрать частоту встречаемости конъюнкций MinNum= 8

2).Для генетического алгоритма выбрать: генную бинарную комбинацию 11**10100**

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

4).Для алгоритма Bagging в функции RandomForest выбрать параметр N.trees= 880, в функции train выбрать параметр method = "Treebag"

5).Для алгоритма Boosting в функции gbm выбрать параметр N.trees = 340, параметр distribution = “bernaully”, параметр bag.Fraction = 0.81

6). Результаты визуализировать и сравнить.

Вариант 13

Разработать логический классификатор с использованием алгоритмов **«Кора», «ID3», «CART», «Бэггинг», «Бустинг», генетического алго-**

ритма 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	
020	1							0.7	0.54	1	1	0.45	
025	0.77											1	
030													
035													
040													
045													
050													
001												0.15	
003	1		1				1			1		0.23	
011	0.58												

1).Для алгоритма Кора: выбрать частоту встречаемости конъюнкций Min-Num= 5

2).Для генетического алгоритма выбрать: генную бинарную комбинацию 010**0101**

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

4).Для алгоритма Bagging в функции RandomForest выбрать параметр N.trees= 710, в функции train выбрать параметр method = "Treebag"

5).Для алгоритма Boosting в функции gbm выбрать параметр N.trees = 340, параметр distribution = “coxph”, параметр bag.Fraction = 0.61

6). Результаты визуализировать и сравнить.

Вариант 14

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	
020	1							0.7	0.54	1	1	0.45	
025	0.77											1	
030													
035													

040													
045													
050													
001												0.15	
003	1		1				1			1		0.23	
011	0.58												

1).Для алгоритма Кора: выбрать частоту встречаемости конъюнкций MinNum= 7

2).Для генетического алгоритма выбрать: генную бинарную комбинацию 1**010111**

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

4).Для алгоритма Bagging в функции RandomForest выбрать параметр N.trees= 624, в функции train выбрать параметр method = "bagFDA"

5).Для алгоритма Boosting в функции gbm выбрать параметр N.trees = 340, параметр distribution = “bernoully”, параметр bag.Fraction = 0.441

6). Результаты визуализировать и сравнить.

Вариант 15

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	
050	1	1	0.9	0.7				0.7	0.54	1	1	0.45	
030	0.77											1	
034		0.1											
021													
014	1		1			1			1			1	
046													
036													
049												0.15	
009	1		1				1			1		0.23	
014	0.58												

1).Для алгоритма Кора: выбрать частоту встречаемости конъюнкций MinNum= 8

2).Для генетического алгоритма выбрать: генную бинарную комбинацию 11**10001**

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

4).Для алгоритма Bagging в функции RandomForest выбрать параметр N.trees= 200, в функции train выбрать параметр method = "bagEarth"

5).Для алгоритма Boosting в функции gbm выбрать параметр N.trees = 510, параметр distribution = “laplace”, параметр bag.Fraction = 0.49

6). Результаты визуализировать и сравнить.