

Bagging and Random Forests

ADVANCED COMPUTING AND DIGITAL TECHNOLOGY RESEARCH CENTER

Pattern Recognition and Machine Learning



Lecture 13 Bagging and Random Forests

Chizhi Chris ZHANG

zhangchizhi@ciomp.ac.cn

Advanced Computing and Digital Technology Research Center

University of Chinese Academy of Sciences

Autumn 2026

Lecture plan and time allocation

ADVANCED COMPUTING AND DIGITAL TECHNOLOGY RESEARCH CENTER

Topic	Minutes
Bagging	45
Random forests and OOB evaluation	45
Tuning and the link to boosting	45

Several trees predicting the same case

Measure shared errors; construct bootstrap learners; evaluate the right predictions.

The numerical steps are shown on the slides.



A tree can fit well and still be unstable

ADVANCED COMPUTING AND DIGITAL TECHNOLOGY RESEARCH CENTER

We first calculate how independent votes and correlated numeric predictions combine. Bootstrap samples then provide a practical way to train different learners for the same task.

In a repair-cost regression illustration (costs in hundreds of currency units), a deep tree can change its first split when a small number of training cases change. This is sensitivity to the training sample, not merely a large training error.

Training sample	Training MSE	Validation MSE	x_0 : predicted repair cost
A	0.00	0.22	18
B	0.00	0.24	30
C	0.00	0.21	12

Good learners also need different mistakes

ADVANCED COMPUTING AND DIGITAL TECHNOLOGY RESEARCH CENTER

Read the figure. Each column is one test case; each row is a classifier. Count correct votes in each column before inspecting the ensemble row.

测试 测试 测试				测试 测试 测试				测试 测试 测试			
例 1 例 2 例 3				例 1 例 2 例 3				例 1 例 2 例 3			
f_1	✓	✓	✗	f_1	✓	✓	✗	f_1	✓	✗	✗
f_2	✗	✓	✓	f_2	✓	✓	✗	f_2	✗	✓	✗
f_3	✓	✗	✓	f_3	✓	✓	✗	f_3	✗	✗	✓
集成	✓	✓	✓	集成	✓	✓	✗	集成	✗	✗	✗

(a) 集成提升性能

(b) 集成不起作用

(c) 集成起负作用



Three independent votes: calculate the benefit

ADVANCED COMPUTING AND DIGITAL TECHNOLOGY RESEARCH CENTER

For three binary classifiers, assume equal error probability $q = 0.30$ and independent errors on a test case. Majority voting fails when at least two classifiers fail.

$$q = 0.30$$

$$\begin{aligned} P(\text{wrong majority}) &= \binom{3}{2} q^2 (1 - q) + q^3 \\ &= 3(0.3)^2(0.7) + (0.3)^3 = 0.216. \end{aligned}$$



Separate bias, variance, and observation noise

ADVANCED COMPUTING AND DIGITAL TECHNOLOGY RESEARCH CENTER

For squared-error regression at a fixed input, compare predictions over repeated training samples. The mean prediction may miss the true function; individual predictions may fluctuate around that mean.

$$\mathbb{E}[(y - \hat{f}(x))^2] = \underbrace{(f(x) - \mathbb{E}\hat{f}(x))^2}_{\text{bias squared}} + \underbrace{\text{Var}(\hat{f}(x))}_{\text{sample sensitivity}} + \underbrace{\sigma_\epsilon^2}_{\text{noise}}.$$

$$\text{bias} = 2 \quad \text{variance} = 9$$

$$\text{Noise variance} = 4 \quad 4 + 9 + 4 = 17$$



The average contains diagonal and cross terms

ADVANCED COMPUTING AND DIGITAL TECHNOLOGY RESEARCH CENTER

Fix one input. Four fitted numeric predictions each have variance 16. Averaging uses all four outputs with weight 1/4.

Case 1: zero pairwise covariance

$$\text{Var}(\bar{f}) = \frac{1}{16} [16 + 16 + 16 + 16] = 4.$$

More generally,

$$\bar{f} = \frac{1}{B} \sum_b f_b, \quad \text{Var}(\bar{f}) = \frac{\sigma^2}{B}.$$

This result needs the covariance terms to be zero.

The learner count is unchanged. Shared fluctuations explain the difference between 4 and 10.

