Bagging, Random Forests

UBCO MDS — DATA 571

Motivation

- We've just wrapped up some discussion on decisions trees, or CARTs which suggests that they are not particularly 'good' models in many scenarios
- Now we will discuss three computational ways (split into two lectures) to address their primary flaws
- But as with all model improvements, we will have to accept some drawbacks

Bagging

- Bagging (bootstrap-aggregating) can be thought of as applying the bootstrap to an entire model-fitting process, rather than just to estimate standard errors of estimators.

- By generating B bootstrapped samples, we can train B models $\hat{f}^{*b}$ and then average all of their predictions

$$\hat{f}_{bag}(x) = \frac{\sum_{b=1}^{B} f^{*b}(x_b^*)}{B}$$

- Because of the pitfalls of CARTs mentioned earlier, this is a common strategy to improve CART predictions and stability. Textbook suggests $B = 100$ as generally sufficient, but probably 500+ is safer.

"Out of Bag"

- When bootstrapping (sampling n observations with replacement), some observations are randomly left out while others are duplicated.
- It is relatively straightforward to compute that for each bootstrap sample, on average roughly $\frac{1}{3}$ of the observations will not appear. Discuss!
- Observations left out of a model fit are called out of bag or OOB observations. Note the similarity to cross-validation.
- This leads us to an interesting option for bootstrap-aggregated models...

OOB Estimation

- Even though bagging is fairly robust to overfitting (since it is already averaging across many varied model fits), we can report error using the observed data only through models fit with the observation was OOB

- In order to avoid nasty notation, let's just suppose we fit a bagged model with B=10, and found that x_1 was OOB for the set $S = \{3, 5, 6\}$

- In this case, the OOB estimate

$$\hat{y}_1 = \frac{\sum_{b \in S} f^{*b}(x_1)}{3}$$

- And we can estimate long-term model error using all of the OOB estimates!

Bagging

- Major Con: Bagging loses the interpretability of our model! We no longer have one model, but an average of many models! Prediction vs Inference tradeoff...
- Major Pro: We can measure variable importance by averaging the decrease in error (RSS or Gini) found by splitting on each predictor across all B models.

Example: Beer Regression

- Recall the beer data we discussed last lecture

```
> library(DAAG)
> beercv <- cv.lm(data=beerdat, beerlm, m=10)
> sum((beercv$price-beercv$cvpred)^2) #10CV RSS for lm
[1] 54.6
> min(cv.beert$dev) #10CV RSS for 6 terminal tree
[1] 74.3
```

- Above are the cross-validated squared errors.

Example: Bagging Regression Tree

```
> library(randomForest)
> set.seed(134891)
> beerbag <- randomForest(price~., data=beerdat, mtry=5, impo
> beerbag

Call:
 randomForest(formula = price ~ ., data = beerdat, mtry = 5,
               Type of random forest: regression
                     Number of trees: 500
No. of variables tried at each split: 5

          Mean of squared residuals: 0.911
                    % Var explained: 55.8
```

- So OOB RSS would be 0.911 * 69 = 62.9, an improvement over the tree, but not over LMs!

Error Comments

- It's worth noting that comparison we just made...
- Technically, we have not performed cross validation on our bagged trees, and yet we are comparing the (OOB) error to properly cross-validated errors for LM and CART
- BUT, again, OOB estimation is a similar (and more efficient, when applicable) way for providing realistic long term error estimates
- Takeaway: For practical purposes, OOB error from bagged models can be compared to CV error from other models.

Variable Importance

- The package that performs bagging provides variable importance plots (if requested).
- Two measures are provided as a proxy for variable 'importance'. They may not completely agree
- Both are worth further explanation...

