1. Министерство науки и высшего образования Российской Федерации
2. Санкт-Петербургский Политехнический Университет Петра Великого
3. —

Институт компьютерных наук и технологий

Высшая школа искусственного интеллекта

Отчёт по лабораторной работе «Деревья решений»

по дисциплине «Машинное обучение»

Выполнила

студентка гр. 3540201/20302 Обидина А.И.

Проверил

профессор Уткин Л.В.

Санкт-Петербург

2022

**Задание**

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) 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

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

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

Исходные обучающие данные для классификации – в файле Titanic\_train.csv.

Данные для тестирования – в файле Titanic\_test.csv

# Ход работы

## **Пункт 1**

Построенное без оптимизаций дерево представлено на рисунке 1, оно избыточно, следовательно, нуждается в оптимизации. Избыточно, так как в некоторых случаях разветвления ведут к одинаковым классам, как в случаях 12 и 13, 18 и 19 (при проведении оптимизации были удалены некоторые узлы).

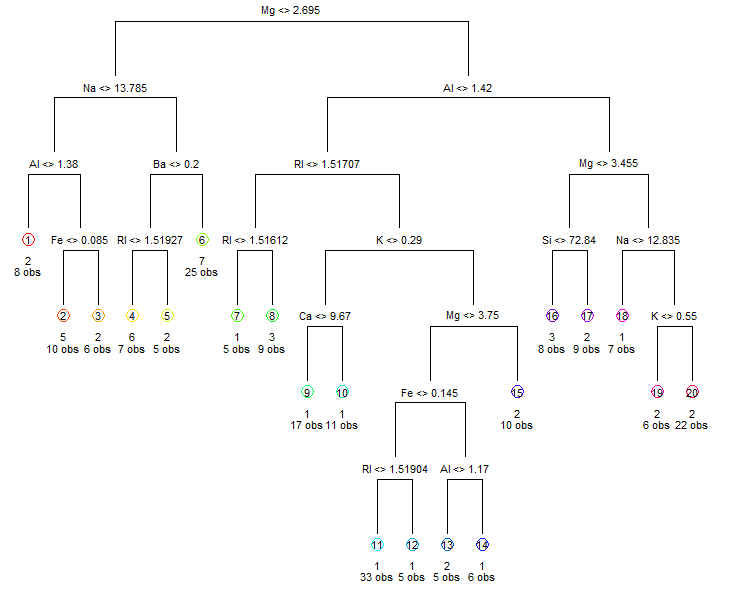


Рисунок 1 – Дерево без применения оптимизации

После применения оптимизации дерево приняло вид, который продемонстрирован на рисунке 2.

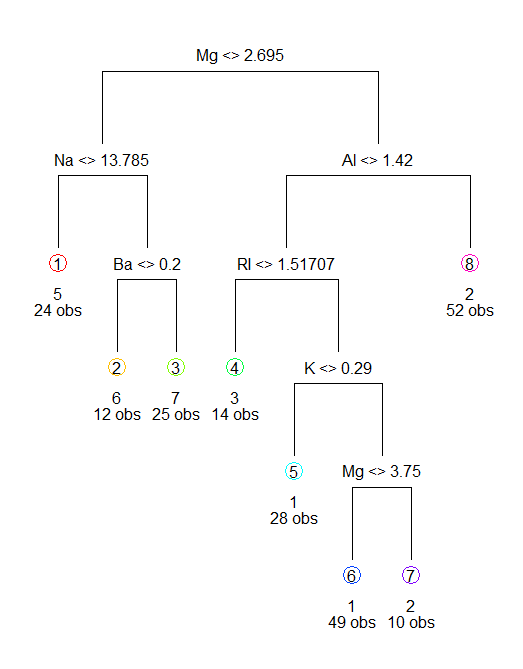


Рисунок 2 – Дерево после применения оптимизации

Код:

library(mlbench)

library(kknn)

library(maptree)

library(tree)

data(Glass)

glass <- Glass[, -1] #убрать признак Id

m <- dim(glass)[1]

glass.tr <- tree(Type ~., Glass)

print(summary(glass.tr))

draw.tree(glass.tr, cex = 0.7)

glass.tr

glass.tr1 <- snip.tree(glass.tr, nodes = c(8,9,20,21,26,27,52,53,54,55,14,15))

draw.tree(glass.tr1)

glass.tr2 <- prune.tree(glass.tr,best=8)

draw.tree(glass.tr2)

