Отчет по лабораторной работе
Деревья решений

19 января 2024 г.

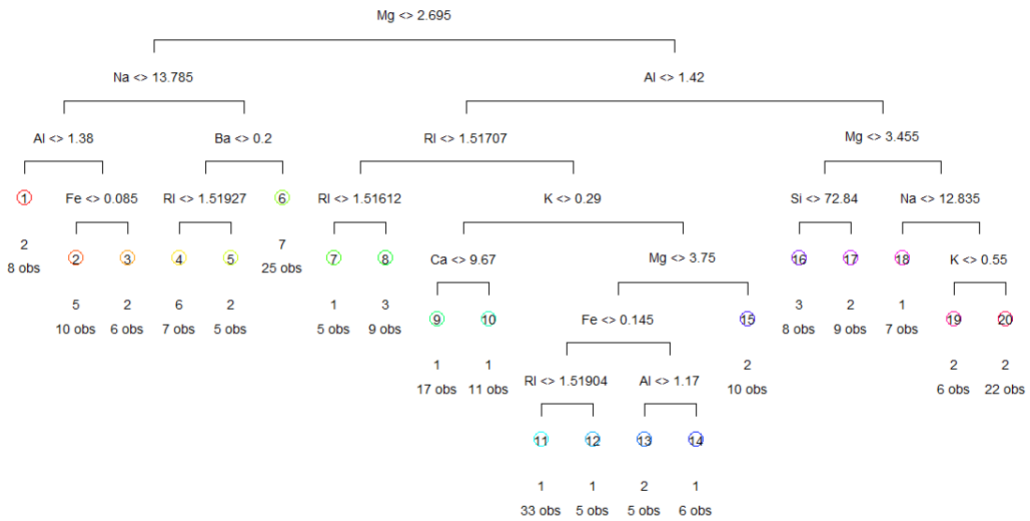
Шаронов Артем
Группа: 5140201/30302

Задание №1

1) Загрузите набор данных Glass из пакета “mlbench”. Набор данных (признаки, классы) был изучен в работе «Метод ближайших соседей». Постройте дерево классификации для модели, задаваемой следующей формулой: Type ., дайте интерпретацию полученным результатам. При рисовании дерева используйте параметр cex=0.7 для уменьшения размера текста на рисунке, например, text(bc.tr,cex=0.7) или draw.tree(bc.tr,cex=0.7). Является ли построенное дерево избыточным? Выполните все операции оптимизации дерева.

```
# Код на R
# загрузка необходимых пакетов

# Загрузка данных Glass из пакета mlbench
data(Glass)

# Построение дерева классификации
bc.tree <- tree(Type~., Glass)
draw.tree(bc.tree, cex = 0.7)

# Оптимизация дерева
bc.tree <- prune.tree(bc.tree, best = 7)
draw.tree(bc.tree, cex = 0.7)
```

Отрисовка дерева классификации:

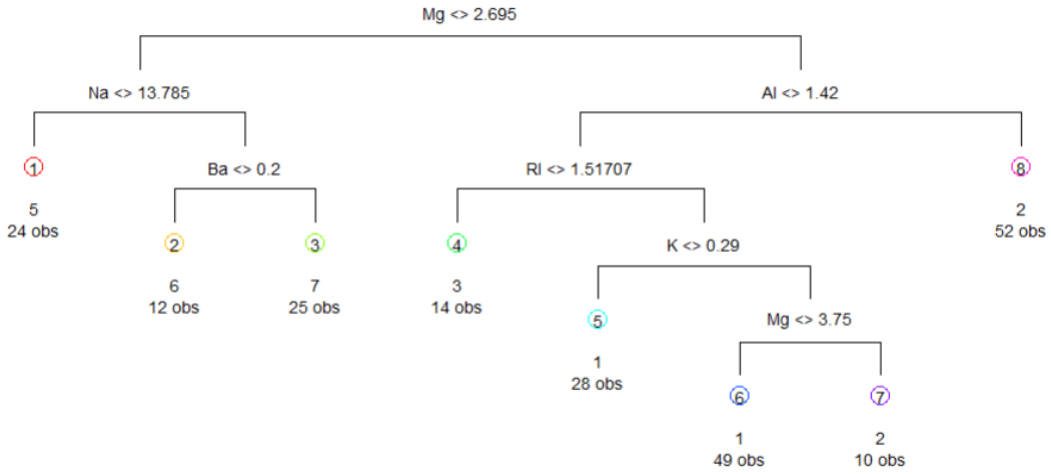


Наиболее значимый признак это количество Mg в образце. Дерево является избыточным, так как оно содержит листья выходящие из одного узла и имеющие один и

тот же класс.

