Homework3

YikunHan42

目录

1 导入第三方库

```
library(mlr)
```

```
## Warning: 程辑包'mlr'是用R版本4.1.3 来建造的

## 载入需要的程辑包：ParamHelpers

## Warning: 程辑包'ParamHelpers'是用R版本4.1.3 来建造的

## Warning message: 'mlr' is in 'maintenance-only' mode since July 2019.
## Future development will only happen in 'mlr3'
## (<https://mlr3.mlr-org.com>). Due to the focus on 'mlr3' there might be
## uncaught bugs meanwhile in {mlr} - please consider switching.
```

```
library(ISLR)
```

```
## Warning: 程辑包'ISLR'是用R版本4.1.3 来建造的
```

```
library(ggplot2)
library(reshape2)
```

```
## Warning: 程辑包'reshape2'是用R版本4.1.3 来建造的
```

```
library(plyr)
```

```
## Warning: 程辑包'plyr'是用R版本4.1.3 来建造的
```

```
library(dplyr)
```

```
##
## 载入程辑包：'dplyr'
## The following objects are masked from 'package:plyr':
##
##     arrange, count, desc, failwith, id, mutate, rename, summarise,
##     summarize

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(class)
```

```
## Warning: 程辑包'class'是用R版本4.1.3 来建造的
```

```
library(tidyverse)
```

```
## Warning: 程辑包'tidyverse'是用R版本4.1.3 来建造的
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --
```

```
## v tibble  3.1.6     v purrr   0.3.4
## v tidyr   1.2.0     v stringr 1.4.0
## v readr   2.1.2     v forcats 0.5.1
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::arrange()   masks plyr::arrange()
## x purrr::compact()   masks plyr::compact()
## x dplyr::count()     masks plyr::count()
## x dplyr::failwith()  masks plyr::failwith()
## x dplyr::filter()    masks stats::filter()
## x dplyr::id()        masks plyr::id()
## x dplyr::lag()       masks stats::lag()
## x dplyr::mutate()    masks plyr::mutate()
## x dplyr::rename()    masks plyr::rename()
## x dplyr::summarise() masks plyr::summarise()
## x dplyr::summarize() masks plyr::summarize()
```

```
library(plotly)
```

```
## Warning: 程辑包'plotly'是用R版本4.1.3 来建造的
```

```
##
## 载入程辑包：'plotly'
```

```
## The following objects are masked from 'package:plyr':
##
##     arrange, mutate, rename, summarise
```

```
## The following object is masked from 'package:ggplot2':
##
##     last_plot
```

```
## The following object is masked from 'package:stats':
##
##     filter
```

```
## The following object is masked from 'package:graphics':
##
##     layout
```

2 读入和清洗数据

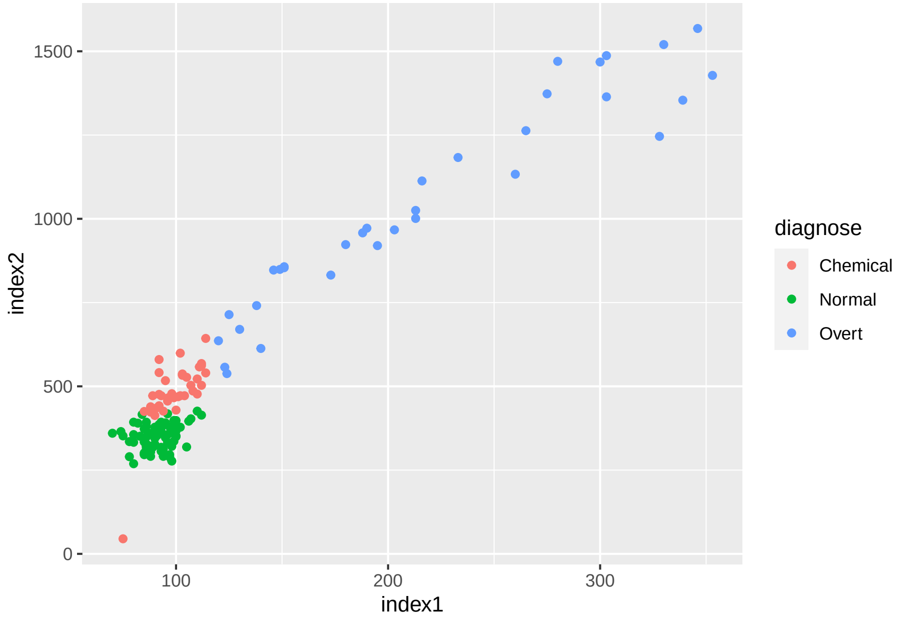
```
data <- read.csv("D://Study/DSBI/Task3/data.csv")
data[!complete.cases(data),]
```

```
## [1] diagnose index1   index2   index3
## <0 行> (或0-长度的row.names)
```

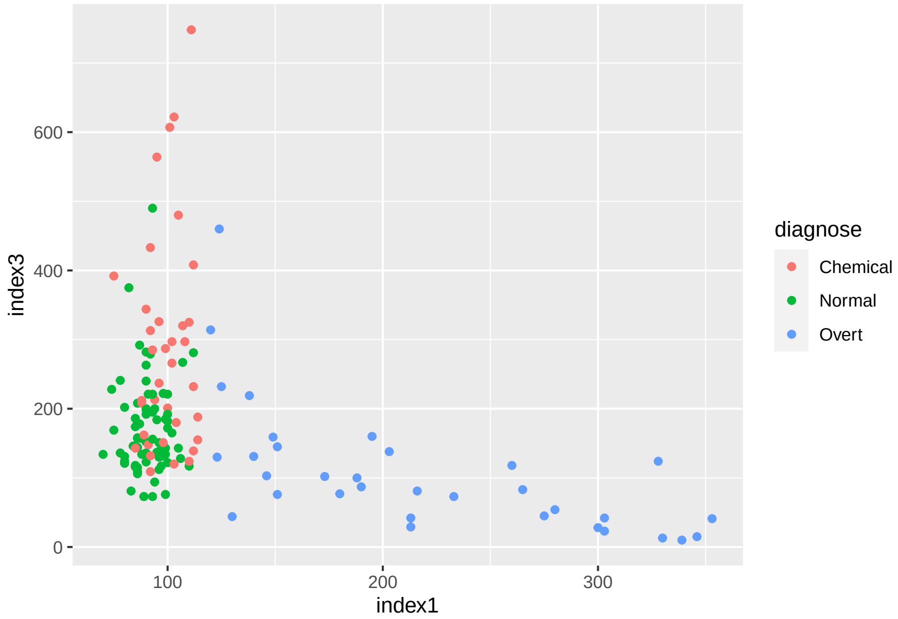
```
# 确认数据是否有缺失值
```

3 探索性数据分析

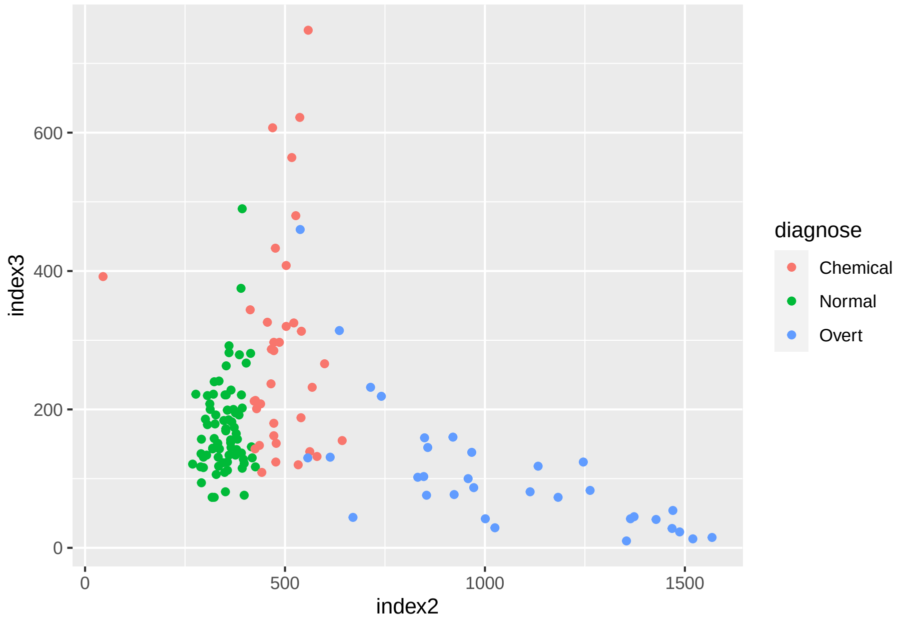
```
# 诊断结果作为分类依据，其他分别作为 x,y 坐标进行尝试探索
p1 <- ggplot(data,aes(color = diagnose))
p1 + geom_point(aes(x = index1,y = index2))
```



```
p1 + geom_point(aes(x = index1,y = index3))
```



```
p1 + geom_point(aes(x = index2,y = index3))
```



```
str(data)
```

```
## 'data.frame':    145 obs. of  4 variables:
##  $ diagnose: chr  "Normal" "Normal" "Normal" "Normal" ...
##  $ index1  : int  80 97 105 90 90 86 100 85 97 97 ...
##  $ index2  : int  356 289 319 356 323 381 350 301 379 296 ...
##  $ index3  : int  124 117 143 199 240 157 221 186 142 131 ...
```

```
summary(data)
```

```
##    diagnose              index1         index2            index3
##  Length:145          Min.   : 70   Min.   :  45.0   Min.   : 10.0
##  Class :character   1st Qu.: 90   1st Qu.: 352.0   1st Qu.:118.0
##  Mode  :character   Median : 97   Median : 403.0   Median :156.0
##                      Mean   :122   Mean   : 540.8   Mean   :186.1
##                      3rd Qu.:112   3rd Qu.: 558.0   3rd Qu.:221.0
##                      Max.   :353   Max.   :1568.0   Max.   :748.0
```

```
View(data)
#
p1 <- plot_ly(data, x = data[, 2], y =data[, 3], z = data[, 4],
             color = ~data$diagnose,
             marker = list(size = 3)) %>% add_markers()
p1
```

4 KNN

4.1 模型构建

```
# 创建训练任务-task
task <- makeClassifTask(data = data, target = "diagnose")
```

```
# 选择学习算法-learner
learner<- makeLearner("classif.knn", par.vals=list("k" = 5))
# 训练模型-train
model <- train(learner,task)
```

4.2 预测与评估

```
newdata <- data
pred <- predict(model, newdata = newdata)
pred
```

```
## Prediction: 145 observations
## predict.type: response
## threshold:
## time: 0.00
##    truth response
## 1 Normal   Normal
## 2 Normal   Normal
## 3 Normal   Normal
## 4 Normal   Normal
## 5 Normal   Normal
## 6 Normal   Normal
## ... (#rows: 145, #cols: 2)
```

```
calculateConfusionMatrix(pred)  # 计算混淆矩阵
```

```
##           predicted
## true       Chemical Normal Overt -err.-
##   Chemical       32      4     0      4
##   Normal          1     75     0      1
##   Overt           4      0    29      4
##   -err.-          5      4     0      9
```

```
performance(pred, measures=list(mmce,acc))
```

```
##       mmce        acc
## 0.06206897 0.93793103
```

4.3 调参

4.3.1 留出法 (Holdout CV)

```
cv.holdout <- makeResampleDesc(method="Holdout",
                               split=0.7,
                               stratify=T)
# 0.7 分割比例
resa.holdout <- resample(learner = learner,
                         task = task,
                         resampling=cv.holdout,
                         measures=list(mmce,acc))
```

```
## Resampling: holdout

## Measures:             mmce      acc

## [Resample] iter 1:    0.0681818 0.9318182

##

## Aggregated Result: mmce.test.mean=0.0681818,acc.test.mean=0.9318182

##
```

```
resa.holdout$aggr
```

```
## mmce.test.mean  acc.test.mean
##     0.06818182     0.93181818
```

4.3.2 K 折法 (k-Fold CV)

```
cv.10fold <- makeResampleDesc(method="CV",iters = 10,
                              stratify = T)
# 10 折
resa.10fold <- resample(learner = learner,
                        task = task,
                        cv.10fold,list(mmce,acc))
```

```
## Resampling: cross-validation

## Measures:             mmce      acc

## [Resample] iter 1:    0.0625000 0.9375000

## [Resample] iter 2:    0.0000000 1.0000000

## [Resample] iter 3:    0.0666667 0.9333333

## [Resample] iter 4:    0.0000000 1.0000000

## [Resample] iter 5:    0.2000000 0.8000000

## [Resample] iter 6:    0.0000000 1.0000000

## [Resample] iter 7:    0.0000000 1.0000000

## [Resample] iter 8:    0.0714286 0.9285714

## [Resample] iter 9:    0.2000000 0.8000000

## [Resample] iter 10:   0.1428571 0.8571429

##
```

```
## Aggregated Result: mmce.test.mean=0.0743452,acc.test.mean=0.9256548

##
```

```
resa.10fold$aggr
```

```
## mmce.test.mean  acc.test.mean
##     0.07434524     0.92565476
```

另一种实现，并非传统 mlr 实现

```
set.seed(1)
index = round(nrow(data) * 0.2,digits = 0) # 82 开训练集和测试集
test.indices = sample(1:nrow(data), index)
data.train=data[-test.indices,] # 切分训练集测试集
data.test=data[test.indices,]
YTrain = data.train$diagnose # 自变量因变量指定
XTrain = data.train %>% select(-diagnose)
YTest = data.test$diagnose
XTest = data.test %>% select(-diagnose)
```

```
calc_error_rate <- function(predicted.value, true.value){
 return(mean(true.value!=predicted.value)) # 定义错误函数
}
```

```
nfold = 10 # 10 折，与之前的实现保持一致
set.seed(1)
folds = seq.int(nrow(data.train)) %>% cut(breaks = nfold, labels=FALSE) %>% sample
do.chunk <- function(chunkid, folddef, Xdat, Ydat, k){
     train = (folddef!=chunkid)
Xtr = Xdat[train,] # 训练集
Ytr = Ydat[train]
Xvl = Xdat[!train,] # 测试集
Yvl = Ydat[!train]
predYtr = knn(train = Xtr, test = Xtr, cl = Ytr, k = k) # 预测训练集标签
predYvl = knn(train = Xtr, test = Xvl, cl = Ytr, k = k) # 预测测试集标签
data.frame(fold =chunkid, train.error = calc_error_rate(predYtr, Ytr),# k 折, 每折训练误
 val.error = calc_error_rate(predYvl, Yvl))} # 每折测试误差
error.folds=NULL # 存储 validation error
kvec = c(1, seq(10, 50, length.out=5)) # 创建间隔为 10 的序列
set.seed(1)
for (j in kvec){
 tmp = ldply(1:nfold, do.chunk, folddef=folds, Xdat=XTrain, Ydat=YTrain, k=j) # 对每折应
 tmp$neighbors = j
 error.folds = rbind(error.folds, tmp) # 组合结果
 }
# 将宽数据变成长数据
errors = melt(error.folds, id.vars=c("fold","neighbors"), value.name= "error")
```

```
val.error.means = errors %>% # 选中所有行
    filter(variable== "val.error" ) %>% # 分组
    group_by(neighbors, variable) %>% # 计算 CV error
    summarise_each(funs(mean), error) %>%
    ungroup() %>%
    filter(error==min(error))
```

```
## Warning: `summarise_each_()` was deprecated in dplyr 0.7.0.
## Please use `across()` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was generated.

## Warning: `funs()` was deprecated in dplyr 0.8.0.
## Please use a list of either functions or lambdas:
##
##   # Simple named list:
##   list(mean = mean, median = median)
##
```

```
##   # Auto named with `tibble::lst()`:
##   tibble::lst(mean, median)
##
##   # Using lambdas
##   list(~ mean(., trim = .2), ~ median(., na.rm = TRUE))
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was generated.
```

```
# 最佳近邻个数
numneighbor = max(val.error.means$neighbors)
```

```
set.seed(20)
pred.YTtrain = knn(train = XTrain, test = XTrain, cl = YTrain, k = 20)
knn_traing_error <- calc_error_rate(predicted.value=pred.YTtrain, true.value=YTrain)
knn_traing_error # 训练误差
```

```
## [1] 0.1206897
```

```
data <- data[complete.cases(data[,0:3]),]
set.seed(20)
pred.YTest = knn(train=XTrain, test=XTest, cl=YTrain, k=20)
knn_test_error <- calc_error_rate(predicted.value=pred.YTest, true.value=YTest)
knn_test_error # 测试误差
```

```
## [1] 0.1034483
```

```
conf.matrix = table(predicted=pred.YTest, true=YTest)
conf.matrix # 混淆矩阵
```

```
##            true
## predicted  Chemical Normal Overt
##   Chemical        4      0     1
##   Normal          2     18     0
##   Overt           0      0     4
```

```
sum(diag(conf.matrix)/sum(conf.matrix)) # 正确率
```

```
## [1] 0.8965517
```

```
# Test error rate
1 - sum(diag(conf.matrix)/sum(conf.matrix))
```

```
## [1] 0.1034483
```

```
# 绘制 ROC 曲线并计算 AUC 值
library(pROC)
```

```
## Warning: 程辑包'pROC'是用R版本4.1.3 来建造的
```

```
## Type 'citation("pROC")' for a citation.
```

```
##
## 载入程辑包：'pROC'
```

```
## The following objects are masked from 'package:stats':
##
##     cov, smooth, var
```

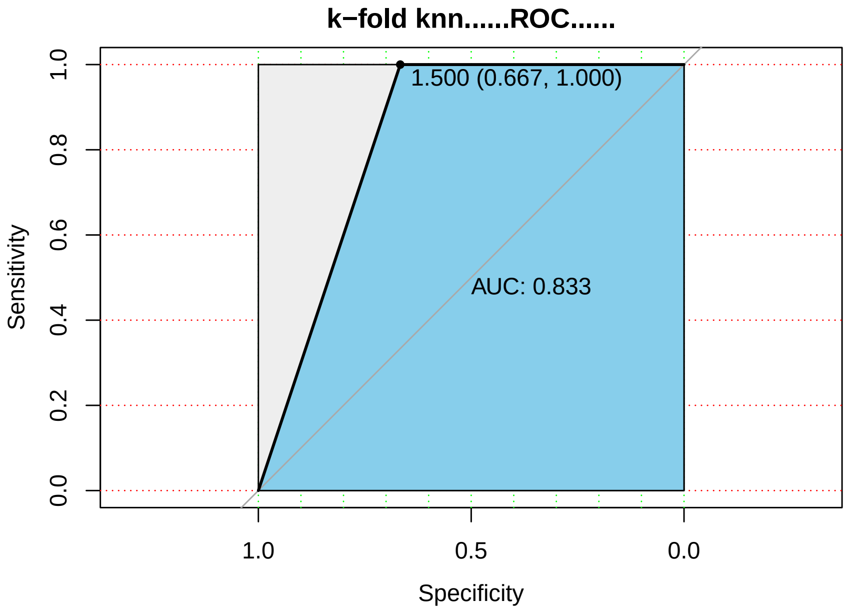
```
knn_roc <- roc(data.test$diagnose,as.numeric(pred.YTest))
```

```
## Warning in roc.default(data.test$diagnose, as.numeric(pred.YTest)): 'response'
## has more than two levels. Consider setting 'levels' explicitly or using
## 'multiclass.roc' instead
```

```
## Setting levels: control = Chemical, case = Normal
```

```
## Setting direction: controls < cases
```

```
plot(knn_roc, print.auc=TRUE, auc.polygon=TRUE, grid=c(0.1, 0.2),grid.col=c("green", "r
```



4.3.3 重复 k 折法 (Repeat k-Fold CV)

```
cvRepKFold <-makeResampleDesc(method="RepCV",folds=12,reps=2,stratify=T)
resa.rep12fold<-resample(learner,task,cvRepKFold,measures=list(mmce,acc))
```

```
## Resampling: repeated cross-validation

## Measures:              mmce      acc

## [Resample] iter 1:     0.0000000 1.0000000

## [Resample] iter 2:     0.0000000 1.0000000

## [Resample] iter 3:     0.1538462 0.8461538
## [Resample] iter 4:    0.0000000 1.0000000

## [Resample] iter 5:    0.0833333 0.9166667

## [Resample] iter 6:    0.2307692 0.7692308

## [Resample] iter 7:    0.0833333 0.9166667

## [Resample] iter 8:    0.0833333 0.9166667

## [Resample] iter 9:    0.1666667 0.8333333

## [Resample] iter 10:   0.0909091 0.9090909

## [Resample] iter 11:   0.0000000 1.0000000

## [Resample] iter 12:   0.0000000 1.0000000

## [Resample] iter 13:   0.0833333 0.9166667

## [Resample] iter 14:   0.0000000 1.0000000

## [Resample] iter 15:   0.0769231 0.9230769

## [Resample] iter 16:   0.1666667 0.8333333

## [Resample] iter 17:   0.0000000 1.0000000

## [Resample] iter 18:   0.0000000 1.0000000

## [Resample] iter 19:   0.1538462 0.8461538

## [Resample] iter 20:   0.0000000 1.0000000

## [Resample] iter 21:   0.0833333 0.9166667
## [Resample] iter 22:   0.0000000 1.0000000

## [Resample] iter 23:   0.2500000 0.7500000

## [Resample] iter 24:   0.0909091 0.9090909

##

## Aggregated Result: mmce.test.mean=0.0748834,acc.test.mean=0.9251166

##
```

```
calculateConfusionMatrix(resa.rep12fold$pred, relative=T)
```

```
## Relative confusion matrix (normalized by row/column):
##           predicted
## true       Chemical  Normal    Overt     -err.-
##   Chemical 0.86/0.84 0.12/0.06 0.01/0.02 0.14
##   Normal   0.01/0.03 0.99/0.94 0.00/0.00 0.01
##   Overt    0.15/0.14 0.00/0.00 0.85/0.98 0.15
##   -err.-        0.16      0.06      0.02 0.08
##
##
## Absolute confusion matrix:
##           predicted
## true       Chemical Normal Overt -err.-
##   Chemical       62      9     1     10
##   Normal          2    150     0      2
##   Overt          10      0    56     10
##   -err.-         12      9     1     22
```

4.3.4 留一法 (Leave-One-Out CV)

```
cv.loo <- makeResampleDesc(method="LOO")
resa.loo <- resample(learner,
                     task,
                     cv.loo,
                     list(mmce,acc))
```

```
## Resampling: LOO

## Measures:             mmce      acc

## [Resample] iter 1:    0.0000000 1.0000000

## [Resample] iter 2:    0.0000000 1.0000000

## [Resample] iter 3:    0.0000000 1.0000000

## [Resample] iter 4:    0.0000000 1.0000000

## [Resample] iter 5:    0.0000000 1.0000000

## [Resample] iter 6:    0.0000000 1.0000000

## [Resample] iter 7:    0.0000000 1.0000000

## [Resample] iter 8:    0.0000000 1.0000000

## [Resample] iter 9:    0.0000000 1.0000000

## [Resample] iter 10:   0.0000000 1.0000000

## [Resample] iter 11:   0.0000000 1.0000000

## [Resample] iter 12:   0.0000000 1.0000000
## [Resample] iter 13:   0.0000000 1.0000000

## [Resample] iter 14:   0.0000000 1.0000000

## [Resample] iter 15:   0.0000000 1.0000000

## [Resample] iter 16:   0.0000000 1.0000000

## [Resample] iter 17:   0.0000000 1.0000000

## [Resample] iter 18:   0.0000000 1.0000000

## [Resample] iter 19:   0.0000000 1.0000000

## [Resample] iter 20:   0.0000000 1.0000000

## [Resample] iter 21:   0.0000000 1.0000000

## [Resample] iter 22:   0.0000000 1.0000000

## [Resample] iter 23:   0.0000000 1.0000000

## [Resample] iter 24:   0.0000000 1.0000000

## [Resample] iter 25:   0.0000000 1.0000000

## [Resample] iter 26:   0.0000000 1.0000000

## [Resample] iter 27:   0.0000000 1.0000000

## [Resample] iter 28:   0.0000000 1.0000000

## [Resample] iter 29:   0.0000000 1.0000000

## [Resample] iter 30:   0.0000000 1.0000000
## [Resample] iter 31:   0.0000000 1.0000000

## [Resample] iter 32:   0.0000000 1.0000000

## [Resample] iter 33:   0.0000000 1.0000000

## [Resample] iter 34:   0.0000000 1.0000000

## [Resample] iter 35:   0.0000000 1.0000000

## [Resample] iter 36:   0.0000000 1.0000000

## [Resample] iter 37:   0.0000000 1.0000000

## [Resample] iter 38:   0.0000000 1.0000000

## [Resample] iter 39:   0.0000000 1.0000000

## [Resample] iter 40:   0.0000000 1.0000000

## [Resample] iter 41:   0.0000000 1.0000000

## [Resample] iter 42:   0.0000000 1.0000000

## [Resample] iter 43:   0.0000000 1.0000000

## [Resample] iter 44:   0.0000000 1.0000000

## [Resample] iter 45:   0.0000000 1.0000000

## [Resample] iter 46:   0.0000000 1.0000000

## [Resample] iter 47:   0.0000000 1.0000000

## [Resample] iter 48:   0.0000000 1.0000000
## [Resample] iter 49:   0.0000000 1.0000000

## [Resample] iter 50:   0.0000000 1.0000000

## [Resample] iter 51:   0.0000000 1.0000000

## [Resample] iter 52:   0.0000000 1.0000000

## [Resample] iter 53:   0.0000000 1.0000000

## [Resample] iter 54:   0.0000000 1.0000000

## [Resample] iter 55:   0.0000000 1.0000000

## [Resample] iter 56:   0.0000000 1.0000000

## [Resample] iter 57:   0.0000000 1.0000000

## [Resample] iter 58:   0.0000000 1.0000000

## [Resample] iter 59:   0.0000000 1.0000000

## [Resample] iter 60:   0.0000000 1.0000000

## [Resample] iter 61:   0.0000000 1.0000000

## [Resample] iter 62:   0.0000000 1.0000000

## [Resample] iter 63:   0.0000000 1.0000000

## [Resample] iter 64:   0.0000000 1.0000000

## [Resample] iter 65:   0.0000000 1.0000000

## [Resample] iter 66:   1.0000000 0.0000000
## [Resample] iter 67:   0.0000000 1.0000000

## [Resample] iter 68:   0.0000000 1.0000000

## [Resample] iter 69:   0.0000000 1.0000000

## [Resample] iter 70:   0.0000000 1.0000000

## [Resample] iter 71:   1.0000000 0.0000000

## [Resample] iter 72:   0.0000000 1.0000000

## [Resample] iter 73:   0.0000000 1.0000000

## [Resample] iter 74:   0.0000000 1.0000000

## [Resample] iter 75:   0.0000000 1.0000000

## [Resample] iter 76:   0.0000000 1.0000000

## [Resample] iter 77:   0.0000000 1.0000000

## [Resample] iter 78:   0.0000000 1.0000000

## [Resample] iter 79:   0.0000000 1.0000000

## [Resample] iter 80:   0.0000000 1.0000000

## [Resample] iter 81:   0.0000000 1.0000000

## [Resample] iter 82:   1.0000000 0.0000000

## [Resample] iter 83:   1.0000000 0.0000000

## [Resample] iter 84:   0.0000000 1.0000000
## [Resample] iter 85:   0.0000000 1.0000000

## [Resample] iter 86:   0.0000000 1.0000000

## [Resample] iter 87:   0.0000000 1.0000000

## [Resample] iter 88:   0.0000000 1.0000000

## [Resample] iter 89:   0.0000000 1.0000000

## [Resample] iter 90:   0.0000000 1.0000000

## [Resample] iter 91:   0.0000000 1.0000000

## [Resample] iter 92:   0.0000000 1.0000000

## [Resample] iter 93:   0.0000000 1.0000000

## [Resample] iter 94:   0.0000000 1.0000000

## [Resample] iter 95:   0.0000000 1.0000000

## [Resample] iter 96:   0.0000000 1.0000000

## [Resample] iter 97:   0.0000000 1.0000000

## [Resample] iter 98:   0.0000000 1.0000000

## [Resample] iter 99:   0.0000000 1.0000000

## [Resample] iter 100:  0.0000000 1.0000000

## [Resample] iter 101:  0.0000000 1.0000000

## [Resample] iter 102:  1.0000000 0.0000000
## [Resample] iter 103:  0.0000000 1.0000000

## [Resample] iter 104:  1.0000000 0.0000000

## [Resample] iter 105:  0.0000000 1.0000000

## [Resample] iter 106:  0.0000000 1.0000000

## [Resample] iter 107:  0.0000000 1.0000000

## [Resample] iter 108:  0.0000000 1.0000000

## [Resample] iter 109:  0.0000000 1.0000000

## [Resample] iter 110:  0.0000000 1.0000000

## [Resample] iter 111:  1.0000000 0.0000000

## [Resample] iter 112:  0.0000000 1.0000000

## [Resample] iter 113:  0.0000000 1.0000000

## [Resample] iter 114:  0.0000000 1.0000000

## [Resample] iter 115:  0.0000000 1.0000000

## [Resample] iter 116:  0.0000000 1.0000000

## [Resample] iter 117:  0.0000000 1.0000000

## [Resample] iter 118:  0.0000000 1.0000000

## [Resample] iter 119:  0.0000000 1.0000000

## [Resample] iter 120:  0.0000000 1.0000000
## [Resample] iter 121:  0.0000000 1.0000000

## [Resample] iter 122:  0.0000000 1.0000000

## [Resample] iter 123:  0.0000000 1.0000000

## [Resample] iter 124:  1.0000000 0.0000000

## [Resample] iter 125:  0.0000000 1.0000000

## [Resample] iter 126:  0.0000000 1.0000000

## [Resample] iter 127:  0.0000000 1.0000000

## [Resample] iter 128:  0.0000000 1.0000000

## [Resample] iter 129:  0.0000000 1.0000000

## [Resample] iter 130:  0.0000000 1.0000000

## [Resample] iter 131:  1.0000000 0.0000000

## [Resample] iter 132:  0.0000000 1.0000000

## [Resample] iter 133:  0.0000000 1.0000000

## [Resample] iter 134:  1.0000000 0.0000000

## [Resample] iter 135:  1.0000000 0.0000000

## [Resample] iter 136:  1.0000000 0.0000000

## [Resample] iter 137:  0.0000000 1.0000000

## [Resample] iter 138:  0.0000000 1.0000000
## [Resample] iter 139:  0.0000000 1.0000000

## [Resample] iter 140:  0.0000000 1.0000000

## [Resample] iter 141:  0.0000000 1.0000000

## [Resample] iter 142:  0.0000000 1.0000000

## [Resample] iter 143:  0.0000000 1.0000000

## [Resample] iter 144:  0.0000000 1.0000000

## [Resample] iter 145:  0.0000000 1.0000000

##

## Aggregated Result: mmce.test.mean=0.0827586,acc.test.mean=0.9172414

##
```

```
resa.loo$aggr
```

```
## mmce.test.mean  acc.test.mean
##     0.08275862     0.91724138
```

