A survey of methods for model validation with applications in R

Christian Thiele
@thie1e
c.thiele@hs-osnabrueck.de

12.01.2017

Hochschule Osnabrück
University of Applied Sciences

Motivating example

At the data dive last year the teams were asked to predict visitors of the Nettebad in Osnabrück:

- Get training data

Motivating example

At the data dive last year the teams were asked to predict visitors of the Nettebad in Osnabrück:

- Get training data
- Build your model

Motivating example

At the data dive last year the teams were asked to predict visitors of the Nettebad in Osnabrück:

- Get training data
- Build your model
- (validate your model??)

Motivating example

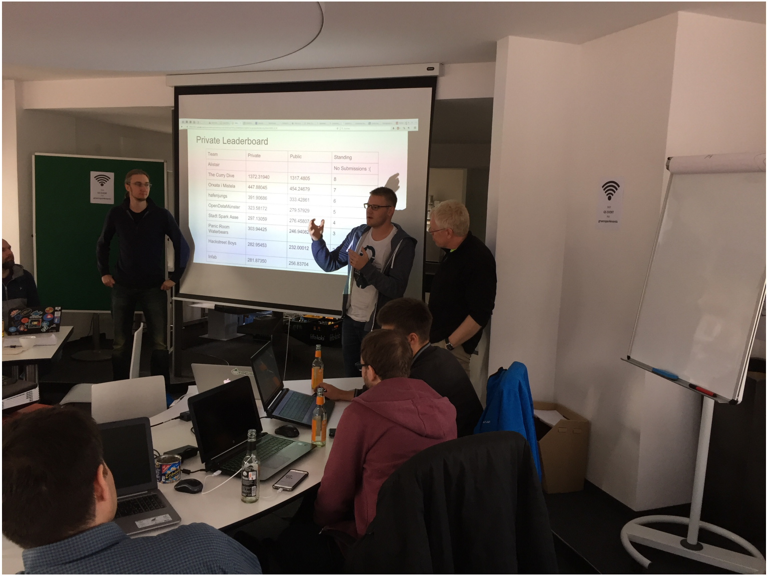
At the data dive last year the teams were asked to predict visitors of the Nettebad in Osnabrück:

- Get training data
- Build your model
- (validate your model??)
- Get score on a public data set

Motivating example

At the data dive last year the teams were asked to predict visitors of the Nettebad in Osnabrück:

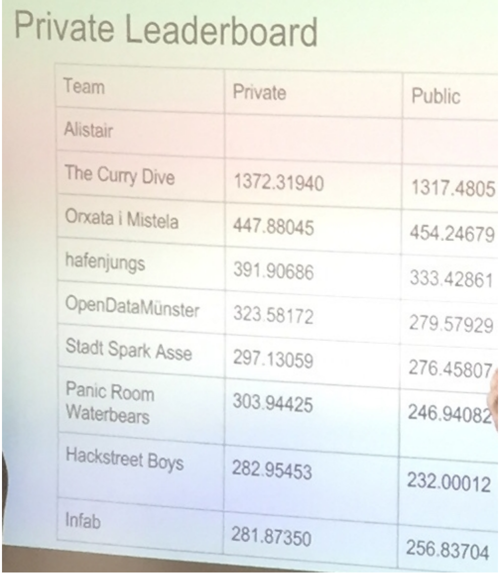
- Get training data
- Build your model
- (validate your model??)
- Get score on a public data set
- Be ranked **solely** on unknown data



Private Leaderboard

Team	Private	Public	Standing
Alistair			No Submissions :(
The Curry Dive	1372.31940	1317.4805	8
Orxata i Mistela	447.88045	454.24679	7
hafenjungs	391.90686	333.42861	6
OpenDataMünster	323.58172	279.57929	5
Stadt Spark Asse	297.13059	276.45807	4
Panic Room Waterbears	303.94425	246.94082	3
Hackstreet Boys	282.95453	232.00012	2
Infab	281.87350	256.83704	[illegible]

Figure 1: After the Data Dive



Private Leaderboard

Team	Private	Public
Alistair		
The Curry Dive	1372.31940	1317.4805
Orxata i Mistela	447.88045	454.24679
hafenjungs	391.90686	333.42861
OpenDataMünster	323.58172	279.57929
Stadt Spark Asse	297.13059	276.45807
Panic Room Waterbears	303.94425	246.94082
Hackstreet Boys	282.95453	232.00012
Infab	281.87350	256.83704

