Lesson Objectives

1. Frame. Why an ensemble of trees can outperform a single model.
2. Apply. Grow a classification tree and a random forest, and tune the forest on out-of-bag error.
3. Mean. What variable importance does and does not say.
4. Communicate. What the forest predicts, and how confident to be when you cannot show the mechanism.

Bottom Line Up Front

1. **A tree asks yes/no questions.** Each split is the cut that leaves the two boxes least mixed. It finds thresholds and interactions on its own.
2. **One tree overfits or changes with the data.** Grown all the way, it memorizes; a slightly different sample grows a different tree.
3. **A forest averages the noise away.** 500 trees on bootstrap samples, a few predictors per split, majority vote. You trade a readable rule for accuracy.

Key Definitions

Gini impurity: how mixed a box is between passers and failers. 0 is pure; 0.5 is half and half. Lower is better.

Overfitting: a model that memorizes its training data and does not predict well out of sample.

Bootstrap sample: n rows drawn from the data with replacement. Some rows appear more than once, some not at all.

Out-of-bag (OOB): the rows a tree never saw. Scoring each cadet with only those trees gives a free test set.

Variable importance: scramble one predictor among the OOB cadets; how much worse does the forest predict?

Trees and Forests

Gini, one box	$G = 2p(1-p), \quad p =$ share who passed
Gini, a split	$G = \frac{n_L}{n} G_L + \frac{n_R}{n} G_R, \quad$ keep the cut with the lowest G
Tree	`rpart(pass ~ ., data = train, method = "class")` knobs: `cp` (0.01), `minsplit` (20), `maxdepth` (30)
Forest	`randomForest(pass ~ ., data = train, ntree = 500, mtry = 2, nodesize = 1)` knobs: `ntree` trees, `mtry` predictors per split, `nodesize` smallest box

Who Will Pass the IOCT?

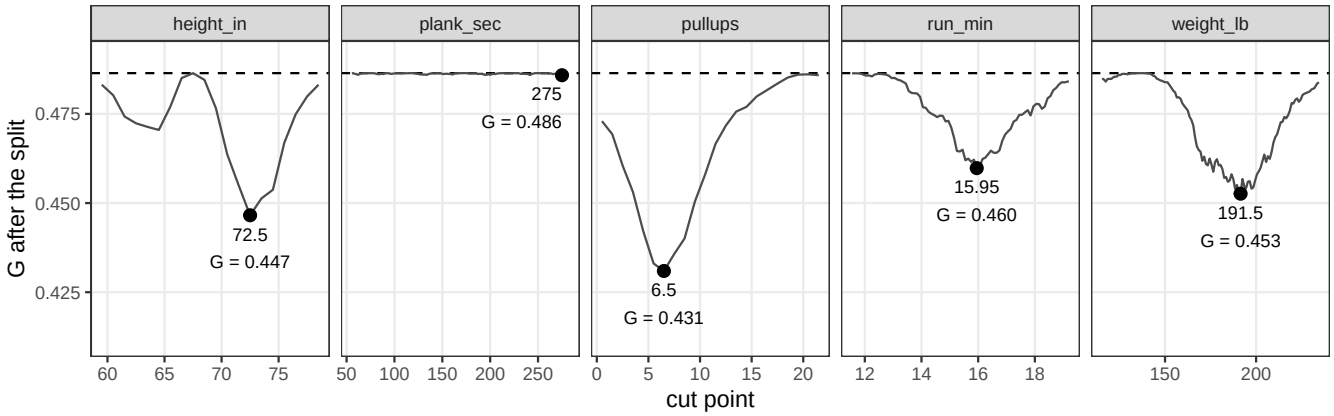
Last year's 2,100 cadets (`train`) have IOCT results; 1,223 passed. Predict which of this year's 900 (`test`) will pass. Predictors: `class_year`, `height_in`, `weight_lb`, `pullups`, `run_min`, `plank_sec`. Every model is fit on `train` and scored on `test`.

