Random Forests

Aquiles Farias

Random Forests - References

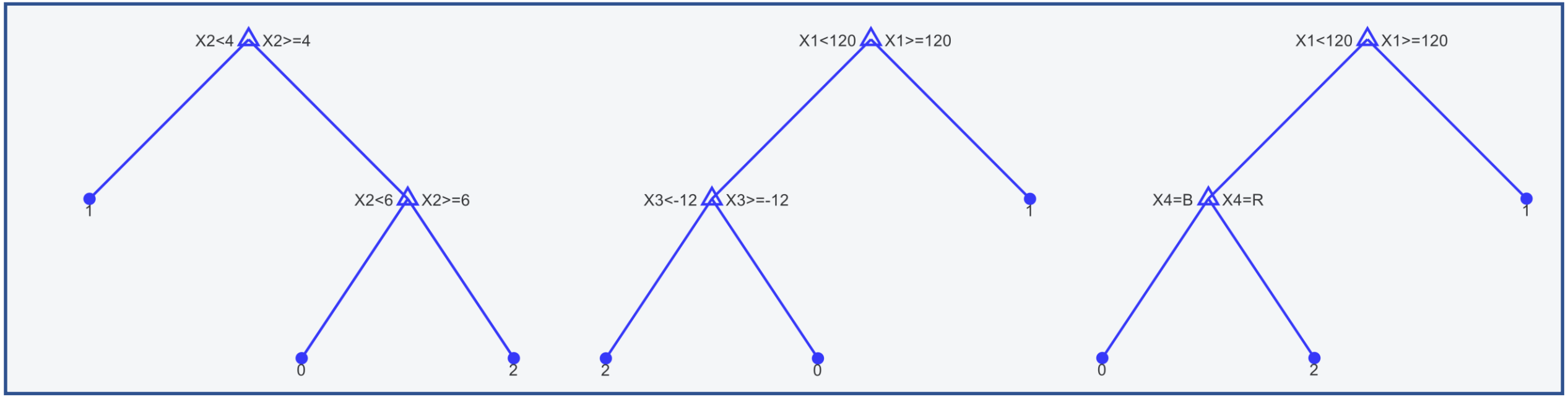
- **[1] Breiman, L. "Bagging Predictors."** ***Machine Learning*****. Vol. 26, pp. 123–140, 1996.**
- **[2] Breiman, L. "Random Forests."** ***Machine Learning*****. Vol. 45, pp. 5–32, 2001.**
- [3] Freund, Y. "A more robust boosting algorithm." *arXiv:0905.2138v1*, 2009.
- [4] Freund, Y. and R. E. Schapire. "A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting." *J. of Computer and System Sciences*, Vol. 55, pp. 119–139, 1997.
- [5] Friedman, J. "Greedy function approximation: A gradient boosting machine." *Annals of Statistics*, Vol. 29, No. 5, pp. 1189–1232, 2001.
- [6] Friedman, J., T. Hastie, and R. Tibshirani. "Additive logistic regression: A statistical view of boosting." *Annals of Statistics*, Vol. 28, No. 2, pp. 337–407, 2000.
- [7] Ho, T. K. "The random subspace method for constructing decision forests." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 20, No. 8, pp. 832–844, 1998.
- [8] Schapire, R. E., Y. Freund, P. Bartlett, and W.S. Lee. "Boosting the margin: A new explanation for the effectiveness of voting methods." *Annals of Statistics*, Vol. 26, No. 5, pp. 1651–1686, 1998.
- [9] Seiffert, C., T. Khoshgoftaar, J. Hulse, and A. Napolitano. "RUSBoost: Improving classification performance when training data is skewed." *19th International Conference on Pattern Recognition*, pp. 1–4, 2008.
- [10] Warmuth, M., J. Liao, and G. Ratsch. "Totally corrective boosting algorithms that maximize the margin." *Proc. 23rd Int'l. Conf. on Machine Learning, ACM*, New York, pp. 1001–1008, 2006.