## **Пункт 2**

Для проведения правильной оптимизации нужно было подобрать параметр best в prune.tree. Опытным путем было подобрано значение параметра, оптимальное его значение - 5. Исходное дерево решений представлено на рисунке 3, на рисунке 4 – дерево после проведения оптимизации.

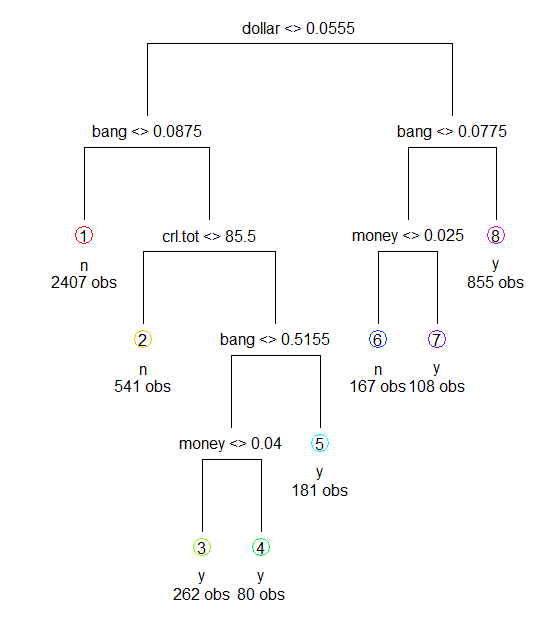


Рисунок 3 – Дерево без применения оптимизации

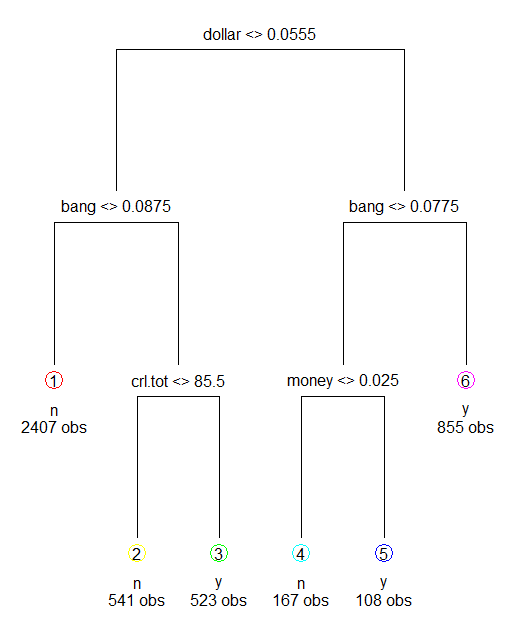


Рисунок 4 – Дерево после применения оптимизации

Код:

library(maptree)

library(tree)

install.packages("DAAG")

install.packages("png")

install.packages("jpeg")

library(DAAG)

data(spam7)

spam.tr<-tree(yesno ~., spam7)

draw.tree(spam.tr)

print(summary(spam.tr))