%IncMSE/MeanDecreaseAccuracy

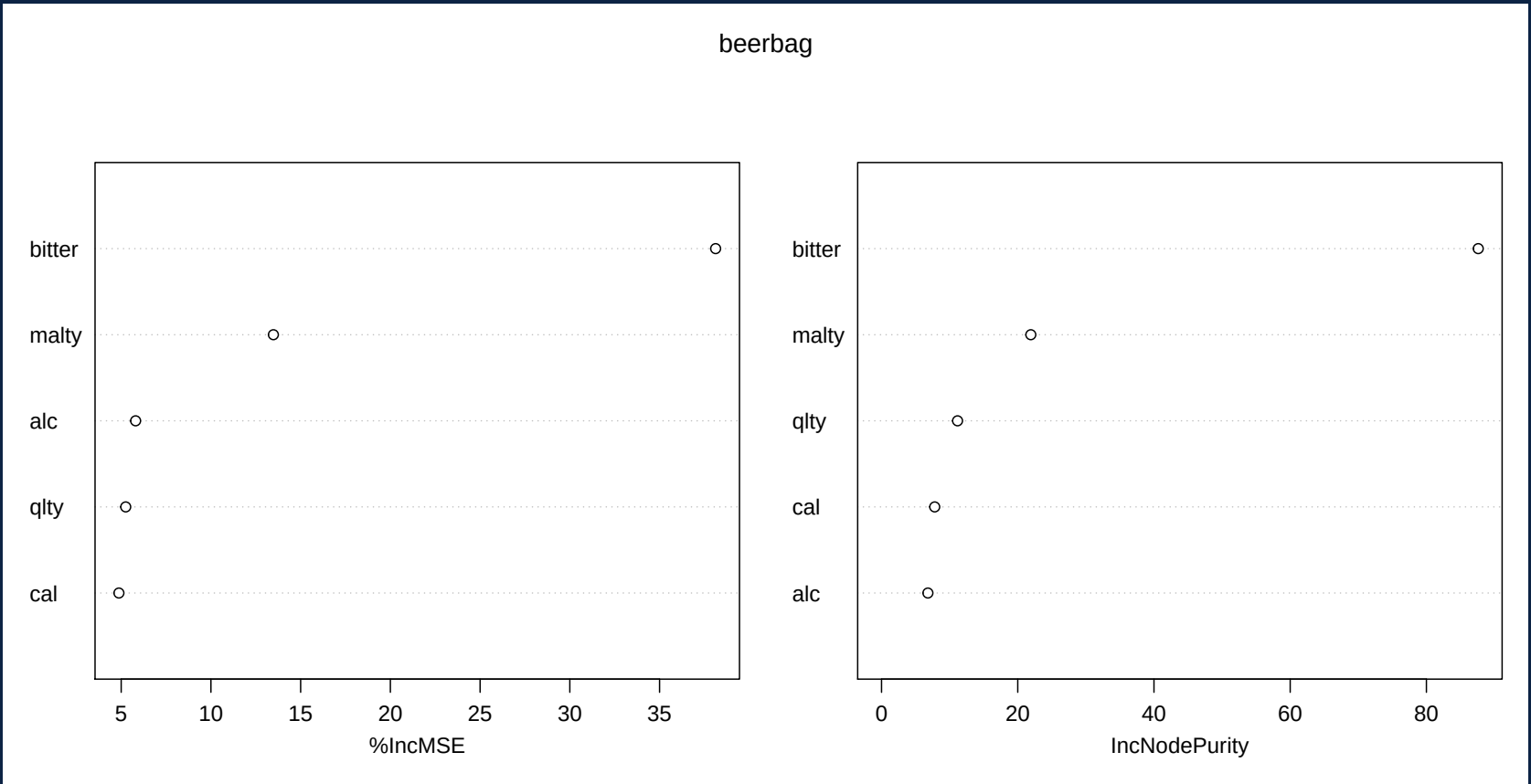
- Percent increase of MSE or average decrease in classification accuracy
  1. Calculate MSE/Misclass for the full fit of the random forest (say, MSEfull or Misfull)
  2. For the variable in question, randomly shuffle the observations (leave the order correct for the other columns).
  3. Carry out predictions, calculate MSEshuff/Misshuff then check the change from the full version for the variable in question: $\frac{MSEshuff - MSEfull}{MSEfull}$ or $Misshuff - Misfull$.

- Higher indicates that the variable is more important

IncNodePurity

- "IncNodePurity" — increase in node purity
  - Increase in node purity found by splitting on the variable, averaged across all trees in the forest