Case 2: pairwise correlation .5

Each covariance is $.5(16) = 8$. There are 4 diagonal terms and $4(3) = 12$ ordered cross terms.

$$\text{Var}(\bar{f}) = \frac{1}{16} \left[\sum_i \text{Var}(f_i) + \sum_{i \neq j} \text{Cov}(f_i, f_j) \right].$$

$$\text{Var}(\bar{f}) = \frac{4(16) + 12(8)}{16} = 10.$$

Correlation leaves a floor under the variance

ADVANCED COMPUTING AND DIGITAL TECHNOLOGY RESEARCH CENTER

Key idea

Now assume equal member variance σ^2 and equal pairwise correlation ρ . There are B variance terms and $B(B - 1)$ covariance terms.

$$\text{Var}(\bar{f}) = \frac{B\sigma^2 + B(B-1)\rho\sigma^2}{B^2} = \sigma^2 \left[\rho + \frac{1-\rho}{B} \right].$$

$$\sigma^2 = 16 \quad \rho = 0.5$$

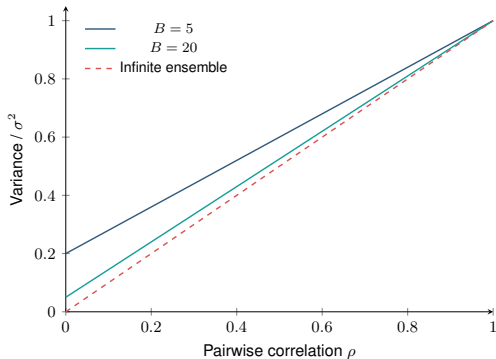
$$B = 4 \quad 16(0.5 + 0.5/4) = 10$$

$$\rho\sigma^2 = 8$$



Compare tree counts at the same correlation

ADVANCED COMPUTING AND DIGITAL TECHNOLOGY RESEARCH CENTER



The horizontal axis is pairwise correlation. Compare different tree counts at the same correlation.



Choose between more trees and less correlation

ADVANCED COMPUTING AND DIGITAL TECHNOLOGY RESEARCH CENTER

Compare two ways to improve an ensemble with member variance 16. Increasing B helps the independent part; reducing ρ changes the limiting variance.

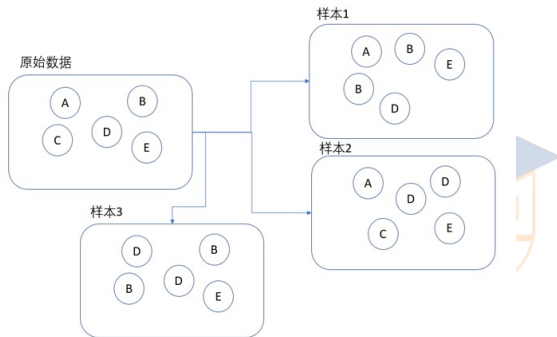
Correlation	10 trees	100 trees	Limit
$\rho = 0.5$	8.80	8.08	8.00
$\rho = 0.2$	4.48	3.33	3.20



Bootstrap samples reuse some rows and omit others

ADVANCED COMPUTING AND DIGITAL TECHNOLOGY RESEARCH CENTER

Start with A, B, C, D, E. Each sample contains five draws with replacement. In sample 1, B appears twice and C is absent. In sample 2, D appears twice and B is absent.



Duplicates act like observation multiplicities

ADVANCED COMPUTING AND DIGITAL TECHNOLOGY RESEARCH CENTER

For one bootstrap sample, store the number of times each original observation is drawn. Repeated rows can be represented by these counts when the learner supports equivalent sample weights.

Original row	A	B	C	D	E
Draw count	1	2	0	1	1
Share of draws	0.2	0.4	0	0.2	0.2

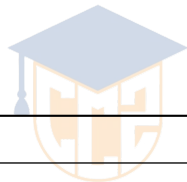
Why about 36.8 percent of rows are out of bag

ADVANCED COMPUTING AND DIGITAL TECHNOLOGY RESEARCH CENTER

Key idea

With N independent draws from N rows, a fixed row is missed once with probability $1 - 1/N$. To be OOB, it must be missed on every draw.

