

```

1  ###決定木 (rpart:2進木解析)#####
2  ##決定木は機械学習系の解析方法なので、本来練習用とテスト用にデータを分割し
3  #学習器を作るという手続きが必要である。本稿は解き方だけ
4  ###Classification:rpart (再帰分割回帰木)
5  install.packages("rpart")
6  install.packages("partykit")
7  install.packages("rattle")
8  install.packages("DALEX")          #Titanic data
9  #install.packages("stablelearner") #Titanic data
10
11  ###分類木:rpart
12  ### Iris.Data #####
13  #data
14  data(iris)
15  colnames(iris)
16  ##欠測値処理
17  anyNA(iris) #NAがあるか
18
19  library(rpart)
20  library(partykit)
21  iris.out <- rpart(Species~., data=iris , #y(Species) ~ ., : x(All Items)
22                  method = "class")
23                  #control = rpart.control(cp = 0.005)) #まずはcontrolなしで
24  #method:anova,poisson(y;2 columns),class(y;factor),exp ;methodは指定がbetter
25  #control:minsplint(nodeの最少観測数),minbucket(leafの最少観測数),cp(複雑パラメータ)
26  #xval(number of cross-validations)
27  names(iris.out) #出力内容
28  as.party(iris.out) #Fitと分割内容
29  iris.out$sctable
30  iris.out$variable.importance #重要度
31  plot(as.party(iris.out))
32  printcp(iris.out)
33  plotcp(iris.out)#CP.Graf
34  ###剪定:prune()関数を用いる方法
35  iris.out2 <- prune(iris.out , cp = 0.06) #printcp(iris.out)のcpを確認して
36  as.party(iris.out2) #剪定の必要なし
37  plot(as.party(iris.out2)) #棒グラフ
38  plot(as.party(iris.out2),
39        terminal_panel=node_barplot(as.party(iris.out2),beside=F)) #帯グラフ
40  #クロス集計:的中率
41  est1 <- predict(iris.out2 , type = "class") #予測値
42  est2 <- xtabs(~ Species + est1 , data = iris ) #予測値と観測値のクロス集計
43  est2
44  round((est2[1,1]+est2[2,2]+est2[3,3])/length(iris$Species),3)*100 #的中率
45
46  ### titanic.Data #####
47  #data
48  library("DALEX")
49  data("titanic")
50  colnames(titanic)
51  #library(stablelearner)
52  #data("titanic") #11 varriables(name,ticket noがある)
53  #colnames(titanic)
54  ##欠測値処理
55  anyNA(titanic[,c(1,2,3,9)])#1,2,3,9列にNAがあるか
56  t.na.omit <- na.omit(titanic[,c(1,2,3,9)])#NA omitファイル作成
57  head(t.na.omit)
58  nrow(na.omit(t.na.omit))##リストワイズ後のサンプル数
59
60  library(rpart)

```

```
61 library(partykit)
62 tita.fit <- rpart( survived ~. ,
63                 data = t.na.omit ,
64                 method = "class" )
65 names(tita.fit)
66 as.party(tita.fit) #Fitと分割内容
67 tita.fit$cptable
68 tita.fit$variable.importance#重要度
69 plot(as.party(tita.fit))
70 printcp(tita.fit)
71 plotcp(tita.fit)#CP.Graf
72 ###剪定:prune()
73 tita.fit2 <- prune(tita.fit , cp = 0.015)
74 as.party(tita.fit2) #Fit
75 tita.fit2$cptable
76 tita.fit2$variable.importance#重要度
77 plot(as.party(tita.fit2),
78       terminal_panel=node_barplot(as.party(tita.fit2),beside=T))
79 #クロス集計:的中率
80 estt1 <- predict(tita.fit2 , type = "class")
81 estt2 <- xtabs( ~ survived + estt1 , data = t.na.omit )
82 estt2
83 round((estt2[1,1]+estt2[2,2])/length(t.na.omit$survived),3)*100
84 #剪定:的中率の差は2%
85
86 #####回帰木#####
87 ### Cars93.Data
88 library(MASS)
89 data(Cars93)
90 colnames(Cars93)
91 ###欠測値処理
92 anyNA(Cars93) #NAがあるか
93 c.na.omit <- na.omit(Cars93) #NA omitファイル作成
94 head(c.na.omit)
95 nrow(na.omit(c.na.omit)) ##リストワイズ後のサンプル数
96
97 library(rpart)
98 library(partykit)
99 out.c <- rpart( Price ~ Type+EngineSize+Length,
100               data=c.na.omit ,
101               method = "anova" )
102 names(out.c)
103 as.party(out.c) #Fitと分割内容
104 out.c$cptable
105 out.c$variable.importance#重要度
106 plot(as.party(out.c))
107 printcp(out.c)
108 plotcp(out.c)#CP.Graf
109
110 ###剪定
111 out.c2 <- prune(out.c , cp = 0.07)
112 as.party(out.c2) #Fit
113 out.c2$cptable
114 out.c2$variable.importance#重要度
115 plot(as.party(out.c2))
116 sum(residuals(out.c2)^2) #残差二乗和
117 #回帰推定による残差二乗和と比較
118 out.lm <- lm( Price ~ Type+EngineSize+Length,
119              data=c.na.omit) #回帰推定
120 sum(residuals(out.lm)^2) #回帰推定による残差二乗和
```

```
121  boxplot (data.frame (Rpart=residuals (out.c2) ,  
122                      LM=residuals (out.lm) ) ) #残差二乗和の比較  
123  
124
```