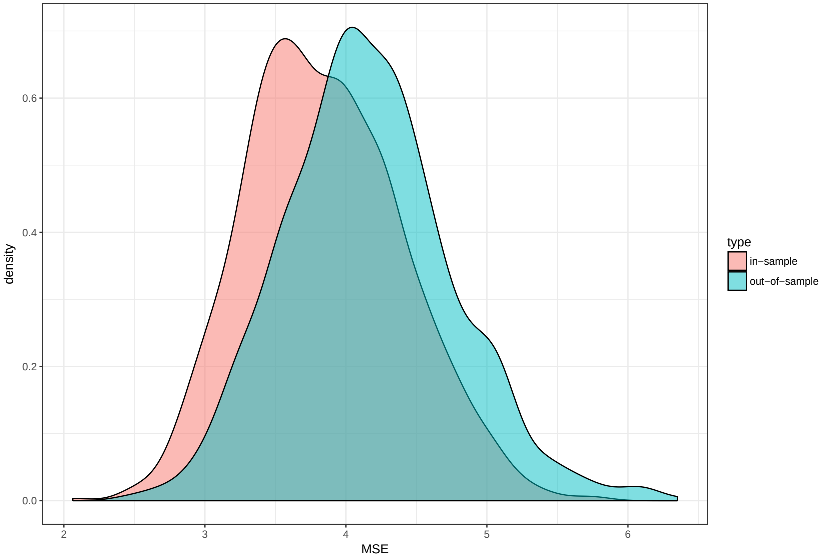
Figure 2: After the Data Dive (zoomed in)

Goals for today

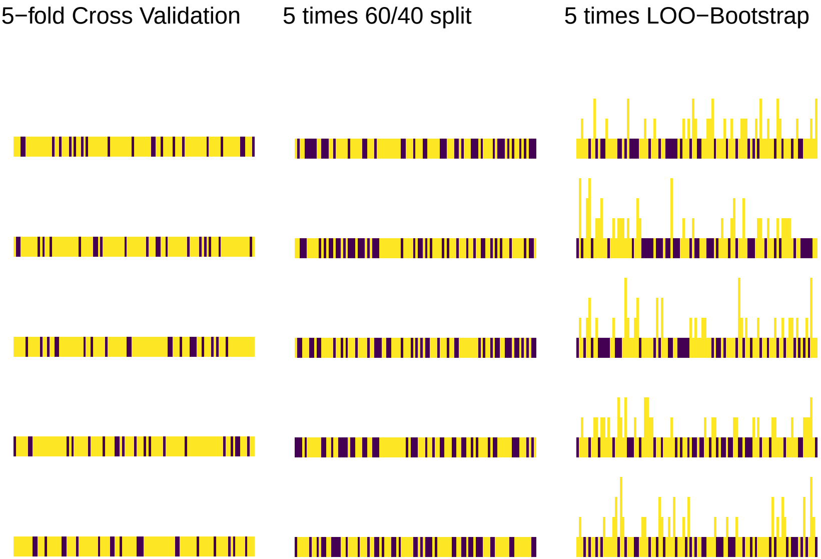
- See how different validation methods stack up
- Some examples in R (`caret` package)

Is it worthwile to consider the performance on known data?

```r
# Adapted from: http://eranraviv.com
n_sim <- 1000
MSE <- oosMSE <- rep(NA, n_sim)
# define the noise level
sig <- 2
# 3 explanatories
pp <-  3
# define the signal level
signall <- 10
for (i in 1:n_sim) {
    beta = seq(from = 1, to = signall, length.out = 4)
    X = cbind(rep(1, 100), matrix(rnorm(100 * pp), nrow = 100))
    Y = X %*% beta + sig * rnorm(100)
    Ynew = X %*% beta + sig * rnorm(100)
    mod = lm(Y ~ X - 1)
    Yhat = predict(mod)
    MSE[i] = mean((Y - Yhat)^2)
    oosMSE[i] = mean((Ynew - Yhat)^2)
}
```


Even if we get the model exactly right (which we don't), the training performance tends to be optimistic

Resampling methods



Training / Test split

- **Idea:** Split dataset in two sets
- **Variants:**
  - Repetitions: Can be repeated by randomly sampling new split s n times and averaging the results

Cross validation

- **Idea:** Split into k distinct sets of size $\frac{n}{k}$. For 1 to k:
  - Fit the model using the remaining $n - \frac{n}{k}$ observations
  - Average the k test set results
- **Variants:**
  - Number of folds (k usually between 2 and 10)
  - Repetitions: Average the results of multiple CV runs

Bootstrapping

- **Idea:** Draw with replacement a random bootstrap-sample x_b with size equal to the original sample x
- **Variants:**
  - Conventional Bootstrap: Train on x_b, test on x
  - Leave-one-out bootstrap: Train on x_b, test on $x \notin x_b$

Validation methods are just estimators, too!