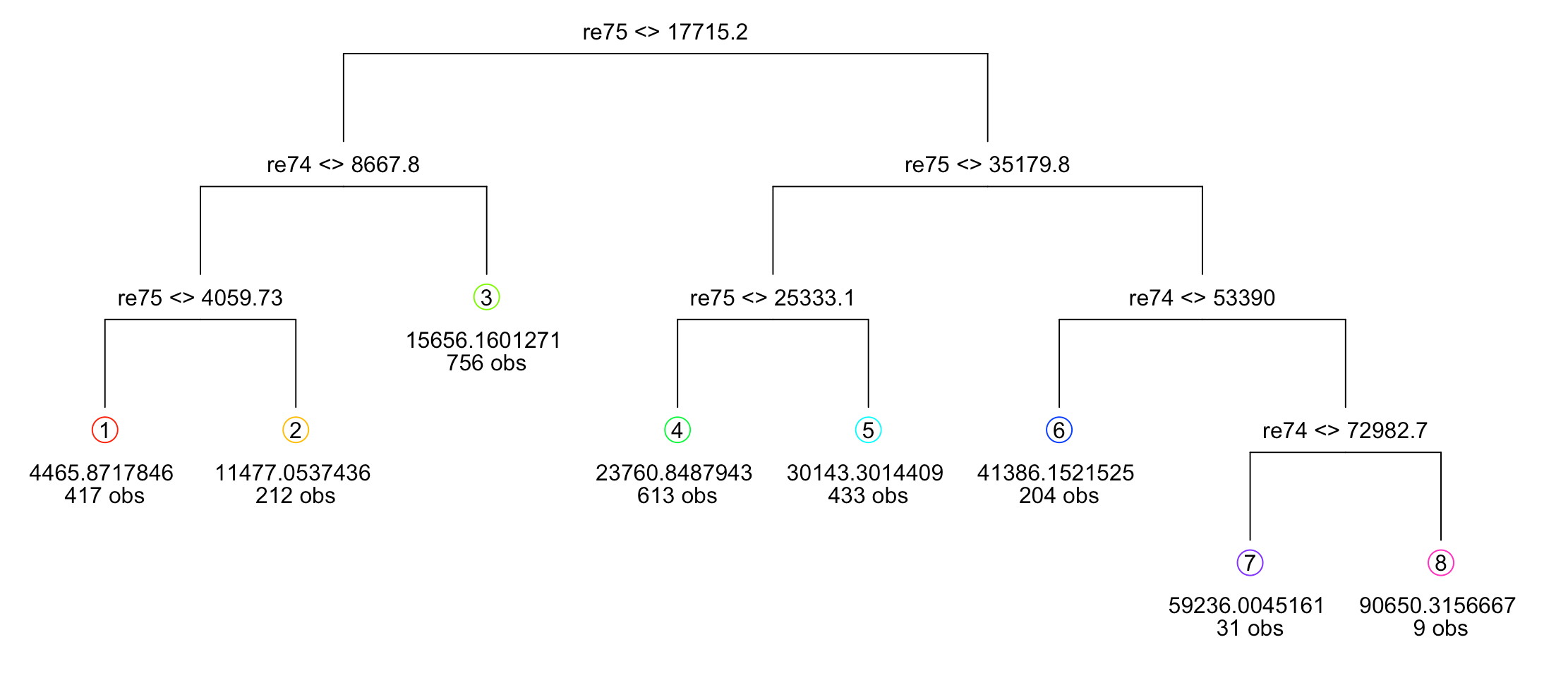
spam.tr1<-prune.tree(spam.tr, method = "misclass", best = 5)

draw.tree(spam.tr1)

print(summary(spam.tr1))

## **Пункт 3**

Загрузите набор данных nsw74psid1 из пакета DAAG. Постройте регрессионное дерево для модели, задаваемой следующей формулой: **re78 ~.**.

Рисунок 5 – Регрессионное дерево

Был загружен набор данных nsw74psid1 из пакета DAAG и построено регрессионное дерево для модели, задаваемой следующей формулой: **re78 ~.** (представлено на рисунке 5).

Также были построены регрессионная модель и SVM-регрессия для данной формулы. У построенных моделей было сравнено качество с помощью расчета среднеквадратичных ошибок, самая большая среднеквадратичная ошибка получилась у регрессионного дерева, примерно одинаковые – у регрессионной модели и SVM-регрессии. Таким образом, для решения задачи регрессии наиболее оптимальные модели из предложенных – регрессионная и SVM-регрессия (в дереве число листьев ограничено, решать задачу регрессии с помощью него не очень оптимально).

