Practical machine learning Q2

Davy Meesemaecker

18/1/2018

Practical machine learning - Quiz 2

Question 2 :

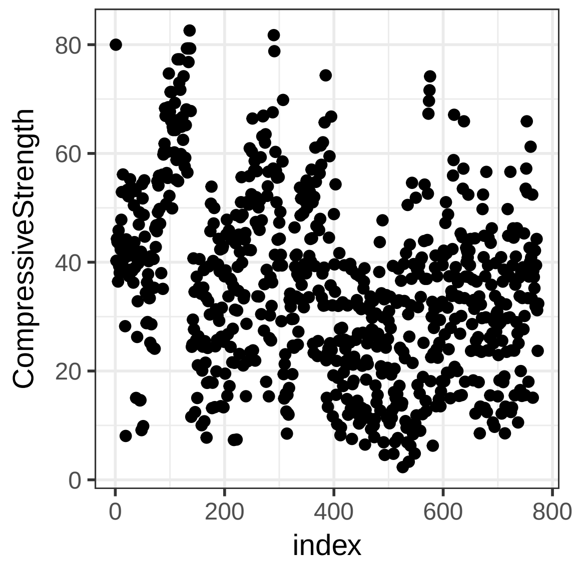
Load the cement data using the commands:

```
library(AppliedPredictiveModeling)
data(concrete)
library(caret)
set.seed(1000)
inTrain = createDataPartition(mixtures$CompressiveStrength, p = 3/4)[[1]]
training = mixtures[ inTrain,]
testing = mixtures[-inTrain,]
```

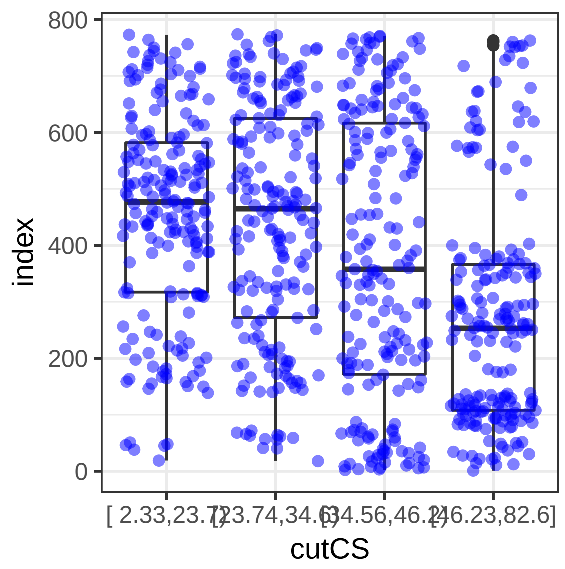
Make a plot of the outcome (CompressiveStrength) versus the index of the samples. Color by each of the variables in the data set (you may find the cut2() function in the Hmisc package useful for turning continuous covariates into factors). What do you notice in these plots?

Answer

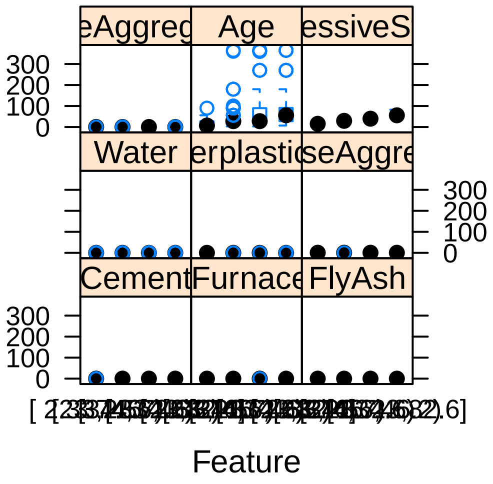
```
library(Hmisc)
index <- seq_along(1:nrow(training))
ggplot(data = training, aes(x = index, y = CompressiveStrength)) + geom_point() + theme_bw()
```



```
cutCS <- cut2(training$CompressiveStrength, g = 4)
ggplot(data = training, aes(y = index, x = cutCS)) + geom_boxplot() + geom_jitter(col = "blue", alpha =
```



```r
featurePlot(x = training, y = cutCS, plot = "box")
```



There is a non-random pattern in the plot of the outcome versus index that does not appear to be perfectly explained by any predictor suggesting a variable may be missing.