- We know of bias / variance tradeoffs depending on model complexity

Validation methods are just estimators, too!

- We know of bias / variance tradeoffs depending on model complexity
- By retraining and retesting the model on resampled data (say, during CV) we want an estimate of the out-of-sample performance

Validation methods are just estimators, too!

- We know of bias / variance tradeoffs depending on model complexity
- By retraining and retesting the model on resampled data (say, during CV) we want an estimate of the out-of-sample performance
- These methods suffer from bias and variance just like an “ordinary” Machine Learning model!

Validation methods are just estimators, too!

- We know of bias / variance tradeoffs depending on model complexity
- By retraining and retesting the model on resampled data (say, during CV) we want an estimate of the out-of-sample performance
- These methods suffer from bias and variance just like an "ordinary" Machine Learning model!
- Bias: Resampling can result in a systematically higher (or lower) estimate of the true external performance

Validation methods are just estimators, too!

- We know of bias / variance tradeoffs depending on model complexity
- By retraining and retesting the model on resampled data (say, during CV) we want an estimate of the out-of-sample performance
- These methods suffer from bias and variance just like an "ordinary" Machine Learning model!
- Bias: Resampling can result in a systematically higher (or lower) estimate of the true external performance
- Variance: Even if unbiased, the estimate may vary considerably around the true external performance

Simulation

Data

- 795 observations
- Binary outcome: Migraine yes (56%) / no (44%)
- 9 independent variables (frequency of pain, auxiliary symptoms, age, etc.)

Classification and regression tree (CART, via `rpart`)

- Variables: All 9 variables
- Maximum tree depth: 4
- Minimum observations in a node to attempt a split: 6

Simulation

Nested simulation procedure

1. Draw a very small sample ($n = 30, 45, 60$ or 80) from the data set. Mimics analyzing a sample from a large population

Simulation

Nested simulation procedure

1. Draw a very small sample ($n = 30$, 45, 60 or 80) from the data set. Mimics analyzing a sample from a large population
2. Use this sample for model fitting and the validation procedure

Simulation

Nested simulation procedure

1. Draw a very small sample (n = 30, 45, 60 or 80) from the data set. Mimics analyzing a sample from a large population
2. Use this sample for model fitting and the validation procedure
3. Check pseudo out-of-sample performance on the rest of the data set and compare with the estimate from the internal validation