  - For regression, purity measured by RSS
  - For classification, purity measured by Gini index
- Higher increase indicates that the variable is more important

Example: Bagging Regression Tree



Random Forests

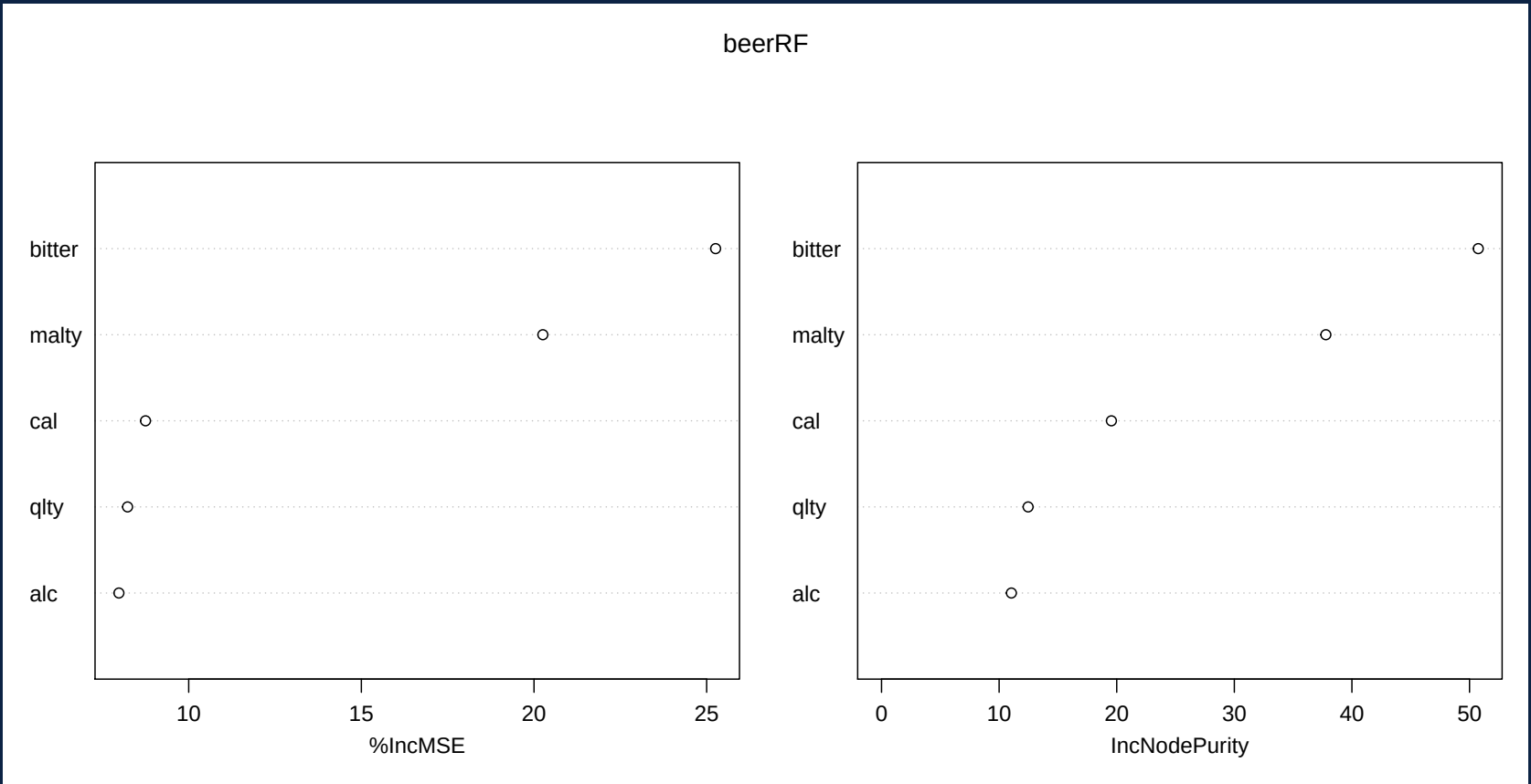
- Random forests take bagging a step further.
- During tree growth for all B samples, at each split a number of the predictors (default different for classification vs regression) are removed randomly from consideration.
- This effectively decreases the dependency (or correlation) across the B trees, decreasing the variance of the averaged model (think back to LOOCV) and therefore increasing stability.
- Still lose interpretability over simple trees, but still gain the variable importance measure.

