

# Approximating Boosted Decision Trees with Differential Privacy

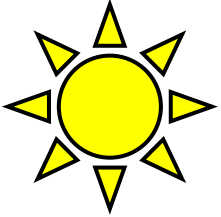
Thorsten Peinemann

My personal website: [tpein160.github.io](https://tpein160.github.io)

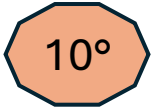


UNIVERSITÄT ZU LÜBECK  
INSTITUTE FOR IT SECURITY  
PRIVACY & SECURITY GROUP

# Temperature prediction



Sun in the last hour, yes/no?



Current temperature in celsius



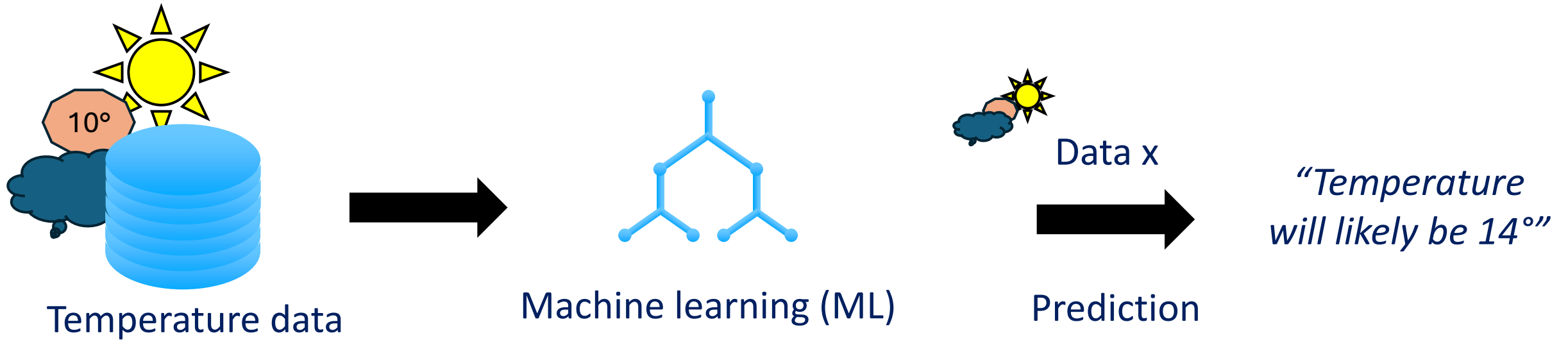
Rain in the last hour, yes/no?



