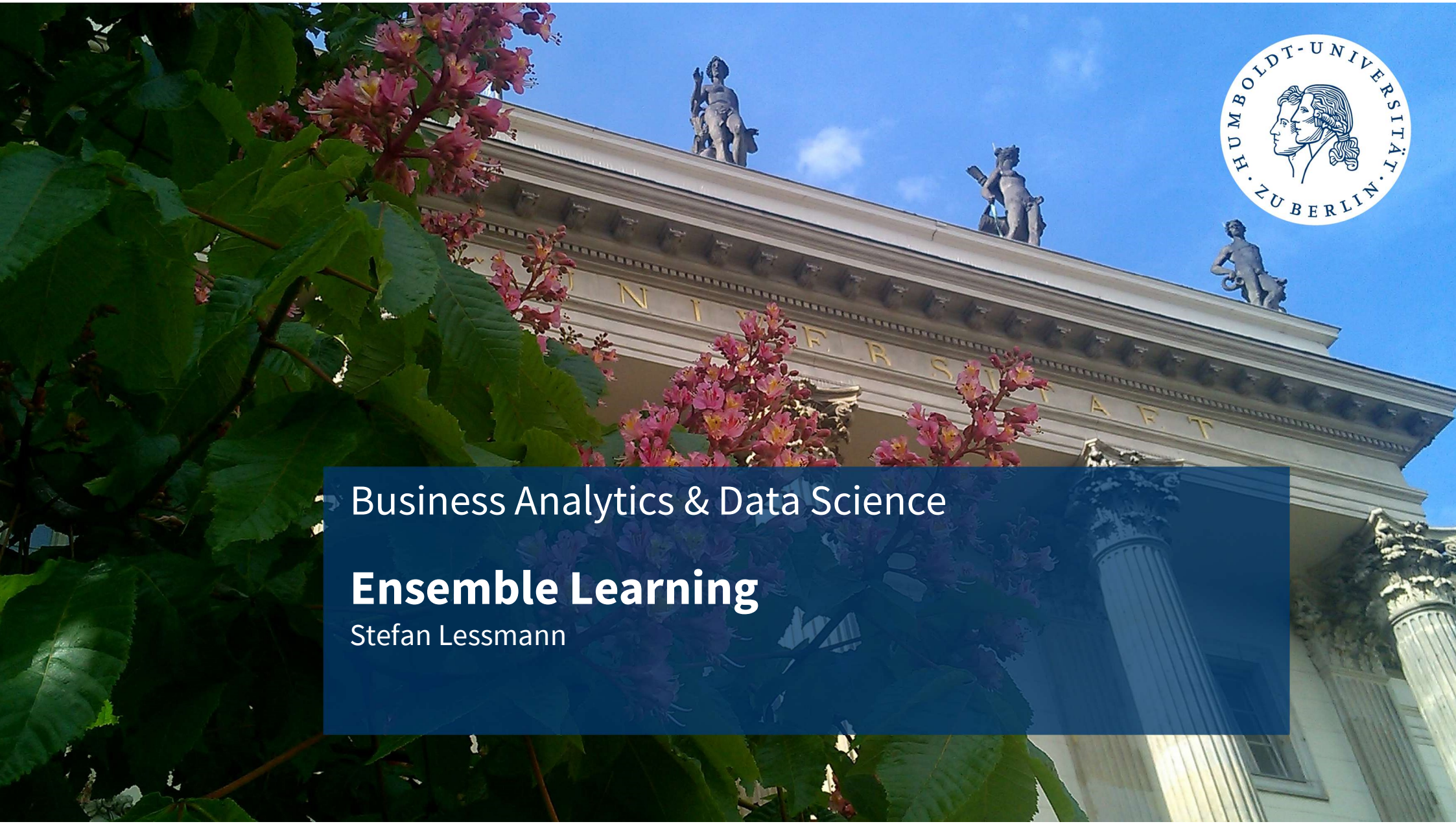


Business Analytics & Data Science

Ensemble Learning

Stefan Lessmann

Agenda

Introduction
Ensemble learning fundamentals, bias-variance trade-off

Bagging and Random Forest
Functioning, meta-parameters, and a little bit of theory

The Boosting Algorithm
Boosting principle and Adaboost

The Gradient Boosting Algorithm
Intuition, gradient descent, state-of-the-art implementations

Heterogeneous Ensemble Learning
Ensemble learning strategies, the stacking algorithm

Summary

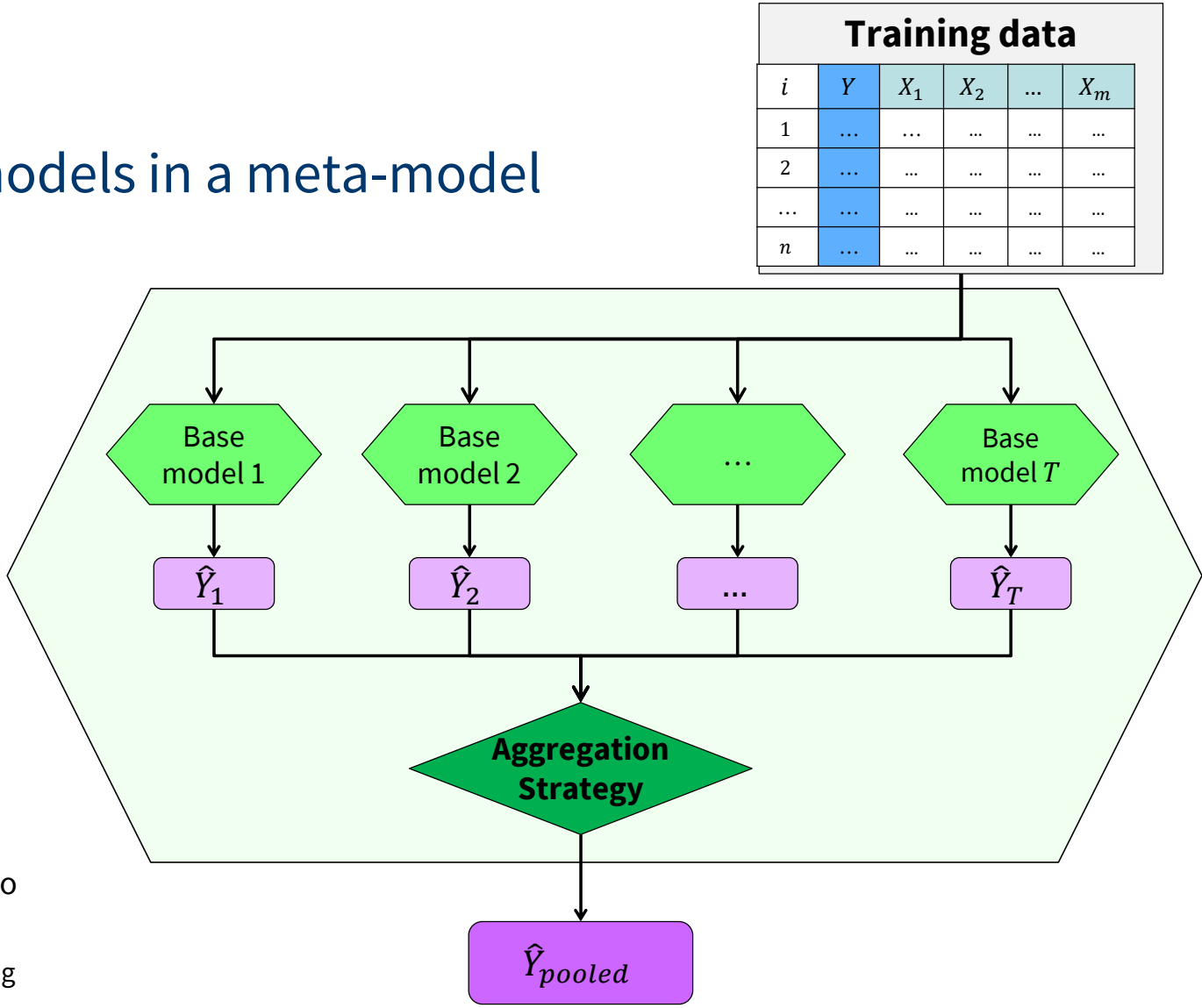
Introduction

Ensemble learning fundamentals, bias-variance trade-off

Ensemble Learning

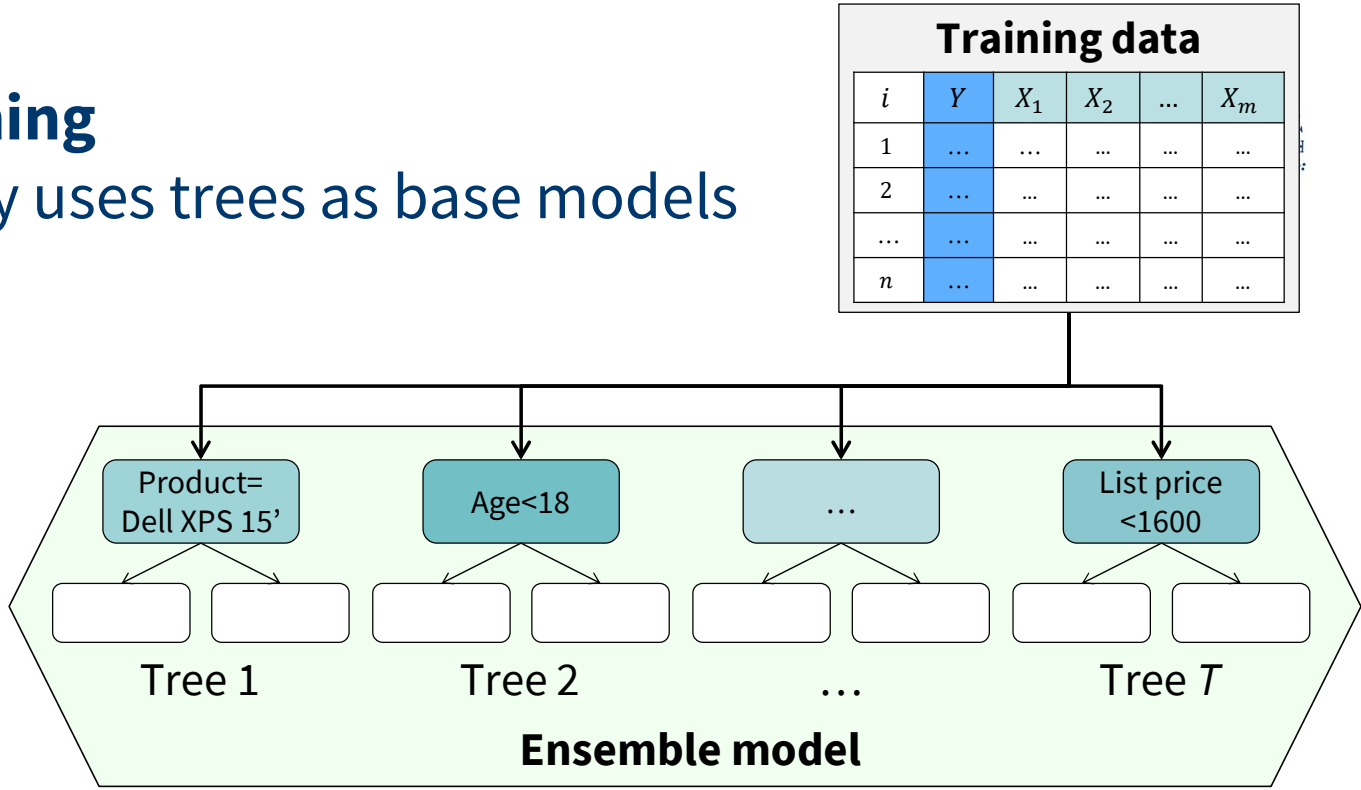
Combining multiple (base) models in a meta-model

- **Multi-step modeling approach**
  - Develop a set of (base) models
  - Aggregate their predictions
- **Goal is to raise predictive accuracy**
  - Models learn different patterns from data and can complement each other
  - Many simple base models can give a powerful ensemble model
- **Different ensemble learning strategies**
  - How to derive base models
    - Homogeneous ensembles
    - Heterogeneous ensembles
  - How to integrate base model predictions into composite forecast
    - Simple or weighted averages with or w/o pruning
    - Let all base models have a vote in the forecast or call base models selectively



Tree-Based Ensemble Learning

Ensemble learning commonly uses trees as base models

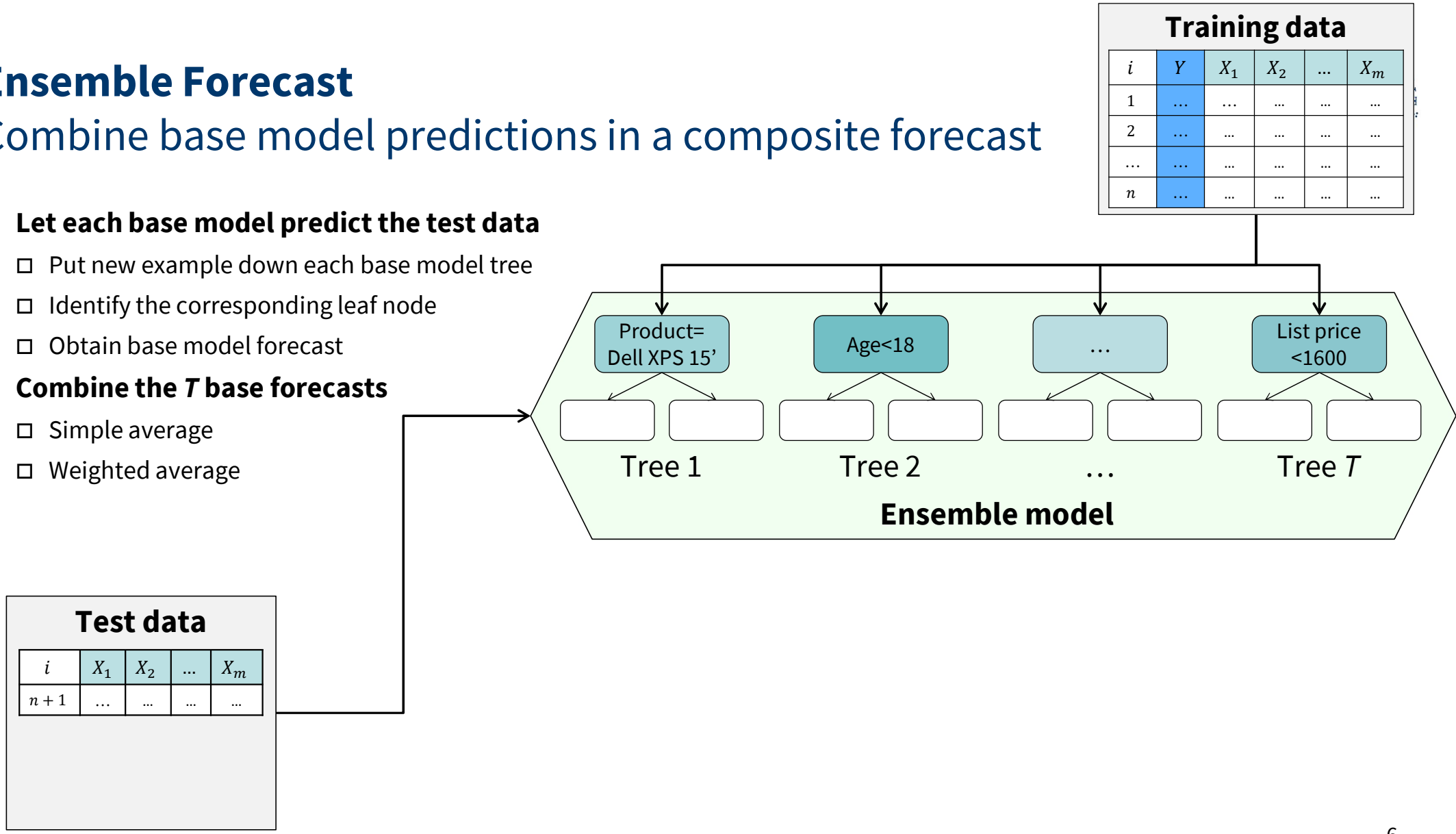


- **Tree learning is unstable**
  - Small changes in the data (e.g., due to sampling) can lead to different trees
  - Easy to derive many different models from a data set
- **Here we consider a set of *T* decision stumps**
  - Stump = tree with only one split
  - No loss of generality
  - Following considerations equally apply to proper tree models

Ensemble Forecast

Combine base model predictions in a composite forecast

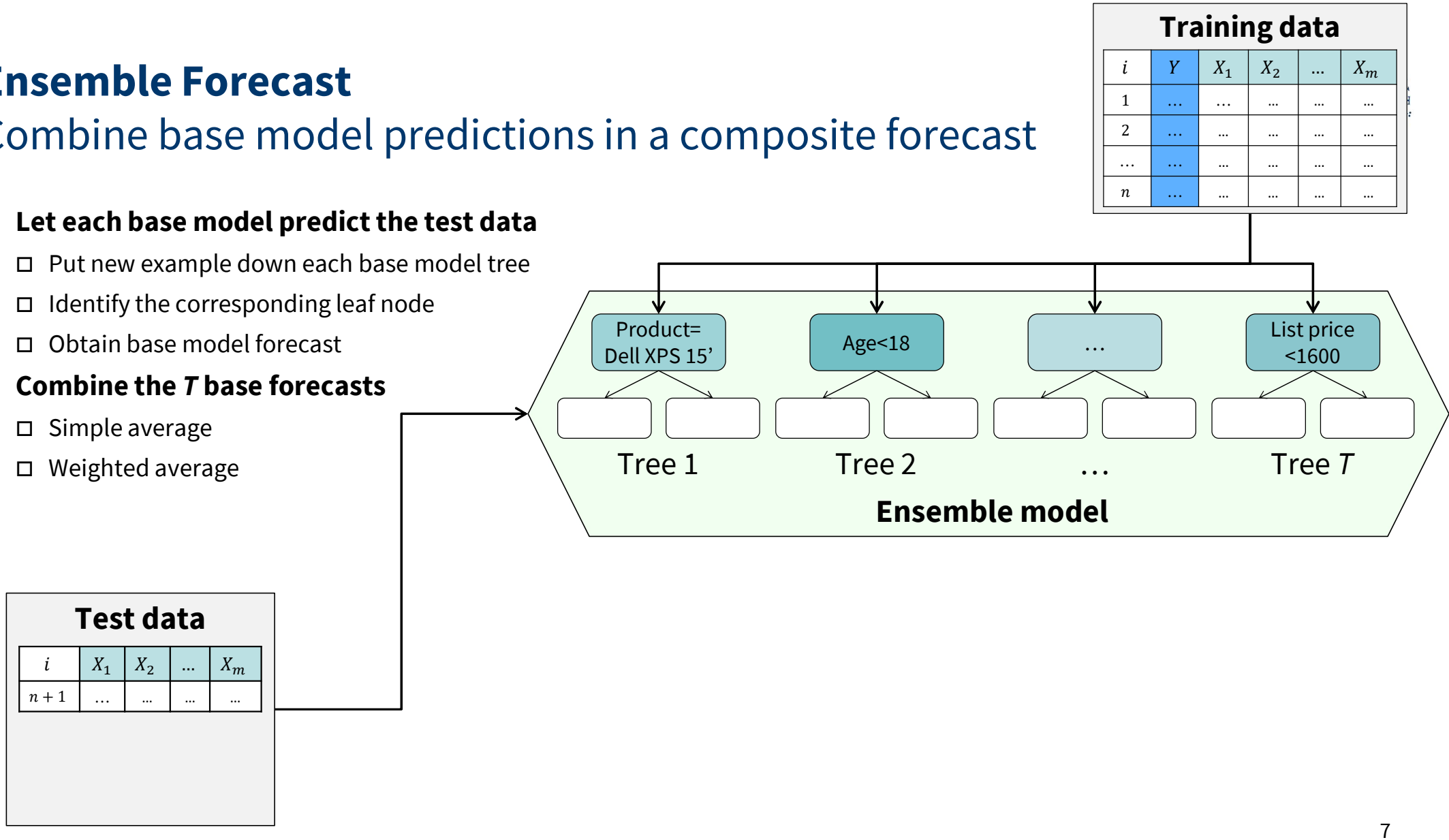
- **Let each base model predict the test data**
  - Put new example down each base model tree
  - Identify the corresponding leaf node
  - Obtain base model forecast
- **Combine the *T* base forecasts**
  - Simple average
  - Weighted average



Ensemble Forecast

Combine base model predictions in a composite forecast

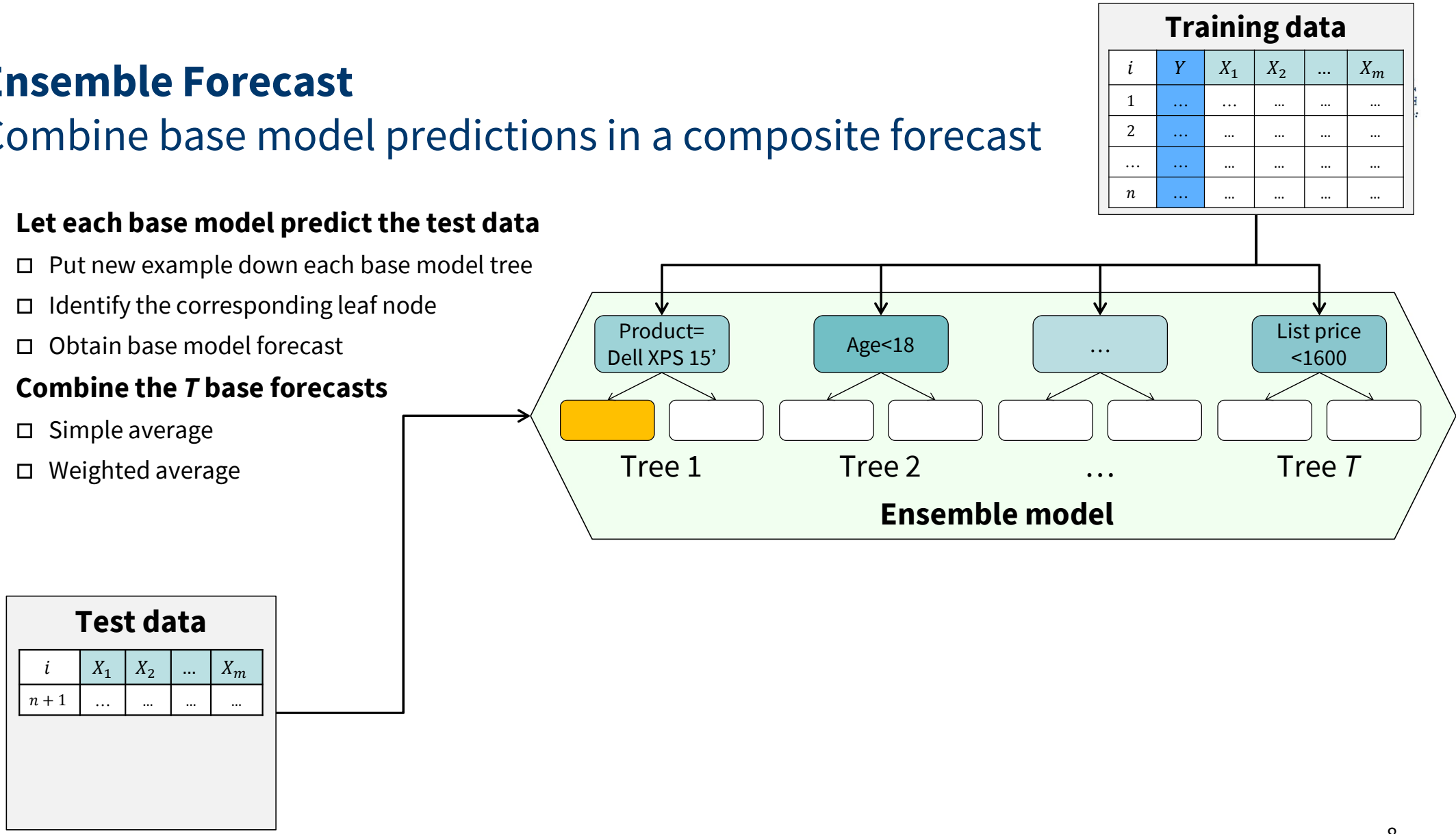
- **Let each base model predict the test data**
  - Put new example down each base model tree
  - Identify the corresponding leaf node
  - Obtain base model forecast
- **Combine the *T* base forecasts**
  - Simple average
  - Weighted average



Ensemble Forecast

Combine base model predictions in a composite forecast

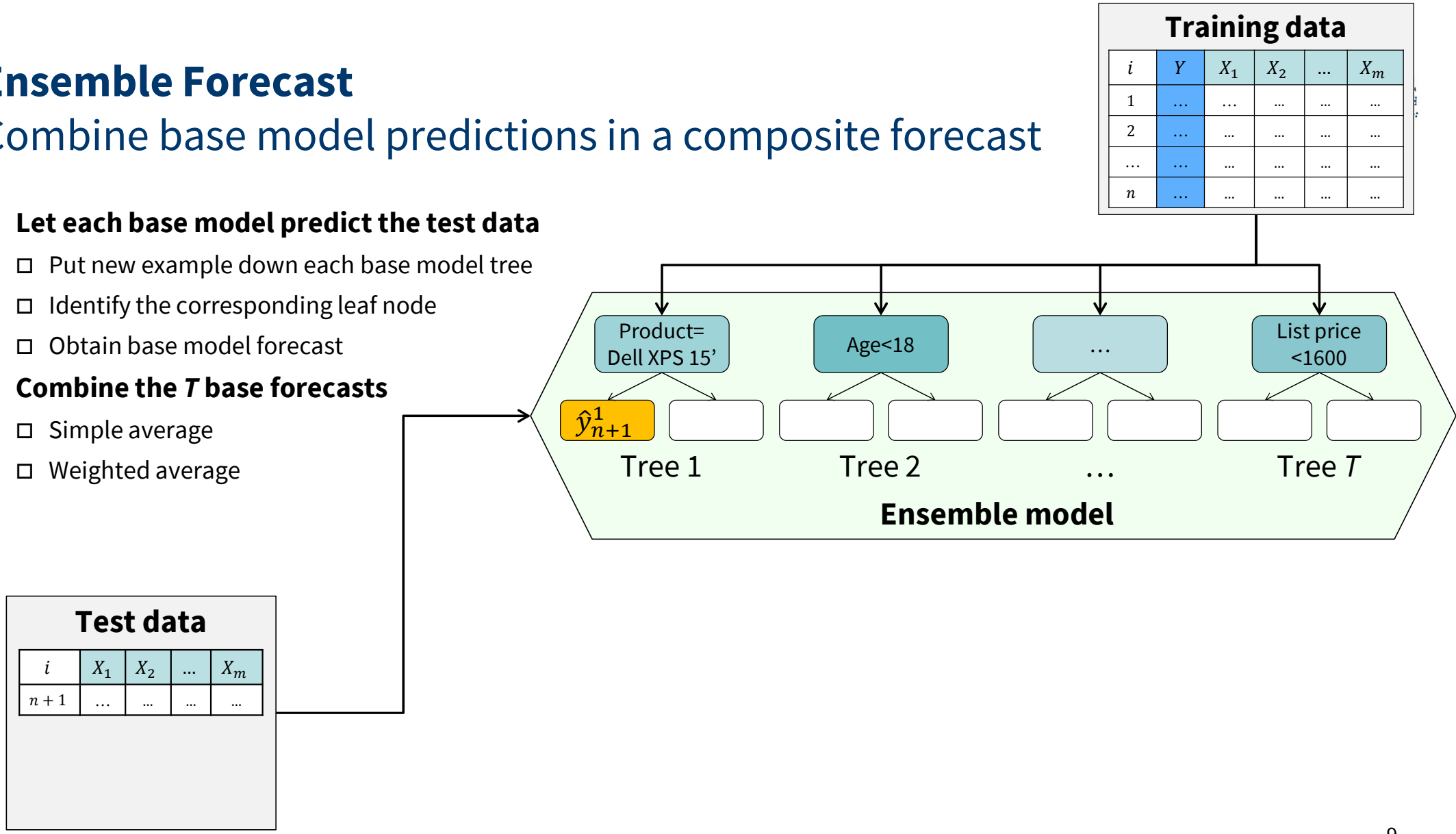
- **Let each base model predict the test data**
  - Put new example down each base model tree
  - Identify the corresponding leaf node
  - Obtain base model forecast
- **Combine the *T* base forecasts**
  - Simple average
  - Weighted average



