Random forest

If there is a strong predictor in the dataset, the decision trees produced by each of the bootstrap samples in the bagging algorithm becomes very similar: Most of the trees will use the same strong predictor in the top split.

Random forests is a solution to this problem and a method for decorrelating the trees. The hope is to improve the variance reduction.

The effect of correlation on the variance of the mean

The variance of the average of B observations of i.i.d random variables X, each with variance σ^2 is $\frac{\sigma^2}{B}$. Now, suppose we have B observations of a random variable X which are identically distributed, each with mean μ and variance σ^2, but not independent.

That is, suppose the variables have a positive correlation ρ

$$\text{Cov}(X_i, X_j) = \rho\sigma^2, \quad i \neq j.$$

This is called compound symmetry.

Then the variance of the average is

$$\rho\sigma^2 + \frac{1-\rho}{B}\sigma^2$$

(Exercise to prove this.)

Check: $\rho = 0$ and $\rho = 1$? Observe the linearity in ρ.

(Most negative values of ρ will not give a positive definite covariance matrix. The covariance matrix is positive definite if $\rho > -1/(B-1)$.)

Core modifications to bagging

The idea behind random forest is to *improve the variance reduction of bagging* by reducing the correlation between the trees - while hoping the possible increase in variance in each tree doesn´t cancel the improvement.

The procedure is thus as in bagging, but with the important difference, that

- at each split we are only allowed to consider $m < p$ of the predictors.

A new sample of m predictors is taken at each split and

- typically $m = \text{floor}(\sqrt{p})$ (classificaton) and $m = \text{floor}(p/3)$ (regression)

The general idea is at for very correlated predictors m is chosen to be small.

Random forest algorithm

(We write out in class.)

- Regression: average of trees
- Classification: majority vote based on vote from each tree.

Study ELS Figure 15.3 to see that the rule of thumb for m may not always be the best choice.

Study ELS Figure 15.9 to se that the correlation between trees is dependent on m.

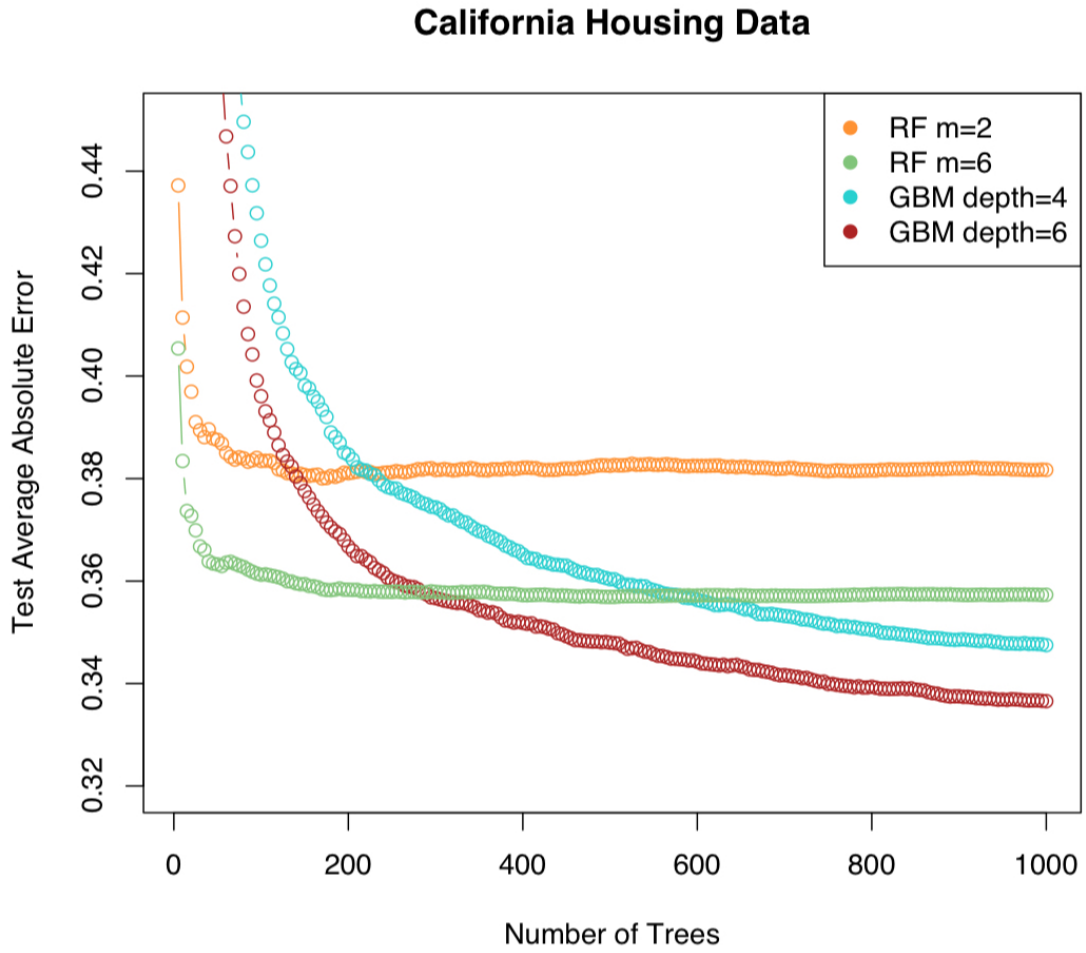


FIGURE 15.3. *Random forests compared to gradient boosting on the California housing data. The curves represent mean absolute error on the test data as a function of the number of trees in the models. Two random forests are shown, with* $m = 2$ *and* $m = 6$*. The two gradient boosted models use a shrinkage parameter* $\nu = 0.05$ *in (10.41), and have interaction depths of* 4 *and* 6*. The boosted models outperform random forests.*

