Министерство образования и науки РФ
Санкт-Петербургский Политехнический университет Петра Великого
Институт компьютерных наук и технологий
Высшая школа искусственного интеллекта

ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ № 4

«Деревья решений»
по дисциплине «Машинное обучение»

Выполнила:
студентка гр. 3540201/20301 _____ Климова О. А.
подпись, дата

Проверил:
д.т.н., проф. _____ Уткин Л. В.
подпись, дата

Санкт-Петербург
2022

Содержание

Постановка задачи

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.

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.

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

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?

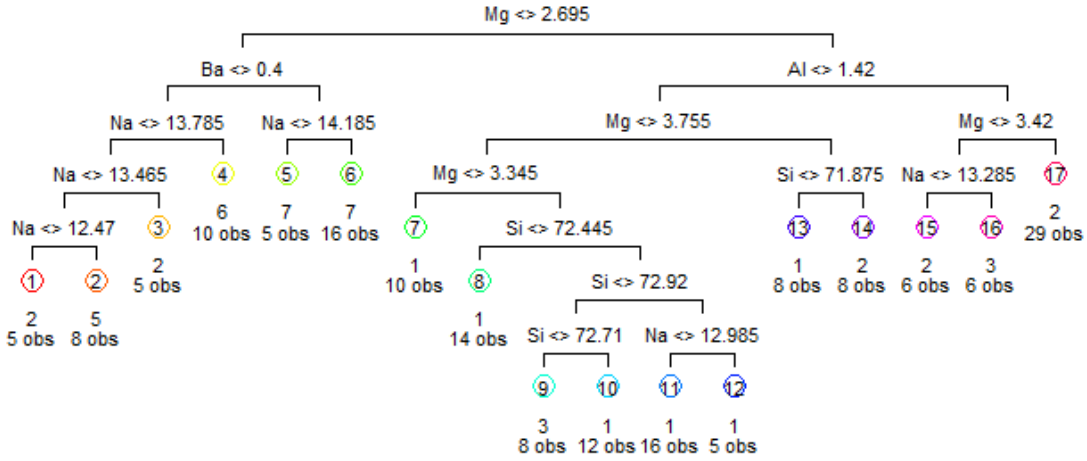
5) Для построения классификатора используйте заранее сгенерированные обучающие и тестовые выборки, хранящиеся в файлах svmdata4.txt, svmdata4test.txt.

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

1 Задание 1

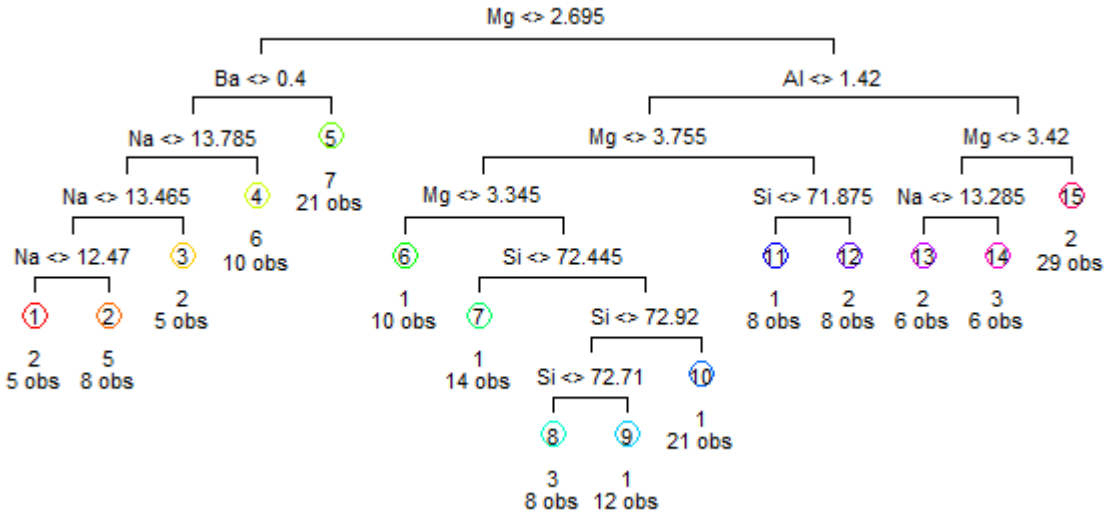
В данном задании был загружен набор данных Glass из пакета “mlbench” и построено дерево классификации. Код представлен в Приложении 1.