Ensemble Forecast

Combine base model predictions in a composite forecast

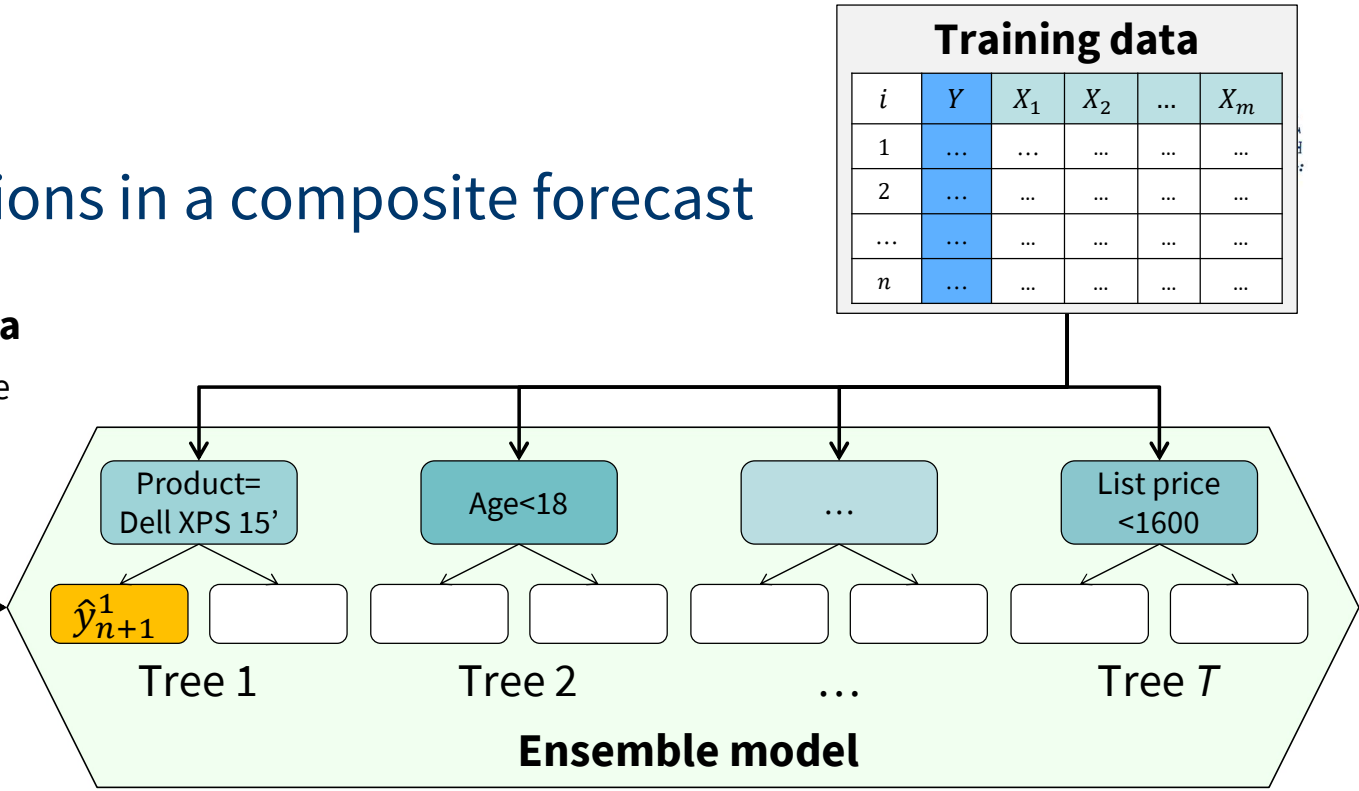
- **Let each base model predict the test data**
  - Put new example down each base model tree
  - Identify the corresponding leaf node
  - Obtain base model forecast
- **Combine the *T* base forecasts**
  - Simple average
  - Weighted average

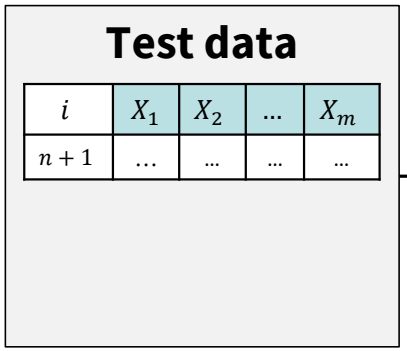


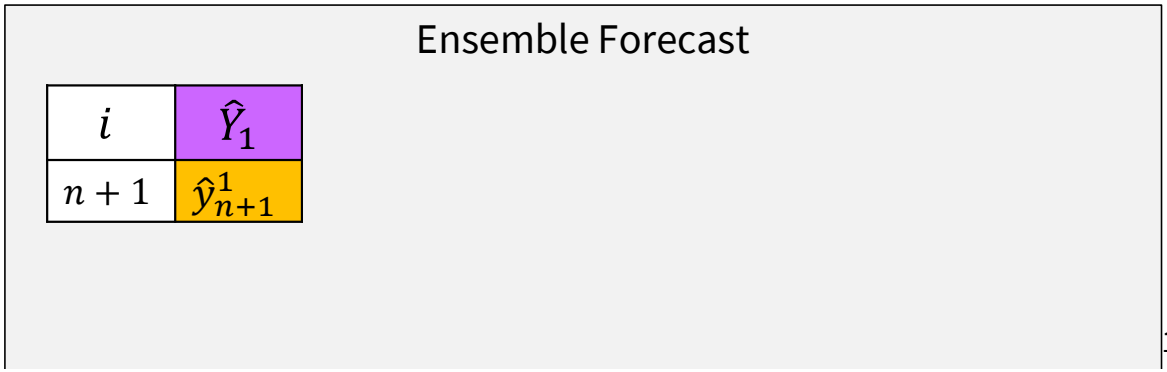
Ensemble Forecast

Combine base model predictions in a composite forecast

- **Let each base model predict the test data**
  - Put new example down each base model tree
  - Identify the corresponding leaf node
  - Obtain base model forecast
- **Combine the *T* base forecasts**
  - Simple average
  - Weighted average



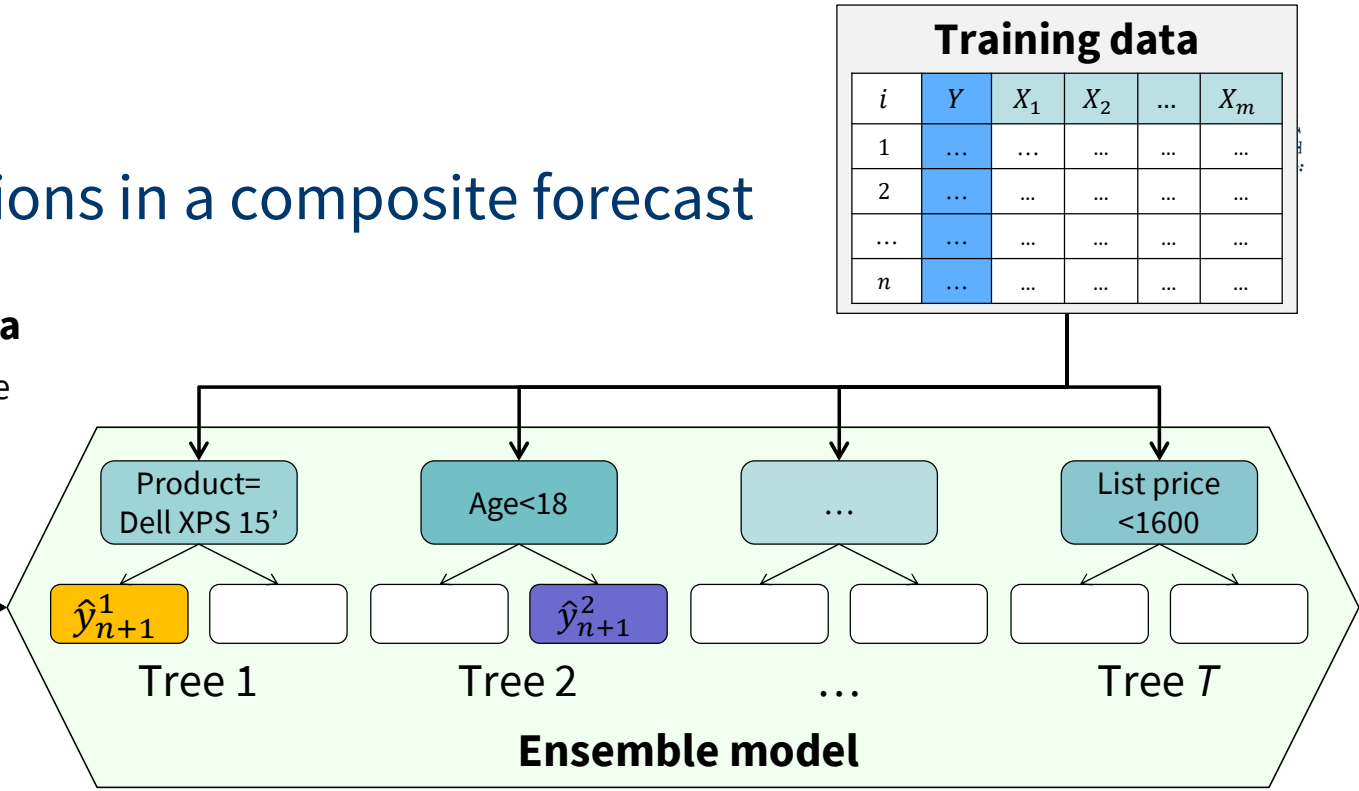




Ensemble Forecast

Combine base model predictions in a composite forecast

- **Let each base model predict the test data**
  - Put new example down each base model tree
  - Identify the corresponding leaf node
  - Obtain base model forecast
- **Combine the *T* base forecasts**
  - Simple average
  - Weighted average



Test data

i	X_1	X_2	...	X_m
$n+1$	...	...	...	...

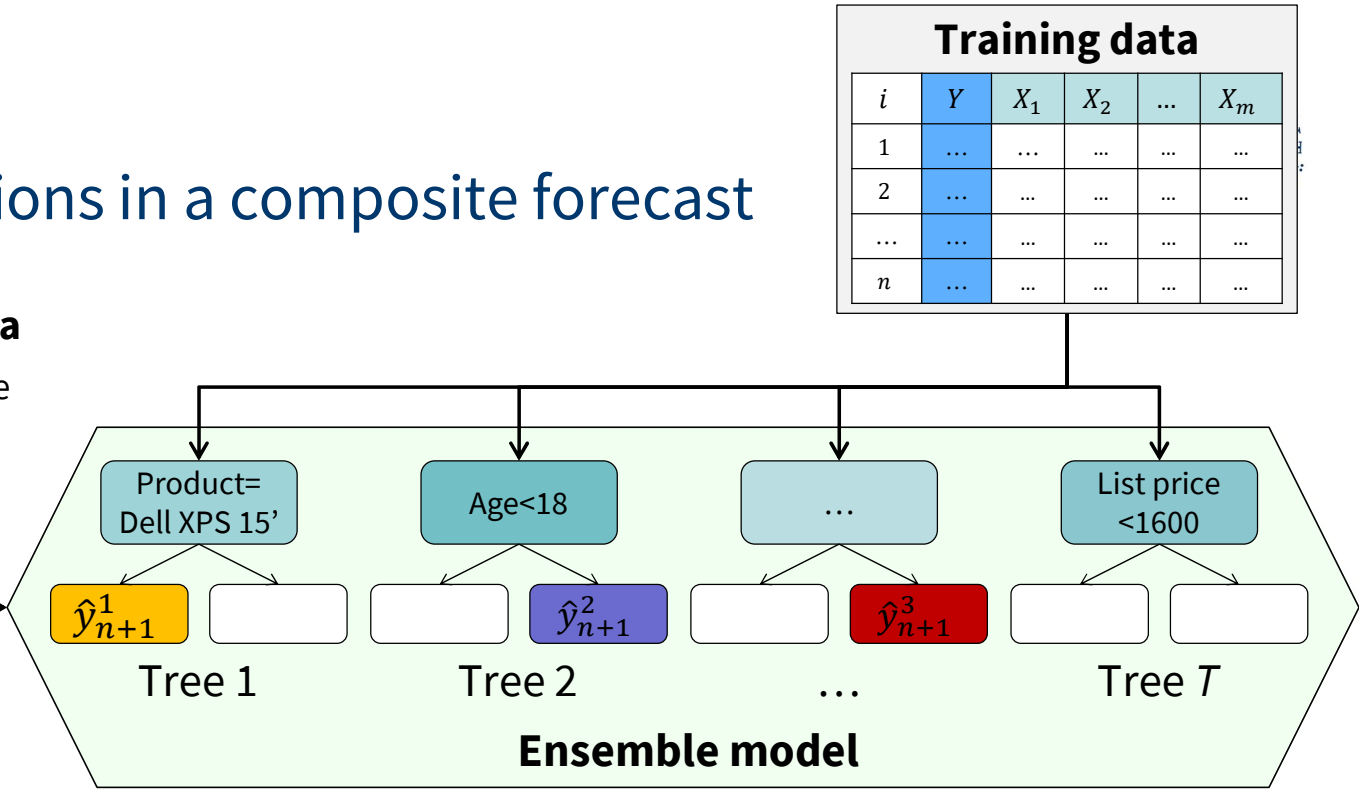
Ensemble Forecast

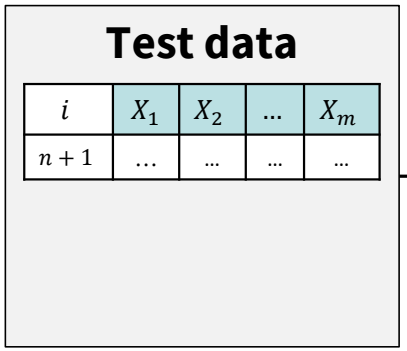
i	$\hat{Y}_1$	$\hat{Y}_2$
$n+1$	$\hat{y}^1_{n+1}$	$\hat{y}^2_{n+1}$

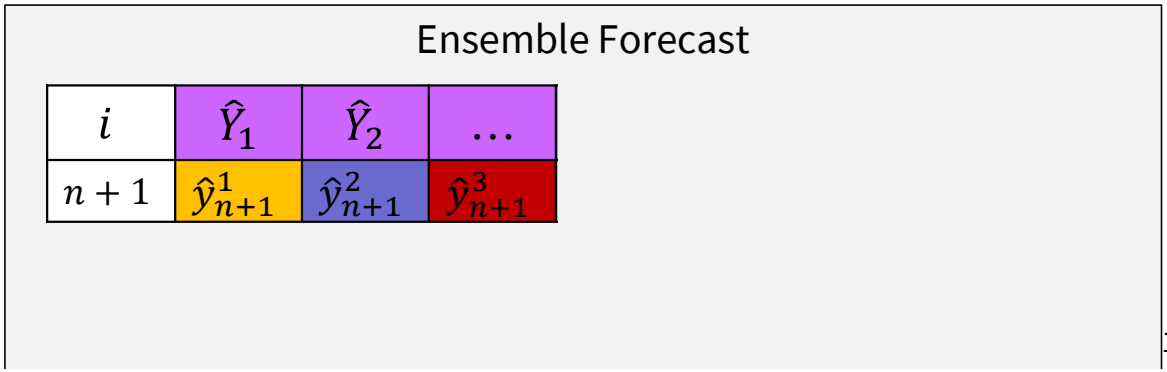
Ensemble Forecast

Combine base model predictions in a composite forecast

- **Let each base model predict the test data**
  - Put new example down each base model tree
  - Identify the corresponding leaf node
  - Obtain base model forecast
- **Combine the *T* base forecasts**
  - Simple average
  - Weighted average



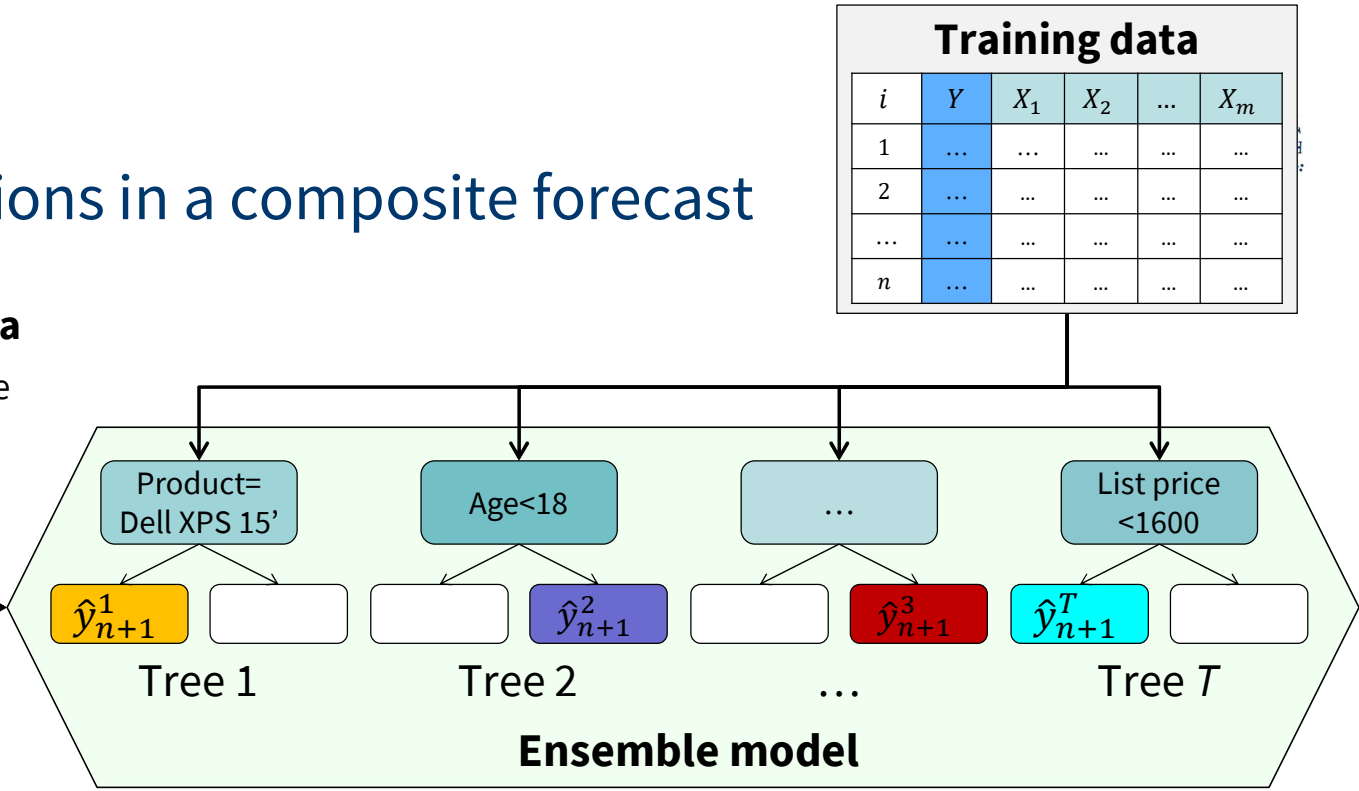




Ensemble Forecast

Combine base model predictions in a composite forecast

- **Let each base model predict the test data**
  - Put new example down each base model tree
  - Identify the corresponding leaf node
  - Obtain base model forecast
- **Combine the *T* base forecasts**
  - Simple average
  - Weighted average



Test data

i	X_1	X_2	...	X_m
$n+1$	...	...	...	...

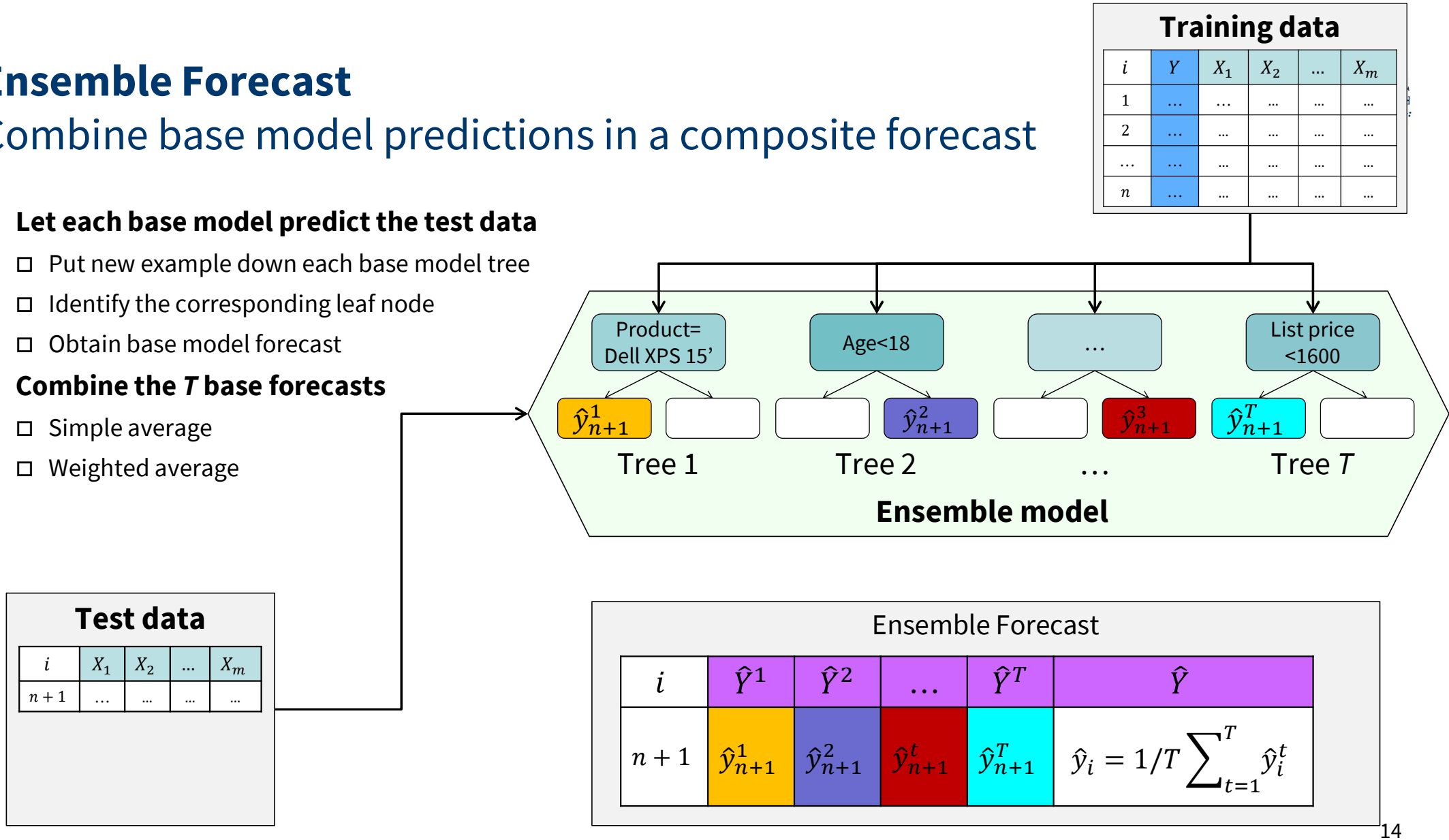
Ensemble Forecast

i	$\hat{Y}_1$	$\hat{Y}_2$	...	$\hat{Y}_T$
$n+1$	$\hat{y}^1_{n+1}$	$\hat{y}^2_{n+1}$	$\hat{y}^3_{n+1}$	$\hat{y}^T_{n+1}$

Ensemble Forecast

Combine base model predictions in a composite forecast

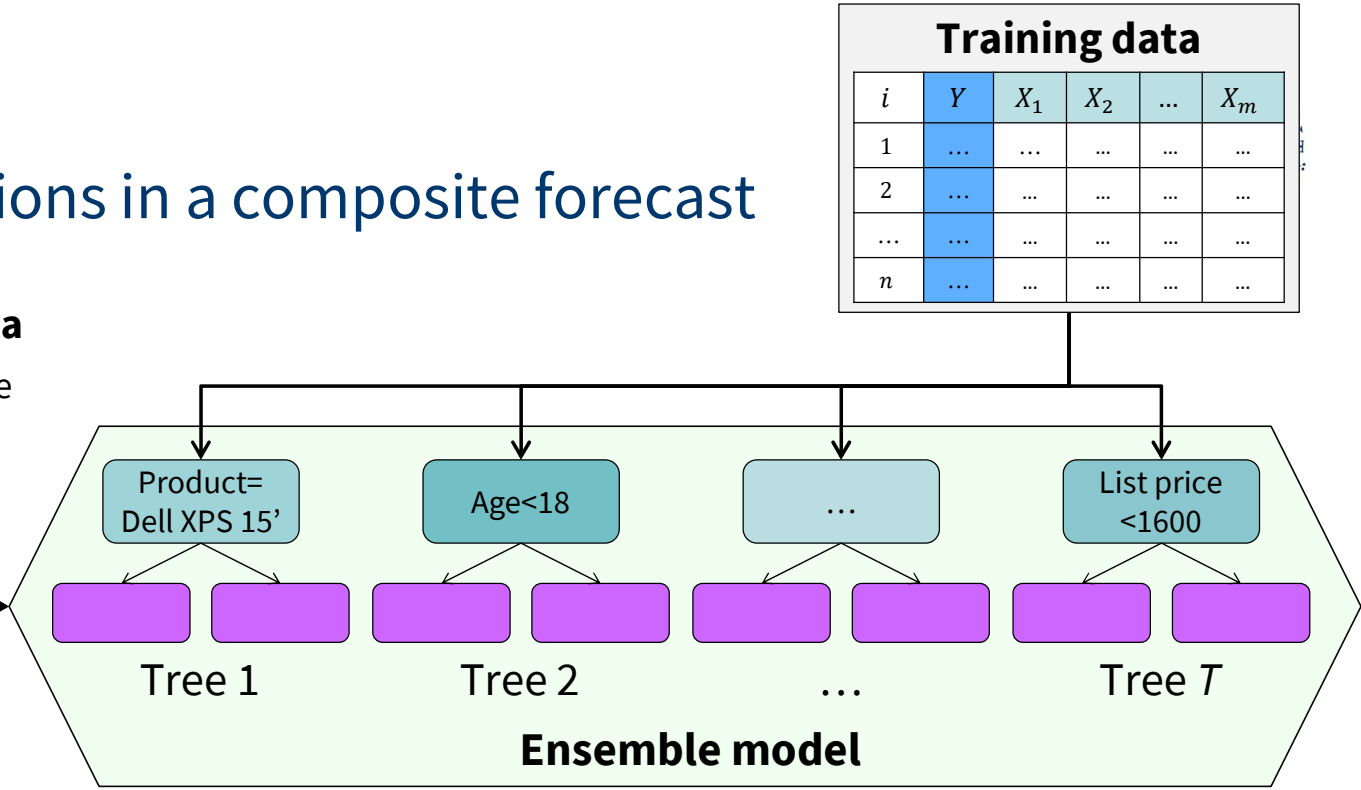
- **Let each base model predict the test data**
  - Put new example down each base model tree
  - Identify the corresponding leaf node
  - Obtain base model forecast
- **Combine the *T* base forecasts**
  - Simple average
  - Weighted average



Ensemble Forecast

Combine base model predictions in a composite forecast

- **Let each base model predict the test data**
  - Put new example down each base model tree
  - Identify the corresponding leaf node
  - Obtain base model forecast
- **Combine the *T* base forecasts**
  - Simple average
  - Weighted average



Test data

i	X_1	X_2	...	X_m
$n+1$	...	...	...	...
$n+2$	...	...	...	...