4.4 不同种方法的混淆矩阵

```
calculateConfusionMatrix(pred=resa.holdout$pred,
                         relative=T)
```

```
## Relative confusion matrix (normalized by row/column):
##           predicted
## true       Chemical  Normal    Overt     -err.-
##   Chemical 1.00/0.79 0.00/0.00 0.00/0.00 0.00
##   Normal   0.04/0.07 0.96/1.00 0.00/0.00 0.04
##   Overt    0.20/0.14 0.00/0.00 0.80/1.00 0.20
##   -err.-        0.21      0.00      0.00 0.07
##
##
## Absolute confusion matrix:
##             predicted
## true         Chemical Normal Overt -err.-
##   Chemical         11      0     0      0
##   Normal            1     22     0      1
##   Overt             2      0     8      2
##   -err.-            3      0     0      3
```

```
calculateConfusionMatrix(pred=resa.10fold$pred,
                         relative=T)
```

```
## Relative confusion matrix (normalized by row/column):
##             predicted
## true         Chemical  Normal    Overt     -err.-
##   Chemical 0.86/0.84 0.14/0.06 0.00/0.00 0.14
##   Normal   0.01/0.03 0.99/0.94 0.00/0.00 0.01
##   Overt    0.15/0.14 0.00/0.00 0.85/1.00 0.15
##   -err.-        0.16      0.06      0.00 0.08
##
##
## Absolute confusion matrix:
##             predicted
## true         Chemical Normal Overt -err.-
##   Chemical         31      5     0      5
##   Normal            1     75     0      1
##   Overt             5      0    28      5
##   -err.-            6      5     0     11
```

```
calculateConfusionMatrix(pred=resa.rep12fold$pred,
                         relative=T)
```

```
## Relative confusion matrix (normalized by row/column):
##           predicted
## true       Chemical  Normal    Overt     -err.-
##   Chemical 0.86/0.84 0.12/0.06 0.01/0.02 0.14
##   Normal   0.01/0.03 0.99/0.94 0.00/0.00 0.01
##   Overt    0.15/0.14 0.00/0.00 0.85/0.98 0.15
##   -err.-        0.16      0.06      0.02 0.08
##
##
## Absolute confusion matrix:
##           predicted
## true       Chemical Normal Overt -err.-
##   Chemical       62      9     1     10
##   Normal          2    150     0      2
##   Overt          10      0    56     10
##   -err.-         12      9     1     22
```

```
calculateConfusionMatrix(pred=resa.loo$pred,
                         relative=T)
```

```
## Relative confusion matrix (normalized by row/column):
##           predicted
## true       Chemical  Normal    Overt     -err.-
##   Chemical 0.83/0.83 0.14/0.06 0.03/0.03 0.17
##   Normal   0.01/0.03 0.99/0.94 0.00/0.00 0.01
##   Overt    0.15/0.14 0.00/0.00 0.85/0.97 0.15
##   -err.-        0.17      0.06      0.03 0.08
##
##
## Absolute confusion matrix:
```

```
##           predicted
## true       Chemical Normal Overt -err.-
##   Chemical       30      5     1      6
##   Normal          1     75     0      1
##   Overt           5      0    28      5
##   -err.-          6      5     1     12
```

得到结果为 holdout 方法最好，后面选用此方法作为 baseline 进行优化

4.5 提升模型性能

```
getParamSet("classif.knn")
```

```
##             Type len   Def   Constr Req Tunable Trafo
## k        integer   -     1 1 to Inf   -    TRUE     -
## l        numeric   -     0 0 to Inf   -    TRUE     -
## prob     logical   - FALSE        -   -   FALSE     -
## use.all logical   -  TRUE        -   -    TRUE     -
```

```
tuneHyperParmSet <- makeParamSet(makeDiscreteParam("k",1:20))
gridSearch<-makeTuneControlGrid()
tunedHyperParam<-tuneParams(learner, task,
                            resampling=cv.holdout, # 交叉验证
                            par.set=tuneHyperParmSet,
                            control=gridSearch) # 调参
```

```
## [Tune] Started tuning learner classif.knn for parameter set:
```

```
##       Type len Def                                  Constr Req Tunable Trafo
## k discrete   -   - 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,1...   -    TRUE     -
```

```
## With control class: TuneControlGrid
## Imputation value: 1

## [Tune-x] 1: k=1

## [Tune-y] 1: mmce.test.mean=0.0454545; time: 0.0 min

## [Tune-x] 2: k=2

## [Tune-y] 2: mmce.test.mean=0.1136364; time: 0.0 min

## [Tune-x] 3: k=3

## [Tune-y] 3: mmce.test.mean=0.0454545; time: 0.0 min

## [Tune-x] 4: k=4

## [Tune-y] 4: mmce.test.mean=0.0909091; time: 0.0 min

## [Tune-x] 5: k=5

## [Tune-y] 5: mmce.test.mean=0.1136364; time: 0.0 min

## [Tune-x] 6: k=6

## [Tune-y] 6: mmce.test.mean=0.0909091; time: 0.0 min

## [Tune-x] 7: k=7

## [Tune-y] 7: mmce.test.mean=0.0909091; time: 0.0 min

## [Tune-x] 8: k=8

## [Tune-y] 8: mmce.test.mean=0.0909091; time: 0.0 min

## [Tune-x] 9: k=9
## [Tune-y] 9: mmce.test.mean=0.0909091; time: 0.0 min

## [Tune-x] 10: k=10

## [Tune-y] 10: mmce.test.mean=0.0909091; time: 0.0 min

## [Tune-x] 11: k=11

## [Tune-y] 11: mmce.test.mean=0.0909091; time: 0.0 min

## [Tune-x] 12: k=12

## [Tune-y] 12: mmce.test.mean=0.0909091; time: 0.0 min

## [Tune-x] 13: k=13

## [Tune-y] 13: mmce.test.mean=0.1136364; time: 0.0 min

## [Tune-x] 14: k=14

## [Tune-y] 14: mmce.test.mean=0.1363636; time: 0.0 min

## [Tune-x] 15: k=15

## [Tune-y] 15: mmce.test.mean=0.1363636; time: 0.0 min

## [Tune-x] 16: k=16

## [Tune-y] 16: mmce.test.mean=0.1363636; time: 0.0 min

## [Tune-x] 17: k=17

## [Tune-y] 17: mmce.test.mean=0.1363636; time: 0.0 min

## [Tune-x] 18: k=18
## [Tune-y] 18: mmce.test.mean=0.1363636; time: 0.0 min
```

```
## [Tune-x] 19: k=19
```

```
## [Tune-y] 19: mmce.test.mean=0.1363636; time: 0.0 min
```

```
## [Tune-x] 20: k=20
```

```
## [Tune-y] 20: mmce.test.mean=0.1363636; time: 0.0 min
```

```
## [Tune] Result: k=3 : mmce.test.mean=0.0454545
```

```
tunedHyperParam$x # 通过 $x 获取调参后的超参值
```

```
## $k
## [1] 3
```

k = 7 最好

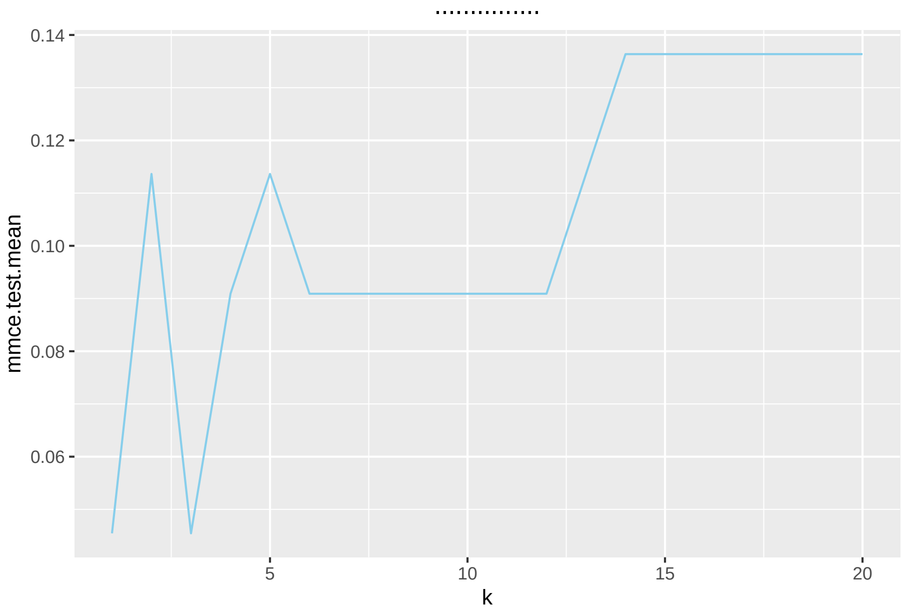
4.6 可视化调参过程

```
tuningData <- generateHyperParsEffectData(tunedHyperParam)
p <- ggplot(tuningData$data, aes(x=k, y=mmce.test.mean)) # 指定 x,y 坐标
p + geom_line(color="skyblue") + labs(title = " 可视化调参") + theme(plot.title = elemen
```

```
## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## 'mbcsToSbcs'里转换'鍙 鍖栬皟鍙 '出错: <e5>代替了dot
```

```
## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## 'mbcsToSbcs'里转换'鍙 鍖栬皟鍙 '出错: <8f>代替了dot
```

```
## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## 'mbcsToSbcs'里转换'鍙 鍖栬皟鍙 '出错: <af>代替了dot
## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## 'mbcsToSbcs'里转换'鎏 鍖栬皟鎏'出错：<e8>代替了dot

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## 'mbcsToSbcs'里转换'鎏 鍖栬皟鎏'出错：<a7>代替了dot

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## 'mbcsToSbcs'里转换'鎏 鍖栬皟鎏'出错：<86>代替了dot

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## 'mbcsToSbcs'里转换'鎏 鍖栬皟鎏'出错：<e5>代替了dot

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## 'mbcsToSbcs'里转换'鎏 鍖栬皟鎏'出错：<8c>代替了dot

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## 'mbcsToSbcs'里转换'鎏 鍖栬皟鎏'出错：<96>代替了dot

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## 'mbcsToSbcs'里转换'鎏 鍖栬皟鎏'出错：<e8>代替了dot

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## 'mbcsToSbcs'里转换'鎏 鍖栬皟鎏'出错：<b0>代替了dot

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## 'mbcsToSbcs'里转换'鎏 鍖栬皟鎏'出错：<83>代替了dot

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## 'mbcsToSbcs'里转换'鎏 鍖栬皟鎏'出错：<e5>代替了dot

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## 'mbcsToSbcs'里转换'鎏 鍖栬皟鎏'出错：<8f>代替了dot

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## 'mbcsToSbcs'里转换'鎏 鍖栬皟鎏'出错：<82>代替了dot
## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## 'mbcsToSbcs'里转换'鍙 鍖栬賾鍙 '出错: <e5>代替了dot

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## 'mbcsToSbcs'里转换'鍙 鍖栬賾鍙 '出错: <8f>代替了dot

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## 'mbcsToSbcs'里转换'鍙 鍖栬賾鍙 '出错: <af>代替了dot

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## 'mbcsToSbcs'里转换'鍙 鍖栬賾鍙 '出错: <e8>代替了dot

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## 'mbcsToSbcs'里转换'鍙 鍖栬賾鍙 '出错: <a7>代替了dot

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## 'mbcsToSbcs'里转换'鍙 鍖栬賾鍙 '出错: <86>代替了dot

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## 'mbcsToSbcs'里转换'鍙 鍖栬賾鍙 '出错: <e5>代替了dot

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## 'mbcsToSbcs'里转换'鍙 鍖栬賾鍙 '出错: <8c>代替了dot

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## 'mbcsToSbcs'里转换'鍙 鍖栬賾鍙 '出错: <96>代替了dot

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## 'mbcsToSbcs'里转换'鍙 鍖栬賾鍙 '出错: <e8>代替了dot

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## 'mbcsToSbcs'里转换'鍙 鍖栬賾鍙 '出错: <b0>代替了dot

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## 'mbcsToSbcs'里转换'鍙 鍖栬賾鍙 '出错: <83>代替了dot
## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## 'mbcsToSbcs'里转换'鎏 鍖栬皟鎏 '出错: <e5>代替了dot

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## 'mbcsToSbcs'里转换'鎏 鍖栬皟鎏 '出错: <8f>代替了dot

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## 'mbcsToSbcs'里转换'鎏 鍖栬皟鎏 '出错: <82>代替了dot

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## 'mbcsToSbcs'里转换'鎏 鍖栬皟鎏 '出错: <e5>代替了dot

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## 'mbcsToSbcs'里转换'鎏 鍖栬皟鎏 '出错: <8f>代替了dot

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## 'mbcsToSbcs'里转换'鎏 鍖栬皟鎏 '出错: <af>代替了dot

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## 'mbcsToSbcs'里转换'鎏 鍖栬皟鎏 '出错: <e8>代替了dot

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## 'mbcsToSbcs'里转换'鎏 鍖栬皟鎏 '出错: <a7>代替了dot

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## 'mbcsToSbcs'里转换'鎏 鍖栬皟鎏 '出错: <86>代替了dot

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## 'mbcsToSbcs'里转换'鎏 鍖栬皟鎏 '出错: <e5>代替了dot

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## 'mbcsToSbcs'里转换'鎏 鍖栬皟鎏 '出错: <8c>代替了dot

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## 'mbcsToSbcs'里转换'鎏 鍖栬皟鎏 '出错: <96>代替了dot
## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## 'mbcsToSbcs'里转换'鎏 鍖栬瞔鎏 '出错: <e8>代替了dot

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## 'mbcsToSbcs'里转换'鎏 鍖栬瞔鎏 '出错: <b0>代替了dot

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## 'mbcsToSbcs'里转换'鎏 鍖栬瞔鎏 '出错: <83>代替了dot

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## 'mbcsToSbcs'里转换'鎏 鍖栬瞔鎏 '出错: <e5>代替了dot

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## 'mbcsToSbcs'里转换'鎏 鍖栬瞔鎏 '出错: <8f>代替了dot

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## 'mbcsToSbcs'里转换'鎏 鍖栬瞔鎏 '出错: <82>代替了dot

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## 'mbcsToSbcs'里转换'鎏 鍖栬瞔鎏 '出错: <e5>代替了dot

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## 'mbcsToSbcs'里转换'鎏 鍖栬瞔鎏 '出错: <8f>代替了dot

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## 'mbcsToSbcs'里转换'鎏 鍖栬瞔鎏 '出错: <af>代替了dot

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## 'mbcsToSbcs'里转换'鎏 鍖栬瞔鎏 '出错: <e8>代替了dot

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## 'mbcsToSbcs'里转换'鎏 鍖栬瞔鎏 '出错: <a7>代替了dot

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## 'mbcsToSbcs'里转换'鎏 鍖栬瞔鎏 '出错: <86>代替了dot
## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## 'mbcsToSbcs'里转换'鎏 鍖栬瞔鎏 '出错：<e5>代替了dot

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## 'mbcsToSbcs'里转换'鎏 鍖栬瞔鎏 '出错：<8c>代替了dot

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## 'mbcsToSbcs'里转换'鎏 鍖栬瞔鎏 '出错：<96>代替了dot

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## 'mbcsToSbcs'里转换'鎏 鍖栬瞔鎏 '出错：<e8>代替了dot

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## 'mbcsToSbcs'里转换'鎏 鍖栬瞔鎏 '出错：<b0>代替了dot

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## 'mbcsToSbcs'里转换'鎏 鍖栬瞔鎏 '出错：<83>代替了dot

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## 'mbcsToSbcs'里转换'鎏 鍖栬瞔鎏 '出错：<e5>代替了dot

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## 'mbcsToSbcs'里转换'鎏 鍖栬瞔鎏 '出错：<8f>代替了dot

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## 'mbcsToSbcs'里转换'鎏 鍖栬瞔鎏 '出错：<82>代替了dot

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## 'mbcsToSbcs'里转换'鎏 鍖栬瞔鎏 '出错：<e5>代替了dot

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## 'mbcsToSbcs'里转换'鎏 鍖栬瞔鎏 '出错：<8f>代替了dot

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## 'mbcsToSbcs'里转换'鎏 鍖栬瞔鎏 '出错：<af>代替了dot
## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## 'mbcsToSbcs'里转换'鍙 鍖栬瘑鍙 '出错：<e8>代替了dot

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## 'mbcsToSbcs'里转换'鍙 鍖栬瘑鍙 '出错：<a7>代替了dot

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## 'mbcsToSbcs'里转换'鍙 鍖栬瘑鍙 '出错：<86>代替了dot

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## 'mbcsToSbcs'里转换'鍙 鍖栬瘑鍙 '出错：<e5>代替了dot

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## 'mbcsToSbcs'里转换'鍙 鍖栬瘑鍙 '出错：<8c>代替了dot

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## 'mbcsToSbcs'里转换'鍙 鍖栬瘑鍙 '出错：<96>代替了dot

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## 'mbcsToSbcs'里转换'鍙 鍖栬瘑鍙 '出错：<e8>代替了dot

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## 'mbcsToSbcs'里转换'鍙 鍖栬瘑鍙 '出错：<b0>代替了dot

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## 'mbcsToSbcs'里转换'鍙 鍖栬瘑鍙 '出错：<83>代替了dot

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## 'mbcsToSbcs'里转换'鍙 鍖栬瘑鍙 '出错：<e5>代替了dot

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## 'mbcsToSbcs'里转换'鍙 鍖栬瘑鍙 '出错：<8f>代替了dot

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## 'mbcsToSbcs'里转换'鍙 鍖栬瘑鍙 '出错：<82>代替了dot
## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## 'mbcsToSbcs'里转换'鎏 錤栬瞔鎏 '出错：<e5>代替了dot

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## 'mbcsToSbcs'里转换'鎏 錤栬瞔鎏 '出错：<8f>代替了dot

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## 'mbcsToSbcs'里转换'鎏 錤栬瞔鎏 '出错：<af>代替了dot

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## 'mbcsToSbcs'里转换'鎏 錤栬瞔鎏 '出错：<e8>代替了dot

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## 'mbcsToSbcs'里转换'鎏 錤栬瞔鎏 '出错：<a7>代替了dot

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## 'mbcsToSbcs'里转换'鎏 錤栬瞔鎏 '出错：<86>代替了dot

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## 'mbcsToSbcs'里转换'鎏 錤栬瞔鎏 '出错：<e5>代替了dot

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## 'mbcsToSbcs'里转换'鎏 錤栬瞔鎏 '出错：<8c>代替了dot

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## 'mbcsToSbcs'里转换'鎏 錤栬瞔鎏 '出错：<96>代替了dot

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## 'mbcsToSbcs'里转换'鎏 錤栬瞔鎏 '出错：<e8>代替了dot

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## 'mbcsToSbcs'里转换'鎏 錤栬瞔鎏 '出错：<b0>代替了dot

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## 'mbcsToSbcs'里转换'鎏 錤栬瞔鎏 '出错：<83>代替了dot
## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## 'mbcsToSbcs'里转换'鎏 鍖栬瞔鎏 '出错：<e5>代替了dot

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## 'mbcsToSbcs'里转换'鎏 鍖栬瞔鎏 '出错：<8f>代替了dot

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## 'mbcsToSbcs'里转换'鎏 鍖栬瞔鎏 '出错：<82>代替了dot

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## 'mbcsToSbcs'里转换'鎏 鍖栬瞔鎏 '出错：<e5>代替了dot

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## 'mbcsToSbcs'里转换'鎏 鍖栬瞔鎏 '出错：<8f>代替了dot

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## 'mbcsToSbcs'里转换'鎏 鍖栬瞔鎏 '出错：<af>代替了dot

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## 'mbcsToSbcs'里转换'鎏 鍖栬瞔鎏 '出错：<e8>代替了dot

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## 'mbcsToSbcs'里转换'鎏 鍖栬瞔鎏 '出错：<a7>代替了dot

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## 'mbcsToSbcs'里转换'鎏 鍖栬瞔鎏 '出错：<86>代替了dot

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## 'mbcsToSbcs'里转换'鎏 鍖栬瞔鎏 '出错：<e5>代替了dot

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## 'mbcsToSbcs'里转换'鎏 鍖栬瞔鎏 '出错：<8c>代替了dot

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## 'mbcsToSbcs'里转换'鎏 鍖栬瞔鎏 '出错：<96>代替了dot
## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## 'mbcsToSbcs'里转换'鍙 鍖栬嵂鍙 '出错：<e8>代替了dot

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## 'mbcsToSbcs'里转换'鍙 鍖栬嵂鍙 '出错：<b0>代替了dot

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## 'mbcsToSbcs'里转换'鍙 鍖栬嵂鍙 '出错：<83>代替了dot

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## 'mbcsToSbcs'里转换'鍙 鍖栬嵂鍙 '出错：<e5>代替了dot

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## 'mbcsToSbcs'里转换'鍙 鍖栬嵂鍙 '出错：<8f>代替了dot

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## 'mbcsToSbcs'里转换'鍙 鍖栬嵂鍙 '出错：<82>代替了dot

## Warning in grid.Call.graphics(C_text, as.graphicsAnnot(x$label), x$x, x$y, :
## 'mbcsToSbcs'里转换'鍙 鍖栬嵂鍙 '出错：<e5>代替了dot

## Warning in grid.Call.graphics(C_text, as.graphicsAnnot(x$label), x$x, x$y, :
## 'mbcsToSbcs'里转换'鍙 鍖栬嵂鍙 '出错：<8f>代替了dot

## Warning in grid.Call.graphics(C_text, as.graphicsAnnot(x$label), x$x, x$y, :
## 'mbcsToSbcs'里转换'鍙 鍖栬嵂鍙 '出错：<af>代替了dot

## Warning in grid.Call.graphics(C_text, as.graphicsAnnot(x$label), x$x, x$y, :
## 'mbcsToSbcs'里转换'鍙 鍖栬嵂鍙 '出错：<e8>代替了dot

## Warning in grid.Call.graphics(C_text, as.graphicsAnnot(x$label), x$x, x$y, :
## 'mbcsToSbcs'里转换'鍙 鍖栬嵂鍙 '出错：<a7>代替了dot

## Warning in grid.Call.graphics(C_text, as.graphicsAnnot(x$label), x$x, x$y, :
## 'mbcsToSbcs'里转换'鍙 鍖栬嵂鍙 '出错：<86>代替了dot
## Warning in grid.Call.graphics(C_text, as.graphicsAnnot(x$label), x$x, x$y, :
## 'mbcsToSbcs'里转换'鎏 鍖栬瞔鎏 '出错：<e5>代替了dot

## Warning in grid.Call.graphics(C_text, as.graphicsAnnot(x$label), x$x, x$y, :
## 'mbcsToSbcs'里转换'鎏 鍖栬瞔鎏 '出错：<8c>代替了dot

## Warning in grid.Call.graphics(C_text, as.graphicsAnnot(x$label), x$x, x$y, :
## 'mbcsToSbcs'里转换'鎏 鍖栬瞔鎏 '出错：<96>代替了dot

## Warning in grid.Call.graphics(C_text, as.graphicsAnnot(x$label), x$x, x$y, :
## 'mbcsToSbcs'里转换'鎏 鍖栬瞔鎏 '出错：<e8>代替了dot

## Warning in grid.Call.graphics(C_text, as.graphicsAnnot(x$label), x$x, x$y, :
## 'mbcsToSbcs'里转换'鎏 鍖栬瞔鎏 '出错：<b0>代替了dot

## Warning in grid.Call.graphics(C_text, as.graphicsAnnot(x$label), x$x, x$y, :
## 'mbcsToSbcs'里转换'鎏 鍖栬瞔鎏 '出错：<83>代替了dot

## Warning in grid.Call.graphics(C_text, as.graphicsAnnot(x$label), x$x, x$y, :
## 'mbcsToSbcs'里转换'鎏 鍖栬瞔鎏 '出错：<e5>代替了dot

## Warning in grid.Call.graphics(C_text, as.graphicsAnnot(x$label), x$x, x$y, :
## 'mbcsToSbcs'里转换'鎏 鍖栬瞔鎏 '出错：<8f>代替了dot

## Warning in grid.Call.graphics(C_text, as.graphicsAnnot(x$label), x$x, x$y, :
## 'mbcsToSbcs'里转换'鎏 鍖栬瞔鎏 '出错：<82>代替了dot
```



