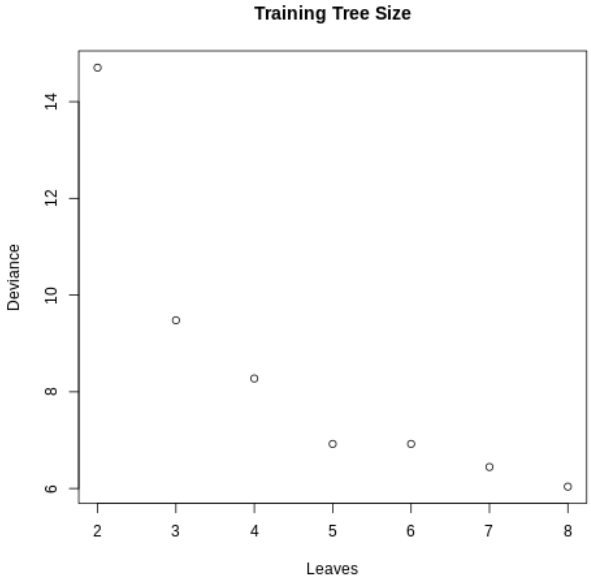
Machine Learning

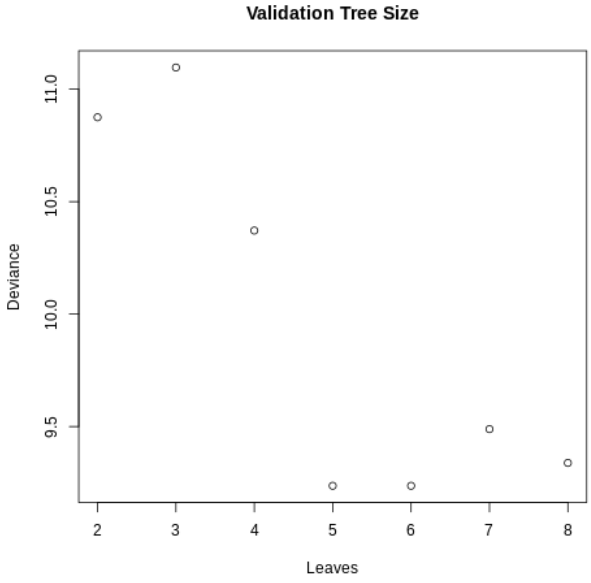
Report – Erik S. V. Jansson

Assignment 1

We are given the task to analyze the *Aluminium* chemical contents in *glass observations*, and see if it can be *described/related* to other chemicals. We use *CART* and *Partial Least Squares* as the chosen models for the task. See given Appendix.

Part 1: Regression Tree Model

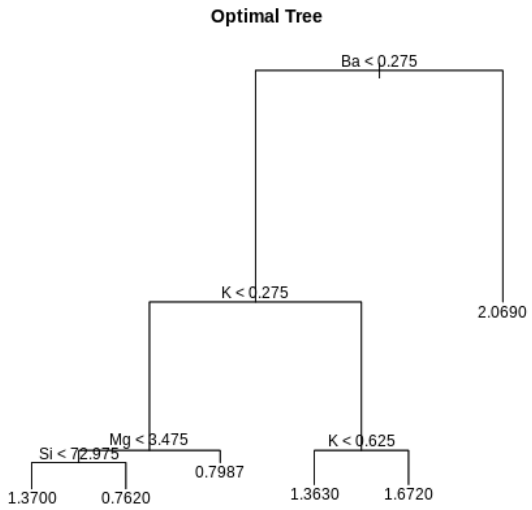




After fitting the model and pruning the tree one leaf addition at a time, we calculate the deviance of each increase in the leaf size to produce the plots in the previous page. As can be seen, the training data will continue to improve (because we fit the model with it) as the number of leaves increase. While the validation deviance will stop decreasing after a certain point, when the model is overfitted to the training data set. The optimal number of leaves is 5 since the deviance starts to become larger for the validation set after this point. In bias-variance tradeoff terms, the model will start to be a bit more biased to training data.

Part 2: Analysis of the Results

Below follows the optimal tree that was selected based on the lowest deviance found when doing a consideration of both training and validation data sets. The chosen variables were *Ba, K, Mg, and Si*, as can also be seen in the figure below...