Question 4 :

Load the Alzheimer's disease data using the commands:

```r
library(caret)
library(AppliedPredictiveModeling)
set.seed(3433)
data(AlzheimerDisease)
adData = data.frame(diagnosis,predictors)
inTrain = createDataPartition(adData$diagnosis, p = 3/4)[[1]]
training = adData[ inTrain,]
```

```
testing = adData[-inTrain,]
```

Find all the predictor variables in the training set that begin with IL. Perform principal components on these variables with the preProcess() function from the caret package. Calculate the number of principal components needed to capture 90% of the variance. How many are there?

Answer

```
names(training)
```

```
##   [1] "diagnosis"                        "ACE_CD143_Angiotensin_Converti"
##   [3] "ACTH_Adrenocorticotropic_Hormon"  "AXL"
##   [5] "Adiponectin"                      "Alpha_1_Antichymotrypsin"
##   [7] "Alpha_1_Antitrypsin"              "Alpha_1_Microglobulin"
##   [9] "Alpha_2_Macroglobulin"            "Angiopoietin_2_ANG_2"
##  [11] "Angiotensinogen"                  "Apolipoprotein_A_IV"
##  [13] "Apolipoprotein_A1"                "Apolipoprotein_A2"
##  [15] "Apolipoprotein_B"                 "Apolipoprotein_CI"
##  [17] "Apolipoprotein_CIII"              "Apolipoprotein_D"
##  [19] "Apolipoprotein_E"                 "Apolipoprotein_H"
##  [21] "B_Lymphocyte_Chemoattractant_BL"  "BMP_6"
##  [23] "Beta_2_Microglobulin"             "Betacellulin"
##  [25] "C_Reactive_Protein"               "CD40"
##  [27] "CD5L"                             "Calbindin"
##  [29] "Calcitonin"                       "CgA"
##  [31] "Clusterin_Apo_J"                  "Complement_3"
##  [33] "Complement_Factor_H"              "Connective_Tissue_Growth_Factor"
##  [35] "Cortisol"                         "Creatine_Kinase_MB"
##  [37] "Cystatin_C"                       "EGF_R"
##  [39] "EN_RAGE"                          "ENA_78"
##  [41] "Eotaxin_3"                        "FAS"
##  [43] "FSH_Follicle_Stimulation_Hormon"  "Fas_Ligand"
##  [45] "Fatty_Acid_Binding_Protein"       "Ferritin"
##  [47] "Fetuin_A"                         "Fibrinogen"
##  [49] "GRO_alpha"                        "Gamma_Interferon_induced_Monokin"
##  [51] "Glutathione_S_Transferase_alpha"  "HB_EGF"
##  [53] "HCC_4"                            "Hepatocyte_Growth_Factor_HGF"
##  [55] "I_309"                            "ICAM_1"
##  [57] "IGF_BP_2"                         "IL_11"
##  [59] "IL_13"                            "IL_16"
##  [61] "IL_17E"                           "IL_1alpha"
##  [63] "IL_3"                             "IL_4"
##  [65] "IL_5"                             "IL_6"
##  [67] "IL_6_Receptor"                    "IL_7"
##  [69] "IL_8"                             "IP_10_Inducible_Protein_10"
##  [71] "IgA"                              "Insulin"
##  [73] "Kidney_Injury_Molecule_1_KIM_1"   "LOX_1"
##  [75] "Leptin"                           "Lipoprotein_a"
##  [77] "MCP_1"                            "MCP_2"
##  [79] "MIF"                              "MIP_1alpha"
##  [81] "MIP_1beta"                        "MMP_2"
##  [83] "MMP_3"                            "MMP10"
##  [85] "MMP7"                             "Myoglobin"
##  [87] "NT_proBNP"                         "NrCAM"
##  [89] "Osteopontin"                       "PAI_1"
##  [91] "PAPP_A"                            "PLGF"
##  [93] "PYY"                               "Pancreatic_polypeptide"
##  [95] "Prolactin"                         "Prostatic_Acid_Phosphatase"
##  [97] "Protein_S"                         "Pulmonary_and_Activation_Regulat"
##  [99] "RANTES"                            "Resistin"
## [101] "S100b"                             "SGOT"
## [103] "SHBG"                              "SOD"
## [105] "Serum_Amyloid_P"                   "Sortilin"
## [107] "Stem_Cell_Factor"                  "TGF_alpha"
## [109] "TIMP_1"                            "TNF_RII"
## [111] "TRAIL_R3"                          "TTR_prealbumin"
## [113] "Tamm_Horsfall_Protein_THP"         "Thrombomodulin"
## [115] "Thrombopoietin"                    "Thymus_Expressed_Chemokine_TECK"
## [117] "Thyroid_Stimulating_Hormone"       "Thyroxine_Binding_Globulin"
## [119] "Tissue_Factor"                     "Transferrin"
## [121] "Trefoil_Factor_3_TFF3"             "VCAM_1"
## [123] "VEGF"                              "Vitronectin"
## [125] "von_Willebrand_Factor"             "age"
## [127] "tau"                               "p_tau"
## [129] "Ab_42"                             "male"
## [131] "Genotype"
```

```
preproc <- preProcess(training[, c(58:69)], method = "pca", thresh = 0.9)
preproc
```

```
## Created from 251 samples and 12 variables
##
## Pre-processing:
##   - centered (12)
##   - ignored (0)
##   - principal component signal extraction (12)
##   - scaled (12)
##
## PCA needed 9 components to capture 90 percent of the variance
```

