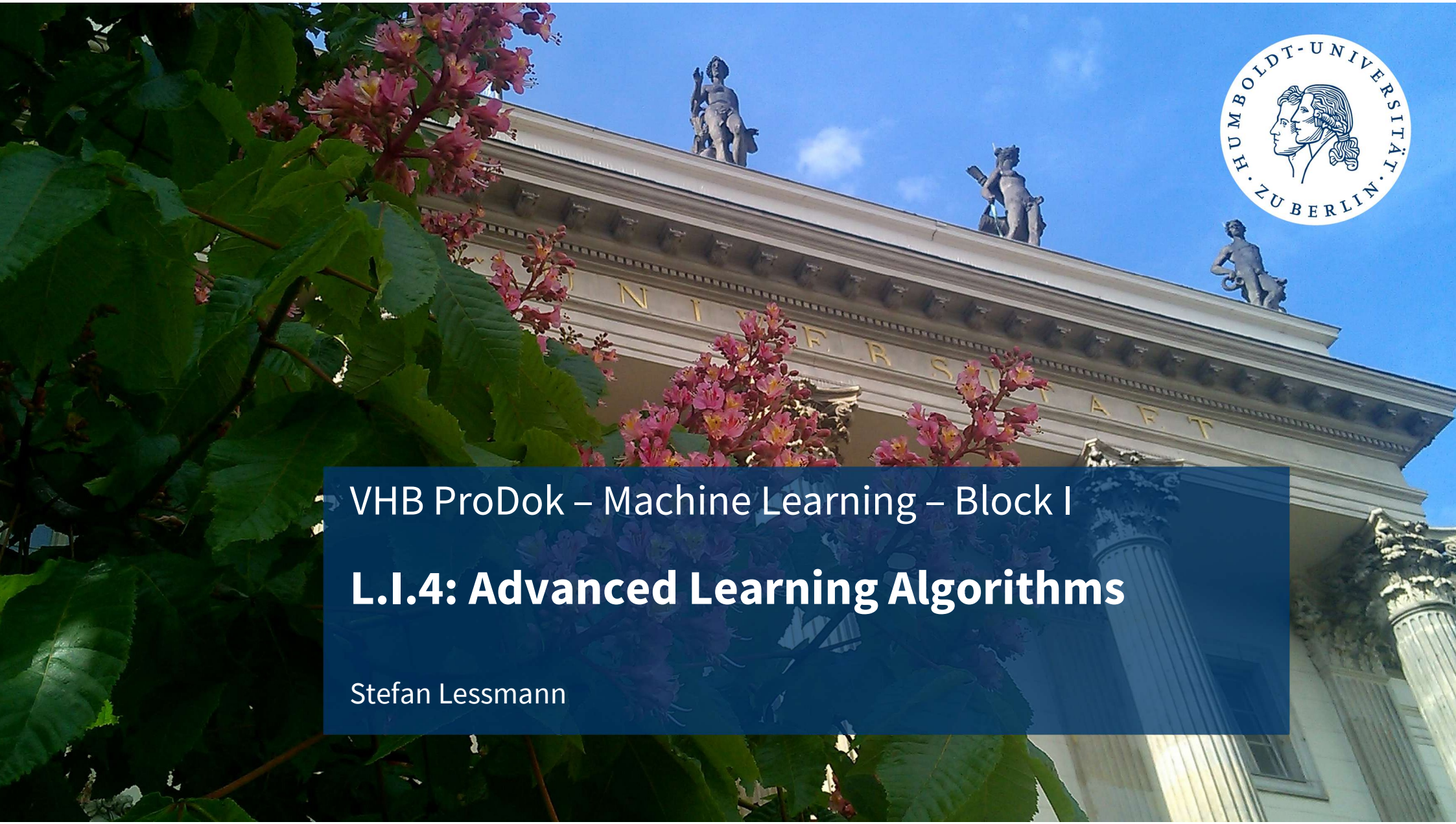
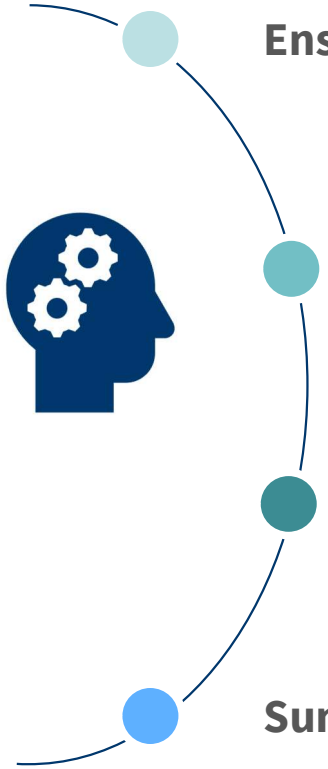