Additionally, the *mean square error* of doing a prediction on the *testing data set* is ≈0.1280426, if we assume that the target variable is normally distributed. We don't use training or validation data when estimating the error, since these were used to fit and choose the "optimal" tree model. Notice in the figure above that the *K* variable is being used "twice" for the decision tree model...

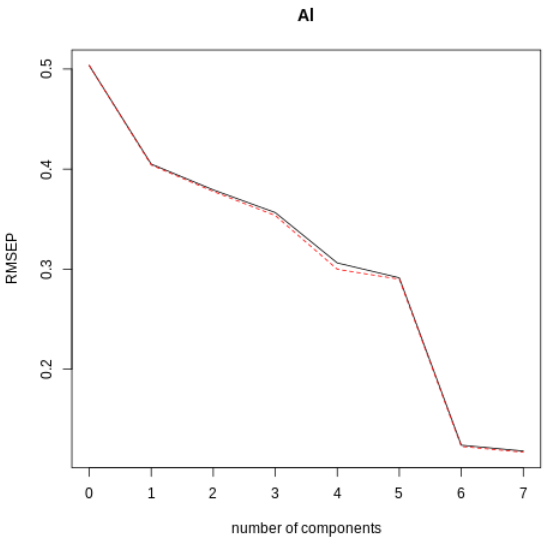
Part 3: PLS Regression Model

Now we attempt to fit a model using *Partial Least Squares Regression* instead. By using the *pls package* with the *plsr function*, the *summary* give us quite a lot of information. Here they are:

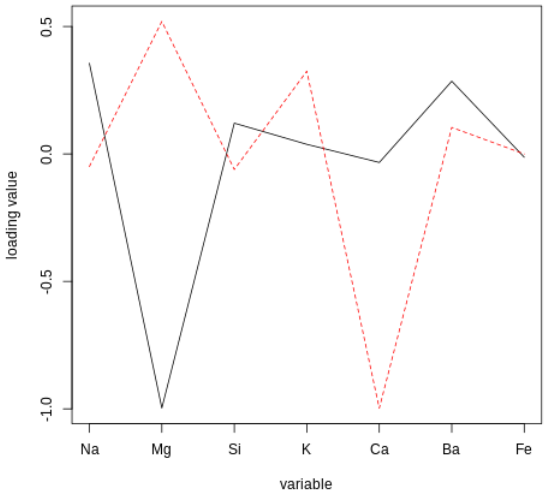
a) **How many variables are required to explain at least 90 % of the feature space?:** it seems that we need 5 to at least explain 90 % of the feature space, however, 4 components is *almost* there, explaining 89.90 % (5 explains 97.92%).

b) **How many variables are enough to explain at least 90 % of the target space?:** while for the target space we do need a bit more, at least 6 components is needed to explain 90 % of the space. The 5 components are not very close to 90%, being only 84.97 % (6 being the 95.6%).

c) **What is the optimal number of variables according to the cross-validation?:** it seems that of the 7 total variables considered, by cross-validation the optimal number is the entire 7 variables. The following plot can explain this:



d) **Which variables contribute mostly to the first principal component?:** see the plot on the next page for PCA p*loadings*:



As can be seen, Na *(Natrium)* has a high correlation with the first principal (*Mg)* component while Mg + *K* too has high correlation with the second component...

e) **What is the equation of the target in the coordinates of the principal components:** according to *summary(fit)*: 0.5036 + PC1*0.405 + PC2*0.3792 + PC3*0.3565 + PC4*0.3062 + PC5*0.291 + PC6*0.124 + PC7*0.1182. Where the first number is the intercept.

f) **What is the prediction error of the optimal PLS model for the test data?:** by taking the *mean square error* between the predicted and observed values, we get the prediction error of: 0.06939215.

