Apr 24, 2023

Machine Learning Methods and Applications

Week 8. Ensemble learning | Random forests

Remember

Ensemble learning

Ensemble learning

Ensembling is a technique of combining two or more models of similar or dissimilar types.

An ensemble occurs when probability predictions or numerical predictions of multiple ML models are combined by averaging, weighting each model and adding them together or using the most common observation between models.

The predictions are averaged in regression models, while are voted in classification models.

Ensemble methods

1. **Bagging**: Training multiple models from different subsamples of the training set.

2. **Boosting:** Training multiple models each of which learns to fix the prediction errors of a prior model in the chain.

3. **Stacking:** Training multiple models and supervisor model that learns how to best combine the predictions of the primary models.

Why ensemble models?

- Trees can easily overfit, leading to poor generalizability to test/unseen data.

- To overcome high variance problem in the trees.

- **The primary goal is to improve the performance of the model.**

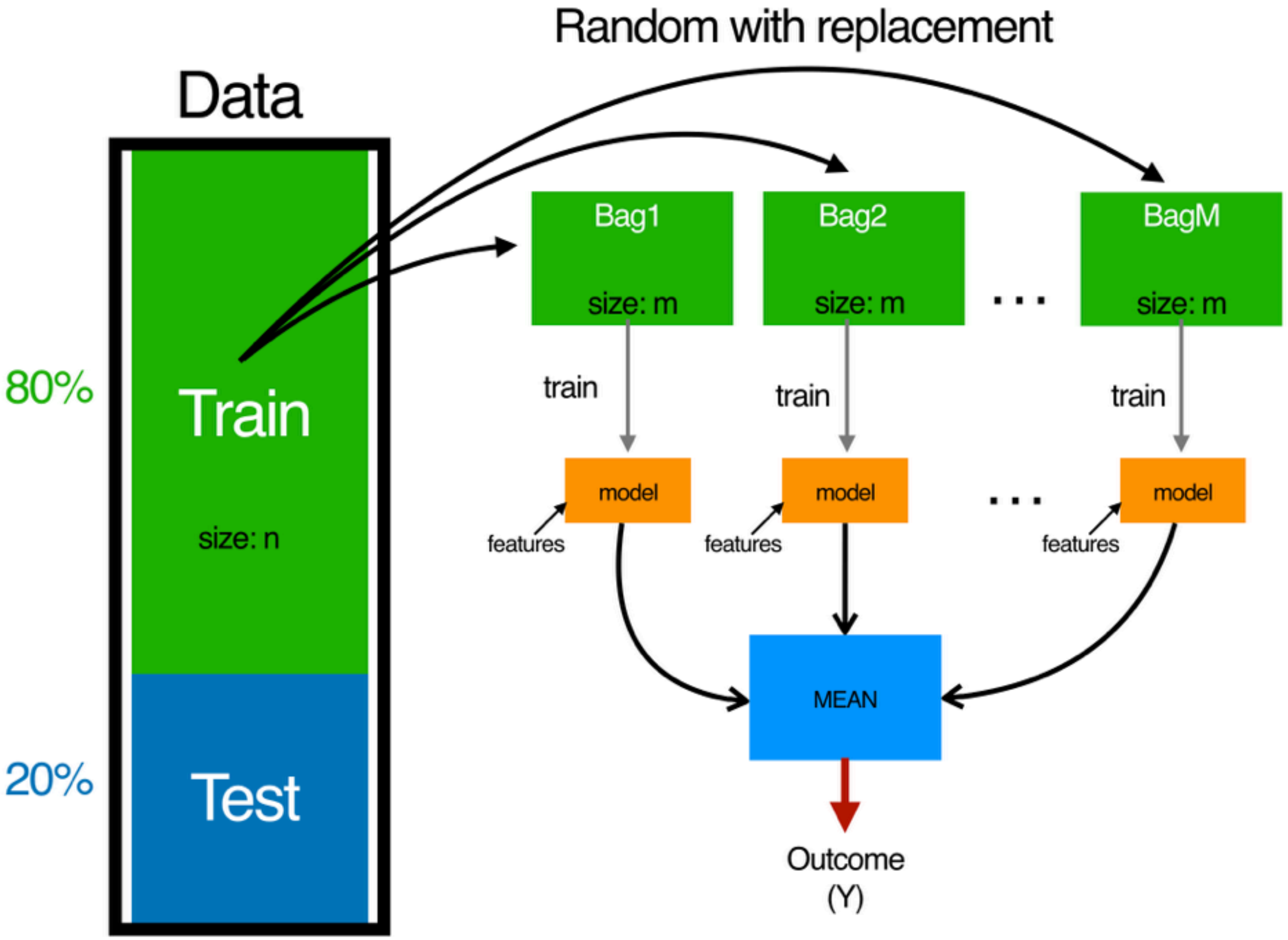
Bagging trees

Bagging trees

- **B**ootstrap **agg**regat**ing** trees: is a general method for fitting multiple versions of a decision tree and then combining them into an aggregated prediction.

- “wisdom of the crowd!”

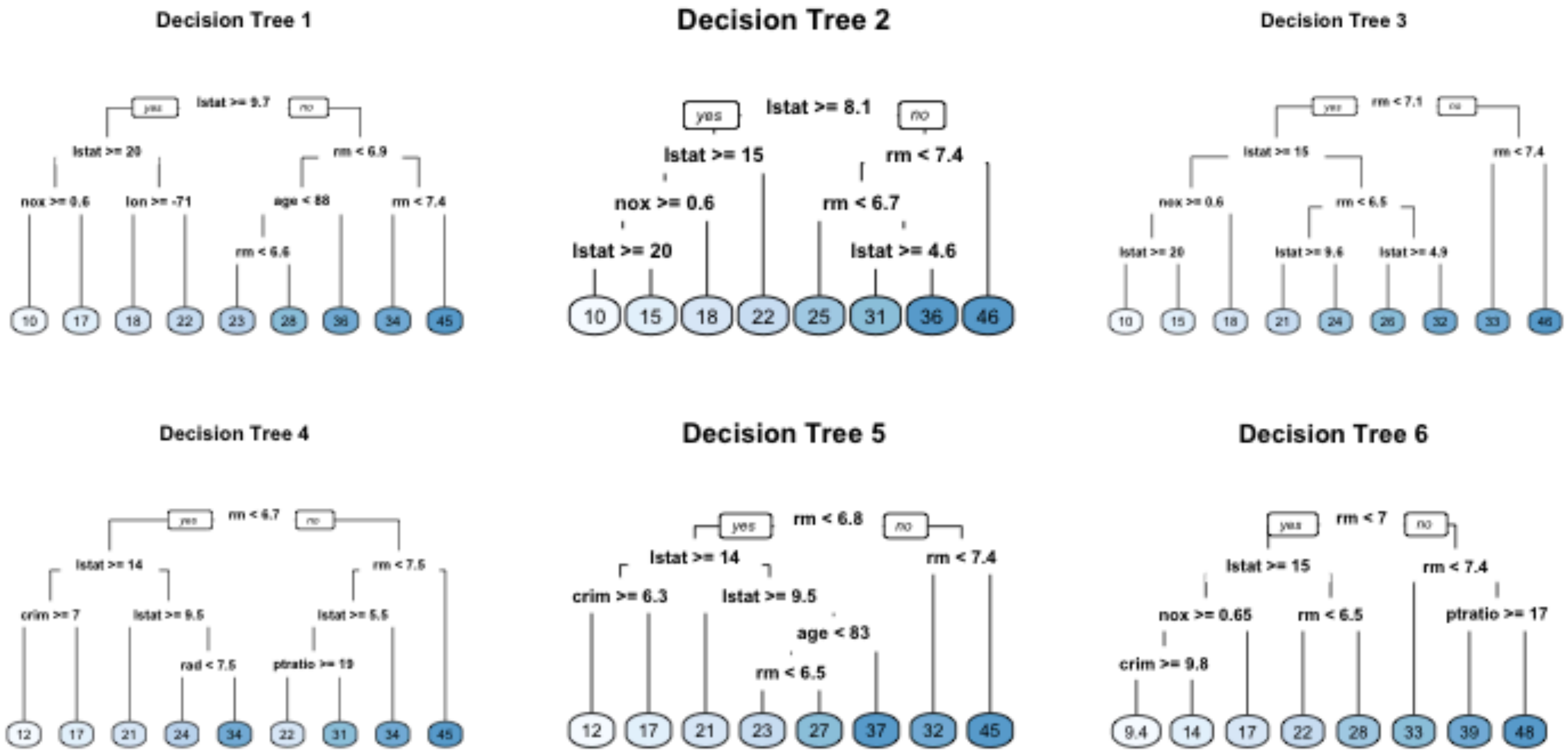
- The bootstrap can be used to estimate the standard errors of the coefficients from a learning model.