Simulation

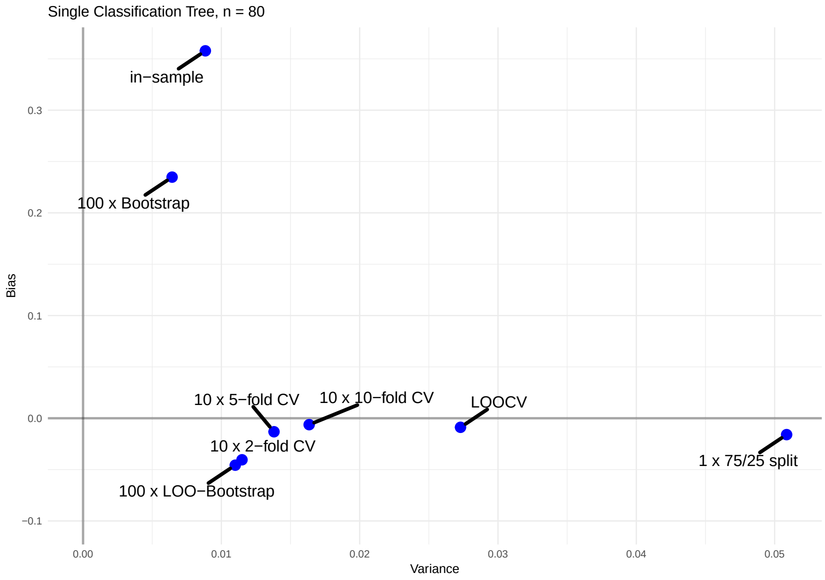
Nested simulation procedure

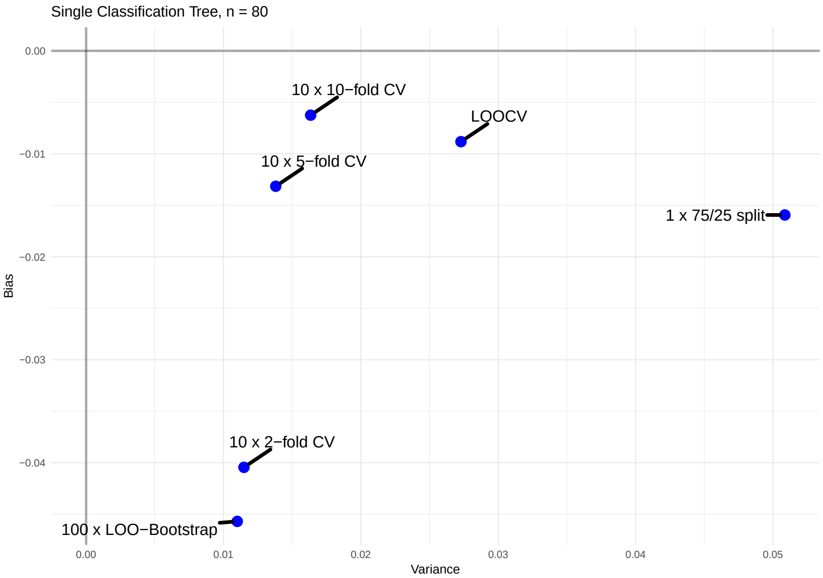
1. Draw a very small sample (n = 30, 45, 60 or 80) from the data set. Mimics analyzing a sample from a large population
2. Use this sample for model fitting and the validation procedure
3. Check pseudo out-of-sample performance on the rest of the data set and compare with the estimate from the internal validation
4. Compare the performance of the resampling methods in terms of bias and variance