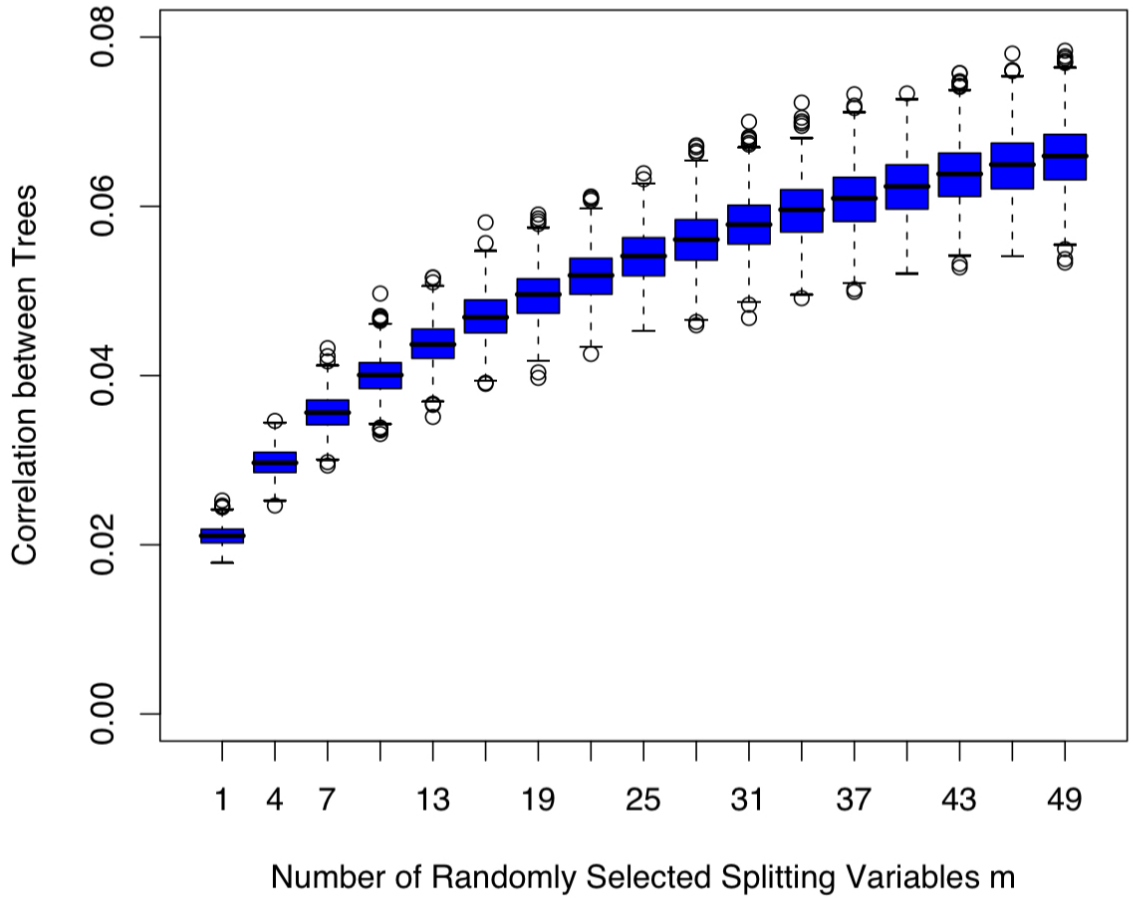


FIGURE 15.9. *Correlations between pairs of trees drawn by a random-forest regression algorithm, as a function of* m*. The boxplots represent the correlations at* 600 *randomly chosen prediction points* x*.*

In addition the recommendations from the Random forest authors were also on *node size* (the minimum number of observations in a leaf node):

- classification: 1
- regression: 5

(ELS page 592)

This is an indication that node size is an hyperparameter, but ESL argue that is is maybe not worth the extra effort to optimize on this parameter.

Study ELS Figure 15.8 for effect of node size.