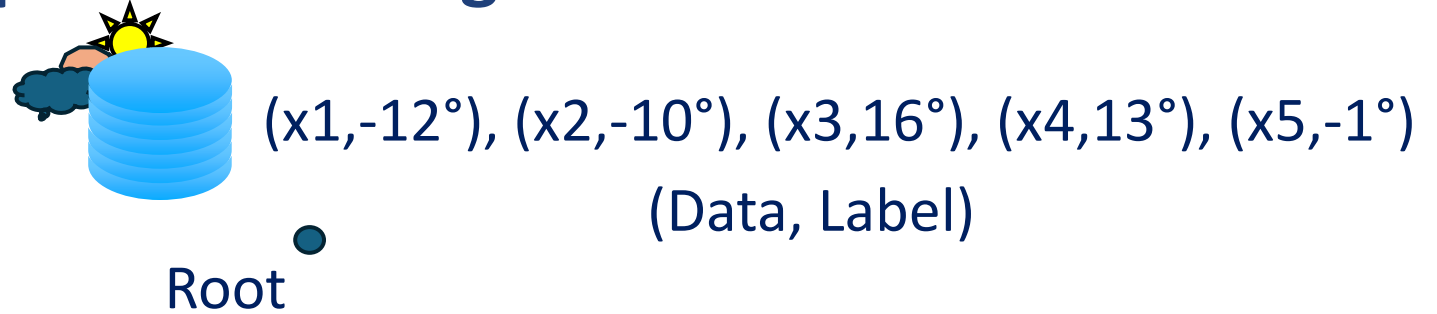
Predict temperature  
in 3 hours



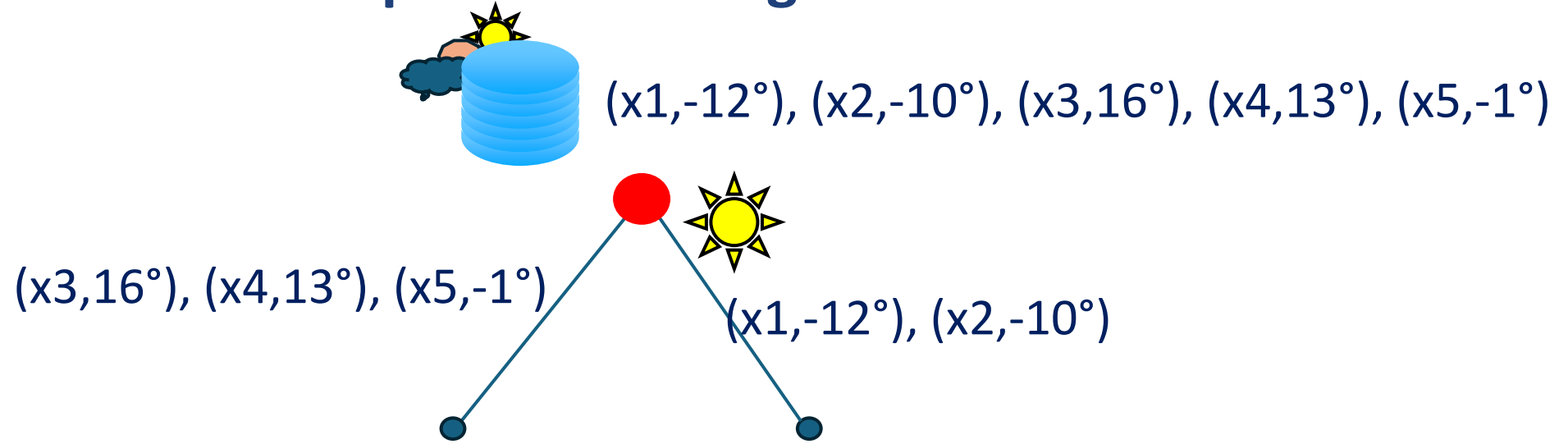
# Temperature prediction using boosted decision tree (BDT) model