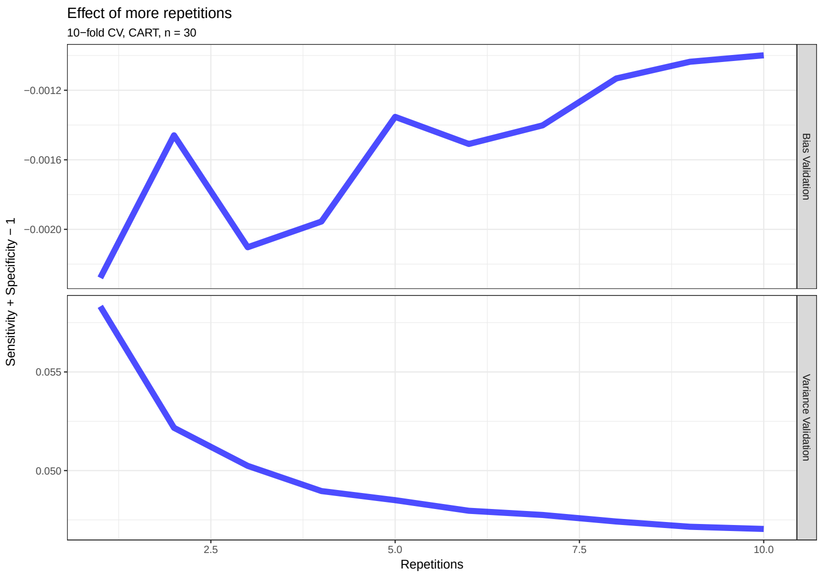
Simulation

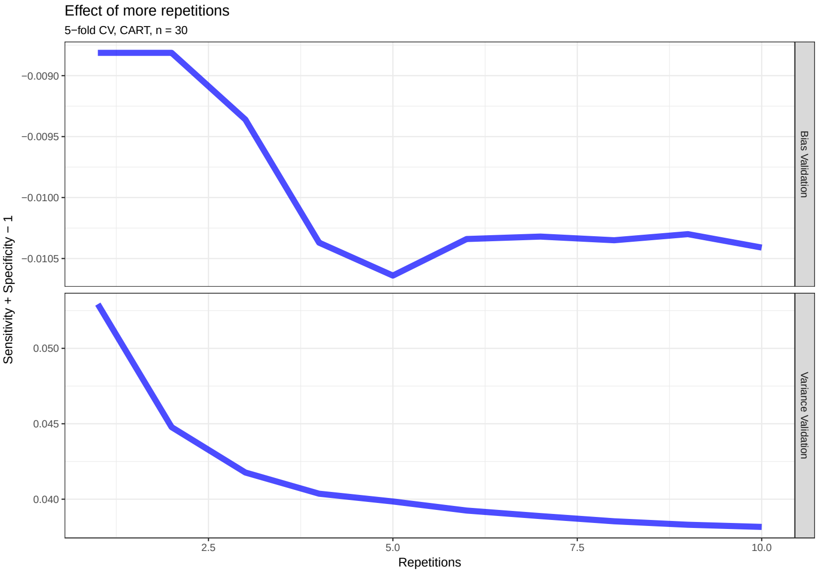
Nested simulation procedure

1. Draw a very small sample (n = 30, 45, 60 or 80) from the data set. Mimics analyzing a sample from a large population
2. Use this sample for model fitting and the validation procedure
3. Check pseudo out-of-sample performance on the rest of the data set and compare with the estimate from the internal validation
4. Compare the performance of the resampling methods in terms of bias and variance
5. Repeat simulation steps at least 6000 times (13000 in the case of training / test split)


Single Classification Tree, n = 80
in−sample
100 x Bootstrap
10 x 5−fold CV
10 x 10−fold CV
LOOCV
10 x 2−fold CV
100 x LOO−Bootstrap
1 x 75/25 split
Bias
0.3
0.2
0.1
0.0
−0.1
0.00
0.01
0.02
0.03
0.04
0.05
Variance


Single Classification Tree, n = 80
Bias
Variance
0.00
−0.01
−0.02
−0.03
−0.04
0.00
0.01
0.02
0.03
0.04
0.05
10 x 10−fold CV
LOOCV
10 x 5−fold CV
1 x 75/25 split
10 x 2−fold CV
100 x LOO−Bootstrap


Effect of more repetitions
10−fold CV, CART, n = 30
Sensitivity + Specificity − 1
−0.0012
−0.0016
−0.0020
Bias Validation
0.055
0.050
Variance Validation
2.5
5.0
7.5
10.0
Repetitions


Effect of more repetitions
5−fold CV, CART, n = 30
Sensitivity + Specificity − 1
−0.0090
−0.0095
−0.0100
−0.0105
Bias Validation
0.050
0.045
0.040
Variance Validation
2.5
5.0
7.5
10.0
Repetitions

Preliminary wrap-up

We've seen that all methods are somewhat biased and variable.
By doing more repetitions we can lower the variance.

Q: "But what's the method I should use for my task at hand?"

Preliminary wrap-up

We've seen that all methods are somewhat biased and variable. By doing more repetitions we can lower the variance.

Q: "But what's the method I should use for my task at hand?"

A:

Figure 3: Well. . .

Preliminary wrap-up

We've seen that all methods are somewhat biased and variable.
By doing more repetitions we can lower the variance.

Q: "But what's the method I should use for my task at hand?"

A2: Think of model selection vs. error estimation (is bias or variance more important?)