Bagging trees

- Produce the final prediction by averaging the predictions for a regression task, while by voting for a classification tasks.
- Improve the stability and accuracy of regression and classification models.
- Help to reduce variance and minimize overfitting.

Bagging trees: Example



Decision Tree 1

Decision Tree 2

Decision Tree 3

Decision Tree 4

Decision Tree 5

Decision Tree 6

Image credit: Boehmke & Greenwell - Hands-On Machine Learning with R

Random forests

Random forests

- The random forest model is an extension of bagging trees.

- The difference is random forests use the subset of the features to train each tree while bagging trees use the all features.

- Bagging trees results in tree correlation that limits the effect of variance reduction.

- Random forests help to reduce tree correlation by injecting more randomness into the tree-growing process.

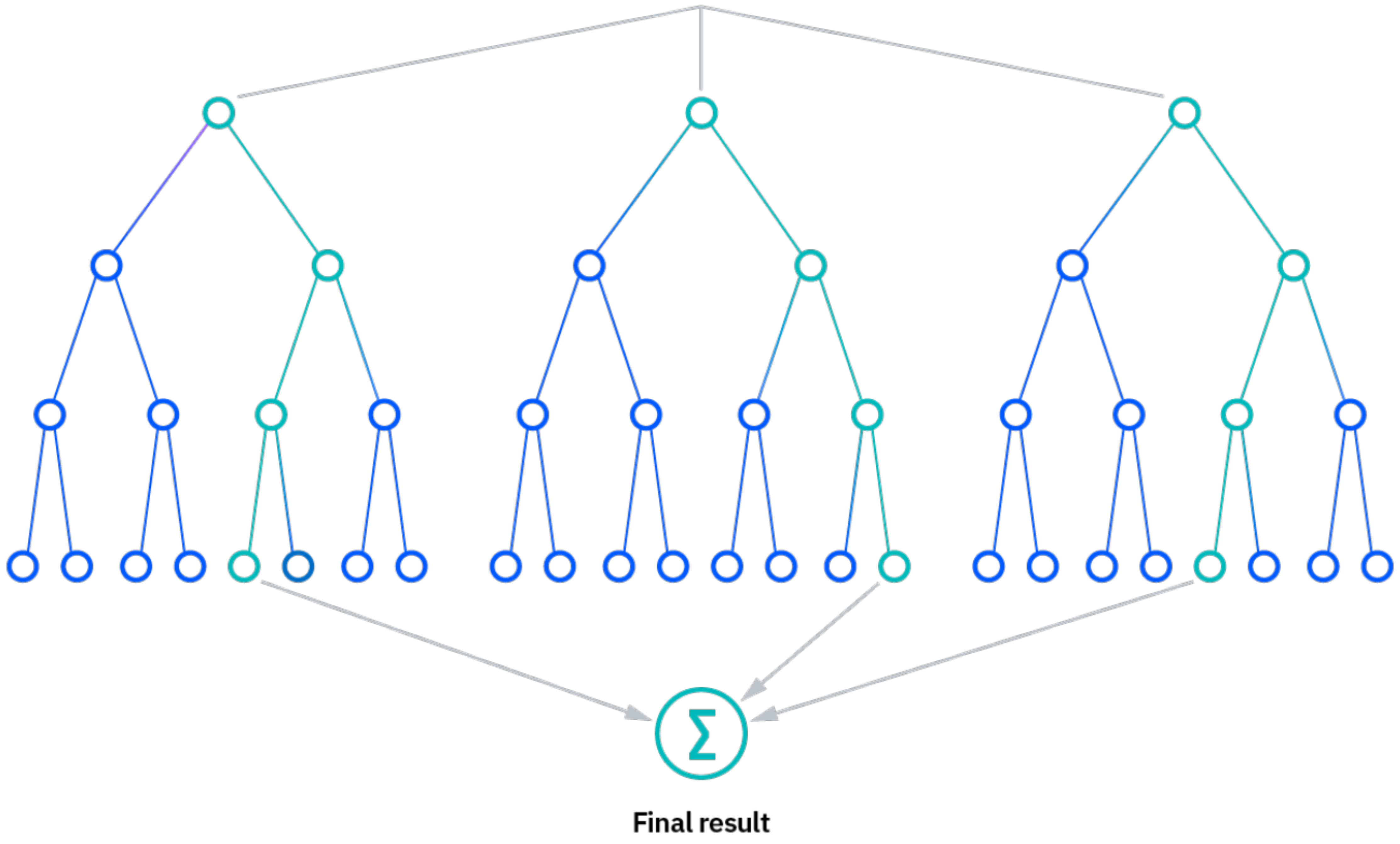


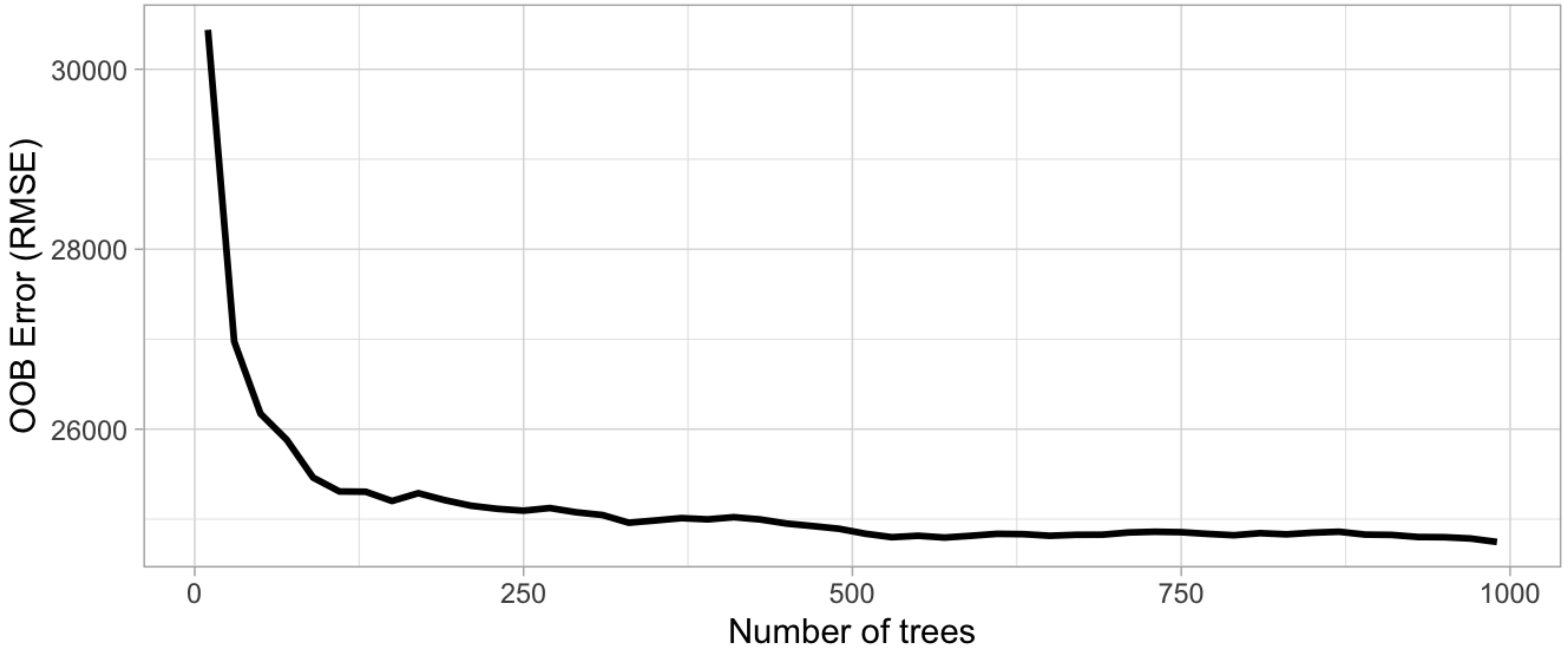
Image Credit: IBM

Random forests: Hyperparameters

1. The number of trees in the forest.
2. The number of features to consider at any given split: mtry
3. The complexity of each tree
4. The sampling scheme
5. The splitting rule to use during tree construction

1. Number of trees

- The number of trees needs to be sufficiently to stabilize the error rate.
- A good rule of thumb is to start with 10 times the number of features.
- More trees provide more robust and stable error estimates; however, the impact on computation time increases linearly with the number of trees.