Полученное дерево имеет вид:



Данное дерево является избыточным, так как есть разветвления, где выбор идет между двумя одинаковыми классами (на узлах 5-6 и 11-12).

Для построения оптимального дерева необходимо удалить избыточные, тогда дерево привет вид:

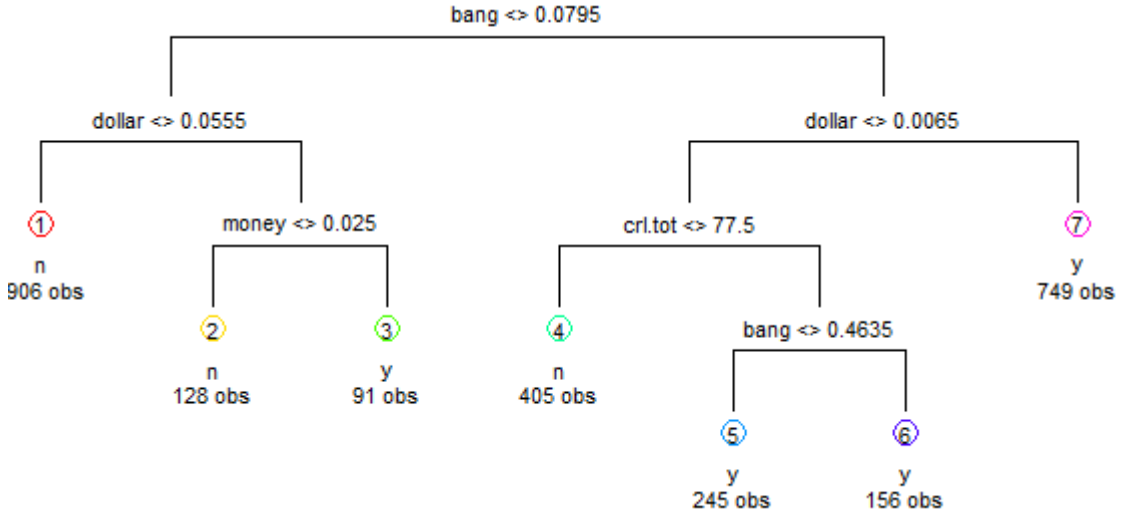


Можно видеть, что избыточные ветви были удалены.

Пример при 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 относится к классу 2 с вероятностью 60% и классу 5 с вероятностью 60%.

