knn_hw.R

alexaubrey

Sun May 20 13:03:01 2018

```
library(class)
library(caret)
```

```
## Warning: package 'caret' was built under R version 3.4.4
```

```
## Loading required package: lattice
```

```
## Loading required package: ggplot2
```

```
## Warning in as.POSIXlt.POSIXct(Sys.time()): unknown timezone 'zone/tz/2018c.
## 1.0/zoneinfo/America/Chicago'
```

```
data <- read.csv("https://archive.ics.uci.edu/ml/machine-learning-databases/00267/data_b
anknote_authentication.txt",
                 head=FALSE,
                 col.names=c("varOWTi","skwOWTi","curtOWTI","entropy","class"))

# This is a class value not a continuous attribute
data$class <- as.factor(data$class)

preProcess(data, method=c("center", "scale"))
```

```
## Created from 1372 samples and 5 variables
##
## Pre-processing:
##   - centered (4)
##   - ignored (1)
##   - scaled (4)
```

```
n <- dim(data)[1]

t1 <- sample(1:n, n*.8)
t2 <- setdiff(1:n, t1)

c1 <- data[t1,]$class

train <- subset(data[t1,], select =- class)
test <- subset(data[t2,], select =- class)

ktune <- train(train, c1, method="knn",
               tuneGrid=data.frame(.k = 1:50),
               trControl = trainControl(method = "cv"))

ktune #k = 14 suggestion
```

```
## k-Nearest Neighbors
##
## 1097 samples
##    4 predictor
##    2 classes: '0', '1'
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 988, 987, 987, 988, 987, 987, ...
## Resampling results across tuning parameters:
##
##   k   Accuracy   Kappa
##    1  1.0000000  1.0000000
##    2  1.0000000  1.0000000
##    3  1.0000000  1.0000000
##    4  1.0000000  1.0000000
##    5  1.0000000  1.0000000
##    6  1.0000000  1.0000000
##    7  1.0000000  1.0000000
##    8  1.0000000  1.0000000
##    9  1.0000000  1.0000000
##   10  1.0000000  1.0000000
##   11  1.0000000  1.0000000
##   12  1.0000000  1.0000000
##   13  1.0000000  1.0000000
##   14  0.9990909  0.9981562
##   15  0.9972727  0.9944773
##   16  0.9936364  0.9871194
##   17  0.9936364  0.9871194
##   18  0.9936364  0.9871194
##   19  0.9927189  0.9852539
##   20  0.9927189  0.9852539
##   21  0.9918098  0.9834102
##   22  0.9918098  0.9834102
##   23  0.9918098  0.9834102
##   24  0.9918098  0.9834102
##   25  0.9918098  0.9834102
##   26  0.9918098  0.9834102
##   27  0.9909008  0.9815750
##   28  0.9909008  0.9815750
##   29  0.9918098  0.9834102
##   30  0.9909008  0.9815664
##   31  0.9909008  0.9815750
##   32  0.9890826  0.9778961
##   33  0.9899917  0.9797312
##   34  0.9899917  0.9797312
##   35  0.9899917  0.9797312
##   36  0.9890826  0.9779047
##   37  0.9899917  0.9797312
##   38  0.9890826  0.9778875
##   39  0.9890826  0.9778875
##   40  0.9899917  0.9797312
##   41  0.9890826  0.9778875
```

```
##   42  0.9890826  0.9779047
##   43  0.9881735  0.9760609
##   44  0.9872644  0.9742343
##   45  0.9881735  0.9760609
##   46  0.9872644  0.9742343
##   47  0.9881735  0.9760609
##   48  0.9836280  0.9669365
##   49  0.9836280  0.9669365
##   50  0.9845371  0.9687630
##
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was k = 13.
```

```
pred <- predict(ktune, test)
confusionMatrix(pred, data[t2,]$class)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 142   0
##          1   0 133
##
##                Accuracy : 1
##                  95% CI : (0.9867, 1)
##     No Information Rate : 0.5164
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 1
##  Mcnemar's Test P-Value : NA
##
##             Sensitivity : 1.0000
##             Specificity : 1.0000
##          Pos Pred Value : 1.0000
##          Neg Pred Value : 1.0000
##              Prevalence : 0.5164
##          Detection Rate : 0.5164
##    Detection Prevalence : 0.5164
##       Balanced Accuracy : 1.0000
##
##        'Positive' Class : 0
##
```