Example: Random Forests

```
> set.seed(134891)
> beerRF <- randomForest(price~., data=beerdat, mtry=2, impor
> beerRF

Call:
 randomForest(formula = price ~ ., data = beerdat, mtry = 2,
               Type of random forest: regression
                     Number of trees: 500
No. of variables tried at each split: 2

          Mean of squared residuals: 0.854
                    % Var explained: 58.6
```

- Note mtry=2 (approx sqrt(5)). OOB RSS is .854 * 69 = 58.9 — better than trees and bagging, still not better than LMs!

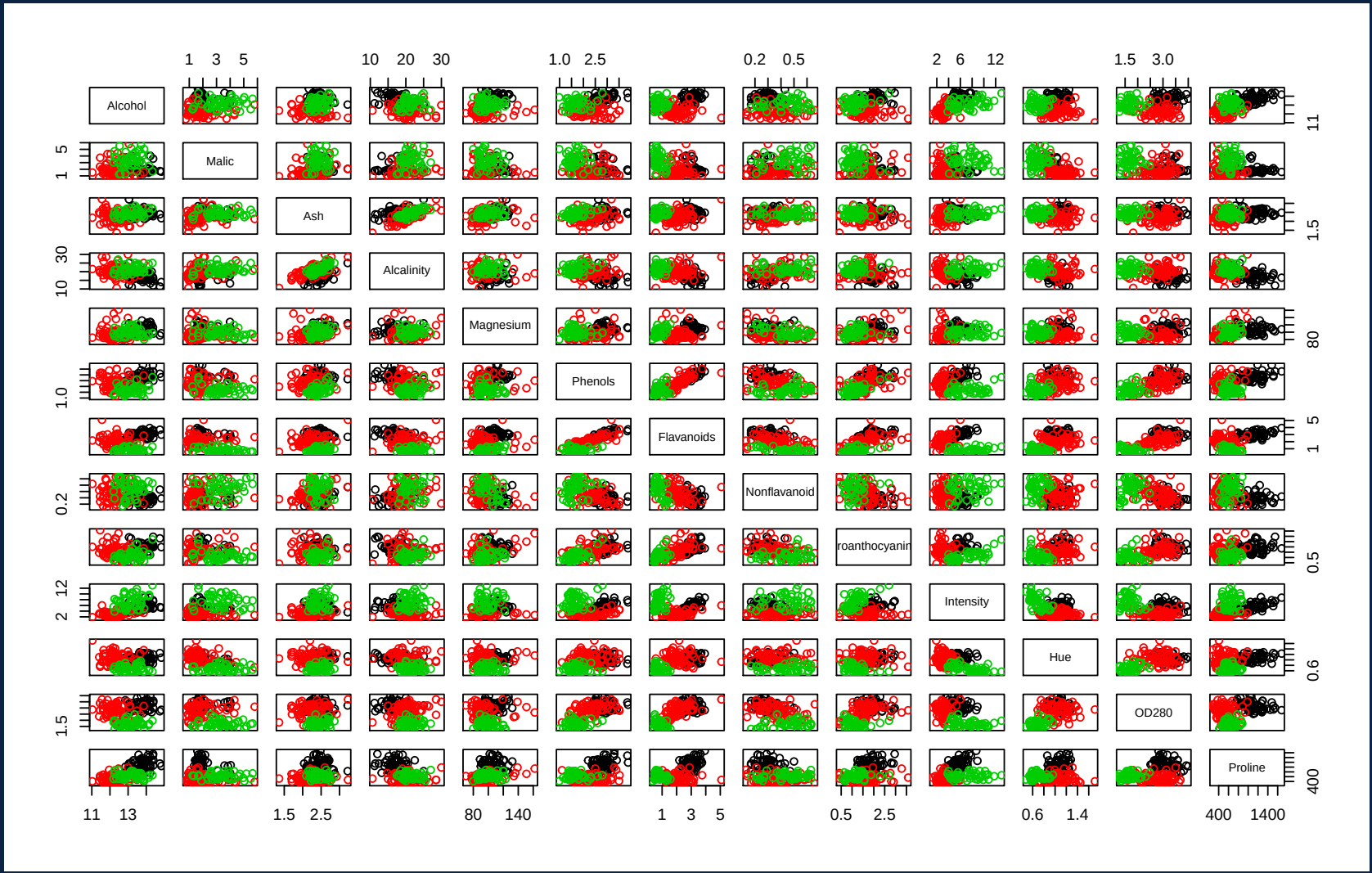
Example: Random Forests



Classification Example

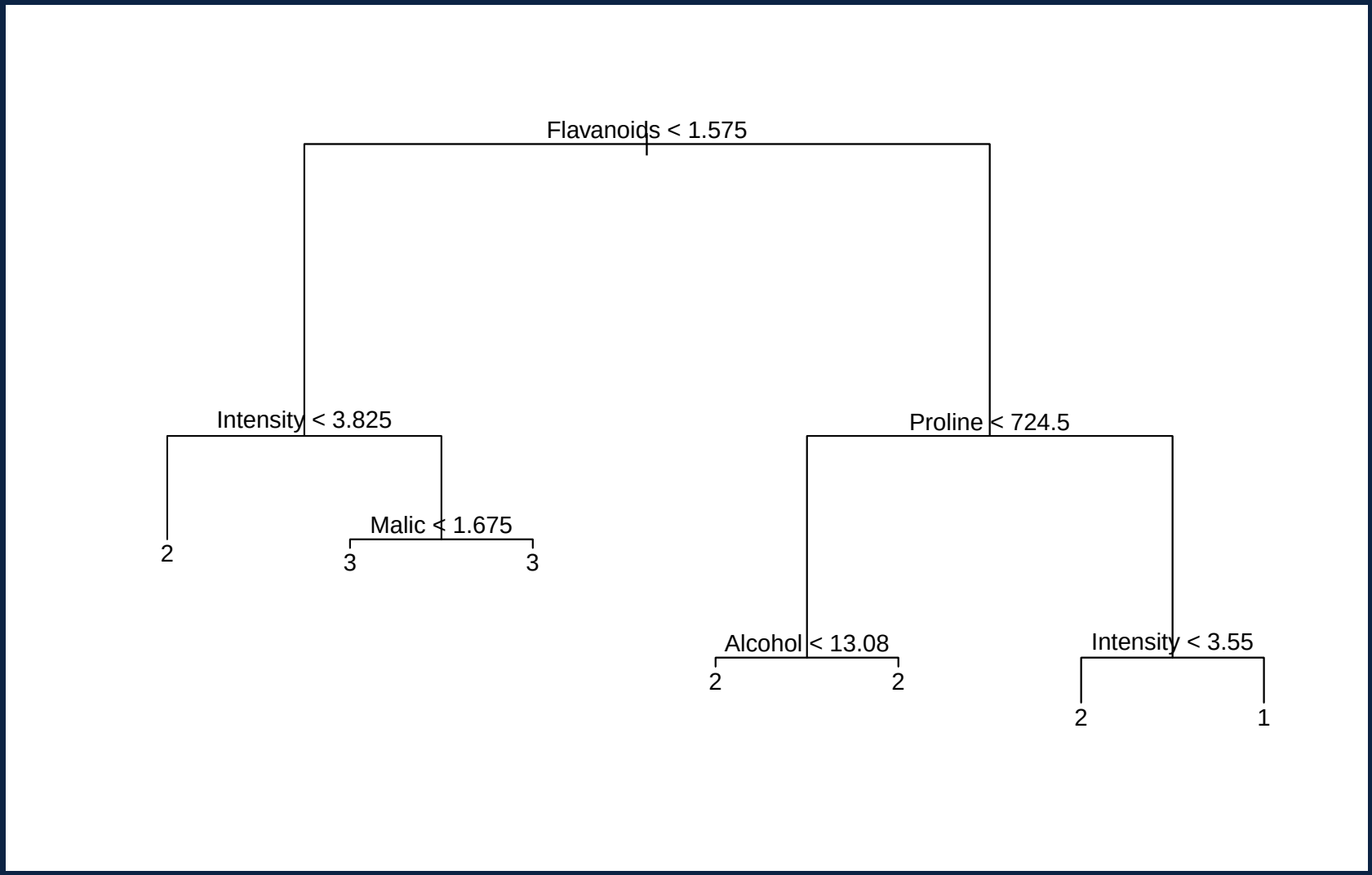
- The smaller (more common) Italian red wine data set
- 178 samples of wine from three varietals grown in the Piedmont region of Italy
- 13 chemical measurements...