# Non-private training of BDT model

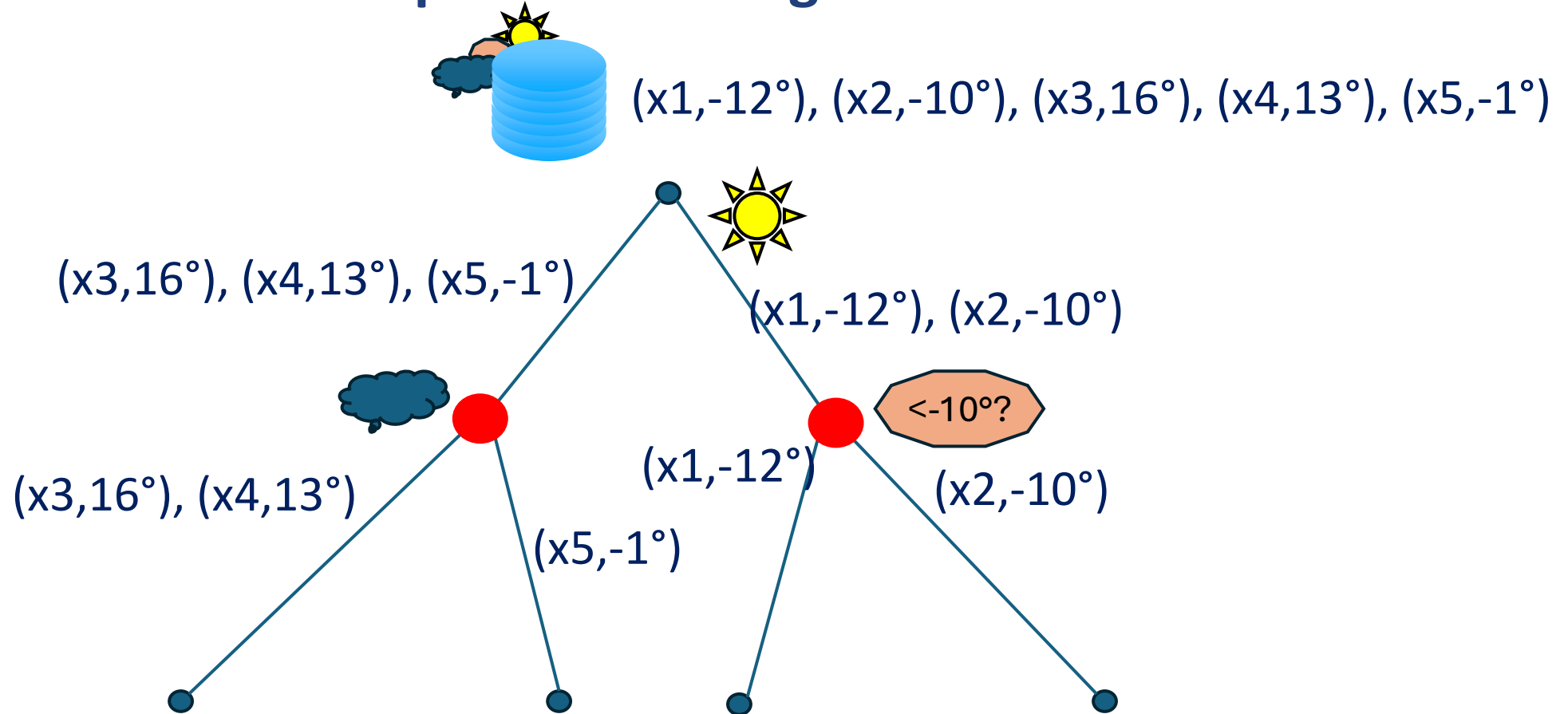


# Non-private training of BDT model



