Homework 3 Solutions

Brian Lois

2/4/2017

1.

a)

The maximum R^2 value is 0.545 with a standard error of .0308. Using the one standard error method, 0.545 - .0308 = 0.5142. The least number of components with $R^2 > 0.5142$ is 3.

b)

The tolerance values can be computed as

```
R2 <- c(.444, .5, .533, .545, .542, .537, .534, .534, .520, .507)
( tolerances <-  1 - (R2 / max(R2)) )
```

```
##  [1] 0.185321101 0.082568807 0.022018349 0.000000000 0.005504587
##  [6] 0.014678899 0.020183486 0.020183486 0.045871560 0.069724771
```

The best choice is then

```
min(which(tolerances <= .1))
```

```
## [1] 2
```

c)

To maximize R^2, choose the random forest model.

d)

If time is also a consideration, support vector machine or boosted linear regression would be good options.

2.

a)

```
data(oil)
original_distribution <- table(oilType) / length(oilType)
random_sample <- sample(oilType, 60)
sampled_distribution <- table(random_sample) / length(random_sample)
(sampled_distribution - original_distribution) / original_distribution
```

```
## random_sample
##           A           B           C           D           E           F
##  0.08108108 -0.20000000  0.06666667  0.37142857 -0.12727273  0.12000000
##           G
## -0.20000000
```

In category C the two distributions differ by 100%. The next highest is category E with a 27% difference. (Your values may differ due to randomness.) The problem is more pronounced for the smaller categories.

b)

Doing a stratefied sample

```
frac_to_keep <- 60 / length(oilType)
random_sample_indices <- createDataPartition(oilType, p = frac_to_keep)
random_sample <- oilType[random_sample_indices$Resample1]
sampled_distribution <- table(random_sample) / length(random_sample)
(sampled_distribution - original_distribution) / original_distribution
```

```
## random_sample
##           A           B           C           D           E           F
## -0.02702703 -0.01923077  0.00000000  0.07142857 -0.04545455  0.05000000
##           G
##  0.50000000
```

All categories except G are now within 10% of the original.

```
?createDataPartition
```

When `y` is numeric, the argument `groups` controls the number of quantiles into which `y` is split for sampling.

c)

For small sample sizes, repeated k-fold cross validation, or bootstrapping are good choices A test set is not a good use of samples for small datasets.

3.

a)

```
data(Glass)
train_samples <- createDataPartition(Glass$Type, p = .8)

train_data = Glass[train_samples$Resample1, ]
head(train_data)
```

```
##        RI    Na   Mg   Al    Si    K   Ca Ba   Fe Type
## 1 1.52101 13.64 4.49 1.10 71.78 0.06 8.75  0 0.00    1
## 2 1.51761 13.89 3.60 1.36 72.73 0.48 7.83  0 0.00    1
```

```
## 3 1.51618 13.53 3.55 1.54 72.99 0.39 7.78  0 0.00    1
## 4 1.51766 13.21 3.69 1.29 72.61 0.57 8.22  0 0.00    1
## 5 1.51742 13.27 3.62 1.24 73.08 0.55 8.07  0 0.00    1
## 6 1.51596 12.79 3.61 1.62 72.97 0.64 8.07  0 0.26    1
```

```
test_data = Glass[-train_samples$Resample1, ]
head(test_data)
```

```
##         RI    Na   Mg   Al    Si    K   Ca Ba   Fe Type
## 10 1.51755 13.00 3.60 1.36 72.99 0.57 8.40  0 0.11    1
## 12 1.51763 12.80 3.66 1.27 73.01 0.60 8.56  0 0.00    1
## 16 1.51761 12.81 3.54 1.23 73.24 0.58 8.39  0 0.00    1
## 19 1.51911 13.90 3.73 1.18 72.12 0.06 8.89  0 0.00    1
## 20 1.51735 13.02 3.54 1.69 72.73 0.54 8.44  0 0.07    1
## 22 1.51966 14.77 3.75 0.29 72.02 0.03 9.00  0 0.00    1
```