2 Задание 2

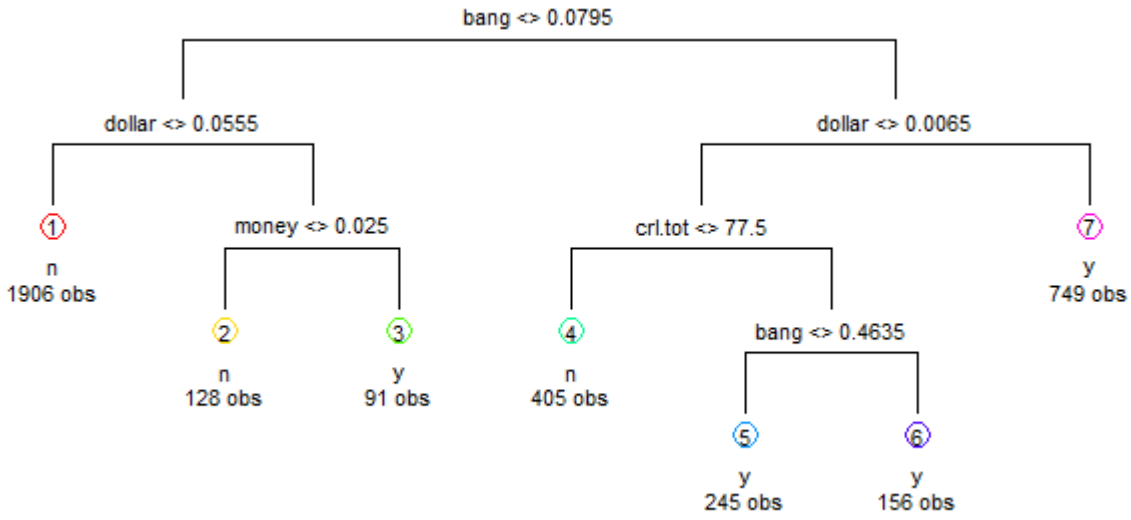
В данном задании был загружен набор данных spam7 из пакета DAAG. Было построено дерево классификации:



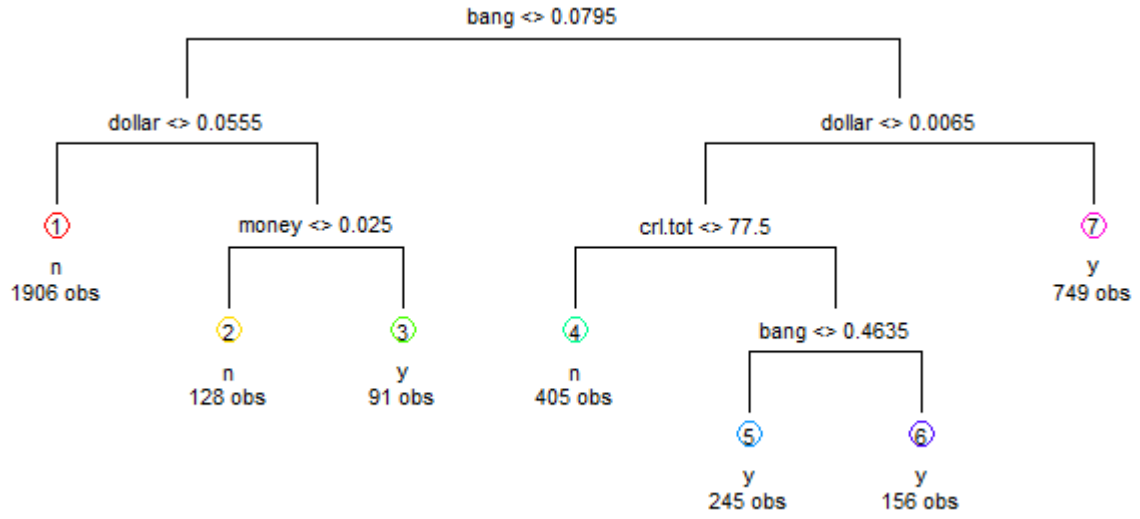
Полученное дерево является избыточным, так как существует выбор между двумя одинаковыми классами (5-6).

Далее была запущена процедура “cost-complexity prunning” с выбором параметра k по умолчанию, method = ’misclass’ и выведена полученная последовательность деревьев:

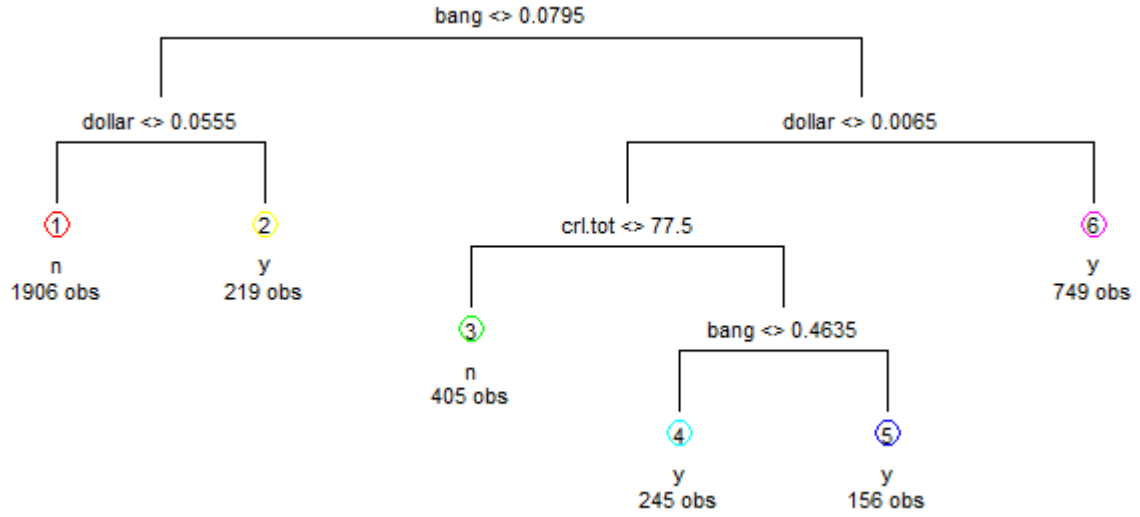
1) k = 0:



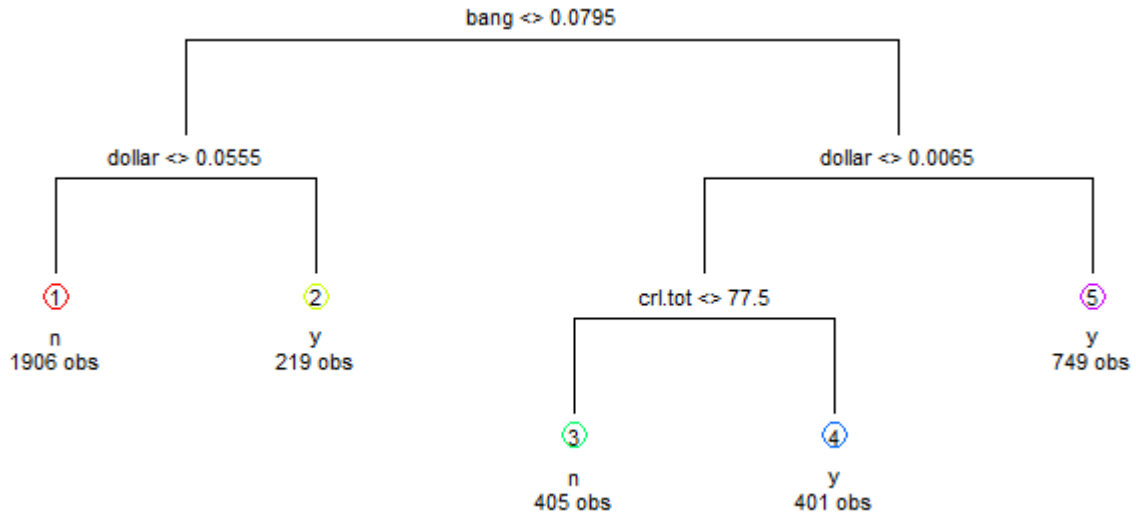
2) k = 12:



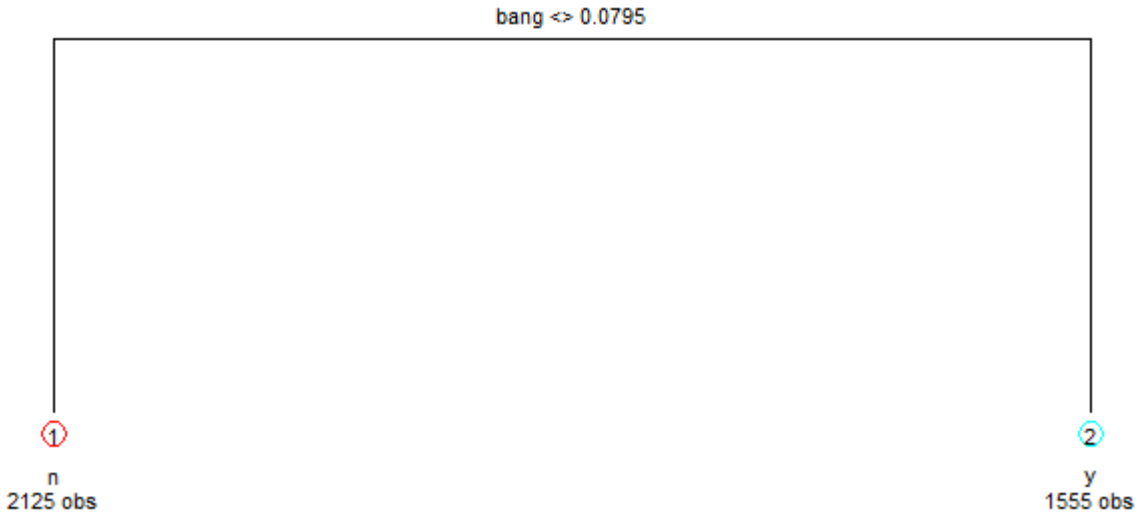
3) k = 63:



4) k = 83.5:



5) k = 83.5:

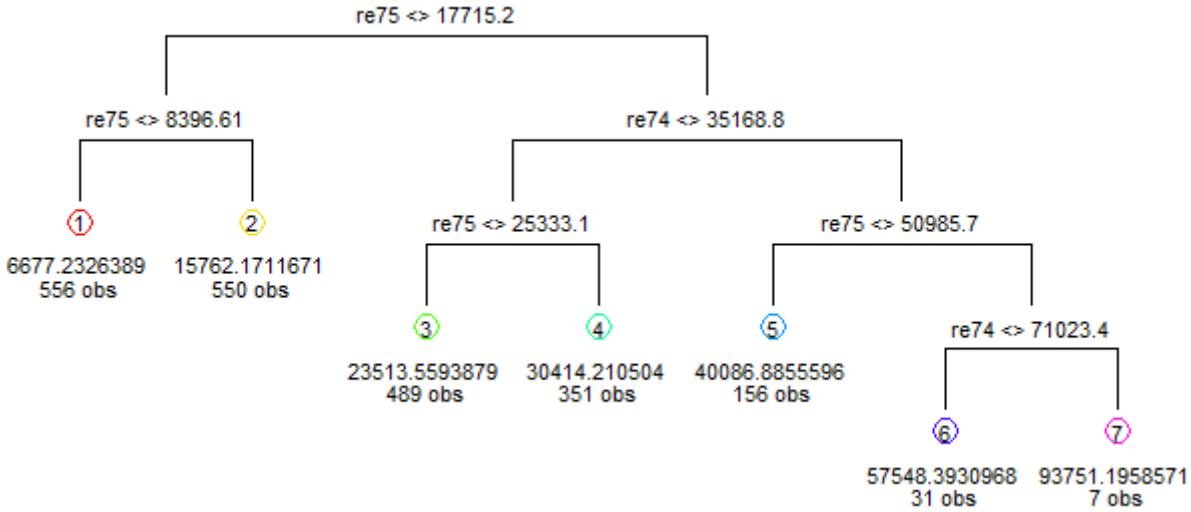


Из полученных деревьев наиболее оптимальным с точки зрения архитектуры является дерево под номером 4, при котором k = 83.5. Данное дерево не является избыточным – нет выбора между одинаковыми классами, но оно является менее точным с точки зрения классификации. Поэтому, наиболее оптимальными с точки зрения классификации являются деревья при k = 0 и k = 12 (левая ветка у этих деревьев разветвляется еще на две в отличии от дерева под номером 4).