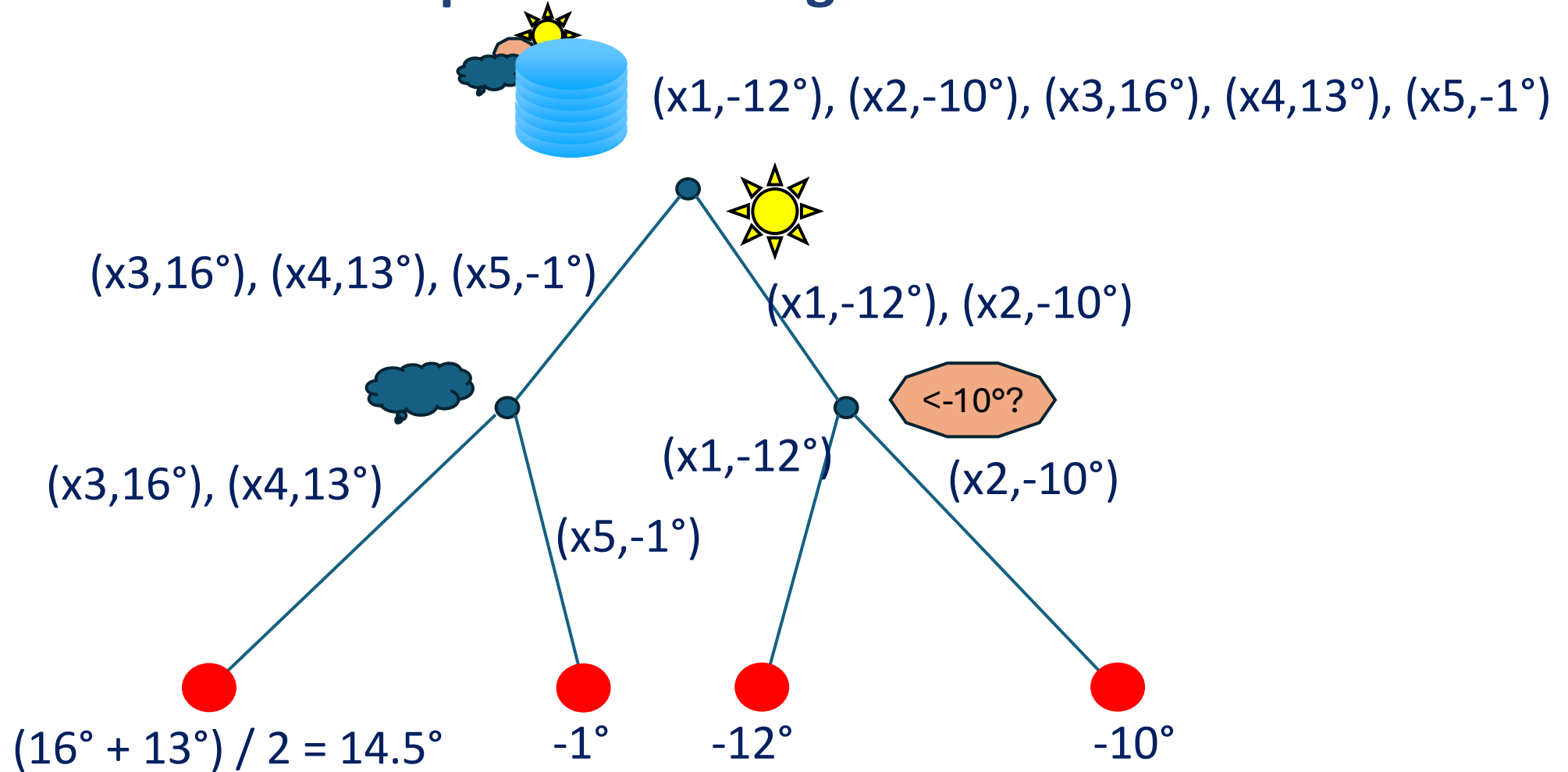
Each node splits the data in two subsets so that each subset groups together alike labels (e.g. gini-coefficient)