Оптимизация дерева происходила по числу равному количеству классов в наборе данных (7).

Отрисовка оптимизированного дерева классификации:

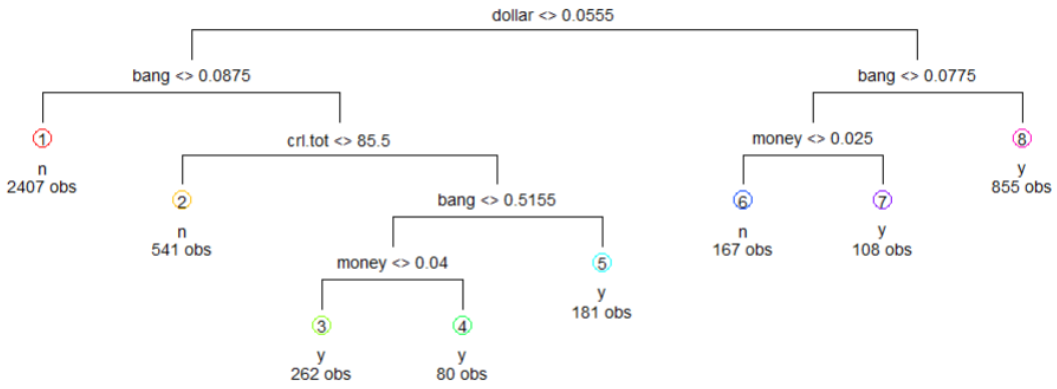


Задание №2

Загрузите набор данных spam7 из пакета DAAG. Постройте дерево классификации для модели, задаваемой следующей формулой: yesno ~ ., дайте интерпретацию полученным результатам. Запустите процедуру “cost-complexity prunning” с выбором параметра k по умолчанию, method = ’misclass’, выведите полученную последовательность деревьев. Какое из полученных деревьев, на Ваш взгляд, является оптимальным? Объясните свой выбор.

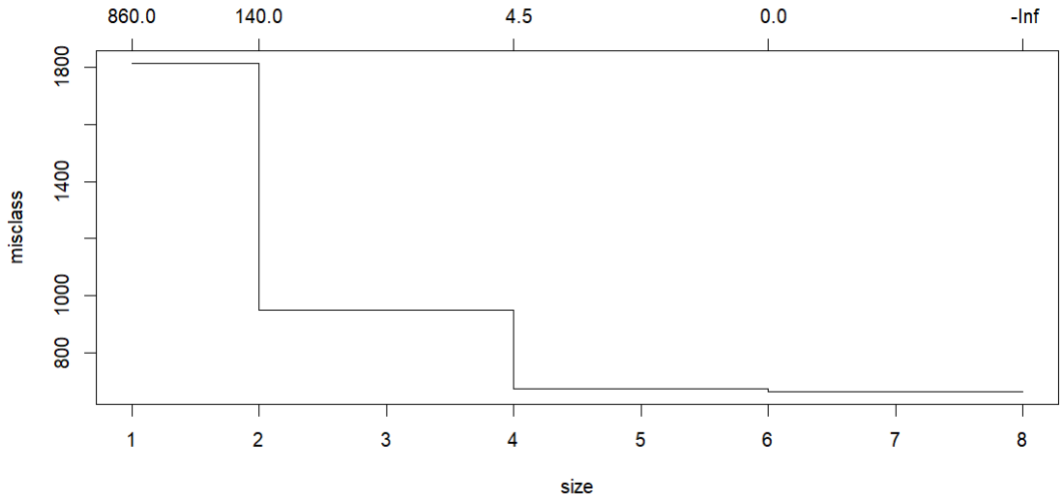
```
# Код на R
# загрузка необходимых пакетов

# Загрузка данных spam7 из пакета DAAG
data(spam7)

# Построение дерева классификации
sp7.tree <- tree(yesno~., spam7)
draw.tree(sp7.tree, cex = 0.7)

# Построение графика влияния k на ошибочность классификации
misc <- prune.misclass(spam7.tree)
plot(misc)

# Построение вариантов оптимизации дерева
sp7.pruned <- prune.tree(sp7.tree, method = "misclass", k=0)
draw.tree(sp7.pruned, cex = 0.7)
sp7.pruned <- prune.tree(sp7.tree, method = "misclass", k=4.5)
draw.tree(sp7.pruned, cex = 0.7)
sp7.pruned <- prune.tree(sp7.tree, method = "misclass", k=137.5)
draw.tree(sp7.pruned, cex = 0.7)
```