Ensemble Forecast

i	$\hat{Y}^1$	$\hat{Y}^2$	...	$\hat{Y}^T$	$\hat{Y}$
$n+1$	$\hat{y}^1_{n+1}$	$\hat{y}^2_{n+1}$	$\hat{y}^t_{n+1}$	$\hat{y}^T_{n+1}$	$\hat{y}_i = 1/T \sum_{t=1}^{T} \hat{y}_i^t$
$n+2$	$\hat{y}^1_{n+2}$	$\hat{y}^2_{n+2}$	$\hat{y}^t_{n+2}$	$\hat{y}^T_{n+2}$	

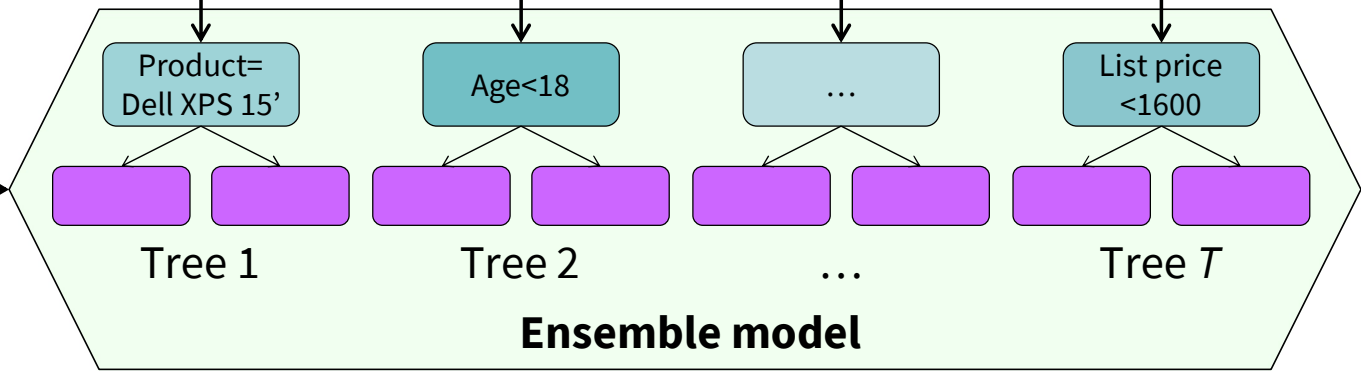
Ensemble Forecast

Combine base model predictions in a composite forecast

- **Let each base model predict the test data**
  - Put new example down each base model tree
  - Identify the corresponding leaf node
  - Obtain base model forecast
- **Combine the *T* base forecasts**
  - Simple average
  - Weighted average

Training data

i	Y	X_1	X_2	...	X_m
1	…	…	…	…	…
2	…	…	…	…	…
…	…	…	…	…	…
n	…	…	…	…	…



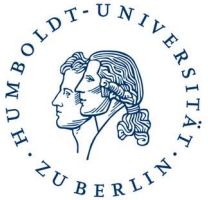
Test data

i	X_1	X_2	...	X_m
$n+1$	…	…	…	…
$n+2$	…	…	…	…
…	…	…	…	…
N	…	…	…	…

Ensemble Forecast

i	$\hat{Y}_1$	$\hat{Y}_2$	…	$\hat{Y}_T$	$\hat{Y}$
$n+1$	$\hat{y}_{n+1}^1$	$\hat{y}_{n+1}^2$	$\hat{y}_{n+1}^t$	$\hat{y}_{n+1}^T$	$\hat{y}_i = 1/T \sum_{t=1}^{T} \hat{y}_i^t$
$n+2$	$\hat{y}_{n+2}^1$	$\hat{y}_{n+2}^2$	$\hat{y}_{n+2}^t$	$\hat{y}_{n+2}^T$	
…	…	…	…	…	
N	$\hat{y}_N^1$	$\hat{y}_N^2$	$\hat{y}_N^t$	$\hat{y}_N^T$	

Why Ensemble Learning Raises Predictive Accuracy



- **A vast amount of empirical evidence shows that model combination improves predictive performance**
- **What factors explain the success of ensemble learning?**

Bias-Variance Trade-Off and Ensemble Learning

- **Model combination and variance**
  - Averaging (forecasts) reduces variance
  - Random forest philosophy
    - Deep trees with low bias as base learner
    - Each tree would suffer high variance
    - Grow large forest to reduce variance
- **Model combination and bias**
  - Combining many weak models can give one strong ensemble model
  - Boosting philosophy
    - Use shallow trees with high bias
    - Add trees to incrementally reduce bias
    - Many shallow trees together form a powerful ensemble model with low bias

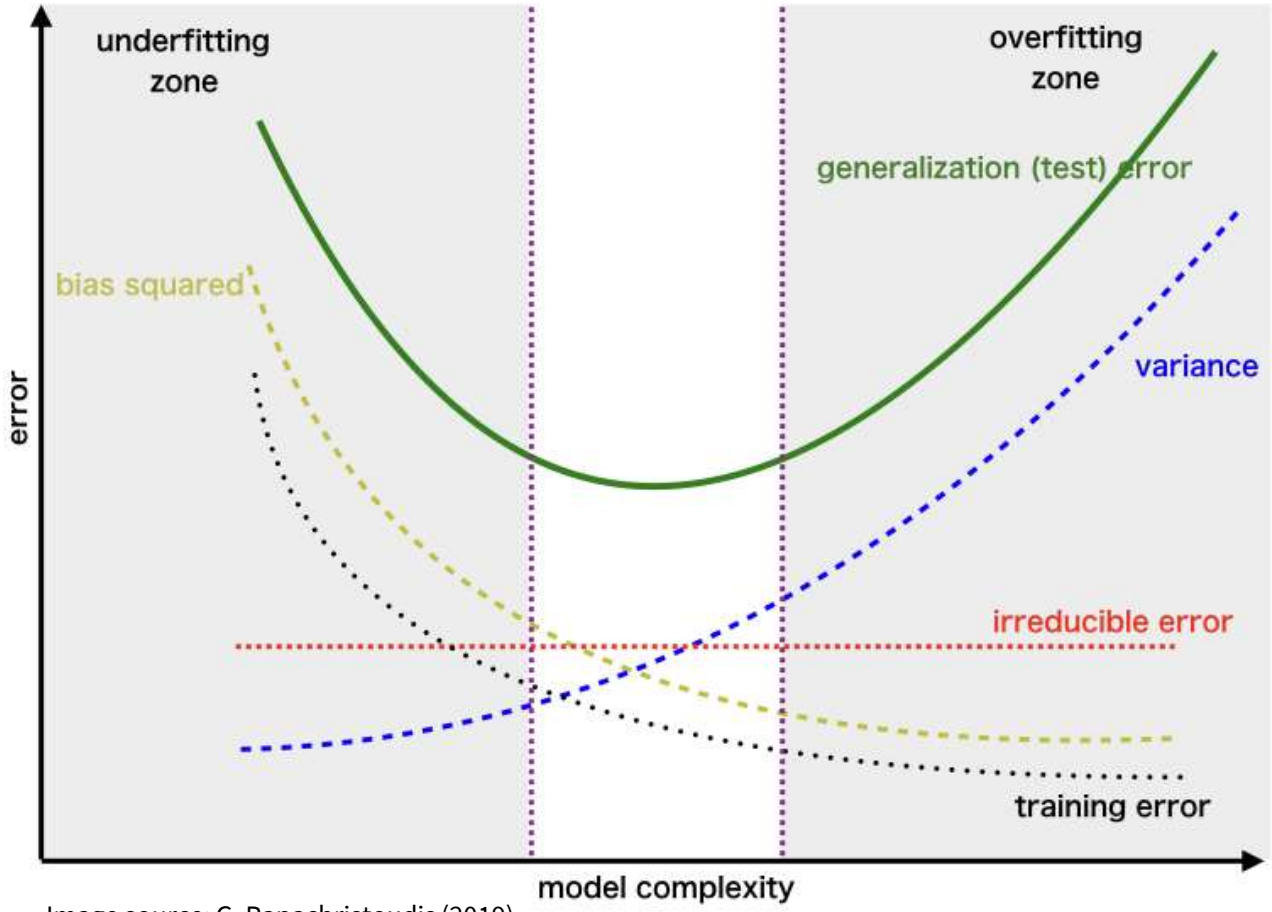
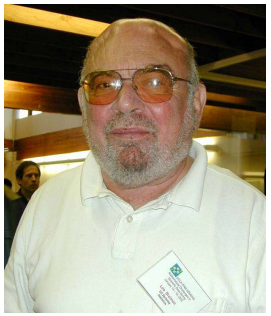


Image source: G. Papachristoudis (2019)
https://towardsdatascience.com/the-bias-variance-tradeoff-8818f41e39e9

Implementing Ensemble Learning

Two main strategies to build (tree-based) ensemble models

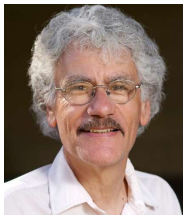
- **Bagging-based approaches**
  - Relies on sampling
    - Draw a random sample from the training data
    - Grow tree an that sample, and repeat
  - No dependence among base models facilitates parallel training
  - Simple average for base model forecast combination
  - Most popular and powerful representative: Random Forest
- **Boosting approaches**
  - Relies on data weighting and forecast errors
    - Grow one tree from the training data and compute residuals
    - Grow tree next tree to reduce those residuals, and repeat
  - Dependence among base models requires sequential training
  - Weighted average for base model forecast combination
  - Most popular and powerful representative: Gradient Boosting



Leo Breiman

R. Shapire

Y. Freund

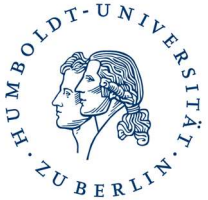


J. Friedman

Bagging and Random Forest

Functioning, meta-parameters, and a little bit of theory

Bootstrap Aggregation (Bagging)



- **Introduced in Breiman (1996)**
- **Homogeneous ensemble strategy**
  - Best to use unstable base learners that are sensitive to manipulation of the training data
  - Examples: decision trees, artificial neural networks, prototype methods
- **Uses bootstrap sampling (see next)**
- **Later extended into Random Forest algorithm (Breiman 2001)**
  - Adds random subspace mechanism (Ho, 1998)
  - Unlike bagging, this extension is specific to tree-based base models
- **Theoretical and much empirical evidence suggest Random Forest typically outperforms bagging *if* using trees as base learners**
- **So ok to focus on random forest in the reminder**

Random Forest Algorithm

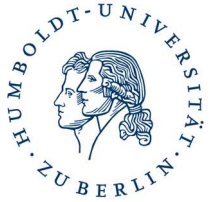
Grow individual trees from bootstrap samples of the training data

- **A bootstrap sample is a random sample of the same size as the original data drawn randomly and with replacement**
  - Some observations occur multiple times
  - Other observations (~30%) do not appear

i	Product	List price	Age	Industry	…	Resale price [$]
1	Dell XPS 15’	2,500	36	Mining	…	347
2	Dell XPS 17’	3,000	36	Health	…	538
3	HP Envy 17’	1,300	24	Office	…	121
4	HP EliteBook 850	1,900	36	Mining	…	172
5	Lenovo Yoga 13’	1,100	12	Office	…	266

Random Forest Algorithm

Grow individual trees from bootstrap samples of the training data



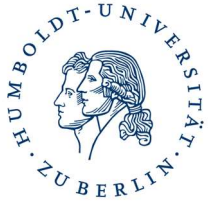
- **A bootstrap sample is a random sample of the same size as the original data drawn randomly and with replacement**
  - Some observations occur multiple times
  - Other observations (~30%) do not appear

i	Product	List price	Age	Industry	…	Resale price [$]

i	Product	List price	Age	Industry	…	Resale price [$]
1	Dell XPS 15’	2,500	36	Mining	…	347
2	Dell XPS 17’	3,000	36	Health	…	538
3	HP Envy 17’	1,300	24	Office	…	121
4	HP EliteBook 850	1,900	36	Mining	…	172
5	Lenovo Yoga 13’	1,100	12	Office	…	266

Random Forest Algorithm

Grow individual trees from bootstrap samples of the training data



- **A bootstrap sample is a random sample of the same size as the original data drawn randomly and with replacement**
  - Some observations occur multiple times
  - Other observations (~30%) do not appear

i	Product	List price	Age	Industry	…	Resale price [$]

i	Product	List price	Age	Industry	…	Resale price [$]
1	Dell XPS 15’	2,500	36	Mining	…	347
2	Dell XPS 17’	3,000	36	Health	…	538
3	HP Envy 17’	1,300	24	Office	…	121
4	HP EliteBook 850	1,900	36	Mining	…	172
5	Lenovo Yoga 13’	1,100	12	Office	…	266

Random Forest Algorithm

Grow individual trees from bootstrap samples of the training data

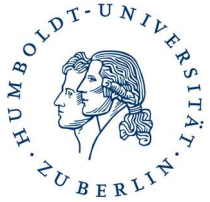
- **A bootstrap sample is a random sample of the same size as the original data drawn randomly and with replacement**
  - Some observations occur multiple times
  - Other observations (~30%) do not appear

i	Product	List price	Age	Industry	…	Resale price [$]
1	Dell XPS 15’	2,500	36	Mining	…	347
2	Dell XPS 17’	3,000	36	Health	…	538
3	HP Envy 17’	1,300	24	Office	…	121
4	HP EliteBook 850	1,900	36	Mining	…	172
5	Lenovo Yoga 13’	1,100	12	Office	…	266

i	Product	List price	Age	Industry	…	Resale price [$]
2	Dell XPS 17’	3,000	36	Health	…	538

Random Forest Algorithm

Grow individual trees from bootstrap samples of the training data



- **A bootstrap sample is a random sample of the same size as the original data drawn randomly and with replacement**
  - Some observations occur multiple times
  - Other observations (~30%) do not appear

i	Product	List price	Age	Industry	…	Resale price [$]
2	Dell XPS 17’	3,000	36	Health	…	538

i	Product	List price	Age	Industry	…	Resale price [$]
1	Dell XPS 15’	2,500	36	Mining	…	347
2	Dell XPS 17’	3,000	36	Health	…	538
3	HP Envy 17’	1,300	24	Office	…	121
4	HP EliteBook 850	1,900	36	Mining	…	172
5	Lenovo Yoga 13’	1,100	12	Office	…	266

Random Forest Algorithm

Grow individual trees from bootstrap samples of the training data

- **A bootstrap sample is a random sample of the same size as the original data drawn randomly and with replacement**
  - Some observations occur multiple times
  - Other observations (~30%) do not appear

i	Product	List price	Age	Industry	…	Resale price [$]
1	Dell XPS 15’	2,500	36	Mining	…	347
2	Dell XPS 17’	3,000	36	Health	…	538
3	HP Envy 17’	1,300	24	Office	…	121
4	HP EliteBook 850	1,900	36	Mining	…	172
5	Lenovo Yoga 13’	1,100	12	Office	…	266

i	Product	List price	Age	Industry	…	Resale price [$]
2	Dell XPS 17’	3,000	36	Health	…	538
5	Lenovo Yoga 13’	1,100	12	Office	…	266

Random Forest Algorithm

Grow individual trees from bootstrap samples of the training data

- **A bootstrap sample is a random sample of the same size as the original data drawn randomly and with replacement**
  - Some observations occur multiple times
  - Other observations (~30%) do not appear

i	Product	List price	Age	Industry	…	Resale price [$]
1	Dell XPS 15’	2,500	36	Mining	…	347
2	Dell XPS 17’	3,000	36	Health	…	538
3	HP Envy 17’	1,300	24	Office	…	121
4	HP EliteBook 850	1,900	36	Mining	…	172
5	Lenovo Yoga 13’	1,100	12	Office	…	266

i	Product	List price	Age	Industry	…	Resale price [$]
2	Dell XPS 17’	3,000	36	Health	…	538
5	Lenovo Yoga 13’	1,100	12	Office	…	266

Random Forest Algorithm

Grow individual trees from bootstrap samples of the training data

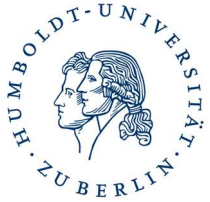
- **A bootstrap sample is a random sample of the same size as the original data drawn randomly and with replacement**
  - Some observations occur multiple times
  - Other observations (~30%) do not appear

i	Product	List price	Age	Industry	…	Resale price [$]
1	Dell XPS 15’	2,500	36	Mining	…	347
2	Dell XPS 17’	3,000	36	Health	…	538
3	HP Envy 17’	1,300	24	Office	…	121
4	HP EliteBook 850	1,900	36	Mining	…	172
5	Lenovo Yoga 13’	1,100	12	Office	…	266

i	Product	List price	Age	Industry	…	Resale price [$]
2	Dell XPS 17’	3,000	36	Health	…	538
5	Lenovo Yoga 13’	1,100	12	Office	…	266
1	Dell XPS 15’	2,500	36	Mining	…	347

Random Forest Algorithm

Grow individual trees from bootstrap samples of the training data



- **A bootstrap sample is a random sample of the same size as the original data drawn randomly and with replacement**
  - Some observations occur multiple times
  - Other observations (~30%) do not appear

i	Product	List price	Age	Industry	…	Resale price [$]
1	Dell XPS 15’	2,500	36	Mining	…	347
2	Dell XPS 17’	3,000	36	Health	…	538
3	HP Envy 17’	1,300	24	Office	…	121
4	HP EliteBook 850	1,900	36	Mining	…	172
5	Lenovo Yoga 13’	1,100	12	Office	…	266

i	Product	List price	Age	Industry	…	Resale price [$]
2	Dell XPS 17’	3,000	36	Health	…	538
5	Lenovo Yoga 13’	1,100	12	Office	…	266
1	Dell XPS 15’	2,500	36	Mining	…	347

Random Forest Algorithm

Grow individual trees from bootstrap samples of the training data

- **A bootstrap sample is a random sample of the same size as the original data drawn randomly and with replacement**
  - Some observations occur multiple times
  - Other observations (~30%) do not appear

i	Product	List price	Age	Industry	…	Resale price [$]
1	Dell XPS 15’	2,500	36	Mining	…	347
2	Dell XPS 17’	3,000	36	Health	…	538
3	HP Envy 17’	1,300	24	Office	…	121
4	HP EliteBook 850	1,900	36	Mining	…	172
5	Lenovo Yoga 13’	1,100	12	Office	…	266

i	Product	List price	Age	Industry	…	Resale price [$]
2	Dell XPS 17’	3,000	36	Health	…	538
5	Lenovo Yoga 13’	1,100	12	Office	…	266
1	Dell XPS 15’	2,500	36	Mining	…	347
5	Lenovo Yoga 13’	1,100	12	Office	…	266

Random Forest Algorithm

Grow individual trees from bootstrap samples of the training data

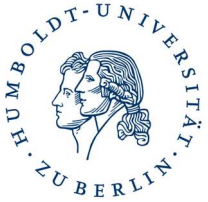
- **A bootstrap sample is a random sample of the same size as the original data drawn randomly and with replacement**
  - Some observations occur multiple times
  - Other observations (~30%) do not appear

i	Product	List price	Age	Industry	…	Resale price [$]
1	Dell XPS 15’	2,500	36	Mining	…	347
2	Dell XPS 17’	3,000	36	Health	…	538
3	HP Envy 17’	1,300	24	Office	…	121
4	HP EliteBook 850	1,900	36	Mining	…	172
5	Lenovo Yoga 13’	1,100	12	Office	…	266

i	Product	List price	Age	Industry	…	Resale price [$]
2	Dell XPS 17’	3,000	36	Health	…	538
5	Lenovo Yoga 13’	1,100	12	Office	…	266
1	Dell XPS 15’	2,500	36	Mining	…	347
5	Lenovo Yoga 13’	1,100	12	Office	…	266

Random Forest Algorithm

Grow individual trees from bootstrap samples of the training data



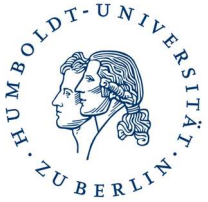
- **A bootstrap sample is a random sample of the same size as the original data drawn randomly and with replacement**
  - Some observations occur multiple times
  - Other observations (~30%) do not appear

i	Product	List price	Age	Industry	…	Resale price [$]
1	Dell XPS 15’	2,500	36	Mining	…	347
2	Dell XPS 17’	3,000	36	Health	…	538
3	HP Envy 17’	1,300	24	Office	…	121
4	HP EliteBook 850	1,900	36	Mining	…	172
5	Lenovo Yoga 13’	1,100	12	Office	…	266

i	Product	List price	Age	Industry	…	Resale price [$]
2	Dell XPS 17’	3,000	36	Health	…	538
5	Lenovo Yoga 13’	1,100	12	Office	…	266
1	Dell XPS 15’	2,500	36	Mining	…	347
5	Lenovo Yoga 13’	1,100	12	Office	…	266
2	Dell XPS 17’	3,000	36	Health	…	538

Random Forest Algorithm

Grow individual trees from bootstrap samples of the training data



- **A bootstrap sample is a random sample of the same size as the original data drawn randomly and with replacement**
  - Some observations occur multiple times
  - Other observations (~30%) do not appear

i	Product	List price	Age	Industry	…	Resale price [$]
2	Dell XPS 17’	3,000	36	Health	…	538
5	Lenovo Yoga 13’	1,100	12	Office	…	266
1	Dell XPS 15’	2,500	36	Mining	…	347
5	Lenovo Yoga 13’	1,100	12	Office	…	266
2	Dell XPS 17’	3,000	36	Health	…	538

i	Product	List price	Age	Industry	…	Resale price [$]
1	Dell XPS 15’	2,500	36	Mining	…	347
2	Dell XPS 17’	3,000	36	Health	…	538
3	HP Envy 17’	1,300	24	Office	…	121
4	HP EliteBook 850	1,900	36	Mining	…	172
5	Lenovo Yoga 13’	1,100	12	Office	…	266

Note how the out-of-bag (OOB) observations can serve as *hold-out test* data to evaluate a model derived from the bootstrap sample.