From trees to forests

Random Forest

- Supervised learning
- Non-parametric
- Ensemble method
  - Bagging

- Very flexible
- Classification and Regression
- No need to rescale nor center the data

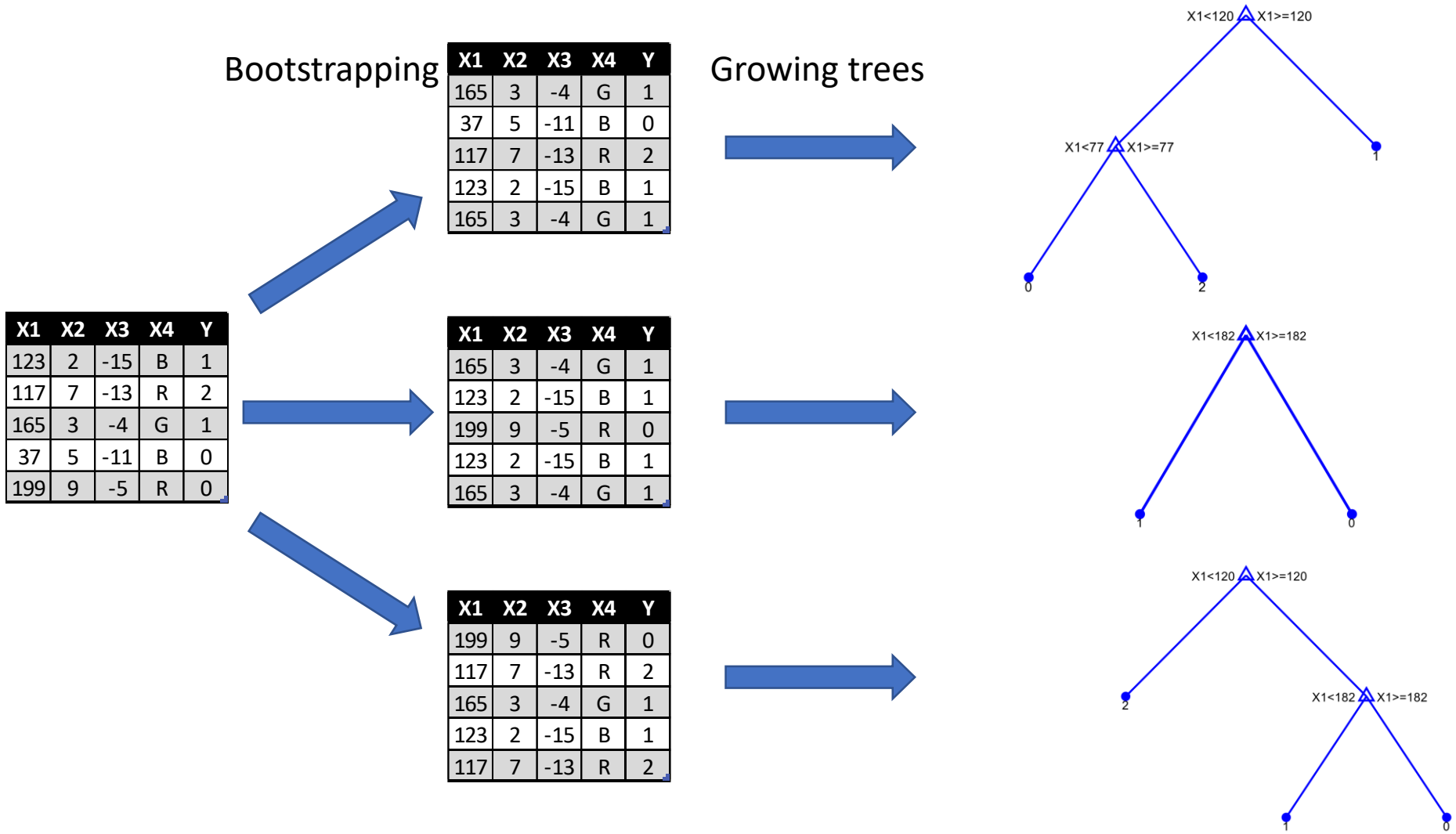


Growing a random forest

- Let T = # of trees in the forest, N = # of observations in the training dataset, F = # of features and S the stop criterion.
- For each tree in T:
  - Sample from the training dataset, with replacement, n observations. Usually $n = N$.
  - Grow the tree. For each split decision:
    - Sample from the features available f features. Usually $f = \sqrt{F}$. If you use all features, then it's a bag of trees
    - Split the tree
    - Check if it's time to stop using S