# Non-private training of BDT model



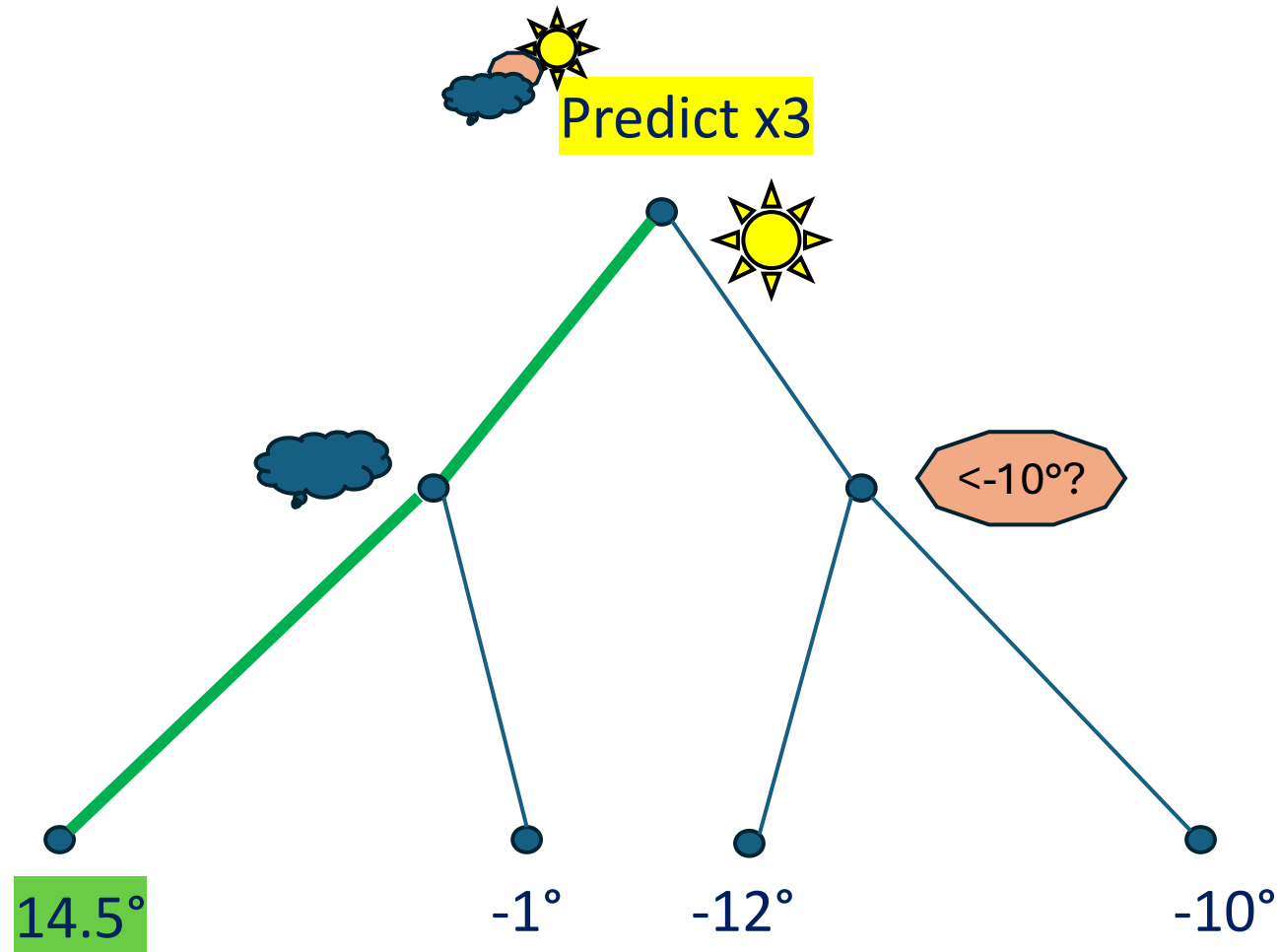
Each node splits the data in two subsets so that each subset groups together alike labels (e.g. gini-coefficient)