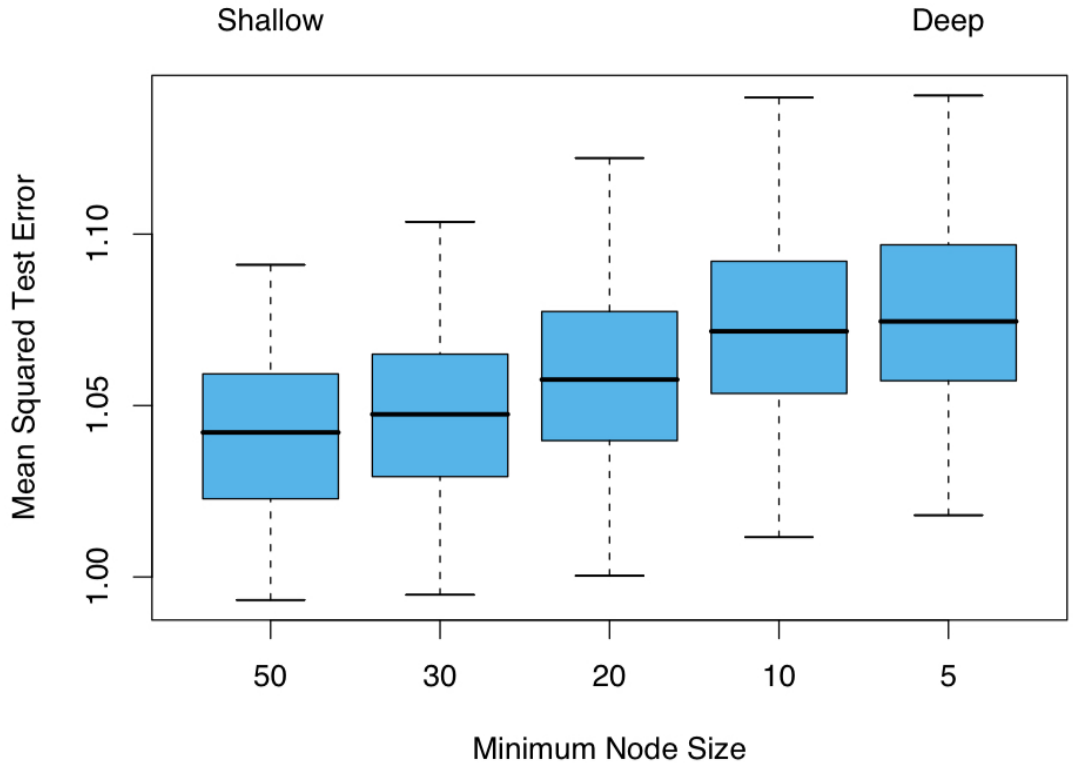


FIGURE 15.8. *The effect of* tree size *on the error in random forest regression. In this example, the true surface was additive in two of the* 12 *variables, plus additive unit-variance Gaussian noise. Tree depth is controlled here by the minimum node size; the smaller the minimum node size, the deeper the trees.*

The number of trees, B, is not a tuning parameter (according to the ISLR-authors), and the best is to choose it large enough.

But - in lecture L7 we will hear that we may also look at many of the choices for how to fit a tree as model hyperparameters - in addition to both B and m.

Study Figure 15.1 in ELS.

We will look at comparing error rates (using statistical tests) for different methods later in Part 2.