Bag of trees

Bag of trees



Random Forest

X1	X2	X3	X4	Y
123	2	-15	B	1
117	7	-13	R	2
165	3	-4	G	1
37	5	-11	B	0
199	9	-5	R	0

Bootstrapping

X2	X3	Y
3	-4	1
5	-11	0
7	-13	2
2	-15	1
3	-4	1

X1	X3	Y
165	-4	1
123	-15	1
199	-5	0
123	-15	1
165	-4	1

X1	X4	Y
199	R	0
117	R	2
165	G	1
123	B	1
117	R	2

Sampling 2 Features

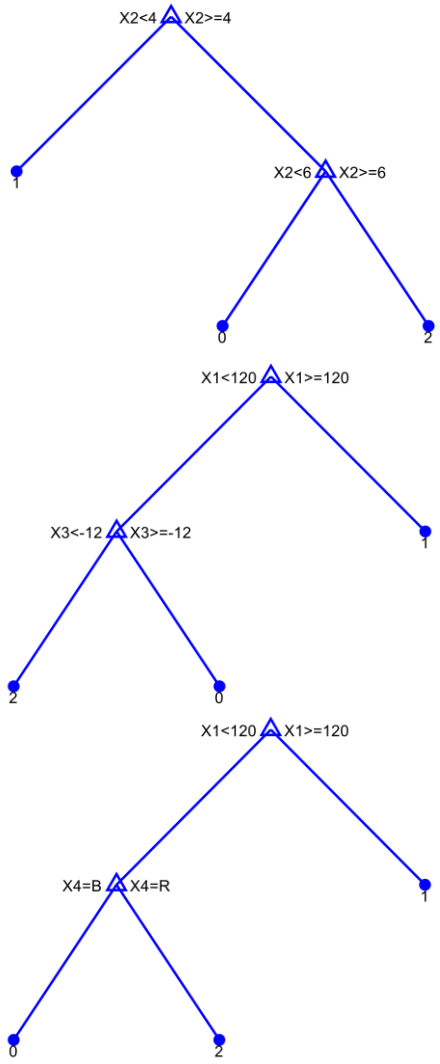
X2	X3	Y
3	-4	1
5	-11	0
7	-13	2
2	-15	1
3	-4	1

X3	X1	Y
3	-4	1
2	-15	1
9	-5	0
2	-15	1
3	-4	1

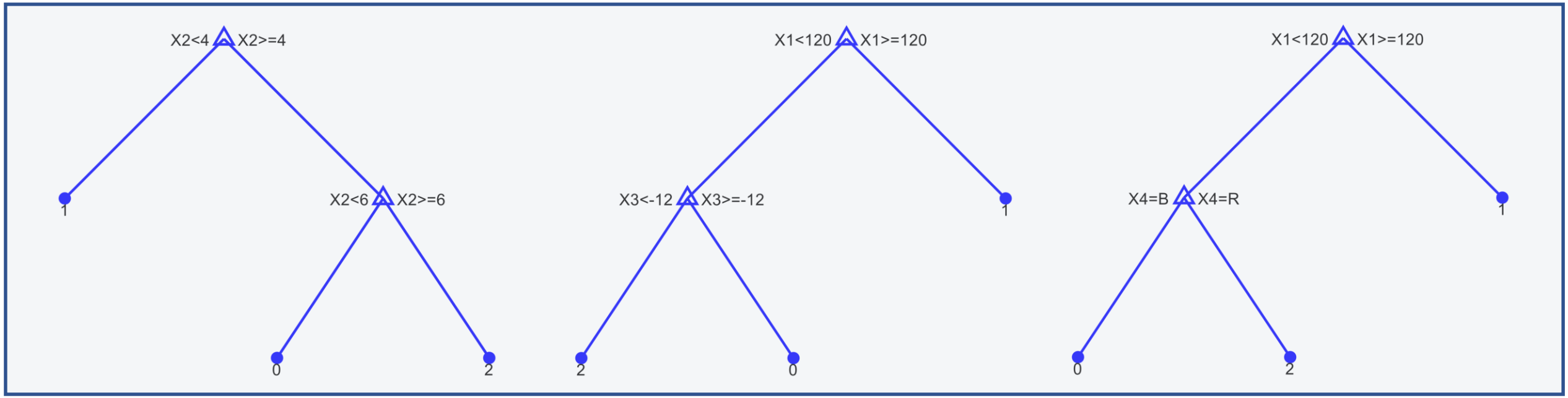
X4	X1	Y
9	-5	0
7	-13	2
3	-4	1
2	-15	1
7	-13	2