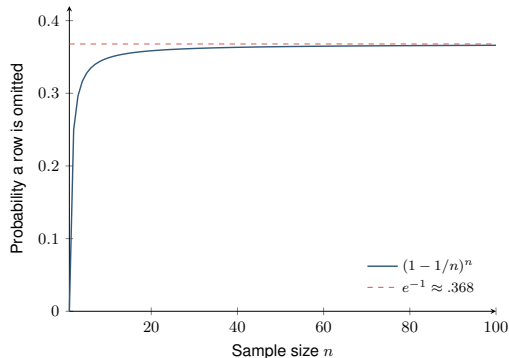
$$P(\text{row is OOB}) = \left(1 - \frac{1}{N}\right)^N \rightarrow e^{-1} \approx 0.368.$$



Dataset size	OOB probability	Expected distinct rows
$N = 5$	$(4/5)^5 = 0.32768$	$5(1 - 0.32768) = 3.36$
N Larger	about 0.368	approximately $0.632N$

How often is one row omitted from a bootstrap?

ADVANCED COMPUTING AND DIGITAL TECHNOLOGY RESEARCH CENTER



A fixed row is omitted on one draw with probability $1 - 1/N$. Independent draws multiply, giving $(1 - 1/N)^N$ after N draws. The curve approaches $e^{-1} \approx .368$; finite samples need not equal this limit.



Activity: label the in-bag and OOB rows

ADVANCED COMPUTING AND DIGITAL TECHNOLOGY RESEARCH CENTER

Original data contain rows A through F. Two learners receive the bootstrap draws below. The table identifies omitted rows.

$$D_1 = (A, A, C, D, D, F), \quad D_2 = (B, C, C, E, F, F).$$

Tree 1 omits $\{B, E\}$; tree 2 omits $\{A, D\}$. Those rows were excluded from the corresponding tree's training.



Regression averages values, including their common bias

ADVANCED COMPUTING AND DIGITAL TECHNOLOGY RESEARCH CENTER

For each learner b , draw a bootstrap dataset D_b and fit the same learning algorithm. Every learner solves the original task; no learner is trained specifically to repair the previous one.

$b = 1, \dots, B$

Three trees estimate a repair cost as 80, 110, and 95. Bagging gives equal weight to these numeric predictions.

$$\hat{y} = \frac{80 + 110 + 95}{3} = 95.$$



Classification can combine labels or probabilities

ADVANCED COMPUTING AND DIGITAL TECHNOLOGY RESEARCH CENTER

Bootstrap trees may still repeatedly choose the same dominant feature.

Hard voting counts predicted class labels. Soft voting averages class probabilities and then chooses a class or applies a decision threshold. Neither operation changes how the trees were trained.

$$\hat{y}_{\text{hard}} = \text{mode}\{\hat{y}_b\}, \quad \bar{p}_k = \frac{1}{B} \sum_b p_{b,k}, \quad \hat{y}_{\text{soft}} = \arg \max_k \bar{p}_k.$$



The same three trees can produce different decisions

ADVANCED COMPUTING AND DIGITAL TECHNOLOGY RESEARCH CENTER

Hard voting predicts failure, two votes to one. Probability averaging predicts normal. Choose using probability quality and the intended decision rule.

Tree	Failure probability	Hard class
1	.51	Failure
2	.51	Failure
3	.01	Normal

$$\bar{p} = (.51 + .51 + .01)/3 \approx .343.$$



A tree probability comes from its reached leaf

ADVANCED COMPUTING AND DIGITAL TECHNOLOGY RESEARCH CENTER

A classification tree routes the input to one leaf. A simple probability estimate is the class fraction among the training observations reaching that leaf.

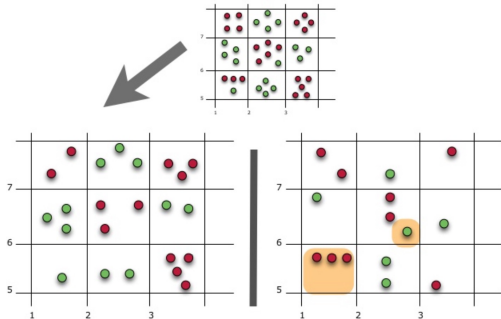
$$p_b(\text{Failure} \mid x) = \frac{n_{\text{Failure, leaf}}}{n_{\text{leaf}}}.$$



Out-of-bag evaluation uses the omitted observations

ADVANCED COMPUTING AND DIGITAL TECHNOLOGY RESEARCH CENTER

OOBE



The top panel is the original data. A bootstrap learner is fitted using the selected observations in the lower-left panel; the lower-right observations were omitted from that learner.