4.7 重新训练并预测

```
tunedKnn <- setHyperPars(learner, par.vals=tunedHyperParam$x)
tunedModel <- train(tunedKnn, task) # 训练得到最终模型
pred1 <- predict(tunedModel, newdata=newdata) # 预测结果
calculateConfusionMatrix(pred)
```

```
##           predicted
## true        Chemical Normal Overt -err.-
##    Chemical       32      4     0      4
##    Normal          1     75     0      1
##    Overt           4      0    29      4
##    -err.-          5      4     0      9
```

```
# 引入 control
search.grid <- makeTuneControlGrid()
cv.inner <- makeResampleDesc("CV",stratify=T)
cv.outer <- makeResampleDesc("RepCV",
                             folds=12,
                             reps=5,
                             stratify=T)

wrapper.kNN <- makeTuneWrapper(learner="classif.knn",
                               resampling=cv.inner,
                               par.set=tuneHyperParmSet,
                               control=search.grid)

resa.tune.embeded.cv <- resample(learner=wrapper.kNN,
                                 task,
                                 resampling=cv.outer,
                                 measures=list(mmce,acc))
```

```
## Resampling: repeated cross-validation

## Measures:             mmce      acc

## [Tune] Started tuning learner classif.knn for parameter set:

##       Type len Def                                     Constr Req Tunable Trafo
## k discrete   -   - 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,1...   -    TRUE     -

## With control class: TuneControlGrid

## Imputation value: 1

## [Tune-x] 1: k=1

## [Tune-y] 1: mmce.test.mean=0.1054945; time: 0.0 min

## [Tune-x] 2: k=2
## [Tune-y] 2: mmce.test.mean=0.0912088; time: 0.0 min

## [Tune-x] 3: k=3

## [Tune-y] 3: mmce.test.mean=0.0835165; time: 0.0 min

## [Tune-x] 4: k=4

## [Tune-y] 4: mmce.test.mean=0.0758242; time: 0.0 min

## [Tune-x] 5: k=5

## [Tune-y] 5: mmce.test.mean=0.0835165; time: 0.0 min

## [Tune-x] 6: k=6

## [Tune-y] 6: mmce.test.mean=0.0681319; time: 0.0 min

## [Tune-x] 7: k=7

## [Tune-y] 7: mmce.test.mean=0.0758242; time: 0.0 min

## [Tune-x] 8: k=8

## [Tune-y] 8: mmce.test.mean=0.0989011; time: 0.0 min

## [Tune-x] 9: k=9

## [Tune-y] 9: mmce.test.mean=0.0835165; time: 0.0 min

## [Tune-x] 10: k=10

## [Tune-y] 10: mmce.test.mean=0.0752747; time: 0.0 min

## [Tune-x] 11: k=11
## [Tune-y] 11: mmce.test.mean=0.0906593; time: 0.0 min

## [Tune-x] 12: k=12

## [Tune-y] 12: mmce.test.mean=0.0758242; time: 0.0 min

## [Tune-x] 13: k=13

## [Tune-y] 13: mmce.test.mean=0.0829670; time: 0.0 min

## [Tune-x] 14: k=14

## [Tune-y] 14: mmce.test.mean=0.0829670; time: 0.0 min

## [Tune-x] 15: k=15

## [Tune-y] 15: mmce.test.mean=0.0906593; time: 0.0 min

## [Tune-x] 16: k=16

## [Tune-y] 16: mmce.test.mean=0.0983516; time: 0.0 min

## [Tune-x] 17: k=17

## [Tune-y] 17: mmce.test.mean=0.1060440; time: 0.0 min

## [Tune-x] 18: k=18

## [Tune-y] 18: mmce.test.mean=0.1060440; time: 0.0 min

## [Tune-x] 19: k=19

## [Tune-y] 19: mmce.test.mean=0.1131868; time: 0.0 min

## [Tune-x] 20: k=20
## [Tune-y] 20: mmce.test.mean=0.1054945; time: 0.0 min

## [Tune] Result: k=6 : mmce.test.mean=0.0681319

## [Resample] iter 1:    0.0833333 0.9166667

## [Tune] Started tuning learner classif.knn for parameter set:

##       Type len Def                                   Constr Req Tunable Trafo
## k discrete   -   - 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,1...   -    TRUE     -

## With control class: TuneControlGrid

## Imputation value: 1

## [Tune-x] 1: k=1

## [Tune-y] 1: mmce.test.mean=0.1351648; time: 0.0 min

## [Tune-x] 2: k=2

## [Tune-y] 2: mmce.test.mean=0.1197802; time: 0.0 min

## [Tune-x] 3: k=3

## [Tune-y] 3: mmce.test.mean=0.1126374; time: 0.0 min

## [Tune-x] 4: k=4

## [Tune-y] 4: mmce.test.mean=0.1049451; time: 0.0 min

## [Tune-x] 5: k=5

## [Tune-y] 5: mmce.test.mean=0.0824176; time: 0.0 min
## [Tune-x] 6: k=6

## [Tune-y] 6: mmce.test.mean=0.0978022; time: 0.0 min

## [Tune-x] 7: k=7

## [Tune-y] 7: mmce.test.mean=0.0978022; time: 0.0 min

## [Tune-x] 8: k=8

## [Tune-y] 8: mmce.test.mean=0.0901099; time: 0.0 min

## [Tune-x] 9: k=9

## [Tune-y] 9: mmce.test.mean=0.0901099; time: 0.0 min

## [Tune-x] 10: k=10

## [Tune-y] 10: mmce.test.mean=0.1054945; time: 0.0 min

## [Tune-x] 11: k=11

## [Tune-y] 11: mmce.test.mean=0.1054945; time: 0.0 min

## [Tune-x] 12: k=12

## [Tune-y] 12: mmce.test.mean=0.0978022; time: 0.0 min

## [Tune-x] 13: k=13

## [Tune-y] 13: mmce.test.mean=0.0906593; time: 0.0 min

## [Tune-x] 14: k=14

## [Tune-y] 14: mmce.test.mean=0.0978022; time: 0.0 min
## [Tune-x] 15: k=15

## [Tune-y] 15: mmce.test.mean=0.1131868; time: 0.0 min

## [Tune-x] 16: k=16

## [Tune-y] 16: mmce.test.mean=0.1208791; time: 0.0 min

## [Tune-x] 17: k=17

## [Tune-y] 17: mmce.test.mean=0.1054945; time: 0.0 min

## [Tune-x] 18: k=18

## [Tune-y] 18: mmce.test.mean=0.1054945; time: 0.0 min

## [Tune-x] 19: k=19

## [Tune-y] 19: mmce.test.mean=0.1131868; time: 0.0 min

## [Tune-x] 20: k=20

## [Tune-y] 20: mmce.test.mean=0.1131868; time: 0.0 min

## [Tune] Result: k=5 : mmce.test.mean=0.0824176

## [Resample] iter 2:    0.0000000 1.0000000

## [Tune] Started tuning learner classif.knn for parameter set:

##       Type len Def                                  Constr Req Tunable Trafo
## k discrete   -   - 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,1...   -    TRUE     -

## With control class: TuneControlGrid
## Imputation value: 1

## [Tune-x] 1: k=1

## [Tune-y] 1: mmce.test.mean=0.0972527; time: 0.0 min

## [Tune-x] 2: k=2

## [Tune-y] 2: mmce.test.mean=0.1049451; time: 0.0 min

## [Tune-x] 3: k=3

## [Tune-y] 3: mmce.test.mean=0.0758242; time: 0.0 min

## [Tune-x] 4: k=4

## [Tune-y] 4: mmce.test.mean=0.0824176; time: 0.0 min

## [Tune-x] 5: k=5

## [Tune-y] 5: mmce.test.mean=0.0835165; time: 0.0 min

## [Tune-x] 6: k=6

## [Tune-y] 6: mmce.test.mean=0.0978022; time: 0.0 min

## [Tune-x] 7: k=7

## [Tune-y] 7: mmce.test.mean=0.0681319; time: 0.0 min

## [Tune-x] 8: k=8

## [Tune-y] 8: mmce.test.mean=0.0906593; time: 0.0 min

## [Tune-x] 9: k=9
## [Tune-y] 9: mmce.test.mean=0.0829670; time: 0.0 min

## [Tune-x] 10: k=10

## [Tune-y] 10: mmce.test.mean=0.0901099; time: 0.0 min

## [Tune-x] 11: k=11

## [Tune-y] 11: mmce.test.mean=0.0978022; time: 0.0 min

## [Tune-x] 12: k=12

## [Tune-y] 12: mmce.test.mean=0.0978022; time: 0.0 min

## [Tune-x] 13: k=13

## [Tune-y] 13: mmce.test.mean=0.0978022; time: 0.0 min

## [Tune-x] 14: k=14

## [Tune-y] 14: mmce.test.mean=0.0978022; time: 0.0 min

## [Tune-x] 15: k=15

## [Tune-y] 15: mmce.test.mean=0.1054945; time: 0.0 min

## [Tune-x] 16: k=16

## [Tune-y] 16: mmce.test.mean=0.0978022; time: 0.0 min

## [Tune-x] 17: k=17

## [Tune-y] 17: mmce.test.mean=0.0983516; time: 0.0 min

## [Tune-x] 18: k=18
## [Tune-y] 18: mmce.test.mean=0.1126374; time: 0.0 min

## [Tune-x] 19: k=19

## [Tune-y] 19: mmce.test.mean=0.1131868; time: 0.0 min

## [Tune-x] 20: k=20

## [Tune-y] 20: mmce.test.mean=0.1203297; time: 0.0 min

## [Tune] Result: k=7 : mmce.test.mean=0.0681319

## [Resample] iter 3:    0.0833333 0.9166667

## [Tune] Started tuning learner classif.knn for parameter set:

##       Type len Def                                   Constr Req Tunable Trafo
## k discrete   -   - 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,1...   -    TRUE     -

## With control class: TuneControlGrid

## Imputation value: 1

## [Tune-x] 1: k=1

## [Tune-y] 1: mmce.test.mean=0.1027473; time: 0.0 min

## [Tune-x] 2: k=2

## [Tune-y] 2: mmce.test.mean=0.0961538; time: 0.0 min

## [Tune-x] 3: k=3

## [Tune-y] 3: mmce.test.mean=0.0730769; time: 0.0 min
## [Tune-x] 4: k=4

## [Tune-y] 4: mmce.test.mean=0.0802198; time: 0.0 min

## [Tune-x] 5: k=5

## [Tune-y] 5: mmce.test.mean=0.0725275; time: 0.0 min

## [Tune-x] 6: k=6

## [Tune-y] 6: mmce.test.mean=0.0659341; time: 0.0 min

## [Tune-x] 7: k=7

## [Tune-y] 7: mmce.test.mean=0.0582418; time: 0.0 min

## [Tune-x] 8: k=8

## [Tune-y] 8: mmce.test.mean=0.0582418; time: 0.0 min

## [Tune-x] 9: k=9

## [Tune-y] 9: mmce.test.mean=0.0659341; time: 0.0 min

## [Tune-x] 10: k=10

## [Tune-y] 10: mmce.test.mean=0.0653846; time: 0.0 min

## [Tune-x] 11: k=11

## [Tune-y] 11: mmce.test.mean=0.0884615; time: 0.0 min

## [Tune-x] 12: k=12

## [Tune-y] 12: mmce.test.mean=0.0807692; time: 0.0 min
## [Tune-x] 13: k=13
## [Tune-y] 13: mmce.test.mean=0.0807692; time: 0.0 min
## [Tune-x] 14: k=14
## [Tune-y] 14: mmce.test.mean=0.0879121; time: 0.0 min
## [Tune-x] 15: k=15
## [Tune-y] 15: mmce.test.mean=0.0950549; time: 0.0 min
## [Tune-x] 16: k=16
## [Tune-y] 16: mmce.test.mean=0.0879121; time: 0.0 min
## [Tune-x] 17: k=17
## [Tune-y] 17: mmce.test.mean=0.0950549; time: 0.0 min
## [Tune-x] 18: k=18
## [Tune-y] 18: mmce.test.mean=0.1027473; time: 0.0 min
## [Tune-x] 19: k=19
## [Tune-y] 19: mmce.test.mean=0.1027473; time: 0.0 min
## [Tune-x] 20: k=20
## [Tune-y] 20: mmce.test.mean=0.1027473; time: 0.0 min
## [Tune] Result: k=8 : mmce.test.mean=0.0582418
## [Resample] iter 4:    0.1818182 0.8181818
## [Tune] Started tuning learner classif.knn for parameter set:

##       Type len Def                                 Constr Req Tunable Trafo
## k discrete   -   - 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,1...   -    TRUE     -

## With control class: TuneControlGrid

## Imputation value: 1

## [Tune-x] 1: k=1

## [Tune-y] 1: mmce.test.mean=0.1043956; time: 0.0 min

## [Tune-x] 2: k=2

## [Tune-y] 2: mmce.test.mean=0.1351648; time: 0.0 min

## [Tune-x] 3: k=3

## [Tune-y] 3: mmce.test.mean=0.0895604; time: 0.0 min

## [Tune-x] 4: k=4

## [Tune-y] 4: mmce.test.mean=0.1049451; time: 0.0 min

## [Tune-x] 5: k=5

## [Tune-y] 5: mmce.test.mean=0.0824176; time: 0.0 min

## [Tune-x] 6: k=6

## [Tune-y] 6: mmce.test.mean=0.0829670; time: 0.0 min

## [Tune-x] 7: k=7
## [Tune-y] 7: mmce.test.mean=0.0752747; time: 0.0 min

## [Tune-x] 8: k=8

## [Tune-y] 8: mmce.test.mean=0.0906593; time: 0.0 min

## [Tune-x] 9: k=9

## [Tune-y] 9: mmce.test.mean=0.0829670; time: 0.0 min

## [Tune-x] 10: k=10

## [Tune-y] 10: mmce.test.mean=0.0752747; time: 0.0 min

## [Tune-x] 11: k=11

## [Tune-y] 11: mmce.test.mean=0.0675824; time: 0.0 min

## [Tune-x] 12: k=12

## [Tune-y] 12: mmce.test.mean=0.0901099; time: 0.0 min

## [Tune-x] 13: k=13

## [Tune-y] 13: mmce.test.mean=0.0824176; time: 0.0 min

## [Tune-x] 14: k=14

## [Tune-y] 14: mmce.test.mean=0.0901099; time: 0.0 min

## [Tune-x] 15: k=15

## [Tune-y] 15: mmce.test.mean=0.0978022; time: 0.0 min

## [Tune-x] 16: k=16
## [Tune-y] 16: mmce.test.mean=0.0978022; time: 0.0 min

## [Tune-x] 17: k=17

## [Tune-y] 17: mmce.test.mean=0.0978022; time: 0.0 min

## [Tune-x] 18: k=18

## [Tune-y] 18: mmce.test.mean=0.0978022; time: 0.0 min

## [Tune-x] 19: k=19

## [Tune-y] 19: mmce.test.mean=0.0978022; time: 0.0 min

## [Tune-x] 20: k=20

## [Tune-y] 20: mmce.test.mean=0.1054945; time: 0.0 min

## [Tune] Result: k=11 : mmce.test.mean=0.0675824

## [Resample] iter 5:    0.1666667 0.8333333

## [Tune] Started tuning learner classif.knn for parameter set:

##       Type len Def                                  Constr Req Tunable Trafo
## k discrete   -   - 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,1...   -    TRUE     -

## With control class: TuneControlGrid

## Imputation value: 1

## [Tune-x] 1: k=1

## [Tune-y] 1: mmce.test.mean=0.0989927; time: 0.0 min
## [Tune-x] 2: k=2

## [Tune-y] 2: mmce.test.mean=0.0996337; time: 0.0 min

## [Tune-x] 3: k=3

## [Tune-y] 3: mmce.test.mean=0.1001832; time: 0.0 min

## [Tune-x] 4: k=4

## [Tune-y] 4: mmce.test.mean=0.1007326; time: 0.0 min

## [Tune-x] 5: k=5

## [Tune-y] 5: mmce.test.mean=0.0770147; time: 0.0 min

## [Tune-x] 6: k=6

## [Tune-y] 6: mmce.test.mean=0.0841575; time: 0.0 min

## [Tune-x] 7: k=7

## [Tune-y] 7: mmce.test.mean=0.0621795; time: 0.0 min

## [Tune-x] 8: k=8

## [Tune-y] 8: mmce.test.mean=0.0770147; time: 0.0 min

## [Tune-x] 9: k=9

## [Tune-y] 9: mmce.test.mean=0.0698718; time: 0.0 min

## [Tune-x] 10: k=10

## [Tune-y] 10: mmce.test.mean=0.0693223; time: 0.0 min
## [Tune-x] 11: k=11

## [Tune-y] 11: mmce.test.mean=0.0775641; time: 0.0 min

## [Tune-x] 12: k=12

## [Tune-y] 12: mmce.test.mean=0.0764652; time: 0.0 min

## [Tune-x] 13: k=13

## [Tune-y] 13: mmce.test.mean=0.0913004; time: 0.0 min

## [Tune-x] 14: k=14

## [Tune-y] 14: mmce.test.mean=0.0836081; time: 0.0 min

## [Tune-x] 15: k=15

## [Tune-y] 15: mmce.test.mean=0.0984432; time: 0.0 min

## [Tune-x] 16: k=16

## [Tune-y] 16: mmce.test.mean=0.0918498; time: 0.0 min

## [Tune-x] 17: k=17

## [Tune-y] 17: mmce.test.mean=0.0989927; time: 0.0 min

## [Tune-x] 18: k=18

## [Tune-y] 18: mmce.test.mean=0.1066850; time: 0.0 min

## [Tune-x] 19: k=19

## [Tune-y] 19: mmce.test.mean=0.1066850; time: 0.0 min
## [Tune-x] 20: k=20

## [Tune-y] 20: mmce.test.mean=0.0995421; time: 0.0 min

## [Tune] Result: k=7 : mmce.test.mean=0.0621795

## [Resample] iter 6:    0.1538462 0.8461538

## [Tune] Started tuning learner classif.knn for parameter set:

##       Type len Def                                  Constr Req Tunable Trafo
## k discrete   -   - 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,1...   -    TRUE     -

## With control class: TuneControlGrid

## Imputation value: 1

## [Tune-x] 1: k=1

## [Tune-y] 1: mmce.test.mean=0.1144689; time: 0.0 min

## [Tune-x] 2: k=2

## [Tune-y] 2: mmce.test.mean=0.0919414; time: 0.0 min

## [Tune-x] 3: k=3

## [Tune-y] 3: mmce.test.mean=0.0759158; time: 0.0 min

## [Tune-x] 4: k=4

## [Tune-y] 4: mmce.test.mean=0.0841575; time: 0.0 min

## [Tune-x] 5: k=5
## [Tune-y] 5: mmce.test.mean=0.0836081; time: 0.0 min

## [Tune-x] 6: k=6

## [Tune-y] 6: mmce.test.mean=0.0830586; time: 0.0 min

## [Tune-x] 7: k=7

## [Tune-y] 7: mmce.test.mean=0.0836081; time: 0.0 min

## [Tune-x] 8: k=8

## [Tune-y] 8: mmce.test.mean=0.0830586; time: 0.0 min

## [Tune-x] 9: k=9

## [Tune-y] 9: mmce.test.mean=0.0682234; time: 0.0 min

## [Tune-x] 10: k=10

## [Tune-y] 10: mmce.test.mean=0.0907509; time: 0.0 min

## [Tune-x] 11: k=11

## [Tune-y] 11: mmce.test.mean=0.0830586; time: 0.0 min

## [Tune-x] 12: k=12

## [Tune-y] 12: mmce.test.mean=0.0830586; time: 0.0 min

## [Tune-x] 13: k=13

## [Tune-y] 13: mmce.test.mean=0.1138278; time: 0.0 min

## [Tune-x] 14: k=14
## [Tune-y] 14: mmce.test.mean=0.1138278; time: 0.0 min

## [Tune-x] 15: k=15

## [Tune-y] 15: mmce.test.mean=0.1138278; time: 0.0 min

## [Tune-x] 16: k=16

## [Tune-y] 16: mmce.test.mean=0.1215201; time: 0.0 min

## [Tune-x] 17: k=17

## [Tune-y] 17: mmce.test.mean=0.1138278; time: 0.0 min

## [Tune-x] 18: k=18

## [Tune-y] 18: mmce.test.mean=0.1061355; time: 0.0 min

## [Tune-x] 19: k=19

## [Tune-y] 19: mmce.test.mean=0.1138278; time: 0.0 min

## [Tune-x] 20: k=20

## [Tune-y] 20: mmce.test.mean=0.1061355; time: 0.0 min

## [Tune] Result: k=9 : mmce.test.mean=0.0682234

## [Resample] iter 7:    0.0769231 0.9230769

## [Tune] Started tuning learner classif.knn for parameter set:

##       Type len Def                                   Constr Req Tunable Trafo
## k discrete   -   - 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,1...   -    TRUE     -
## With control class: TuneControlGrid

## Imputation value: 1

## [Tune-x] 1: k=1

## [Tune-y] 1: mmce.test.mean=0.1203297; time: 0.0 min

## [Tune-x] 2: k=2

## [Tune-y] 2: mmce.test.mean=0.0901099; time: 0.0 min

## [Tune-x] 3: k=3

## [Tune-y] 3: mmce.test.mean=0.0906593; time: 0.0 min

## [Tune-x] 4: k=4

## [Tune-y] 4: mmce.test.mean=0.1060440; time: 0.0 min

## [Tune-x] 5: k=5

## [Tune-y] 5: mmce.test.mean=0.0829670; time: 0.0 min

## [Tune-x] 6: k=6

## [Tune-y] 6: mmce.test.mean=0.0829670; time: 0.0 min

## [Tune-x] 7: k=7

## [Tune-y] 7: mmce.test.mean=0.0829670; time: 0.0 min

## [Tune-x] 8: k=8

## [Tune-y] 8: mmce.test.mean=0.0598901; time: 0.0 min
## [Tune-x] 9: k=9

## [Tune-y] 9: mmce.test.mean=0.0829670; time: 0.0 min

## [Tune-x] 10: k=10

## [Tune-y] 10: mmce.test.mean=0.0906593; time: 0.0 min

## [Tune-x] 11: k=11

## [Tune-y] 11: mmce.test.mean=0.0983516; time: 0.0 min

## [Tune-x] 12: k=12

## [Tune-y] 12: mmce.test.mean=0.0983516; time: 0.0 min

## [Tune-x] 13: k=13

## [Tune-y] 13: mmce.test.mean=0.0983516; time: 0.0 min

## [Tune-x] 14: k=14

## [Tune-y] 14: mmce.test.mean=0.0835165; time: 0.0 min

## [Tune-x] 15: k=15

## [Tune-y] 15: mmce.test.mean=0.0912088; time: 0.0 min

## [Tune-x] 16: k=16

## [Tune-y] 16: mmce.test.mean=0.0989011; time: 0.0 min

## [Tune-x] 17: k=17

## [Tune-y] 17: mmce.test.mean=0.1137363; time: 0.0 min
## [Tune-x] 18: k=18

## [Tune-y] 18: mmce.test.mean=0.1208791; time: 0.0 min

## [Tune-x] 19: k=19

## [Tune-y] 19: mmce.test.mean=0.1280220; time: 0.0 min

## [Tune-x] 20: k=20

## [Tune-y] 20: mmce.test.mean=0.1208791; time: 0.0 min

## [Tune] Result: k=8 : mmce.test.mean=0.0598901

## [Resample] iter 8:    0.0000000 1.0000000

## [Tune] Started tuning learner classif.knn for parameter set:

##       Type len Def                                  Constr Req Tunable Trafo
## k discrete   -   - 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,1...   -    TRUE     -

## With control class: TuneControlGrid

## Imputation value: 1

## [Tune-x] 1: k=1

## [Tune-y] 1: mmce.test.mean=0.1049451; time: 0.0 min

## [Tune-x] 2: k=2

## [Tune-y] 2: mmce.test.mean=0.0972527; time: 0.0 min

## [Tune-x] 3: k=3
## [Tune-y] 3: mmce.test.mean=0.0824176; time: 0.0 min

## [Tune-x] 4: k=4

## [Tune-y] 4: mmce.test.mean=0.0670330; time: 0.0 min

## [Tune-x] 5: k=5

## [Tune-y] 5: mmce.test.mean=0.0675824; time: 0.0 min

## [Tune-x] 6: k=6

## [Tune-y] 6: mmce.test.mean=0.0818681; time: 0.0 min

## [Tune-x] 7: k=7

## [Tune-y] 7: mmce.test.mean=0.0890110; time: 0.0 min

## [Tune-x] 8: k=8

## [Tune-y] 8: mmce.test.mean=0.0747253; time: 0.0 min

## [Tune-x] 9: k=9

## [Tune-y] 9: mmce.test.mean=0.1043956; time: 0.0 min

## [Tune-x] 10: k=10

## [Tune-y] 10: mmce.test.mean=0.0967033; time: 0.0 min

## [Tune-x] 11: k=11

## [Tune-y] 11: mmce.test.mean=0.0890110; time: 0.0 min

## [Tune-x] 12: k=12
## [Tune-y] 12: mmce.test.mean=0.0818681; time: 0.0 min

## [Tune-x] 13: k=13

## [Tune-y] 13: mmce.test.mean=0.0967033; time: 0.0 min

## [Tune-x] 14: k=14

## [Tune-y] 14: mmce.test.mean=0.0972527; time: 0.0 min

## [Tune-x] 15: k=15

## [Tune-y] 15: mmce.test.mean=0.1043956; time: 0.0 min

## [Tune-x] 16: k=16

## [Tune-y] 16: mmce.test.mean=0.1120879; time: 0.0 min

## [Tune-x] 17: k=17

## [Tune-y] 17: mmce.test.mean=0.1120879; time: 0.0 min

## [Tune-x] 18: k=18

## [Tune-y] 18: mmce.test.mean=0.1049451; time: 0.0 min

## [Tune-x] 19: k=19

## [Tune-y] 19: mmce.test.mean=0.1120879; time: 0.0 min

## [Tune-x] 20: k=20

## [Tune-y] 20: mmce.test.mean=0.1043956; time: 0.0 min

## [Tune] Result: k=4 : mmce.test.mean=0.0670330
## [Resample] iter 9:    0.1666667 0.8333333

## [Tune] Started tuning learner classif.knn for parameter set:

##       Type len Def                                   Constr Req Tunable Trafo
## k discrete   -   - 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,1...   -    TRUE     -

## With control class: TuneControlGrid

## Imputation value: 1

## [Tune-x] 1: k=1

## [Tune-y] 1: mmce.test.mean=0.0994505; time: 0.0 min

## [Tune-x] 2: k=2

## [Tune-y] 2: mmce.test.mean=0.1131868; time: 0.0 min

## [Tune-x] 3: k=3

## [Tune-y] 3: mmce.test.mean=0.1054945; time: 0.0 min

## [Tune-x] 4: k=4

## [Tune-y] 4: mmce.test.mean=0.0989011; time: 0.0 min

## [Tune-x] 5: k=5

## [Tune-y] 5: mmce.test.mean=0.0989011; time: 0.0 min

## [Tune-x] 6: k=6

## [Tune-y] 6: mmce.test.mean=0.0912088; time: 0.0 min
## [Tune-x] 7: k=7

## [Tune-y] 7: mmce.test.mean=0.0840659; time: 0.0 min

## [Tune-x] 8: k=8

## [Tune-y] 8: mmce.test.mean=0.0835165; time: 0.0 min

## [Tune-x] 9: k=9

## [Tune-y] 9: mmce.test.mean=0.0840659; time: 0.0 min

## [Tune-x] 10: k=10

## [Tune-y] 10: mmce.test.mean=0.0763736; time: 0.0 min

## [Tune-x] 11: k=11

## [Tune-y] 11: mmce.test.mean=0.0840659; time: 0.0 min

## [Tune-x] 12: k=12

## [Tune-y] 12: mmce.test.mean=0.0989011; time: 0.0 min

## [Tune-x] 13: k=13

## [Tune-y] 13: mmce.test.mean=0.0917582; time: 0.0 min

## [Tune-x] 14: k=14

## [Tune-y] 14: mmce.test.mean=0.0917582; time: 0.0 min

## [Tune-x] 15: k=15

## [Tune-y] 15: mmce.test.mean=0.0917582; time: 0.0 min
## [Tune-x] 16: k=16

## [Tune-y] 16: mmce.test.mean=0.0917582; time: 0.0 min

## [Tune-x] 17: k=17

## [Tune-y] 17: mmce.test.mean=0.0989011; time: 0.0 min

## [Tune-x] 18: k=18

## [Tune-y] 18: mmce.test.mean=0.1060440; time: 0.0 min

## [Tune-x] 19: k=19

## [Tune-y] 19: mmce.test.mean=0.0912088; time: 0.0 min

## [Tune-x] 20: k=20

## [Tune-y] 20: mmce.test.mean=0.0912088; time: 0.0 min

## [Tune] Result: k=10 : mmce.test.mean=0.0763736

## [Resample] iter 10:   0.1666667 0.8333333

## [Tune] Started tuning learner classif.knn for parameter set:

##       Type len Def                                   Constr Req Tunable Trafo
## k discrete   -   - 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,1...   -    TRUE     -

## With control class: TuneControlGrid

## Imputation value: 1

## [Tune-x] 1: k=1
## [Tune-y] 1: mmce.test.mean=0.1291209; time: 0.0 min

## [Tune-x] 2: k=2

## [Tune-y] 2: mmce.test.mean=0.0906593; time: 0.0 min

## [Tune-x] 3: k=3

## [Tune-y] 3: mmce.test.mean=0.0912088; time: 0.0 min

## [Tune-x] 4: k=4

## [Tune-y] 4: mmce.test.mean=0.0835165; time: 0.0 min

## [Tune-x] 5: k=5

## [Tune-y] 5: mmce.test.mean=0.0989011; time: 0.0 min

## [Tune-x] 6: k=6

## [Tune-y] 6: mmce.test.mean=0.0835165; time: 0.0 min

## [Tune-x] 7: k=7

## [Tune-y] 7: mmce.test.mean=0.0829670; time: 0.0 min

## [Tune-x] 8: k=8

## [Tune-y] 8: mmce.test.mean=0.0752747; time: 0.0 min

## [Tune-x] 9: k=9

## [Tune-y] 9: mmce.test.mean=0.0675824; time: 0.0 min

## [Tune-x] 10: k=10
## [Tune-y] 10: mmce.test.mean=0.0978022; time: 0.0 min

## [Tune-x] 11: k=11

## [Tune-y] 11: mmce.test.mean=0.0901099; time: 0.0 min

## [Tune-x] 12: k=12

## [Tune-y] 12: mmce.test.mean=0.1054945; time: 0.0 min

## [Tune-x] 13: k=13

## [Tune-y] 13: mmce.test.mean=0.1060440; time: 0.0 min

## [Tune-x] 14: k=14

## [Tune-y] 14: mmce.test.mean=0.0978022; time: 0.0 min

## [Tune-x] 15: k=15

## [Tune-y] 15: mmce.test.mean=0.1054945; time: 0.0 min

## [Tune-x] 16: k=16

## [Tune-y] 16: mmce.test.mean=0.1131868; time: 0.0 min

## [Tune-x] 17: k=17

## [Tune-y] 17: mmce.test.mean=0.1060440; time: 0.0 min

## [Tune-x] 18: k=18

## [Tune-y] 18: mmce.test.mean=0.1137363; time: 0.0 min

## [Tune-x] 19: k=19
## [Tune-y] 19: mmce.test.mean=0.1137363; time: 0.0 min

## [Tune-x] 20: k=20

## [Tune-y] 20: mmce.test.mean=0.1214286; time: 0.0 min

## [Tune] Result: k=9 : mmce.test.mean=0.0675824

## [Resample] iter 11:   0.0000000 1.0000000

## [Tune] Started tuning learner classif.knn for parameter set:

##       Type len Def                                  Constr Req Tunable Trafo
## k discrete   -   - 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,1...   -    TRUE     -

## With control class: TuneControlGrid

## Imputation value: 1

## [Tune-x] 1: k=1

## [Tune-y] 1: mmce.test.mean=0.1263736; time: 0.0 min

## [Tune-x] 2: k=2

## [Tune-y] 2: mmce.test.mean=0.1115385; time: 0.0 min

## [Tune-x] 3: k=3

## [Tune-y] 3: mmce.test.mean=0.0813187; time: 0.0 min

## [Tune-x] 4: k=4

## [Tune-y] 4: mmce.test.mean=0.0972527; time: 0.0 min
## [Tune-x] 5: k=5

## [Tune-y] 5: mmce.test.mean=0.0824176; time: 0.0 min

## [Tune-x] 6: k=6

## [Tune-y] 6: mmce.test.mean=0.0890110; time: 0.0 min

## [Tune-x] 7: k=7

## [Tune-y] 7: mmce.test.mean=0.0895604; time: 0.0 min

## [Tune-x] 8: k=8

## [Tune-y] 8: mmce.test.mean=0.1043956; time: 0.0 min

## [Tune-x] 9: k=9

## [Tune-y] 9: mmce.test.mean=0.0967033; time: 0.0 min

## [Tune-x] 10: k=10

## [Tune-y] 10: mmce.test.mean=0.0895604; time: 0.0 min

## [Tune-x] 11: k=11

## [Tune-y] 11: mmce.test.mean=0.0961538; time: 0.0 min

## [Tune-x] 12: k=12

## [Tune-y] 12: mmce.test.mean=0.0961538; time: 0.0 min

## [Tune-x] 13: k=13

## [Tune-y] 13: mmce.test.mean=0.0967033; time: 0.0 min
## [Tune-x] 14: k=14

## [Tune-y] 14: mmce.test.mean=0.0967033; time: 0.0 min

## [Tune-x] 15: k=15

## [Tune-y] 15: mmce.test.mean=0.1038462; time: 0.0 min

## [Tune-x] 16: k=16

## [Tune-y] 16: mmce.test.mean=0.0967033; time: 0.0 min

## [Tune-x] 17: k=17

## [Tune-y] 17: mmce.test.mean=0.0961538; time: 0.0 min

## [Tune-x] 18: k=18

## [Tune-y] 18: mmce.test.mean=0.1038462; time: 0.0 min

## [Tune-x] 19: k=19

## [Tune-y] 19: mmce.test.mean=0.1192308; time: 0.0 min

## [Tune-x] 20: k=20

## [Tune-y] 20: mmce.test.mean=0.1192308; time: 0.0 min

## [Tune] Result: k=3 : mmce.test.mean=0.0813187

## [Resample] iter 12:   0.0909091 0.9090909

## [Tune] Started tuning learner classif.knn for parameter set:

##       Type len Def                                   Constr Req Tunable Trafo
## k discrete   -   - 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,1...   -    TRUE     -
## With control class: TuneControlGrid

## Imputation value: 1

## [Tune-x] 1: k=1

## [Tune-y] 1: mmce.test.mean=0.1110806; time: 0.0 min

## [Tune-x] 2: k=2

## [Tune-y] 2: mmce.test.mean=0.1109890; time: 0.0 min

## [Tune-x] 3: k=3

## [Tune-y] 3: mmce.test.mean=0.0807692; time: 0.0 min

## [Tune-x] 4: k=4

## [Tune-y] 4: mmce.test.mean=0.0819597; time: 0.0 min

## [Tune-x] 5: k=5

## [Tune-y] 5: mmce.test.mean=0.0819597; time: 0.0 min

## [Tune-x] 6: k=6

## [Tune-y] 6: mmce.test.mean=0.0818681; time: 0.0 min

## [Tune-x] 7: k=7

## [Tune-y] 7: mmce.test.mean=0.0742674; time: 0.0 min

## [Tune-x] 8: k=8

## [Tune-y] 8: mmce.test.mean=0.0891026; time: 0.0 min
## [Tune-x] 9: k=9

## [Tune-y] 9: mmce.test.mean=0.0807692; time: 0.0 min

## [Tune-x] 10: k=10

## [Tune-y] 10: mmce.test.mean=0.0736264; time: 0.0 min

## [Tune-x] 11: k=11

## [Tune-y] 11: mmce.test.mean=0.0736264; time: 0.0 min

## [Tune-x] 12: k=12

## [Tune-y] 12: mmce.test.mean=0.0884615; time: 0.0 min

## [Tune-x] 13: k=13

## [Tune-y] 13: mmce.test.mean=0.0813187; time: 0.0 min

## [Tune-x] 14: k=14

## [Tune-y] 14: mmce.test.mean=0.0813187; time: 0.0 min

## [Tune-x] 15: k=15

## [Tune-y] 15: mmce.test.mean=0.0813187; time: 0.0 min

## [Tune-x] 16: k=16

## [Tune-y] 16: mmce.test.mean=0.0967033; time: 0.0 min

## [Tune-x] 17: k=17

## [Tune-y] 17: mmce.test.mean=0.0967033; time: 0.0 min
## [Tune-x] 18: k=18

## [Tune-y] 18: mmce.test.mean=0.0967033; time: 0.0 min

## [Tune-x] 19: k=19

## [Tune-y] 19: mmce.test.mean=0.1044872; time: 0.0 min

## [Tune-x] 20: k=20

## [Tune-y] 20: mmce.test.mean=0.1121795; time: 0.0 min

## [Tune] Result: k=11 : mmce.test.mean=0.0736264

## [Resample] iter 13:   0.0769231 0.9230769

## [Tune] Started tuning learner classif.knn for parameter set:

##       Type len Def                                   Constr Req Tunable Trafo
## k discrete   -   - 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,1...   -    TRUE     -

## With control class: TuneControlGrid

## Imputation value: 1

## [Tune-x] 1: k=1

## [Tune-y] 1: mmce.test.mean=0.1115385; time: 0.0 min

## [Tune-x] 2: k=2

## [Tune-y] 2: mmce.test.mean=0.0895604; time: 0.0 min

## [Tune-x] 3: k=3
## [Tune-y] 3: mmce.test.mean=0.0884615; time: 0.0 min

## [Tune-x] 4: k=4

## [Tune-y] 4: mmce.test.mean=0.0741758; time: 0.0 min

## [Tune-x] 5: k=5

## [Tune-y] 5: mmce.test.mean=0.0587912; time: 0.0 min

## [Tune-x] 6: k=6

## [Tune-y] 6: mmce.test.mean=0.0741758; time: 0.0 min

## [Tune-x] 7: k=7

## [Tune-y] 7: mmce.test.mean=0.0741758; time: 0.0 min

## [Tune-x] 8: k=8

## [Tune-y] 8: mmce.test.mean=0.0813187; time: 0.0 min

## [Tune-x] 9: k=9

## [Tune-y] 9: mmce.test.mean=0.0884615; time: 0.0 min

## [Tune-x] 10: k=10

## [Tune-y] 10: mmce.test.mean=0.0664835; time: 0.0 min

## [Tune-x] 11: k=11

## [Tune-y] 11: mmce.test.mean=0.0813187; time: 0.0 min

## [Tune-x] 12: k=12
## [Tune-y] 12: mmce.test.mean=0.0884615; time: 0.0 min

## [Tune-x] 13: k=13

## [Tune-y] 13: mmce.test.mean=0.0890110; time: 0.0 min

## [Tune-x] 14: k=14

## [Tune-y] 14: mmce.test.mean=0.0890110; time: 0.0 min

## [Tune-x] 15: k=15

## [Tune-y] 15: mmce.test.mean=0.1115385; time: 0.0 min

## [Tune-x] 16: k=16

## [Tune-y] 16: mmce.test.mean=0.1104396; time: 0.0 min

## [Tune-x] 17: k=17

## [Tune-y] 17: mmce.test.mean=0.1032967; time: 0.0 min

## [Tune-x] 18: k=18

## [Tune-y] 18: mmce.test.mean=0.1032967; time: 0.0 min

## [Tune-x] 19: k=19

## [Tune-y] 19: mmce.test.mean=0.1032967; time: 0.0 min

## [Tune-x] 20: k=20

## [Tune-y] 20: mmce.test.mean=0.1032967; time: 0.0 min

## [Tune] Result: k=5 : mmce.test.mean=0.0587912
## [Resample] iter 14:   0.0833333 0.9166667

## [Tune] Started tuning learner classif.knn for parameter set:

##       Type len Def                                   Constr Req Tunable Trafo
## k discrete   -   - 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,1...   -    TRUE     -

## With control class: TuneControlGrid

## Imputation value: 1

## [Tune-x] 1: k=1

## [Tune-y] 1: mmce.test.mean=0.1056777; time: 0.0 min

## [Tune-x] 2: k=2

## [Tune-y] 2: mmce.test.mean=0.1293040; time: 0.0 min

## [Tune-x] 3: k=3

## [Tune-y] 3: mmce.test.mean=0.0913919; time: 0.0 min

## [Tune-x] 4: k=4

## [Tune-y] 4: mmce.test.mean=0.0830586; time: 0.0 min

## [Tune-x] 5: k=5

## [Tune-y] 5: mmce.test.mean=0.0830586; time: 0.0 min

## [Tune-x] 6: k=6

## [Tune-y] 6: mmce.test.mean=0.1061355; time: 0.0 min
## [Tune-x] 7: k=7

## [Tune-y] 7: mmce.test.mean=0.0984432; time: 0.0 min

## [Tune-x] 8: k=8

## [Tune-y] 8: mmce.test.mean=0.1055861; time: 0.0 min

## [Tune-x] 9: k=9

## [Tune-y] 9: mmce.test.mean=0.0902015; time: 0.0 min

## [Tune-x] 10: k=10

## [Tune-y] 10: mmce.test.mean=0.0824176; time: 0.0 min

## [Tune-x] 11: k=11

## [Tune-y] 11: mmce.test.mean=0.0824176; time: 0.0 min

## [Tune-x] 12: k=12

## [Tune-y] 12: mmce.test.mean=0.0824176; time: 0.0 min

## [Tune-x] 13: k=13

## [Tune-y] 13: mmce.test.mean=0.0895604; time: 0.0 min

## [Tune-x] 14: k=14

## [Tune-y] 14: mmce.test.mean=0.1049451; time: 0.0 min

## [Tune-x] 15: k=15

## [Tune-y] 15: mmce.test.mean=0.0972527; time: 0.0 min
## [Tune-x] 16: k=16

## [Tune-y] 16: mmce.test.mean=0.0972527; time: 0.0 min

## [Tune-x] 17: k=17

## [Tune-y] 17: mmce.test.mean=0.1055861; time: 0.0 min

## [Tune-x] 18: k=18

## [Tune-y] 18: mmce.test.mean=0.1127289; time: 0.0 min

## [Tune-x] 19: k=19

## [Tune-y] 19: mmce.test.mean=0.1127289; time: 0.0 min

## [Tune-x] 20: k=20

## [Tune-y] 20: mmce.test.mean=0.1204212; time: 0.0 min

## [Tune] Result: k=10 : mmce.test.mean=0.0824176

## [Resample] iter 15:   0.0000000 1.0000000

## [Tune] Started tuning learner classif.knn for parameter set:

##       Type len Def                                   Constr Req Tunable Trafo
## k discrete   -   - 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,1...   -    TRUE     -

## With control class: TuneControlGrid

## Imputation value: 1

## [Tune-x] 1: k=1
## [Tune-y] 1: mmce.test.mean=0.0978755; time: 0.0 min

## [Tune-x] 2: k=2

## [Tune-y] 2: mmce.test.mean=0.1050183; time: 0.0 min

## [Tune-x] 3: k=3

## [Tune-y] 3: mmce.test.mean=0.0890842; time: 0.0 min

## [Tune-x] 4: k=4

## [Tune-y] 4: mmce.test.mean=0.0968498; time: 0.0 min

## [Tune-x] 5: k=5

## [Tune-y] 5: mmce.test.mean=0.0968498; time: 0.0 min

## [Tune-x] 6: k=6

## [Tune-y] 6: mmce.test.mean=0.0891575; time: 0.0 min

## [Tune-x] 7: k=7

## [Tune-y] 7: mmce.test.mean=0.0814652; time: 0.0 min

## [Tune-x] 8: k=8

## [Tune-y] 8: mmce.test.mean=0.0886081; time: 0.0 min

## [Tune-x] 9: k=9

## [Tune-y] 9: mmce.test.mean=0.0963004; time: 0.0 min

## [Tune-x] 10: k=10
## [Tune-y] 10: mmce.test.mean=0.0809158; time: 0.0 min

## [Tune-x] 11: k=11

## [Tune-y] 11: mmce.test.mean=0.0737729; time: 0.0 min

## [Tune-x] 12: k=12

## [Tune-y] 12: mmce.test.mean=0.0809158; time: 0.0 min

## [Tune-x] 13: k=13

## [Tune-y] 13: mmce.test.mean=0.0880586; time: 0.0 min

## [Tune-x] 14: k=14

## [Tune-y] 14: mmce.test.mean=0.0952015; time: 0.0 min

## [Tune-x] 15: k=15

## [Tune-y] 15: mmce.test.mean=0.1028938; time: 0.0 min

## [Tune-x] 16: k=16

## [Tune-y] 16: mmce.test.mean=0.1028938; time: 0.0 min

## [Tune-x] 17: k=17

## [Tune-y] 17: mmce.test.mean=0.0957509; time: 0.0 min

## [Tune-x] 18: k=18

## [Tune-y] 18: mmce.test.mean=0.0957509; time: 0.0 min

## [Tune-x] 19: k=19
## [Tune-y] 19: mmce.test.mean=0.1034432; time: 0.0 min

## [Tune-x] 20: k=20

## [Tune-y] 20: mmce.test.mean=0.0957509; time: 0.0 min

## [Tune] Result: k=11 : mmce.test.mean=0.0737729

## [Resample] iter 16:   0.0000000 1.0000000

## [Tune] Started tuning learner classif.knn for parameter set:

##        Type len Def                                   Constr Req Tunable Trafo
## k discrete   -   - 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,1...   -    TRUE     -

## With control class: TuneControlGrid

## Imputation value: 1

## [Tune-x] 1: k=1

## [Tune-y] 1: mmce.test.mean=0.1049451; time: 0.0 min

## [Tune-x] 2: k=2

## [Tune-y] 2: mmce.test.mean=0.0983516; time: 0.0 min

## [Tune-x] 3: k=3

## [Tune-y] 3: mmce.test.mean=0.1054945; time: 0.0 min

## [Tune-x] 4: k=4

## [Tune-y] 4: mmce.test.mean=0.0978022; time: 0.0 min
## [Tune-x] 5: k=5

## [Tune-y] 5: mmce.test.mean=0.0901099; time: 0.0 min

## [Tune-x] 6: k=6

## [Tune-y] 6: mmce.test.mean=0.1049451; time: 0.0 min

## [Tune-x] 7: k=7

## [Tune-y] 7: mmce.test.mean=0.0752747; time: 0.0 min

## [Tune-x] 8: k=8

## [Tune-y] 8: mmce.test.mean=0.0972527; time: 0.0 min

## [Tune-x] 9: k=9

## [Tune-y] 9: mmce.test.mean=0.1054945; time: 0.0 min

## [Tune-x] 10: k=10

## [Tune-y] 10: mmce.test.mean=0.1049451; time: 0.0 min

## [Tune-x] 11: k=11

## [Tune-y] 11: mmce.test.mean=0.1049451; time: 0.0 min

## [Tune-x] 12: k=12

## [Tune-y] 12: mmce.test.mean=0.0972527; time: 0.0 min

## [Tune-x] 13: k=13

## [Tune-y] 13: mmce.test.mean=0.1049451; time: 0.0 min
## [Tune-x] 14: k=14

## [Tune-y] 14: mmce.test.mean=0.1120879; time: 0.0 min

## [Tune-x] 15: k=15

## [Tune-y] 15: mmce.test.mean=0.1120879; time: 0.0 min

## [Tune-x] 16: k=16

## [Tune-y] 16: mmce.test.mean=0.1120879; time: 0.0 min

## [Tune-x] 17: k=17

## [Tune-y] 17: mmce.test.mean=0.1043956; time: 0.0 min

## [Tune-x] 18: k=18

## [Tune-y] 18: mmce.test.mean=0.1120879; time: 0.0 min

## [Tune-x] 19: k=19

## [Tune-y] 19: mmce.test.mean=0.1049451; time: 0.0 min

## [Tune-x] 20: k=20

## [Tune-y] 20: mmce.test.mean=0.1203297; time: 0.0 min

## [Tune] Result: k=7 : mmce.test.mean=0.0752747

## [Resample] iter 17:   0.0769231 0.9230769

## [Tune] Started tuning learner classif.knn for parameter set:

##       Type len Def                                   Constr Req Tunable Trafo
## k discrete   -   - 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,1...   -    TRUE     -
## With control class: TuneControlGrid

## Imputation value: 1

## [Tune-x] 1: k=1

## [Tune-y] 1: mmce.test.mean=0.1186813; time: 0.0 min

## [Tune-x] 2: k=2

## [Tune-y] 2: mmce.test.mean=0.1038462; time: 0.0 min

## [Tune-x] 3: k=3

## [Tune-y] 3: mmce.test.mean=0.0884615; time: 0.0 min

## [Tune-x] 4: k=4

## [Tune-y] 4: mmce.test.mean=0.0884615; time: 0.0 min

## [Tune-x] 5: k=5

## [Tune-y] 5: mmce.test.mean=0.0807692; time: 0.0 min

## [Tune-x] 6: k=6

## [Tune-y] 6: mmce.test.mean=0.1038462; time: 0.0 min

## [Tune-x] 7: k=7

## [Tune-y] 7: mmce.test.mean=0.0736264; time: 0.0 min

## [Tune-x] 8: k=8

## [Tune-y] 8: mmce.test.mean=0.0961538; time: 0.0 min
## [Tune-x] 9: k=9

## [Tune-y] 9: mmce.test.mean=0.1032967; time: 0.0 min

## [Tune-x] 10: k=10

## [Tune-y] 10: mmce.test.mean=0.0884615; time: 0.0 min

## [Tune-x] 11: k=11

## [Tune-y] 11: mmce.test.mean=0.1032967; time: 0.0 min

## [Tune-x] 12: k=12

## [Tune-y] 12: mmce.test.mean=0.1115385; time: 0.0 min

## [Tune-x] 13: k=13

## [Tune-y] 13: mmce.test.mean=0.0972527; time: 0.0 min

## [Tune-x] 14: k=14

## [Tune-y] 14: mmce.test.mean=0.0967033; time: 0.0 min

## [Tune-x] 15: k=15

## [Tune-y] 15: mmce.test.mean=0.0967033; time: 0.0 min

## [Tune-x] 16: k=16

## [Tune-y] 16: mmce.test.mean=0.1115385; time: 0.0 min

## [Tune-x] 17: k=17

## [Tune-y] 17: mmce.test.mean=0.1109890; time: 0.0 min
## [Tune-x] 18: k=18

## [Tune-y] 18: mmce.test.mean=0.1115385; time: 0.0 min

## [Tune-x] 19: k=19

## [Tune-y] 19: mmce.test.mean=0.1115385; time: 0.0 min

## [Tune-x] 20: k=20

## [Tune-y] 20: mmce.test.mean=0.1192308; time: 0.0 min

## [Tune] Result: k=7 : mmce.test.mean=0.0736264

## [Resample] iter 18:   0.0000000 1.0000000

## [Tune] Started tuning learner classif.knn for parameter set:

##       Type len Def                                  Constr Req Tunable Trafo
## k discrete   -   - 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,1...   -    TRUE     -

## With control class: TuneControlGrid

## Imputation value: 1

## [Tune-x] 1: k=1

## [Tune-y] 1: mmce.test.mean=0.1269231; time: 0.0 min

## [Tune-x] 2: k=2

## [Tune-y] 2: mmce.test.mean=0.0967033; time: 0.0 min

## [Tune-x] 3: k=3
## [Tune-y] 3: mmce.test.mean=0.0961538; time: 0.0 min

## [Tune-x] 4: k=4

## [Tune-y] 4: mmce.test.mean=0.0890110; time: 0.0 min

## [Tune-x] 5: k=5

## [Tune-y] 5: mmce.test.mean=0.0895604; time: 0.0 min

## [Tune-x] 6: k=6

## [Tune-y] 6: mmce.test.mean=0.0818681; time: 0.0 min

## [Tune-x] 7: k=7

## [Tune-y] 7: mmce.test.mean=0.0670330; time: 0.0 min

## [Tune-x] 8: k=8

## [Tune-y] 8: mmce.test.mean=0.0884615; time: 0.0 min

## [Tune-x] 9: k=9

## [Tune-y] 9: mmce.test.mean=0.0741758; time: 0.0 min

## [Tune-x] 10: k=10

## [Tune-y] 10: mmce.test.mean=0.0967033; time: 0.0 min

## [Tune-x] 11: k=11

## [Tune-y] 11: mmce.test.mean=0.0813187; time: 0.0 min

## [Tune-x] 12: k=12
## [Tune-y] 12: mmce.test.mean=0.0741758; time: 0.0 min

## [Tune-x] 13: k=13

## [Tune-y] 13: mmce.test.mean=0.0884615; time: 0.0 min

## [Tune-x] 14: k=14

## [Tune-y] 14: mmce.test.mean=0.0890110; time: 0.0 min

## [Tune-x] 15: k=15

## [Tune-y] 15: mmce.test.mean=0.0961538; time: 0.0 min

## [Tune-x] 16: k=16

## [Tune-y] 16: mmce.test.mean=0.0967033; time: 0.0 min

## [Tune-x] 17: k=17

## [Tune-y] 17: mmce.test.mean=0.1038462; time: 0.0 min

## [Tune-x] 18: k=18

## [Tune-y] 18: mmce.test.mean=0.1115385; time: 0.0 min

## [Tune-x] 19: k=19

## [Tune-y] 19: mmce.test.mean=0.1038462; time: 0.0 min

## [Tune-x] 20: k=20

## [Tune-y] 20: mmce.test.mean=0.1038462; time: 0.0 min

## [Tune] Result: k=7 : mmce.test.mean=0.0670330
## [Resample] iter 19:   0.0000000 1.0000000

## [Tune] Started tuning learner classif.knn for parameter set:

##       Type len Def                                   Constr Req Tunable Trafo
## k discrete   -   - 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,1...   -    TRUE     -

## With control class: TuneControlGrid

## Imputation value: 1

## [Tune-x] 1: k=1

## [Tune-y] 1: mmce.test.mean=0.0913919; time: 0.0 min

## [Tune-x] 2: k=2

## [Tune-y] 2: mmce.test.mean=0.0830586; time: 0.0 min

## [Tune-x] 3: k=3

## [Tune-y] 3: mmce.test.mean=0.0760073; time: 0.0 min

## [Tune-x] 4: k=4

## [Tune-y] 4: mmce.test.mean=0.0760073; time: 0.0 min

## [Tune-x] 5: k=5

## [Tune-y] 5: mmce.test.mean=0.0606227; time: 0.0 min

## [Tune-x] 6: k=6

## [Tune-y] 6: mmce.test.mean=0.0611722; time: 0.0 min
## [Tune-x] 7: k=7

## [Tune-y] 7: mmce.test.mean=0.0831502; time: 0.0 min

## [Tune-x] 8: k=8

## [Tune-y] 8: mmce.test.mean=0.0908425; time: 0.0 min

## [Tune-x] 9: k=9

## [Tune-y] 9: mmce.test.mean=0.0831502; time: 0.0 min

## [Tune-x] 10: k=10

## [Tune-y] 10: mmce.test.mean=0.0836996; time: 0.0 min

## [Tune-x] 11: k=11

## [Tune-y] 11: mmce.test.mean=0.0979853; time: 0.0 min

## [Tune-x] 12: k=12

## [Tune-y] 12: mmce.test.mean=0.0908425; time: 0.0 min

## [Tune-x] 13: k=13

## [Tune-y] 13: mmce.test.mean=0.0908425; time: 0.0 min

## [Tune-x] 14: k=14

## [Tune-y] 14: mmce.test.mean=0.0896520; time: 0.0 min

## [Tune-x] 15: k=15

## [Tune-y] 15: mmce.test.mean=0.0825092; time: 0.0 min
## [Tune-x] 16: k=16

## [Tune-y] 16: mmce.test.mean=0.0908425; time: 0.0 min

## [Tune-x] 17: k=17

## [Tune-y] 17: mmce.test.mean=0.0979853; time: 0.0 min

## [Tune-x] 18: k=18

## [Tune-y] 18: mmce.test.mean=0.0986264; time: 0.0 min

## [Tune-x] 19: k=19

## [Tune-y] 19: mmce.test.mean=0.1134615; time: 0.0 min

## [Tune-x] 20: k=20

## [Tune-y] 20: mmce.test.mean=0.1057692; time: 0.0 min

## [Tune] Result: k=5 : mmce.test.mean=0.0606227

## [Resample] iter 20:   0.1538462 0.8461538

## [Tune] Started tuning learner classif.knn for parameter set:

##       Type len Def                                   Constr Req Tunable Trafo
## k discrete   -   - 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,1...   -    TRUE     -

## With control class: TuneControlGrid

## Imputation value: 1

## [Tune-x] 1: k=1
## [Tune-y] 1: mmce.test.mean=0.1054945; time: 0.0 min

## [Tune-x] 2: k=2

## [Tune-y] 2: mmce.test.mean=0.0972527; time: 0.0 min

## [Tune-x] 3: k=3

## [Tune-y] 3: mmce.test.mean=0.1049451; time: 0.0 min

## [Tune-x] 4: k=4

## [Tune-y] 4: mmce.test.mean=0.0972527; time: 0.0 min

## [Tune-x] 5: k=5

## [Tune-y] 5: mmce.test.mean=0.0824176; time: 0.0 min

## [Tune-x] 6: k=6

## [Tune-y] 6: mmce.test.mean=0.0747253; time: 0.0 min

## [Tune-x] 7: k=7

## [Tune-y] 7: mmce.test.mean=0.0818681; time: 0.0 min

## [Tune-x] 8: k=8

## [Tune-y] 8: mmce.test.mean=0.0818681; time: 0.0 min

## [Tune-x] 9: k=9

## [Tune-y] 9: mmce.test.mean=0.0747253; time: 0.0 min

## [Tune-x] 10: k=10
## [Tune-y] 10: mmce.test.mean=0.0593407; time: 0.0 min

## [Tune-x] 11: k=11

## [Tune-y] 11: mmce.test.mean=0.0956044; time: 0.0 min

## [Tune-x] 12: k=12

## [Tune-y] 12: mmce.test.mean=0.0956044; time: 0.0 min

## [Tune-x] 13: k=13

## [Tune-y] 13: mmce.test.mean=0.0807692; time: 0.0 min

## [Tune-x] 14: k=14

## [Tune-y] 14: mmce.test.mean=0.0807692; time: 0.0 min

## [Tune-x] 15: k=15

## [Tune-y] 15: mmce.test.mean=0.0950549; time: 0.0 min

## [Tune-x] 16: k=16

## [Tune-y] 16: mmce.test.mean=0.1032967; time: 0.0 min

## [Tune-x] 17: k=17

## [Tune-y] 17: mmce.test.mean=0.0956044; time: 0.0 min

## [Tune-x] 18: k=18

## [Tune-y] 18: mmce.test.mean=0.1032967; time: 0.0 min

## [Tune-x] 19: k=19
## [Tune-y] 19: mmce.test.mean=0.1186813; time: 0.0 min

## [Tune-x] 20: k=20

## [Tune-y] 20: mmce.test.mean=0.1115385; time: 0.0 min

## [Tune] Result: k=10 : mmce.test.mean=0.0593407

## [Resample] iter 21:   0.0909091 0.9090909

## [Tune] Started tuning learner classif.knn for parameter set:

##       Type len Def                                   Constr Req Tunable Trafo
## k discrete   -   - 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,1...   -    TRUE     -

## With control class: TuneControlGrid

## Imputation value: 1

## [Tune-x] 1: k=1

## [Tune-y] 1: mmce.test.mean=0.1203297; time: 0.0 min

## [Tune-x] 2: k=2

## [Tune-y] 2: mmce.test.mean=0.1126374; time: 0.0 min

## [Tune-x] 3: k=3

## [Tune-y] 3: mmce.test.mean=0.0972527; time: 0.0 min

## [Tune-x] 4: k=4

## [Tune-y] 4: mmce.test.mean=0.0978022; time: 0.0 min
## [Tune-x] 5: k=5

## [Tune-y] 5: mmce.test.mean=0.0895604; time: 0.0 min

## [Tune-x] 6: k=6

## [Tune-y] 6: mmce.test.mean=0.0895604; time: 0.0 min

## [Tune-x] 7: k=7

## [Tune-y] 7: mmce.test.mean=0.0895604; time: 0.0 min

## [Tune-x] 8: k=8

## [Tune-y] 8: mmce.test.mean=0.1049451; time: 0.0 min

## [Tune-x] 9: k=9

## [Tune-y] 9: mmce.test.mean=0.0818681; time: 0.0 min

## [Tune-x] 10: k=10

## [Tune-y] 10: mmce.test.mean=0.1043956; time: 0.0 min

## [Tune-x] 11: k=11

## [Tune-y] 11: mmce.test.mean=0.1043956; time: 0.0 min

## [Tune-x] 12: k=12

## [Tune-y] 12: mmce.test.mean=0.1043956; time: 0.0 min

## [Tune-x] 13: k=13

## [Tune-y] 13: mmce.test.mean=0.0967033; time: 0.0 min
## [Tune-x] 14: k=14

## [Tune-y] 14: mmce.test.mean=0.0890110; time: 0.0 min

## [Tune-x] 15: k=15

## [Tune-y] 15: mmce.test.mean=0.0961538; time: 0.0 min

## [Tune-x] 16: k=16

## [Tune-y] 16: mmce.test.mean=0.1038462; time: 0.0 min

## [Tune-x] 17: k=17

## [Tune-y] 17: mmce.test.mean=0.1192308; time: 0.0 min

## [Tune-x] 18: k=18

## [Tune-y] 18: mmce.test.mean=0.1269231; time: 0.0 min

## [Tune-x] 19: k=19

## [Tune-y] 19: mmce.test.mean=0.1043956; time: 0.0 min

## [Tune-x] 20: k=20

## [Tune-y] 20: mmce.test.mean=0.1120879; time: 0.0 min

## [Tune] Result: k=9 : mmce.test.mean=0.0818681

## [Resample] iter 22:   0.0000000 1.0000000

## [Tune] Started tuning learner classif.knn for parameter set:

##       Type len Def                                   Constr Req Tunable Trafo
## k discrete   -   - 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,1...   -    TRUE     -
## With control class: TuneControlGrid

## Imputation value: 1

## [Tune-x] 1: k=1

## [Tune-y] 1: mmce.test.mean=0.0906593; time: 0.0 min

## [Tune-x] 2: k=2

## [Tune-y] 2: mmce.test.mean=0.0906593; time: 0.0 min

## [Tune-x] 3: k=3

## [Tune-y] 3: mmce.test.mean=0.0829670; time: 0.0 min

## [Tune-x] 4: k=4

## [Tune-y] 4: mmce.test.mean=0.0675824; time: 0.0 min

## [Tune-x] 5: k=5

## [Tune-y] 5: mmce.test.mean=0.0598901; time: 0.0 min

## [Tune-x] 6: k=6

## [Tune-y] 6: mmce.test.mean=0.0670330; time: 0.0 min

## [Tune-x] 7: k=7

## [Tune-y] 7: mmce.test.mean=0.0741758; time: 0.0 min

## [Tune-x] 8: k=8

## [Tune-y] 8: mmce.test.mean=0.0813187; time: 0.0 min
## [Tune-x] 9: k=9

## [Tune-y] 9: mmce.test.mean=0.0890110; time: 0.0 min

## [Tune-x] 10: k=10

## [Tune-y] 10: mmce.test.mean=0.0890110; time: 0.0 min

## [Tune-x] 11: k=11

## [Tune-y] 11: mmce.test.mean=0.0813187; time: 0.0 min

## [Tune-x] 12: k=12

## [Tune-y] 12: mmce.test.mean=0.0890110; time: 0.0 min

## [Tune-x] 13: k=13

## [Tune-y] 13: mmce.test.mean=0.0813187; time: 0.0 min

## [Tune-x] 14: k=14

## [Tune-y] 14: mmce.test.mean=0.1038462; time: 0.0 min

## [Tune-x] 15: k=15

## [Tune-y] 15: mmce.test.mean=0.0961538; time: 0.0 min

## [Tune-x] 16: k=16

## [Tune-y] 16: mmce.test.mean=0.1038462; time: 0.0 min

## [Tune-x] 17: k=17

## [Tune-y] 17: mmce.test.mean=0.1038462; time: 0.0 min
## [Tune-x] 18: k=18

## [Tune-y] 18: mmce.test.mean=0.1181319; time: 0.0 min

## [Tune-x] 19: k=19

## [Tune-y] 19: mmce.test.mean=0.1032967; time: 0.0 min

## [Tune-x] 20: k=20

## [Tune-y] 20: mmce.test.mean=0.0961538; time: 0.0 min

## [Tune] Result: k=5 : mmce.test.mean=0.0598901

## [Resample] iter 23:   0.2500000 0.7500000

## [Tune] Started tuning learner classif.knn for parameter set:

##       Type len Def                                   Constr Req Tunable Trafo
## k discrete   -   - 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,1...   -    TRUE     -

## With control class: TuneControlGrid

## Imputation value: 1

## [Tune-x] 1: k=1

## [Tune-y] 1: mmce.test.mean=0.1038462; time: 0.0 min

## [Tune-x] 2: k=2

## [Tune-y] 2: mmce.test.mean=0.0747253; time: 0.0 min

## [Tune-x] 3: k=3
## [Tune-y] 3: mmce.test.mean=0.0521978; time: 0.0 min

## [Tune-x] 4: k=4

## [Tune-y] 4: mmce.test.mean=0.0450549; time: 0.0 min

## [Tune-x] 5: k=5

## [Tune-y] 5: mmce.test.mean=0.0598901; time: 0.0 min

## [Tune-x] 6: k=6

## [Tune-y] 6: mmce.test.mean=0.0670330; time: 0.0 min

## [Tune-x] 7: k=7

## [Tune-y] 7: mmce.test.mean=0.0593407; time: 0.0 min

## [Tune-x] 8: k=8

## [Tune-y] 8: mmce.test.mean=0.0813187; time: 0.0 min

## [Tune-x] 9: k=9

## [Tune-y] 9: mmce.test.mean=0.0736264; time: 0.0 min

## [Tune-x] 10: k=10

## [Tune-y] 10: mmce.test.mean=0.0818681; time: 0.0 min

## [Tune-x] 11: k=11

## [Tune-y] 11: mmce.test.mean=0.0884615; time: 0.0 min

## [Tune-x] 12: k=12
## [Tune-y] 12: mmce.test.mean=0.0961538; time: 0.0 min

## [Tune-x] 13: k=13

## [Tune-y] 13: mmce.test.mean=0.0967033; time: 0.0 min

## [Tune-x] 14: k=14

## [Tune-y] 14: mmce.test.mean=0.0967033; time: 0.0 min

## [Tune-x] 15: k=15

## [Tune-y] 15: mmce.test.mean=0.1043956; time: 0.0 min

## [Tune-x] 16: k=16

## [Tune-y] 16: mmce.test.mean=0.1043956; time: 0.0 min

## [Tune-x] 17: k=17

## [Tune-y] 17: mmce.test.mean=0.0967033; time: 0.0 min

## [Tune-x] 18: k=18

## [Tune-y] 18: mmce.test.mean=0.1032967; time: 0.0 min

## [Tune-x] 19: k=19

## [Tune-y] 19: mmce.test.mean=0.0961538; time: 0.0 min

## [Tune-x] 20: k=20

## [Tune-y] 20: mmce.test.mean=0.1032967; time: 0.0 min

## [Tune] Result: k=4 : mmce.test.mean=0.0450549
## [Resample] iter 24:   0.1818182 0.8181818

## [Tune] Started tuning learner classif.knn for parameter set:

##       Type len Def                                   Constr Req Tunable Trafo
## k discrete   -   - 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,1...   -    TRUE     -

## With control class: TuneControlGrid

## Imputation value: 1

## [Tune-x] 1: k=1

## [Tune-y] 1: mmce.test.mean=0.1115385; time: 0.0 min

## [Tune-x] 2: k=2

## [Tune-y] 2: mmce.test.mean=0.0967033; time: 0.0 min

## [Tune-x] 3: k=3

## [Tune-y] 3: mmce.test.mean=0.0813187; time: 0.0 min

## [Tune-x] 4: k=4

## [Tune-y] 4: mmce.test.mean=0.0664835; time: 0.0 min

## [Tune-x] 5: k=5

## [Tune-y] 5: mmce.test.mean=0.0813187; time: 0.0 min

## [Tune-x] 6: k=6

## [Tune-y] 6: mmce.test.mean=0.0890110; time: 0.0 min
## [Tune-x] 7: k=7

## [Tune-y] 7: mmce.test.mean=0.0741758; time: 0.0 min

## [Tune-x] 8: k=8

## [Tune-y] 8: mmce.test.mean=0.0884615; time: 0.0 min

## [Tune-x] 9: k=9

## [Tune-y] 9: mmce.test.mean=0.0813187; time: 0.0 min

## [Tune-x] 10: k=10

## [Tune-y] 10: mmce.test.mean=0.0741758; time: 0.0 min

## [Tune-x] 11: k=11

## [Tune-y] 11: mmce.test.mean=0.0813187; time: 0.0 min

## [Tune-x] 12: k=12

## [Tune-y] 12: mmce.test.mean=0.0884615; time: 0.0 min

## [Tune-x] 13: k=13

## [Tune-y] 13: mmce.test.mean=0.0884615; time: 0.0 min

## [Tune-x] 14: k=14

## [Tune-y] 14: mmce.test.mean=0.0884615; time: 0.0 min

## [Tune-x] 15: k=15

## [Tune-y] 15: mmce.test.mean=0.0961538; time: 0.0 min
## [Tune-x] 16: k=16

## [Tune-y] 16: mmce.test.mean=0.1104396; time: 0.0 min

## [Tune-x] 17: k=17

## [Tune-y] 17: mmce.test.mean=0.1104396; time: 0.0 min

## [Tune-x] 18: k=18

## [Tune-y] 18: mmce.test.mean=0.1104396; time: 0.0 min

## [Tune-x] 19: k=19

## [Tune-y] 19: mmce.test.mean=0.1027473; time: 0.0 min

## [Tune-x] 20: k=20

## [Tune-y] 20: mmce.test.mean=0.1027473; time: 0.0 min

## [Tune] Result: k=4 : mmce.test.mean=0.0664835

## [Resample] iter 25:   0.0909091 0.9090909

## [Tune] Started tuning learner classif.knn for parameter set:

##       Type len Def                                  Constr Req Tunable Trafo
## k discrete   -   - 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,1...   -    TRUE     -

## With control class: TuneControlGrid

## Imputation value: 1

## [Tune-x] 1: k=1
## [Tune-y] 1: mmce.test.mean=0.1285714; time: 0.0 min

## [Tune-x] 2: k=2

## [Tune-y] 2: mmce.test.mean=0.1280220; time: 0.0 min

## [Tune-x] 3: k=3

## [Tune-y] 3: mmce.test.mean=0.0978022; time: 0.0 min

## [Tune-x] 4: k=4

## [Tune-y] 4: mmce.test.mean=0.0901099; time: 0.0 min

## [Tune-x] 5: k=5

## [Tune-y] 5: mmce.test.mean=0.0758242; time: 0.0 min

## [Tune-x] 6: k=6

## [Tune-y] 6: mmce.test.mean=0.0835165; time: 0.0 min

## [Tune-x] 7: k=7

## [Tune-y] 7: mmce.test.mean=0.0829670; time: 0.0 min

## [Tune-x] 8: k=8

## [Tune-y] 8: mmce.test.mean=0.0983516; time: 0.0 min

## [Tune-x] 9: k=9

## [Tune-y] 9: mmce.test.mean=0.0829670; time: 0.0 min

## [Tune-x] 10: k=10
## [Tune-y] 10: mmce.test.mean=0.0829670; time: 0.0 min

## [Tune-x] 11: k=11

## [Tune-y] 11: mmce.test.mean=0.0835165; time: 0.0 min

## [Tune-x] 12: k=12

## [Tune-y] 12: mmce.test.mean=0.0989011; time: 0.0 min

## [Tune-x] 13: k=13

## [Tune-y] 13: mmce.test.mean=0.0906593; time: 0.0 min

## [Tune-x] 14: k=14

## [Tune-y] 14: mmce.test.mean=0.0978022; time: 0.0 min

## [Tune-x] 15: k=15

## [Tune-y] 15: mmce.test.mean=0.1131868; time: 0.0 min

## [Tune-x] 16: k=16

## [Tune-y] 16: mmce.test.mean=0.1131868; time: 0.0 min

## [Tune-x] 17: k=17

## [Tune-y] 17: mmce.test.mean=0.1054945; time: 0.0 min

## [Tune-x] 18: k=18

## [Tune-y] 18: mmce.test.mean=0.0901099; time: 0.0 min

## [Tune-x] 19: k=19
## [Tune-y] 19: mmce.test.mean=0.1131868; time: 0.0 min

## [Tune-x] 20: k=20

## [Tune-y] 20: mmce.test.mean=0.1208791; time: 0.0 min

## [Tune] Result: k=5 : mmce.test.mean=0.0758242

## [Resample] iter 26:   0.0000000 1.0000000

## [Tune] Started tuning learner classif.knn for parameter set:

##       Type len Def                                   Constr Req Tunable Trafo
## k discrete   -   - 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,1...   -    TRUE     -

## With control class: TuneControlGrid

## Imputation value: 1

## [Tune-x] 1: k=1

## [Tune-y] 1: mmce.test.mean=0.0972527; time: 0.0 min

## [Tune-x] 2: k=2

## [Tune-y] 2: mmce.test.mean=0.0747253; time: 0.0 min

## [Tune-x] 3: k=3

## [Tune-y] 3: mmce.test.mean=0.0752747; time: 0.0 min

## [Tune-x] 4: k=4

## [Tune-y] 4: mmce.test.mean=0.0752747; time: 0.0 min
## [Tune-x] 5: k=5

## [Tune-y] 5: mmce.test.mean=0.0675824; time: 0.0 min

## [Tune-x] 6: k=6

## [Tune-y] 6: mmce.test.mean=0.0675824; time: 0.0 min

## [Tune-x] 7: k=7

## [Tune-y] 7: mmce.test.mean=0.0675824; time: 0.0 min

## [Tune-x] 8: k=8

## [Tune-y] 8: mmce.test.mean=0.0752747; time: 0.0 min

## [Tune-x] 9: k=9

## [Tune-y] 9: mmce.test.mean=0.0829670; time: 0.0 min

## [Tune-x] 10: k=10

## [Tune-y] 10: mmce.test.mean=0.0829670; time: 0.0 min

## [Tune-x] 11: k=11

## [Tune-y] 11: mmce.test.mean=0.0972527; time: 0.0 min

## [Tune-x] 12: k=12

## [Tune-y] 12: mmce.test.mean=0.1043956; time: 0.0 min

## [Tune-x] 13: k=13

## [Tune-y] 13: mmce.test.mean=0.0972527; time: 0.0 min
## [Tune-x] 14: k=14

## [Tune-y] 14: mmce.test.mean=0.0895604; time: 0.0 min

## [Tune-x] 15: k=15

## [Tune-y] 15: mmce.test.mean=0.1043956; time: 0.0 min

## [Tune-x] 16: k=16

## [Tune-y] 16: mmce.test.mean=0.1043956; time: 0.0 min

## [Tune-x] 17: k=17

## [Tune-y] 17: mmce.test.mean=0.1043956; time: 0.0 min

## [Tune-x] 18: k=18

## [Tune-y] 18: mmce.test.mean=0.1043956; time: 0.0 min

## [Tune-x] 19: k=19

## [Tune-y] 19: mmce.test.mean=0.0972527; time: 0.0 min

## [Tune-x] 20: k=20

## [Tune-y] 20: mmce.test.mean=0.1120879; time: 0.0 min

## [Tune] Result: k=5 : mmce.test.mean=0.0675824

## [Resample] iter 27:   0.0833333 0.9166667

## [Tune] Started tuning learner classif.knn for parameter set:

##       Type len Def                                   Constr Req Tunable Trafo
## k discrete   -   - 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,1...   -    TRUE     -
## With control class: TuneControlGrid

## Imputation value: 1

## [Tune-x] 1: k=1

## [Tune-y] 1: mmce.test.mean=0.1208791; time: 0.0 min

## [Tune-x] 2: k=2

## [Tune-y] 2: mmce.test.mean=0.1197802; time: 0.0 min

## [Tune-x] 3: k=3

## [Tune-y] 3: mmce.test.mean=0.0978022; time: 0.0 min

## [Tune-x] 4: k=4

## [Tune-y] 4: mmce.test.mean=0.0983516; time: 0.0 min

## [Tune-x] 5: k=5

## [Tune-y] 5: mmce.test.mean=0.0906593; time: 0.0 min

## [Tune-x] 6: k=6

## [Tune-y] 6: mmce.test.mean=0.0824176; time: 0.0 min

## [Tune-x] 7: k=7

## [Tune-y] 7: mmce.test.mean=0.0681319; time: 0.0 min

## [Tune-x] 8: k=8

## [Tune-y] 8: mmce.test.mean=0.0978022; time: 0.0 min
## [Tune-x] 9: k=9

## [Tune-y] 9: mmce.test.mean=0.1049451; time: 0.0 min

## [Tune-x] 10: k=10

## [Tune-y] 10: mmce.test.mean=0.1054945; time: 0.0 min

## [Tune-x] 11: k=11

## [Tune-y] 11: mmce.test.mean=0.0978022; time: 0.0 min

## [Tune-x] 12: k=12

## [Tune-y] 12: mmce.test.mean=0.1049451; time: 0.0 min

## [Tune-x] 13: k=13

## [Tune-y] 13: mmce.test.mean=0.1115385; time: 0.0 min

## [Tune-x] 14: k=14

## [Tune-y] 14: mmce.test.mean=0.1115385; time: 0.0 min

## [Tune-x] 15: k=15

## [Tune-y] 15: mmce.test.mean=0.0967033; time: 0.0 min

## [Tune-x] 16: k=16

## [Tune-y] 16: mmce.test.mean=0.1038462; time: 0.0 min

## [Tune-x] 17: k=17

## [Tune-y] 17: mmce.test.mean=0.1115385; time: 0.0 min
## [Tune-x] 18: k=18

## [Tune-y] 18: mmce.test.mean=0.1263736; time: 0.0 min

## [Tune-x] 19: k=19

## [Tune-y] 19: mmce.test.mean=0.1263736; time: 0.0 min

## [Tune-x] 20: k=20

## [Tune-y] 20: mmce.test.mean=0.1263736; time: 0.0 min

## [Tune] Result: k=7 : mmce.test.mean=0.0681319

## [Resample] iter 28:   0.0000000 1.0000000

## [Tune] Started tuning learner classif.knn for parameter set:

##       Type len Def                                  Constr Req Tunable Trafo
## k discrete   -   - 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,1...   -    TRUE     -

## With control class: TuneControlGrid

## Imputation value: 1

## [Tune-x] 1: k=1

## [Tune-y] 1: mmce.test.mean=0.1269231; time: 0.0 min

## [Tune-x] 2: k=2

## [Tune-y] 2: mmce.test.mean=0.0906593; time: 0.0 min

## [Tune-x] 3: k=3
## [Tune-y] 3: mmce.test.mean=0.0989011; time: 0.0 min

## [Tune-x] 4: k=4

## [Tune-y] 4: mmce.test.mean=0.0983516; time: 0.0 min

## [Tune-x] 5: k=5

## [Tune-y] 5: mmce.test.mean=0.0912088; time: 0.0 min

## [Tune-x] 6: k=6

## [Tune-y] 6: mmce.test.mean=0.0912088; time: 0.0 min

## [Tune-x] 7: k=7

## [Tune-y] 7: mmce.test.mean=0.0912088; time: 0.0 min

## [Tune-x] 8: k=8

## [Tune-y] 8: mmce.test.mean=0.0906593; time: 0.0 min

## [Tune-x] 9: k=9

## [Tune-y] 9: mmce.test.mean=0.0989011; time: 0.0 min

## [Tune-x] 10: k=10

## [Tune-y] 10: mmce.test.mean=0.0983516; time: 0.0 min

## [Tune-x] 11: k=11

## [Tune-y] 11: mmce.test.mean=0.0906593; time: 0.0 min

## [Tune-x] 12: k=12
## [Tune-y] 12: mmce.test.mean=0.0835165; time: 0.0 min

## [Tune-x] 13: k=13

## [Tune-y] 13: mmce.test.mean=0.0978022; time: 0.0 min

## [Tune-x] 14: k=14

## [Tune-y] 14: mmce.test.mean=0.0835165; time: 0.0 min

## [Tune-x] 15: k=15

## [Tune-y] 15: mmce.test.mean=0.0912088; time: 0.0 min

## [Tune-x] 16: k=16

## [Tune-y] 16: mmce.test.mean=0.0912088; time: 0.0 min

## [Tune-x] 17: k=17

## [Tune-y] 17: mmce.test.mean=0.0983516; time: 0.0 min

## [Tune-x] 18: k=18

## [Tune-y] 18: mmce.test.mean=0.1054945; time: 0.0 min

## [Tune-x] 19: k=19

## [Tune-y] 19: mmce.test.mean=0.1054945; time: 0.0 min

## [Tune-x] 20: k=20

## [Tune-y] 20: mmce.test.mean=0.1054945; time: 0.0 min

## [Tune] Result: k=12 : mmce.test.mean=0.0835165
## [Resample] iter 29:   0.0833333 0.9166667

## [Tune] Started tuning learner classif.knn for parameter set:

##       Type len Def                                   Constr Req Tunable Trafo
## k discrete   -   - 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,1...   -    TRUE     -

## With control class: TuneControlGrid

## Imputation value: 1

## [Tune-x] 1: k=1

## [Tune-y] 1: mmce.test.mean=0.1002747; time: 0.0 min

## [Tune-x] 2: k=2

## [Tune-y] 2: mmce.test.mean=0.1074176; time: 0.0 min

## [Tune-x] 3: k=3

## [Tune-y] 3: mmce.test.mean=0.0777473; time: 0.0 min

## [Tune-x] 4: k=4

## [Tune-y] 4: mmce.test.mean=0.0777473; time: 0.0 min

## [Tune-x] 5: k=5

## [Tune-y] 5: mmce.test.mean=0.0765568; time: 0.0 min

## [Tune-x] 6: k=6

## [Tune-y] 6: mmce.test.mean=0.0925824; time: 0.0 min
## [Tune-x] 7: k=7

## [Tune-y] 7: mmce.test.mean=0.0854396; time: 0.0 min

## [Tune-x] 8: k=8

## [Tune-y] 8: mmce.test.mean=0.1002747; time: 0.0 min

## [Tune-x] 9: k=9

## [Tune-y] 9: mmce.test.mean=0.0854396; time: 0.0 min

## [Tune-x] 10: k=10

## [Tune-y] 10: mmce.test.mean=0.0847985; time: 0.0 min

## [Tune-x] 11: k=11

## [Tune-y] 11: mmce.test.mean=0.0924908; time: 0.0 min

## [Tune-x] 12: k=12

## [Tune-y] 12: mmce.test.mean=0.0847985; time: 0.0 min

## [Tune-x] 13: k=13

## [Tune-y] 13: mmce.test.mean=0.1008242; time: 0.0 min

## [Tune-x] 14: k=14

## [Tune-y] 14: mmce.test.mean=0.1085165; time: 0.0 min

## [Tune-x] 15: k=15

## [Tune-y] 15: mmce.test.mean=0.1008242; time: 0.0 min
## [Tune-x] 16: k=16

## [Tune-y] 16: mmce.test.mean=0.0931319; time: 0.0 min

## [Tune-x] 17: k=17

## [Tune-y] 17: mmce.test.mean=0.1079670; time: 0.0 min

## [Tune-x] 18: k=18

## [Tune-y] 18: mmce.test.mean=0.1002747; time: 0.0 min

## [Tune-x] 19: k=19

## [Tune-y] 19: mmce.test.mean=0.1228022; time: 0.0 min

## [Tune-x] 20: k=20

## [Tune-y] 20: mmce.test.mean=0.1228022; time: 0.0 min

## [Tune] Result: k=5 : mmce.test.mean=0.0765568

## [Resample] iter 30:   0.1538462 0.8461538

## [Tune] Started tuning learner classif.knn for parameter set:

##       Type len Def                                   Constr Req Tunable Trafo
## k discrete   -   - 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,1...   -    TRUE     -

## With control class: TuneControlGrid

## Imputation value: 1

## [Tune-x] 1: k=1
## [Tune-y] 1: mmce.test.mean=0.0890110; time: 0.0 min

## [Tune-x] 2: k=2

## [Tune-y] 2: mmce.test.mean=0.0895604; time: 0.0 min

## [Tune-x] 3: k=3

## [Tune-y] 3: mmce.test.mean=0.0741758; time: 0.0 min

## [Tune-x] 4: k=4

## [Tune-y] 4: mmce.test.mean=0.0741758; time: 0.0 min

## [Tune-x] 5: k=5

## [Tune-y] 5: mmce.test.mean=0.0664835; time: 0.0 min

## [Tune-x] 6: k=6

## [Tune-y] 6: mmce.test.mean=0.0895604; time: 0.0 min

## [Tune-x] 7: k=7

## [Tune-y] 7: mmce.test.mean=0.0895604; time: 0.0 min

## [Tune-x] 8: k=8

## [Tune-y] 8: mmce.test.mean=0.0741758; time: 0.0 min

## [Tune-x] 9: k=9

## [Tune-y] 9: mmce.test.mean=0.0664835; time: 0.0 min

## [Tune-x] 10: k=10
## [Tune-y] 10: mmce.test.mean=0.0664835; time: 0.0 min

## [Tune-x] 11: k=11

## [Tune-y] 11: mmce.test.mean=0.0664835; time: 0.0 min

## [Tune-x] 12: k=12

## [Tune-y] 12: mmce.test.mean=0.0818681; time: 0.0 min

## [Tune-x] 13: k=13

## [Tune-y] 13: mmce.test.mean=0.0818681; time: 0.0 min

## [Tune-x] 14: k=14

## [Tune-y] 14: mmce.test.mean=0.0967033; time: 0.0 min

## [Tune-x] 15: k=15

## [Tune-y] 15: mmce.test.mean=0.0890110; time: 0.0 min

## [Tune-x] 16: k=16

## [Tune-y] 16: mmce.test.mean=0.0890110; time: 0.0 min

## [Tune-x] 17: k=17

## [Tune-y] 17: mmce.test.mean=0.1043956; time: 0.0 min

## [Tune-x] 18: k=18

## [Tune-y] 18: mmce.test.mean=0.1043956; time: 0.0 min

## [Tune-x] 19: k=19
## [Tune-y] 19: mmce.test.mean=0.1043956; time: 0.0 min

## [Tune-x] 20: k=20

## [Tune-y] 20: mmce.test.mean=0.1043956; time: 0.0 min

## [Tune] Result: k=9 : mmce.test.mean=0.0664835

## [Resample] iter 31:   0.1666667 0.8333333

## [Tune] Started tuning learner classif.knn for parameter set:

##       Type len Def                                  Constr Req Tunable Trafo
## k discrete   -   - 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,1...   -    TRUE     -

## With control class: TuneControlGrid

## Imputation value: 1

## [Tune-x] 1: k=1

## [Tune-y] 1: mmce.test.mean=0.1352564; time: 0.0 min

## [Tune-x] 2: k=2

## [Tune-y] 2: mmce.test.mean=0.1215201; time: 0.0 min

## [Tune-x] 3: k=3

## [Tune-y] 3: mmce.test.mean=0.0913004; time: 0.0 min

## [Tune-x] 4: k=4

## [Tune-y] 4: mmce.test.mean=0.1220696; time: 0.0 min
## [Tune-x] 5: k=5

## [Tune-y] 5: mmce.test.mean=0.0913004; time: 0.0 min

## [Tune-x] 6: k=6

## [Tune-y] 6: mmce.test.mean=0.1001832; time: 0.0 min

## [Tune-x] 7: k=7

## [Tune-y] 7: mmce.test.mean=0.0996337; time: 0.0 min

## [Tune-x] 8: k=8

## [Tune-y] 8: mmce.test.mean=0.1001832; time: 0.0 min

## [Tune-x] 9: k=9

## [Tune-y] 9: mmce.test.mean=0.0924908; time: 0.0 min

## [Tune-x] 10: k=10

## [Tune-y] 10: mmce.test.mean=0.1078755; time: 0.0 min

## [Tune-x] 11: k=11

## [Tune-y] 11: mmce.test.mean=0.0930403; time: 0.0 min

## [Tune-x] 12: k=12

## [Tune-y] 12: mmce.test.mean=0.1001832; time: 0.0 min

## [Tune-x] 13: k=13

## [Tune-y] 13: mmce.test.mean=0.1007326; time: 0.0 min
## [Tune-x] 14: k=14

## [Tune-y] 14: mmce.test.mean=0.0930403; time: 0.0 min

## [Tune-x] 15: k=15

## [Tune-y] 15: mmce.test.mean=0.1078755; time: 0.0 min

## [Tune-x] 16: k=16

## [Tune-y] 16: mmce.test.mean=0.1078755; time: 0.0 min

## [Tune-x] 17: k=17

## [Tune-y] 17: mmce.test.mean=0.1078755; time: 0.0 min

## [Tune-x] 18: k=18

## [Tune-y] 18: mmce.test.mean=0.1078755; time: 0.0 min

## [Tune-x] 19: k=19

## [Tune-y] 19: mmce.test.mean=0.1001832; time: 0.0 min

## [Tune-x] 20: k=20

## [Tune-y] 20: mmce.test.mean=0.1078755; time: 0.0 min

## [Tune] Result: k=3 : mmce.test.mean=0.0913004

## [Resample] iter 32:   0.0000000 1.0000000

## [Tune] Started tuning learner classif.knn for parameter set:

##       Type len Def                                  Constr Req Tunable Trafo
## k discrete   -   - 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,1...   -    TRUE     -
## With control class: TuneControlGrid

## Imputation value: 1

## [Tune-x] 1: k=1

## [Tune-y] 1: mmce.test.mean=0.1434066; time: 0.0 min

## [Tune-x] 2: k=2

## [Tune-y] 2: mmce.test.mean=0.1351648; time: 0.0 min

## [Tune-x] 3: k=3

## [Tune-y] 3: mmce.test.mean=0.0829670; time: 0.0 min

## [Tune-x] 4: k=4

## [Tune-y] 4: mmce.test.mean=0.0675824; time: 0.0 min

## [Tune-x] 5: k=5

## [Tune-y] 5: mmce.test.mean=0.0912088; time: 0.0 min

## [Tune-x] 6: k=6

## [Tune-y] 6: mmce.test.mean=0.0906593; time: 0.0 min

## [Tune-x] 7: k=7

## [Tune-y] 7: mmce.test.mean=0.0829670; time: 0.0 min

## [Tune-x] 8: k=8

## [Tune-y] 8: mmce.test.mean=0.0978022; time: 0.0 min
## [Tune-x] 9: k=9

## [Tune-y] 9: mmce.test.mean=0.1054945; time: 0.0 min

## [Tune-x] 10: k=10

## [Tune-y] 10: mmce.test.mean=0.1120879; time: 0.0 min

## [Tune-x] 11: k=11

## [Tune-y] 11: mmce.test.mean=0.0901099; time: 0.0 min

## [Tune-x] 12: k=12

## [Tune-y] 12: mmce.test.mean=0.1049451; time: 0.0 min

## [Tune-x] 13: k=13

## [Tune-y] 13: mmce.test.mean=0.1043956; time: 0.0 min

## [Tune-x] 14: k=14

## [Tune-y] 14: mmce.test.mean=0.1043956; time: 0.0 min

## [Tune-x] 15: k=15

## [Tune-y] 15: mmce.test.mean=0.1043956; time: 0.0 min

## [Tune-x] 16: k=16

## [Tune-y] 16: mmce.test.mean=0.1043956; time: 0.0 min

## [Tune-x] 17: k=17

## [Tune-y] 17: mmce.test.mean=0.1115385; time: 0.0 min
## [Tune-x] 18: k=18

## [Tune-y] 18: mmce.test.mean=0.1120879; time: 0.0 min

## [Tune-x] 19: k=19

## [Tune-y] 19: mmce.test.mean=0.1192308; time: 0.0 min

## [Tune-x] 20: k=20

## [Tune-y] 20: mmce.test.mean=0.0978022; time: 0.0 min

## [Tune] Result: k=4 : mmce.test.mean=0.0675824

## [Resample] iter 33:   0.0909091 0.9090909

## [Tune] Started tuning learner classif.knn for parameter set:

##       Type len Def                                  Constr Req Tunable Trafo
## k discrete   -   - 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,1...   -    TRUE     -

## With control class: TuneControlGrid

## Imputation value: 1

## [Tune-x] 1: k=1

## [Tune-y] 1: mmce.test.mean=0.1056777; time: 0.0 min

## [Tune-x] 2: k=2

## [Tune-y] 2: mmce.test.mean=0.1139194; time: 0.0 min

## [Tune-x] 3: k=3
## [Tune-y] 3: mmce.test.mean=0.0913919; time: 0.0 min

## [Tune-x] 4: k=4

## [Tune-y] 4: mmce.test.mean=0.0907509; time: 0.0 min

## [Tune-x] 5: k=5

## [Tune-y] 5: mmce.test.mean=0.0896520; time: 0.0 min

## [Tune-x] 6: k=6

## [Tune-y] 6: mmce.test.mean=0.0902015; time: 0.0 min

## [Tune-x] 7: k=7

## [Tune-y] 7: mmce.test.mean=0.0682234; time: 0.0 min

## [Tune-x] 8: k=8

## [Tune-y] 8: mmce.test.mean=0.0682234; time: 0.0 min

## [Tune-x] 9: k=9

## [Tune-y] 9: mmce.test.mean=0.0902015; time: 0.0 min

## [Tune-x] 10: k=10

## [Tune-y] 10: mmce.test.mean=0.0896520; time: 0.0 min

## [Tune-x] 11: k=11

## [Tune-y] 11: mmce.test.mean=0.0748168; time: 0.0 min

## [Tune-x] 12: k=12
## [Tune-y] 12: mmce.test.mean=0.0825092; time: 0.0 min

## [Tune-x] 13: k=13

## [Tune-y] 13: mmce.test.mean=0.0676740; time: 0.0 min

## [Tune-x] 14: k=14

## [Tune-y] 14: mmce.test.mean=0.0748168; time: 0.0 min

## [Tune-x] 15: k=15

## [Tune-y] 15: mmce.test.mean=0.0896520; time: 0.0 min

## [Tune-x] 16: k=16

## [Tune-y] 16: mmce.test.mean=0.1121795; time: 0.0 min

## [Tune-x] 17: k=17

## [Tune-y] 17: mmce.test.mean=0.0891026; time: 0.0 min

## [Tune-x] 18: k=18

## [Tune-y] 18: mmce.test.mean=0.0819597; time: 0.0 min

## [Tune-x] 19: k=19

## [Tune-y] 19: mmce.test.mean=0.0967949; time: 0.0 min

## [Tune-x] 20: k=20

## [Tune-y] 20: mmce.test.mean=0.1044872; time: 0.0 min

## [Tune] Result: k=13 : mmce.test.mean=0.0676740
## [Resample] iter 34:   0.1538462 0.8461538

## [Tune] Started tuning learner classif.knn for parameter set:

##       Type len Def                                   Constr Req Tunable Trafo
## k discrete   -   - 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,1...   -    TRUE     -

## With control class: TuneControlGrid

## Imputation value: 1

## [Tune-x] 1: k=1

## [Tune-y] 1: mmce.test.mean=0.0763736; time: 0.0 min

## [Tune-x] 2: k=2

## [Tune-y] 2: mmce.test.mean=0.0373626; time: 0.0 min

## [Tune-x] 3: k=3

## [Tune-y] 3: mmce.test.mean=0.0758242; time: 0.0 min

## [Tune-x] 4: k=4

## [Tune-y] 4: mmce.test.mean=0.0758242; time: 0.0 min

## [Tune-x] 5: k=5

## [Tune-y] 5: mmce.test.mean=0.0609890; time: 0.0 min

## [Tune-x] 6: k=6

## [Tune-y] 6: mmce.test.mean=0.0609890; time: 0.0 min
## [Tune-x] 7: k=7

## [Tune-y] 7: mmce.test.mean=0.0609890; time: 0.0 min

## [Tune-x] 8: k=8

## [Tune-y] 8: mmce.test.mean=0.0769231; time: 0.0 min

## [Tune-x] 9: k=9

## [Tune-y] 9: mmce.test.mean=0.0769231; time: 0.0 min

## [Tune-x] 10: k=10

## [Tune-y] 10: mmce.test.mean=0.0615385; time: 0.0 min

## [Tune-x] 11: k=11

## [Tune-y] 11: mmce.test.mean=0.0769231; time: 0.0 min

## [Tune-x] 12: k=12

## [Tune-y] 12: mmce.test.mean=0.0763736; time: 0.0 min

## [Tune-x] 13: k=13

## [Tune-y] 13: mmce.test.mean=0.0692308; time: 0.0 min

## [Tune-x] 14: k=14

## [Tune-y] 14: mmce.test.mean=0.0763736; time: 0.0 min

## [Tune-x] 15: k=15

## [Tune-y] 15: mmce.test.mean=0.0912088; time: 0.0 min
## [Tune-x] 16: k=16

## [Tune-y] 16: mmce.test.mean=0.0912088; time: 0.0 min

## [Tune-x] 17: k=17

## [Tune-y] 17: mmce.test.mean=0.0923077; time: 0.0 min

## [Tune-x] 18: k=18

## [Tune-y] 18: mmce.test.mean=0.1065934; time: 0.0 min

## [Tune-x] 19: k=19

## [Tune-y] 19: mmce.test.mean=0.0989011; time: 0.0 min

## [Tune-x] 20: k=20

## [Tune-y] 20: mmce.test.mean=0.0989011; time: 0.0 min

## [Tune] Result: k=2 : mmce.test.mean=0.0373626

## [Resample] iter 35:   0.2307692 0.7692308

## [Tune] Started tuning learner classif.knn for parameter set:

##       Type len Def                                   Constr Req Tunable Trafo
## k discrete   -   - 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,1...   -    TRUE     -

## With control class: TuneControlGrid

## Imputation value: 1

## [Tune-x] 1: k=1
## [Tune-y] 1: mmce.test.mean=0.1192308; time: 0.0 min

## [Tune-x] 2: k=2

## [Tune-y] 2: mmce.test.mean=0.1203297; time: 0.0 min

## [Tune-x] 3: k=3

## [Tune-y] 3: mmce.test.mean=0.0978022; time: 0.0 min

## [Tune-x] 4: k=4

## [Tune-y] 4: mmce.test.mean=0.0989011; time: 0.0 min

## [Tune-x] 5: k=5

## [Tune-y] 5: mmce.test.mean=0.0758242; time: 0.0 min

## [Tune-x] 6: k=6

## [Tune-y] 6: mmce.test.mean=0.0906593; time: 0.0 min

## [Tune-x] 7: k=7

## [Tune-y] 7: mmce.test.mean=0.0829670; time: 0.0 min

## [Tune-x] 8: k=8

## [Tune-y] 8: mmce.test.mean=0.0752747; time: 0.0 min

## [Tune-x] 9: k=9

## [Tune-y] 9: mmce.test.mean=0.0901099; time: 0.0 min

## [Tune-x] 10: k=10
## [Tune-y] 10: mmce.test.mean=0.0752747; time: 0.0 min

## [Tune-x] 11: k=11

## [Tune-y] 11: mmce.test.mean=0.0818681; time: 0.0 min

## [Tune-x] 12: k=12

## [Tune-y] 12: mmce.test.mean=0.0824176; time: 0.0 min

## [Tune-x] 13: k=13

## [Tune-y] 13: mmce.test.mean=0.0901099; time: 0.0 min

## [Tune-x] 14: k=14

## [Tune-y] 14: mmce.test.mean=0.0978022; time: 0.0 min

## [Tune-x] 15: k=15

## [Tune-y] 15: mmce.test.mean=0.0978022; time: 0.0 min

## [Tune-x] 16: k=16

## [Tune-y] 16: mmce.test.mean=0.1054945; time: 0.0 min

## [Tune-x] 17: k=17

## [Tune-y] 17: mmce.test.mean=0.0978022; time: 0.0 min

## [Tune-x] 18: k=18

## [Tune-y] 18: mmce.test.mean=0.0978022; time: 0.0 min

## [Tune-x] 19: k=19
## [Tune-y] 19: mmce.test.mean=0.1120879; time: 0.0 min

## [Tune-x] 20: k=20

## [Tune-y] 20: mmce.test.mean=0.1120879; time: 0.0 min

## [Tune] Result: k=10 : mmce.test.mean=0.0752747

## [Resample] iter 36:   0.0000000 1.0000000

## [Tune] Started tuning learner classif.knn for parameter set:

##       Type len Def                                   Constr Req Tunable Trafo
## k discrete   -   - 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,1...   -    TRUE     -

## With control class: TuneControlGrid

## Imputation value: 1

## [Tune-x] 1: k=1

## [Tune-y] 1: mmce.test.mean=0.1043956; time: 0.0 min

## [Tune-x] 2: k=2

## [Tune-y] 2: mmce.test.mean=0.0906593; time: 0.0 min

## [Tune-x] 3: k=3

## [Tune-y] 3: mmce.test.mean=0.1109890; time: 0.0 min

## [Tune-x] 4: k=4

## [Tune-y] 4: mmce.test.mean=0.0890110; time: 0.0 min
## [Tune-x] 5: k=5

## [Tune-y] 5: mmce.test.mean=0.0741758; time: 0.0 min

## [Tune-x] 6: k=6

## [Tune-y] 6: mmce.test.mean=0.0664835; time: 0.0 min

## [Tune-x] 7: k=7

## [Tune-y] 7: mmce.test.mean=0.0659341; time: 0.0 min

## [Tune-x] 8: k=8

## [Tune-y] 8: mmce.test.mean=0.0961538; time: 0.0 min

## [Tune-x] 9: k=9

## [Tune-y] 9: mmce.test.mean=0.0884615; time: 0.0 min

## [Tune-x] 10: k=10

## [Tune-y] 10: mmce.test.mean=0.0890110; time: 0.0 min

## [Tune-x] 11: k=11

## [Tune-y] 11: mmce.test.mean=0.0890110; time: 0.0 min

## [Tune-x] 12: k=12

## [Tune-y] 12: mmce.test.mean=0.0967033; time: 0.0 min

## [Tune-x] 13: k=13

## [Tune-y] 13: mmce.test.mean=0.0813187; time: 0.0 min
## [Tune-x] 14: k=14

## [Tune-y] 14: mmce.test.mean=0.0890110; time: 0.0 min

## [Tune-x] 15: k=15

## [Tune-y] 15: mmce.test.mean=0.0967033; time: 0.0 min

## [Tune-x] 16: k=16

## [Tune-y] 16: mmce.test.mean=0.1043956; time: 0.0 min

## [Tune-x] 17: k=17

## [Tune-y] 17: mmce.test.mean=0.0967033; time: 0.0 min

## [Tune-x] 18: k=18

## [Tune-y] 18: mmce.test.mean=0.0967033; time: 0.0 min

## [Tune-x] 19: k=19

## [Tune-y] 19: mmce.test.mean=0.1043956; time: 0.0 min

## [Tune-x] 20: k=20

## [Tune-y] 20: mmce.test.mean=0.1043956; time: 0.0 min

## [Tune] Result: k=7 : mmce.test.mean=0.0659341

## [Resample] iter 37:   0.0000000 1.0000000

## [Tune] Started tuning learner classif.knn for parameter set:

##       Type len Def                                   Constr Req Tunable Trafo
## k discrete   -   - 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,1...   -    TRUE     -
## With control class: TuneControlGrid

## Imputation value: 1

## [Tune-x] 1: k=1

## [Tune-y] 1: mmce.test.mean=0.0907509; time: 0.0 min

## [Tune-x] 2: k=2

## [Tune-y] 2: mmce.test.mean=0.1061355; time: 0.0 min

## [Tune-x] 3: k=3

## [Tune-y] 3: mmce.test.mean=0.0759158; time: 0.0 min

## [Tune-x] 4: k=4

## [Tune-y] 4: mmce.test.mean=0.0913004; time: 0.0 min

## [Tune-x] 5: k=5

## [Tune-y] 5: mmce.test.mean=0.0902015; time: 0.0 min

## [Tune-x] 6: k=6

## [Tune-y] 6: mmce.test.mean=0.0830586; time: 0.0 min

## [Tune-x] 7: k=7

## [Tune-y] 7: mmce.test.mean=0.0830586; time: 0.0 min

## [Tune-x] 8: k=8

## [Tune-y] 8: mmce.test.mean=0.0984432; time: 0.0 min
## [Tune-x] 9: k=9

## [Tune-y] 9: mmce.test.mean=0.0984432; time: 0.0 min

## [Tune-x] 10: k=10

## [Tune-y] 10: mmce.test.mean=0.0978938; time: 0.0 min

## [Tune-x] 11: k=11

## [Tune-y] 11: mmce.test.mean=0.0907509; time: 0.0 min

## [Tune-x] 12: k=12

## [Tune-y] 12: mmce.test.mean=0.0907509; time: 0.0 min

## [Tune-x] 13: k=13

## [Tune-y] 13: mmce.test.mean=0.0907509; time: 0.0 min

## [Tune-x] 14: k=14

## [Tune-y] 14: mmce.test.mean=0.0902015; time: 0.0 min

## [Tune-x] 15: k=15

## [Tune-y] 15: mmce.test.mean=0.1132784; time: 0.0 min

## [Tune-x] 16: k=16

## [Tune-y] 16: mmce.test.mean=0.1050366; time: 0.0 min

## [Tune-x] 17: k=17

## [Tune-y] 17: mmce.test.mean=0.1121795; time: 0.0 min
## [Tune-x] 18: k=18

## [Tune-y] 18: mmce.test.mean=0.1050366; time: 0.0 min

## [Tune-x] 19: k=19

## [Tune-y] 19: mmce.test.mean=0.1121795; time: 0.0 min

## [Tune-x] 20: k=20

## [Tune-y] 20: mmce.test.mean=0.1121795; time: 0.0 min

## [Tune] Result: k=3 : mmce.test.mean=0.0759158

## [Resample] iter 38:   0.1538462 0.8461538

## [Tune] Started tuning learner classif.knn for parameter set:

##       Type len Def                                   Constr Req Tunable Trafo
## k discrete   -   - 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,1...   -    TRUE     -

## With control class: TuneControlGrid

## Imputation value: 1

## [Tune-x] 1: k=1

## [Tune-y] 1: mmce.test.mean=0.0748168; time: 0.0 min

## [Tune-x] 2: k=2

## [Tune-y] 2: mmce.test.mean=0.0814103; time: 0.0 min

## [Tune-x] 3: k=3
## [Tune-y] 3: mmce.test.mean=0.0907509; time: 0.0 min

## [Tune-x] 4: k=4

## [Tune-y] 4: mmce.test.mean=0.0825092; time: 0.0 min

## [Tune-x] 5: k=5

## [Tune-y] 5: mmce.test.mean=0.0836081; time: 0.0 min

## [Tune-x] 6: k=6

## [Tune-y] 6: mmce.test.mean=0.0753663; time: 0.0 min

## [Tune-x] 7: k=7

## [Tune-y] 7: mmce.test.mean=0.0759158; time: 0.0 min

## [Tune-x] 8: k=8

## [Tune-y] 8: mmce.test.mean=0.0759158; time: 0.0 min

## [Tune-x] 9: k=9

## [Tune-y] 9: mmce.test.mean=0.0687729; time: 0.0 min

## [Tune-x] 10: k=10

## [Tune-y] 10: mmce.test.mean=0.0764652; time: 0.0 min

## [Tune-x] 11: k=11

## [Tune-y] 11: mmce.test.mean=0.0764652; time: 0.0 min

## [Tune-x] 12: k=12
## [Tune-y] 12: mmce.test.mean=0.0687729; time: 0.0 min

## [Tune-x] 13: k=13

## [Tune-y] 13: mmce.test.mean=0.0836081; time: 0.0 min

## [Tune-x] 14: k=14

## [Tune-y] 14: mmce.test.mean=0.0836081; time: 0.0 min

## [Tune-x] 15: k=15

## [Tune-y] 15: mmce.test.mean=0.0841575; time: 0.0 min

## [Tune-x] 16: k=16

## [Tune-y] 16: mmce.test.mean=0.0984432; time: 0.0 min

## [Tune-x] 17: k=17

## [Tune-y] 17: mmce.test.mean=0.0913004; time: 0.0 min

## [Tune-x] 18: k=18

## [Tune-y] 18: mmce.test.mean=0.0907509; time: 0.0 min

## [Tune-x] 19: k=19

## [Tune-y] 19: mmce.test.mean=0.0984432; time: 0.0 min

## [Tune-x] 20: k=20

## [Tune-y] 20: mmce.test.mean=0.0989927; time: 0.0 min

## [Tune] Result: k=12 : mmce.test.mean=0.0687729
## [Resample] iter 39:   0.3076923 0.6923077

## [Tune] Started tuning learner classif.knn for parameter set:

##       Type len Def                                  Constr Req Tunable Trafo
## k discrete   -   - 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,1...   -    TRUE     -

## With control class: TuneControlGrid

## Imputation value: 1

## [Tune-x] 1: k=1

## [Tune-y] 1: mmce.test.mean=0.1038462; time: 0.0 min

## [Tune-x] 2: k=2

## [Tune-y] 2: mmce.test.mean=0.1340659; time: 0.0 min

## [Tune-x] 3: k=3

## [Tune-y] 3: mmce.test.mean=0.0741758; time: 0.0 min

## [Tune-x] 4: k=4

## [Tune-y] 4: mmce.test.mean=0.0521978; time: 0.0 min

## [Tune-x] 5: k=5

## [Tune-y] 5: mmce.test.mean=0.0741758; time: 0.0 min

## [Tune-x] 6: k=6

## [Tune-y] 6: mmce.test.mean=0.0967033; time: 0.0 min
## [Tune-x] 7: k=7

## [Tune-y] 7: mmce.test.mean=0.0741758; time: 0.0 min

## [Tune-x] 8: k=8

## [Tune-y] 8: mmce.test.mean=0.0895604; time: 0.0 min

## [Tune-x] 9: k=9

## [Tune-y] 9: mmce.test.mean=0.0824176; time: 0.0 min

## [Tune-x] 10: k=10

## [Tune-y] 10: mmce.test.mean=0.0895604; time: 0.0 min

## [Tune-x] 11: k=11

## [Tune-y] 11: mmce.test.mean=0.0824176; time: 0.0 min

## [Tune-x] 12: k=12

## [Tune-y] 12: mmce.test.mean=0.0824176; time: 0.0 min

## [Tune-x] 13: k=13

## [Tune-y] 13: mmce.test.mean=0.0895604; time: 0.0 min

## [Tune-x] 14: k=14

## [Tune-y] 14: mmce.test.mean=0.0895604; time: 0.0 min

## [Tune-x] 15: k=15

## [Tune-y] 15: mmce.test.mean=0.1049451; time: 0.0 min
## [Tune-x] 16: k=16

## [Tune-y] 16: mmce.test.mean=0.1043956; time: 0.0 min

## [Tune-x] 17: k=17

## [Tune-y] 17: mmce.test.mean=0.1043956; time: 0.0 min

## [Tune-x] 18: k=18

## [Tune-y] 18: mmce.test.mean=0.1043956; time: 0.0 min

## [Tune-x] 19: k=19

## [Tune-y] 19: mmce.test.mean=0.1038462; time: 0.0 min

## [Tune-x] 20: k=20

## [Tune-y] 20: mmce.test.mean=0.1043956; time: 0.0 min

## [Tune] Result: k=4 : mmce.test.mean=0.0521978

## [Resample] iter 40:   0.0833333 0.9166667

## [Tune] Started tuning learner classif.knn for parameter set:

##       Type len Def                                   Constr Req Tunable Trafo
## k discrete   -   - 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,1...   -    TRUE     -

## With control class: TuneControlGrid

## Imputation value: 1

## [Tune-x] 1: k=1
## [Tune-y] 1: mmce.test.mean=0.0901099; time: 0.0 min

## [Tune-x] 2: k=2

## [Tune-y] 2: mmce.test.mean=0.1054945; time: 0.0 min

## [Tune-x] 3: k=3

## [Tune-y] 3: mmce.test.mean=0.0681319; time: 0.0 min

## [Tune-x] 4: k=4

## [Tune-y] 4: mmce.test.mean=0.0978022; time: 0.0 min

## [Tune-x] 5: k=5

## [Tune-y] 5: mmce.test.mean=0.0675824; time: 0.0 min

## [Tune-x] 6: k=6

## [Tune-y] 6: mmce.test.mean=0.0758242; time: 0.0 min

## [Tune-x] 7: k=7

## [Tune-y] 7: mmce.test.mean=0.0681319; time: 0.0 min

## [Tune-x] 8: k=8

## [Tune-y] 8: mmce.test.mean=0.0906593; time: 0.0 min

## [Tune-x] 9: k=9

## [Tune-y] 9: mmce.test.mean=0.0906593; time: 0.0 min

## [Tune-x] 10: k=10
## [Tune-y] 10: mmce.test.mean=0.0901099; time: 0.0 min

## [Tune-x] 11: k=11

## [Tune-y] 11: mmce.test.mean=0.0752747; time: 0.0 min

## [Tune-x] 12: k=12

## [Tune-y] 12: mmce.test.mean=0.0752747; time: 0.0 min

## [Tune-x] 13: k=13

## [Tune-y] 13: mmce.test.mean=0.0824176; time: 0.0 min

## [Tune-x] 14: k=14

## [Tune-y] 14: mmce.test.mean=0.0972527; time: 0.0 min

## [Tune-x] 15: k=15

## [Tune-y] 15: mmce.test.mean=0.1043956; time: 0.0 min

## [Tune-x] 16: k=16

## [Tune-y] 16: mmce.test.mean=0.1120879; time: 0.0 min

## [Tune-x] 17: k=17

## [Tune-y] 17: mmce.test.mean=0.1120879; time: 0.0 min

## [Tune-x] 18: k=18

## [Tune-y] 18: mmce.test.mean=0.1120879; time: 0.0 min

## [Tune-x] 19: k=19
## [Tune-y] 19: mmce.test.mean=0.1120879; time: 0.0 min

## [Tune-x] 20: k=20

## [Tune-y] 20: mmce.test.mean=0.1120879; time: 0.0 min

## [Tune] Result: k=5 : mmce.test.mean=0.0675824

## [Resample] iter 41:   0.1818182 0.8181818

## [Tune] Started tuning learner classif.knn for parameter set:

##       Type len Def                                   Constr Req Tunable Trafo
## k discrete   -   - 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,1...   -    TRUE     -

## With control class: TuneControlGrid

## Imputation value: 1

## [Tune-x] 1: k=1

## [Tune-y] 1: mmce.test.mean=0.1032967; time: 0.0 min

## [Tune-x] 2: k=2

## [Tune-y] 2: mmce.test.mean=0.0747253; time: 0.0 min

## [Tune-x] 3: k=3

## [Tune-y] 3: mmce.test.mean=0.0956044; time: 0.0 min

## [Tune-x] 4: k=4

## [Tune-y] 4: mmce.test.mean=0.0736264; time: 0.0 min
## [Tune-x] 5: k=5

## [Tune-y] 5: mmce.test.mean=0.0670330; time: 0.0 min

## [Tune-x] 6: k=6

## [Tune-y] 6: mmce.test.mean=0.0593407; time: 0.0 min

## [Tune-x] 7: k=7

## [Tune-y] 7: mmce.test.mean=0.0670330; time: 0.0 min

## [Tune-x] 8: k=8

## [Tune-y] 8: mmce.test.mean=0.0741758; time: 0.0 min

## [Tune-x] 9: k=9

## [Tune-y] 9: mmce.test.mean=0.0813187; time: 0.0 min

## [Tune-x] 10: k=10

## [Tune-y] 10: mmce.test.mean=0.0664835; time: 0.0 min

## [Tune-x] 11: k=11

## [Tune-y] 11: mmce.test.mean=0.0670330; time: 0.0 min

## [Tune-x] 12: k=12

## [Tune-y] 12: mmce.test.mean=0.0747253; time: 0.0 min

## [Tune-x] 13: k=13

## [Tune-y] 13: mmce.test.mean=0.0890110; time: 0.0 min
## [Tune-x] 14: k=14

## [Tune-y] 14: mmce.test.mean=0.0890110; time: 0.0 min

## [Tune-x] 15: k=15

## [Tune-y] 15: mmce.test.mean=0.1038462; time: 0.0 min

## [Tune-x] 16: k=16

## [Tune-y] 16: mmce.test.mean=0.1038462; time: 0.0 min

## [Tune-x] 17: k=17

## [Tune-y] 17: mmce.test.mean=0.0967033; time: 0.0 min

## [Tune-x] 18: k=18

## [Tune-y] 18: mmce.test.mean=0.0961538; time: 0.0 min

## [Tune-x] 19: k=19

## [Tune-y] 19: mmce.test.mean=0.1044872; time: 0.0 min

## [Tune-x] 20: k=20

## [Tune-y] 20: mmce.test.mean=0.1044872; time: 0.0 min

## [Tune] Result: k=6 : mmce.test.mean=0.0593407

## [Resample] iter 42:   0.0833333 0.9166667

## [Tune] Started tuning learner classif.knn for parameter set:

##       Type len Def                                   Constr Req Tunable Trafo
## k discrete   -   - 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,1...   -    TRUE     -
## With control class: TuneControlGrid

## Imputation value: 1

## [Tune-x] 1: k=1

## [Tune-y] 1: mmce.test.mean=0.1192308; time: 0.0 min

## [Tune-x] 2: k=2

## [Tune-y] 2: mmce.test.mean=0.1351648; time: 0.0 min

## [Tune-x] 3: k=3

## [Tune-y] 3: mmce.test.mean=0.1120879; time: 0.0 min

## [Tune-x] 4: k=4

## [Tune-y] 4: mmce.test.mean=0.1049451; time: 0.0 min

## [Tune-x] 5: k=5

## [Tune-y] 5: mmce.test.mean=0.0895604; time: 0.0 min

## [Tune-x] 6: k=6

## [Tune-y] 6: mmce.test.mean=0.0890110; time: 0.0 min

## [Tune-x] 7: k=7

## [Tune-y] 7: mmce.test.mean=0.0972527; time: 0.0 min

## [Tune-x] 8: k=8

## [Tune-y] 8: mmce.test.mean=0.0972527; time: 0.0 min
## [Tune-x] 9: k=9

## [Tune-y] 9: mmce.test.mean=0.1049451; time: 0.0 min

## [Tune-x] 10: k=10

## [Tune-y] 10: mmce.test.mean=0.0972527; time: 0.0 min

## [Tune-x] 11: k=11

## [Tune-y] 11: mmce.test.mean=0.1049451; time: 0.0 min

## [Tune-x] 12: k=12

## [Tune-y] 12: mmce.test.mean=0.0978022; time: 0.0 min

## [Tune-x] 13: k=13

## [Tune-y] 13: mmce.test.mean=0.0978022; time: 0.0 min

## [Tune-x] 14: k=14

## [Tune-y] 14: mmce.test.mean=0.0978022; time: 0.0 min

## [Tune-x] 15: k=15

## [Tune-y] 15: mmce.test.mean=0.1126374; time: 0.0 min

## [Tune-x] 16: k=16

## [Tune-y] 16: mmce.test.mean=0.1274725; time: 0.0 min

## [Tune-x] 17: k=17

## [Tune-y] 17: mmce.test.mean=0.1197802; time: 0.0 min
## [Tune-x] 18: k=18

## [Tune-y] 18: mmce.test.mean=0.1203297; time: 0.0 min

## [Tune-x] 19: k=19

## [Tune-y] 19: mmce.test.mean=0.1203297; time: 0.0 min

## [Tune-x] 20: k=20

## [Tune-y] 20: mmce.test.mean=0.1126374; time: 0.0 min

## [Tune] Result: k=6 : mmce.test.mean=0.0890110

## [Resample] iter 43:   0.0000000 1.0000000

## [Tune] Started tuning learner classif.knn for parameter set:

##       Type len Def                                  Constr Req Tunable Trafo
## k discrete   -   - 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,1...   -    TRUE     -

## With control class: TuneControlGrid

## Imputation value: 1

## [Tune-x] 1: k=1

## [Tune-y] 1: mmce.test.mean=0.0972527; time: 0.0 min

## [Tune-x] 2: k=2

## [Tune-y] 2: mmce.test.mean=0.1120879; time: 0.0 min

## [Tune-x] 3: k=3
## [Tune-y] 3: mmce.test.mean=0.0824176; time: 0.0 min

## [Tune-x] 4: k=4

## [Tune-y] 4: mmce.test.mean=0.0824176; time: 0.0 min

## [Tune-x] 5: k=5

## [Tune-y] 5: mmce.test.mean=0.0675824; time: 0.0 min

## [Tune-x] 6: k=6

## [Tune-y] 6: mmce.test.mean=0.0670330; time: 0.0 min

## [Tune-x] 7: k=7

## [Tune-y] 7: mmce.test.mean=0.0895604; time: 0.0 min

## [Tune-x] 8: k=8

## [Tune-y] 8: mmce.test.mean=0.0747253; time: 0.0 min

## [Tune-x] 9: k=9

## [Tune-y] 9: mmce.test.mean=0.0670330; time: 0.0 min

## [Tune-x] 10: k=10

## [Tune-y] 10: mmce.test.mean=0.0741758; time: 0.0 min

## [Tune-x] 11: k=11

## [Tune-y] 11: mmce.test.mean=0.0818681; time: 0.0 min

## [Tune-x] 12: k=12
## [Tune-y] 12: mmce.test.mean=0.0818681; time: 0.0 min

## [Tune-x] 13: k=13

## [Tune-y] 13: mmce.test.mean=0.0890110; time: 0.0 min

## [Tune-x] 14: k=14

## [Tune-y] 14: mmce.test.mean=0.0961538; time: 0.0 min

## [Tune-x] 15: k=15

## [Tune-y] 15: mmce.test.mean=0.0967033; time: 0.0 min

## [Tune-x] 16: k=16

## [Tune-y] 16: mmce.test.mean=0.1120879; time: 0.0 min

## [Tune-x] 17: k=17

## [Tune-y] 17: mmce.test.mean=0.1192308; time: 0.0 min

## [Tune-x] 18: k=18

## [Tune-y] 18: mmce.test.mean=0.1120879; time: 0.0 min

## [Tune-x] 19: k=19

## [Tune-y] 19: mmce.test.mean=0.1115385; time: 0.0 min

## [Tune-x] 20: k=20

## [Tune-y] 20: mmce.test.mean=0.1115385; time: 0.0 min

## [Tune] Result: k=9 : mmce.test.mean=0.0670330
## [Resample] iter 44:   0.0833333 0.9166667

## [Tune] Started tuning learner classif.knn for parameter set:

##       Type len Def                                  Constr Req Tunable Trafo
## k discrete   -   - 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,1...   -    TRUE     -

## With control class: TuneControlGrid

## Imputation value: 1

## [Tune-x] 1: k=1

## [Tune-y] 1: mmce.test.mean=0.1204029; time: 0.0 min

## [Tune-x] 2: k=2

## [Tune-y] 2: mmce.test.mean=0.0830403; time: 0.0 min

## [Tune-x] 3: k=3

## [Tune-y] 3: mmce.test.mean=0.1045421; time: 0.0 min

## [Tune-x] 4: k=4

## [Tune-y] 4: mmce.test.mean=0.1193773; time: 0.0 min

## [Tune-x] 5: k=5

## [Tune-y] 5: mmce.test.mean=0.1105861; time: 0.0 min

## [Tune-x] 6: k=6

## [Tune-y] 6: mmce.test.mean=0.1183516; time: 0.0 min
## [Tune-x] 7: k=7

## [Tune-y] 7: mmce.test.mean=0.0958242; time: 0.0 min

## [Tune-x] 8: k=8

## [Tune-y] 8: mmce.test.mean=0.0820147; time: 0.0 min

## [Tune-x] 9: k=9

## [Tune-y] 9: mmce.test.mean=0.0881319; time: 0.0 min

## [Tune-x] 10: k=10

## [Tune-y] 10: mmce.test.mean=0.0963004; time: 0.0 min

## [Tune-x] 11: k=11

## [Tune-y] 11: mmce.test.mean=0.0886813; time: 0.0 min

## [Tune-x] 12: k=12

## [Tune-y] 12: mmce.test.mean=0.1035165; time: 0.0 min

## [Tune-x] 13: k=13

## [Tune-y] 13: mmce.test.mean=0.1035165; time: 0.0 min

## [Tune-x] 14: k=14

## [Tune-y] 14: mmce.test.mean=0.1029670; time: 0.0 min

## [Tune-x] 15: k=15

## [Tune-y] 15: mmce.test.mean=0.0958242; time: 0.0 min
## [Tune-x] 16: k=16

## [Tune-y] 16: mmce.test.mean=0.1035165; time: 0.0 min

## [Tune-x] 17: k=17

## [Tune-y] 17: mmce.test.mean=0.1035165; time: 0.0 min

## [Tune-x] 18: k=18

## [Tune-y] 18: mmce.test.mean=0.1029670; time: 0.0 min

## [Tune-x] 19: k=19

## [Tune-y] 19: mmce.test.mean=0.0958242; time: 0.0 min

## [Tune-x] 20: k=20

## [Tune-y] 20: mmce.test.mean=0.1096337; time: 0.0 min

## [Tune] Result: k=8 : mmce.test.mean=0.0820147

## [Resample] iter 45:   0.0909091 0.9090909

## [Tune] Started tuning learner classif.knn for parameter set:

##       Type len Def                                  Constr Req Tunable Trafo
## k discrete   -   - 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,1...   -    TRUE     -

## With control class: TuneControlGrid

## Imputation value: 1

## [Tune-x] 1: k=1
## [Tune-y] 1: mmce.test.mean=0.1208791; time: 0.0 min

## [Tune-x] 2: k=2

## [Tune-y] 2: mmce.test.mean=0.1142857; time: 0.0 min

## [Tune-x] 3: k=3

## [Tune-y] 3: mmce.test.mean=0.0994505; time: 0.0 min

## [Tune-x] 4: k=4

## [Tune-y] 4: mmce.test.mean=0.0917582; time: 0.0 min

## [Tune-x] 5: k=5

## [Tune-y] 5: mmce.test.mean=0.1060440; time: 0.0 min

## [Tune-x] 6: k=6

## [Tune-y] 6: mmce.test.mean=0.1060440; time: 0.0 min

## [Tune-x] 7: k=7

## [Tune-y] 7: mmce.test.mean=0.0989011; time: 0.0 min

## [Tune-x] 8: k=8

## [Tune-y] 8: mmce.test.mean=0.1142857; time: 0.0 min

## [Tune-x] 9: k=9

## [Tune-y] 9: mmce.test.mean=0.0989011; time: 0.0 min

## [Tune-x] 10: k=10
## [Tune-y] 10: mmce.test.mean=0.0917582; time: 0.0 min

## [Tune-x] 11: k=11

## [Tune-y] 11: mmce.test.mean=0.0983516; time: 0.0 min

## [Tune-x] 12: k=12

## [Tune-y] 12: mmce.test.mean=0.1060440; time: 0.0 min

## [Tune-x] 13: k=13

## [Tune-y] 13: mmce.test.mean=0.1060440; time: 0.0 min

## [Tune-x] 14: k=14

## [Tune-y] 14: mmce.test.mean=0.1137363; time: 0.0 min

## [Tune-x] 15: k=15

## [Tune-y] 15: mmce.test.mean=0.1065934; time: 0.0 min

## [Tune-x] 16: k=16

## [Tune-y] 16: mmce.test.mean=0.1137363; time: 0.0 min

## [Tune-x] 17: k=17

## [Tune-y] 17: mmce.test.mean=0.1137363; time: 0.0 min

## [Tune-x] 18: k=18

## [Tune-y] 18: mmce.test.mean=0.1137363; time: 0.0 min

## [Tune-x] 19: k=19
## [Tune-y] 19: mmce.test.mean=0.1060440; time: 0.0 min

## [Tune-x] 20: k=20

## [Tune-y] 20: mmce.test.mean=0.1060440; time: 0.0 min

## [Tune] Result: k=4 : mmce.test.mean=0.0917582

## [Resample] iter 46:   0.0000000 1.0000000

## [Tune] Started tuning learner classif.knn for parameter set:

##       Type len Def                                   Constr Req Tunable Trafo
## k discrete   -   - 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,1...   -    TRUE     -

## With control class: TuneControlGrid

## Imputation value: 1

## [Tune-x] 1: k=1

## [Tune-y] 1: mmce.test.mean=0.1263736; time: 0.0 min

## [Tune-x] 2: k=2

## [Tune-y] 2: mmce.test.mean=0.1109890; time: 0.0 min

## [Tune-x] 3: k=3

## [Tune-y] 3: mmce.test.mean=0.0741758; time: 0.0 min

## [Tune-x] 4: k=4

## [Tune-y] 4: mmce.test.mean=0.0598901; time: 0.0 min
## [Tune-x] 5: k=5

## [Tune-y] 5: mmce.test.mean=0.0741758; time: 0.0 min

## [Tune-x] 6: k=6

## [Tune-y] 6: mmce.test.mean=0.0741758; time: 0.0 min

## [Tune-x] 7: k=7

## [Tune-y] 7: mmce.test.mean=0.0670330; time: 0.0 min

## [Tune-x] 8: k=8

## [Tune-y] 8: mmce.test.mean=0.0747253; time: 0.0 min

## [Tune-x] 9: k=9

## [Tune-y] 9: mmce.test.mean=0.0747253; time: 0.0 min

## [Tune-x] 10: k=10

## [Tune-y] 10: mmce.test.mean=0.0895604; time: 0.0 min

## [Tune-x] 11: k=11

## [Tune-y] 11: mmce.test.mean=0.0818681; time: 0.0 min

## [Tune-x] 12: k=12

## [Tune-y] 12: mmce.test.mean=0.0818681; time: 0.0 min

## [Tune-x] 13: k=13

## [Tune-y] 13: mmce.test.mean=0.0895604; time: 0.0 min
## [Tune-x] 14: k=14

## [Tune-y] 14: mmce.test.mean=0.0967033; time: 0.0 min

## [Tune-x] 15: k=15

## [Tune-y] 15: mmce.test.mean=0.1043956; time: 0.0 min

## [Tune-x] 16: k=16

## [Tune-y] 16: mmce.test.mean=0.1120879; time: 0.0 min

## [Tune-x] 17: k=17

## [Tune-y] 17: mmce.test.mean=0.1043956; time: 0.0 min

## [Tune-x] 18: k=18

## [Tune-y] 18: mmce.test.mean=0.1043956; time: 0.0 min

## [Tune-x] 19: k=19

## [Tune-y] 19: mmce.test.mean=0.0967033; time: 0.0 min

## [Tune-x] 20: k=20

## [Tune-y] 20: mmce.test.mean=0.0967033; time: 0.0 min

## [Tune] Result: k=4 : mmce.test.mean=0.0598901

## [Resample] iter 47:   0.1666667 0.8333333

## [Tune] Started tuning learner classif.knn for parameter set:

##       Type len Def                                  Constr Req Tunable Trafo
## k discrete   -   - 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,1...   -    TRUE     -
## With control class: TuneControlGrid

## Imputation value: 1

## [Tune-x] 1: k=1

## [Tune-y] 1: mmce.test.mean=0.1116300; time: 0.0 min

## [Tune-x] 2: k=2

## [Tune-y] 2: mmce.test.mean=0.1205128; time: 0.0 min

## [Tune-x] 3: k=3

## [Tune-y] 3: mmce.test.mean=0.0967949; time: 0.0 min

## [Tune-x] 4: k=4

## [Tune-y] 4: mmce.test.mean=0.1110806; time: 0.0 min

## [Tune-x] 5: k=5

## [Tune-y] 5: mmce.test.mean=0.0825092; time: 0.0 min

## [Tune-x] 6: k=6

## [Tune-y] 6: mmce.test.mean=0.1127289; time: 0.0 min

## [Tune-x] 7: k=7

## [Tune-y] 7: mmce.test.mean=0.0896520; time: 0.0 min

## [Tune-x] 8: k=8

## [Tune-y] 8: mmce.test.mean=0.0973443; time: 0.0 min
## [Tune-x] 9: k=9

## [Tune-y] 9: mmce.test.mean=0.1204212; time: 0.0 min

## [Tune-x] 10: k=10

## [Tune-y] 10: mmce.test.mean=0.1055861; time: 0.0 min

## [Tune-x] 11: k=11

## [Tune-y] 11: mmce.test.mean=0.1132784; time: 0.0 min

## [Tune-x] 12: k=12

## [Tune-y] 12: mmce.test.mean=0.1127289; time: 0.0 min

## [Tune-x] 13: k=13

## [Tune-y] 13: mmce.test.mean=0.1132784; time: 0.0 min

## [Tune-x] 14: k=14

## [Tune-y] 14: mmce.test.mean=0.1132784; time: 0.0 min

## [Tune-x] 15: k=15

## [Tune-y] 15: mmce.test.mean=0.1132784; time: 0.0 min

## [Tune-x] 16: k=16

## [Tune-y] 16: mmce.test.mean=0.1127289; time: 0.0 min

## [Tune-x] 17: k=17

## [Tune-y] 17: mmce.test.mean=0.1127289; time: 0.0 min
## [Tune-x] 18: k=18

## [Tune-y] 18: mmce.test.mean=0.1050366; time: 0.0 min

## [Tune-x] 19: k=19

## [Tune-y] 19: mmce.test.mean=0.1050366; time: 0.0 min

## [Tune-x] 20: k=20

## [Tune-y] 20: mmce.test.mean=0.1204212; time: 0.0 min

## [Tune] Result: k=5 : mmce.test.mean=0.0825092

## [Resample] iter 48:   0.0000000 1.0000000

## [Tune] Started tuning learner classif.knn for parameter set:

##       Type len Def                                   Constr Req Tunable Trafo
## k discrete   -   - 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,1...   -    TRUE     -

## With control class: TuneControlGrid

## Imputation value: 1

## [Tune-x] 1: k=1

## [Tune-y] 1: mmce.test.mean=0.1120879; time: 0.0 min

## [Tune-x] 2: k=2

## [Tune-y] 2: mmce.test.mean=0.1120879; time: 0.0 min

## [Tune-x] 3: k=3
## [Tune-y] 3: mmce.test.mean=0.0818681; time: 0.0 min

## [Tune-x] 4: k=4

## [Tune-y] 4: mmce.test.mean=0.0741758; time: 0.0 min

## [Tune-x] 5: k=5

## [Tune-y] 5: mmce.test.mean=0.0741758; time: 0.0 min

## [Tune-x] 6: k=6

## [Tune-y] 6: mmce.test.mean=0.0741758; time: 0.0 min

## [Tune-x] 7: k=7

## [Tune-y] 7: mmce.test.mean=0.0818681; time: 0.0 min

## [Tune-x] 8: k=8

## [Tune-y] 8: mmce.test.mean=0.0890110; time: 0.0 min

## [Tune-x] 9: k=9

## [Tune-y] 9: mmce.test.mean=0.0741758; time: 0.0 min

## [Tune-x] 10: k=10

## [Tune-y] 10: mmce.test.mean=0.0747253; time: 0.0 min

## [Tune-x] 11: k=11

## [Tune-y] 11: mmce.test.mean=0.0747253; time: 0.0 min

## [Tune-x] 12: k=12
## [Tune-y] 12: mmce.test.mean=0.0818681; time: 0.0 min

## [Tune-x] 13: k=13

## [Tune-y] 13: mmce.test.mean=0.0967033; time: 0.0 min

## [Tune-x] 14: k=14

## [Tune-y] 14: mmce.test.mean=0.1038462; time: 0.0 min

## [Tune-x] 15: k=15

## [Tune-y] 15: mmce.test.mean=0.0895604; time: 0.0 min

## [Tune-x] 16: k=16

## [Tune-y] 16: mmce.test.mean=0.0972527; time: 0.0 min

## [Tune-x] 17: k=17

## [Tune-y] 17: mmce.test.mean=0.1049451; time: 0.0 min

## [Tune-x] 18: k=18

## [Tune-y] 18: mmce.test.mean=0.1126374; time: 0.0 min

## [Tune-x] 19: k=19

## [Tune-y] 19: mmce.test.mean=0.1203297; time: 0.0 min

## [Tune-x] 20: k=20

## [Tune-y] 20: mmce.test.mean=0.1126374; time: 0.0 min

## [Tune] Result: k=4 : mmce.test.mean=0.0741758
## [Resample] iter 49:   0.0833333 0.9166667

## [Tune] Started tuning learner classif.knn for parameter set:

##       Type len Def                                  Constr Req Tunable Trafo
## k discrete   -   - 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,1...   -    TRUE     -

## With control class: TuneControlGrid

## Imputation value: 1

## [Tune-x] 1: k=1

## [Tune-y] 1: mmce.test.mean=0.1137363; time: 0.0 min

## [Tune-x] 2: k=2

## [Tune-y] 2: mmce.test.mean=0.1137363; time: 0.0 min

## [Tune-x] 3: k=3

## [Tune-y] 3: mmce.test.mean=0.0906593; time: 0.0 min

## [Tune-x] 4: k=4

## [Tune-y] 4: mmce.test.mean=0.0906593; time: 0.0 min

## [Tune-x] 5: k=5

## [Tune-y] 5: mmce.test.mean=0.0675824; time: 0.0 min

## [Tune-x] 6: k=6

## [Tune-y] 6: mmce.test.mean=0.0598901; time: 0.0 min
## [Tune-x] 7: k=7

## [Tune-y] 7: mmce.test.mean=0.0747253; time: 0.0 min

## [Tune-x] 8: k=8

## [Tune-y] 8: mmce.test.mean=0.0901099; time: 0.0 min

## [Tune-x] 9: k=9

## [Tune-y] 9: mmce.test.mean=0.0829670; time: 0.0 min

## [Tune-x] 10: k=10

## [Tune-y] 10: mmce.test.mean=0.0824176; time: 0.0 min

## [Tune-x] 11: k=11

## [Tune-y] 11: mmce.test.mean=0.0901099; time: 0.0 min

## [Tune-x] 12: k=12

## [Tune-y] 12: mmce.test.mean=0.0824176; time: 0.0 min

## [Tune-x] 13: k=13

## [Tune-y] 13: mmce.test.mean=0.0824176; time: 0.0 min

## [Tune-x] 14: k=14

## [Tune-y] 14: mmce.test.mean=0.0901099; time: 0.0 min

## [Tune-x] 15: k=15

## [Tune-y] 15: mmce.test.mean=0.0901099; time: 0.0 min
## [Tune-x] 16: k=16

## [Tune-y] 16: mmce.test.mean=0.0824176; time: 0.0 min

## [Tune-x] 17: k=17

## [Tune-y] 17: mmce.test.mean=0.1126374; time: 0.0 min

## [Tune-x] 18: k=18

## [Tune-y] 18: mmce.test.mean=0.1126374; time: 0.0 min

## [Tune-x] 19: k=19

## [Tune-y] 19: mmce.test.mean=0.1049451; time: 0.0 min

## [Tune-x] 20: k=20

## [Tune-y] 20: mmce.test.mean=0.1126374; time: 0.0 min

## [Tune] Result: k=6 : mmce.test.mean=0.0598901

## [Resample] iter 50:   0.0833333 0.9166667

## [Tune] Started tuning learner classif.knn for parameter set:

##       Type len Def                                  Constr Req Tunable Trafo
## k discrete   -   - 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,1...   -    TRUE     -

## With control class: TuneControlGrid

## Imputation value: 1

## [Tune-x] 1: k=1
## [Tune-y] 1: mmce.test.mean=0.0813187; time: 0.0 min

## [Tune-x] 2: k=2

## [Tune-y] 2: mmce.test.mean=0.1115385; time: 0.0 min

## [Tune-x] 3: k=3

## [Tune-y] 3: mmce.test.mean=0.0884615; time: 0.0 min

## [Tune-x] 4: k=4

## [Tune-y] 4: mmce.test.mean=0.0967033; time: 0.0 min

## [Tune-x] 5: k=5

## [Tune-y] 5: mmce.test.mean=0.0587912; time: 0.0 min

## [Tune-x] 6: k=6

## [Tune-y] 6: mmce.test.mean=0.0664835; time: 0.0 min

## [Tune-x] 7: k=7

## [Tune-y] 7: mmce.test.mean=0.0741758; time: 0.0 min

## [Tune-x] 8: k=8

## [Tune-y] 8: mmce.test.mean=0.0741758; time: 0.0 min

## [Tune-x] 9: k=9

## [Tune-y] 9: mmce.test.mean=0.0736264; time: 0.0 min

## [Tune-x] 10: k=10
## [Tune-y] 10: mmce.test.mean=0.0884615; time: 0.0 min

## [Tune-x] 11: k=11

## [Tune-y] 11: mmce.test.mean=0.0890110; time: 0.0 min

## [Tune-x] 12: k=12

## [Tune-y] 12: mmce.test.mean=0.0967033; time: 0.0 min

## [Tune-x] 13: k=13

## [Tune-y] 13: mmce.test.mean=0.0967033; time: 0.0 min

## [Tune-x] 14: k=14

## [Tune-y] 14: mmce.test.mean=0.0972527; time: 0.0 min

## [Tune-x] 15: k=15

## [Tune-y] 15: mmce.test.mean=0.0895604; time: 0.0 min

## [Tune-x] 16: k=16

## [Tune-y] 16: mmce.test.mean=0.0967033; time: 0.0 min

## [Tune-x] 17: k=17

## [Tune-y] 17: mmce.test.mean=0.1049451; time: 0.0 min

## [Tune-x] 18: k=18

## [Tune-y] 18: mmce.test.mean=0.1126374; time: 0.0 min

## [Tune-x] 19: k=19
## [Tune-y] 19: mmce.test.mean=0.1126374; time: 0.0 min

## [Tune-x] 20: k=20

## [Tune-y] 20: mmce.test.mean=0.1126374; time: 0.0 min

## [Tune] Result: k=5 : mmce.test.mean=0.0587912

## [Resample] iter 51:   0.0000000 1.0000000

## [Tune] Started tuning learner classif.knn for parameter set:

##       Type len Def                                   Constr Req Tunable Trafo
## k discrete   -   - 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,1...   -    TRUE     -

## With control class: TuneControlGrid

## Imputation value: 1

## [Tune-x] 1: k=1

## [Tune-y] 1: mmce.test.mean=0.1043956; time: 0.0 min

## [Tune-x] 2: k=2

## [Tune-y] 2: mmce.test.mean=0.1353480; time: 0.0 min

## [Tune-x] 3: k=3

## [Tune-y] 3: mmce.test.mean=0.1050366; time: 0.0 min

## [Tune-x] 4: k=4

## [Tune-y] 4: mmce.test.mean=0.0807692; time: 0.0 min
## [Tune-x] 5: k=5

## [Tune-y] 5: mmce.test.mean=0.0593407; time: 0.0 min

## [Tune-x] 6: k=6

## [Tune-y] 6: mmce.test.mean=0.0890110; time: 0.0 min

## [Tune-x] 7: k=7

## [Tune-y] 7: mmce.test.mean=0.0664835; time: 0.0 min

## [Tune-x] 8: k=8

## [Tune-y] 8: mmce.test.mean=0.0807692; time: 0.0 min

## [Tune-x] 9: k=9

## [Tune-y] 9: mmce.test.mean=0.0741758; time: 0.0 min

## [Tune-x] 10: k=10

## [Tune-y] 10: mmce.test.mean=0.0664835; time: 0.0 min

## [Tune-x] 11: k=11

## [Tune-y] 11: mmce.test.mean=0.0741758; time: 0.0 min

## [Tune-x] 12: k=12

## [Tune-y] 12: mmce.test.mean=0.0747253; time: 0.0 min

## [Tune-x] 13: k=13

## [Tune-y] 13: mmce.test.mean=0.0967033; time: 0.0 min
## [Tune-x] 14: k=14

## [Tune-y] 14: mmce.test.mean=0.1038462; time: 0.0 min

## [Tune-x] 15: k=15

## [Tune-y] 15: mmce.test.mean=0.0890110; time: 0.0 min

## [Tune-x] 16: k=16

## [Tune-y] 16: mmce.test.mean=0.0967033; time: 0.0 min

## [Tune-x] 17: k=17

## [Tune-y] 17: mmce.test.mean=0.0967033; time: 0.0 min

## [Tune-x] 18: k=18

## [Tune-y] 18: mmce.test.mean=0.0972527; time: 0.0 min

## [Tune-x] 19: k=19

## [Tune-y] 19: mmce.test.mean=0.1043956; time: 0.0 min

## [Tune-x] 20: k=20

## [Tune-y] 20: mmce.test.mean=0.1043956; time: 0.0 min

## [Tune] Result: k=5 : mmce.test.mean=0.0593407

## [Resample] iter 52:   0.0769231 0.9230769

## [Tune] Started tuning learner classif.knn for parameter set:

##       Type len Def                                  Constr Req Tunable Trafo
## k discrete   -   - 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,1...   -    TRUE     -
## With control class: TuneControlGrid

## Imputation value: 1

## [Tune-x] 1: k=1

## [Tune-y] 1: mmce.test.mean=0.1043956; time: 0.0 min

## [Tune-x] 2: k=2

## [Tune-y] 2: mmce.test.mean=0.1131868; time: 0.0 min

## [Tune-x] 3: k=3

## [Tune-y] 3: mmce.test.mean=0.0747253; time: 0.0 min

## [Tune-x] 4: k=4

## [Tune-y] 4: mmce.test.mean=0.0741758; time: 0.0 min

## [Tune-x] 5: k=5

## [Tune-y] 5: mmce.test.mean=0.0604396; time: 0.0 min

## [Tune-x] 6: k=6

## [Tune-y] 6: mmce.test.mean=0.0675824; time: 0.0 min

## [Tune-x] 7: k=7

## [Tune-y] 7: mmce.test.mean=0.0829670; time: 0.0 min

## [Tune-x] 8: k=8

## [Tune-y] 8: mmce.test.mean=0.0829670; time: 0.0 min
## [Tune-x] 9: k=9

## [Tune-y] 9: mmce.test.mean=0.0829670; time: 0.0 min

## [Tune-x] 10: k=10

## [Tune-y] 10: mmce.test.mean=0.0829670; time: 0.0 min

## [Tune-x] 11: k=11

## [Tune-y] 11: mmce.test.mean=0.0829670; time: 0.0 min

## [Tune-x] 12: k=12

## [Tune-y] 12: mmce.test.mean=0.0972527; time: 0.0 min

## [Tune-x] 13: k=13

## [Tune-y] 13: mmce.test.mean=0.0972527; time: 0.0 min

## [Tune-x] 14: k=14

## [Tune-y] 14: mmce.test.mean=0.1043956; time: 0.0 min

## [Tune-x] 15: k=15

## [Tune-y] 15: mmce.test.mean=0.0967033; time: 0.0 min

## [Tune-x] 16: k=16

## [Tune-y] 16: mmce.test.mean=0.1043956; time: 0.0 min

## [Tune-x] 17: k=17

## [Tune-y] 17: mmce.test.mean=0.1120879; time: 0.0 min
## [Tune-x] 18: k=18

## [Tune-y] 18: mmce.test.mean=0.1120879; time: 0.0 min

## [Tune-x] 19: k=19

## [Tune-y] 19: mmce.test.mean=0.1043956; time: 0.0 min

## [Tune-x] 20: k=20

## [Tune-y] 20: mmce.test.mean=0.1043956; time: 0.0 min

## [Tune] Result: k=5 : mmce.test.mean=0.0604396

## [Resample] iter 53:   0.1818182 0.8181818

## [Tune] Started tuning learner classif.knn for parameter set:

##       Type len Def                                  Constr Req Tunable Trafo
## k discrete   -   - 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,1...   -    TRUE     -

## With control class: TuneControlGrid

## Imputation value: 1

## [Tune-x] 1: k=1

## [Tune-y] 1: mmce.test.mean=0.1131868; time: 0.0 min

## [Tune-x] 2: k=2

## [Tune-y] 2: mmce.test.mean=0.0906593; time: 0.0 min

## [Tune-x] 3: k=3
## [Tune-y] 3: mmce.test.mean=0.0906593; time: 0.0 min

## [Tune-x] 4: k=4

## [Tune-y] 4: mmce.test.mean=0.0901099; time: 0.0 min

## [Tune-x] 5: k=5

## [Tune-y] 5: mmce.test.mean=0.0901099; time: 0.0 min

## [Tune-x] 6: k=6

## [Tune-y] 6: mmce.test.mean=0.0972527; time: 0.0 min

## [Tune-x] 7: k=7

## [Tune-y] 7: mmce.test.mean=0.1043956; time: 0.0 min

## [Tune-x] 8: k=8

## [Tune-y] 8: mmce.test.mean=0.0967033; time: 0.0 min

## [Tune-x] 9: k=9

## [Tune-y] 9: mmce.test.mean=0.1043956; time: 0.0 min

## [Tune-x] 10: k=10

## [Tune-y] 10: mmce.test.mean=0.0895604; time: 0.0 min

## [Tune-x] 11: k=11

## [Tune-y] 11: mmce.test.mean=0.1120879; time: 0.0 min

## [Tune-x] 12: k=12
## [Tune-y] 12: mmce.test.mean=0.0972527; time: 0.0 min

## [Tune-x] 13: k=13

## [Tune-y] 13: mmce.test.mean=0.0978022; time: 0.0 min

## [Tune-x] 14: k=14

## [Tune-y] 14: mmce.test.mean=0.1049451; time: 0.0 min

## [Tune-x] 15: k=15

## [Tune-y] 15: mmce.test.mean=0.1126374; time: 0.0 min

## [Tune-x] 16: k=16

## [Tune-y] 16: mmce.test.mean=0.1054945; time: 0.0 min

## [Tune-x] 17: k=17

## [Tune-y] 17: mmce.test.mean=0.1131868; time: 0.0 min

## [Tune-x] 18: k=18

## [Tune-y] 18: mmce.test.mean=0.1131868; time: 0.0 min

## [Tune-x] 19: k=19

## [Tune-y] 19: mmce.test.mean=0.1131868; time: 0.0 min

## [Tune-x] 20: k=20

## [Tune-y] 20: mmce.test.mean=0.1203297; time: 0.0 min

## [Tune] Result: k=10 : mmce.test.mean=0.0895604
## [Resample] iter 54:   0.0000000 1.0000000

## [Tune] Started tuning learner classif.knn for parameter set:

##       Type len Def                                  Constr Req Tunable Trafo
## k discrete   -   - 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,1...   -    TRUE     -

## With control class: TuneControlGrid

## Imputation value: 1

## [Tune-x] 1: k=1

## [Tune-y] 1: mmce.test.mean=0.1126374; time: 0.0 min

## [Tune-x] 2: k=2

## [Tune-y] 2: mmce.test.mean=0.1126374; time: 0.0 min

## [Tune-x] 3: k=3

## [Tune-y] 3: mmce.test.mean=0.0758242; time: 0.0 min

## [Tune-x] 4: k=4

## [Tune-y] 4: mmce.test.mean=0.1043956; time: 0.0 min

## [Tune-x] 5: k=5

## [Tune-y] 5: mmce.test.mean=0.0901099; time: 0.0 min

## [Tune-x] 6: k=6

## [Tune-y] 6: mmce.test.mean=0.0675824; time: 0.0 min
## [Tune-x] 7: k=7

## [Tune-y] 7: mmce.test.mean=0.0901099; time: 0.0 min

## [Tune-x] 8: k=8

## [Tune-y] 8: mmce.test.mean=0.0978022; time: 0.0 min

## [Tune-x] 9: k=9

## [Tune-y] 9: mmce.test.mean=0.1049451; time: 0.0 min

## [Tune-x] 10: k=10

## [Tune-y] 10: mmce.test.mean=0.1049451; time: 0.0 min

## [Tune-x] 11: k=11

## [Tune-y] 11: mmce.test.mean=0.0972527; time: 0.0 min

## [Tune-x] 12: k=12

## [Tune-y] 12: mmce.test.mean=0.1049451; time: 0.0 min

## [Tune-x] 13: k=13

## [Tune-y] 13: mmce.test.mean=0.1049451; time: 0.0 min

## [Tune-x] 14: k=14

## [Tune-y] 14: mmce.test.mean=0.0901099; time: 0.0 min

## [Tune-x] 15: k=15

## [Tune-y] 15: mmce.test.mean=0.0978022; time: 0.0 min
## [Tune-x] 16: k=16

## [Tune-y] 16: mmce.test.mean=0.0978022; time: 0.0 min

## [Tune-x] 17: k=17

## [Tune-y] 17: mmce.test.mean=0.0978022; time: 0.0 min

## [Tune-x] 18: k=18

## [Tune-y] 18: mmce.test.mean=0.0978022; time: 0.0 min

## [Tune-x] 19: k=19

## [Tune-y] 19: mmce.test.mean=0.0978022; time: 0.0 min

## [Tune-x] 20: k=20

## [Tune-y] 20: mmce.test.mean=0.0978022; time: 0.0 min

## [Tune] Result: k=6 : mmce.test.mean=0.0675824

## [Resample] iter 55:   0.1666667 0.8333333

## [Tune] Started tuning learner classif.knn for parameter set:

##       Type len Def                                   Constr Req Tunable Trafo
## k discrete   -   - 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,1...   -    TRUE     -

## With control class: TuneControlGrid

## Imputation value: 1

## [Tune-x] 1: k=1
## [Tune-y] 1: mmce.test.mean=0.0884615; time: 0.0 min

## [Tune-x] 2: k=2

## [Tune-y] 2: mmce.test.mean=0.1126374; time: 0.0 min

## [Tune-x] 3: k=3

## [Tune-y] 3: mmce.test.mean=0.0741758; time: 0.0 min

## [Tune-x] 4: k=4

## [Tune-y] 4: mmce.test.mean=0.0807692; time: 0.0 min

## [Tune-x] 5: k=5

## [Tune-y] 5: mmce.test.mean=0.0736264; time: 0.0 min

## [Tune-x] 6: k=6

## [Tune-y] 6: mmce.test.mean=0.0807692; time: 0.0 min

## [Tune-x] 7: k=7

## [Tune-y] 7: mmce.test.mean=0.0736264; time: 0.0 min

## [Tune-x] 8: k=8

## [Tune-y] 8: mmce.test.mean=0.0807692; time: 0.0 min

## [Tune-x] 9: k=9

## [Tune-y] 9: mmce.test.mean=0.0659341; time: 0.0 min

## [Tune-x] 10: k=10
## [Tune-y] 10: mmce.test.mean=0.0730769; time: 0.0 min

## [Tune-x] 11: k=11

## [Tune-y] 11: mmce.test.mean=0.0659341; time: 0.0 min

## [Tune-x] 12: k=12

## [Tune-y] 12: mmce.test.mean=0.0884615; time: 0.0 min

## [Tune-x] 13: k=13

## [Tune-y] 13: mmce.test.mean=0.0956044; time: 0.0 min

## [Tune-x] 14: k=14

## [Tune-y] 14: mmce.test.mean=0.0879121; time: 0.0 min

## [Tune-x] 15: k=15

## [Tune-y] 15: mmce.test.mean=0.0956044; time: 0.0 min

## [Tune-x] 16: k=16

## [Tune-y] 16: mmce.test.mean=0.0884615; time: 0.0 min

## [Tune-x] 17: k=17

## [Tune-y] 17: mmce.test.mean=0.0884615; time: 0.0 min

## [Tune-x] 18: k=18

## [Tune-y] 18: mmce.test.mean=0.0890110; time: 0.0 min

## [Tune-x] 19: k=19
## [Tune-y] 19: mmce.test.mean=0.0890110; time: 0.0 min

## [Tune-x] 20: k=20

## [Tune-y] 20: mmce.test.mean=0.0961538; time: 0.0 min

## [Tune] Result: k=11 : mmce.test.mean=0.0659341

## [Resample] iter 56:   0.2727273 0.7272727

## [Tune] Started tuning learner classif.knn for parameter set:

##       Type len Def                                   Constr Req Tunable Trafo
## k discrete   -   - 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,1...   -    TRUE     -

## With control class: TuneControlGrid

## Imputation value: 1

## [Tune-x] 1: k=1

## [Tune-y] 1: mmce.test.mean=0.1221612; time: 0.0 min

## [Tune-x] 2: k=2

## [Tune-y] 2: mmce.test.mean=0.0687729; time: 0.0 min

## [Tune-x] 3: k=3

## [Tune-y] 3: mmce.test.mean=0.1144689; time: 0.0 min

## [Tune-x] 4: k=4

## [Tune-y] 4: mmce.test.mean=0.1067766; time: 0.0 min
## [Tune-x] 5: k=5

## [Tune-y] 5: mmce.test.mean=0.0847985; time: 0.0 min

## [Tune-x] 6: k=6

## [Tune-y] 6: mmce.test.mean=0.0836081; time: 0.0 min

## [Tune-x] 7: k=7

## [Tune-y] 7: mmce.test.mean=0.0847070; time: 0.0 min

## [Tune-x] 8: k=8

## [Tune-y] 8: mmce.test.mean=0.0989927; time: 0.0 min

## [Tune-x] 9: k=9

## [Tune-y] 9: mmce.test.mean=0.0841575; time: 0.0 min

## [Tune-x] 10: k=10

## [Tune-y] 10: mmce.test.mean=0.0919414; time: 0.0 min

## [Tune-x] 11: k=11

## [Tune-y] 11: mmce.test.mean=0.0913004; time: 0.0 min

## [Tune-x] 12: k=12

## [Tune-y] 12: mmce.test.mean=0.1073260; time: 0.0 min

## [Tune-x] 13: k=13

## [Tune-y] 13: mmce.test.mean=0.0996337; time: 0.0 min
## [Tune-x] 14: k=14

## [Tune-y] 14: mmce.test.mean=0.0996337; time: 0.0 min

## [Tune-x] 15: k=15

## [Tune-y] 15: mmce.test.mean=0.0996337; time: 0.0 min

## [Tune-x] 16: k=16

## [Tune-y] 16: mmce.test.mean=0.1073260; time: 0.0 min

## [Tune-x] 17: k=17

## [Tune-y] 17: mmce.test.mean=0.1216117; time: 0.0 min

## [Tune-x] 18: k=18

## [Tune-y] 18: mmce.test.mean=0.1067766; time: 0.0 min

## [Tune-x] 19: k=19

## [Tune-y] 19: mmce.test.mean=0.1139194; time: 0.0 min

## [Tune-x] 20: k=20

## [Tune-y] 20: mmce.test.mean=0.1139194; time: 0.0 min

## [Tune] Result: k=2 : mmce.test.mean=0.0687729

## [Resample] iter 57:   0.1538462 0.8461538

## [Tune] Started tuning learner classif.knn for parameter set:

##       Type len Def                                  Constr Req Tunable Trafo
## k discrete   -   - 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,1...   -    TRUE     -
## With control class: TuneControlGrid

## Imputation value: 1

## [Tune-x] 1: k=1

## [Tune-y] 1: mmce.test.mean=0.1208791; time: 0.0 min

## [Tune-x] 2: k=2

## [Tune-y] 2: mmce.test.mean=0.1208791; time: 0.0 min

## [Tune-x] 3: k=3

## [Tune-y] 3: mmce.test.mean=0.0978022; time: 0.0 min

## [Tune-x] 4: k=4

## [Tune-y] 4: mmce.test.mean=0.1137363; time: 0.0 min

## [Tune-x] 5: k=5

## [Tune-y] 5: mmce.test.mean=0.1060440; time: 0.0 min

## [Tune-x] 6: k=6

## [Tune-y] 6: mmce.test.mean=0.0752747; time: 0.0 min

## [Tune-x] 7: k=7

## [Tune-y] 7: mmce.test.mean=0.0829670; time: 0.0 min

## [Tune-x] 8: k=8

## [Tune-y] 8: mmce.test.mean=0.0983516; time: 0.0 min
## [Tune-x] 9: k=9

## [Tune-y] 9: mmce.test.mean=0.0906593; time: 0.0 min

## [Tune-x] 10: k=10

## [Tune-y] 10: mmce.test.mean=0.0829670; time: 0.0 min

## [Tune-x] 11: k=11

## [Tune-y] 11: mmce.test.mean=0.0829670; time: 0.0 min

## [Tune-x] 12: k=12

## [Tune-y] 12: mmce.test.mean=0.1049451; time: 0.0 min

## [Tune-x] 13: k=13

## [Tune-y] 13: mmce.test.mean=0.0972527; time: 0.0 min

## [Tune-x] 14: k=14

## [Tune-y] 14: mmce.test.mean=0.1049451; time: 0.0 min

## [Tune-x] 15: k=15

## [Tune-y] 15: mmce.test.mean=0.0978022; time: 0.0 min

## [Tune-x] 16: k=16

## [Tune-y] 16: mmce.test.mean=0.0978022; time: 0.0 min

## [Tune-x] 17: k=17

## [Tune-y] 17: mmce.test.mean=0.0978022; time: 0.0 min
## [Tune-x] 18: k=18

## [Tune-y] 18: mmce.test.mean=0.0978022; time: 0.0 min

## [Tune-x] 19: k=19

## [Tune-y] 19: mmce.test.mean=0.1049451; time: 0.0 min

## [Tune-x] 20: k=20

## [Tune-y] 20: mmce.test.mean=0.1049451; time: 0.0 min

## [Tune] Result: k=6 : mmce.test.mean=0.0752747

## [Resample] iter 58:   0.0000000 1.0000000

## [Tune] Started tuning learner classif.knn for parameter set:

##       Type len Def                                   Constr Req Tunable Trafo
## k discrete   -   - 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,1...   -    TRUE     -

## With control class: TuneControlGrid

## Imputation value: 1

## [Tune-x] 1: k=1

## [Tune-y] 1: mmce.test.mean=0.1065934; time: 0.0 min

## [Tune-x] 2: k=2

## [Tune-y] 2: mmce.test.mean=0.1054945; time: 0.0 min

## [Tune-x] 3: k=3
## [Tune-y] 3: mmce.test.mean=0.0912088; time: 0.0 min

## [Tune-x] 4: k=4

## [Tune-y] 4: mmce.test.mean=0.0901099; time: 0.0 min

## [Tune-x] 5: k=5

## [Tune-y] 5: mmce.test.mean=0.0829670; time: 0.0 min

## [Tune-x] 6: k=6

## [Tune-y] 6: mmce.test.mean=0.0835165; time: 0.0 min

## [Tune-x] 7: k=7

## [Tune-y] 7: mmce.test.mean=0.0763736; time: 0.0 min

## [Tune-x] 8: k=8

## [Tune-y] 8: mmce.test.mean=0.0917582; time: 0.0 min

## [Tune-x] 9: k=9

## [Tune-y] 9: mmce.test.mean=0.0994505; time: 0.0 min

## [Tune-x] 10: k=10

## [Tune-y] 10: mmce.test.mean=0.0994505; time: 0.0 min

## [Tune-x] 11: k=11

## [Tune-y] 11: mmce.test.mean=0.0917582; time: 0.0 min

## [Tune-x] 12: k=12
## [Tune-y] 12: mmce.test.mean=0.0994505; time: 0.0 min

## [Tune-x] 13: k=13

## [Tune-y] 13: mmce.test.mean=0.0994505; time: 0.0 min

## [Tune-x] 14: k=14

## [Tune-y] 14: mmce.test.mean=0.1071429; time: 0.0 min

## [Tune-x] 15: k=15

## [Tune-y] 15: mmce.test.mean=0.1148352; time: 0.0 min

## [Tune-x] 16: k=16

## [Tune-y] 16: mmce.test.mean=0.1225275; time: 0.0 min

## [Tune-x] 17: k=17

## [Tune-y] 17: mmce.test.mean=0.1225275; time: 0.0 min

## [Tune-x] 18: k=18

## [Tune-y] 18: mmce.test.mean=0.1225275; time: 0.0 min

## [Tune-x] 19: k=19

## [Tune-y] 19: mmce.test.mean=0.1148352; time: 0.0 min

## [Tune-x] 20: k=20

## [Tune-y] 20: mmce.test.mean=0.1225275; time: 0.0 min

## [Tune] Result: k=7 : mmce.test.mean=0.0763736
## [Resample] iter 59:   0.0000000 1.0000000

## [Tune] Started tuning learner classif.knn for parameter set:

##       Type len Def                                   Constr Req Tunable Trafo
## k discrete   -   - 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,1...   -    TRUE     -

## With control class: TuneControlGrid

## Imputation value: 1

## [Tune-x] 1: k=1

## [Tune-y] 1: mmce.test.mean=0.0890110; time: 0.0 min

## [Tune-x] 2: k=2

## [Tune-y] 2: mmce.test.mean=0.0675824; time: 0.0 min

## [Tune-x] 3: k=3

## [Tune-y] 3: mmce.test.mean=0.0824176; time: 0.0 min

## [Tune-x] 4: k=4

## [Tune-y] 4: mmce.test.mean=0.0747253; time: 0.0 min

## [Tune-x] 5: k=5

## [Tune-y] 5: mmce.test.mean=0.0747253; time: 0.0 min

## [Tune-x] 6: k=6

## [Tune-y] 6: mmce.test.mean=0.0670330; time: 0.0 min
## [Tune-x] 7: k=7

## [Tune-y] 7: mmce.test.mean=0.0593407; time: 0.0 min

## [Tune-x] 8: k=8

## [Tune-y] 8: mmce.test.mean=0.0741758; time: 0.0 min

## [Tune-x] 9: k=9

## [Tune-y] 9: mmce.test.mean=0.0747253; time: 0.0 min

## [Tune-x] 10: k=10

## [Tune-y] 10: mmce.test.mean=0.0967033; time: 0.0 min

## [Tune-x] 11: k=11

## [Tune-y] 11: mmce.test.mean=0.0741758; time: 0.0 min

## [Tune-x] 12: k=12

## [Tune-y] 12: mmce.test.mean=0.0818681; time: 0.0 min

## [Tune-x] 13: k=13

## [Tune-y] 13: mmce.test.mean=0.0813187; time: 0.0 min

## [Tune-x] 14: k=14

## [Tune-y] 14: mmce.test.mean=0.0890110; time: 0.0 min

## [Tune-x] 15: k=15

## [Tune-y] 15: mmce.test.mean=0.0895604; time: 0.0 min
## [Tune-x] 16: k=16

## [Tune-y] 16: mmce.test.mean=0.1043956; time: 0.0 min

## [Tune-x] 17: k=17

## [Tune-y] 17: mmce.test.mean=0.1120879; time: 0.0 min

## [Tune-x] 18: k=18

## [Tune-y] 18: mmce.test.mean=0.0895604; time: 0.0 min

## [Tune-x] 19: k=19

## [Tune-y] 19: mmce.test.mean=0.1049451; time: 0.0 min

## [Tune-x] 20: k=20

## [Tune-y] 20: mmce.test.mean=0.1197802; time: 0.0 min

## [Tune] Result: k=7 : mmce.test.mean=0.0593407

## [Resample] iter 60:   0.0833333 0.9166667

##

## Aggregated Result: mmce.test.mean=0.0898407,acc.test.mean=0.9101593

##
```

```
resa.tune.embeded.cv
```

```
## Resample Result
## Task: data
## Learner: classif.knn.tuned
## Aggr perf: mmce.test.mean=0.0898407,acc.test.mean=0.9101593
## Runtime: 49.4146
```