Growing trees

Random Forest



Growing a random forest

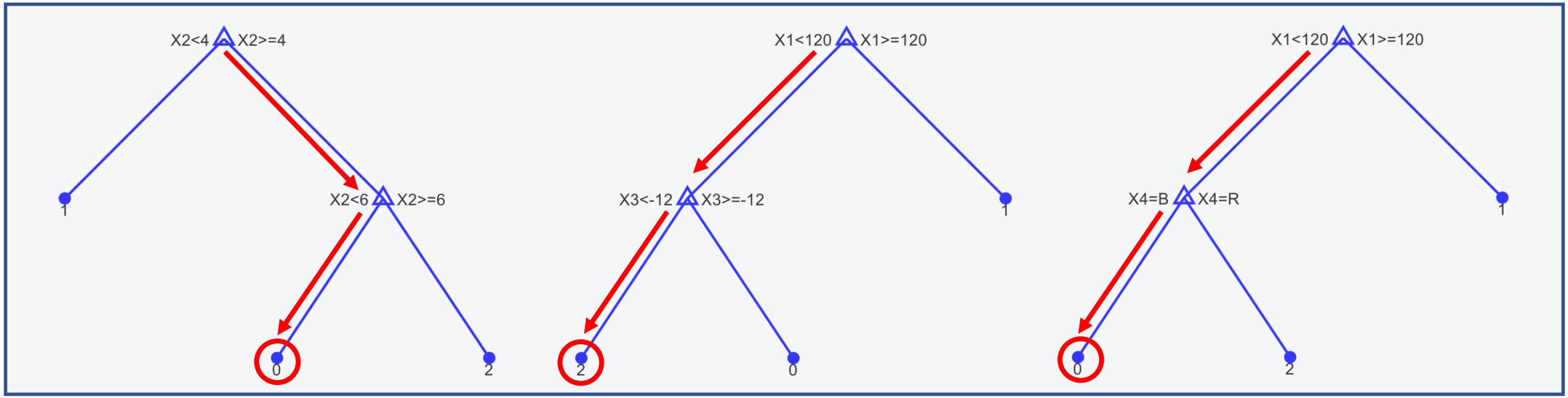


Prediction using a random forest

- Evaluate your new data in each one of the trees in the Random Forest model
- Classification Random Forests
  - Take the statistical mode. Each tree would have 1 vote on deciding the classification.
  - What if there's a tie? Randomize between (among) them!!!!
- Regression Random Forests
  - Take an average across the predictions from all trees

Growing a random forest

Predicting new data: X1=100, X2=5, X3=-10, X4=B



Mode = 0

Mean = 0.67

Out-of-bag evaluation

- You can validate the Random Forest as a whole averaging out the out-of-bag evaluations of each tree – **no need to cross-validate**
- With Random Forest (bagging), for each tree, only about 63% of the unique observations are sampled
- About 37% are not seen by that tree, so its performance can be evaluated on these data points

- $P(x_1 \in \mathcal{B}) = 1 - P(x_1 \notin \mathcal{B}) = 1 - \left(\frac{n-1}{n} \times \frac{n-1}{n} \times \cdots \times \frac{n-1}{n}\right)$