Calculate one OOB probability and one OOB error

ADVANCED COMPUTING AND DIGITAL TECHNOLOGY RESEARCH CENTER

The table marks whether each tree used each row. For an OOB prediction, average only the trees that omitted that particular row. The eligible set changes from row to row.

Rows	Tree 1	Tree 2	Tree 3	Eligible trees
A	in	OOB	OOB	2, 3
B	OOB	in	OOB	1, 3
C	in	in	OOB	3

For A, tree 1 trained on A and gives probability 0.95; trees 2 and 3 omitted A and give 0.60 and 0.20. A is truly normal.

Worked calculation

$$\hat{p}_{\text{OOB}}(A) = \frac{0.60 + 0.20}{2} = 0.40.$$

With 4 mistakes among 20 OOB predictions, the error rate is $= 4/20 = 0.20$

OOB and cross-validation answer related questions

ADVANCED COMPUTING AND DIGITAL TECHNOLOGY RESEARCH CENTER

For large N , one tree omits a fixed row with probability about 0.368. With B independent bootstrap draws, its expected eligible-tree count is approximately $0.368B$.

$$\mathbb{E}[B_{\text{OOB}}] = 0.368B, \quad P(B_{\text{OOB}} = 0) \approx (0.632)^B.$$

$B = 10$ gives about 3.68 eligible OOB trees per row on average; $B = 100$ gives about 36.8 eligible trees.

OOB evaluates rows using trees that omitted those rows. Cross-validation refits the entire chosen learning procedure on separate training folds. They are useful estimates, but neither removes leakage automatically.

Choose a new feature subset at each node

ADVANCED COMPUTING AND DIGITAL TECHNOLOGY RESEARCH CENTER

Suppose temperature is much more predictive than every other sensor. Most bootstrap trees will choose temperature near the root, then make similar errors. Their samples differ, but their structures remain strongly related.

There are six features: temperature, vibration, current, humidity, age, and load. Let each node consider two randomly chosen features.

vibration < 2.1 Temperature < 70

Age < 8



Still choose the best split among the allowed features

ADVANCED COMPUTING AND DIGITAL TECHNOLOGY RESEARCH CENTER

A node contains 6 failures and 4 normal cases, so its Gini impurity is $1 - 0.6^2 - 0.4^2 = 0.48$. Only vibration and age are allowed at this node.

$$1 - 0.6^2 - 0.4^2 = 0.48$$

Candidate split	Child class counts (F,N)	Weighted child Gini	Reduction
vibration	(4,0) and (2,4)	0.267	0.213
age	(3,2) and (3,2)	0.480	0.000

$$0.48 - [4/10 \cdot 0 + 6/10 \cdot (1 - (2/6)^2 - (4/6)^2)] = 0.213$$

How often does a useful feature enter the candidates?

ADVANCED COMPUTING AND DIGITAL TECHNOLOGY RESEARCH CENTER

Choose two candidate features without replacement. Count all unordered pairs, then count pairs containing the feature of interest. This separates candidate availability from whether its split will actually win.

$$\binom{6}{2} = \frac{6(5)}{2} = 15 \quad \text{pairs in total .}$$

$$\binom{4}{2} = \frac{4(3)}{2} = 6,$$

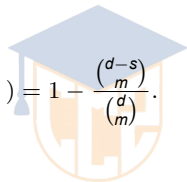
$$P(\text{ miss all useful }) = 6/15 = .4.$$

$$P(\text{ at least one useful feature }) = 1 - .4 = .6.$$

With d available features and m candidates sampled without replacement, one particular feature is available with probability m/d . For s useful features, count candidate sets that miss all of them.

$$P(\text{ at least one useful feature }) = 1 - \frac{\binom{d-s}{m}}{\binom{d}{m}}.$$

$d = 6$	$s = 2$
$m = 2$	$1 - 6/15 = 0.60$
$m = 3$	$1 - 4/20 = 0.80$



Useful randomness balances strength and similarity

ADVANCED COMPUTING AND DIGITAL TECHNOLOGY RESEARCH CENTER

The table illustrates a controlled comparison using the same training split, tree count, leaf settings, and evaluation data. Only the candidate-feature fraction changes.