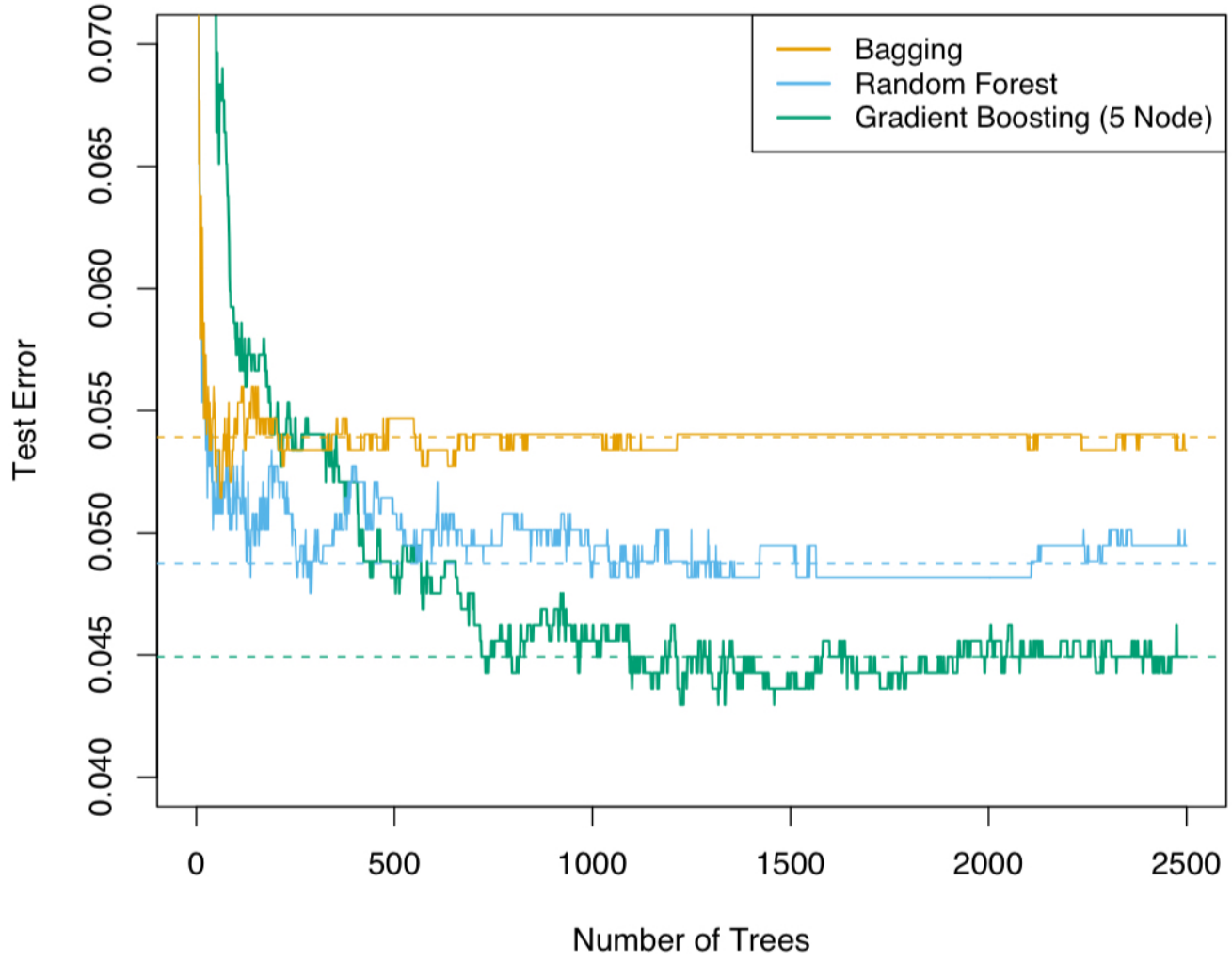


FIGURE 15.1. *Bagging, random forest, and gradient boosting, applied to the spam data. For boosting,* 5*-node trees were used, and the number of trees were chosen by* 10*-fold cross-validation (*2500 *trees). Each "step" in the figure corresponds to a change in a single misclassification (in a test set of* 1536*).*

OOB again

When the OOB error stabilizes the B is large enough and we may stop training.

Study Figure 15.4 in ELS.

If B is sufficiently large (three times the number needed for the random forest to stabilize), the OOB error estimate is equivalent to LOOCV (CASI: Efron and Hastie, 2016, p 330).

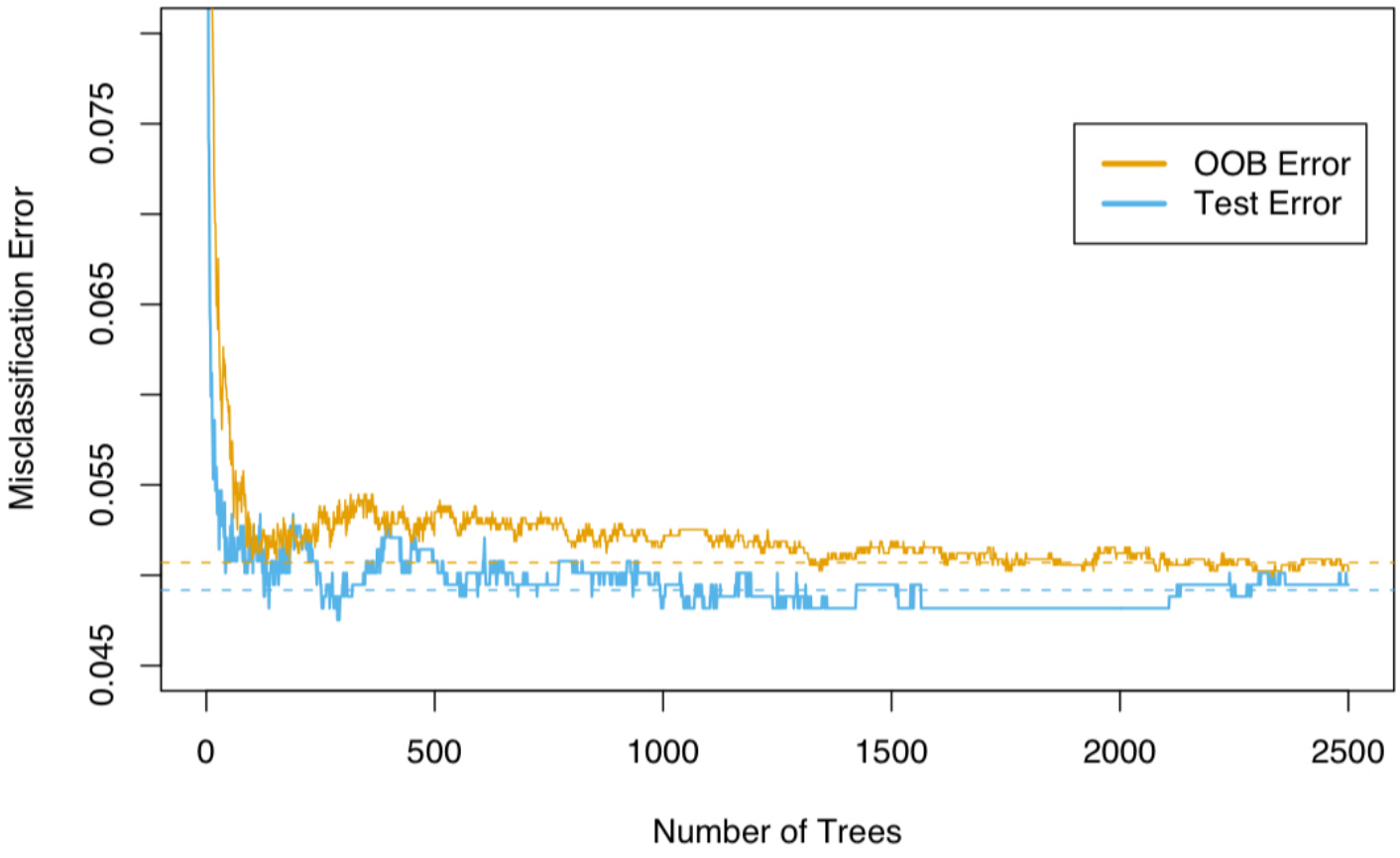


FIGURE 15.4. OOB *error computed on the* `spam` *training data, compared to the test error computed on the test set.*

