CARET PACKAGE

**1. Neural Networks with Feature Extraction (Rank Features by Importance)**

**Packages :** caret, mlbench, e1071, pROC

**Code :**

traindata<-waterdata2[,1:37]

trainclasses<-waterdata2[,38]

model <- train(traindata,trainclasses,method="pcaNNet", preProcess="scale", trControl=trainControl(method="cv"))

importance <- varImp(model, scale=FALSE)

print(importance)

plot(importance)

**Results**

only 20 most important variables shown (out of 37)

Importance

DBO.D 0.8204

PH.E 0.8143

PH.P 0.8097

DQO.D 0.8090

DBO.P 0.8086

PH.D 0.8082

DBO.E 0.7714

DQO.E 0.7663

COND.S 0.7556

COND.D 0.7554

SSV.E 0.7514

COND.E 0.7431

SS.D 0.7411

COND.P 0.7408

SSV.P 0.7370

SED.D 0.7359

SED.E 0.7090

SED.P 0.6798

RD.DBO.S 0.6509

RD.DQO.G 0.6497



**2. Remove Redundant Feature using Correlation**

**Ref :**<http://machinelearningmastery.com/feature-selection-with-the-caret-r-package/>

**Packages :** caret, mlbench

**Code :**

set.seed(7)

library(caret)

library(mlbench)

View(waterdata2)

correlationMatrix <- cor(waterdata2[,1:37])

print(correlationMatrix)

highlyCorrelated <- findCorrelation( correlationMatrix, cutoff = 0.5) # Ideal >0.75

print(highlyCorrelated)

**Results :**

highlyCorrelated : 17 16 36 23 24 6 12 4 25 10 32 35 21 11 30 5 14 15 8 13 9 18 27 20

Important Attributes/Predictors :

"ZN.E" "PH.E" "DBO.E" "SED.E" "SSV.D" "PH.S" "SSV.S" "COND.S" "RD.DBO.P" "RD.SED.P" "RD.DQO.S" "RD.DBO.G" "RD.SED.G"

**3. Rank Features by Importance using Linear Vector Quantization (LVQ) Model**

**Ref :** <http://machinelearningmastery.com/feature-selection-with-the-caret-r-package/>

**Packages :** mlbench, caret

**Code :**

set.seed(7)

library(caret)

library(mlbench)

traindata <- waterdata2[,1:37]

trainclasses <- waterdata2[,38]

control <- trainControl(method = "repeatedcv", number = 10, repeats = 3)

model <- train(traindata, trainclasses, method = "lvq", preProcess = "scale", trControl = control)

importance <- varImp(model, scale = FALSE)

print(importance)

plot(importance)

**Results :**

ROC curve variable importance

only 20 most important variables shown (out of 37)

Importance

DBO.D 0.8204

PH.E 0.8143

PH.P 0.8097

DQO.D 0.8090

DBO.P 0.8086

PH.D 0.8082

DBO.E 0.7714

DQO.E 0.7663

COND.S 0.7556

COND.D 0.7554

SSV.E 0.7514

COND.E 0.7431

SS.D 0.7411

COND.P 0.7408

SSV.P 0.7370

SED.D 0.7359

SED.E 0.7090

SED.P 0.6798

RD.DBO.S 0.6509

RD.DQO.G 0.6497



**4. Automatic Feature Selection Methods using Recursive Feature Elimination (RFE) with Random Forest Algorithm for each iteration to evaluate the model**

**Ref :** <http://machinelearningmastery.com/feature-selection-with-the-caret-r-package/>

**Packages :** caret, mlbench

**Code :**

set.seed(7)

library(mlbench)

library(caret)

traindata <- waterdata2[,1:37]

trainclasses <- waterdata2[,38]

control <- rfeControl(functions = rfFuncs, method = "cv", number = 10) # rfFuncs indicates Random Forest Algorithm

results <- rfe(traindata, trainclasses, sizes = c(1:37), rfeControl = control)

print(results)

predictors(results)

plot(results, type = c("g","o"))

**Results :**

Recursive feature selection

Outer resampling method: Cross-Validated (10 fold)

Resampling performance over subset size:

Variables/ Accuracy/ Kappa/ AccuracySD/ KappaSD/ Selected

1 0.8573 0.07835 0.03209 0.1135

2 0.9378 0.67038 0.02758 0.1461

3 0.9300 0.60649 0.02431 0.1585

4 0.9379 0.65597 0.03661 0.2144

5 0.9353 0.63642 0.03000 0.1588

6 0.9352 0.63569 0.02764 0.1468

7 0.9352 0.63569 0.02764 0.1468

8 0.9352 0.62535 0.02470 0.1321

9 0.9404 0.65457 0.02420 0.1380

10 0.9352 0.62663 0.02795 0.1508

11 0.9378 0.64145 0.03011 0.1681

12 0.9430 0.66396 0.02888 0.1577

13 0.9378 0.62812 0.03256 0.1988

14 0.9378 0.62592 0.03011 0.1858

15 0.9404 0.62293 0.03436 0.2219

16 0.9430 0.64972 0.03372 0.2039

17 0.9404 0.63718 0.02963 0.1827

18 0.9352 0.59528 0.03696 0.2324

19 0.9352 0.57755 0.03481 0.2313

20 0.9404 0.62204 0.03216 0.2132

21 0.9378 0.62503 0.02758 0.1755

22 0.9326 0.58054 0.03014 0.1978

23 0.9377 0.59525 0.03260 0.2269

24 0.9377 0.61078 0.03260 0.2151

25 0.9404 0.62204 0.03216 0.2132

26 0.9430 0.64883 0.03148 0.1946

27 0.9430 0.64883 0.03148 0.1946

28 0.9456 0.66898 0.03295 0.2039

29 0.9430 0.64021 0.03148 0.1977

30 0.9430 0.64883 0.03148 0.1946

31 0.9430 0.65488 0.02643 0.1704

32 0.9455 0.66395 0.02247 0.1602

33 0.9482 0.68410 0.02418 0.1694 \*

34 0.9404 0.61751 0.02994 0.2576

35 0.9429 0.64063 0.03165 0.2132

36 0.9455 0.65790 0.02823 0.1860

37 0.9455 0.66395 0.02247 0.1602

The top 5 variables (out of 33):

PH.D, DBO.P, DBO.D, PH.P, PH.E

Chosen Features :

"PH.D" "DBO.P" "DBO.D" "PH.P" "PH.E" "SSV.E" "DQO.D" "SSV.P" "DBO.E" "RD.DQO.G" "RD.SS.G" "SS.P" "DBO.S" "SS.S" "DQO.E" "SS.D" "COND.E" "SED.P" "COND.S" "COND.P" "COND.D" "SED.E" "RD.SS.P" "RD.DBO.S" "DQO.S" "RD.DBO.G" "SS.E" "RD.DBO.P" "RD.DQO.S" "SED.D" "PH.S" "ZN.E" "RD.SED.P"



**5. Boruta Package (Feature Selection Wrapper Algorithm)**

**Packages :** Boruta, mlbench

**Code :**

library(mlbench)

library(Boruta)

Boruta(c1~. , data = waterdata2, doTrace = 2) -> Bor.waterdata2

print(Bor.waterdata2)

Bor.waterdata2$finalDecision

plot(Bor.waterdata2)

plotImpHistory(Bor.waterdata2)

**Result :**

Boruta performed 99 iterations in 1.027557 mins. p-Value = 0.01

28 attributes confirmed important: COND.D, COND.E, COND.P, COND.S, DBO.D

and 23 more.

8 attributes confirmed unimportant: PH.S, RD.SED.G, RD.SED.P, SED.D, SED.S

and 3 more.

1 tentative attributes left: RD.DQO.S.

**ZN.E PH.E DBO.E DQO.E SS.E SSV.E SED.E COND.E**

**Rejected Confirmed Confirmed Confirmed Confirmed Confirmed Confirmed Confirmed**

**PH.P DBO.P SS.P SSV.P SED.P COND.P PH.D DBO.D**

**Confirmed Confirmed Confirmed Confirmed Confirmed Confirmed Confirmed Confirmed**

**DQO.D SS.D SSV.D SED.D COND.D PH.S DBO.S DQO.S**

**Confirmed Confirmed Rejected Rejected Confirmed Rejected Confirmed Confirmed**

**SS.S SSV.S SED.S COND.S RD.DBO.P RD.SS.P RD.SED.P RD.DBO.S**

**Confirmed Rejected Rejected Confirmed Confirmed Confirmed Rejected Confirmed**

**RD.DQO.S RD.DBO.G RD.DQO.G RD.SS.G RD.SED.G**

**Tentative Confirmed Confirmed Confirmed Rejected**

Levels: Tentative Confirmed Rejected





**6. Partial Least Squares Discriminant Analysis (PLS) for Resampling**

**Packages :** caret, mlbench

**Code :**

control <- trainControl(method = "repeatedcv", repeats = 3)

plsFit <- train(c1~., data = waterdata2, method = "pls", trControl = control, preProc = c("center","scale"))

plsFit

plot(plsFit)

**Result :**

Partial Least Squares

385 samples

37 predictor

2 classes: 'class1', 'class2'

Pre-processing: centered (37), scaled (37)

Resampling: Cross-Validated (10 fold, repeated 3 times)

Summary of sample sizes: 347, 346, 346, 347, 346, 347, ...

Resampling results across tuning parameters:

ncomp Accuracy Kappa Accuracy SD Kappa SD

1 0.9022492 0.2261767 0.02070739 0.2295315

2 0.9004723 0.1987270 0.02334476 0.2551592

3 0.9048133 0.2522108 0.02149547 0.2336183

Accuracy was used to select the optimal model using the largest value.

The final value used for the model was ncomp = 3.