Candidate fraction	Single-tree error	Error correlation	Forest error
All features	0.20	0.80	0.18
Half	0.23	0.45	0.13
Very small	0.36	0.15	0.24

Extra Trees randomizes split thresholds too

ADVANCED COMPUTING AND DIGITAL TECHNOLOGY RESEARCH CENTER

Read the arguments. 300 trees average many perturbed learners; square-root candidates add feature diversity; five observations per leaf restrain very small local estimates.

```
from sklearn.ensemble import RandomForestClassifier
forest = RandomForestClassifier(
    n_estimators=300, max_features="sqrt",
    min_samples_leaf=5, bootstrap=True,
    oob_score=True, random_state=13, n_jobs=-1)
forest.fit(X_train, y_train)
p_valid = forest.predict_proba(X_valid)[: , 1]
```

A random forest usually searches useful thresholds within randomly selected features. Extra Trees proposes random thresholds and chooses a good split among those proposals. Common implementations may also use different bootstrap defaults.

Add trees until the prediction and score stabilize

ADVANCED COMPUTING AND DIGITAL TECHNOLOGY RESEARCH CENTER

Bagging learners are peers whose predictions are averaged.

A practical start. Set enough trees for stable results; then compare a small grid of feature and leaf settings using the task metric. Change one mechanism at a time when learning how it affects the model.

Unlike making each tree deeper, adding trees mostly refines the average over the specified random-tree procedure. It increases training, storage, and prediction cost.

Trees	OOB error	Validation log loss	Prediction time
50	0.145	0.410	1 unit
200	0.137	0.398	4 units
800	0.136	0.397	16 units

Leaf size changes probability support and fit

ADVANCED COMPUTING AND DIGITAL TECHNOLOGY RESEARCH CENTER

Compare two settings on the same data split. Accuracy measures decisions at a threshold; log loss also penalizes confident wrong probabilities.

Minimum leaf size	Train accuracy	Validation accuracy	Validation log loss
1	1.00	0.89	0.47
10	0.95	0.90	0.35



Rare failures make accuracy an incomplete answer

ADVANCED COMPUTING AND DIGITAL TECHNOLOGY RESEARCH CENTER

Suppose each device contributes 20 measurements. A random row split lets the same device appear in training and validation. The model may recognize device-specific patterns instead of learning to generalize to a new device.

Out of 1,000 devices, 20 fail. Predicting every device as normal gives 98% accuracy and detects no failure. A forest threshold yields the following illustrative counts.

Actual / predicted	Failure	Normal
Failure	15	5
Normal	35	945

$$\text{recall} = 15/20 = 0.75, \quad \text{precision} = 15/50 = 0.30.$$

Class weights change the training impurity

ADVANCED COMPUTING AND DIGITAL TECHNOLOGY RESEARCH CENTER

If failures receive weight 5 and normal cases weight 1, a leaf with 2 failures and 8 normal cases has weighted class mass 10 versus 8.

$$p_{\text{weighted}}(F) = \frac{5 \cdot 2}{5 \cdot 2 + 1 \cdot 8} = \frac{10}{18}, \quad G = 1 - (10/18)^2 - (8/18)^2 \approx 0.494.$$

Unweighted failure fraction = $2/10 = 0.20$

Weighted leaf fractions describe the weighted training objective. Validate or calibrate population probabilities before applying a probability-based cost threshold.



A decision threshold follows the error costs

ADVANCED COMPUTING AND DIGITAL TECHNOLOGY RESEARCH CENTER

Let p be a calibrated failure probability. A false alarm costs $C_{FP} = 1$ unit, while a missed failure costs $C_{FN} = 9$ units. Correct decisions cost zero.

$$C_{FP} = 1 \quad C_{FN} = 9$$

$$\text{Cost(flag)} = (1 - p)C_{FP}, \quad \text{Cost(normal)} = pC_{FN}.$$

$$\text{Flag when } p > \frac{C_{FP}}{C_{FP} + C_{FN}} = 0.10.$$

$$p = 0.20 : \text{Cost(flag)} = 0.8, \text{Cost(normal)} = 1.8 \Rightarrow \text{triggers an alarm}$$



Impurity importance adds weighted split improvements

ADVANCED COMPUTING AND DIGITAL TECHNOLOGY RESEARCH CENTER

For a split at node t , measure how much weighted child impurity falls below the parent impurity. Weight that reduction by the fraction of training observations reaching the node.