Random Forest Algorithm

Grow individual trees from bootstrap samples of the training data

- **A bootstrap sample is a random sample of the same size as the original data drawn randomly and with replacement**
  - Some observations occur multiple times
  - Other observations (~30%) do not appear

i	Product	List price	Age	Industry	…	Resale price [$]
1	Dell XPS 15’	2,500	36	Mining	…	347
2	Dell XPS 17’	3,000	36	Health	…	538
3	HP Envy 17’	1,300	24	Office	…	121
4	HP EliteBook 850	1,900	36	Mining	…	172
5	Lenovo Yoga 13’	1,100	12	Office	…	266

- **We can create as many somewhat different samples from the training data as we like**

i	Product	List price	Age	Industry	…	Resale price [$]
2	Dell XPS 17’	3,000	36	Health	…	538
5	Lenovo Yoga 13’	1,100	12	Office	…	266
1	Dell XPS 15’	2,500	36	Mining	…	347
5	Lenovo Yoga 13’	1,100	12	Office	…	266
2	Dell XPS 17’	3,000	36	Health	…	538

i	Product	List price	Age	Industry	…	Resale price [$]
4	HP EliteBook 850	1,900	36	Mining	…	172
3	HP Envy 17’	1,300	24	Office	…	121
4	HP EliteBook 850	1,900	36	Mining	…	172
1	Dell XPS 15’	2,500	36	Mining	…	347
1	Dell XPS 15’	2,500	36	Mining	…	347

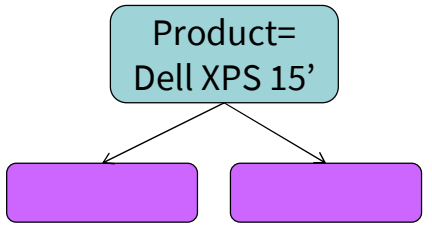
i	Product	List price	Age	Industry	…	Resale price [$]
5	Lenovo Yoga 13’	1,100	12	Office	…	266
3	HP Envy 17’	1,300	24	Office	…	121
4	HP EliteBook 850	1,900	36	Mining	…	172
1	Dell XPS 15’	2,500	36	Mining	…	347
3	HP Envy 17’	1,300	24	Office	…	121

Random Forest Algorithm

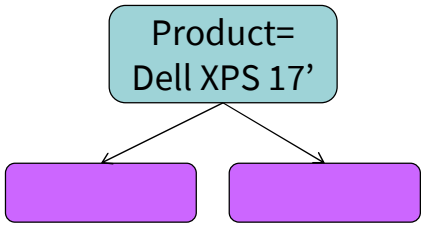
Grow each tree using the random subspace mechanism

- **Random subspace modifies how we grow a tree**
- **Key decision in tree growing is finding splits**
  - Standard/previous approach
    - Step 1: For each feature, find the best split on that feature
    - Step 2: Select the best split over all features

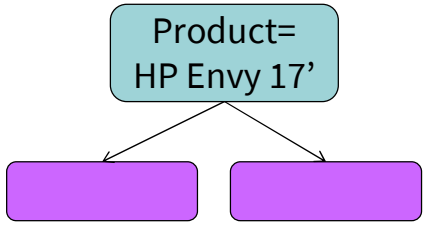
i	Product	List price	Age	Industry	…	Resale price [$]
1	Dell XPS 15’	2,500	36	Mining	…	347
2	Dell XPS 17’	3,000	36	Health	…	538
3	HP Envy 17’	1,300	24	Office	…	121
4	HP EliteBook 850	1,900	36	Mining	…	172
5	Lenovo Yoga 13’	1,100	12	Office	…	266



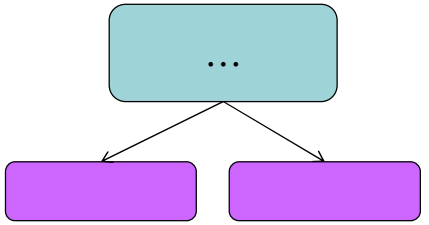
SSR=103,573



SSR=112,693



SSR=116,342

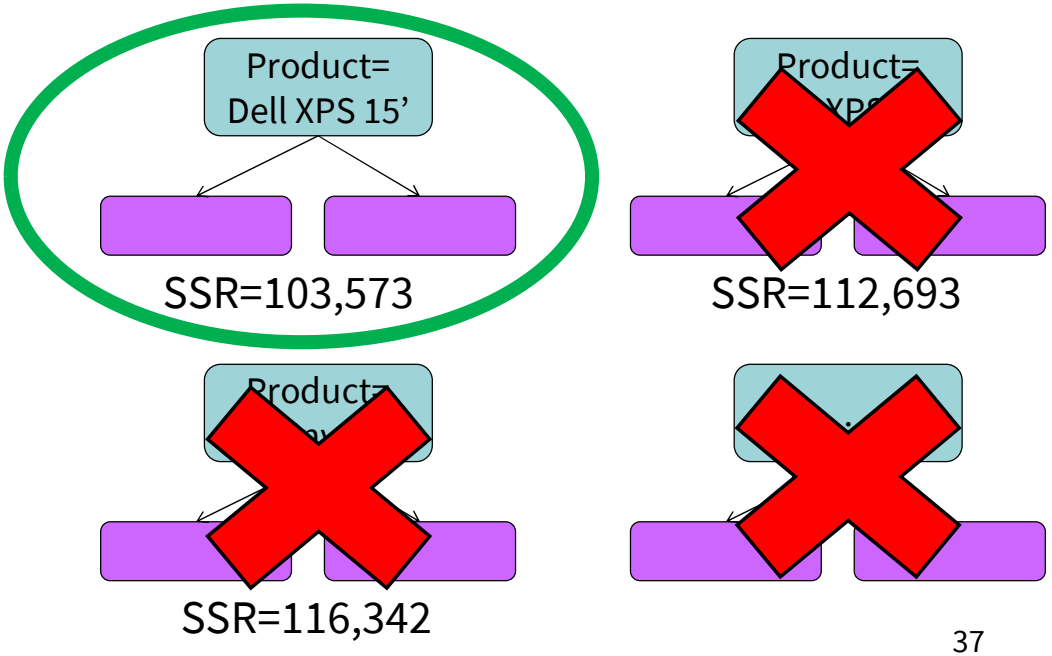


Random Forest Algorithm

Grow each tree using the random subspace mechanism

- **Random subspace modifies how we grow a tree**
- **Key decision in tree growing is finding splits**
  - Standard/previous approach
    - Step 1: For each feature, find the best split on that feature
    - Step 2: Select the best split over all features

i	Product	List price	Age	Industry	…	Resale price [$]
1	Dell XPS 15’	2,500	36	Mining	…	347
2	Dell XPS 17’	3,000	36	Health	…	538
3	HP Envy 17’	1,300	24	Office	…	121
4	HP EliteBook 850	1,900	36	Mining	…	172
5	Lenovo Yoga 13’	1,100	12	Office	…	266

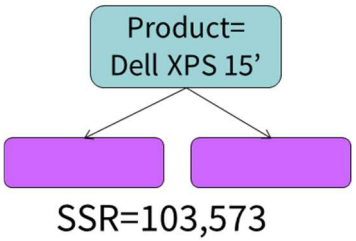


Random Forest Algorithm

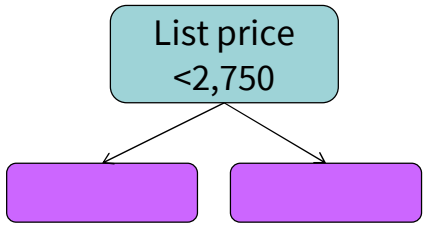
Grow each tree using the random subspace mechanism

- **Random subspace modifies how we grow a tree**
- **Key decision in tree growing is finding splits**
  - Standard/previous approach
    - Step 1: For each feature, find the best split on that feature
    - Step 2: Select the best split over all features

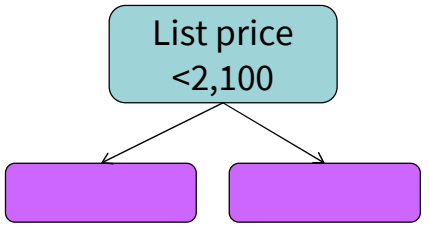
i	Product	List price	Age	Industry	…	Resale price [$]
1	Dell XPS 15’	2,500	36	Mining	…	347
2	Dell XPS 17’	3,000	36	Health	…	538
3	HP Envy 17’	1,300	24	Office	…	121
4	HP EliteBook 850	1,900	36	Mining	…	172
5	Lenovo Yoga 13’	1,100	12	Office	…	266



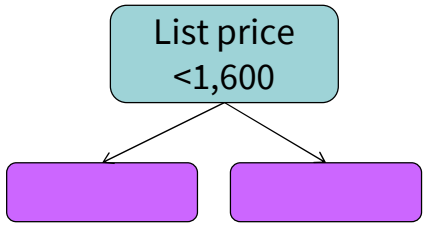
SSR=103,573



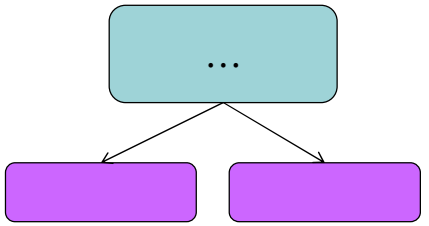
SSR= 112,693



SSR=29,061



SSR=77,533

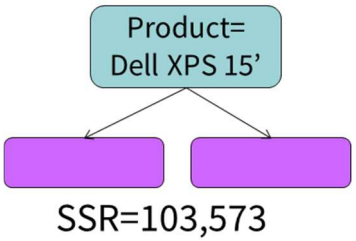


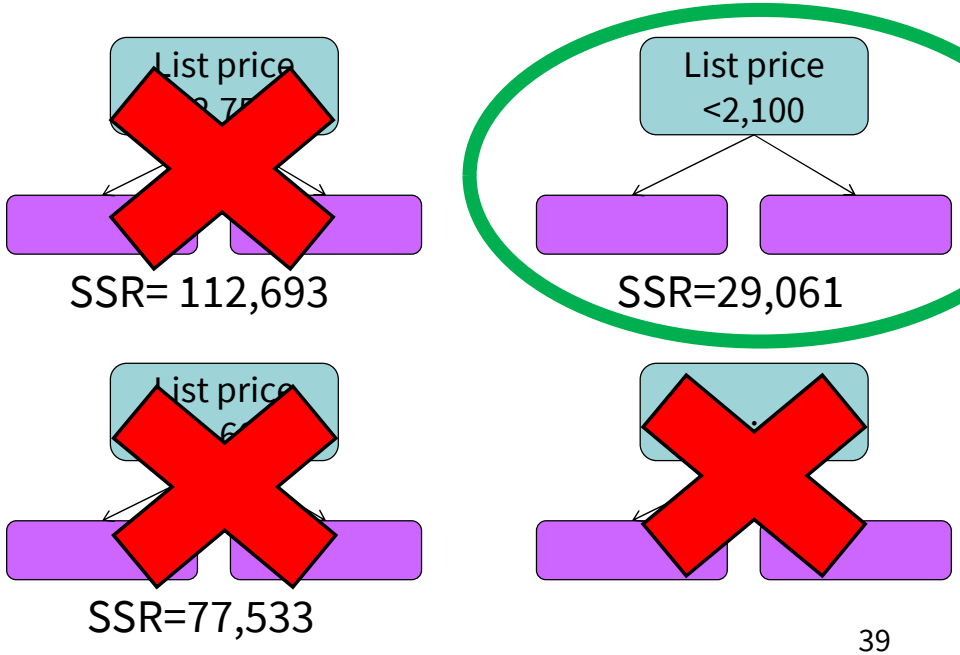
Random Forest Algorithm

Grow each tree using the random subspace mechanism

- **Random subspace modifies how we grow a tree**
- **Key decision in tree growing is finding splits**
  - Standard/previous approach
    - Step 1: For each feature, find the best split on that feature
    - Step 2: Select the best split over all features

i	Product	List price	Age	Industry	…	Resale price [$]
1	Dell XPS 15’	2,500	36	Mining	…	347
2	Dell XPS 17’	3,000	36	Health	…	538
3	HP Envy 17’	1,300	24	Office	…	121
4	HP EliteBook 850	1,900	36	Mining	…	172
5	Lenovo Yoga 13’	1,100	12	Office	…	266



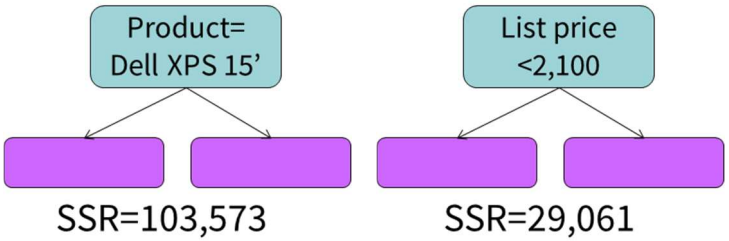


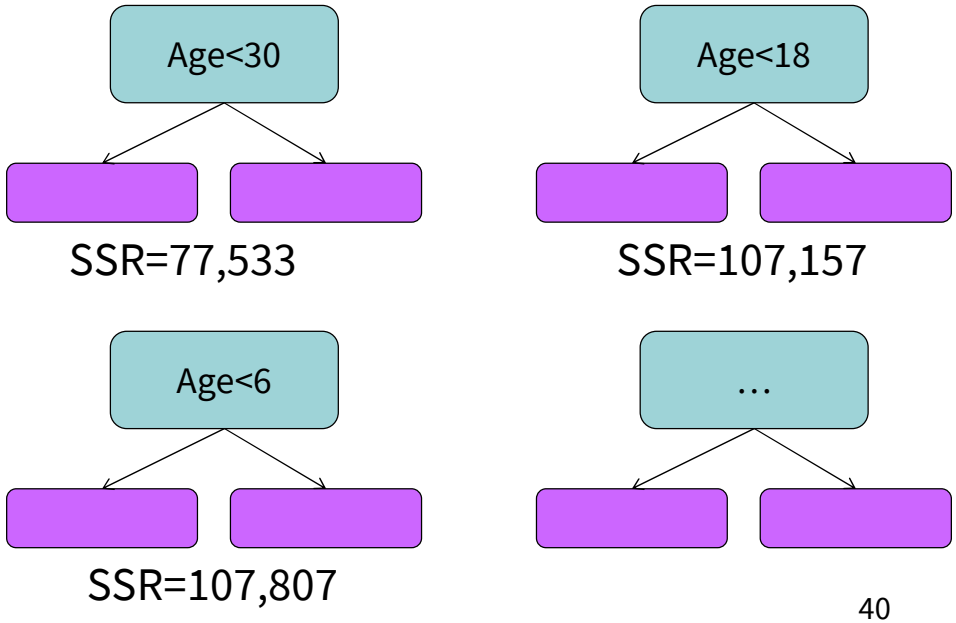
Random Forest Algorithm

Grow each tree using the random subspace mechanism

- **Random subspace modifies how we grow a tree**
- **Key decision in tree growing is finding splits**
  - Standard/previous approach
    - Step 1: For each feature, find the best split on that feature
    - Step 2: Select the best split over all features

i	Product	List price	Age	Industry	…	Resale price [$]
1	Dell XPS 15’	2,500	36	Mining	…	347
2	Dell XPS 17’	3,000	36	Health	…	538
3	HP Envy 17’	1,300	24	Office	…	121
4	HP EliteBook 850	1,900	36	Mining	…	172
5	Lenovo Yoga 13’	1,100	12	Office	…	266



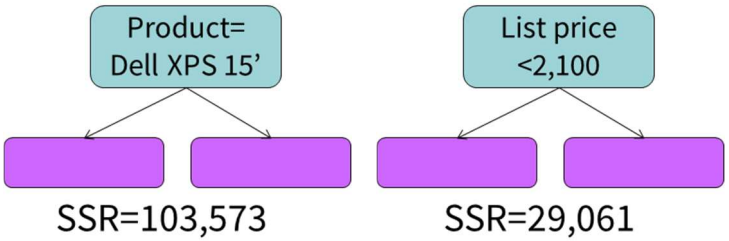


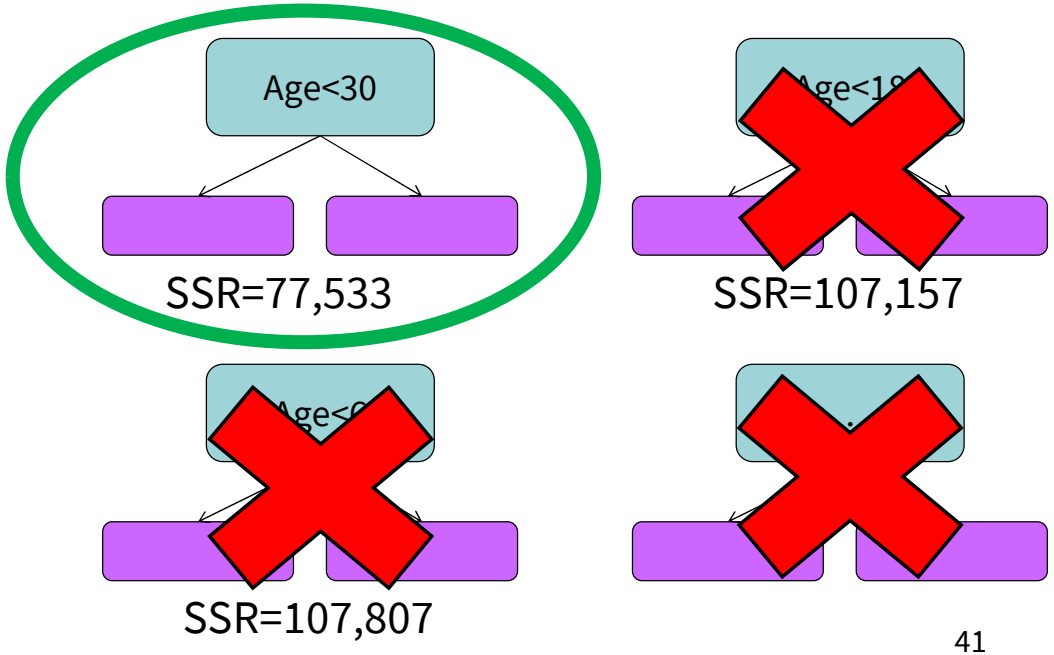
Random Forest Algorithm

Grow each tree using the random subspace mechanism

- **Random subspace modifies how we grow a tree**
- **Key decision in tree growing is finding splits**
  - Standard/previous approach
    - Step 1: For each feature, find the best split on that feature
    - Step 2: Select the best split over all features

i	Product	List price	Age	Industry	…	Resale price [$]
1	Dell XPS 15’	2,500	36	Mining	…	347
2	Dell XPS 17’	3,000	36	Health	…	538
3	HP Envy 17’	1,300	24	Office	…	121
4	HP EliteBook 850	1,900	36	Mining	…	172
5	Lenovo Yoga 13’	1,100	12	Office	…	266



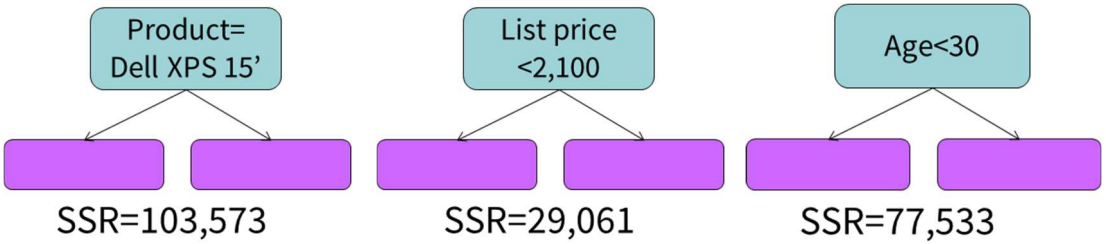


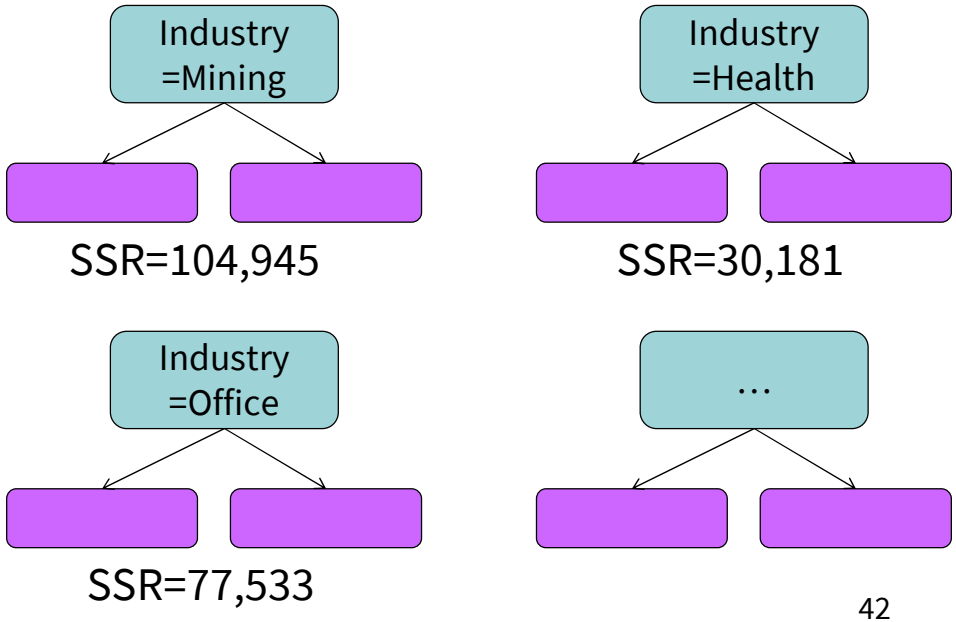
Random Forest Algorithm

Grow each tree using the random subspace mechanism

- **Random subspace modifies how we grow a tree**
- **Key decision in tree growing is finding splits**
  - Standard/previous approach
    - Step 1: For each feature, find the best split on that feature
    - Step 2: Select the best overall approach

i	Product	List price	Age	Industry	…	Resale price [$]
1	Dell XPS 15’	2,500	36	Mining	…	347
2	Dell XPS 17’	3,000	36	Health	…	538
3	HP Envy 17’	1,300	24	Office	…	121
4	HP EliteBook 850	1,900	36	Mining	…	172
5	Lenovo Yoga 13’	1,100	12	Office	…	266



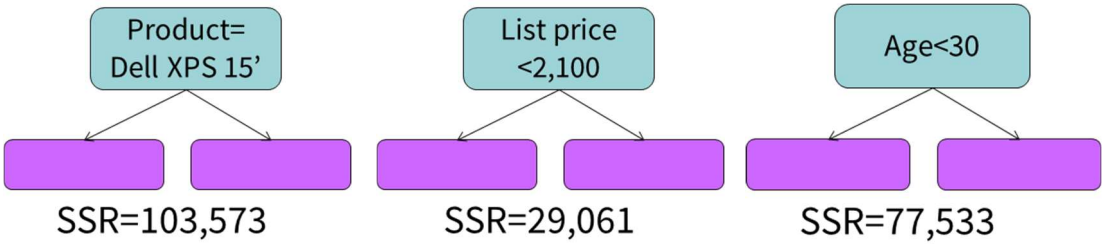


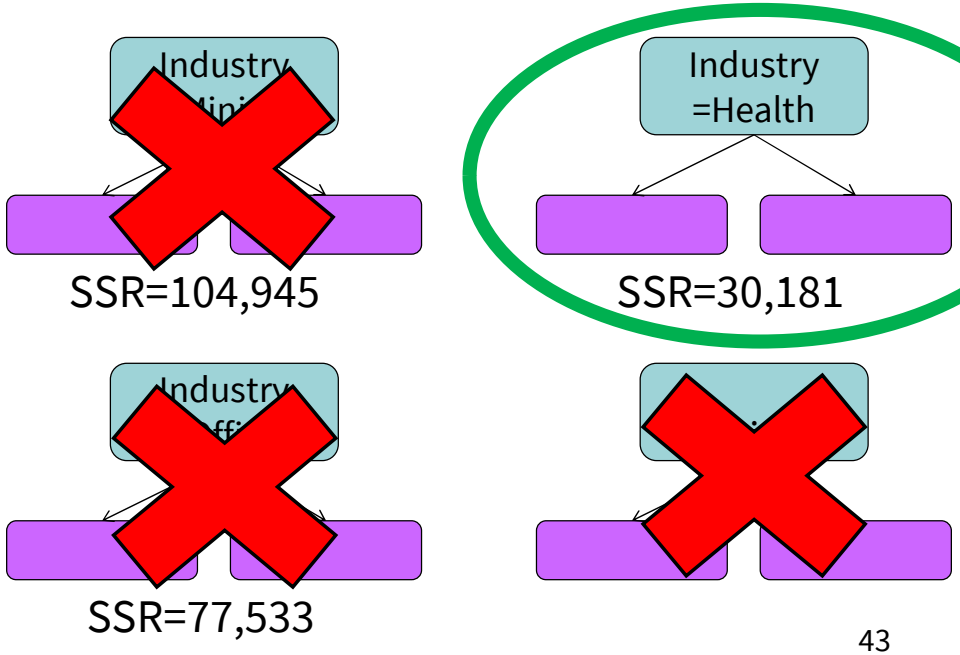
Random Forest Algorithm

Grow each tree using the random subspace mechanism

- **Random subspace modifies how we grow a tree**
- **Key decision in tree growing is finding splits**
  - Standard/previous approach
    - Step 1: For each feature, find the best split on that feature
    - Step 2: Select the best split over all features

i	Product	List price	Age	Industry	…	Resale price [$]
1	Dell XPS 15’	2,500	36	Mining	…	347
2	Dell XPS 17’	3,000	36	Health	…	538
3	HP Envy 17’	1,300	24	Office	…	121
4	HP EliteBook 850	1,900	36	Mining	…	172
5	Lenovo Yoga 13’	1,100	12	Office	…	266



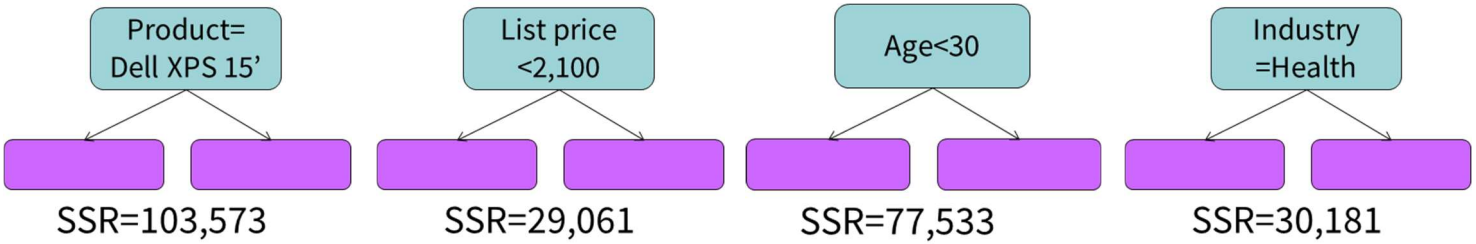


Random Forest Algorithm

Grow each tree using the random subspace mechanism

i	Product	List price	Age	Industry	…	Resale price [$]
1	Dell XPS 15’	2,500	36	Mining	…	347
2	Dell XPS 17’	3,000	36	Health	…	538
3	HP Envy 17’	1,300	24	Office	…	121
4	HP EliteBook 850	1,900	36	Mining	…	172
5	Lenovo Yoga 13’	1,100	12	Office	…	266

- **Random subspace modifies how we grow a tree**
- **Key decision in tree growing is finding splits**
  - Standard/previous approach
    - Step 1: For each feature, find the best split on that feature
    - Step 2: Select the best split over all features

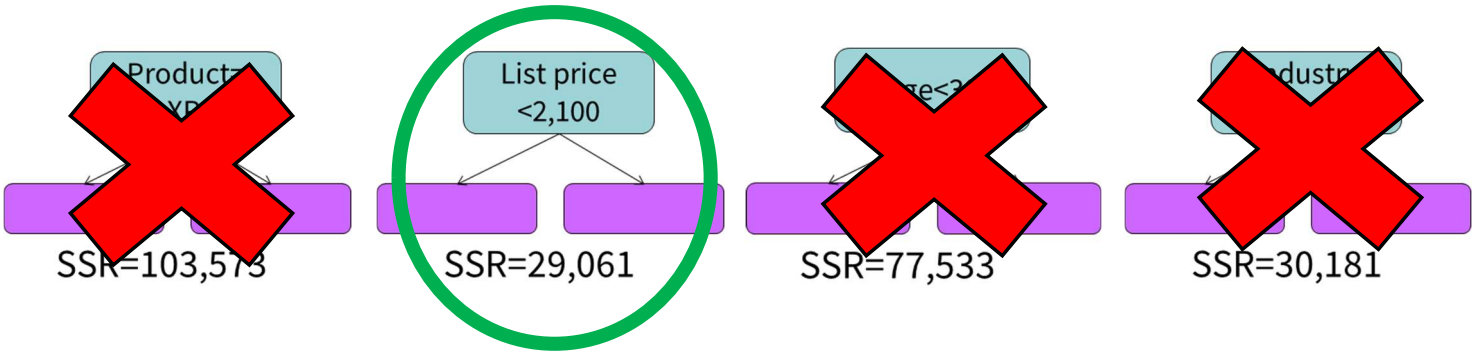


Random Forest Algorithm

Grow each tree using the random subspace mechanism

- **Random subspace modifies how we grow a tree**
- **Key decision in tree growing is finding splits**
  - Standard/previous approach
    - Step 1: For each feature, find the best split on that feature
    - Step 2: Select the best split over all features

i	Product	List price	Age	Industry	…	Resale price [$]
1	Dell XPS 15’	2,500	36	Mining	…	347
2	Dell XPS 17’	3,000	36	Health	…	538
3	HP Envy 17’	1,300	24	Office	…	121
4	HP EliteBook 850	1,900	36	Mining	…	172
5	Lenovo Yoga 13’	1,100	12	Office	…	266

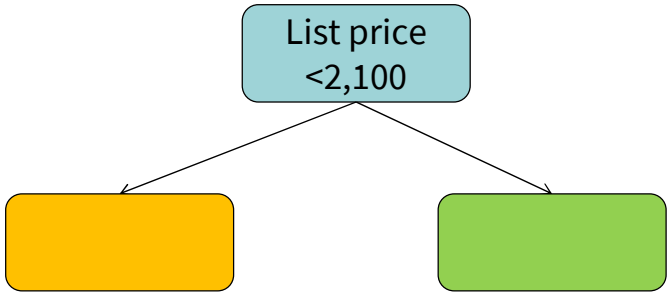


Random Forest Algorithm

Grow each tree using the random subspace mechanism

i	Product	List price	Age	Industry	…	Resale price [$]
1	Dell XPS 15’	2,500	36	Mining	…	347
2	Dell XPS 17’	3,000	36	Health	…	538
3	HP Envy 17’	1,300	24	Office	…	121
4	HP EliteBook 850	1,900	36	Mining	…	172
5	Lenovo Yoga 13’	1,100	12	Office	…	266