# Non-private training of BDT model



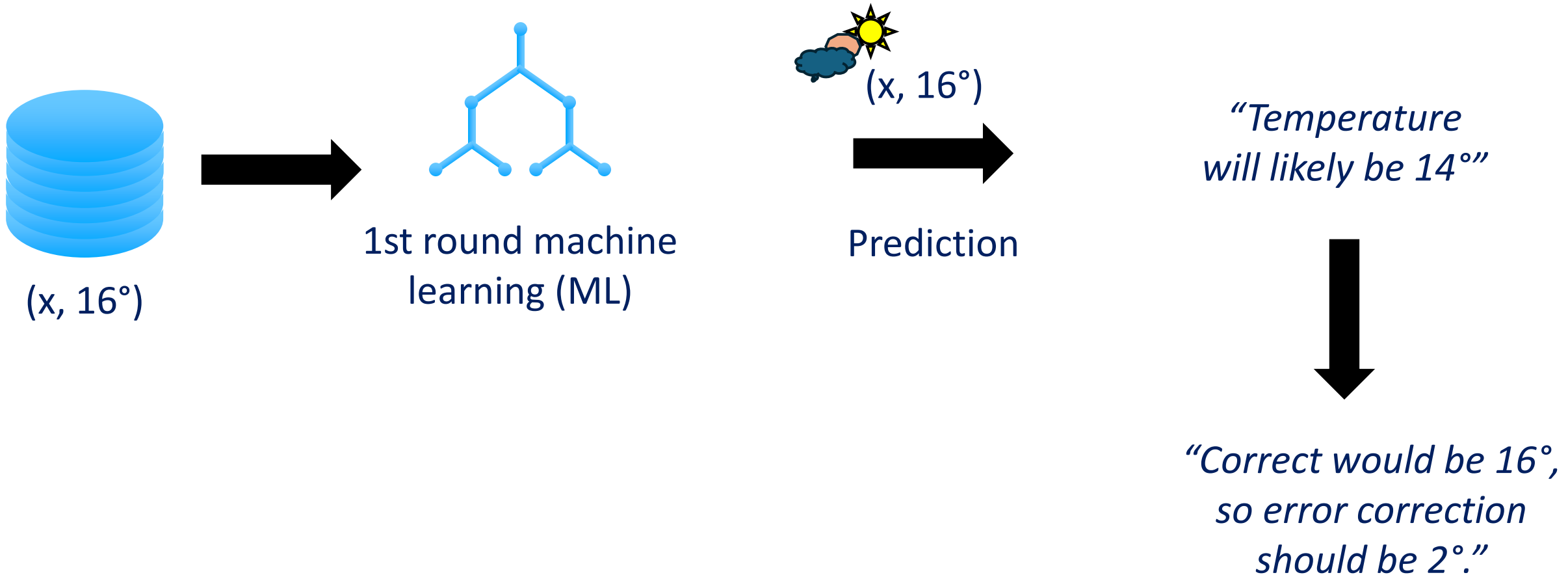
A leaf stores the average label of data points in that leaf