$$I_j \propto \sum_{t: \text{split feature} = j} \frac{n_t}{N} \left[G_t - \frac{n_L}{n_t} G_L - \frac{n_R}{n_t} G_R \right].$$

$$0.4 \times 0.2 = 0.08$$



Many possible thresholds can inflate impurity importance

ADVANCED COMPUTING AND DIGITAL TECHNOLOGY RESEARCH CENTER

A continuous measurement or an identifier with many distinct values offers many candidate splits. Even a useless feature can achieve an apparent training improvement by chance when enough thresholds are tried.

Condition to check

A high training importance for an identifier may reflect memorization. Check permutation results on independent validation data to examine predictive dependence.

Feature	Candidate-split opportunity	Interpretation check
Binary indicator	One nontrivial threshold	A small opportunity to fit chance variation.
Many distinct measurements	Many candidate thresholds	Training improvement can benefit from the larger search.
Almost unique identifier	Many ways to isolate observations	Inspect memorization and independent validation dependence.

Correlated features can substitute for each other

ADVANCED COMPUTING AND DIGITAL TECHNOLOGY RESEARCH CENTER

First score the fitted forest on validation data. Then shuffle one feature column across validation rows, keeping labels and other columns fixed. Rescore without retraining.

Validation input	Log loss	Increase over baseline
Unchanged	0.32	0.00
Shuffle temperature	0.48	0.16
Shuffle vibration	0.39	0.07
Shuffle humidity	0.321	0.001

Temperature sensor A and sensor B measure nearly the same quantity. The forest may substitute one for the other, so shuffling a single sensor can produce only a small loss increase.

Feature selection must be fitted inside the split

ADVANCED COMPUTING AND DIGITAL TECHNOLOGY RESEARCH CENTER

```
from sklearn.pipeline import make_pipeline
from sklearn.impute import SimpleImputer
from sklearn.ensemble import RandomForestClassifier
model = make_pipeline(
    SimpleImputer(strategy="median"),
    RandomForestClassifier(n_estimators=300,
                          max_features="sqrt", min_samples_leaf=5,
                          random_state=13))
model.fit(X_train, y_train)
```

Keep the order. Fit imputation and the forest on training data; evaluate on validation data. If importance is used to select columns, repeat that selection inside each training fold.



Case study: use the evidence to choose a forest

ADVANCED COMPUTING AND DIGITAL TECHNOLOGY RESEARCH CENTER

All rows below use device-grouped validation and the same preprocessing. The figures are illustrative. The goal is useful failure probabilities with limited inspection capacity.

Model	recall	precision	Log loss
Deep single tree	0.60	0.23	0.61
300 bagged trees	0.70	0.28	0.42
300-tree random forest	0.75	0.30	0.35

A preview of changing observation weights

ADVANCED COMPUTING AND DIGITAL TECHNOLOGY RESEARCH CENTER

Train each learner from a randomized version of the training data. Learner $b + 1$ does not require learner b 's predictions. Combine the fitted peers.

The next lecture derives these weights and the classifier coefficient, then completes three rounds on the same five cases.

Case	A	B	C	D	E
True label	+	+	-	-	+
First prediction	+	+	-	+	+
Initial weight	.2	.2	.2	.2	.2
Next normalized weight	.125	.125	.125	.5	.125



Bagging peers average; boosting successors correct

ADVANCED COMPUTING AND DIGITAL TECHNOLOGY RESEARCH CENTER

Idea. Bagging fits every learner to the original target on a perturbed sample. Boosting fits a sequence in which the next learner receives information about the current ensemble error.

$$y = (10, 10), \quad F = (8, 13) \Rightarrow r = y - F = (2, -3).$$



Which forest changes would the evidence justify?

ADVANCED COMPUTING AND DIGITAL TECHNOLOGY RESEARCH CENTER

Question

A forest has tiny leaves, high training accuracy, poor validation log loss and correlated temperature sensors.

Evidence and reasoning

Compare larger leaves under the same split to test whether probabilities improve. Use held-out permutation checks and examine correlated sensors together, since either can substitute for the other. Record both the score change and the interpretation limit.

Check. OOB trees exclude their predicted row, but this does not automatically exclude related devices, future information or leaked preprocessing.