$$= 1 - \left(\frac{n-1}{n}\right)^n = 1 - \left(1 - \frac{1}{n}\right)^n$$

$$\lim_{n\to\infty} 1 - \left(1 - \frac{1}{n}\right)^n = 1 - e^{-1} \cong 0.63$$

Out-of-bag evaluation

Random Forest

Model Validation

X1	X2	X3	X4	Y
123	2	-15	B	1
117	7	-13	R	2
165	3	-4	G	1
37	5	-11	B	0
199	9	-5	R	0

Bootstrapping

X1	X2	X3	X4	Y
165	3	-4	G	1
37	5	-11	B	0
117	7	-13	R	2
123	2	-15	B	1
165	3	-4	G	1

X1	X2	X3	X4	Y
165	3	-4	G	1
123	2	-15	B	1
199	9	-5	R	0
123	2	-15	B	1
165	3	-4	G	1

X1	X2	X3	X4	Y
199	9	-5	R	0
117	7	-13	R	2
165	3	-4	G	1
123	2	-15	B	1
117	7	-13	R	2

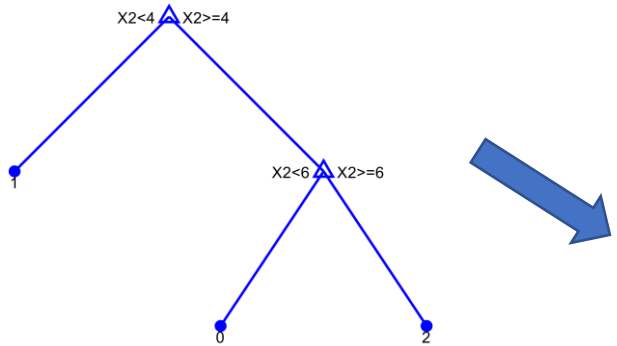
Sampling 2 Features

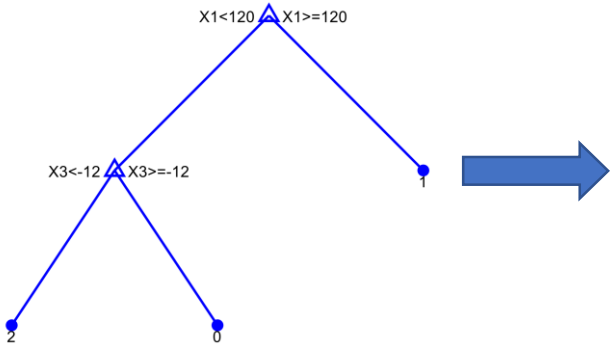
X1	X2	X3	X4	Y
123	2	-15	B	1
117	7	-13	R	2
165	3	-4	G	1
37	5	-11	B	0
199	**9**	**-5**	**R**	**0**

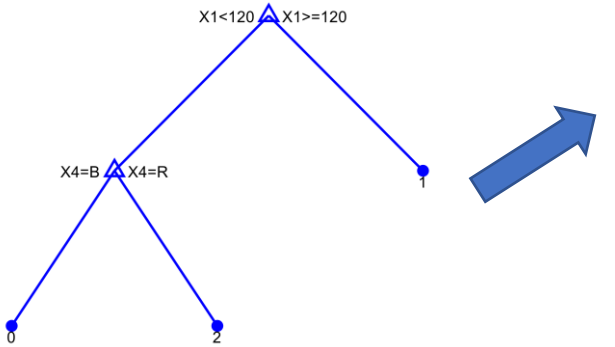
X1	X2	X3	X4	Y
123	2	-15	B	1
117	**7**	**-13**	**R**	**2**
165	3	-4	G	1
37	**5**	**-11**	**B**	**0**
199	9	-5	R	0