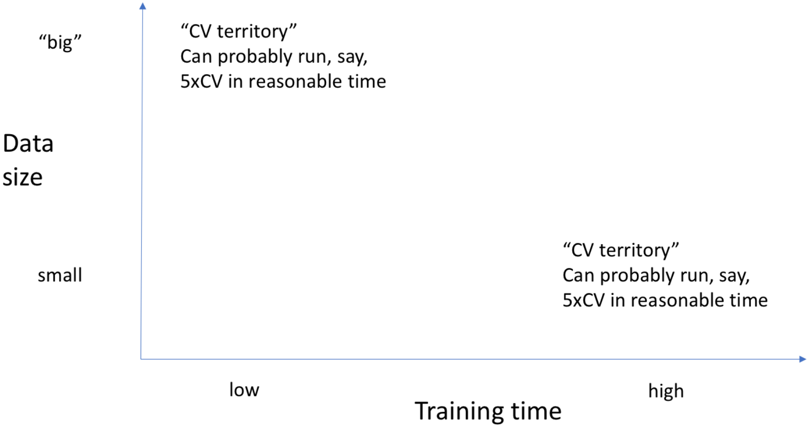
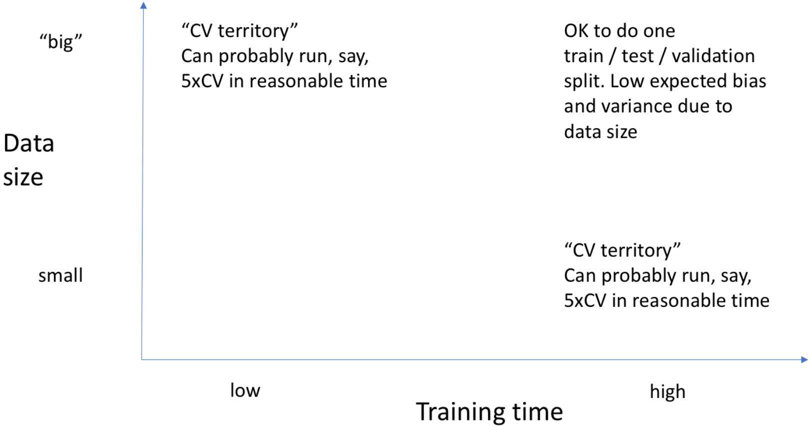
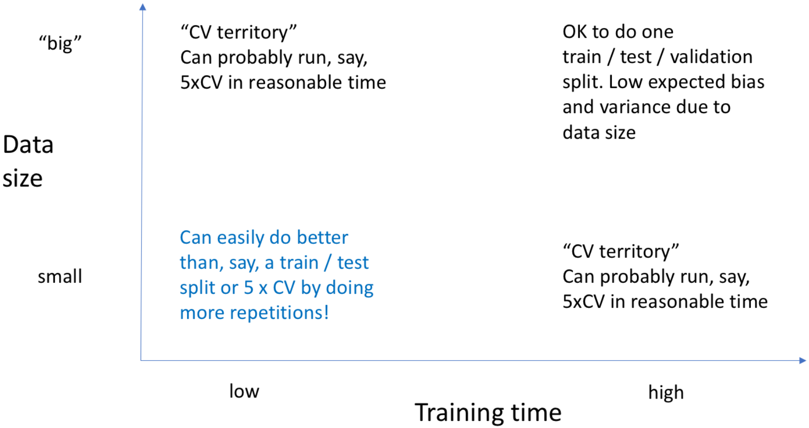

"big"
"CV territory"
Can probably run, say,
5xCV in reasonable time
Data
size
"CV territory"
Can probably run, say,
5xCV in reasonable time
small
low
high
Training time

Figure 4:



"big"
"CV territory"
Can probably run, say,
5xCV in reasonable time
OK to do one
train / test / validation
split. Low expected bias
and variance due to
data size
Data
size
"CV territory"
Can probably run, say,
5xCV in reasonable time
small
low
high
Training time

Figure 5:



"CV territory"
Can probably run, say, 5xCV in reasonable time

OK to do one train / test / validation split. Low expected bias and variance due to data size

Can easily do better than, say, a train / test split or 5 x CV by doing more repetitions!

"CV territory"
Can probably run, say, 5xCV in reasonable time

Figure 6:

Aside: Some attempts to lower the bootstrap bias

- .632: 0.632 * conventional bootstrap + (1 - 0.632) * LOO-bootstrap
- .632+: weight not 0.632 but dependent on amount of in-sample overfitting
- optimism corrected: optimism = performance on out-of-bag data - conventional bootstrap. Subtract optimism from performance on full sample.

Appendix: caret and nested CV

How can I do all this training and testing in R?

Two main options: The `caret` and the `mlr` package. We're going to quickly cover the older `caret`.

```r
library(caret)
data(iris) # Yawn...
head(iris)
```