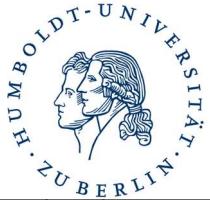
- **Random subspace modifies how we grow a tree**
- **Key decision in tree growing is finding splits**
  - Standard/previous approach
    - Step 1: For each feature, find the best split on that feature
    - Step 2: Select the best split over all features



  - Continue with finding the best split for the left and right child considering only the data in the corresponding subset

Random Forest Algorithm

Grow each tree using the random subspace mechanism

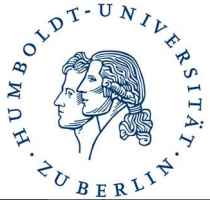


i	Product	List price	Age	Industry	…	Resale price [$]
1	Dell XPS 15’	2,500	36	Mining	…	347
2	Dell XPS 17’	3,000	36	Health	…	538
3	HP Envy 17’	1,300	24	Office	…	121
4	HP EliteBook 850	1,900	36	Mining	…	172
5	Lenovo Yoga 13’	1,100	12	Office	…	266

- **Random subspace modifies how we grow a tree**
- **Key decision in tree growing is finding splits**
  - Standard/previous approach
    - Step 1: For each feature, find the best split on that feature
    - Step 2: Select the best split over all features
  - Random subspace approach
    - Step 1a: Select a subset of features randomly
    - Step 1b: For each feature in the subset, find the best split
    - Step 2: Select the best overall approach
  - Say we set the size of the subset to 2 features and randomly draw Product and Age

Random Forest Algorithm

Grow each tree using the random subspace mechanism



i	Product	List price	Age	Industry	…	Resale price [$]
1	Dell XPS 15’	2,500	36	Mining	…	347
2	Dell XPS 17’	3,000	36	Health	…	538
3	HP Envy 17’	1,300	24	Office	…	121
4	HP EliteBook 850	1,900	36	Mining	…	172
5	Lenovo Yoga 13’	1,100	12	Office	…	266

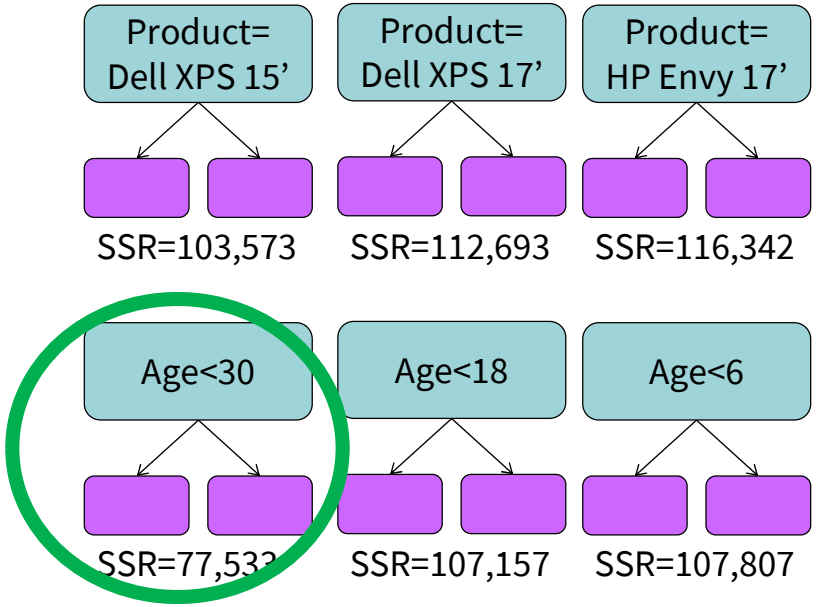
- **Random subspace modifies how we grow a tree**
- **Key decision in tree growing is finding splits**
  - Standard/previous approach
    - Step 1: For each feature, find the best split on that feature
    - Step 2: Select the best split over all features
  - Random subspace approach
    - Step 1a: Select a subset of features randomly
    - Step 1b: For each feature in the subset, find the best split
    - Step 2: Select the best overall approach
  - Say we set the size of the subset to 2 features and randomly draw Product and Age

Random Forest Algorithm

Grow each tree using the random subspace mechanism

- **Random subspace modifies how we grow a tree**
- **Key decision in tree growing is finding splits**
  - Standard/previous approach
    - Step 1: For each feature, find the best split on that feature
    - Step 2: Select the best split over all features
  - Random subspace approach
    - Step 1a: Select a subset of features randomly
    - Step 1b: For each feature in the subset, find the best split
    - Step 2: Select the best overall approach
  - Say we set the size of the subset to 2 features and randomly draw Product and Age

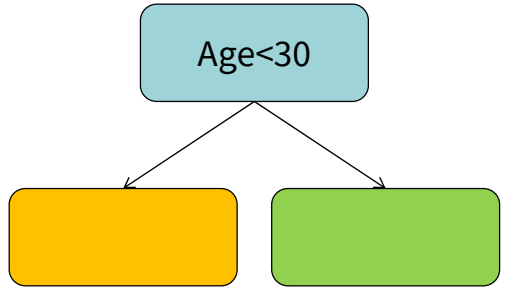
i	Product	List price	Age	Industry	…	Resale price [$]
1	Dell XPS 15’	2,500	36	Mining	…	347
2	Dell XPS 17’	3,000	36	Health	…	538
3	HP Envy 17’	1,300	24	Office	…	121
4	HP EliteBook 850	1,900	36	Mining	…	172
5	Lenovo Yoga 13’	1,100	12	Office	…	266



Random Forest Algorithm

Grow each tree using the random subspace mechanism

i	Product	List price	Age	Industry	…	Resale price [$]
1	Dell XPS 15’	2,500	36	Mining	…	347
2	Dell XPS 17’	3,000	36	Health	…	538
3	HP Envy 17’	1,300	24	Office	…	121
4	HP EliteBook 850	1,900	36	Mining	…	172
5	Lenovo Yoga 13’	1,100	12	Office	…	266



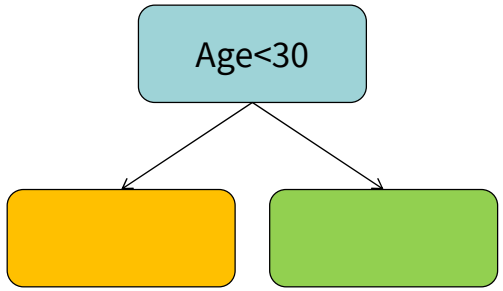
- **Random subspace modifies how we grow a tree**
- **Key decision in tree growing is finding splits**
  - Standard/previous approach
    - Step 1: For each feature, find the best split on that feature
    - Step 2: Select the best split over all features
  - Random subspace approach
    - Step 1a: Select a subset of features randomly
    - Step 1b: For each feature in the subset, find the best split
    - Step 2: Select the best overall approach
  - Say we set the size of the subset to 2 features and randomly draw Product and Age
  - Having found our first split, we continue with the left child
    - We again draw 2 features randomly & find the best split among them
    - Say our second sample includes the features List price and Age

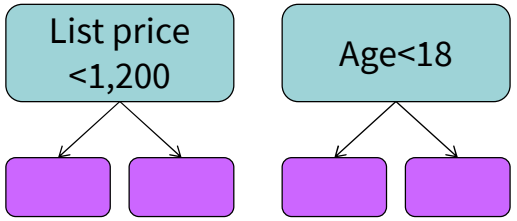
Random Forest Algorithm

Grow each tree using the random subspace mechanism

- **Random subspace modifies how we grow a tree**
- **Key decision in tree growing is finding splits**
  - Standard/previous approach
    - Step 1: For each feature, find the best split on that feature
    - Step 2: Select the best split over all features
  - Random subspace approach
    - Step 1a: Select a subset of features randomly
    - Step 1b: For each feature in the subset, find the best split
    - Step 2: Select the best overall approach
  - Say we set the size of the subset to 2 features and randomly draw Product and Age
  - Having found our first split, we continue with the left child
    - We again draw 2 features randomly & find the best split among them
    - Say our second sample includes the features List price and Age

i	Product	List price	Age	Industry	…	Resale price [$]
1	Dell XPS 15’	2,500	36	Mining	…	347
2	Dell XPS 17’	3,000	36	Health	…	538
3	HP Envy 17’	1,300	24	Office	…	121
4	HP EliteBook 850	1,900	36	Mining	…	172
5	Lenovo Yoga 13’	1,100	12	Office	…	266

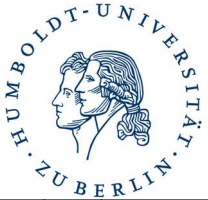




SSR is the same for these splits since we have only two data points left.

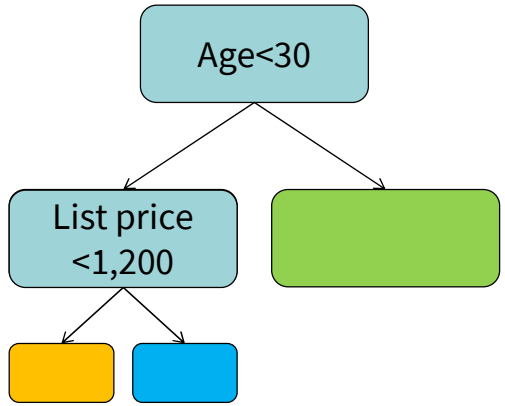
Random Forest Algorithm

Grow each tree using the random subspace mechanism



i	Product	List price	Age	Industry	…	Resale price [$]
1	Dell XPS 15’	2,500	36	Mining	…	347
2	Dell XPS 17’	3,000	36	Health	…	538
3	HP Envy 17’	1,300	24	Office	…	121
4	HP EliteBook 850	1,900	36	Mining	…	172
5	Lenovo Yoga 13’	1,100	12	Office	…	266

- **Random subspace modifies how we grow a tree**
- **Key decision in tree growing is finding splits**
  - Standard/previous approach
    - Step 1: For each feature, find the best split on that feature
    - Step 2: Select the best split over all features
  - Random subspace approach
    - Step 1a: Select a subset of features randomly
    - Step 1b: For each feature in the subset, find the best split
    - Step 2: Select the best overall approach
  - Say we set the size of the subset to 2 features and randomly draw Product and Age
  - Having found our first split, we continue with the left child
    - We again draw 2 features randomly & find the best split among them
    - Say our second sample includes the features List price and Age

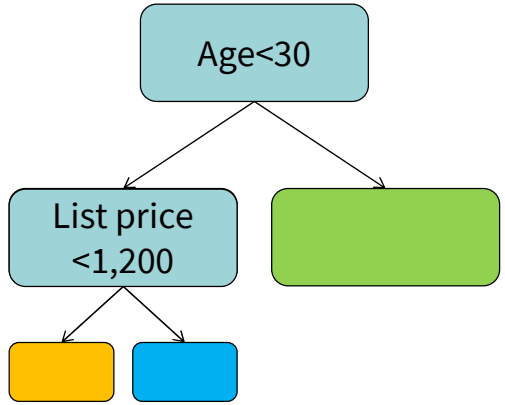


Random Forest Algorithm

Grow each tree using the random subspace mechanism

- **Random subspace modifies how we grow a tree**
- **Key decision in tree growing is finding splits**
  - Standard/previous approach
    - Step 1: For each feature, find the best split on that feature
    - Step 2: Select the best split over all features
  - Random subspace approach
    - Step 1a: Select a subset of features randomly
    - Step 1b: For each feature in the subset, find the best split
    - Step 2: Select the best overall approach
  - And then we continue in just the same way with the right child, and other child nodes further down in the tree
    - Split nodes just as with ordinary trees
    - But search the next split among a random subset of features

i	Product	List price	Age	Industry	…	Resale price [$]
1	Dell XPS 15’	2,500	36	Mining	…	347
2	Dell XPS 17’	3,000	36	Health	…	538
3	HP Envy 17’	1,300	24	Office	…	121
4	HP EliteBook 850	1,900	36	Mining	…	172
5	Lenovo Yoga 13’	1,100	12	Office	…	266



Random Forest in Pseudo-Code

Training

- Draw T bootstrap samples from the training set
- For each bootstrap sample
  - Grow a regression tree
    - Draw a random sample of features
    - Find best split among those features
      - Assess candidate splits as in ordinary regression trees
    - Repeat until tree is fully grown
  - Add grown tree to the ensemble
- Output forest of T trees

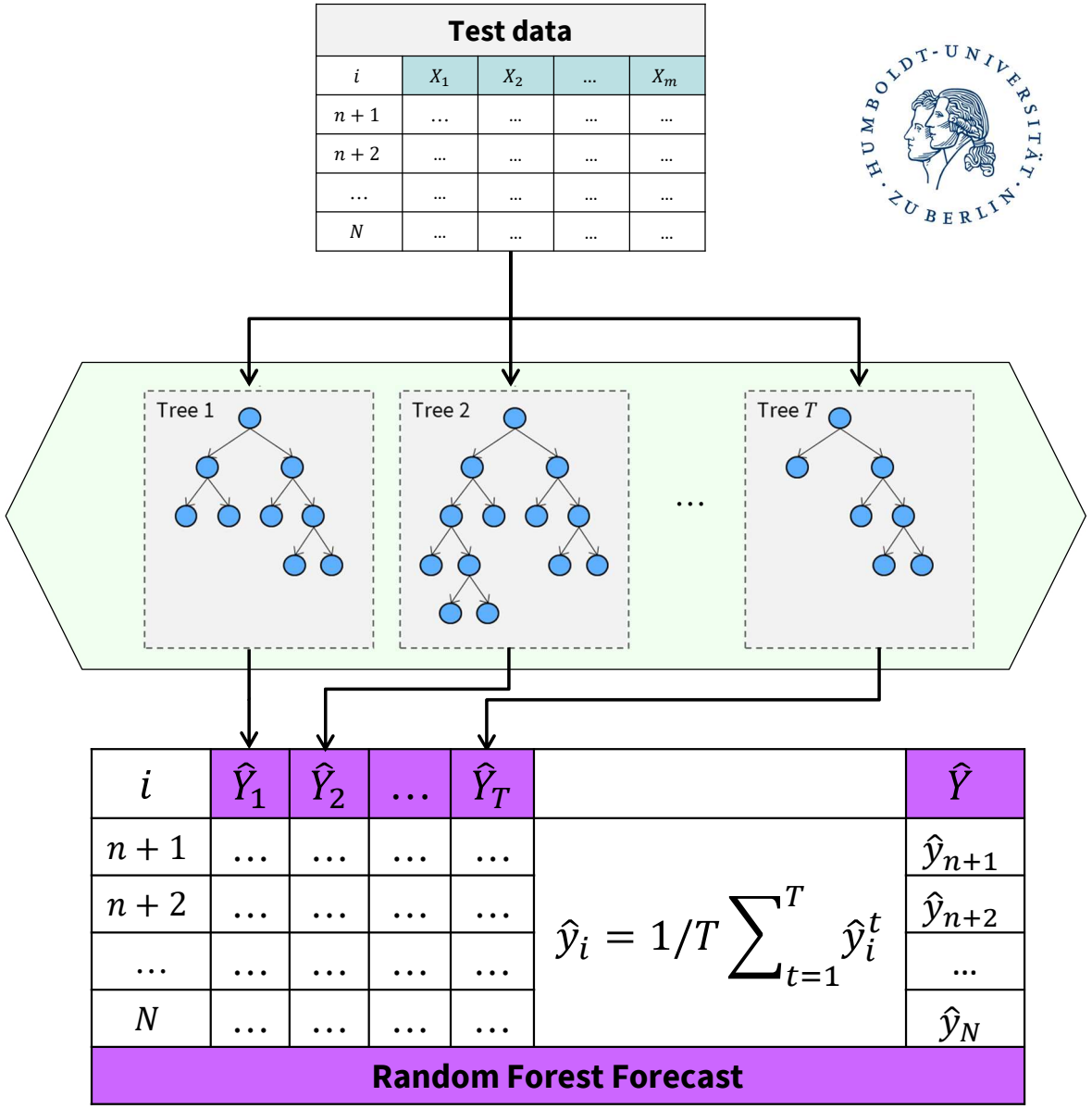
Random Forest in Pseudo-Code

Training

- Draw T bootstrap samples from the training set
- For each bootstrap sample
  - Grow a regression tree
    - Draw a random sample of features
    - Find best split among those features
      - Assess candidate splits as in ordinary regression trees
    - Repeat until tree is fully grown
  - Add grown tree to the ensemble
- Output forest of T trees

Testing

- Let each tree forecast a new example
- Compute ensemble forecast as simple average over the base tree forecasts



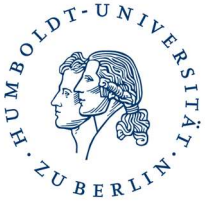
The Random Forest Algorithm

Random Forest and the Bias-Variance-Trade-Off

- **Tree-based models tend to overfit the data**
  - Always possible to obtain a perfect fit of the training data
  - Just find as many rules as there are cases in the training data
- **Pruning is a way to avoid overfitting**
- **Breiman (2001) recommends to not prune the trees of a random forest**
- **Fully-grown decision trees (without pruning)**
  - Have zero bias (by definition)
  - Have high variance (controlled via bootstrapping)
- **How Random Forest increases predictive accuracy**
  - Using base models without bias while
  - Reducing their variance through bootstrapping

The Random Forest Algorithm

Meta-Parameter Tuning in Random Forest



As in bagging

- **Base model algorithm**
  - Preferably tree-based or neural network, but can use any
  - Could tune meta-parameters of the base model
- **How many bootstrap samples (i.e., ensemble size)**
- **Size of bootstrap sample**
  - Often overlooked
  - Useful when working with large data sets
- **Number of attributes sampled at random at each split (often called mtry)**
- **Practical advice:**
  - The larger the ensemble the better (due to variance reduction)
  - Important to tune mtry
    - With m denoting the no. of attributes, rule of thumb is to start with $mtry = \sqrt{m}$
    - Continue with half/twice $\sqrt{m}$

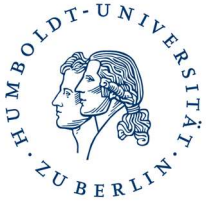
Why Random Forest Works Best with Trees?

- **Developing a random forest can be computationally costly. Assume you work with a large data set and cannot afford building a large forest. The following approaches will reduce runtime**
  - Limit the number of trees in the forest
  - Reduce the size of the bootstrap sample, from which every tree is grown
- **Which strategy will give better results?**

The Gradient Boosting Algorithm

Boosting principle, gradient boosting for regression, state-of-the-art implementations

The Boosting Principle



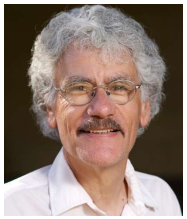
- **Begin with a simple model (often called weak learner)**
- **Calculate the errors of that model**
- **Build a second model that ‘corrects’ those errors**
- **Combine the first and second model to form an ensemble**
- **Continue with**
  - Building and adding base models (one at a time)
  - That ‘correct’ the errors of the current ensemble
  - Until some stopping condition is met (typically max. iterations)
- **Principle is also called additive modeling**

Family Tree of Boosting Algorithms

- **Adaptive boosting (Adaboost) by Freund & Schapire (1997)**
  - Focussed on classification
  - Uses data point weights for base model fitting
- **Gradient boosting (GBM) by Friedman (2001, 2002)**
  - Generic ensemble learning framework for regression, classification, ranking, etc.
  - Fits base models to negative gradients
- **Extreme gradient boosting (XGB) by Chen & Guestrin (2016)**
  - Revision of the gradient-based optimization and addition of regularization
  - Major improvement of the scalability of gradient boosting
- **Latest revisions**
  - LightGBM (Microsoft) by Ke et al. (2017) focuses on scalability
  - Catboost (Yandex) by Prokhorenkova et al. (2018) focuses on categorical features
  - NGBoost (Stanford) by Duan et al. (2019) introduces uncertainty estimates


R. Shapire


Y. Freund

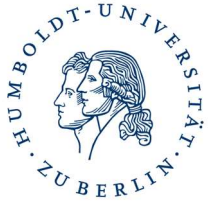

J. Friedman


T. Chen


C. Guestrin

Gradient Boosting Algorithm

Fit trees to the *residuals* of an *incrementally* grown ensemble model

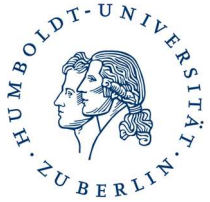


- **Incremental ensemble growing means we start with one base model (tree)**
  - In gradient boosting for regression the first base model is just a constant
  - The best constant forecast is the average of the target over the training set
- **Consider our demo data set for illustration**
  - The average resale price (i.e., target) for this data is $\frac{1}{5}(347 + 538 + 212 + 172 + 266) = 288.8$
  - In regression tree language, the average corresponds to the output of a tree with only one node

i	Product	List price	Age	Industry	…	Resale price [$]
1	Dell XPS 15’	2,500	36	Mining	…	347
2	Dell XPS 17’	3,000	36	Health	…	538
3	HP Envy 17’	1,300	24	Office	…	121
4	HP EliteBook 850	1,900	36	Mining	…	172
5	Lenovo Yoga 13’	1,100	12	Office	…	266

Gradient Boosting Algorithm

Fit trees to the *residuals* of an *incrementally* grown ensemble model

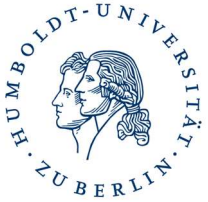


- **Incremental ensemble growing means we start with one base model (tree)**
  - In gradient boosting for regression the first base model is just a constant
  - The best constant forecast is the average of the target over the training set
- **Consider our demo data set for illustration**
  - The average resale price (i.e., target) for this data is $\frac{1}{5}(347 + 538 + 212 + 172 + 266) = 288.8$
  - In regression tree language, the average corresponds to the output of a tree with only one node

i	Product	List price	Age	Industry	…	Resale price [$]	$\hat{Y}^0$
1	Dell XPS 15’	2,500	36	Mining	…	347	288.8
2	Dell XPS 17’	3,000	36	Health	…	538	288.8
3	HP Envy 17’	1,300	24	Office	…	121	288.8
4	HP EliteBook 850	1,900	36	Mining	…	172	288.8
5	Lenovo Yoga 13’	1,100	12	Office	…	266	288.8

Gradient Boosting Algorithm

Fit trees to the *residuals* of an *incrementally* grown ensemble model

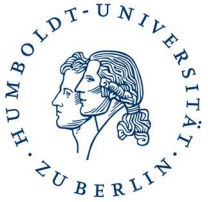


- **Having our first base model, we calculate residuals**

i	Product	List price	Age	Industry	…	Resale price [$]	$\hat{Y}^0$
1	Dell XPS 15’	2,500	36	Mining	…	347	288.8
2	Dell XPS 17’	3,000	36	Health	…	538	288.8
3	HP Envy 17’	1,300	24	Office	…	121	288.8
4	HP EliteBook 850	1,900	36	Mining	…	172	288.8
5	Lenovo Yoga 13’	1,100	12	Office	…	266	288.8

Gradient Boosting Algorithm

Fit trees to the *residuals* of an *incrementally* grown ensemble model



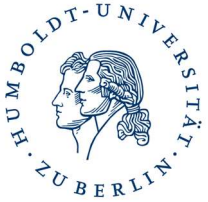
- **Having our first base model, we calculate residuals**
- **Gradient boosting proceeds by fitting the next tree to the residuals of the current ensemble, which for now includes only the first tree**

Price =288.8 t=0

i	Product	List price	Age	Industry	…	Resale price [$]	$\hat{Y}^0$	$Y - \hat{Y}^0$
1	Dell XPS 15’	2,500	36	Mining	…	347	288.8	347 - 288.8 = 58.2
2	Dell XPS 17’	3,000	36	Health	…	538	288.8	538 - 288.8 = 249.2
3	HP Envy 17’	1,300	24	Office	…	121	288.8	121 - 288.8 = -167.8
4	HP EliteBook 850	1,900	36	Mining	…	172	288.8	172 - 288.8 = -116.8
5	Lenovo Yoga 13’	1,100	12	Office	…	266	288.8	266 - 288.8 = -22.8
							Total SSR	**107,807**

Gradient Boosting Algorithm

Fit trees to the *residuals* of an *incrementally* grown ensemble model



- **Data set in next round incorporates residuals as new target**
- **Fit a tree to this new target**

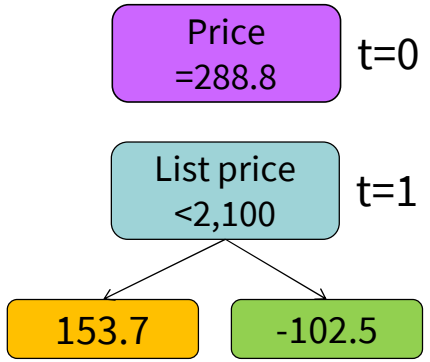
Price =288.8 t=0

i	Product	List price	Age	Industry	…	Resale price [$]	$\epsilon^{\mathrm{GBM}_0}$
1	Dell XPS 15’	2,500	36	Mining	…	347	58.2
2	Dell XPS 17’	3,000	36	Health	…	538	249.2
3	HP Envy 17’	1,300	24	Office	…	121	-167.8
4	HP EliteBook 850	1,900	36	Mining	…	172	-116.8
5	Lenovo Yoga 13’	1,100	12	Office	…	266	-22.8
					Total SSR		**107,807**

Gradient Boosting Algorithm

Fit trees to the *residuals* of an *incrementally* grown ensemble model

- **Data set in next round incorporates residuals as new target**
- **Fit a tree to this new target**

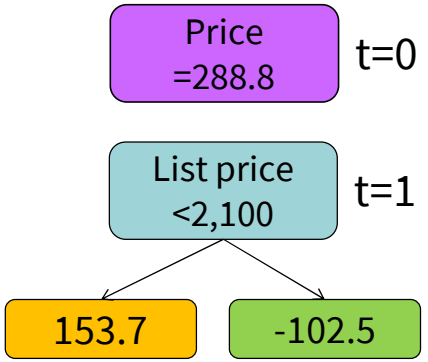


i	Product	List price	Age	Industry	…	Resale price [$]	$\epsilon^{\mathrm{GBM}_0}$
1	Dell XPS 15’	2,500	36	Mining	…	347	58.2
2	Dell XPS 17’	3,000	36	Health	…	538	249.2
3	HP Envy 17’	1,300	24	Office	…	121	-167.8
4	HP EliteBook 850	1,900	36	Mining	…	172	-116.8
5	Lenovo Yoga 13’	1,100	12	Office	…	266	-22.8
					Total SSR		**107,807**

Gradient Boosting Algorithm

Fit trees to the *residuals* of an *incrementally* grown ensemble model

- **Data set in next round incorporates residuals as new target**
- **Fit a tree to this new target**
  - We compute tree predictions as usual
  - Just that we predict the residuals of the previous tree

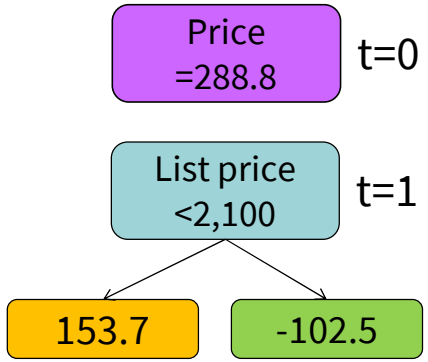


i	Product	List price	Age	Industry	…	Resale price [$]	ϵ^{GBM_0}	$\hat{Y}^1$
1	Dell XPS 15’	2,500	36	Mining	…	347	58.2	153.7
2	Dell XPS 17’	3,000	36	Health	…	538	249.2	153.7
3	HP Envy 17’	1,300	24	Office	…	121	-167.8	-102.5
4	HP EliteBook 850	1,900	36	Mining	…	172	-116.8	-102.5
5	Lenovo Yoga 13’	1,100	12	Office	…	266	-22.8	-102.5
					Total SSR		**107,807**	

Gradient Boosting Algorithm

Fit trees to the *residuals* of an *incrementally* grown ensemble model

- **Data set in next round incorporates residuals as new target**
- **Now that we have two models, we need to decide on how to compute the ensemble forecast**
  - We update the forecast such that residuals will decrease
  - While not changing the forecast too much
  - Rather make many small steps in the right direction



i	Product	List price	Age	Industry	…	Resale price [$]	ϵ^{GBM_0}	$\hat{Y}^1$
1	Dell XPS 15’	2,500	36	Mining	…	347	58.2	153.7
2	Dell XPS 17’	3,000	36	Health	…	538	249.2	153.7
3	HP Envy 17’	1,300	24	Office	…	121	-167.8	-102.5
4	HP EliteBook 850	1,900	36	Mining	…	172	-116.8	-102.5
5	Lenovo Yoga 13’	1,100	12	Office	…	266	-22.8	-102.5
					Total SSR		**107,807**	

Mathematically:

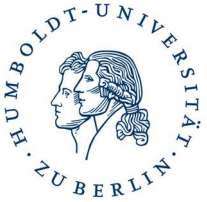
$$\hat{Y}^{\text{GBM}_t} = \hat{Y}^{\text{GBM}_{t-1}} + \eta \epsilon^{\text{GBM}_{t-1}}$$

$$\hat{Y}^{\text{GBM}_t} = \hat{Y}^{\text{GBM}_{t-1}} - \eta \frac{\partial J(w)}{\partial w}$$

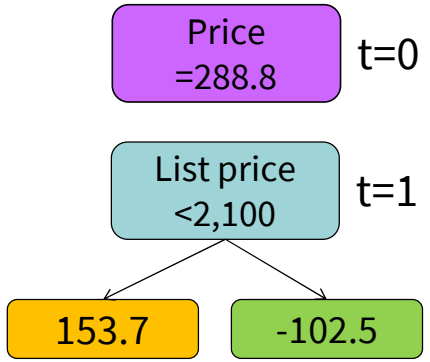
Note: η (eta) is a hyperparameter in (0,1) called the learning rate. It governs how much we update forecasts in each iteration of GBM.