X1	X2	X3	X4	Y
123	2	-15	B	1
117	7	-13	R	2
165	3	-4	G	1
37	**5**	**-11**	**B**	**0**
199	9	-5	R	0

Predict

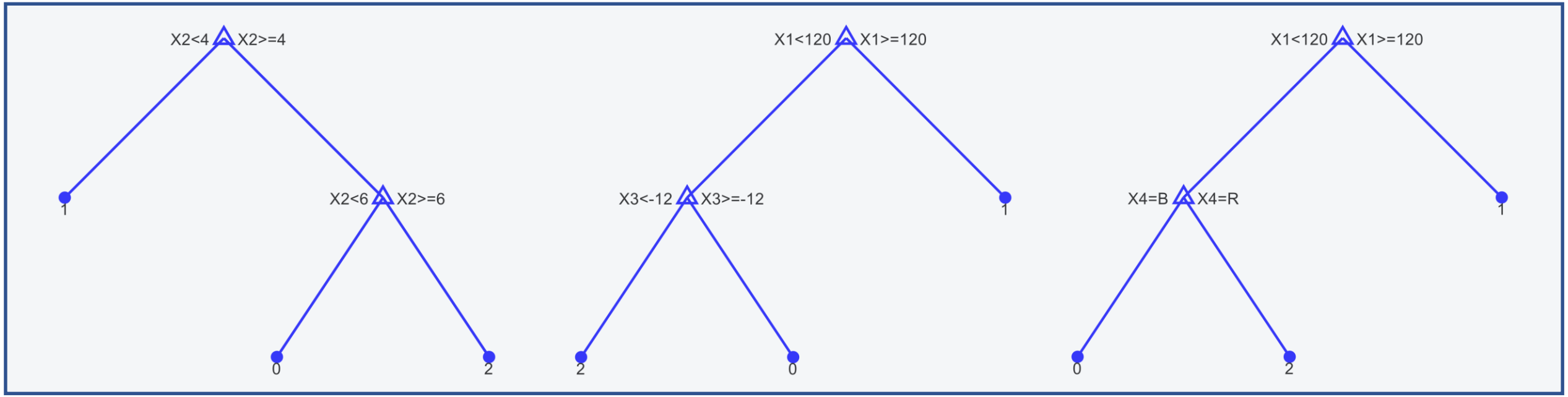






Feature Importance

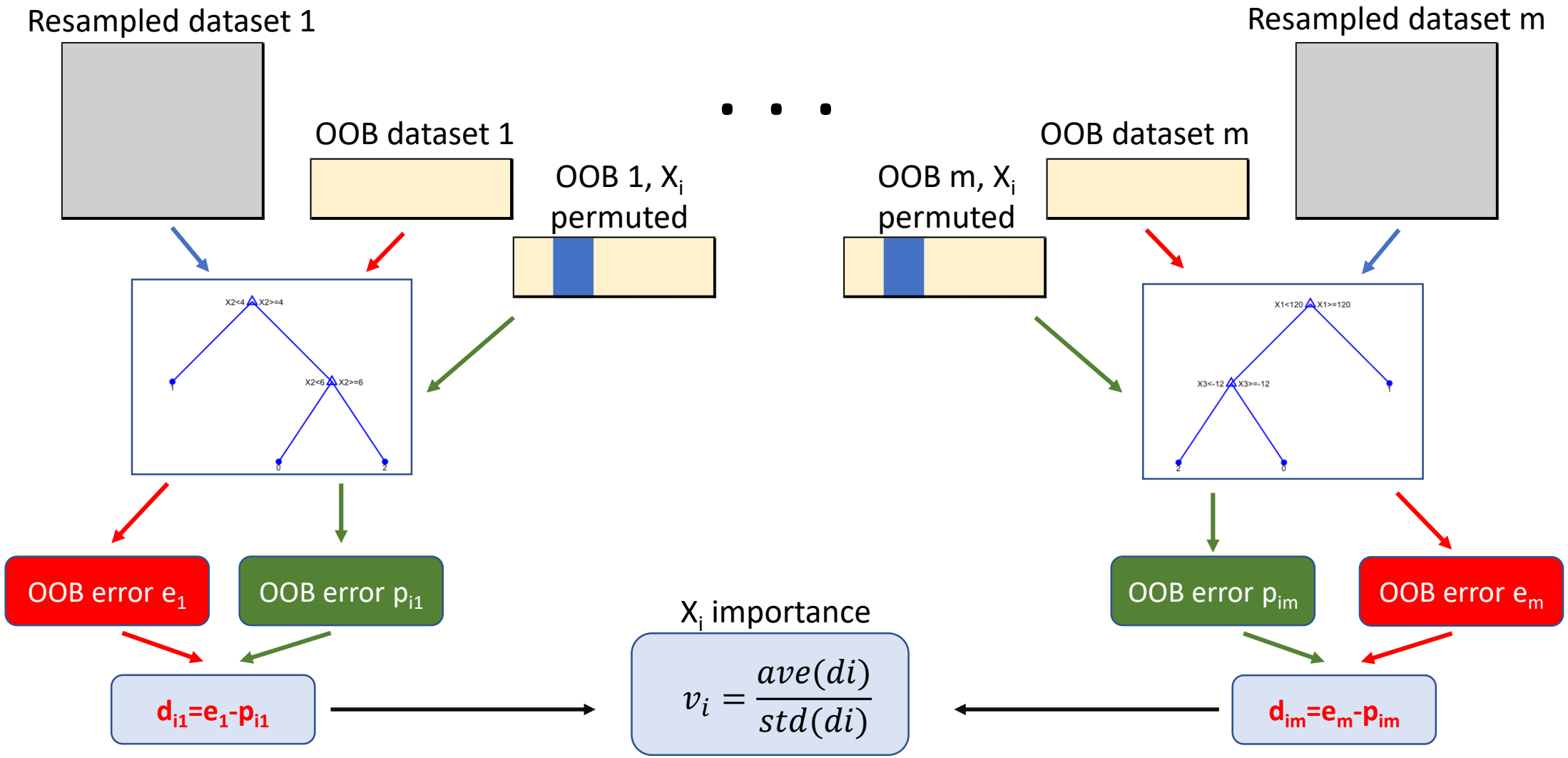
- The number of times each variable is selected by all individual trees in the ensemble.