`coef(fit_logit)`, logistic regression, $F_1 = 0.750$ on `test`

```
(Intercept) class_yearfirstie class_yearplebe class_yearyuk height_in weight_lb pullups run_min plank_sec
      1.613             0.124          -0.309        -0.019     0.024    -0.011   0.137  -0.167     0.001
```

1. Very short and very tall cadets both pass less often. Logistic regression gives height one coefficient, 0.024. Why can't it see both ends?

The tree's first question: try every cut of every predictor, compute G after the split. Dashed line: G before any split, 0.486.



2. The first split is `pullups` at 6.5. Using the figure, why that predictor and that cut?

3. `plank_sec`'s line hugs the dashed line everywhere. What does that say about planks?

Growing the Tree

The one split predicts pass for 7 or more pull-ups (69% passed) and fail for 6 or fewer (32%): $F_1 = 0.757$. The tree repeats the search inside each box.

Best cut inside each box after the first split

box	best cut	G after	pass rate on each side
6 or fewer pull-ups	weight at 179.5	0.381	47% and 14%
7 or more pull-ups	height at 74.5	0.397	72% and 13%

4. The weight cut has the lowest G of anything, yet the two-split tree keeps only the height cut. Why does `rpart` drop the weight cut?

Third split: best cut inside each 7 or more box

box	best cut	G after	pass rate on each side
75 or taller (70)	run at 13.15	0.170	56% (9 cadets) and 7%
under 75 (1,415)	height at 64.5	0.371	35% and 76%

5. The run cut flips the 9 fast, tall runners from a 0 to a 1, yet the three-split tree keeps only the height cut. Why?

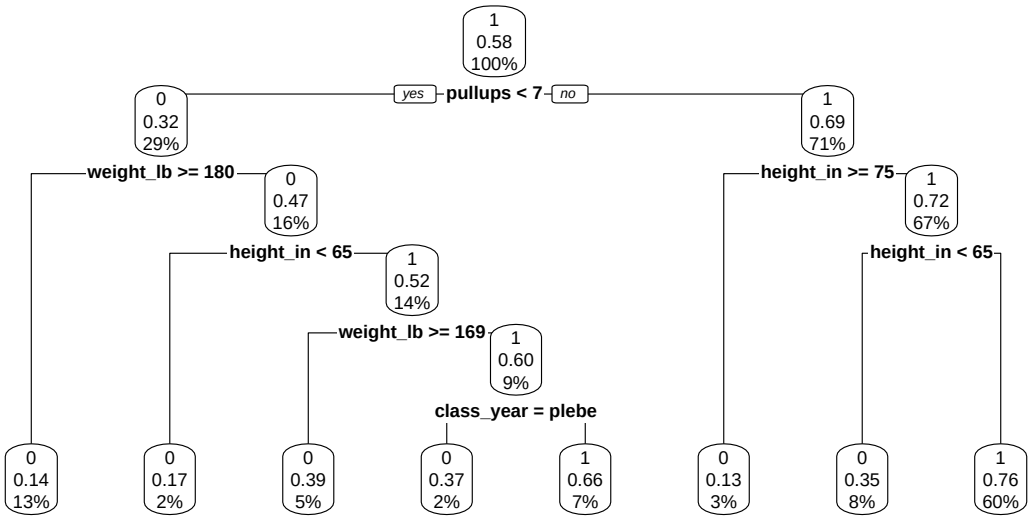
```
fit_big <- rpart(pass ~ ., data = train, method = "class", cp = 0, minsplit = 2)

splits: 478       accuracy on train: 1.000

model         accuracy  sensitivity  precision  f_meas      (all on test)
one split        0.692        0.859      0.677   0.757
two splits       0.717        0.845      0.706   0.769
three splits     0.741        0.791      0.757   0.774
no limits        0.662        0.688      0.702   0.695
limited tree     0.737        0.841      0.729   0.781
```