Gradient Boosting Algorithm

Fit trees to the *residuals* of an *incrementally* grown ensemble model



- **Data set in next round incorporates residuals as new target**
- **Fit a tree to this new target**
- **Having computed the ensemble forecast, we proceed with computing residuals**
  - Given the way we computed the ensemble forecast $\hat{Y}^{\mathrm{GBM}_1}$
  - The residuals $\epsilon^{\mathrm{GBM}_1}$ are guaranteed to be lower than before



i	Product	List price	Age	Industry	…	Resale price [$]	$\epsilon^{\mathrm{GBM}_0}$	$\hat{Y}^1$	$\hat{Y}^{\mathrm{GBM}_1} = \hat{Y}^{\mathrm{GBM}_0} + \eta\hat{Y}^1$	$Y - \hat{Y}^{\mathrm{GBM}_1}$
1	Dell XPS 15’	2,500	36	Mining	…	347	58.2	153.7	288.8+0.1*153.7=304.2	42.8
2	Dell XPS 17’	3,000	36	Health	…	538	249.2	153.7	288.8+0.1*153.7=304.2	233.8
3	HP Envy 17’	1,300	24	Office	…	121	-167.8	-102.5	288.8+0.1*-102.5=278.6	-157.6
4	HP EliteBook 850	1,900	36	Mining	…	172	-116.8	-102.5	288.8+0.1*-102.5=278.6	-106.6
5	Lenovo Yoga 13’	1,100	12	Office	…	266	-22.8	-102.5	288.8+0.1*-102.5=278.6	-12.6
					Total SSR		**107,807**			**92,845**

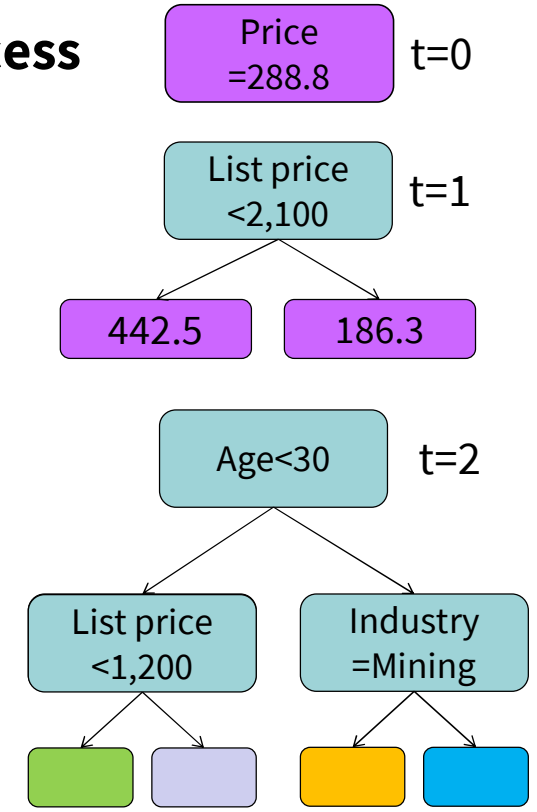
Note: η (eta) is a meta-parameter in (0,1) called the learning rate. We set it to 0.1 in the example.

Gradient Boosting Algorithm

Fit trees to the *residuals* of an *incrementally* grown ensemble model

- **In round 2, and subsequent rounds, we repeat the whole process**
  - Use residuals of previous round as target for the next base model
  - Fit next tree to new target
  - Update ensemble forecast $\widehat{Y}^{\mathrm{GBM}_2}$
  - Compute residuals $\epsilon^{\mathrm{GB}\ 2}$, and so on
- **Note: in practice, GBM uses shallow tree as base models**

i	Product	List price	Age	Industry	…	Resale price [$]	$\epsilon^{\mathrm{GBM}_1}$
1	Dell XPS 15’	2,500	36	Mining	…	347	42.8
2	Dell XPS 17’	3,000	36	Health	…	538	233.8
3	HP Envy 17’	1,300	24	Office	…	121	-157.6
4	HP EliteBook 850	1,900	36	Mining	…	172	-106.6
5	Lenovo Yoga 13’	1,100	12	Office	…	266	-12.6
					Total SSR		**92,845**

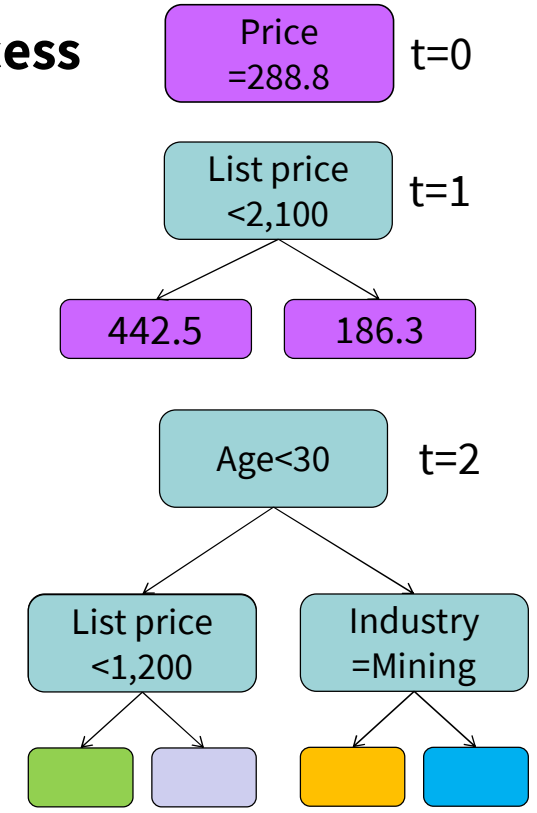


Gradient Boosting Algorithm

Fit trees to the *residuals* of an *incrementally* grown ensemble model

- **In round 2, and subsequent rounds, we repeat the whole process**
  - Use residuals of previous round as target for the next base model
  - Fit next tree to new target
  - Update ensemble forecast $\widehat{Y}^{\mathrm{GB}\ 2}$
  - Compute residuals $\epsilon^{\mathrm{GB}\ 2}$, and so on
- **Note: in practice, GBM uses shallow tree as base models**

i	Product	List price	Age	Industry	…	Resale price [$]	$\epsilon^{\mathrm{GBM}_1}$
1	Dell XPS 15’	2,500	36	Mining	…	347	42.8
2	Dell XPS 17’	3,000	36	Health	…	538	233.8
3	HP Envy 17’	1,300	24	Office	…	121	-157.6
4	HP EliteBook 850	1,900	36	Mining	…	172	-106.6
5	Lenovo Yoga 13’	1,100	12	Office	…	266	-12.6
					Total SSR		**92,845**

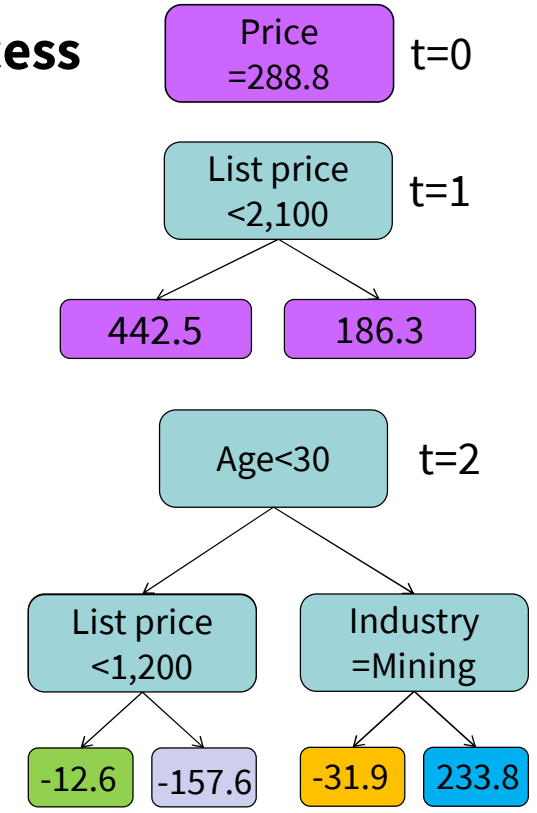


Gradient Boosting Algorithm

Fit trees to the *residuals* of an *incrementally* grown ensemble model

- **In round 2, and subsequent rounds, we repeat the whole process**
  - Use residuals of previous round as target for the next base model
  - Fit next tree to new target
  - Update ensemble forecast $\hat{Y}^{\mathrm{GB}\ 2}$
  - Compute residuals $\epsilon^{\mathrm{GB}\ 2}$, and so on
- **Note: In practice, GBM uses shallow tree as base models**

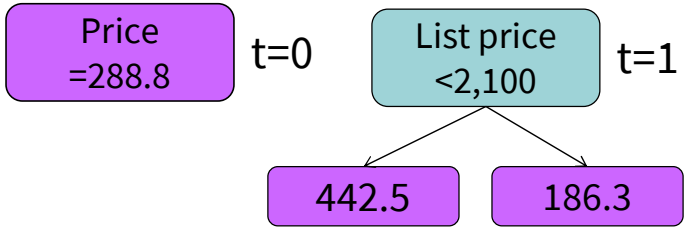
i	Product	List price	Age	Industry	…	Resale price [$]	$\epsilon^{\mathrm{GBM}_1}$	$\hat{Y}^2$
1	Dell XPS 15’	2,500	36	Mining	…	347	42.8	-31.9
2	Dell XPS 17’	3,000	36	Health	…	538	233.8	233.8
3	HP Envy 17’	1,300	24	Office	…	121	-157.6	-157.6
4	HP EliteBook 850	1,900	36	Mining	…	172	-106.6	31.9
5	Lenovo Yoga 13’	1,100	12	Office	…	266	-12.6	-12.6
					Total SSR		**92,845**	

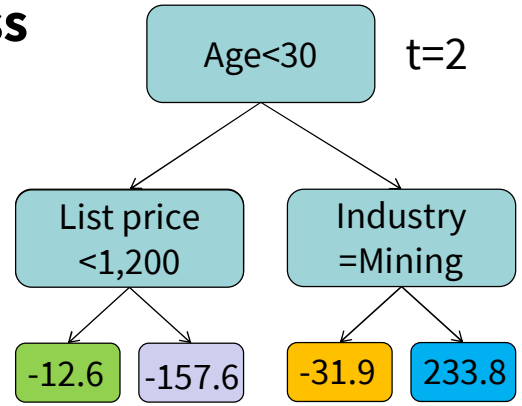


Gradient Boosting Algorithm

Fit trees to the *residuals* of an *incrementally* grown ensemble model

- **In round 2, and subsequent rounds, we repeat the whole process**
  - Use residuals of previous round as target for the next base model
  - Fit next tree to new target
  - Update ensemble forecast $\hat{Y}^{\mathrm{GBM}_2}$
  - Compute residuals $\epsilon^{\mathrm{GBM}_2}$, and so on





- **Note: In practice, GBM uses shallow tree as base models**

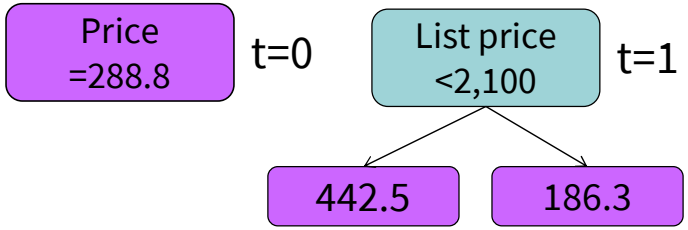
i	Product	List price	Age	Industry	…	Resale price [$]	$\epsilon^{\mathrm{GB}\ 1}$	$\hat{Y}^2$	$\hat{Y}^{\mathrm{GBM}_2} = \hat{Y}^{\mathrm{GB}\ 1} + \eta\hat{Y}^2$
1	Dell XPS 15’	2,500	36	Mining	…	347	42.8	-31.9	304.2+0.1*31.9=301.0
2	Dell XPS 17’	3,000	36	Health	…	538	233.8	233.8	304.2+0.1*233.8=327.6
3	HP Envy 17’	1,300	24	Office	…	121	-157.6	-157.6	278.6+0.1*-157.6=262.8
4	HP EliteBook 850	1,900	36	Mining	…	172	-106.6	31.9	278.6+0.1*31.9=275.4
5	Lenovo Yoga 13’	1,100	12	Office	…	266	-12.6	-12.6	278.6+0.1*-12.6=277.3
					Total SSR		**92,845**		

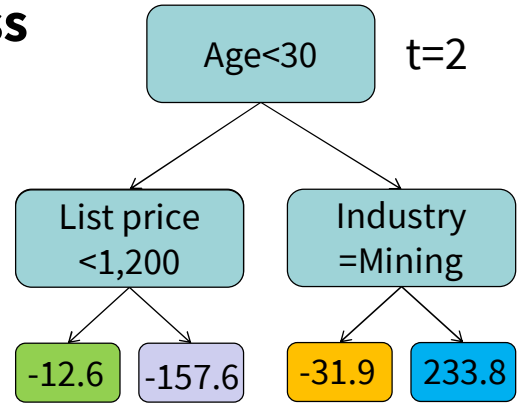
Note: η (eta) is a meta-parameter in (0,1) called the learning rate. We set it to 0.1 in the example.

Gradient Boosting Algorithm

Fit trees to the *residuals* of an *incrementally* grown ensemble model

- **In round 2, and subsequent rounds, we repeat the whole process**
  - Use residuals of previous round as target for the next base model
  - Fit next tree to new target
  - Update ensemble forecast $\hat{Y}^{\mathrm{GB}_2}$
  - Compute residuals ϵ^{GB_2}, and so on





- **Note: In practice, GBM uses shallow tree as base models**

i	Product	List price	Age	Industry	…	Resale price [$]	$\epsilon^{\mathrm{GBM}_1}$	$\hat{Y}^2$	$\hat{Y}^{\mathrm{GBM}_2} = \hat{Y}^{\mathrm{GB}_1} + \eta\hat{Y}^2$	$Y - \hat{Y}^{\mathrm{GBM}_2}$
1	Dell XPS 15’	2,500	36	Mining	…	347	42.8	-31.9	304.2+0.1*31.9=301.0	46.0
2	Dell XPS 17’	3,000	36	Health	…	538	233.8	233.8	304.2+0.1*233.8=327.6	210.4
3	HP Envy 17’	1,300	24	Office	…	121	-157.6	-157.6	278.6+0.1*-157.6=262.8	-141.8
4	HP EliteBook 850	1,900	36	Mining	…	172	-106.6	31.9	278.6+0.1*31.9=275.4	-103.4
5	Lenovo Yoga 13’	1,100	12	Office	…	266	-12.6	-12.6	278.6+0.1*-12.6=277.3	-11.3
					Total SSR		**92,845**			**77,325**

Note: η (eta) is a meta-parameter in (0,1) called the learning rate. We set it to 0.1 in the example.

Gradient Boosting in Pseudo-Code

Training

Compute average of the target and store as initial forecast $\widehat{Y}^{GB\ 0}$

Compute residuals $\boldsymbol{\epsilon}^{GBM_0}$ as $\boldsymbol{\epsilon}^{GBM_0} = Y - \hat{Y}^{GBM_0}$

For $t = 1, \ldots, T$

- Fit regression tree to training set $\mathcal{D}^t = \{\boldsymbol{X}, \boldsymbol{\epsilon}^{GB\ t-1}\}$
- Compute output of tree t, $\hat{Y}^t$, on data $\mathcal{D}^t$
- Compute new GBM forecast as $\hat{Y}^{GB\ t} = \hat{Y}^{GB\ t-1} + \eta\hat{Y}^t$
- Compute new residuals as $\boldsymbol{\epsilon}^{GBM_t} = Y - \hat{Y}^{GBM_t}$
- Monitor development of total SSR on validation data
- Stop if validation error increases (i.e., overfitting)

Testing

Let each tree forecast a new example

Compute ensemble forecast as illustrated in training stage

Gradient Boosting More Formally

Training

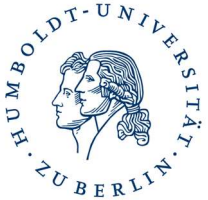
- **Given**
  - Training data $\mathcal{D} = \{Y_i, \boldsymbol{X}_i\}_{i=1}^n$ with target variable Y and feature falues $\boldsymbol{X} \in \mathbb{R}^m$,
  - Base learning algorithm h, with $h(\boldsymbol{X}) = \hat{Y}$ denoting the prediction of a model trained using h for case $\boldsymbol{X}$
  - A differentiable loss function $J(Y, H(\boldsymbol{X}))$, where $H(\boldsymbol{X})$ denotes the ensemble forecast
  - Number of iterations T
- **Initialize** $H(\boldsymbol{X})$ with a constant value: $H^{(0)}(\boldsymbol{X}) := \text{argmin}_{\gamma_0} \sum_{i=1}^n J(y_i, \gamma_0)$
- **For** $t = 1, \dots T$
  - Compute pseudo − residuals: $\epsilon_i^t = -\left[\frac{\partial J\left(y, H^{(t-1)}(\boldsymbol{X})\right)}{\partial H^{(t-1)}(\boldsymbol{X})}\right]_{\boldsymbol{X}=\boldsymbol{x}_i, Y=y_i} \quad \forall\, i = 1, \dots, n$
  - Fit base learner h_t to training set $\left(\boldsymbol{X}_i, \epsilon_i^t\right)_{i=1}^n$
  - Compute base model weight $\gamma_t = \text{argmin}_\gamma \sum_{i=1}^n J\left(y_i, H^{(t-1)}(\boldsymbol{X}_i) + \gamma h_t(\boldsymbol{X}_i)\right)$
  - Update ensemble forecast: $H^{(t)}(\boldsymbol{X}) := H^{(t-1)}(\boldsymbol{X}) + \gamma_t h_t(\boldsymbol{X})$
- **Output** final ensemble $H = \{(h_t, \gamma_t)\}_{i=0}^T$

Testing

- **Compute** ensemble forecast $H(\boldsymbol{X}) = \hat{Y} = \gamma_0 + \sum_{t=1}^T \gamma_t h_t(\boldsymbol{X})$

Gradient Boosting Algorithm

Contemporary libraries add several features and improvements

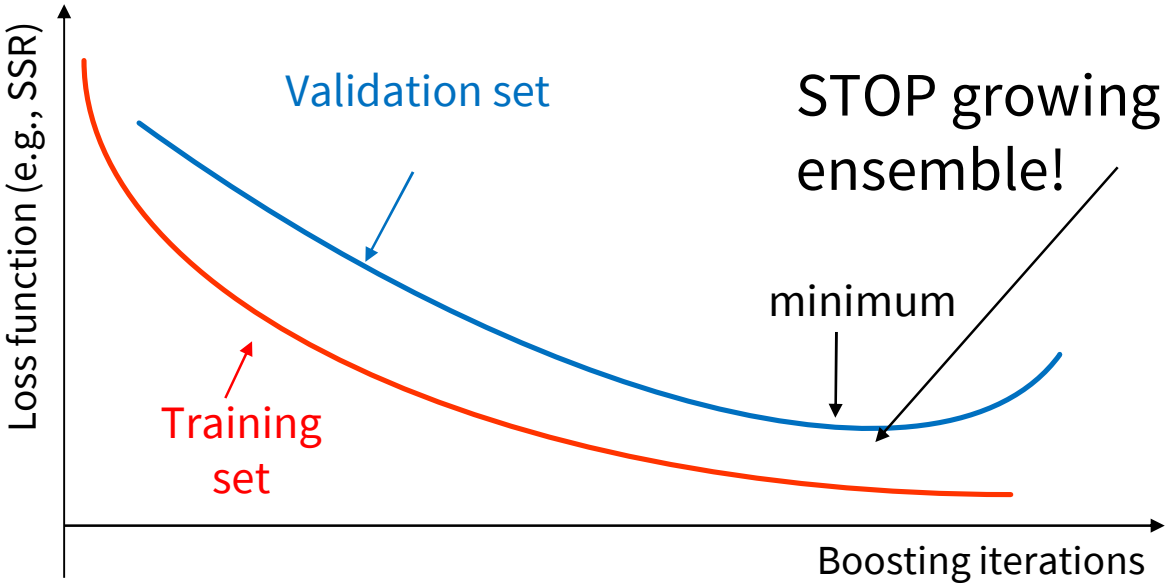


- **Specific adjustments when using decision trees as base learners**
  - Recall GBM approach to compute base model weight $\gamma_t = \text{argmin}_\gamma \sum_{i=1}^{n} J\left(y_i, H^{(t-1)}(\boldsymbol{X}_i) + \gamma h_t(\boldsymbol{X}_i)\right)$
  - Replace weight γ_t with an individual weight per leaf γ_t^j , where j indexes the leaf nodes in tree t
- **Shrinkage**
  - Revise update equation to govern updates explicitly by a learning rate
  - $H^{(t)}(\boldsymbol{X}) = H^{(t-1)}(\boldsymbol{X}) + \eta \cdot \gamma_t h_t(\boldsymbol{X})$ with learning rate or shrinkage parameter η
- **Stochastic gradient boosting (Friedman, 2002)**
  - Borrow variance reducing principles from bagging and random forest
  - Fit base learners on bootstrap samples of the training data $\left(\boldsymbol{X}_i, H^{(0)}(\boldsymbol{X}_i) - Y_i\right)_{i=1}^{n}$
  - Use random subspace for base learner training (i.e., when growing a new tree)
- **Regularization to penalize tree complexity**

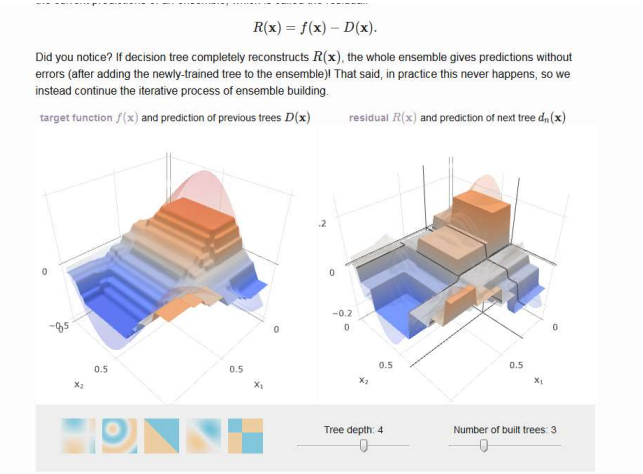
Gradient Boosting Algorithm

Practical use and configurations

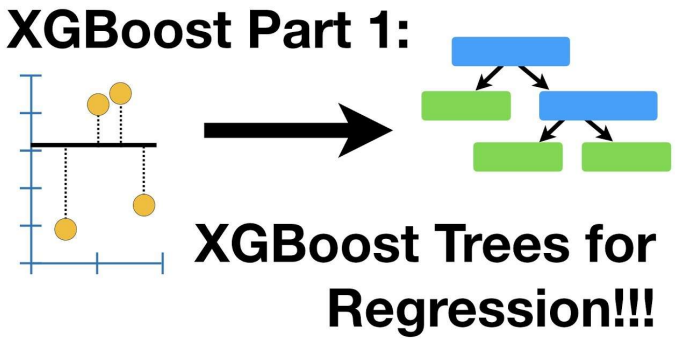
- **GBM repeats the above steps many times**
- **We use validation data to avoid overfitting (just as in ordinary trees)**
- **GBM hyperparameters**
  - Ensemble size *T*, often between 100 and 500
  - Tree depth, often between 3 and 12
  - Learning rate, always between (0,1)
  - Modern SW implementations offer more hyperparameters
    - Bootstrap sampling
    - Random subspace
    - Regularization
  - Hyperparameters need some tuning