Feature Importance

- Gini Importance / Mean Decrease in Impurity (MDI)
  - Gini Importance or Mean Decrease in Impurity (MDI) calculates each feature importance as the sum over the number of splits (across all tress) that include the feature, proportionally to the number of samples it splits.
- Out-of-bag permutation
  - For each variable (feature) X_i in the dataset, keep all others (columns) unchanged and randomly permutate X_i (lines).
  - Calculate the loss in each tree in the forest when predicting using the permutated variable
  - Average the loss across trees to obtain the score

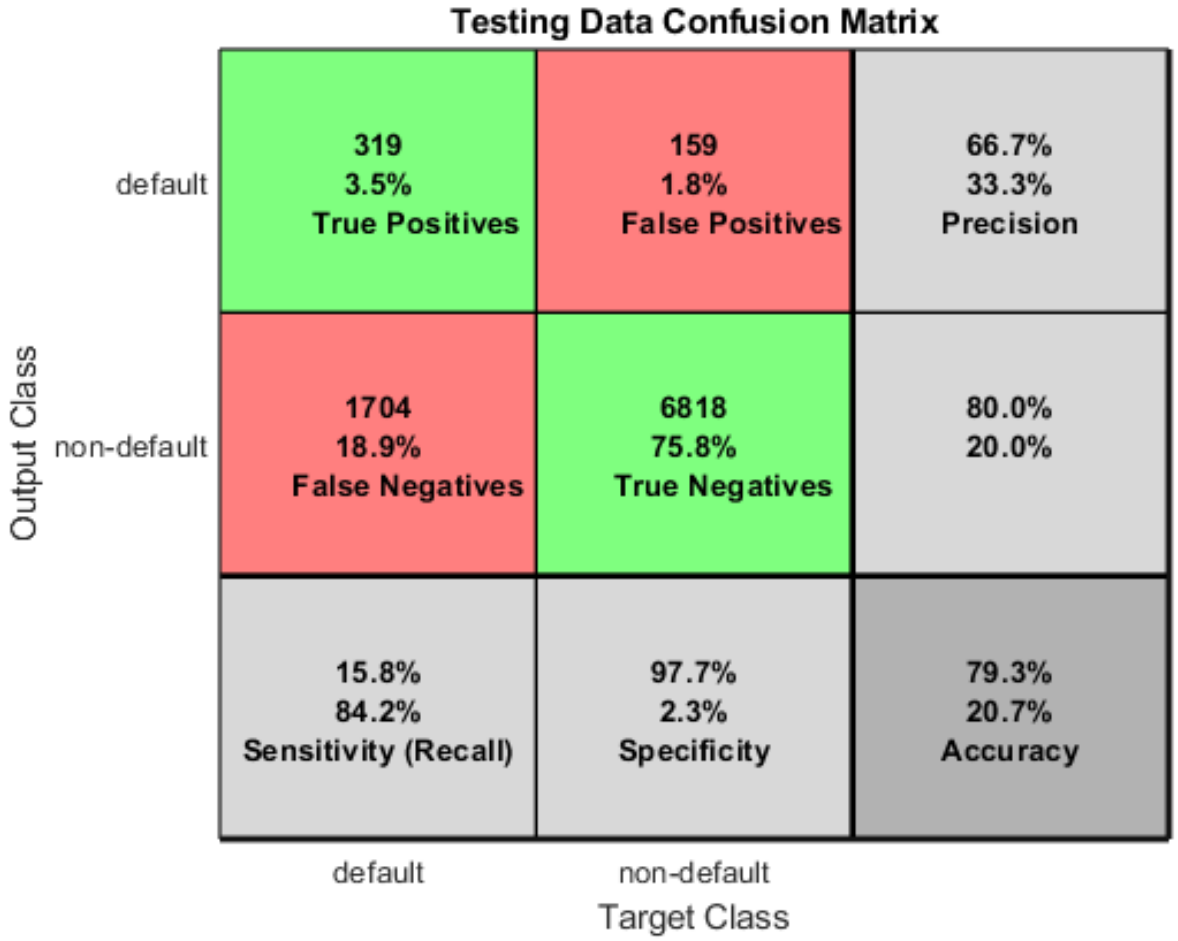
Feature Importance – OOB permutation



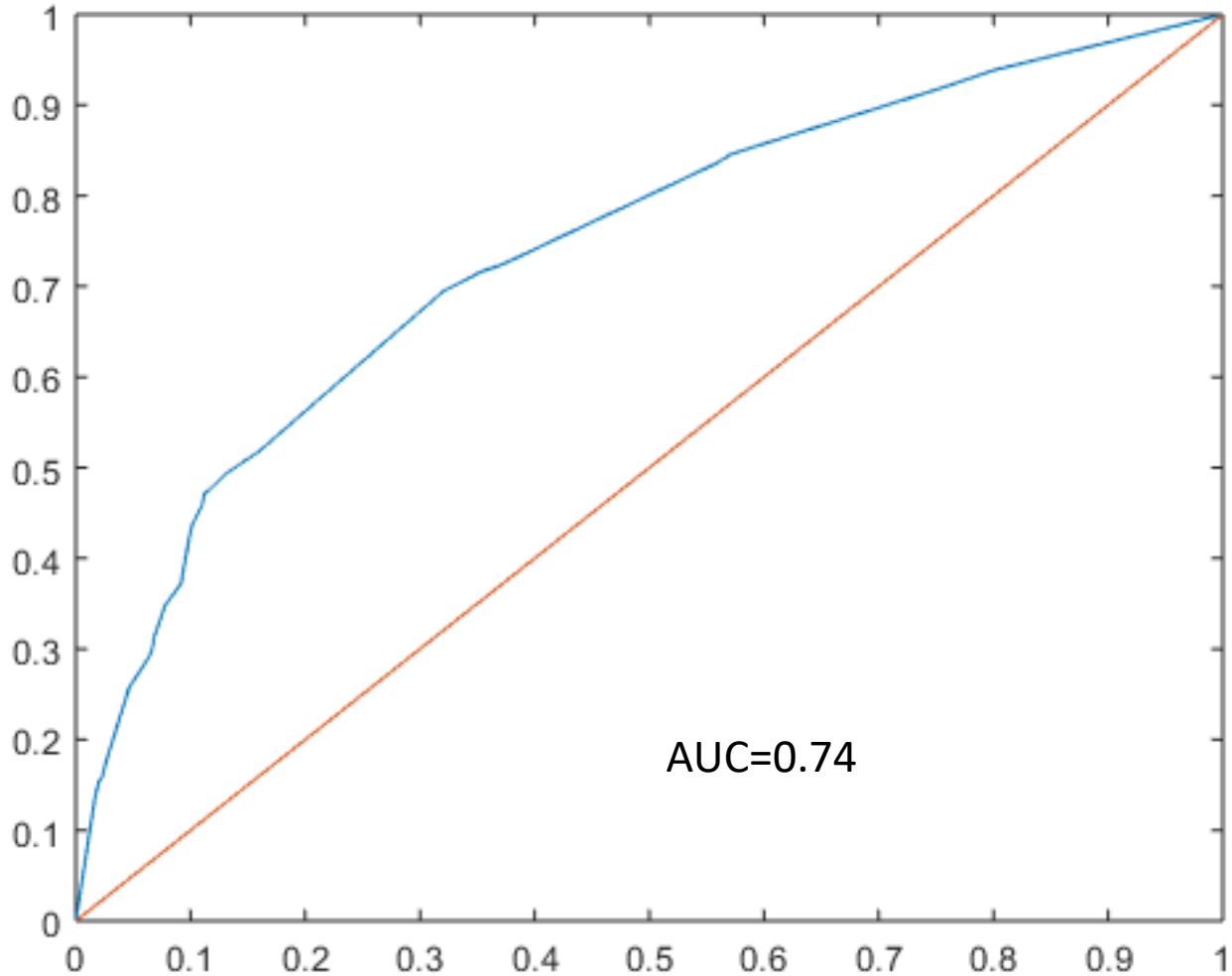
Random Forest – performance evaluation

- Confusion matrix
- Precision
- Accuracy
- Sensitivity (recall)
- Specificity
- The Receiver Operating Characteristic (ROC) curve
- Area under the curve

Random Forest – performance evaluation



Random Forest – performance evaluation



Hands-on