Отрисовка дерева классификации:



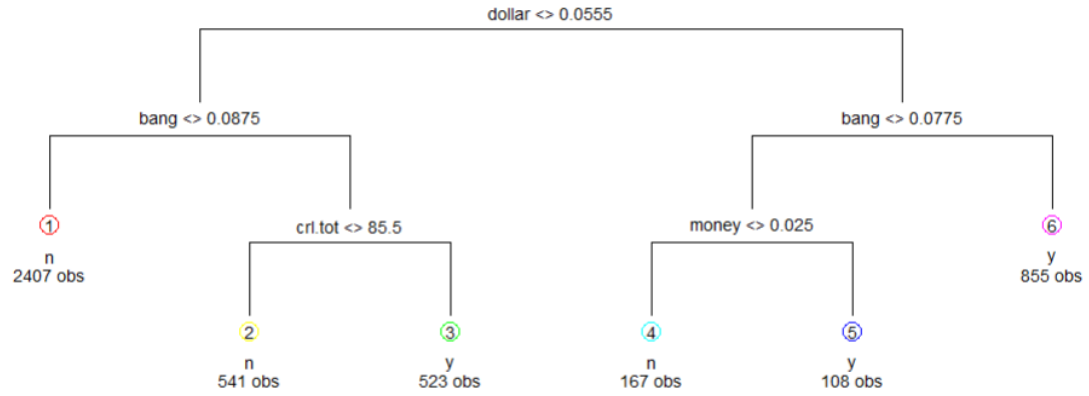
Наиболее значимый признак это dollar. Дерево является избыточным, так как оно содержит листья выходящие из одного узла и имеющие один и тот же класс.

Построение графика влияния k на ошибочность классификации:

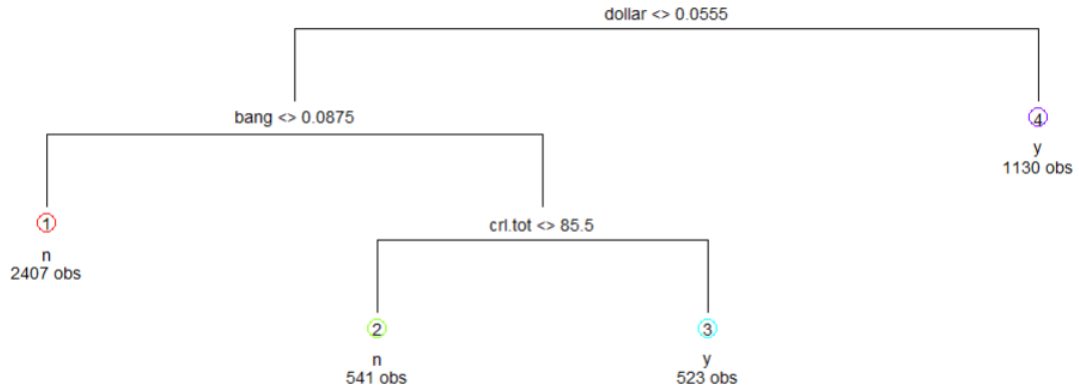


k -> -Inf 0.0 4.5 137.5 864.0

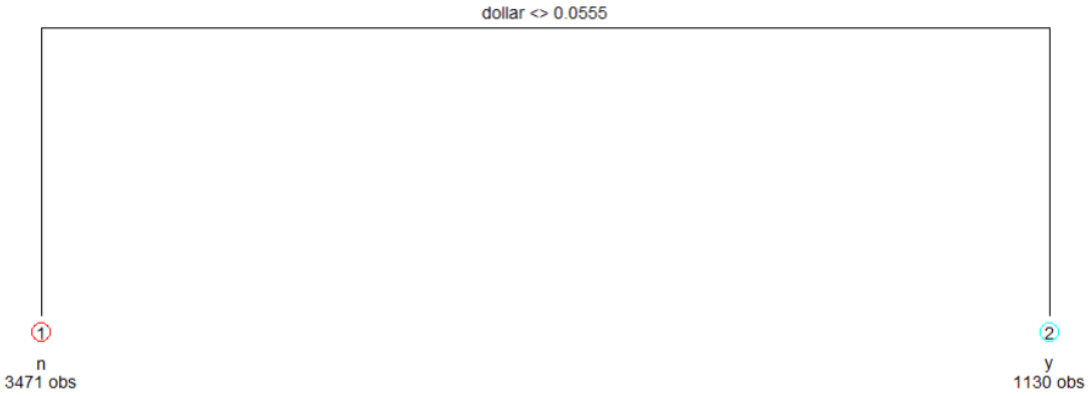
Отрисовка оптимизированных деревьев классификации:



k = [0; 4.5)



$$k = [4.5;\ 137.5)$$



$$k = [137.5;\ 864)$$

В изначальном дереве процент ошибочной классификации составлял 18% при 8 листьях. Оптимальное дерево при $k = [4.5;\ 137.5)$, так как процент ошибочной классификации составляет 18.3% при сокращении количества листьев с 8 до 4.

Задание №3

Загрузите набор данных nsw74psid1 из пакета DAAG. Постройте регрессионное дерево для модели, задаваемой следующей формулой: re78 .. Постройте регрессионную модель и SVMрегрессию для данной формулы. Сравните качество построенных моделей, выберите оптимальную модель и объясните свой выбор.

```r
# Код на R
# загрузка необходимых пакетов
library(DAAG)
library(tree)
library(e1071)
library(Metrics)

# Загрузка данных nsw74psid1 из пакета DAAG
data(nsw74psid1)
nsw <- nsw74psid1

# Построение регрессионного дерева
tree_model <- tree(re78 ~ ., data = nsw)

# Построение регрессионной модели
regress_model <- lm(re78 ~ ., data = nsw)

# Построение SVM-регрессии
svm_model <- svm(re78 ~ ., data = nsw)

# Сравнение моделей
tree_predict <- predict(tree_model)
regress_model_predict <- predict(regress_model)
svm_model_predict <- predict(svm_model)

# Среднеквадратичная ошибка регрессионного дерева
tree_error <- rmse(nsw$re78, tree_predict)
tree_error

# Среднеквадратичная ошибка регрессионной модели
regress_model_error <- rmse(nsw$re78, regress_model_predict)
regress_model_error

# Среднеквадратичная ошибка SVM-регрессии
svm_model_error <- rmse(nsw$re78, svm_model_predict)
svm_model_error
```

Ошибки :

tree_error = 10316.04

regress_model_error = 10051.88

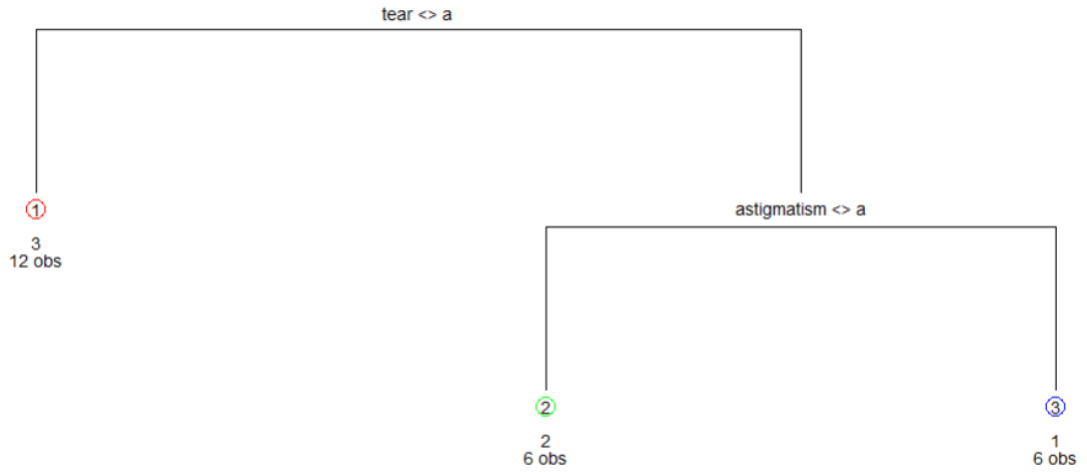
svm_model_error = 9813.287

Оптимальной моделью будет SVMрегрессионная с наименьшей среднеквадратичная ошибкой.