Код:

library(maptree)

library(tree)

library(DAAG)

library(e1071)

data(nsw74psid1)

num<-dim(nsw74psid1)[1]

ns\_r<-nsw74psid1[order(runif(num)),]

ns\_r

num\_train <- as.integer(num\*0.7)

train <- ns\_r[1:num\_train, ]

test <- ns\_r[(num\_train+1):num, ]

ns.tr <- tree(re78~., train)

draw.tree(ns.tr)

summary(ns.tr)

ns.lr <- lm(re78~., train)

summary(ns.lr)

ns.svm <- svm(re78~., data=train, type = "eps-regression",eps=0.5,cost = 1)

summary(ns.svm)

ns\_tr\_predict <- predict(ns.tr, test[-10])

ns\_lr\_predict <- predict(ns.lr, test[-10])

ns\_svm\_predict <- predict(ns.svm, test[-10])

tr\_mistake <- sd(test$re78 - ns\_tr\_predict)

lr\_mistake <- sd(test$re78 - ns\_lr\_predict)

svm\_mistake <- sd(test$re78 - ns\_svm\_predict)

tr\_mistake

lr\_mistake

svm\_mistake

## **Пункт 4**

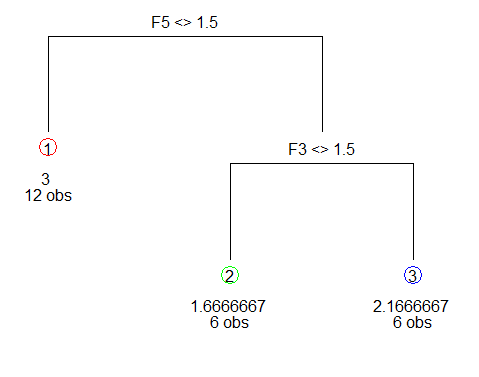


Рисунок 6 – Дерево решений для базы данных «Lenses»

Было построено дерево классификации по базе данных «Lenses», оно представлено на рисунке 6. В построенном дереве для классификации использованы 2 признака – состояние зрения и состояние слезы.

При предстарческой дальнозоркости, близорукости, при наличии астигматизма и сокращенной слезы согласно предсказанию пациенту не следует носить контактные линзы.

Код:

library(tree)

library(maptree)

features <- c("F1", "F2", "F3", "F4", "F5", "Class")

lenses = read.table("D://Lenses.txt", col.names = features)

lenses

lenses = lenses[, -1] #delete ID (F1)

lenses

len.tr <- tree(Class~., lenses)

draw.tree(len.tr)

str <- data.frame("F2" = 2, "F3" = 1, "F4" = 2, "F5" = 1, "Class" = "")

predict(len.tr, str)

summary(len.tr)

len.tr

## **Пункт 5**



Согласно предсказанию, тип стекла – контейнерное.

Код:

str <- 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, "Type" = "" )

predict(gl.tr2, str)

## **Пункт 6**

Для обучающего набора данных было построено дерево решений (рис. 7), после чего оно было оптимизировано (удалена вершина 4). Дерево решений после оптимизации представлено на рисунке 8.