Gradient Boosting Algorithm
Outlook



- **State-of-the-art packages:**
  - XGBoost [Chen & Guestrin, 2016]
  - GBMLight [Microsoft]
  - CatBoost [Yandex]
- **Further readings and resources**
  - Gradient boosting explained
    - Visual demo and playground by Alex Rogozhnikov
    - Available at http://arogozhnikov.github.io/2016/06/24/gradient_boosting_explained.html
  - A gentle introduction to gradient boosting
    - Comprehensive coverage of the original GBM approach by Cheng Li
    - Available at: http://www.chengli.io/tutorials/gradient_boosting.pdf
  - Introduction to boosted trees
    - Boosting tutorial with focus on XGB by one of its developers Tianqi Chen
    - Available at: http://www.chengli.io/tutorials/gradient_boosting.pdf
  - StatsQuest gradient and extreme gradient boosting series on Youtube

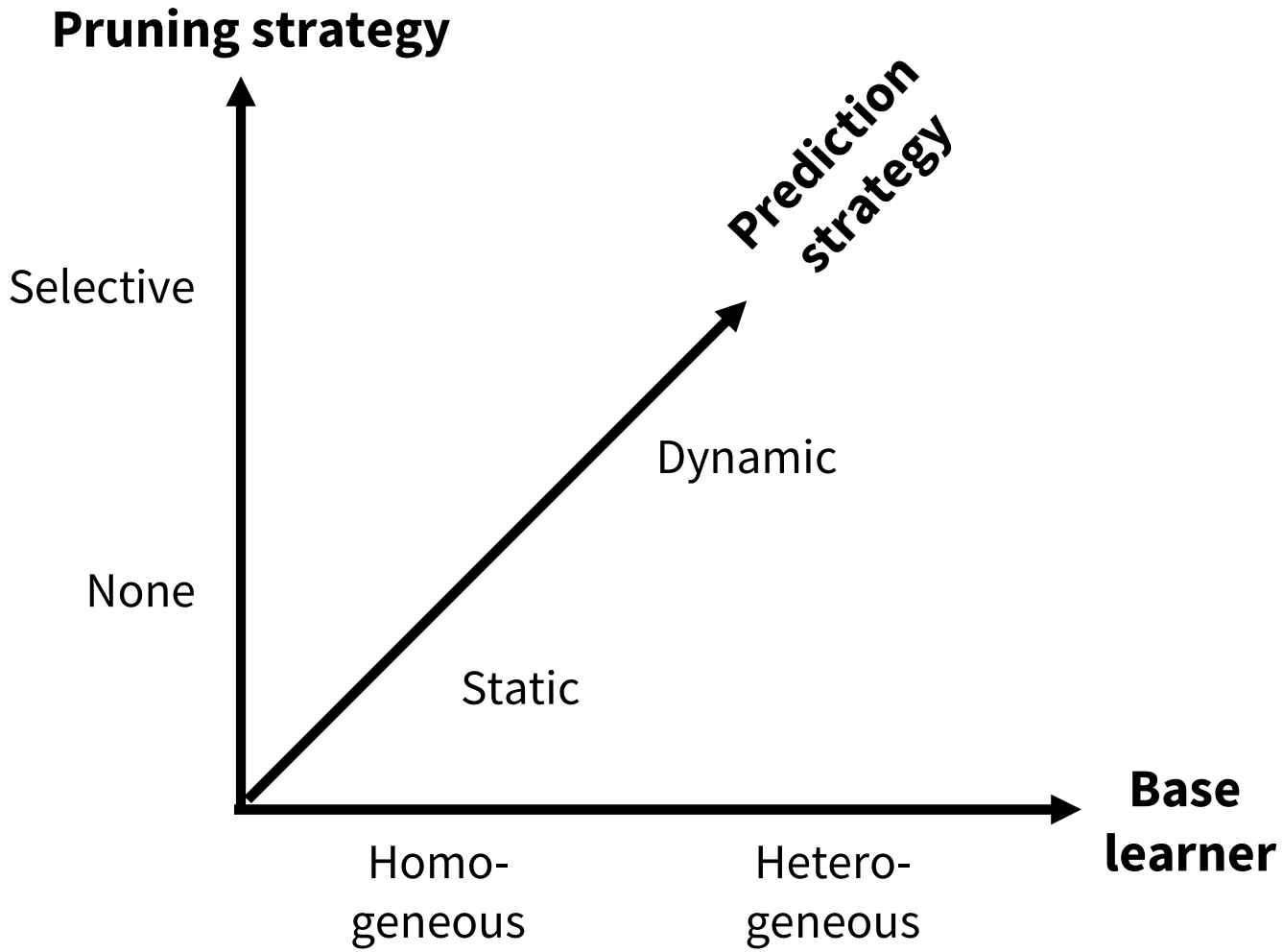


Heterogeneous Ensemble Learning

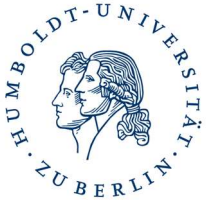
Ensemble learning strategies, the stacking algorithm

Ensemble Learning

Strategies

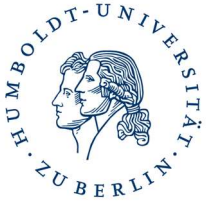


Ensemble Learning Strategies
Dimensions Base Learner



- **Homogeneous ensembles**
  - Inject diversity at the data level
    - Drawing training cases at random (e.g., Bagging)
    - Drawing variables at random (e.g., Random Subspace)
  - Use the same algorithm for base model production
- **Heterogeneous ensembles**
  - Inject diversity at the algorithm level
    - Different prediction methods for base model production (ANN, SVM, …)
    - Different meta-parameter settings per prediction method
  - All methods receive the same training data
  - Also called multiple-classifier-systems (if dealing with classification)

Ensemble Learning Strategies
Dimension Pruning Strategy



- **Ensemble models without pruning**
  - Two-step approach
    - Develop base models (homogeneous or heterogeneous)
    - Put **all** base models into the ensemble
  - Standard practice
- **Selective ensemble models**
  - Three-step approach
    - Develop **candidate** base models (typically heterogeneous)
    - Optimize ensemble composition using some search strategy
    - Put selected base models into the ensemble; discard the rest
  - Active field of research
    - Which search strategy?
    - Which objective?
- **Recommended reading: Tsoumakas et al. (2009)**

Ensemble Learning Strategies

Dimension Prediction Strategy

- **Static ensembles**
  - Develop one ensemble model
  - Use this model to predict novel cases (or cases in a test set)
  - Standard practice
  - All cases are processed by the same ensemble
- **Dynamic ensembles**
  - Identify one ensemble for each case to be predicted
  - Similar to mixture of experts model (Yuksel et al., 2012)
  - Possible actions
    - Change the base models in the ensemble
    - Change the weights of the base models in the ensemble
    - …
  - Need mechanism to decide how to adapt / form the ensemble
- **Recommended reading: Cavalin et al. (2013); Lysiak et al. (2014); Li et al. (2013)**

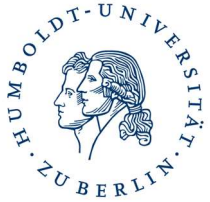
The Stacking Algorithm

- **Idea: Use 2nd level model for base model combination**
- **Classification example**
- **Data set after base model development**

ID	GOOD (0) / BAD (1)	Classifier 1 $p_1(y\|x)$	Classifier 2 $p_2(y\|x)$	…	Classifier t $p_t(y\|x)$
1	1	0.60	0.55	…	0.71
2	1	0.30	0.20	…	0.25
3	0	0.20	0.28	…	0.19
4	0	0.70	0.83	…	0.64
5	0	0.45	0.53		0.41

Actual values of the response taken from a validation sample

The Stacking Algorithm



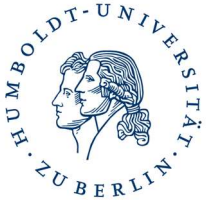
- **Idea: Use 2nd level model for base model combination**
- **Classification example**
- **Data set after base model development**

ID	GOOD (0) / BAD (1)	Classifier 1 $p_1(y\|x)$	Classifier 2 $p_2(y\|x)$	…	Classifier t $p_t(y\|x)$
1	1	0.60	0.55	…	0.71
2	1	0.30	0.20	…	0.25
3	0	0.20	0.28	…	0.19
4	0	0.70	0.83	…	0.64
5	0	0.45	0.52	…	0.41

This is exactly the format, which we use to develop base classifiers:

- A binary response variable
- Multiple "variables" (i.e., base model predictions)
- Train a (2nd level) classifier on this data to combine predictions.

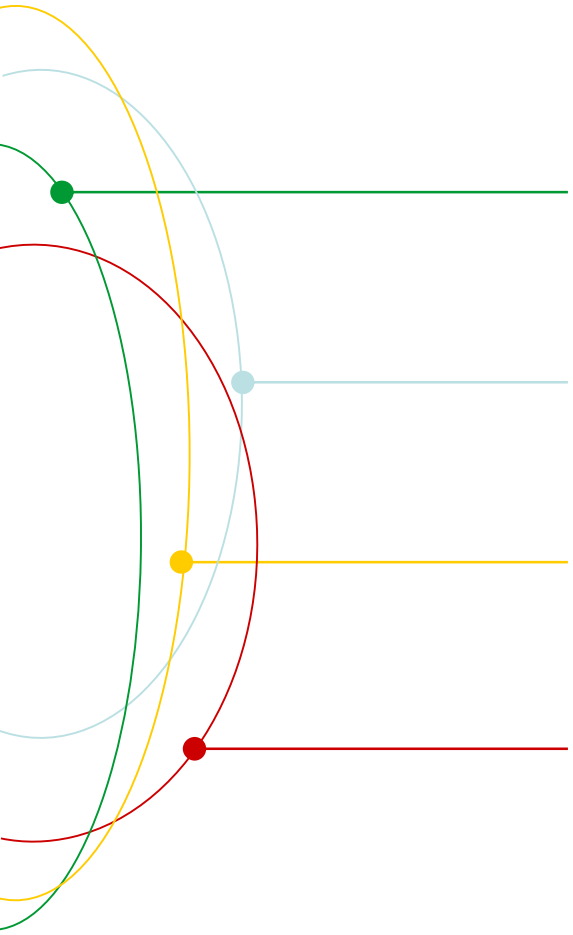
The Stacking Algorithm



- **Base model predictions are correlated**
- **Use *robust* model for 2nd level**
  - Avoid models that suffer from multicollinearity
  - Linear regression, logistic regression, discriminant analysis, etc.
- **Modeling process gets complex when 2nd level classifier requires meta-parameter tuning**
  - Which data to use for model selection?
  - Avoid models with many meta-parameters and/or model that are sensitive toward meta-parameter settings
  - Popular choice
    - Option 1: Regularized (logistic) regression
    - Option 2: Random forest using rule of thumb for mtry
- **Effectiveness of stacking**
  - Many winning entries on kaggle use stacking
  - Careful organization of the data is key to success (e.g., nested cross-validation)
  - Useful tutorial: A Kaggler's Guide to Model Stacking in Practice
    http://blog.kaggle.com/2016/12/27/a-kagglers-guide-to-model-stacking-in-practice/

Summary

Summary



Learning goals

- Principle of ensemble learning
- Overview of relevant ensemble learners

Findings

- Ensembles combine base models (often trees) to obtain a more accurate composite forecast
- Bagging & RF combine independent base models
- Boosting grows ensembles incrementally
- GBM fits base models to residuals/neg. gradients
- Stacking pools predictions by a 2nd stage model

What next

- Demo notebook on ensemble learning for credit scoring
- Lecture on feature engineering and selection

Literature

- Abuzaid, F., Bradley, J. K., Liang, F. T., Feng, A., Yang, L., Zaharia, M., & Talwalkar, A. S. (2016). Yggdrasil: An Optimized System for Training Deep Decision Trees at Scale. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon & R. Garnett (Eds.), *Advances in Neural Information Processing Systems 29*: Curran Associates, Inc.
- Breiman, L. (1996). Bagging predictors, *Machine Learning*, *24*(2), 123-140
- Breiman, L. (2001). Random forests. *Machine Learning, 45*, 5-32.
- Cavalin, P., Sabourin, R., & Suen, C. (2013). Dynamic selection approaches for multiple classifier systems. *Neural Computing and Applications, 22*, 673-688.
- Chen, T., & Guestrin, C. (2016, August 13-17, 2016). XGBoost: A Scalable Tree Boosting System. Paper presented at the Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'16), San Francisco, CA, USA.
- Domingos, P. (2000). A Unified Bias-Variance Decomposition and its Applications. In P. Langley (Ed.), *Proc. of the 17th Intern. Conf. on Machine Learning* (pp. 231-238). Stanford, CA, USA: Morgan Kaufmann.
- Duan, T., Avati, A., Ding, D. Y., Basu, S., Ng, A. Y., & Schuler, A. (2019). NGBoost: Natural Gradient Boosting for Probabilistic Prediction. Archive Preprint, arXiv:1910.03225v2.
- Freund, Y., & Schapire, R. E. (1996). Experiments With a New Boosting Algorithm. In L. Saitta (Ed.), *Proc. of the 13th Intern. Conf. on Machine Learning* (pp. 148-156). Bari, Italy: Morgan Kaufmann.
- Friedman, J., Hastie, T., & Tibshirani, R. (2000). Additive logistic regression: A statistical view of boosting. *The Annals of Statistics, 28*, 337-407.
- Friedman, J. H. (2001). Greedy function approximation: A gradient boosting machine. *The Annals of Statistics, 29*, 1189-1232.
- Friedman, J. H. (2002). Stochastic gradient boosting. *Computational Statistics & Data Analysis, 38*, 367-378.
- Ho, T. K. (1998). The random subspace method for constructing decision forests. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 20*, 832-844.
- Kuncheva, L. I., & Whitaker, C. J. (2003). Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy. Machine Learning, 51, 181-207.
- Li, L., Zou, B., Hu, Q., Wu, X., & Yu, D. (2013). Dynamic classifier ensemble using classification confidence. *Neurocomputing, 99*, 581-591.
- Lysiak, R., Kurzynski, M., & Woloszynski, T. (2014). Optimal selection of ensemble classifiers using measures of competence and diversity of base classifiers. *Neurocomputing, 126*, 29-35.
- Meng, Q., Ke, G., Wang, T., Chen, W., Ye, Q., Ma, Z., & Liu, T.-Y. (2016). A Communication-Efficient Parallel Algorithm for Decision Tree. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon & R. Garnett (Eds.), Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain (pp. 1271-1279): Curran Associates, Inc.
- Tang, E. K., Suganthan, P. N., & Yao, X. (2006). An analysis of diversity measures. Machine Learning, 65, 247-271.
- Tsoumakas, G., Partalas, I., & Vlahavas, I. (2009). An Ensemble Pruning Primer. In O. Okun & G. Valentini (Eds.), *Applications of Supervised and Unsupervised Ensemble Methods* (pp. 1-13). Berlin: Springer.
- Yuksel, S. E., Wilson, J. N., & Gader, P. D. (2012). Twenty years of mixture of experts. *IEEE Transactions on Neural Networks and Learning Systems, 23*, 1177-1193.
- Zhang, H., Si, S., & Hsieh, C.-J. (2017). GPU-acceleration for Large-scale Tree Boosting. CoRR, arXiv:1706.08359v1.

Thank you for your attention!

Stefan Lessmann

Chair of Information Systems
School of Business and Economics
Humboldt-University of Berlin, Germany

Tel. +49.30.2093.5742
Fax. +49.30.2093.5741

stefan.lessmann@hu-berlin.de
http://bit.ly/hu-wi

www.hu-berlin.de

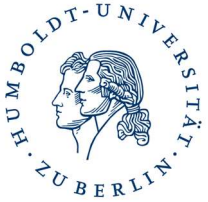
Photo: Heike Zappe

Appendix

Strength Diversity Trade-Off

The Strength Diversity Trade-Off

Theory to explain the success of ensemble learning or lack thereof

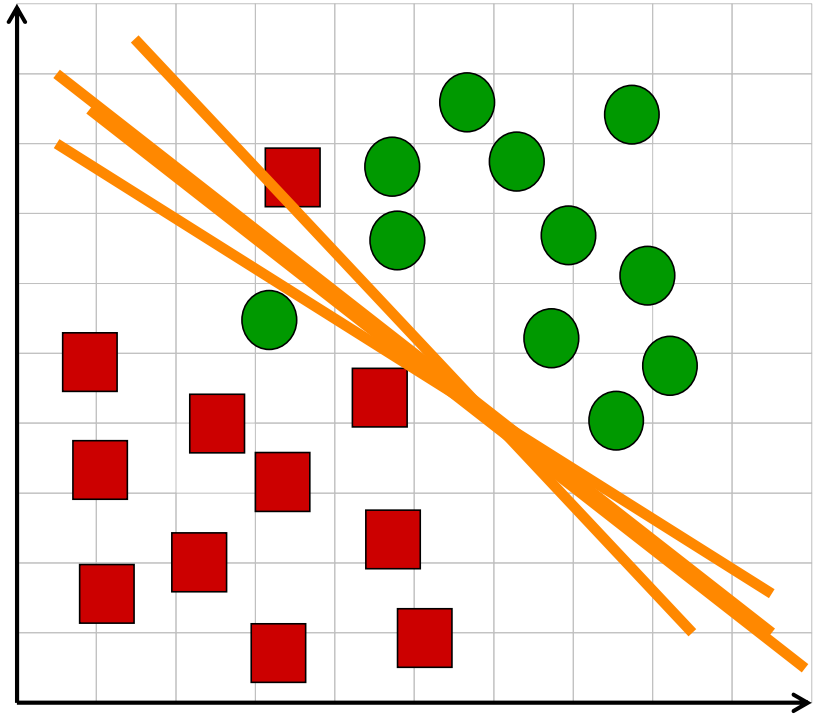


- **The success of an ensemble depends on the strength of and the diversity among the base models**
  - Strength refers to predictive accuracy
  - Diversity captures extent to which base model predictions agree (e.g., forecast or error correlation)
- **Trade-off between strength and diversity**
  - Imagine a perfect model that predicts with 100% accuracy (i.e., max strength)
  - Put this model into an ensemble
    - Average of perfect prediction with other predictions
    - No way this increases accuracy
  - All classifiers predict the same target. They cannot be very strong and diverse at the same time

The Strength Diversity Trade-Off

Ensembles Diversity Exemplified

There is no point in combining identical models. If all base models in an ensemble make the same predictions, combination cannot increase accuracy.



ID	BAD / GOOD	Classifier 1 $p_1(y\|x)$	Classifier 2 $p_2(y\|x)$	…
1	1	0.60	0.59	0.60
2	1	0.30	0.31	0.29
3	0	0.20	0.21	0.20
4	0	0.70	0.69	0.71
5	0	0.45	0.44	0.44

High forecast correlation

Ensemble Learning Theory
Diversity in Classifier Ensembles

- **Output of different classifiers have different meaning**
  - Estimates of posterior class probabilities (numeric)
  - Estimates of class membership (discrete)
- **Motivates research associated with**
  - Developing diversity measures
  - Studying characteristics of different diversity measures
  - Exploring extent to which diversity explains ensemble success
  - Examining whether diversity maximization is a good idea
- **Key take-aways**
  - Diversity is useful to **understand** different ensemble methods
  - Diversity is not necessarily a good objective to **create** ensembles

Ensemble Learning Theory
Diversity Measures Exemplified

- **Categories of diversity measures**
  - Pairwise (e.g., Q-statistic, correlation, disagreement, etc.)
  - Non-pairwise (e.g., entropy, Kohavi-Wolpert variance, etc.)
- **Often ground on cross-table of discrete class predictions**

	Classifier j correct (1)	Classifier j wrong (0)
Classifier i correct (1)	N^{11}	N^{10}
Classifier i wrong (0)	N^{01}	N^{00}

- **Suggested reading**
  - Kuncheva & Whitaker (2003)
  - Tang et al. (2006)

Ensemble Learning Theory

Diversity Measures Exemplified (cont.)

- **Several pairwise diversity measures ground on a cross-table of discrete class predictions**

	Classifier j correct (1)	Classifier j wrong (0)
Classifier i correct (1)	N^{11}	N^{10}
Classifier i wrong (0)	N^{01}	N^{00}

- **Yule's Q**

$$Q_{i,j} = \frac{N^{11} \cdot N^{00} - N^{01} \cdot N^{10}}{N^{11} \cdot N^{00} + N^{01} \cdot N^{10}}$$

- **Correlation**

$$\rho_{i,j} = \frac{N^{11} \cdot N^{00} - N^{01} \cdot N^{10}}{\sqrt{(N^{11} + N^{10})(N^{01} + N^{00})(N^{11} + N^{01})(N^{10} + N^{00})}}$$

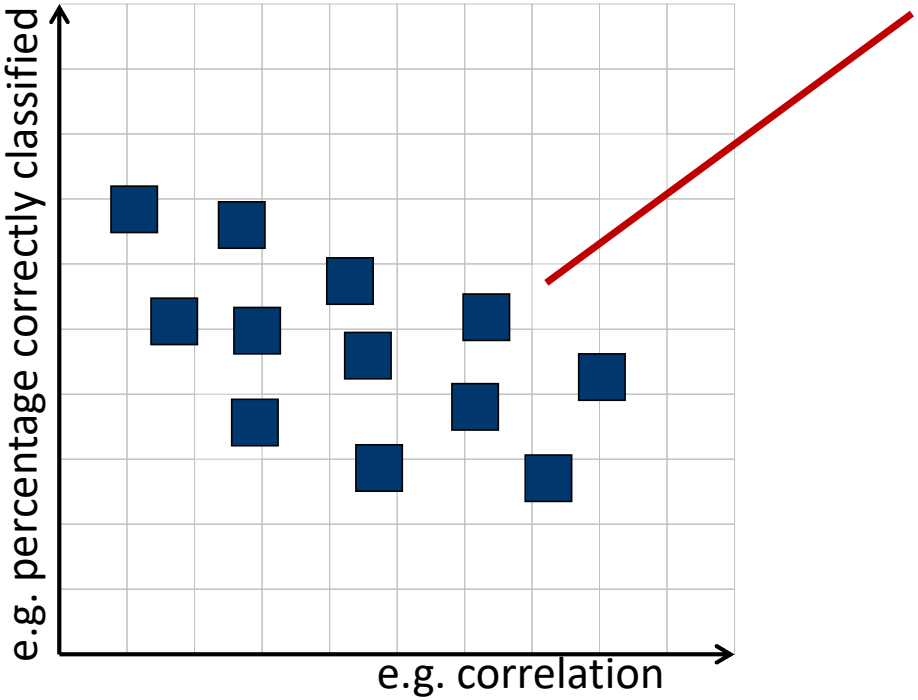
- **Disagreement**

$$Dis_{i,j} = \frac{N^{01} + N^{10}}{N^{11} + N^{00} + N^{01} + N^{10}}$$

Ensemble Learning Theory

Strength-Diversity-Plots

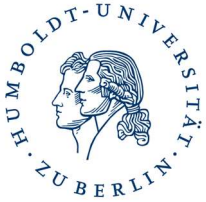
The distribution of base classifier strength and diversity in an ensemble is often illustrated using a scatter plot



- Each point in the plot corresponds to a **pair** of base models in the ensemble
- Y-axis depicts the **average** strength (e.g., PCC) of two base classifiers.
- X-axis depicts a **pairwise** diversity measure such as, e.g., correlation

Random Forest and the Strength Diversity Trade-Off

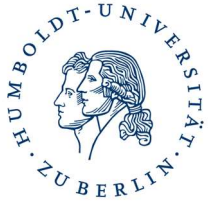
Random Forest Revisited



- **Why Random Forest Works Best with Trees?**
  - Random subspace limits access to attributes
  - Blocking of good attributes **reduces base model strength**
  - Forces tree-growing to explore different ways to split the data
  - Using more attributes **increases diversity among base models**
- **Increasing diversity in the ensemble (e.g., compared to bagging) is instrumental to random forest**
  - In a tree, random subsampling of attributes is performed at every split (i.e., many times per tree)
  - For other base learners, random subspace is performed only once per base model
  - Diversity generating mechanism used more intensively within a tree-based random forest

Random Forest and the Strength Diversity Trade-Off

Back to above question: which strategy will give better results, … ?