Задание №4

Загрузите набор данных Lenses Data Set из файла Lenses.txt: 3 класса (последний столбец): 1 : пациенту следует носить жесткие контактные линзы, 2 : пациенту следует носить мягкие контактные линзы, 3 : пациенту не следует носить контактные линзы. Признаки (категориальные): 1. возраст пациента: (1) молодой, (2) предстарческая дальнозоркость, (3) старческая дальнозоркость 2. состояние зрения: (1) близорукий, (2) дальнозоркий 3. астигматизм: (1) нет, (2) да 4. состояние слезы: (1) сокращенная, (2) нормальная Постройте дерево решений. Какие линзы надо носить при предстарческой дальнозоркости, близорукости, при наличии астигматизма и сокращенной слезы?

```
# Код на R
# Загрузка данных Lenses Data Set из файла Lenses.txt:
lenses <- read.table("C:/Users/Артем/Documents/R-script/Lenses.txt",␣
 ↪header=FALSE)
lenses <- lenses[, -1]

# Присвоение имен столбцам
names(lenses) <- c('age', 'vision', 'astigmatism', 'tear', 'lense')

# Преобразование категориальных признаков в факторы
lenses$age <- as.factor(lenses$age)
lenses$vision <- as.factor(lenses$vision)
lenses$astigmatism <- as.factor(lenses$astigmatism)
lenses$tear <- as.factor(lenses$tear)
lenses$lense <- as.factor(lenses$lense)

# Построение дерева решений
lens.tree <- tree(lenses$lense~., data = lenses, method = "class")
draw.tree(lens.tree, cex = 0.7)

# Прогнозирование класса для нового объекта
example <- data.frame(age = factor(2), vision = factor(1), astigmatism =␣
 ↪factor(2), tear = factor(1))
predict(lens.tree, example, type = "class")
```



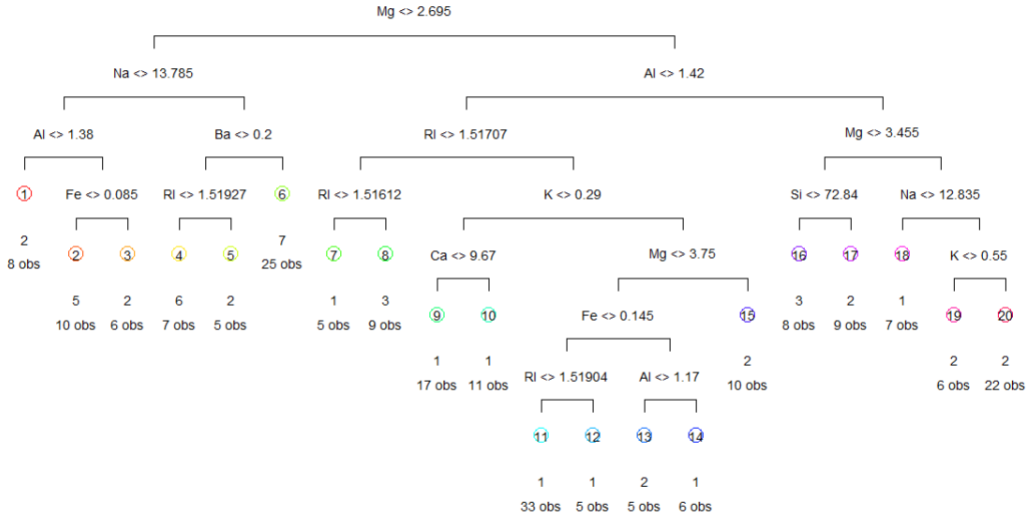
На основе этого дерева было определено, что новый объект относиться к 3 классу - пациенту не следует носить контактные линзы.