Код представлен в Приложении 2.

3 Задание 3

В данном задании был загружен набор данных nsw74psid1 из пакета DAAG. Было построено регрессионное дерево для модели, задаваемой следующей формулой: re78 ~.:

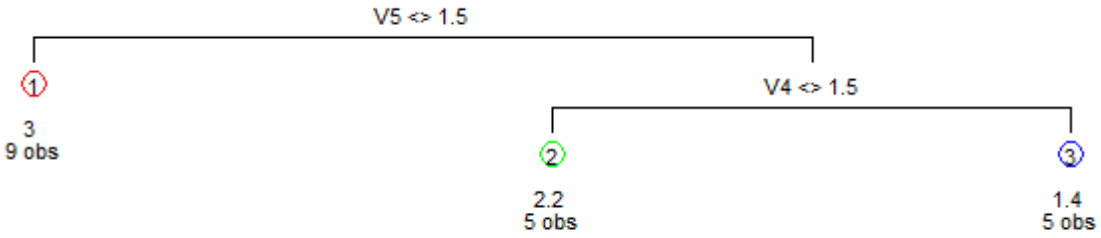


Также была построена регрессионная модель и SVM-регрессия для данной формулы. В результате сравнения двух полученных моделей было выяснено, что ошибка на тестовой выборке при использовании дерева больше, чем при использовании метода опорных векторов (svm).

Код представлен в приложении 3.

4 Задание 4

В данном задании был загружен набор данных Lenses Data Set из файла Lenses.txt. Было построено дерево решений:

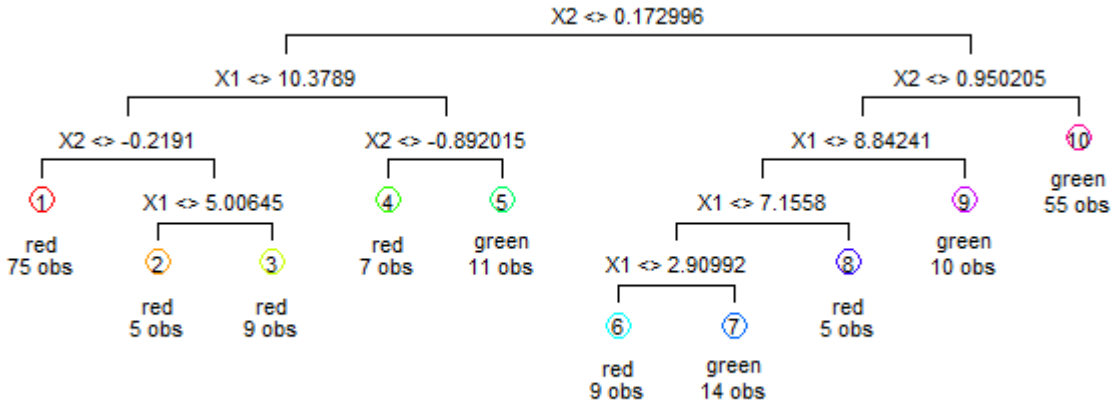


Было выяснено, что при предстарческой дальнозоркости, близорукости, при наличии астигматизма и сокращенной слезы не следует носить контактные линзы.

Код представлен в Приложении 4.