Part 4: Model Comparison

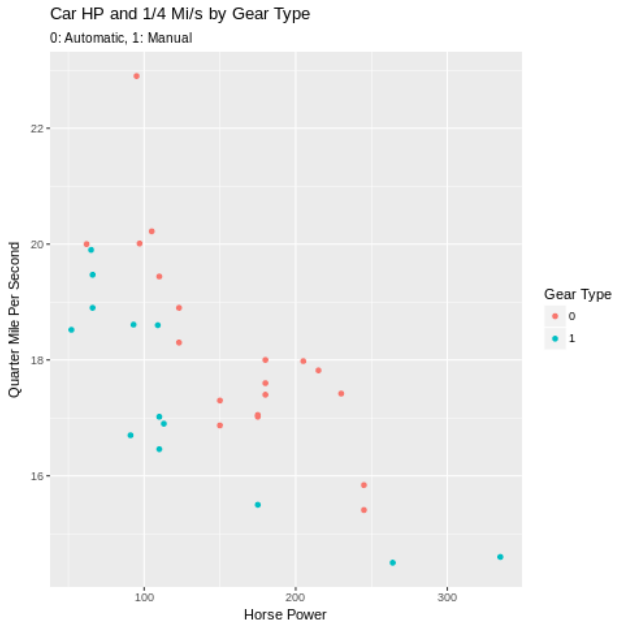
It seems that using *Partial Least Squares* with *cross-validation* paid off since it produces lower prediction errors than *Regression Trees* using the *Hold Out Method*. In this case, using *cross-validation* in the *PLSR* model could be one of the reason why it faired better, since the number of observations are low, the more training data we can get our hands on, the better. This simply isn't true for the hold out method, where we did need to allocate quite a large chunk of the data for the validation data set (better when N is big).

Assignment 2

In this assignment we are tasked with analyzing the relationship between the classification of the *gear method (manual or automatic)* given the *horsepower* and the *quarter of a mile time*. See the Appendix, for the entire source code for this.

Part 1: Analyzing the Data

Plotting the *Horse Power* against the *Quarter Miles Per Second* and identifying those cars which have *manual or automatic transmission*, gives the figure below. As can be seen, the faster the car is, the more inclined it's to be *automatic*. By looking a bit futher, it seems that the classes can be somewhat divided by a linear boundary. However, note that it matters greatly *where* we place this linear boundary, missclassifications can be reduced to a few outliers if done "right". We attempt LDA first, and try kernels later here.



Part 2: Fitting the LDA Model

Part 3: Fitting Kernel Density

Assignment 3

...

Appendix

Assignment 1

```
library(pls)
library(tree)

# Part 1: Regression Tree Model

set.seed(12345)
glass <- read.csv2("glass.csv")
glassy <- glass[3] ; glassx <- glass[-3]
training_indices <- sample(1:nrow(glass),
nrow(glass) / 2)
remaining_indices <- setdiff(1:nrow(glass),
training_indices)
testing_indices <- sample(remaining_indices,
length(remaining_indices) / 2)
validation_indices <- setdiff(remaining_indices,
testing_indices)
validation <- glass[validation_indices,]
training <- glass[training_indices,]
testing <- glass[testing_indices,]

fit <- tree(Al ~ ., data = training)
validation_deviances <- rep(0, 8)
training_deviances <- rep(0, 8)

for (leaves in 2:8) {
    pruned <- prune.tree(fit, best = leaves)
    predicted <- predict(pruned, newdata =
validation, type = "tree")
    validation_deviances[leaves] <-
deviance(predicted)
    training_deviances[leaves] <-
deviance(pruned)
}

png("training_deviance.png")
plot(2:8,training_deviances[2:8],
     xlab = "Leaves", ylab = "Deviance",
     main = "Training Tree Size")
dev.off()

png("validation_deviance.png")
plot(2:8,validation_deviances[2:8],
     xlab = "Leaves", ylab = "Deviance",
     main = "Validation Tree Size")
dev.off()

# Part 2: Analysis of the Results

# Derived from the above graphs.
fit <- prune.tree(fit, best = 5)
summary(fit) # Gives variables.

png("optimal_tree.png")
plot(fit) # Optimal P!
text(fit) # Nice tree!
title("Optimal Tree")
dev.off()

predicted <- predict(fit, newdata = testing,
type = "vector")
mean_square_error <- mean((predicted -
testing$Al)^2)

# Part 3: PLS Regression Model
```

```
full_training_indices <- c(training_indices,
validation_indices)
full_training <- glass[full_training_indices,] #
Using C-V here.
fit <- plsr(Al ~ ., data = full_training,
validation = "CV")

summary(fit) # Gives us most of the relevant
information.

png("validation_plot.png")
validationplot(fit)
dev.off()

png("loading_plot.png")
loadingplot(loadings(fit), label = "names")
dev.off()

predicted <- predict(fit, newdata = testing)
mean_square_error <- mean((predicted -
testing$Al)^2)

# Part 4: Model Comparison
# PLSR+CV seems bit better
```

Assignment 2

Assignment 3