Question 5 :

Load the Alzheimer's disease data using the commands:

```
library(caret)
library(AppliedPredictiveModeling)
set.seed(3433)
data(AlzheimerDisease)
adData = data.frame(diagnosis,predictors)
inTrain = createDataPartition(adData$diagnosis, p = 3/4)[[1]]
training = adData[ inTrain,]
testing = adData[-inTrain,]
```

Create a training data set consisting of only the predictors with variable names beginning with IL and the diagnosis. Build two predictive models, one using the predictors as they are and one using PCA with principal components explaining 80% of the variance in the predictors. Use method="glm" in the train function.

What is the accuracy of each method in the test set? Which is more accurate?

Answer

```
preproc <- preProcess(training[, c(1, 58:69)], method = "pca", thresh = 0.8)
trainpc <- predict(preproc, training[, c(1,58:69)])
modelfit <- train(diagnosis ~ ., method = "glm", data = trainpc)
testpc <- predict(preproc, testing[, c(1,58:69)])
testPCA <- confusionMatrix(testing$diagnosis, predict(modelfit, testpc))
testPCA
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction Impaired Control
##   Impaired        3      19
##   Control         4      56
##
##                Accuracy : 0.7195
##                  95% CI : (0.6094, 0.8132)
##     No Information Rate : 0.9146
##     P-Value [Acc > NIR] : 1.000000
##
##                   Kappa : 0.0889
##  Mcnemar's Test P-Value : 0.003509
##
##             Sensitivity : 0.42857
##             Specificity : 0.74667
##          Pos Pred Value : 0.13636
##          Neg Pred Value : 0.93333
##              Prevalence : 0.08537
##          Detection Rate : 0.03659
##    Detection Prevalence : 0.26829
##       Balanced Accuracy : 0.58762
##
##        'Positive' Class : Impaired
##
```

Now we need to compare it to the one without pca

```
modelfit2 <- train(diagnosis~., method = "glm", data = training[, c(1, 58:69)])
testnonPCA <- confusionMatrix(testing$diagnosis, predict(modelfit2, testing))
testnonPCA
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction Impaired Control
##   Impaired        2      20
##   Control         9      51
##
##                Accuracy : 0.6463
##                  95% CI : (0.533, 0.7488)
##     No Information Rate : 0.8659
##     P-Value [Acc > NIR] : 1.00000
##
##                   Kappa : -0.0702
##  Mcnemar's Test P-Value : 0.06332
##
##             Sensitivity : 0.18182
##             Specificity : 0.71831
##          Pos Pred Value : 0.09091
##          Neg Pred Value : 0.85000
##              Prevalence : 0.13415
##          Detection Rate : 0.02439
##    Detection Prevalence : 0.26829
##       Balanced Accuracy : 0.45006
##
##        'Positive' Class : Impaired
##
```