Variable importance plots

Bagging is an example of an *ensemble method*, so is boosting and random forests (to come next). For all of these methods many trees are grown and combined, and the predictive power can be highly improved. However, this comes at a cost of interpretability. Instead of having one tree, the resulting model consists of B trees, where B often is 300 or 500 (or maybe even 5000 when boosting). Variable importance plots show *the relative importance of the predictors:* the predictors are sorted according to their importance, such that the top variables have a higher importance than the bottom variables. There are in general two types of variable importance plots:

- variable importance based on decrease in node impurity and
- variable importance based on randomization.

Variable importance for a single tree

(10.13.1)

A single tree can be studied to interpret the model fitted. For large trees - and in the coming chapters - for many trees, the concept of *variable importance* is useful.

Consider a covariate. How important is this covariate for the tree prediction?

We have a tree T with $J-1$ *internal nodes* (remark: no leaf nodes - because there is no split at a leaf node).

Let $I_l^2(T)$ be a measure of squared relevance for predictor X_l:

$$I_l^2(T) = \sum_{t=1}^{J-1} \hat{i}_t^2 I(v(t) = l)$$

At each internal node t there is a split, where the covariate to split on is denoted $X_{v(t)}$, and this variable was the one that gave the maximal improvement $\hat{i}_t^2$.

The importance measure is the square root, so $I_l(T) = \sqrt{I_l^2(T)}$.

The term *important* relates to *total decrease in the node impurity, over splits for a predictor*, and is defined differently for regression trees and classification trees.

Regression trees:

- The importance of each predictor is calculated using the MSE.
- The algorithm records the total amount that the MSE is decreased due to splits for each predictor (there may be many spits for one predictor for each tree).
- This decrease in MSE is then averaged over the B trees. The higher the decrease, the more important the predictor.

Classification trees:

- The importance of each predictor is calculated using the Gini index.
- The importance is the mean decrease (over all B trees) in the Gini index by splits of a predictor.

From single to many trees

$$I_l^2 = \frac{1}{B} \sum_{b=1}^{B} I_l^2(T_b)$$

The measure is relative:

- the highest value is set to 100
- the others are scaled according to this

R: `varImpPlot` (or `importance`) in `randomForest` with `type=2`.

Variable importance based on randomization

Variable importance based on randomization is calculated using the OOB sample.

- Computations are carried out for one bootstrap sample at a time.
- Each time a tree is grown the OOB sample is used to test the predictive power of the tree.
- Then for one predictor at a time, repeat the following:
  - permute the OOB observations for the jth variable x_j and calculate the new OOB error.
  - If x_j is important, permuting its observations will decrease the predictive performance.
- The difference between the two is averaged over all trees
- and again highest set to 100, others rescaled.

R: `varImpPlot` (or `importance`) in `randomForest` with `type=1`.