Example: Wine data



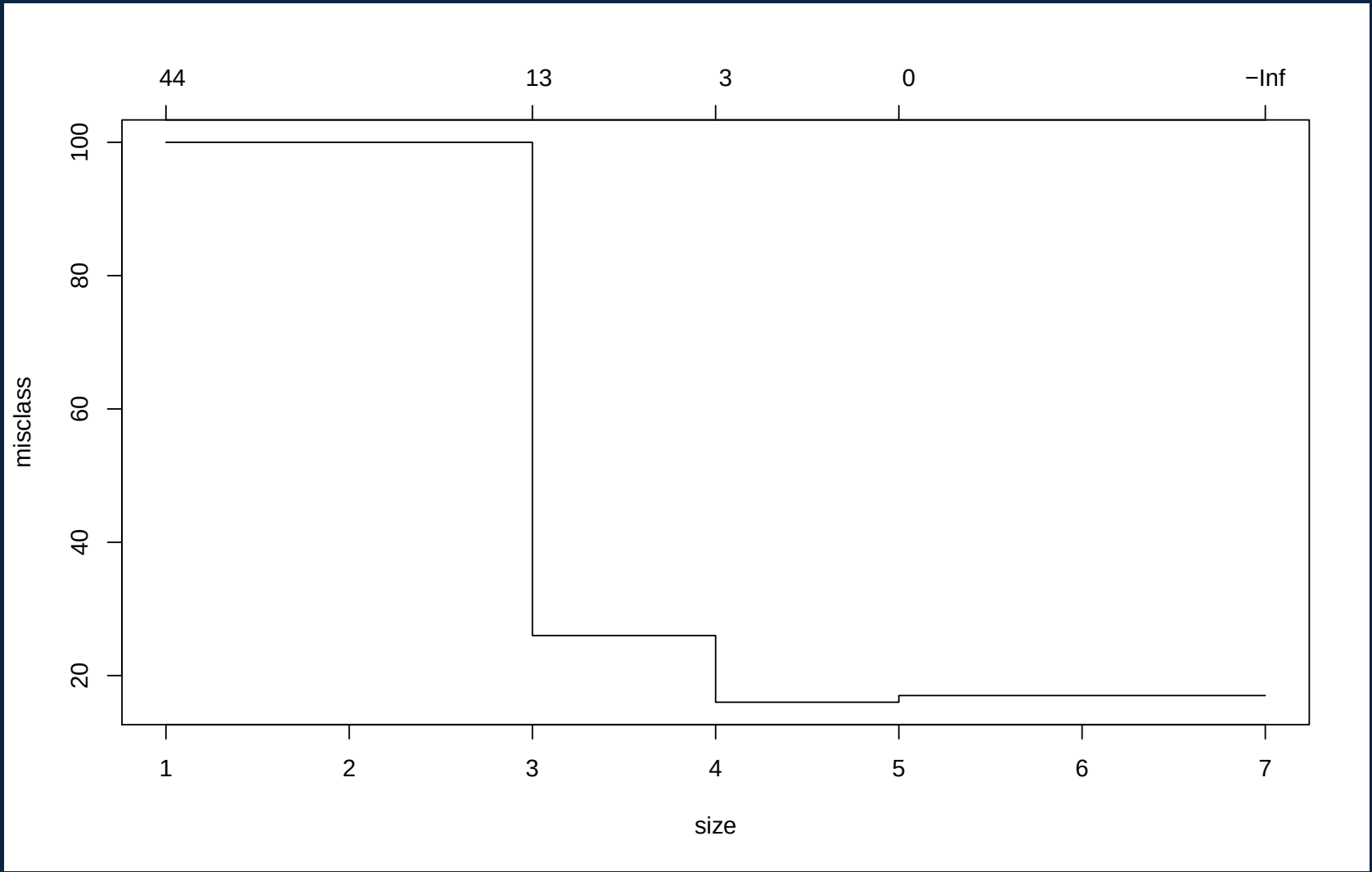
Example: Wine data

First a classification tree



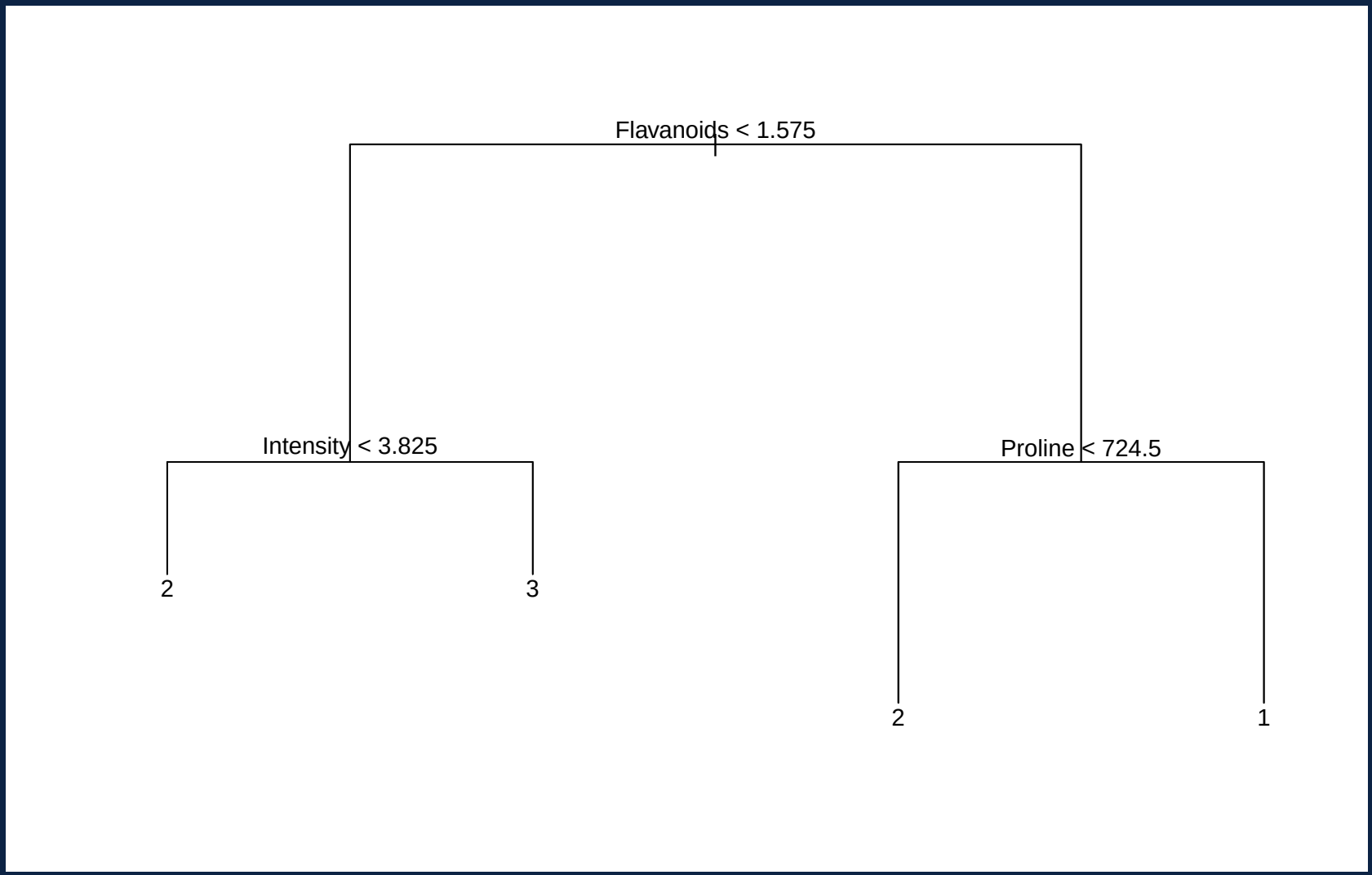
Example: Wine data

BUT, cross-validate it...



Example: Wine data