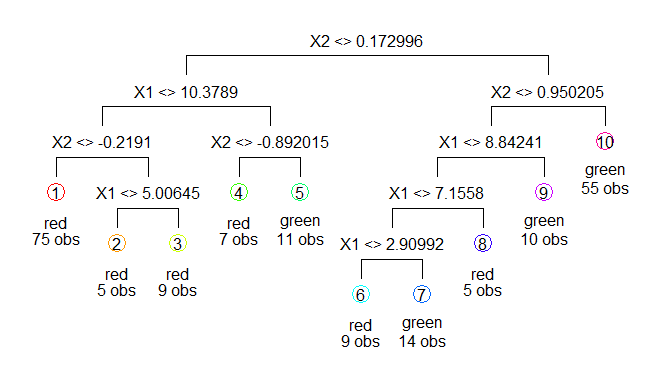


Рисунок 7 – Дерево решений до оптимизации

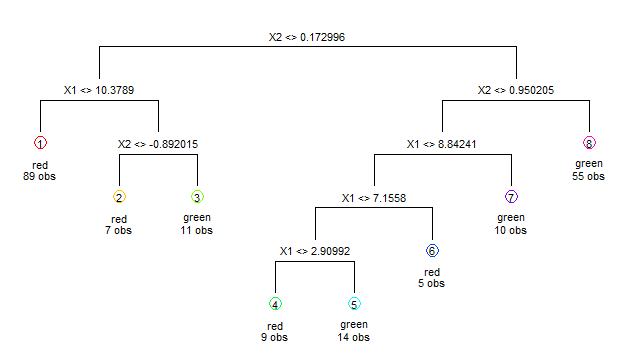


Рисунок 8 – Дерево решений после оптимизации

Точность классификации составила 0,9.

Код:

library(tree)

library(maptree)

train <- read.table("D://svmdata4.txt", stringsAsFactors = TRUE)

test <- read.table("D://svmdata4test.txt", stringsAsFactors = TRUE)

test

svmdata.tr <- tree(Colors ~., train)

draw.tree(svmdata.tr)

svmdata.tr

svmdata.tr1 <- snip.tree(svmdata.tr, nodes = 4)

draw.tree(svmdata.tr1, cex = 0.7)

res <- as.matrix(predict(svmdata.tr1, test))

predict(svmdata.tr1, test)

num\_test <- dim(test)[1]

num\_correct <- 0

i <- 1

while (i <= num\_test){

if(res[[i,1]] < res[[i,2]]) type = "red"

else type = "green"

if(type == test[[i, 3]]) num\_correct = num\_correct +1

i = i+1

}

acc=num\_correct/num\_test

acc

## **Пункт 7**

Для обучающего набора данных было построено дерево решений (рис. 9), при его построении были учтены такие признаки из базы данных «Титаник», как : класс каюты, пол, возраст, число родственников (братьев и сестер, наличие мужа жены, родителей и детей) на корабле, стоимость билета, порт посадки, но не учитывались Id, имя, номер билета и каюты пассажира. Точность классификации – 71,72%.

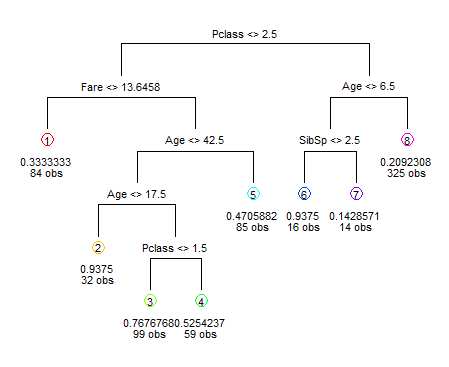


Рисунок 9 – Дерево решений для базы данных «Titanic»

Код:

library(tree)

library(maptree)

train<-read.csv(file="D://train.csv")

test<-read.csv(file="D://test.csv")

test

titanic.tr<-tree(Survived ~ Pclass + Sex + Age + SibSp + Parch + Fare + Embarked, train)

draw.tree(titanic.tr, cex = 0.7)

summary(titanic.tr)

titanic.pred<-predict(titanic.tr, test)

titanic.pred

res <- as.matrix(titanic.pred)

res

titanic.sol<-data.frame(PassengerId = test$PassengerId, Survived = titanic.pred)

num\_test <- dim(test)[1]

num\_correct <- 0

i <- 1

while (i <= num\_test){

if(res[i] < 0.5) type = 0

else type = 1

if(type == test[i, 12]) num\_correct = num\_correct +1

i = i+1

}

acc=num\_correct/num\_test

acc

**Вывод**

При выполнении данной лабораторной работы было изучено построение деревьев решений как для задач классификации и регрессии. Чтобы качество классификации/регрессии было оптимальным, нужно производить оптимизацию дерева. Метод хорош, но имеет ряд недостатков, например ограничения на вложенность дерева 32 в языке R, поэтому классификатор для данных большой размерности построить не получится.