Задание №5

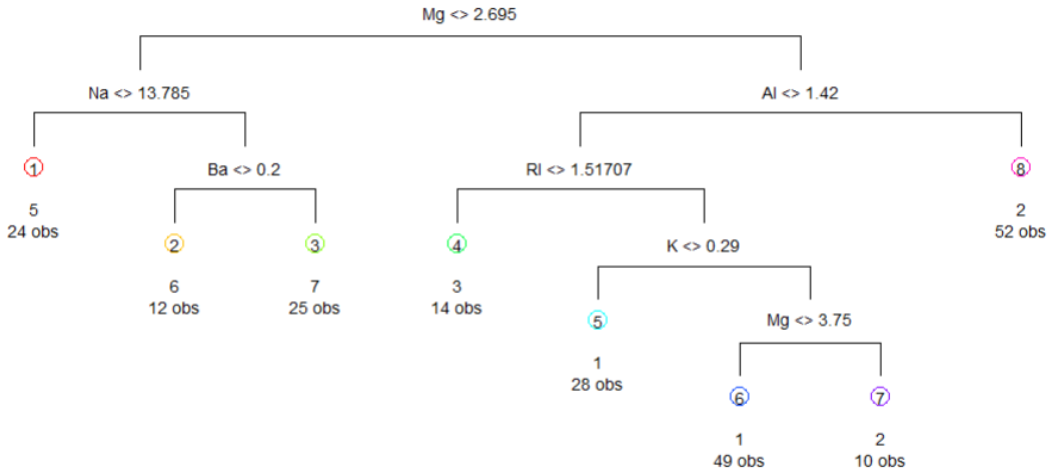
Постройте дерево решений для обучающего множества Glass, данные которого характеризуются 10-ю признаками: 1. Id number: 1 to 214; 2. RI: показатель преломления; 3. Na: сода (процент содержания в соотвествующем оксиде); 4. Mg; 5. Al; 6. Si; 7. K; 8. Ca; 9. Ba; 10. Fe. Классы характеризуют тип стекла: (1) окна зданий, плавильная обработка (2) окна зданий, не плавильная обработка (3) автомобильные окна, плавильная обработка (4) автомобильные окна, не плавильная обработка (нет в базе) (5) контейнеры (6) посуда (7) фары Посмотрите заголовки признаков и классов. Перед построением классификатора необходимо также удалить первый признак Id number, который не несет никакой информационной нагрузки. Это выполняется командой glass <- glass[,-1]. Определите, к какому типу стекла относится экземпляр с характеристиками RI =1.516 Na =11.7 Mg =1.01 Al =1.19 Si =72.59 K=0.43 Ca =11.44 Ba =0.02 Fe =0.1

```
# Код на R
# Загрузка данных Glass из пакета mlbench
data(Glass)
Glass$Type <- as.factor(Glass$Type)

# Построение дерева классификации
bc.tree <- tree(Type ~ ., data = Glass)
draw.tree(bc.tree, cex = 0.7)

# Построение оптимизированного дерева
bc_opti.tree <- prune.tree(bc.tree, best = 7)
draw.tree(bc_opti.tree, cex = 0.7)

# Прогнозирование класса для нового экземпляра
example <- data.frame(RI = 1.516, Na = 11.7, Mg = 1.01, Al = 1.19, Si =␣
 ↪72.59, K = 0.43, Ca = 11.44, Ba = 0.02, Fe = 0.1)
# По начальному дереву
predict(bc.tree, example, type = "class")
# По оптимизированному дереву
predict(bc_opti.tree, example, type = "class")
```

Отрисовка дерева классификации:



Отрисовка оптимизированного дерева классификации:



Класс нового экземпляра по начальному дереву 2 (87.5%) - окна зданий, не плавильная обработка.

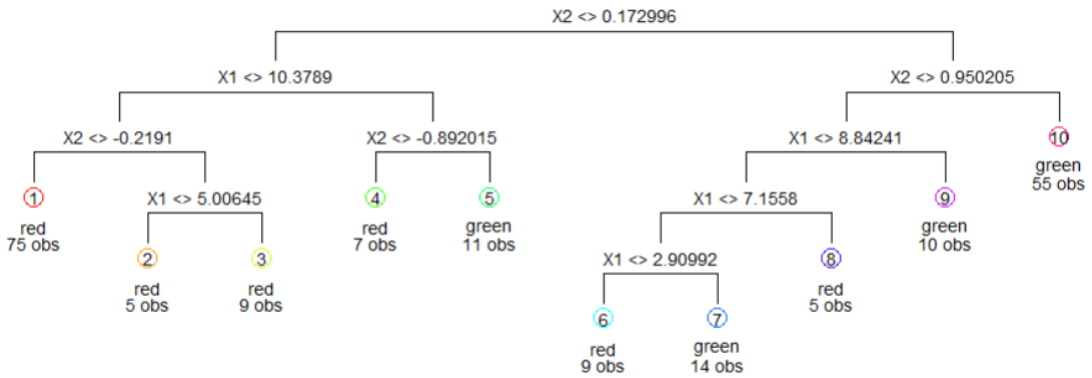
Класс нового экземпляра по оптимизированному дереву 5 (50%) - контейнеры.