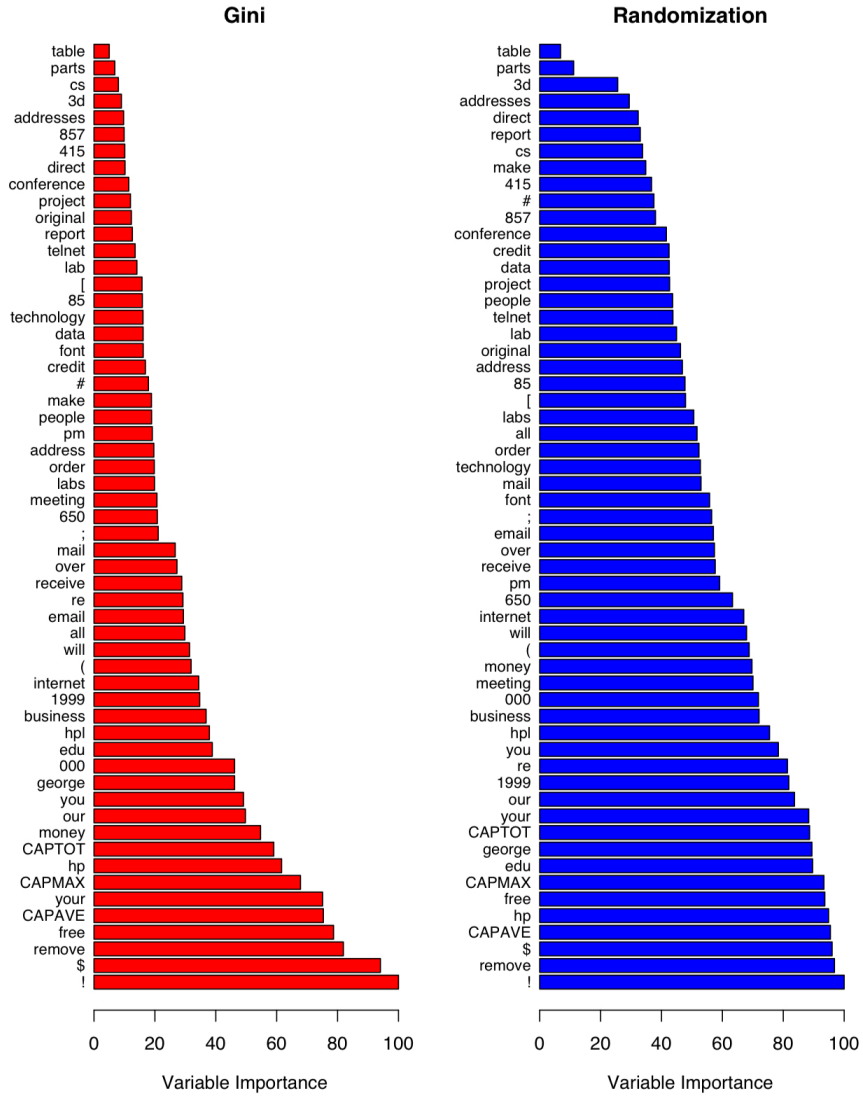


FIGURE 15.5. *Variable importance plots for a classification random forest grown on the* `spam` *data. The left plot bases the importance on the Gini splitting index, as in gradient boosting. The rankings compare well with the rankings produced by gradient boosting (Figure 10.6 on page 354). The right plot uses* OOB *randomization to compute variable importances, and tends to spread the importances more uniformly.*

Variable importance based on randomization

Variable importance based on randomization is calculated using the OOB sample.

- Computations are carried out for one bootstrap sample at a time.
- Each time a tree is grown the OOB sample is used to test the predictive power of the tree.
- Then for one predictor at a time, repeat the following:
  - permute the OOB observations for the jth variable x_j and calculate the new OOB error.
  - If x_j is important, permuting its observations will decrease the predictive performance.
- The difference between the two is averaged over all trees
- and again highest set to 100, others rescaled.

R: `varImpPlot` (or `importance`) in `randomForest` with `type=1`.

Study Figure 15.5 in ELS.

Boston

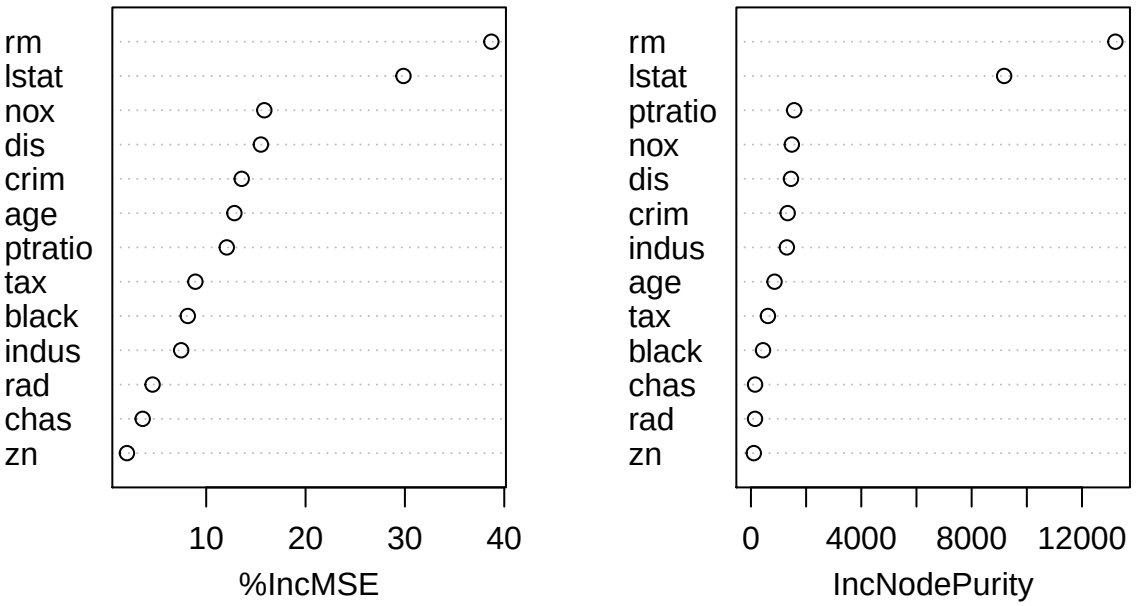
```
set.seed(1)
rf.boston=randomForest(medv~.,data=Boston,subset=train,mtry=6,importance=TRUE)
yhat.rf = predict(rf.boston,newdata=Boston[-train,])
mean((yhat.rf-boston.test)^2)
```

```
## [1] 15.77329
```

```
importance(rf.boston)
```

```
##           %IncMSE IncNodePurity
## crim    13.571040     1331.4214
## zn       2.018357      103.3764
## indus    7.478237     1301.3529
## chas     3.604777      150.6007
## nox     15.847850     1481.9064
## rm      38.703015    13209.6852
## age     12.837457      856.1236
## dis     15.505816     1450.6934
## rad      4.600793      147.7769
## tax      8.910426      615.1269
## ptratio 12.069248     1566.8163
## black    8.144727      438.1747
## lstat   29.854464     9177.8663
```

```
varImpPlot(rf.boston)
```