b)

```
cv_samples = createFolds(Glass$Type, k = 3, list = FALSE)

train_1 <- Glass[cv_samples %in% c(1,2), ]
head(train_1)
```

```
##         RI    Na   Mg   Al    Si    K   Ca Ba   Fe Type
## 1  1.52101 13.64 4.49 1.10 71.78 0.06 8.75  0 0.00    1
## 2  1.51761 13.89 3.60 1.36 72.73 0.48 7.83  0 0.00    1
## 3  1.51618 13.53 3.55 1.54 72.99 0.39 7.78  0 0.00    1
## 8  1.51756 13.15 3.61 1.05 73.24 0.57 8.24  0 0.00    1
## 9  1.51918 14.04 3.58 1.37 72.08 0.56 8.30  0 0.00    1
## 10 1.51755 13.00 3.60 1.36 72.99 0.57 8.40  0 0.11    1
```

```
test_1 <- Glass[cv_samples == 3, ]
head(test_1)
```

```
##         RI    Na   Mg   Al    Si    K   Ca Ba   Fe Type
## 4  1.51766 13.21 3.69 1.29 72.61 0.57 8.22  0 0.00    1
## 5  1.51742 13.27 3.62 1.24 73.08 0.55 8.07  0 0.00    1
## 6  1.51596 12.79 3.61 1.62 72.97 0.64 8.07  0 0.26    1
## 7  1.51743 13.30 3.60 1.14 73.09 0.58 8.17  0 0.00    1
## 11 1.51571 12.72 3.46 1.56 73.20 0.67 8.09  0 0.24    1
## 19 1.51911 13.90 3.73 1.18 72.12 0.06 8.89  0 0.00    1
```

```
train_2 <- Glass[cv_samples %in% c(1,3), ]
head(train_2)
```

```
##        RI    Na   Mg   Al    Si    K   Ca Ba   Fe Type
## 1 1.52101 13.64 4.49 1.10 71.78 0.06 8.75  0 0.00    1
## 2 1.51761 13.89 3.60 1.36 72.73 0.48 7.83  0 0.00    1
## 3 1.51618 13.53 3.55 1.54 72.99 0.39 7.78  0 0.00    1
## 4 1.51766 13.21 3.69 1.29 72.61 0.57 8.22  0 0.00    1
## 5 1.51742 13.27 3.62 1.24 73.08 0.55 8.07  0 0.00    1
## 6 1.51596 12.79 3.61 1.62 72.97 0.64 8.07  0 0.26    1
```

```
test_2 <- Glass[cv_samples == 2, ]
head(test_2)
```

```
##         RI    Na   Mg   Al    Si    K   Ca Ba   Fe Type
## 13 1.51589 12.88 3.43 1.40 73.28 0.69 8.05  0 0.24    1
## 14 1.51748 12.86 3.56 1.27 73.21 0.54 8.38  0 0.17    1
## 15 1.51763 12.61 3.59 1.31 73.29 0.58 8.50  0 0.00    1
## 18 1.52196 14.36 3.85 0.89 71.36 0.15 9.15  0 0.00    1
## 27 1.51793 13.21 3.48 1.41 72.64 0.59 8.43  0 0.00    1
## 29 1.51768 12.56 3.52 1.43 73.15 0.57 8.54  0 0.00    1
```

```
train_3 <- Glass[cv_samples %in% c(2,3), ]
head(train_3)
```

```
##         RI    Na   Mg   Al    Si    K   Ca Ba   Fe Type
## 4  1.51766 13.21 3.69 1.29 72.61 0.57 8.22  0 0.00    1
## 5  1.51742 13.27 3.62 1.24 73.08 0.55 8.07  0 0.00    1
## 6  1.51596 12.79 3.61 1.62 72.97 0.64 8.07  0 0.26    1
## 7  1.51743 13.30 3.60 1.14 73.09 0.58 8.17  0 0.00    1
## 11 1.51571 12.72 3.46 1.56 73.20 0.67 8.09  0 0.24    1
## 13 1.51589 12.88 3.43 1.40 73.28 0.69 8.05  0 0.24    1
```

```
test_3 <- Glass[cv_samples == 1, ]
head(test_3)
```

```
##         RI    Na   Mg   Al    Si    K   Ca Ba   Fe Type
## 1  1.52101 13.64 4.49 1.10 71.78 0.06 8.75  0 0.00    1
## 2  1.51761 13.89 3.60 1.36 72.73 0.48 7.83  0 0.00    1
## 3  1.51618 13.53 3.55 1.54 72.99 0.39 7.78  0 0.00    1
## 8  1.51756 13.15 3.61 1.05 73.24 0.57 8.24  0 0.00    1
## 9  1.51918 14.04 3.58 1.37 72.08 0.56 8.30  0 0.00    1
## 10 1.51755 13.00 3.60 1.36 72.99 0.57 8.40  0 0.11    1
```

c)

```
bootstrap_samples <- createResample(Glass$Type, times = 1, list = FALSE)
bootstrap_train <- Glass[bootstrap_samples, ]
head(bootstrap_train)
```

```
##          RI    Na   Mg   Al    Si    K   Ca Ba Fe Type
## 2   1.51761 13.89 3.60 1.36 72.73 0.48 7.83  0  0    1
## 3   1.51618 13.53 3.55 1.54 72.99 0.39 7.78  0  0    1
## 3.1 1.51618 13.53 3.55 1.54 72.99 0.39 7.78  0  0    1
## 4   1.51766 13.21 3.69 1.29 72.61 0.57 8.22  0  0    1
## 7   1.51743 13.30 3.60 1.14 73.09 0.58 8.17  0  0    1
## 8   1.51756 13.15 3.61 1.05 73.24 0.57 8.24  0  0    1
```

```r
bootstrap_test <- Glass[-bootstrap_samples, ]
head(bootstrap_test)
```

```
##         RI    Na   Mg   Al    Si    K   Ca Ba   Fe Type
## 1  1.52101 13.64 4.49 1.10 71.78 0.06 8.75  0 0.00    1
## 5  1.51742 13.27 3.62 1.24 73.08 0.55 8.07  0 0.00    1
## 6  1.51596 12.79 3.61 1.62 72.97 0.64 8.07  0 0.26    1
## 12 1.51763 12.80 3.66 1.27 73.01 0.60 8.56  0 0.00    1
## 14 1.51748 12.86 3.56 1.27 73.21 0.54 8.38  0 0.17    1
## 15 1.51763 12.61 3.59 1.31 73.29 0.58 8.50  0 0.00    1
```