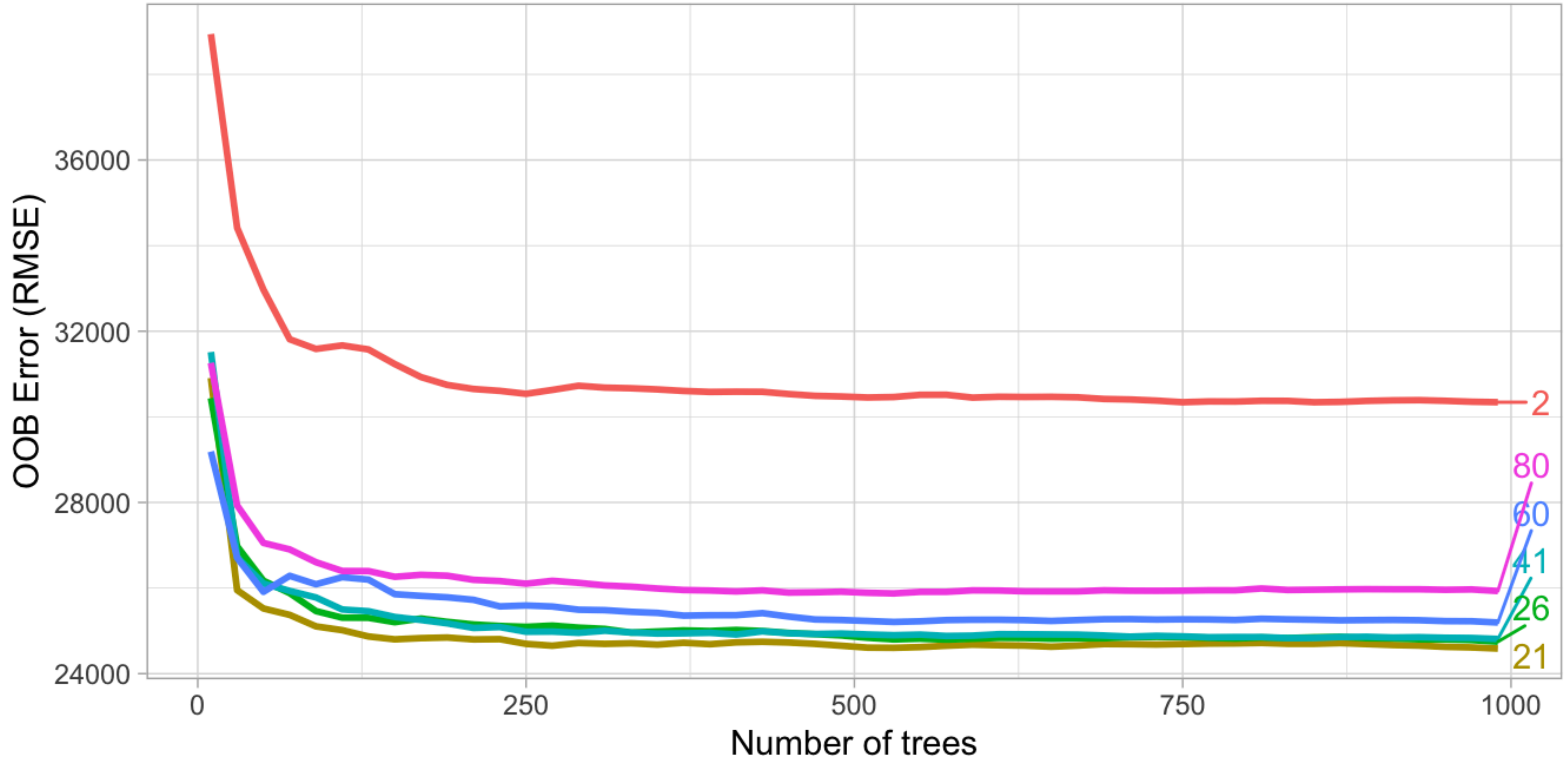


The Ames data has 80 features and starting with 10 times the number of features typically ensures the error estimate converges.