Pima indians

We decorrelate the trees by using the *randomForest()* function again, but this time we set *mtry=3*. This means that the algorithm only considers three of the predictors in each split. We choose 3 because we have 10 predictors in total and $\sqrt{10} \approx 3$.

```
set.seed(1)
rf=randomForest(factor(diabetes)~npreg+glu+bp+skin+bmi+ped+age,data=ctrain,mtry=3,importance=TRUE) #def
rf
```

```
##
## Call:
##  randomForest(formula = factor(diabetes) ~ npreg + glu + bp +      skin + bmi + ped + age, data = ct
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 3
##
##         OOB estimate of  error rate: 22%
## Confusion matrix:
##     0  1 class.error
## 0 172 28        0.14
## 1  38 62        0.38
```

```
test.x=ctest[,-1]
test.y=ctest[,1]
train.y=ctrain[,1]
train.x=ctrain[,-1]
```

```r
train.res=predict(rf,type="prob")[,2]
test.res=predict(rf,newdata=test.x,type="prob")[,2]
train.class=ifelse(train.res>=0.5,1,0)
#train.class2=predict(rf,type="response") #same as train.class
test.class=ifelse(test.res>=0.5,1,0)
print("Evaluation on training data")
```

```
## [1] "Evaluation on training data"
```

```r
confusionMatrix(factor(train.class),factor(train.y))$overall[1]
```

```
##  Accuracy
## 0.7733333
```

```r
print("Evaluation on test data")
```

```
## [1] "Evaluation on test data"
```

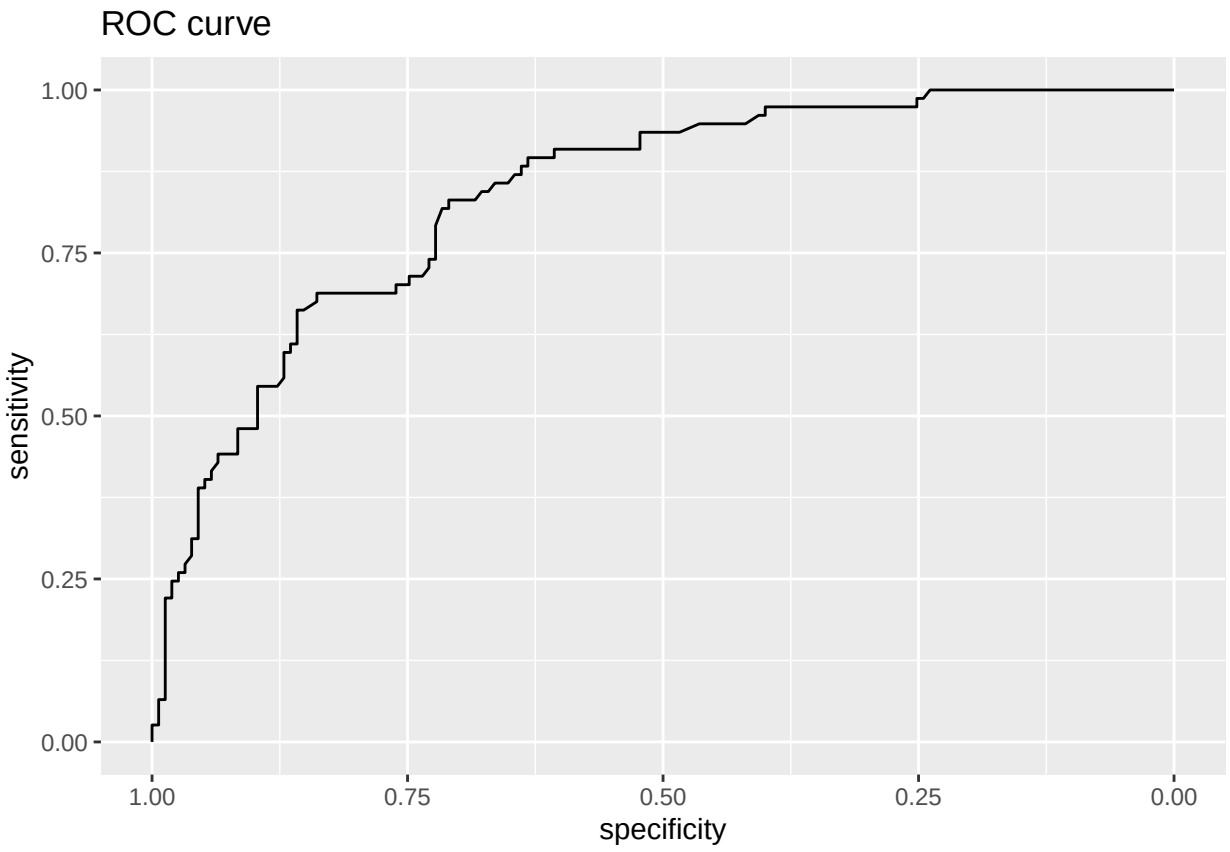
```r
confusionMatrix(factor(test.class),factor(test.y))$overall[1]
```

```
##  Accuracy
## 0.7801724
```