6. The no-limits tree is 100% right on `train` and the worst model on `test`. What happened?

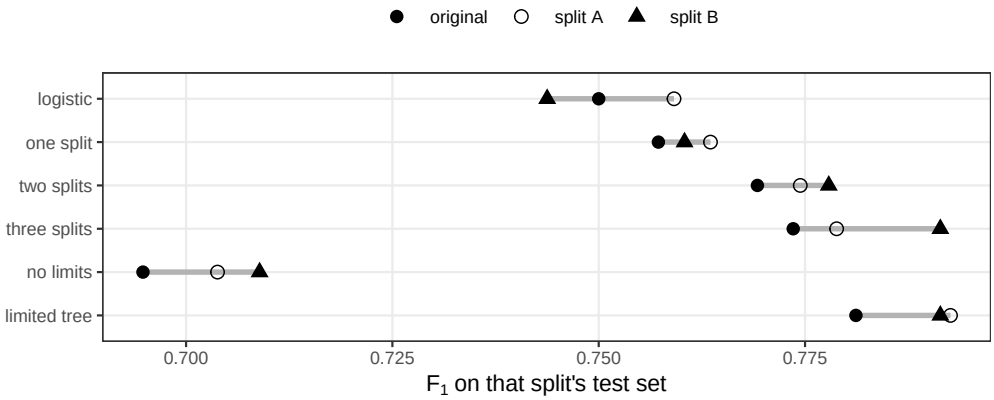
7. The limited tree uses `rpart`'s defaults (`cp = 0.01`, `minsplit = 20`). Below. Why does stopping early beat both three splits and no limits?



Each box, top to bottom: the prediction (1 = pass, 0 = fail); the proportion of `train` cadets in the box who passed; the percent of all `train` cadets who land in the box.

So What's the Downside?

Same 3,000 cadets, two new random 70/30 splits. The limited tree on Split A has 10 boxes; on Split B, 4 boxes, and its first cut is 6 pull-ups, not 7.



8. Same cadets, a 10-box tree and a 4-box tree. What does that say about trusting one tree to tell you what drives passing?

Random Forest

500 trees. Each grows with no limits on its own bootstrap sample of 2,100 cadets, sees only `mtry` random predictors at each split, and votes. The majority wins.

```
randomForest(pass ~ ., data = train, ntree = 500, mtry = 2, nodesize = 1)

        OOB estimate of  error rate: 24.81%
Confusion matrix:
    0    1 class.error
0 557  320   0.3648803
1 201 1022   0.1643500
```

9. Each forest tree grows with no limits, and we just saw that one such tree overfits. Why doesn't the forest?

10. The OOB error is 24.8%. Where does it come from, and why didn't we need `test` for it?

54 forests, `mtry` 1 to 6 × `ntree` 100, 300, 500 × `nodesize` 1, 5, 20, scored on OOB F_1. Top three:

```
 mtry ntree nodesize oob_f1
    1   500        5  0.810
    1   300        5  0.807
    1   500        1  0.807          tuned forest on test: accuracy 0.768, F1 0.805
```

11. `mtry = 1` wins: each split gets one random predictor. Why might fewer predictors per split help, and why tune on OOB F_1 instead of `test`?

So What Do I Make of It?

Logistic regression gives a coefficient for every predictor. The forest gives none.

`importance(fit_best, type = 1)`, the scramble test

```
predictor   MeanDecreaseAccuracy
height_in                 50.2
pullups                   47.8
weight_lb                 29.9
run_min                   23.5
class_year                 3.67
plank_sec                 -2.38
```

12. Height ranks first. A commander asks: "so taller cadets pass more?" What can you and can't you say?

13. In two sentences, tell the commander what the forest predicts and how far to trust it.

Key Takeaway

A tree splits on the cut that leaves the least mixed boxes, finds interactions on its own, and overfits if you let it. A forest of noisy, disagreeing trees votes the noise away and predicts best, but it gives up the readable rule: variable importance says what it uses, not how.