Задание №6

Для построения классификатора используйте заранее сгенерированные обучающие и тестовые выборки, хранящиеся в файлах svmdata4.txt, svmdata4test.txt.

```
# Код на R
# Загрузка обучающей и тестовой выборок
data.train <- read.table("C:/Users/Артем/Documents/Лабы по МО/All_Labs/
 ↪svmdata4.txt", stringsAsFactors = TRUE)
data.test <- read.table("C:/Users/Артем/Documents/Лабы по МО/All_Labs/
 ↪svmdata4test.txt", stringsAsFactors = TRUE)

# Построение дерева классификации
data.tree <- tree(Colors ~ ., data.train)
draw.tree(data.tree, cex = 0.7)

# Прогнозирование на тестовой выборке
predictions <- predict(data.tree, newdata = data.test, type = "class")

# Оценка точности модели на тестовой выборке (зависит от целей задачи)
accuracy <- sum(predictions == data.test$Colors) / length(data.test
 ↪$Colors)
print(accuracy)
```

Отрисовка дерева классификации:



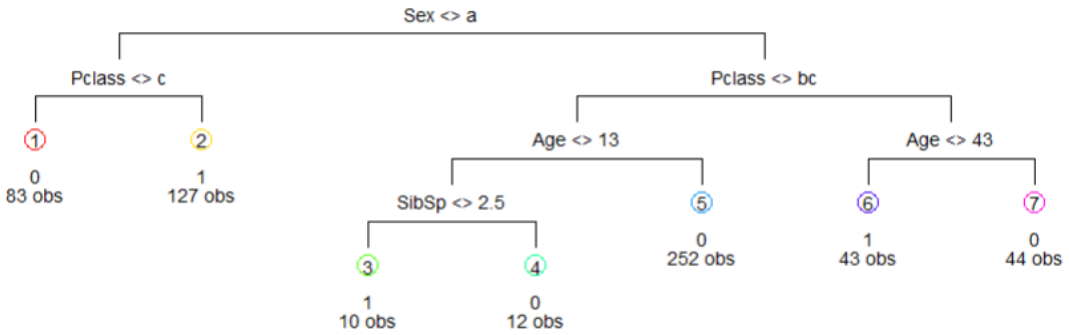
Точность при классификации данных тестовой выборки составила 90%.

Задание №7

Разработать классификатор на основе дерева решений для данных Титаник (Titanic dataset) - https://www.kaggle.com/c/titanic

```
# Код на R
# Загрузка данных
data <- read.csv("C:/Users/Артем/Documents/R-script/train.csv", sep = ',
 ↪', stringsAsFactors = TRUE)

# Рассмотрим только параметры Pclass, Sex, Age, SibSp, Parch и Embarked
data <- select(data, Survived, Pclass, Sex, Age, SibSp, Parch)

# Удаляем строки с пыстыми значениями
data <- na.omit(data)

# Преобразование категориальных признаков в факторы
data$Survived <- factor(data$Survived)
data$Pclass <- factor(data$Pclass)

# Разделение данных на обучающую и тестовую выборку (80% / 20%)
n_row = nrow(data)
total_row = 0.8 * n_row
train_sample <- 1: total_row
data.train <- data[train_sample, ]
data.test <- data[-train_sample, ]

# Построение дерева классификации
tree_model <- tree(Survived ~ ., data = data.train)
draw.tree(tree_model , cex = 0.7)

# Прогнозирование на тестовой выборке
predicted <- predict(tree_model, data.test, type = "class")

# Оценка точности модели на тестовой выборке
accuracy <- sum(predicted == data.test$Survived) / nrow(data.test)
print(accuracy)
```

Отрисовка дерева классификации:



Точность при классификации данных тестовой выборки составила 83%.