Practical Machine Learning

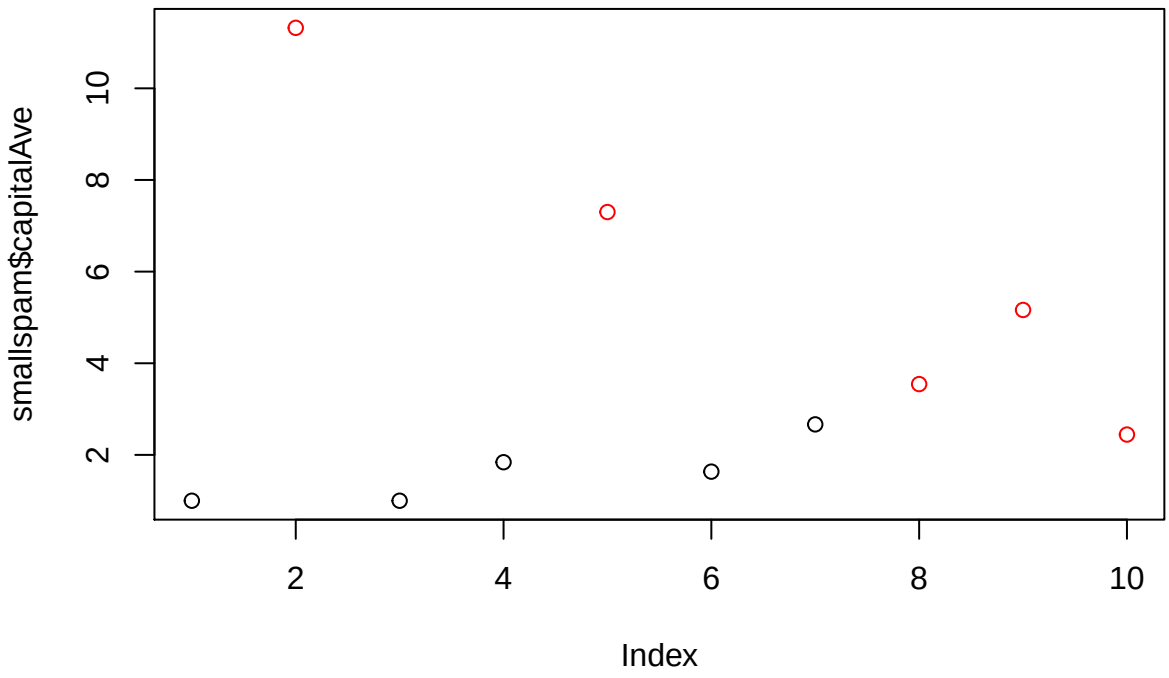
Sandeep Anand

April 30, 2017

Practical Machine Learning

- **Plotting the mails which have capital letter**

```
library(kernlab);data(spam);set.seed(333)
smallspam<-spam[sample(dim(spam)[1], size = 10),]
spamLabel<-(smallspam$type=="spam")*1+1
plot(smallspam$capitalAve, col=spamLabel)
```




```
#Functions to apply Rules to the spam data
rule1<-function(x)
{
  prediction<-rep(NA, length(x))
  prediction[x>2.7]<-"spam"
  prediction[x<2.40]<-"nonspam"
  prediction[x>=2.40 & x<=2.45]<-"spam"
  prediction[x>=2.45 & x<=2.70]<-"nonspam"
  return(prediction)
}
```

```
rule2<-function(x)
{
  prediction<-rep(NA, length(x))
  prediction[x>2.40]<-"spam"
  prediction[x<=2.40]<-"nonspam"
  return(prediction)
}

table(rule1(smallspam$capitalAve), smallspam$type)
```

```
##
##           nonspam spam
##   nonspam       5    0
##   spam          0    5
```

```
table(rule2(smallspam$capitalAve), smallspam$type)
```

```
##
##           nonspam spam
##   nonspam       4    0
##   spam          1    5
```

Applying the above rule functions to all the spam data

- Checking how our rules fit and what are the errors seen
- The diagonal elements provide us with the errors
- Looking at accuracy as well , checking the number of times we are correct for both our rules
- Overfitting - 'Overfitting' A modeling error which occurs when a function is too closely fit to a limited set of data points. Overfitting the model generally takes the form of making an overly complex model to explain idiosyncrasies in the data under study.

```
table(rule1(spam$capitalAve), spam$type)
```

```
##
##           nonspam spam
##   nonspam    2144  589
##   spam        644 1224
```

```
table(rule2(spam$capitalAve), spam$type)
```

```
##
##           nonspam spam
##   nonspam    1985  498
##   spam        803 1315
```

```
sum(rule1(spam$capitalAve)==spam$type)
```

```
## [1] 3368
```

```
sum(rule2(spam$capitalAve)==spam$type)
```

```
## [1] 3300
```