Prune it back to the minimum, estimate 11/178 =¿ 6.2% misclassifications in the long-run



Example: Wine data

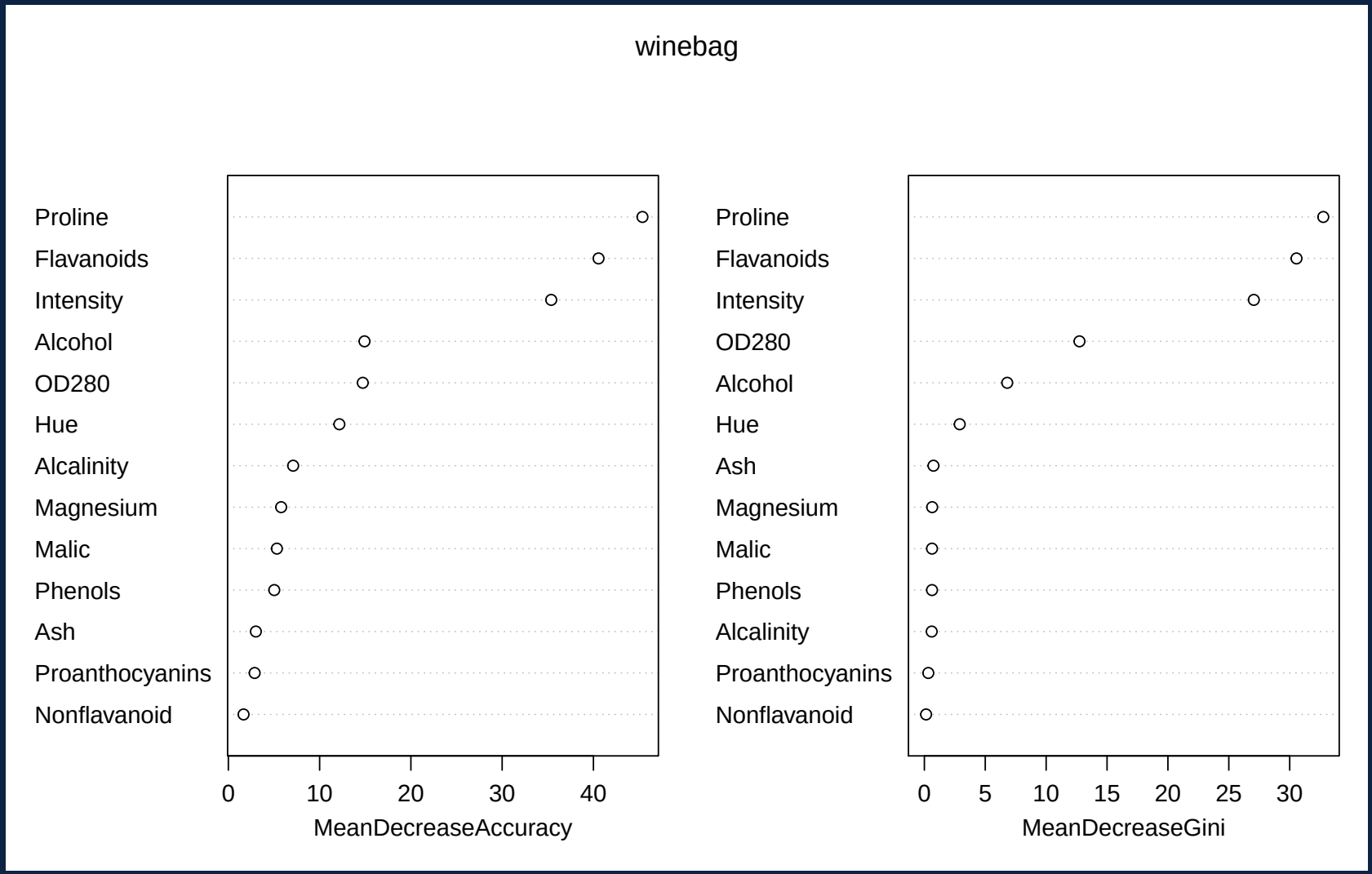
- Not bad...
- Can bagging/random forests improve?