Image credit: Boehmke & Greenwell - Hands-On Machine Learning with R

2. The number of features to consider at any given split: mtry

- mtry controls the split-variable randomization feature of random forests and helps to balance low tree correlation with reasonable predictive performance.
- Default value of mtry is $p/3$ for regression and $\sqrt{p}$ for classification.

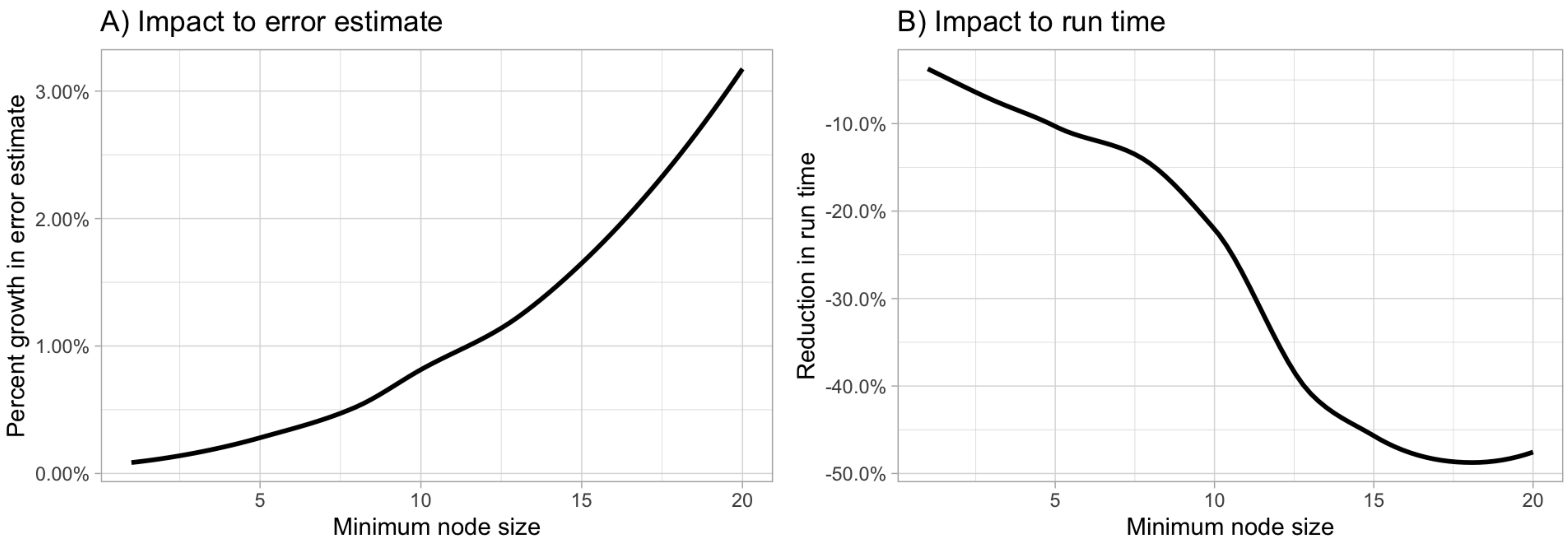


For the Ames data, an mtry value slightly lower (21) than the default (26) improves performance.

Image credit: Boehmke & Greenwell - Hands-On Machine Learning with R

3. Tree complexity

- There are more than one hyper parameter that controls the complexity of trees such as node size, max depth, max number of nodes.



Increasing node size to reduce tree complexity will often have a larger impact on computation speed (right) than on your error estimate.

Image credit: Boehmke & Greenwell - Hands-On Machine Learning with R

Application

See the R codes on the course GitHub repository!

The video recording of today's lecture will be available on YouTube, and slides on GitHub.
Feel free to contact me via e-mail: mustafacavus@eskisehir.edu.tr