# Prediction of BDT model

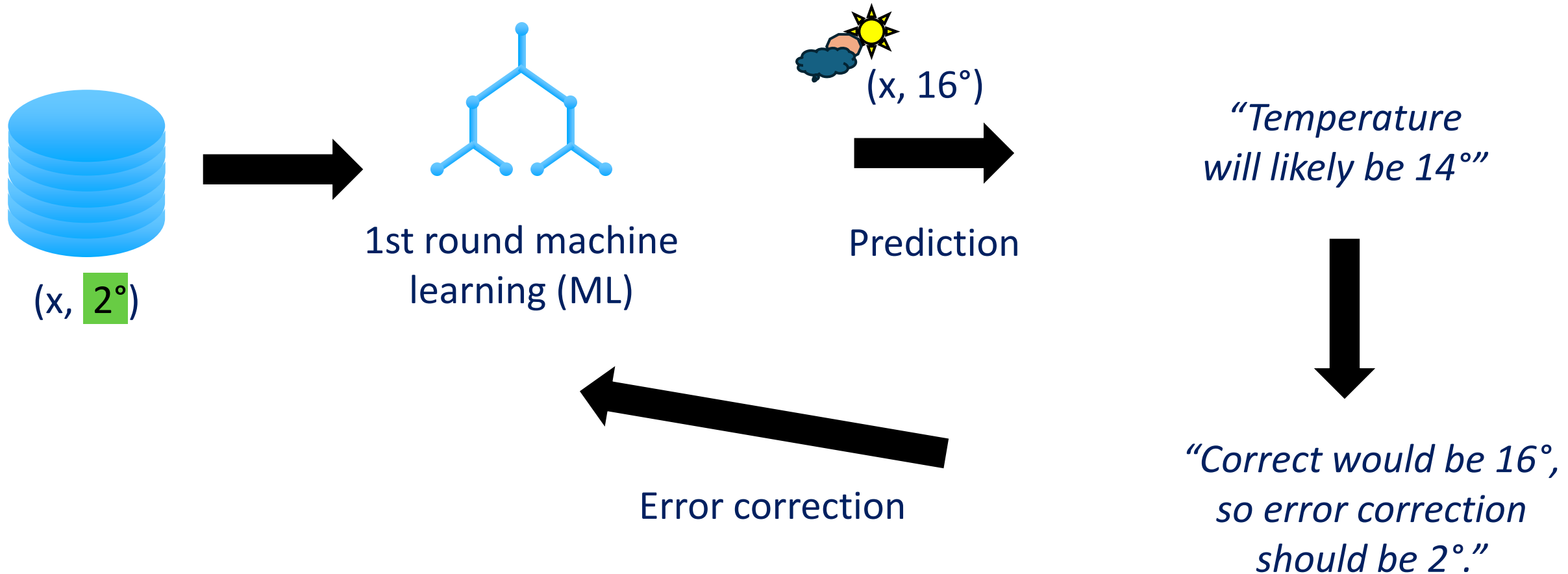




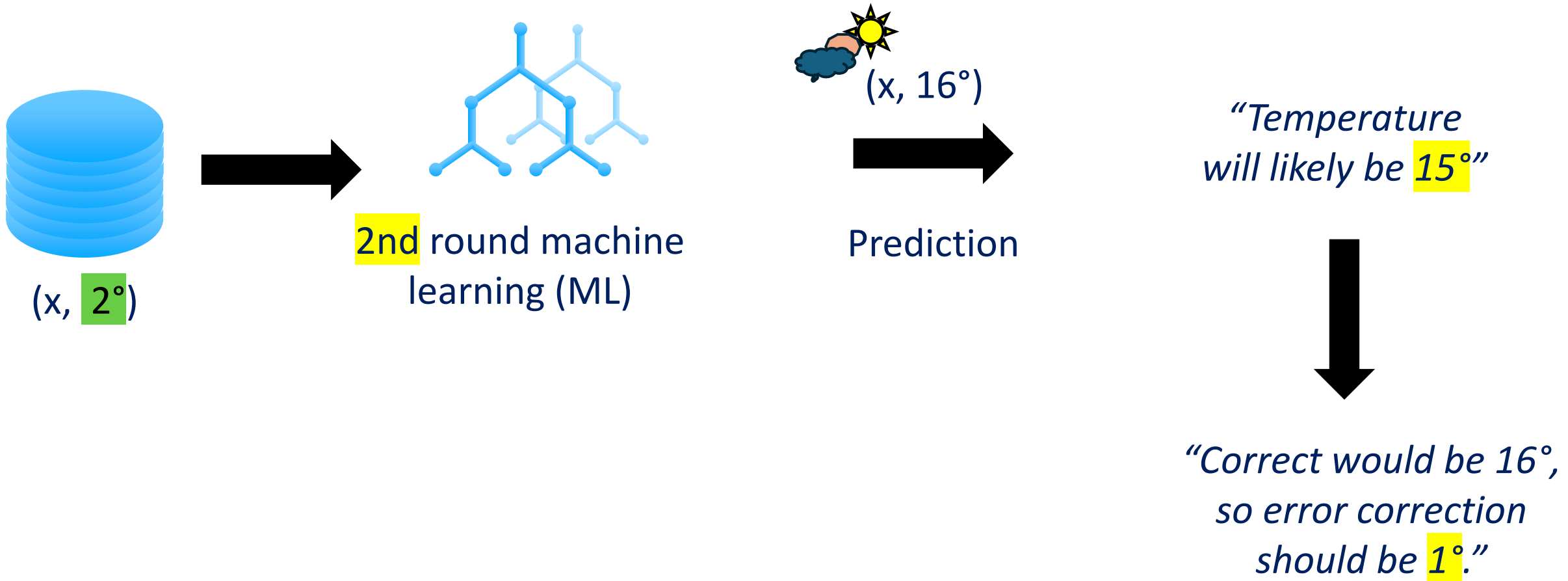
# Error correction for iterative BDT training