5 Задание 5

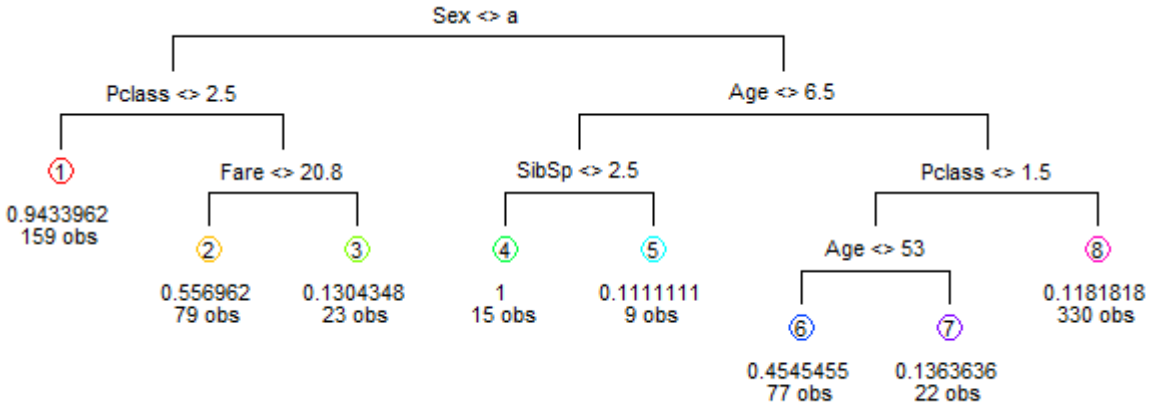
Для построения классификатора были использованы заранее сгенерированные обучающие и тестовые выборки, хранящиеся в файлах svmdata4.txt, svmdata4test.txt:



Код представлен в Приложении 5.

6 Задание 6

Был разработан классификатор на основе дерева решений для данных Титаник (Titanic dataset):



На тестовых данных была получена точность ≈97%:

```
fit   0   1
  0 260   8
  1   6 144
> accuracy
[1] 0.9665072
```

Код представлен в Приложении 6.

Приложение 1. Код для задания 1

```
#ЗАДАНИЕ 1
library(tree)
library(maptree)
library(mlbench)
#загружаем данные Glass
data(Glass)
#определяем общее количество примеров в обучающей выборке
Glass <- Glass[,-1]
n <- dim(Glass)[1]
#делим данные на тестовую и обучающую выборку (80% обучающая)
glass_rand <- Glass[ order(runif(n)),]
nt <- as.integer(n*0.8)
glass_train <- glass_rand[1:nt, ]
glass_test <- glass_rand[(nt+1):n, ]

#построим дерево классификации для данных Glass:
glass.tr <- tree(Type ~., glass_train)
#построение дерева решений
plot(glass.tr, type = "uniform")
text(glass.tr,cex=0.7)
#красивое изображение дает функция draw.tree из библиотеки maptree:
draw.tree(glass.tr, cex=0.7)
glass.tr
#обрежем дерево
glass.tr_opt <- snip.tree(glass.tr, nodes = c(5,103))
draw.tree(glass.tr_opt, cex=0.7)
#пример
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)
predict(glass.tr_opt,example)
```

Приложение 2. Код для задания 2

```
library(tree)
library(maptree)
library(DAAG)
data(spam7)
n <- dim(spam7)[1]
#разделяем на обучающую и тестовую выборку (обучающая 80%)
spam7_rand <- spam7[ order(runif(n)),]
nt <- as.integer(n*0.8)
spam7_train <- spam7_rand[1:nt, ]
spam7_test <- spam7_rand[(nt+1):n, ]
#дерево для spam7
spam7.tr <- tree(yesno ~., spam7_train)
draw.tree(spam7.tr, cex=0.7)
#процедура “cost-complexity prunning”
newspam7.tr <- prune.tree(spam7.tr, method = "misclass")
newspam7.tr$k
draw.tree(prune.tree(spam7.tr, k = 0), cex=0.7)
draw.tree(prune.tree(spam7.tr, k = 12), cex=0.7)
draw.tree(prune.tree(spam7.tr, k = 63), cex=0.7)
draw.tree(prune.tree(spam7.tr, k = 83.5), cex=0.7)
draw.tree(prune.tree(spam7.tr, k = 673), cex=0.7)
```

