Week 3 Homework

Ben Arancibia

June 23, 2015

KJ 7.2 Friedman (1991) introduced several benchmark datasets create by simulation. One of these simulations used the following non–linear equation to create data: $y = 10\sin(\pi x_1 x_2) + 20(x_3 - 0.5)^2 + 10x_4 + 5x_5 + N(0, \sigma^2)$

```
library(mlbench)
library(caret)
```

```
## Warning: package 'caret' was built under R version 3.1.3
```

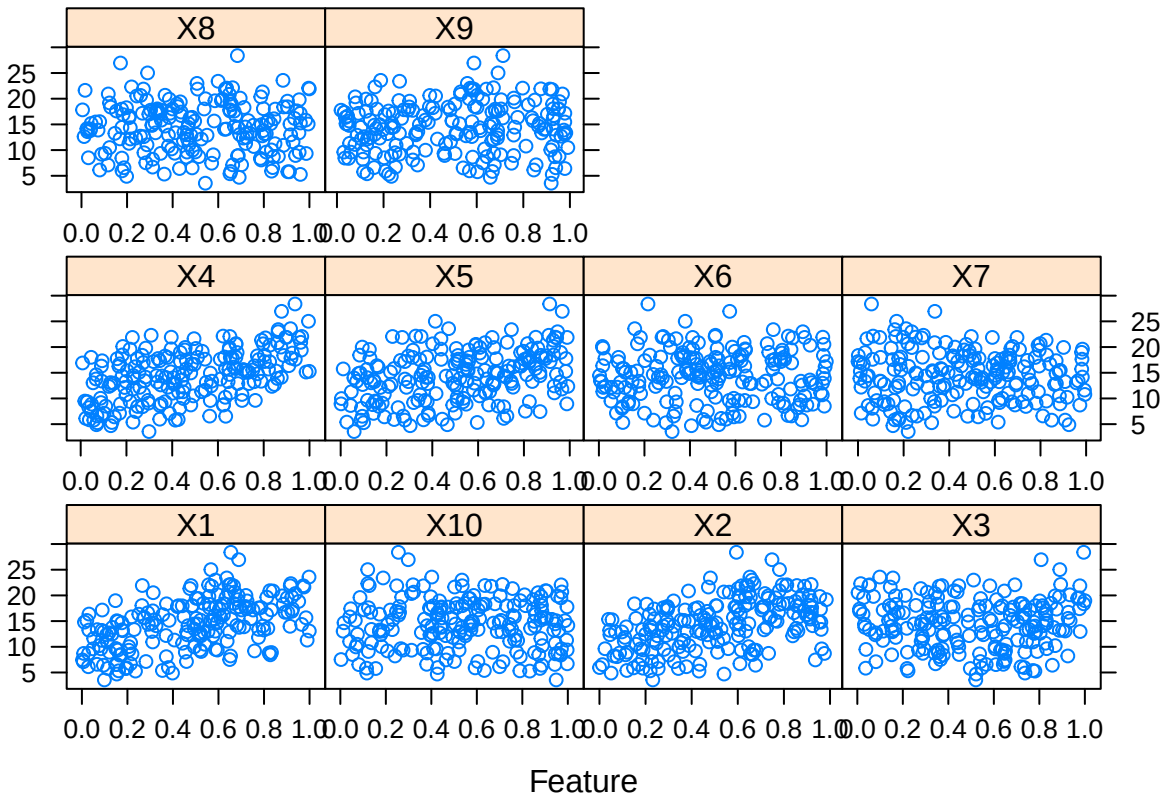
```
## Loading required package: lattice
```

```
## Warning: package 'lattice' was built under R version 3.1.3
```

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

```
## Warning: package 'ggplot2' was built under R version 3.1.3
```

```
set.seed(200)
trainingData <- mlbench.friedman1(200, sd = 1)
trainingData$x <- data.frame(trainingData$x)
featurePlot(trainingData$x, trainingData$y)
```



```
testData <- mlbench.friedman1(5000, sd = 1)
testData$x <- data.frame(testData$x)
```

Tune several models on these data. For example:

```
set.seed(921)
knnModel <- train(x = trainingData$x, y = trainingData$y, method = "knn",
                  preProc = c("center", "scale"),
                  tuneLength = 10)
knnModel
```

```
## k-Nearest Neighbors
##
## 200 samples
##  10 predictor
##
## Pre-processing: centered, scaled
## Resampling: Bootstrapped (25 reps)
##
## Summary of sample sizes: 200, 200, 200, 200, 200, 200, ...
##
## Resampling results across tuning parameters:
##
##   k   RMSE      Rsquared   RMSE SD    Rsquared SD
##    5  3.488933  0.5019753  0.2658769  0.07412999
##    7  3.324957  0.5484600  0.2275136  0.06432697
##    9  3.224541  0.5853589  0.2425946  0.06903863
##   11  3.178450  0.6091595  0.2593909  0.07304742
##   13  3.183655  0.6183553  0.2523970  0.06995701
##   15  3.188007  0.6250729  0.2408867  0.06604822
##   17  3.214343  0.6281943  0.1890541  0.05315353
##   19  3.208743  0.6403733  0.1921773  0.05056336
##   21  3.215199  0.6487431  0.1830866  0.04961129
##   23  3.235167  0.6528070  0.1870124  0.04751686
##
## RMSE was used to select the optimal model using  the smallest value.
## The final value used for the model was k = 11.
```

```
knnPred <- predict(knnModel, newdata = testData$x)
postResample(pred = knnPred, obs = testData$y)
```

```
##      RMSE  Rsquared
## 3.1222641 0.6690472
```

Which models appear to give the best performance? Does MARS select the informative predictors (those named X1–X5)?

K-nearest neighbors models are better when predictors and the response relies on the samples' proximity in the predictor space.