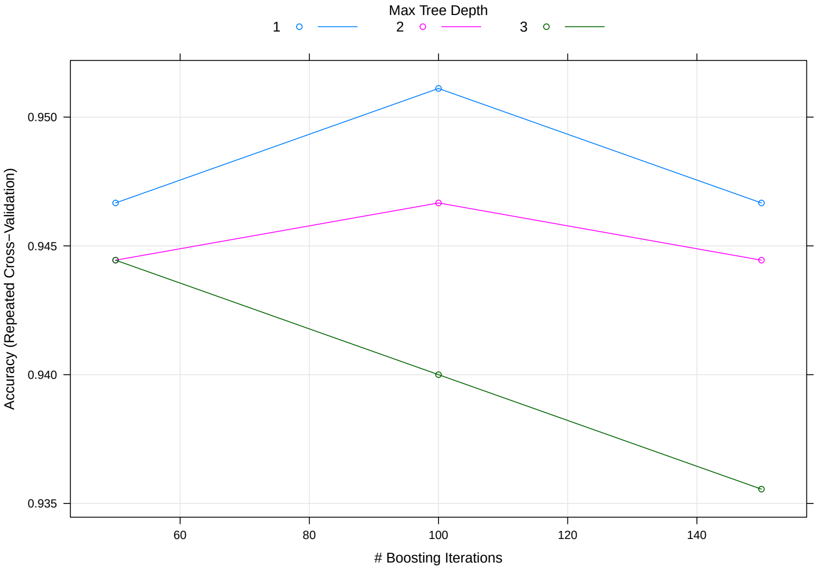
```
##   Sepal.Length Sepal.Width Petal.Length Petal.Width Species
## 1          5.1         3.5          1.4         0.2  setosa
## 2          4.9         3.0          1.4         0.2  setosa
## 3          4.7         3.2          1.3         0.2  setosa
## 4          4.6         3.1          1.5         0.2  setosa
## 5          5.0         3.6          1.4         0.2  setosa
## 6          5.4         3.9          1.7         0.4  setosa
```

```r
trcont <- trainControl(method = "repeatedcv",
                       number = 5, repeats = 3,
                       savePredictions = T,
                       verboseIter = F,
                       classProbs = T)
mod <- train(y = iris$Species, x = iris[, 1:4],
             trControl = trcont, method = "gbm")
```

```
mod
```

```
## Stochastic Gradient Boosting
##
## 150 samples
##   4 predictors
##   3 classes: 'setosa', 'versicolor', 'virginica'
##
## No pre-processing
## Resampling: Cross-Validated (5 fold, repeated 3 times)
## Summary of sample sizes: 120, 120, 120, 120, 120, 120, ...
## Resampling results across tuning parameters:
##
##   interaction.depth  n.trees  Accuracy   Kappa
##   1                   50      0.9466667  0.9200000
##   1                  100      0.9511111  0.9266667
##   1                  150      0.9466667  0.9200000
##   2                   50      0.9444444  0.9166667
##   2                  100      0.9466667  0.9200000
##   2                  150      0.9444444  0.9166667
##   3                   50      0.9444444  0.9166667
##   3                  100      0.9400000  0.9100000
```

```r
plot(mod)
```