不难发现，k=6 或 7 时效果最好，与之前得出的结论一致

4.8 未知数据的预测

```
newdataPred <- predict(tunedModel, newdata=data[1:100,])
newdataPred$data
```

```
##         truth response
## 1      Normal   Normal
## 2      Normal   Normal
## 3      Normal   Normal
## 4      Normal   Normal
## 5      Normal   Normal
## 6      Normal   Normal
## 7      Normal   Normal
## 8      Normal   Normal
## 9      Normal   Normal
## 10     Normal   Normal
## 11     Normal   Normal
## 12     Normal   Normal
## 13     Normal   Normal
## 14     Normal   Normal
## 15     Normal   Normal
## 16     Normal   Normal
## 17     Normal   Normal
## 18     Normal   Normal
## 19     Normal   Normal
## 20     Normal   Normal
## 21     Normal   Normal
## 22     Normal   Normal
## 23     Normal   Normal
## 24     Normal   Normal
## 25     Normal   Normal
## 26     Normal   Normal
## 27     Normal   Normal
## 28    Normal   Normal
## 29    Normal   Normal
## 30    Normal   Normal
## 31    Normal   Normal
## 32    Normal   Normal
## 33    Normal   Normal
## 34    Normal   Normal
## 35    Normal   Normal
## 36    Normal   Normal
## 37    Normal   Normal
## 38    Normal   Normal
## 39    Normal   Normal
## 40    Normal   Normal
## 41    Normal   Normal
## 42    Normal   Normal
## 43    Normal   Normal
## 44    Normal   Normal
## 45    Normal   Normal
## 46    Normal   Normal
## 47    Normal   Normal
## 48    Normal   Normal
## 49    Normal   Normal
## 50    Normal   Normal
## 51    Normal   Normal
## 52    Normal   Normal
## 53    Normal   Normal
## 54    Normal   Normal
## 55    Normal   Normal
## 56    Normal   Normal
## 57    Normal   Normal
## 58    Normal   Normal
## 59  Chemical Chemical
## 60    Normal   Normal
## 61    Normal   Normal
## 62  Chemical Chemical
## 63  Chemical Chemical
## 64    Normal   Normal
## 65  Chemical Chemical
## 66  Chemical Chemical
## 67    Normal   Normal
## 68    Normal   Normal
## 69    Normal Chemical
## 70    Normal   Normal
## 71  Chemical Chemical
## 72    Normal   Normal
## 73    Normal   Normal
## 74    Normal   Normal
## 75    Normal   Normal
## 76    Normal   Normal
## 77  Chemical Chemical
## 78    Normal   Normal
## 79    Normal   Normal
## 80    Normal   Normal
## 81    Normal   Normal
## 82    Normal   Normal
## 83  Chemical Chemical
## 84    Normal   Normal
## 85  Chemical Chemical
## 86  Chemical Chemical
## 87  Chemical Chemical
## 88  Chemical Chemical
## 89  Chemical Chemical
## 90  Chemical Chemical
## 91  Chemical Chemical
## 92  Chemical Chemical
## 93  Chemical Chemical
## 94  Chemical Chemical
## 95  Chemical Chemical
## 96  Chemical Chemical
## 97  Chemical Chemical
## 98  Chemical Chemical
## 99  Chemical Chemical
## 100 Chemical Chemical
```

```
calculateConfusionMatrix(newdataPred)
```

```
##           predicted
## true       Chemical Normal Overt -err.-
##   Chemical       24      0     0      0
##   Normal          1     75     0      1
##   Overt           0      0     0      0
##   -err.-          1      0     0      1
```