

第9章 常用机器学习算法

1. 利用 R 语言内置命令实现 C4.5 的建树。

【参考答案】C4.5 的建树 R 代码如下：

以鸢尾花为例：

```
library(RWeka)
library(party)
oldpar=par(mar=c(3, 3, 1.5, 1), mgp=c(1.5, 0.5, 0), cex=0.3)
data(iris)
m1<-J48(Species ~ Petal.Width+Petal.Length, data=iris)
m1
table(iris$Species, predict(m1))
write_to_dot(m1)
if(require("party", quietly=TRUE))
plot(m1)
```

2. 利用 iris 数据集和 glm 函数进行逻辑回归二分类测试。

【参考答案】参考代码如下：

```
index<- which(iris$Species == 'setosa')
ir<- iris[- index, ]
levels(ir$Species)[1] <- ' '
split<- sample(100, 100* (2/3))
```

#生成训练集

```
ir_train<- ir[split, ]
```

#生成测试集

```
ir_test<- ir[-split, ]
```

```
fit<- glm(Species
```

```
~., family=binomial(link='logit'), data=ir_train)
```

```
summary(fit)
```

```
real<- ir_test$Species
```

```
data.frame(real, predict)
```

```
predict<- predict(fit, type='response', newdata=ir_test)
```

```
res<- data.frame(real, predict
```

```
=ifelse(predict>0.5, 'virginica', 'versicolor'))
```

#查看模型效果(见图 9-1)

```
plot(res)
```

3. 利用 lm 函数拟合数据集，预测女性身高对体重的影响。(提示：女性身高和体重的建模可以参考： $\text{weight} = w * \text{height} + b$)

【参考答案】参考代码如下：

```
fit<- lm(formula = weight ~ height, data = women)
```

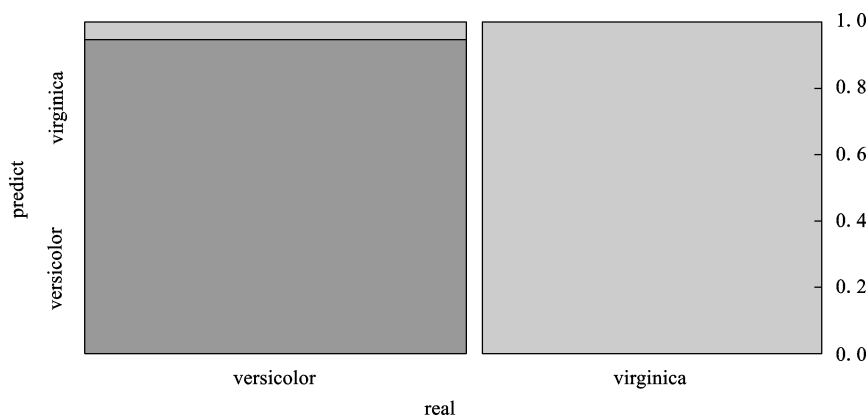


图 9-1 iris 数据集逻辑回归二分类测试图

```
summary(fit)
```

Call:

```
lm(formula = weight ~ height, data = women)
```

4. 利用 nnet 包训练神经网络模型。

(提示: ①先使用安装包以及进行数据分类, 参考代码如下:

```
library(nnet)
```

```
data("iris")
```

```
set.seed(2)
```

```
ind = sample(2, nrow(iris), replace = TRUE, prob = c(0.7, 0.3))
```

```
trainset = iris[ind == 1, ]
```

```
testset = iris[ind == 2, ]
```

②再使用 nnet 包训练神经网络。

③最后调用 summary() 输出训练好的神经网络。)

【参考答案】参考代码如下:

①先使用安装包以及进行数据分类。

```
library(nnet)
```

```
data("iris")
```

```
set.seed(2)
```

```
ind = sample(2, nrow(iris), replace = TRUE, prob = c(0.7, 0.3))
```

```
trainset = iris[ind == 1, ]
```

```
testset = iris[ind == 2, ]
```

②再使用 nnet 包训练神经网络。

```
iris.nn = nnet(Species ~ ., data = trainset, size = 2, rang = 0.1, decay = 5e-4, maxit = 200)
```

```
# weights: 19
```

```
initial value 114.539765
```

```
iter 10 value 52.100312
```

```

iter 20 value 50.231442
iter 30 value 49.526599
iter 40 value 49.402229
iter 50 value 44.680338
iter 60 value 5.254389
iter 70 value 2.836695
iter 80 value 2.744315
iter 90 value 2.687069
iter 100 value 2.621556
iter 110 value 2.589096
iter 120 value 2.410539
iter 130 value 2.096461
iter 140 value 1.938717
iter 150 value 1.857105
iter 160 value 1.825393
iter 170 value 1.817409
iter 180 value 1.815591
iter 190 value 1.815030
iter 200 value 1.814746
final value 1.814746
stopped after 200 iterations

```

③最后调用 `summary()` 输出训练好的神经网络。

```

summary(iris.nn)
a 4-2-3 network with 19 weights
options were - softmaxmodelling decay=5e-04
b->h1 i1->h1 i2->h1 i3->h1 i4->h1
-20.60  0.31  -3.84  3.36  7.72
b->h2 i1->h2 i2->h2 i3->h2 i4->h2
-7.15  1.50  2.49  -4.14  5.59
b->o1 h1->o1 h2->o1
-7.28  -3.67  13.16
b->o2 h1->o2 h2->o2
15.90 -16.64 -19.40
b->o3 h1->o3 h2->o3
-8.62  20.31  6.24

```

在应用函数时可以实现分类观测数据源、隐蔽单元个数 (`size` 参数)、初始随机数权值 (`rang` 参数)、权值衰减参数 (`decay` 参数)、最大迭代次数 (`maxit`)，整个过程会一直重复直至拟合与衰减项收敛。