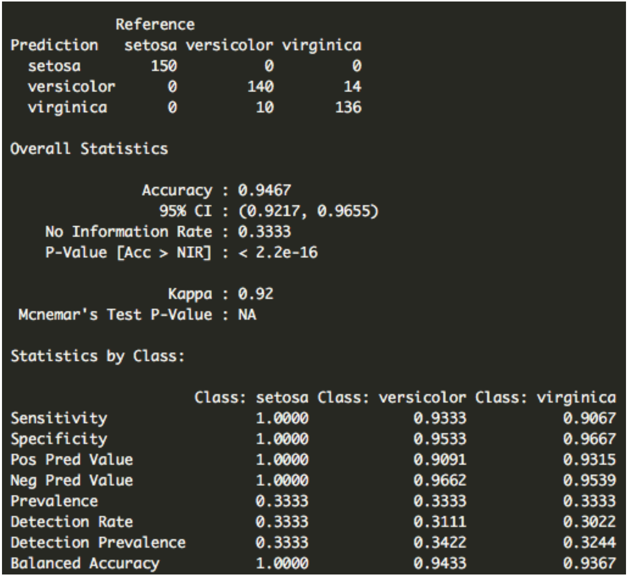


Max Tree Depth
1
2
3
0.950
0.945
0.940
0.935
Accuracy (Repeated Cross-Validation)
60
80
100
120
140
# Boosting Iterations

```r
str(mod$pred)
```

```
## 'data.frame':    4050 obs. of  11 variables:
##  $ pred             : Factor w/ 3 levels "setosa","versicolor
##  $ obs              : Factor w/ 3 levels "setosa","versicolor
##  $ rowIndex         : int  12 16 17 19 22 23 24 30 34 50 ...
##  $ setosa           : num  1 1 1 1 1 ...
##  $ versicolor       : num  5.73e-07 2.94e-06 2.11e-06 3.45e-0
##  $ virginica        : num  3.58e-10 1.05e-10 1.05e-10 1.23e-1
##  $ shrinkage        : num  0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.
##  $ interaction.depth: int  1 1 1 1 1 1 1 1 1 1 ...
##  $ n.minobsinnode   : num  10 10 10 10 10 10 10 10 10 10 ...
##  $ n.trees          : num  150 150 150 150 150 150 150 150 15
##  $ Resample         : chr  "Fold1.Rep1" "Fold1.Rep1" "Fold1.R
```

```r
# Select the "best" parameter set
resamples <- mod$pred %>%
    dplyr::filter(shrinkage == 0.1,
                  n.minobsinnode == 10,
                  n.trees == 100,
                  interaction.depth == 1)
dim(iris)
dim(resamples) # 450 rows because 3 repeats of CV
confusionMatrix(reference = resamples$obs,
                data = resamples$pred)
```



```
            Reference
Prediction   setosa versicolor virginica
  setosa        150          0         0
  versicolor      0        140        14
  virginica       0         10       136

Overall Statistics

               Accuracy : 0.9467
                 95% CI : (0.9217, 0.9655)
    No Information Rate : 0.3333
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.92
 Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: setosa Class: versicolor Class: virginica
Sensitivity                 1.0000            0.9333           0.9067
Specificity                 1.0000            0.9533           0.9667
Pos Pred Value              1.0000            0.9091           0.9315
Neg Pred Value              1.0000            0.9662           0.9539
Prevalence                  0.3333            0.3333           0.3333
Detection Rate              0.3333            0.3111           0.3022
Detection Prevalence        0.3333            0.3422           0.3244
Balanced Accuracy           1.0000            0.9433           0.9367
```

Figure 7: Confusion matrix using the out-of-fold data

Repeated CV is no nested CV!

Let's say we want to get an estimate of our model's performance after selecting the best parameter set or the best single model. We need an additional outer CV loop:

```r
# Outer Loop
inTrain <- createFolds(iris$Species, k = 5,
                       returnTrain = T)

# I hear the cool kids are using purrr now...
nested_cv <- sapply(inTrain, function(train_indices) {
    train_temp <- iris[train_indices, ]
    test_temp  <- iris[-train_indices, ]
    # If nothing is specified, caret trains a Random Forest
    # and evaluates it using Leave-one-out bootstrapping.
    # We're doing selection here, not error estimation,
    # after all. Then picks "best" parameter set.
    mod <- train(x = train_temp[, 1:4],
                 y = train_temp$Species)
    preds_temp <- predict(mod, test_temp)
    # Let's calculate accuracy, could be any other metric
    sum(test_temp$Species == preds_temp) / nrow(test_temp)
})
summary(nested_cv) # Now only "normal" bias/variance left
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.8667  0.9333  0.9667  0.9533  1.0000  1.0000
```

What we did not cover

- calibration
- CV for time series
- t-tests for comparing validation results
- . . .

End of appendix

. . . and of my presentation

Thank you!

References

caret tutorial

- http://topepo.github.io/caret/index.html

Alternative: mlr

- https://mlr-org.github.io/mlr-tutorial/

Arlot, S., & Celisse, A. (2010). A survey of cross-validation procedures for model selection. Statistics Surveys, 4, 40–79.

- http://www.di.ens.fr/sierra/pdfs/2010_Arlot_Celisse_SS.pdf

Hastie, Tibshirani, Friedman. The elements of statistical learning

- http://statweb.stanford.edu/~tibs/ElemStatLearn/printings/ESLII_print10.pdf

Yu, W., Ruibo, W., Huichen, J., & Jihong, L. (2014). Blocked 3x2 Cross-Validated t-Test for Comparing Supervised Classification Learning Algorithms. Neural Computation, 26(1), 208–235.

- https://doi.org/10.1162/NECO_a_00532

Optimism of the training error

- http://eranraviv.com/optimism-training-error-rate/

Altman, D. G., & Royston, P. (n.d.). What do we mean by validating a prognostic model?

- http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.610.4279

Bergmeir, C., Hyndman, R. J., & Koo, B. (2015). A Note on the Validity of Cross-Validation for Evaluating Time Series Prediction.

- https://business.monash.edu/econometrics-and-business-statistics/research/publications/ebs/wp10-15.pdf

Krstajic, D., Buturovic, L. J., Leahy, D. E., & Thomas, S. (2014). Cross-validation pitfalls when selecting and assessing regression and classification models. Journal of Cheminformatics, 6(1), 1.

Steyerberg, E. W., Bleeker, S. E., Moll, H. A., Grobbee, D. E., & Moons, K. G. (2003). Internal and external validation of predictive models: A simulation study of bias and precision in small samples. Journal of Clinical Epidemiology, 56, 441–447.