Example: Wine data

```
Call:
 randomForest(formula = factor(Class) ~ ., data = wine, mtry
               Type of random forest: classification
                     Number of trees: 500
No. of variables tried at each split: 13

        OOB estimate of  error rate: 2.81%
Confusion matrix:
   1  2  3 class.error
1 57  2  0  0.03389831
2  1 68  2  0.04225352
3  0  0 48  0.00000000
```

- Note the OOB error rate of 2.8% versus 6.2% for a standard tree

Example: Wine data

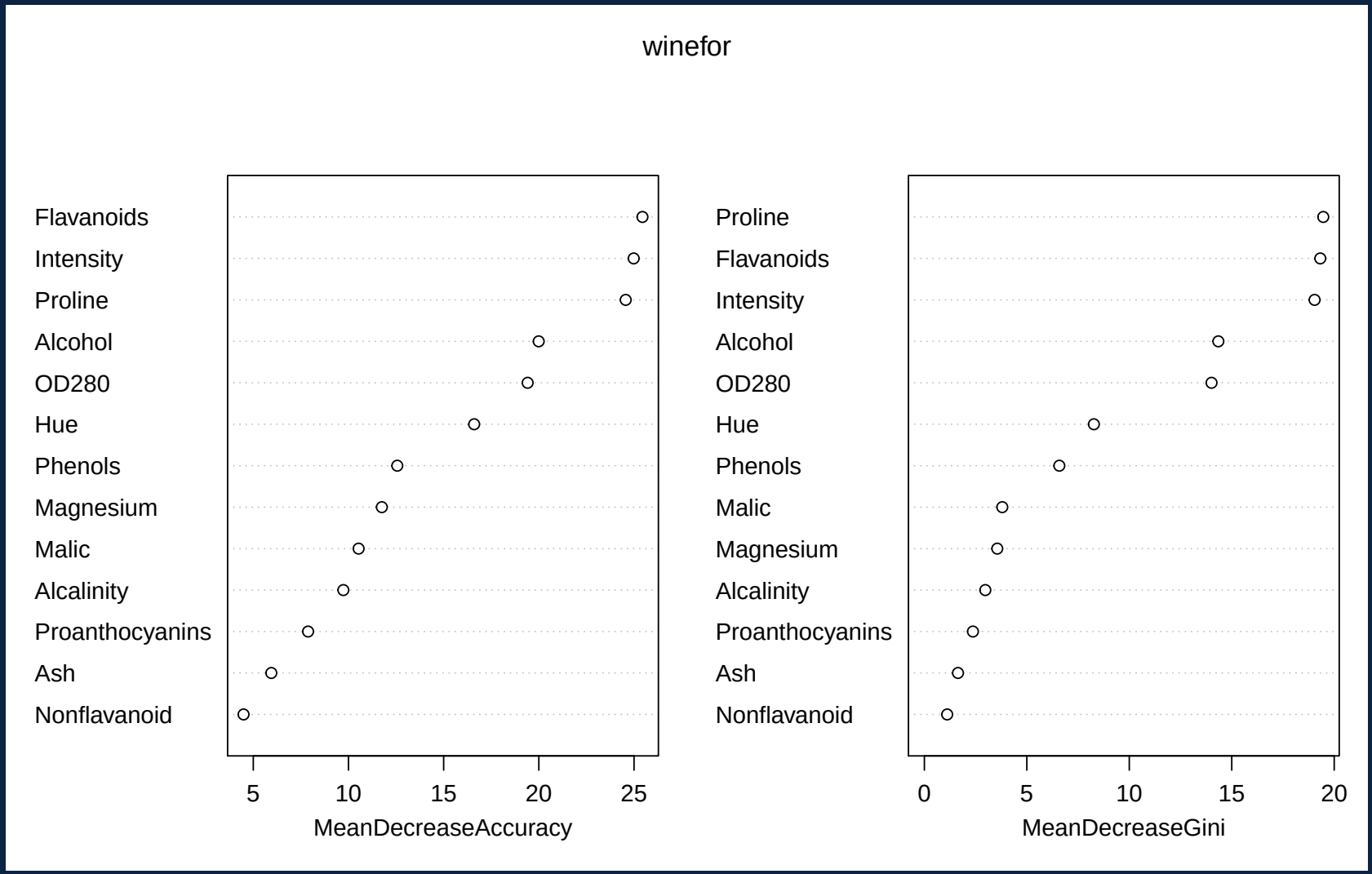


Example: Wine data

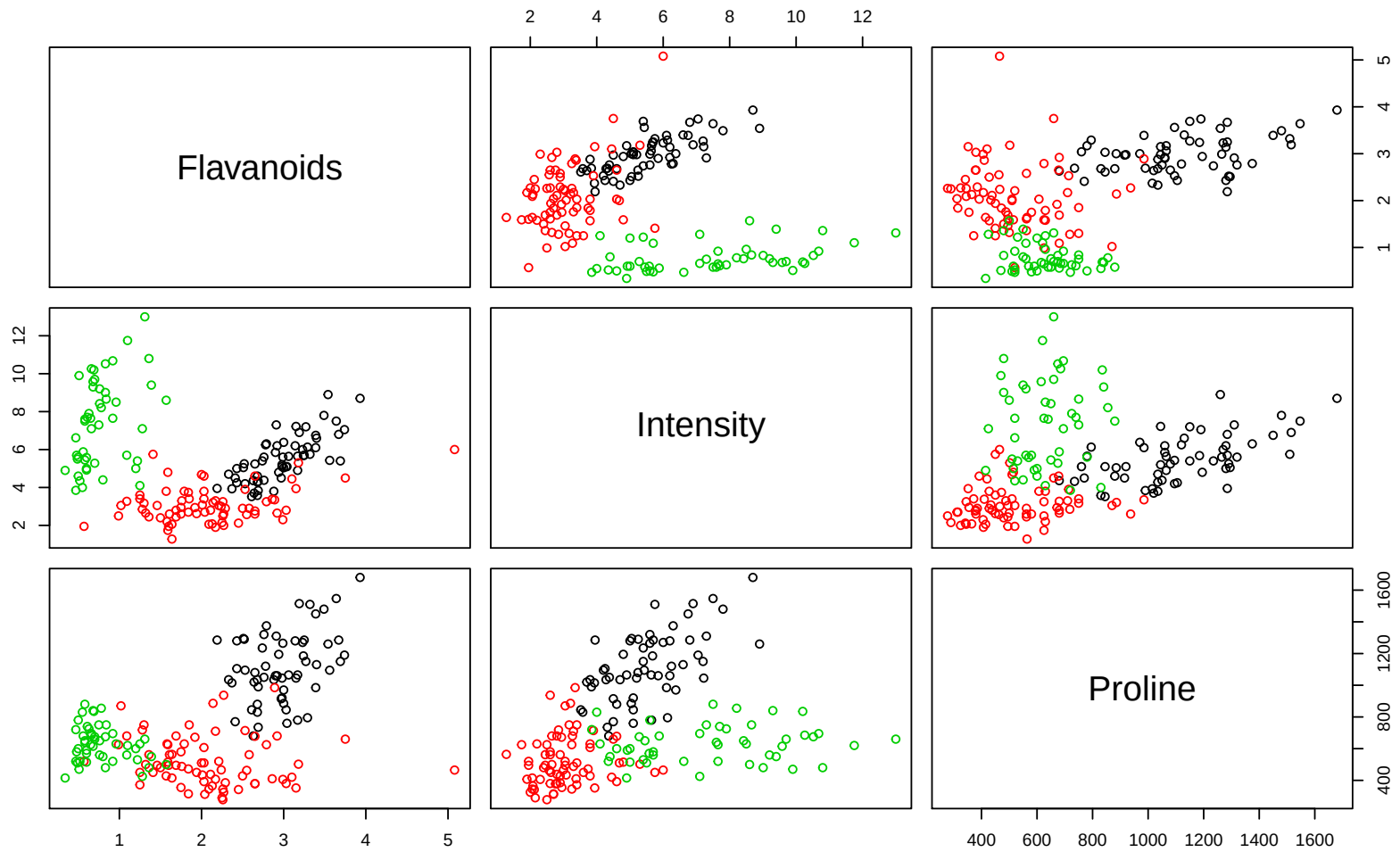
```
Call:
 randomForest(formula = factor(Class) ~ ., data = wine, impor
               Type of random forest: classification
                     Number of trees: 500
No. of variables tried at each split: 3

        OOB estimate of  error rate: 1.69%
Confusion matrix:
   1  2  3 class.error
1 59  0  0  0.00000000
2  1 68  2  0.04225352
3  0  0 48  0.00000000
```

- Note the OOB error rate of 1.7% versus 2.8% versus 6.2%

Example: Wine data



Example: Wine data



Implementation in R

- Quick note that the `ranger` package in R should be considered as a replacement for the `randomForest` package.
- Especially in cases where randomForest might be slow (high observation counts, for example).


THE UNIVERSITY OF BRITISH COLUMBIA