Приложение 3. Код для задания 3

```
#ЗАДАНИЕ 3
library(tree)
library(maptree)
library(DAAG)
library(e1071)
data(nsw74psid1)
n <- dim(nsw74psid1)[1]
#разделяем на тестовую и обучающую
nsw_rand <- nsw74psid1[order(runif(n)),]
nt <- as.integer(n*0.8)
nsw_train <- nsw_rand[1:nt, ]
nsw_test <- nsw_rand[(nt+1):n, ]
#строим дерево решений для nsw74psid1
nsw.tr <- tree(re78 ~., nsw_train)
draw.tree(nsw.tr, cex=0.7)
predictions_tree <- predict(nsw.tr, nsw_test[-10])
#регрессионная модель на методе опорных векторов
svmModel <- svm(nsw_train[-10], nsw_train$re78, type = "eps-regression",
cost=1,kernel="radial")
predictions_svm <- predict(svmModel, nsw_test[-10])
tab2 = table(nsw_test$re78, predictions_svm)
#вычислим ошибки (tree_mistake > svm_mistake)
tree_mistake <- sd(nsw_test$re78 - predictions_tree)
svm_mistake <- sd(nsw_test$re78 - predictions_svm)
tree_mistake
svm_mistake
```

Приложение 4. Код для задания 4

```
#ЗАДАНИЕ 4
library(tree)
library(maptree)
lenses_raw <- read.table("C:/Users/Unicorn/Desktop/Машинное Обучение/Лабы/Lenses.txt",
sep = "", stringsAsFactors = TRUE)
lenses_raw <- lenses_raw[,-1]
n <- dim(lenses_raw)[1]
lenses_rand <- lenses_raw[ order(runif(n)),]
#для обучения возьмем 90%
nt <- as.integer(n*0.8)
lenses_train <- lenses_rand[1:nt, ]
lenses_test <- lenses_rand[(nt+1):n, ]
#дерево решений
lenses.tr <- tree(V6 ~., lenses_train)
draw.tree(lenses.tr, cex=0.7)
#пример
example <- data.frame(V2=2, V3=1, V4=2, V5=1)
predict(lenses.tr,example)
```

Приложение 5. Код для задания 5

```
#ЗАДАНИЕ 5
svmdata4 <- read.table("C:/Users/Unicorn/Desktop/Машинное Обучение/Лабы/svmdata4.txt", sep = "\t", stringsAsFactors = TRUE)
svmdata4test <- read.table("C:/Users/Unicorn/Desktop/Машинное Обучение/Лабы/svmdata4test.txt", sep = "\t", stringsAsFactors = TRUE)
#построение дерева решений
svmdata4.tr <- tree(Colors ~., svmdata4)
draw.tree(svmdata4.tr, cex=0.7)
pred_svmdata4 <- predict(svmdata4.tr, svmdata4test)
```

Приложение 6. Код для задания 6

```
#ЗАДАНИЕ 6
library(dplyr)
Titanic_train<-read.csv("C:/Users/Unicorn/Desktop/Машинное
Обучение/Лабы/Titanic_train.csv", header = TRUE, sep = ",", dec = ".",
                 stringsAsFactors = TRUE)
Titanic_test<-read.csv("C:/Users/Unicorn/Desktop/Машинное
Обучение/Лабы/Titanic_test.csv", header = TRUE, sep = ",", dec = ".",
                 stringsAsFactors = TRUE)
#построение дерева решений
Titanic.tr <- tree(Survived ~ Pclass + Sex + Age + SibSp + Parch + Fare + Embarked,
Titanic_train)
draw.tree(Titanic.tr, cex=0.7)
fit <- round(predict(Titanic.tr,Titanic_test))
Gen_sub=read.csv("C:/Users/Unicorn/Desktop/Машинное
Обучение/Лабы/gender_submission.csv", header = TRUE, sep = ",", dec = ".",
                 stringsAsFactors = FALSE)
tab = table(fit, Gen_sub$Survived)
accuracy = (tab[1,1] + tab[2,2]) / (tab[1,1] + tab[2,2] + tab[1,2] + tab[2,1])
tab
accuracy
```