# Error correction for iterative BDT training



# Error correction for iterative BDT training

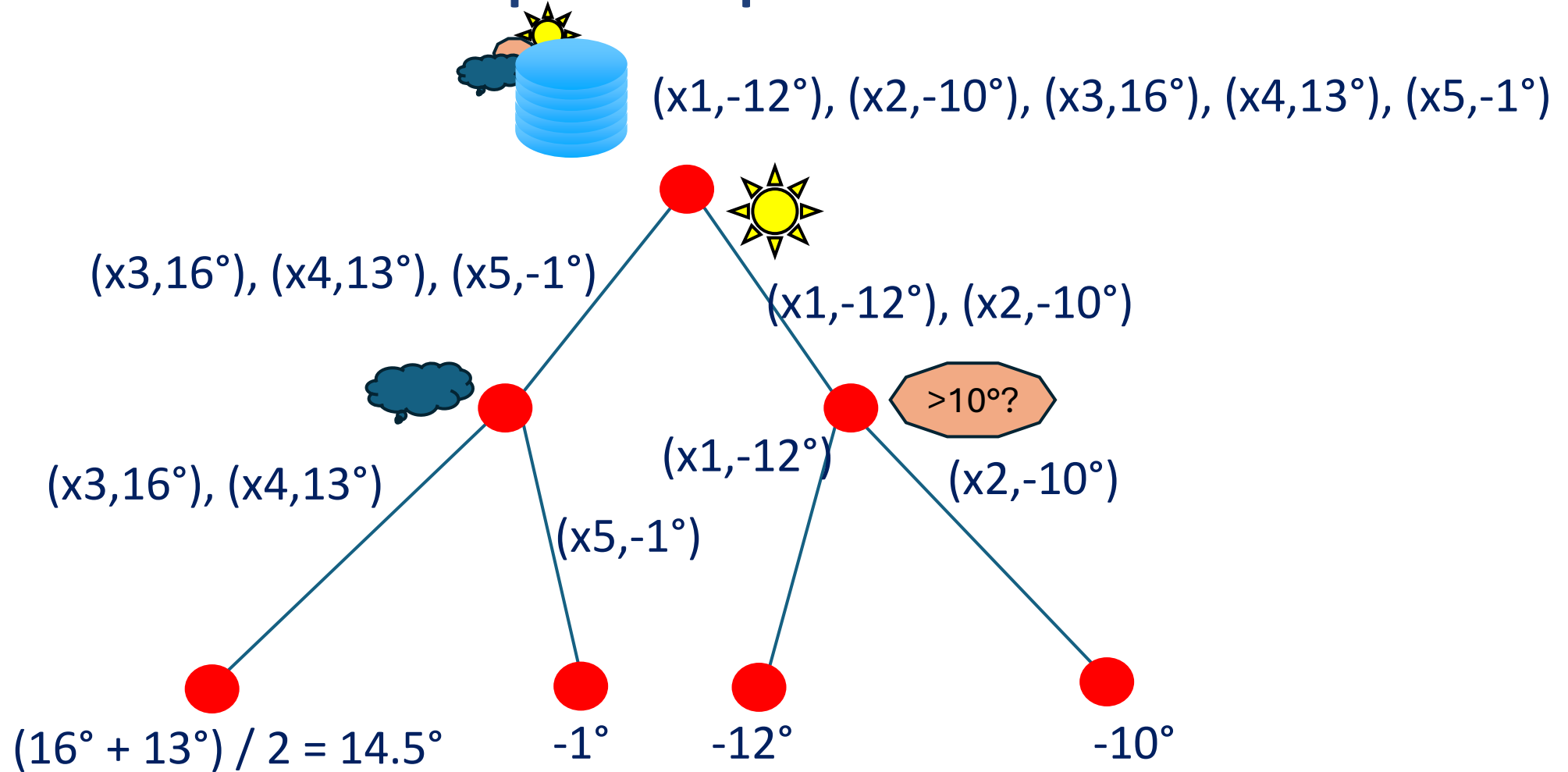


# Differentially private boosted decision trees

*Whence cometh the noise?*



# Data-dependent part of BDT model



# Algorithms for splitting and leaves

## Splitting

From top to bottom:

- Split each node until maximum depth
- Split such that equal error-corrected labels are grouped

## Leaves

For each leaf:

- Find all data points in this leaf
- Sum their error-corrected labels  $\rightarrow S$
- Count number of data points  $\rightarrow C$
- Store  $S/C$



# DP-approximated splitting algorithm

## DP-Splitting

From top to bottom:

Split each node until max depth is reached

Split randomly

## Leaves

For each leaf:

Find all data points in this leaf

Sum their error-corrected labels  $\rightarrow S$

Count number of data points  $\rightarrow C$

Store  $S/C$



# DP-approximated splitting and leaves algorithms

## DP-Splitting

From top to bottom:

Split each node until max depth is reached

Split randomly

## DP-Leaves

For each leaf:

Find all data points in this leaf

Clip their error-corrected labels to length  $L$

Sum clipped error-corrected labels  $\rightarrow S_c$

Add Gaussian noise:  $S_c \rightarrow S'_c$

Count number of data points  $\rightarrow C$

Add Gaussian noise:  $C \rightarrow C'$

Store  $S'_c/C'$





# DP-Proof for DP-approximated splitting and leaves algorithms

## DP-Splitting

Post-processing:

Applying randomized function  
to DP approximated function  
incurs no additional leakage.



# DP-Proof for DP-approximated splitting and leaves algorithms

## DP-Splitting

Post-processing:  
Applying randomized function  
to DP approximated function  
incurs no additional leakage.

## Leaves

$(\epsilon, \delta)$ -Differential Privacy (DP):

$$\Pr[M(x) \in S] \leq e^\epsilon \Pr[M(x') \in S] + \delta$$

Worst case:

$$M(x) = N(0, \sigma^2)$$

$$M(x') = N(L, \sigma^2)$$

M satisfies  $(\epsilon, \delta)$ -DP for any  $\delta > 0$ ,  $\epsilon$  in  $(0,1)$  when  
 $\sigma > \sqrt{2 \cdot \ln(1.25/\delta)} \cdot L/\epsilon$



# Approximating Boosted Decision Trees with Differential Privacy

Thorsten Peinemann

My personal website: [tpein160.github.io](https://tpein160.github.io)



UNIVERSITÄT ZU LÜBECK  
INSTITUTE FOR IT SECURITY  
PRIVACY & SECURITY GROUP