

# In sample and out of sample error

Jeffrey Leek

May 18, 2016

# In sample versus out of sample

**In Sample Error:** The error rate you get on the same data set you used to build your predictor. Sometimes called resubstitution error.

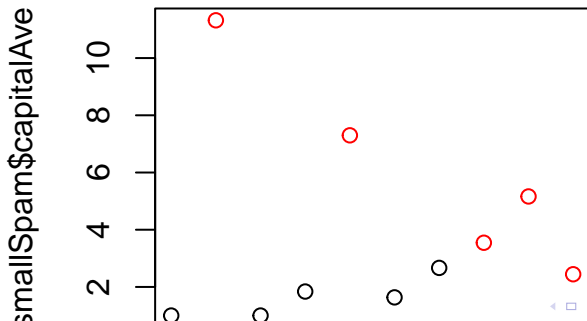
**Out of Sample Error:** The error rate you get on a new data set. Sometimes called generalization error.

## Key ideas

1. Out of sample error is what you care about
  2. In sample error  $<$  out of sample error
  3. The reason is overfitting
- Matching your algorithm to the data you have

## In sample versus out of sample errors

```
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)
```



# Prediction rule 1

- ▶  $\text{capitalAve} > 2.7 = \text{"spam"}$
- ▶  $\text{capitalAve} < 2.40 = \text{"nonspam"}$
- ▶  $\text{capitalAve} \text{ between } 2.40 \text{ and } 2.45 = \text{"spam"}$
- ▶  $\text{capitalAve} \text{ between } 2.45 \text{ and } 2.7 = \text{"nonspam"}$

## Apply Rule 1 to smallSpam

```
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)  
}  
table(rule1(smallSpam$capitalAve),smallSpam$type)
```

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

## Prediction rule 2

- ▶  $\text{capitalAve} > 2.40 = \text{"spam"}$
- ▶  $\text{capitalAve} \leq 2.40 = \text{"nonspam"}$

## Apply Rule 2 to smallSpam

```
rule2 <- function(x){  
  prediction <- rep(NA,length(x))  
  prediction[x > 2.8] <- "spam"  
  prediction[x <= 2.8] <- "nonspam"  
  return(prediction)  
}  
table(rule2(smallSpam$capitalAve),smallSpam$type)
```

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

## Apply to complete spam data

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

```
##  
##           nonspam spam  
## nonspam      2141  588  
## spam         647 1225
```

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

```
##  
##           nonspam spam  
## nonspam      2224  642  
## spam         564 1171
```

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

```
## [1] 0.7315801
```



## Look at accuracy

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

```
## [1] 3366
```

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

```
## [1] 3395
```

# What's going on?

## Overfitting

- ▶ Data have two parts
- ▶ Signal
- ▶ Noise
- ▶ The goal of a predictor is to find signal
- ▶ You can always design a perfect in-sample predictor
- ▶ You capture both signal + noise when you do that
- ▶ Predictor won't perform as well on new samples

<http://en.wikipedia.org/wiki/Overfitting>