- **Consequences of limiting the number of trees in the forest**
  - Using many trees is crucial to reduce variance (especially when using deep trees)
  - Using many trees is also crucial to increase diversity
    - Using more trees increases the overall number of splits
    - More splits lead to more applications of random subspace
    - More applications of random subspace increase diversity in the forest
- **Consequences of reducing the size of the bootstrap sample**
  - Using less training data is always detrimental as it decreases the strength of the model
  - Base model strength is not key to random forest
  - Smaller bootstrap samples also increase diversity
- **Better to use larger forest; reduce bootstrap sample size if needed**

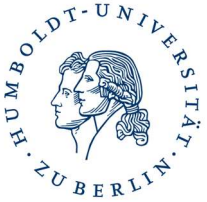
Ensemble Learning Theory
Ensemble Margin

- **Ensemble margin as another measure to explain ensemble success**
- **Formal definition**
  - O_{ij}: “oracle output” of classifier j on example i (equals +1 if x_i is classified correctly, and -1 otherwise)
  - m_i: Ensemble margin on example i
  - w_j: Weight of classifier j within the ensemble (e.g., when averaging predictions)

$$m_i = \sum_{j=1}^{L} w_j O_{i,j} \quad ; \text{ where } w_j \geq 0 \text{ and } \sum_j w_j = 1$$

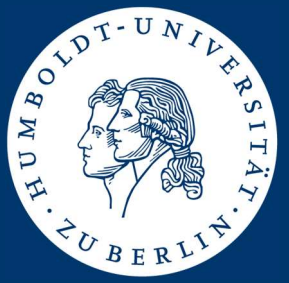
Ensemble Learning Theory

Ensemble Margin



- **Several studies show the generalization ability of an ensemble to depend on the margin distribution on the training sample.**
- **Larger margin improves performance**
- **Generalization error upper-bounded by ensemble margin**

ID	GOOD (0) BAD (1)	Base model class predictions				Ensemble margin m_i
		BM-1 (w=0.25)	**BM-2** (w=0.25)	**BM-3** (w=0.25)	**BM-4** (w=0.25)	
1	0	0	1	0	0	+0.25-0.25+0.25+0.25 = **0.5**
2	0	1	0	0	1	-0.25+0.25+0.25-0.25 = **0.0**
3	0	0	0	0	1	+0.25+0.25+0.25-0.25 = **0.5**
4	1	1	1	1	0	+0.25+0.25+0.25-0.25 = **0.5**
5	1	0	0	1	0	-0.25-0.25+0.25-0.25 = **-0.5**
6	1	1	1	1	1	+0.25+0.25+0.25+0.25 = **1.0**

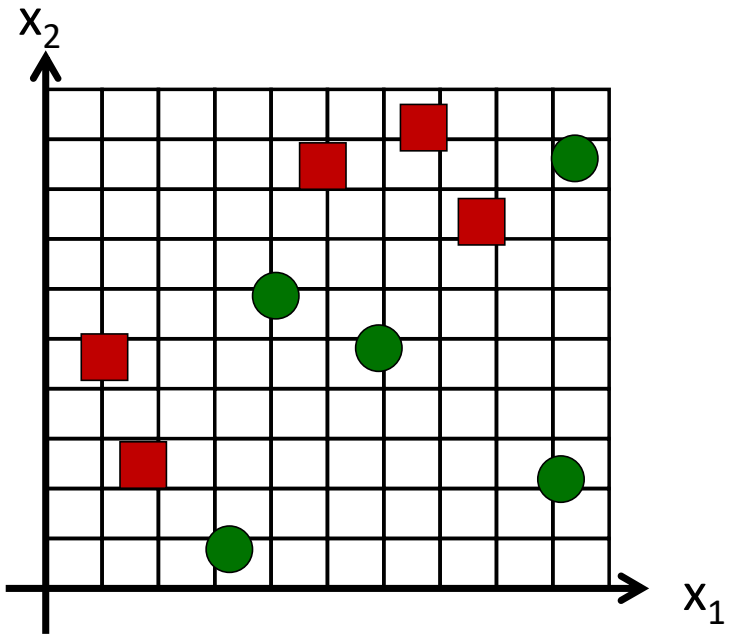


Appendix

The Adaboost algorithm

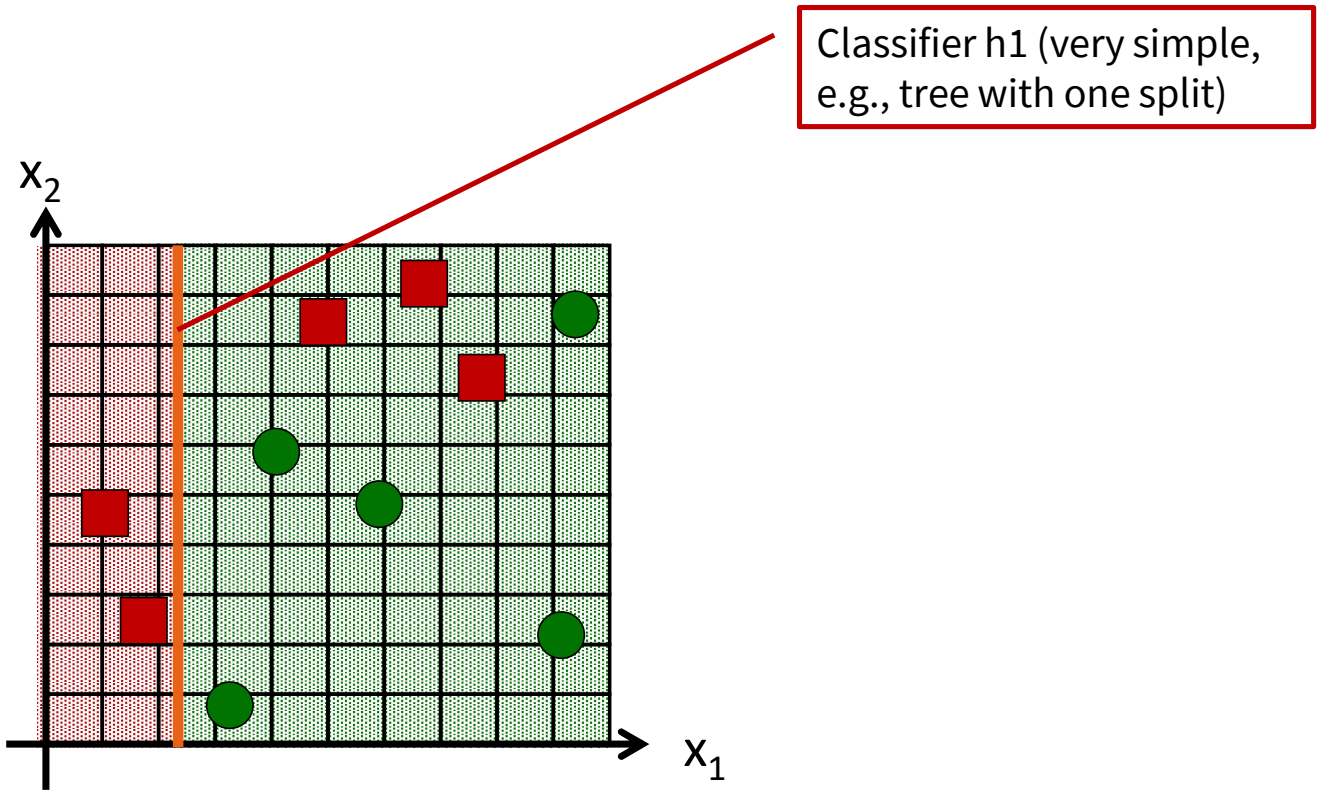
Classification With Adaboost – A Visual Intuition in 2 D

Original data



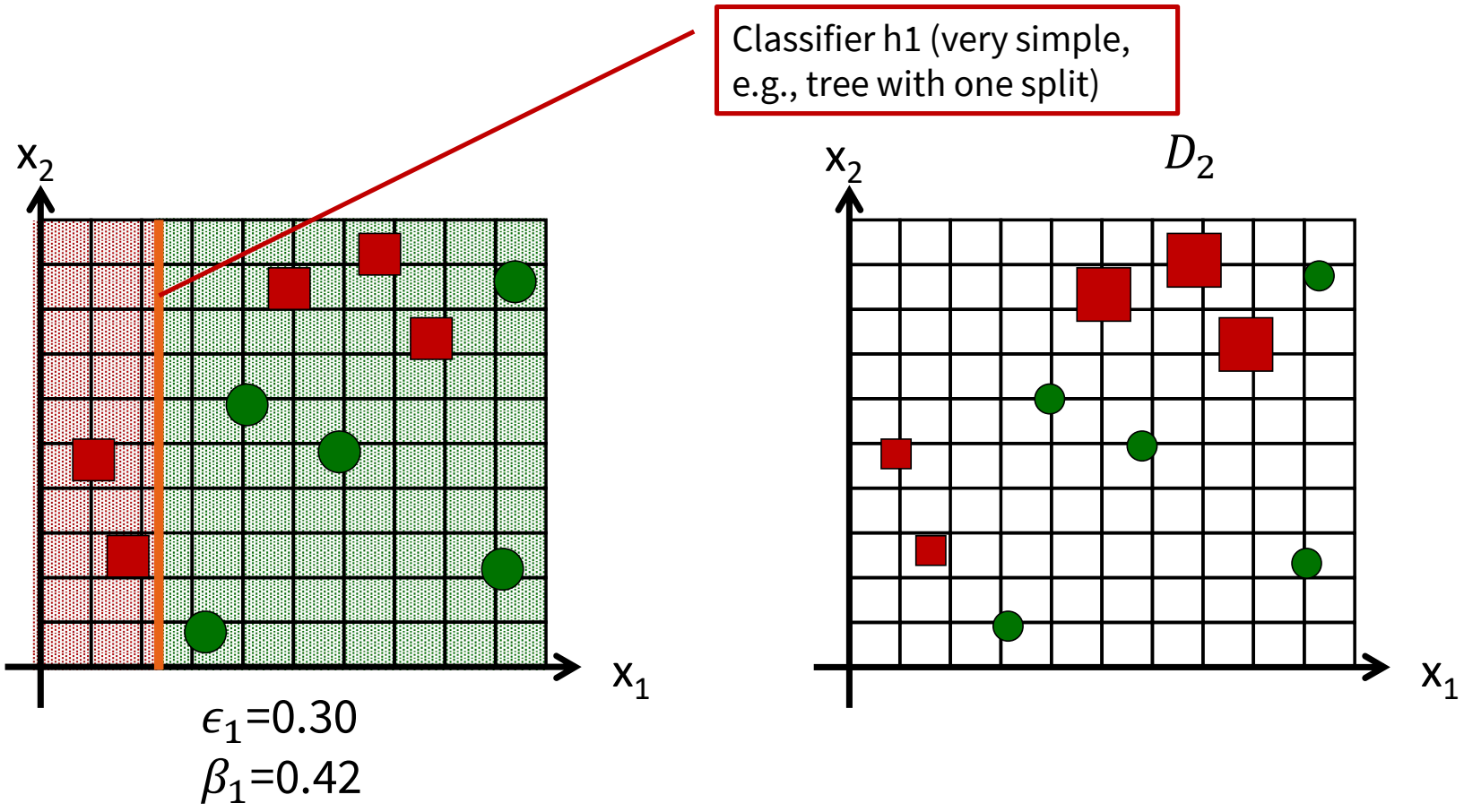
Classification With Adaboost – A Visual Intuition in 2 D

Iteration 1: first weak classifier



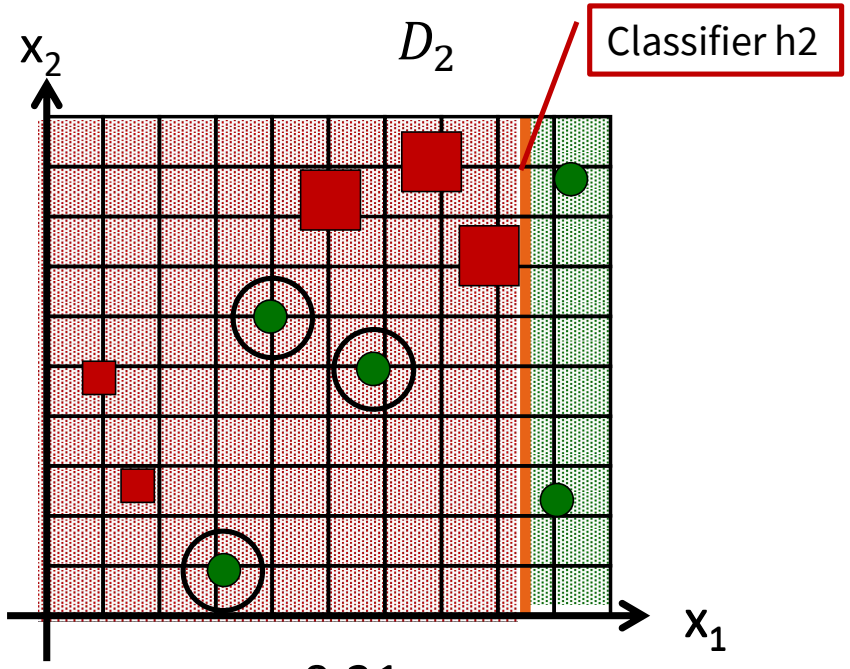
Classification With Adaboost – A Visual Intuition in 2 D

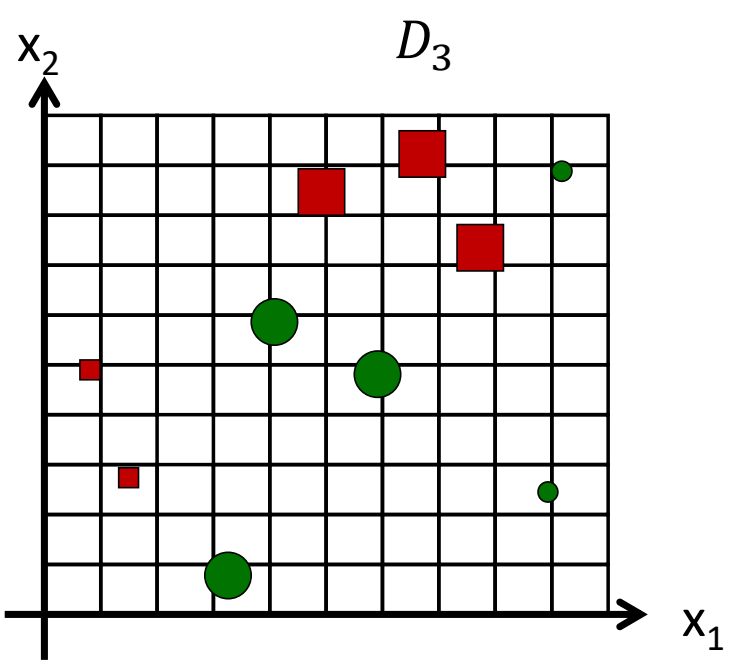
Iteration 1: reweighting examples based on error



Classification With Adaboost – A Visual Intuition in 2 D

Iteration 2: second weak classifier and reweighting of examples

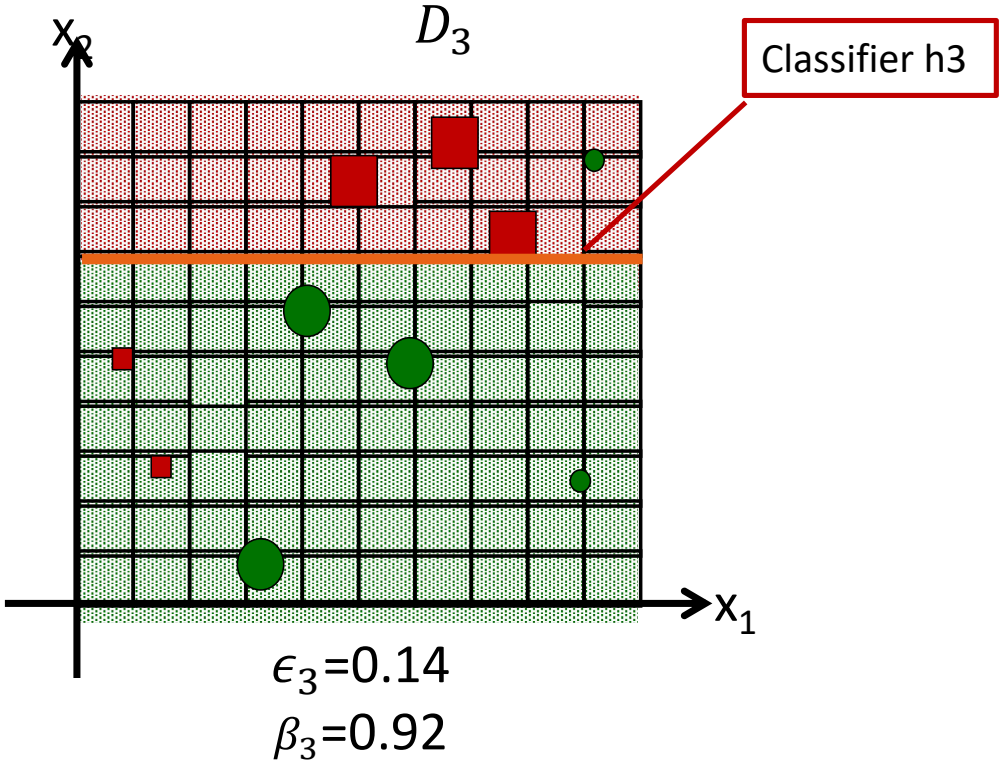




Classification With Adaboost – A Visual Intuition in 2 D

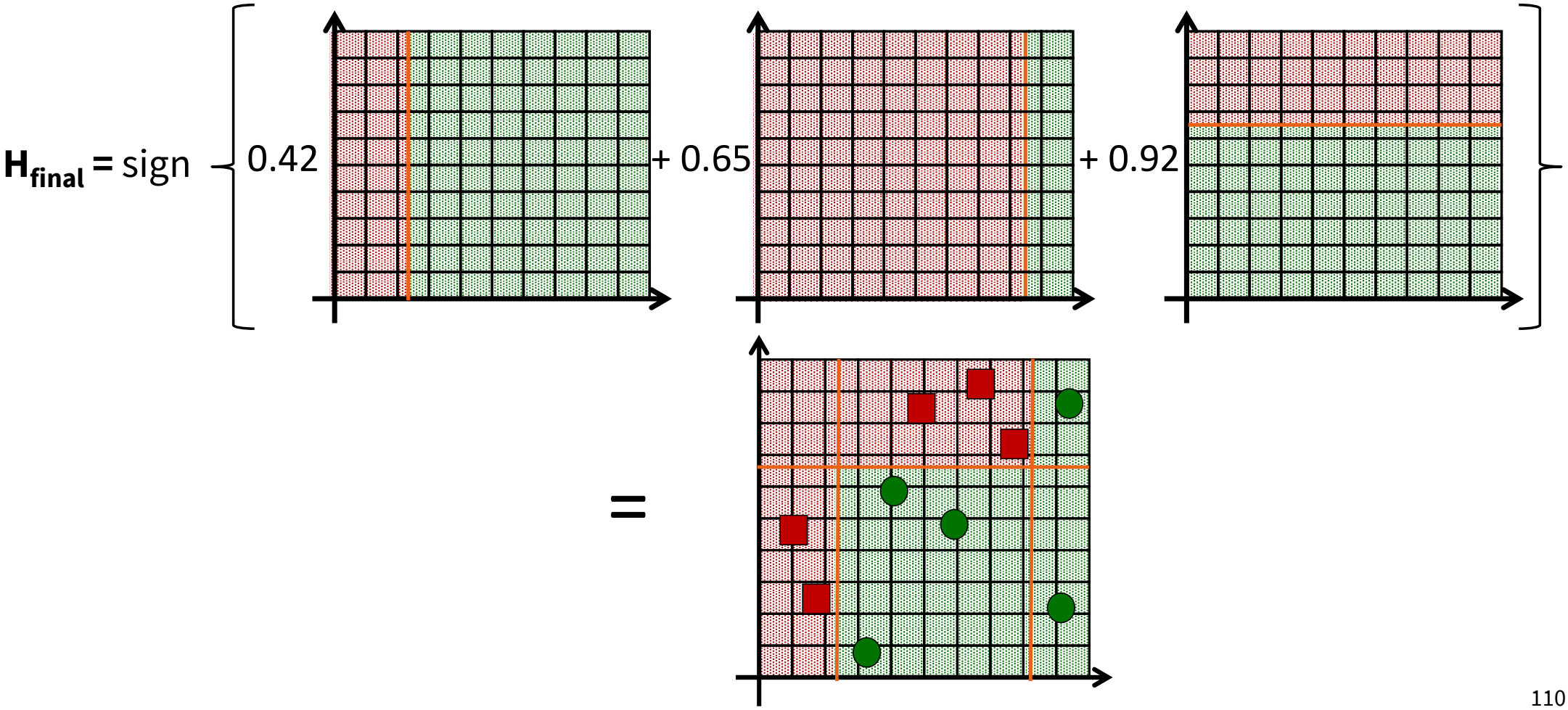
Iteration 3: third weak classifier from reweighted training data

Third round of boosting



Classification With Adaboost – A Visual Intuition in 2 D

Final step: weighted average over base model predictions



The Adaboost Algorithm

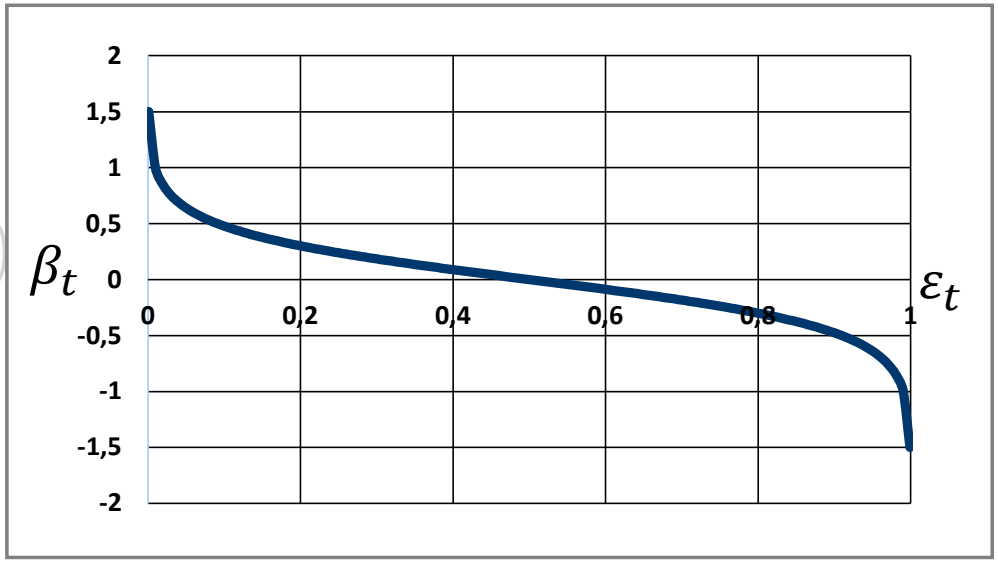
- **Given**
  - Training data $(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \ldots, (\boldsymbol{x}_n, y_n)$ with $\boldsymbol{x} \in \mathbb{R}^m$ and $y \in \{-1, +1\}, i = 1, \ldots n$
  - Base learning algorithm h: $\mathbb{R}^m \rightarrow \{-1, +1\}$
- **Initialize** data point weights $w_i^{(0)} = \frac{1}{\mathrm{n}} \; \forall i$
- **For** $t = 1$ to T
  - Fit classifier $h_t(\boldsymbol{x})$ from weighted sample $\left(x_i, y_i, w_i^{(t-1)}\right)$
  - Calculate a weighted classification error $\epsilon_t = \sum_{i \mid h_t(\boldsymbol{x}_i) \neq y_i} {}^{w_i}\!/_{\sum_{i=1}^n w_i}$
  - Calculate classifier weight $\beta_t = \frac{1}{2} \log\left(\frac{1-\epsilon_t}{\epsilon_t}\right)$
  - Update weights $w_i^{(t)} = \frac{w_i^{(t-1)}}{Z_t} \times \begin{cases} e^{-\beta_t} & \text{if} \quad h_t(\boldsymbol{x}_i) = y_i \\ e^{\beta_t} & \text{if} \quad h_t(\boldsymbol{x}_i) \neq y_i \end{cases}$
    - *where Z_t is a normalization factor such that $\sum_{i=1}^n w_i = 1$*
- **Output** final classifier $H(\boldsymbol{x}) = sign(\sum_{t=1}^T \beta_t h_t(\boldsymbol{x}))$

Use learner that supports data point weights or sample new training set according to weight distribution.

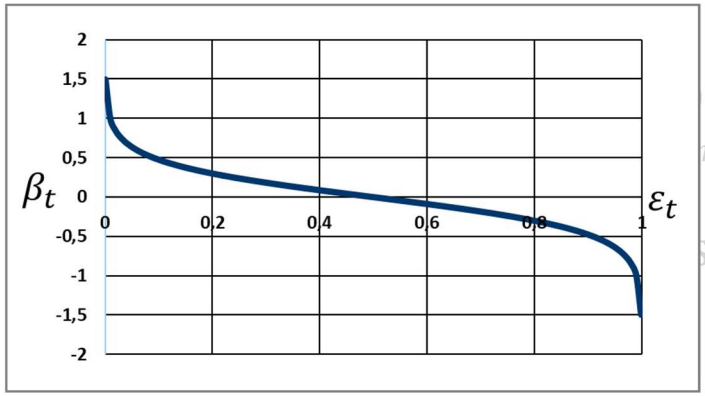
Weights ensure that 'difficult' data points get more attention and affect classifier learning to a larger degree.

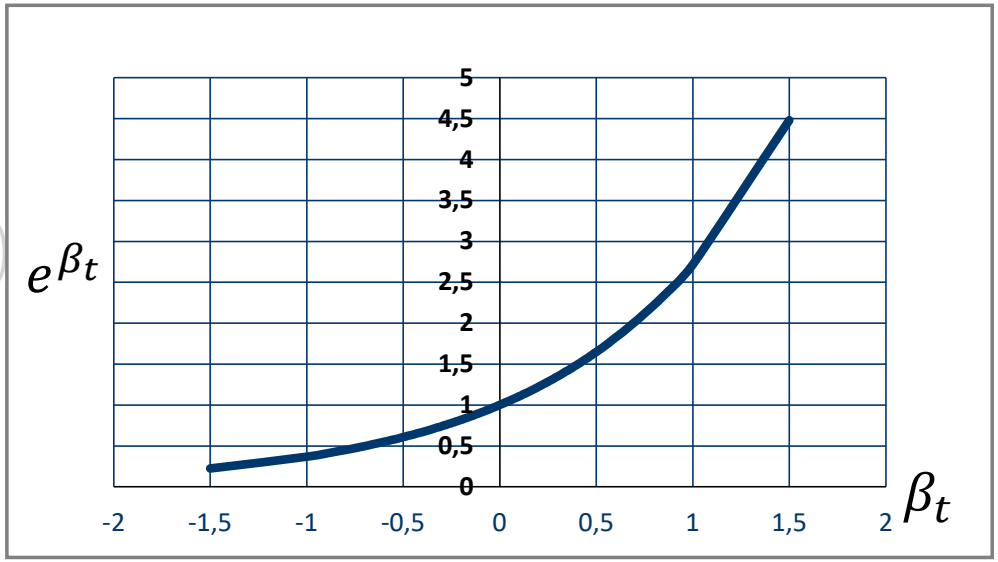
The Adaboost Algorithm in Pseudo Code

- **Given**
  - Training data $(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \ldots, (\boldsymbol{x}_n, y_n)$ with $\boldsymbol{x} \in \mathbb{R}^m$ and $y \in \{-1, +1\}, i = 1, \ldots n$
  - Base learning algorithm $h: \mathbb{R}^m \to \{-1, +1\}$
- **Initialize** data point weights $w_i^{(0)} = \frac{1}{n} \; \forall i$
- **For** $t = 1$ to T
  - Fit classifier $h_t(\boldsymbol{x})$ from weighted sample $\left(x_i, y_i, w_i^{(t-1)}\right)$
  - Calculate classification error $\epsilon_t = \sum_{i \mid h_t(x_i) \neq y_i} {}^{w_i}/_{\sum_{i=1}^n w_i}$
  - Calculate classifier weight $\beta_t = \frac{1}{2} \log\left(\frac{1-\epsilon_t}{\epsilon_t}\right)$
  - Update weights $w_i^{(t)} = \frac{w_i^{(t-1)}}{Z_t} \times \begin{cases} e^{-\beta_t} & \text{if} \quad h_t(\boldsymbol{x}_i) = y_i \\ e^{\beta_t} & \text{if} \quad h_t(\boldsymbol{x}_i) \neq y_i \end{cases}$
    - *where Z_t is a normalization factor such that $\sum_{i=1}^n w_i = 1$*
- **Output** final classifier $H(\boldsymbol{x}) = sign(\sum_{t=1}^T \beta_t h_t(\boldsymbol{x}))$



The Adaboost Algorithm in Pseudo Code





- Given $(x_1, y_1), \ldots, (x_n, y_n)$ with $x \in \mathbb{R}^m$ and $y \in \{-1, +1\}, i = 1, \ldots n$
- Initialize weights $w_i^{(0)} = \frac{1}{n} \; \forall i$
- For $t = 1, \ldots, T$:
  - Fit classifier $h_t(x)$ from weighted sample $\left(x_i, y_i, w_i^{(t-1)}\right)$
  - Calculate classification error $\epsilon_t = \sum_{i \mid h_t(x_i) \neq y_i} {}^{w_i}/_{\sum_{i=1}^{n} w_i}$
  - Calculate classifier weight $\beta_t = \frac{1}{2} \log\left(\frac{1-\epsilon_t}{\epsilon_t}\right)$
  - Update weights $w_i^{(t)} = \frac{w_i^{(t-1)}}{Z_t} \times \begin{cases} e^{-\beta_t} & \text{if} \quad h_t(\boldsymbol{x}_i) = y_i \\ e^{\beta_t} & \text{if} \quad h_t(\boldsymbol{x}_i) \neq y_i \end{cases}$
    - *where Z_t is a normalization factor such that $\sum_{i=1}^{n} w_i = 1$*
- **Output** final classifier $H(\boldsymbol{x}) = sign(\sum_{t=1}^{T} \beta_t h_t(\boldsymbol{x}))$