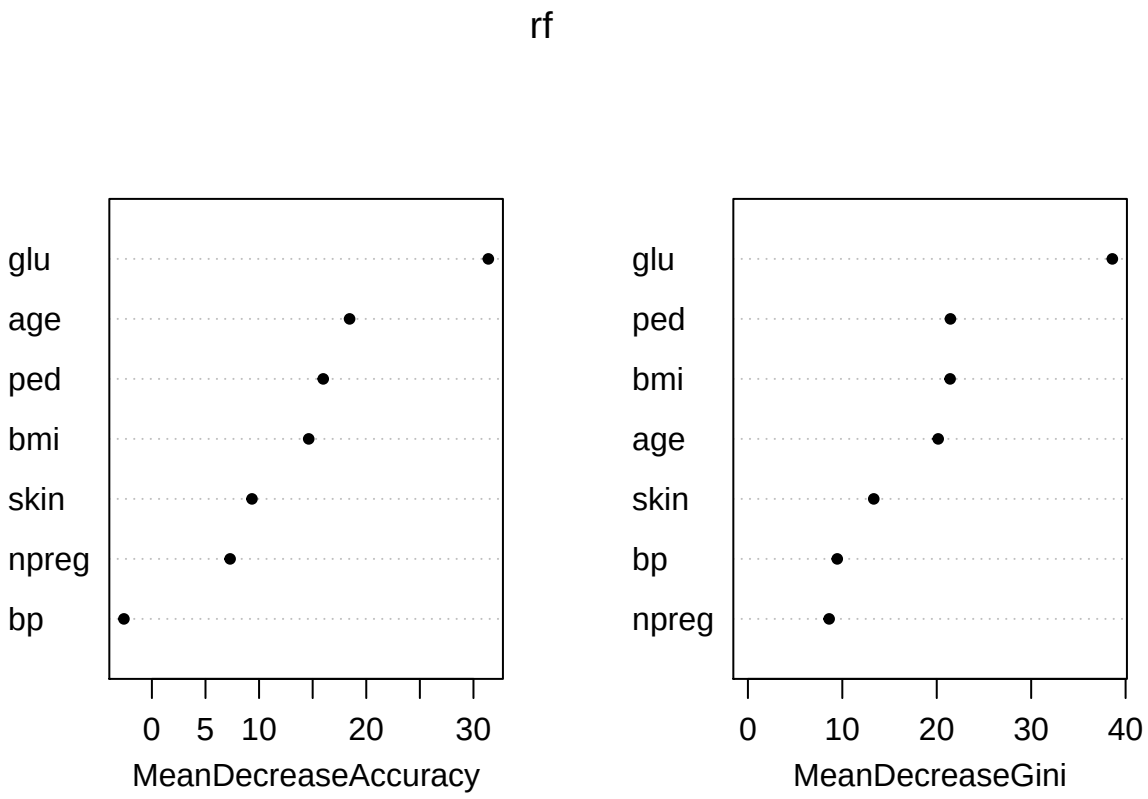
```r
roc.rf = roc(test.y,test.res,legacy.axes=TRUE)
print(auc(roc.rf))
```

```
## Area under the curve: 0.8379
```

```r
ggroc(roc.rf)+ggtitle("ROC curve")
```



```
varImpPlot(rf,pch=20)
```




```
#varImpPlot(rf,pch=20,type=1)
#varImpPlot(rf,pch=20,type=2)
```

Forward - boosting next

Study ELS Figure 15.7 for comparing random forest with boosting as a function of relevant variables.

When the number of relevant predictors * is high, random forest performs well. * is small, random forest performance deteriorate with many noisy variables

Conclusions

Exercises

Small tasks

Look through the many problems sets presented in this document. (Solutions provided for some of the problems.)

Prove the formula

for the variance of the mean with compound symmetry correlation

$$\rho\sigma^2 + \frac{1-\rho}{B}\sigma^2$$

This is also Exercise 15.1 in ELS. Link to solutions

Conclusions

Exercises

Small tasks

Look through the many problems sets presented in this document. (Solutions provided for some of the problems.)

Prove the formula

for the variance of the mean with compound symmetry correlation

$$\rho\sigma^2 + \frac{1-\rho}{B}\sigma^2$$

This is also Exercise 15.1 in ELS. Link to solutions

References

- ISLR: An introduction to statistical learning (2013). James, Gareth and Witten, Daniela and Hastie, Trevor and Tibshirani, Robert. Springer.
- Ripley (1996): Pattern Recognicion and Neural Networks, Cambridge University Press.
- Breiman (1996): Bagging Predictors, Machine Learning, 24, 123-140.
- Breiman (2001): Random Forest, Machine Learning, 45, 5-32.
- Stef van Buuren. "Flexible imputation of missing data", Section 1.2 https://stefvanbuuren.name/fimd/sec-MCAR.html
- CASI: Computer age statistical inference - algorithms, evidence, and data science (2016). Bradley Efron and Trevor Hastie. Cambridge University Press.