VHB ProDok – Machine Learning – Block I

L.I.4: Advanced Learning Algorithms

Stefan Lessmann

Agenda



Ensemble Learning Fundamentals

A Tree-Learning Primer
Tree growing algorithm, split criteria, and tree pruning

Implementing Ensemble Learning
Random Forest and Gradient Boosting

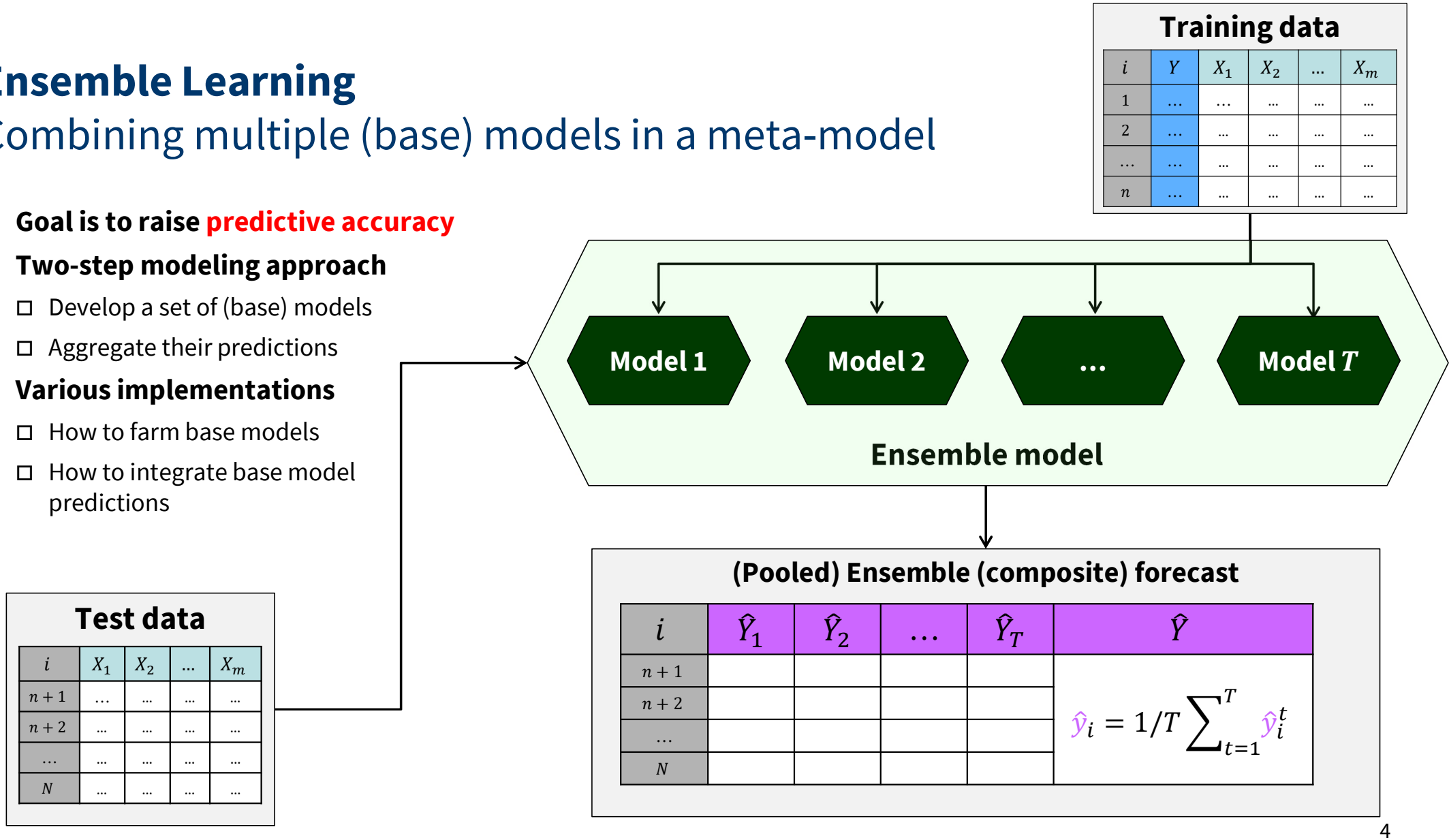
Summary

Ensemble Learning Fundamentals

Ensemble Learning

Combining multiple (base) models in a meta-model

- **Goal is to raise predictive accuracy**
- **Two-step modeling approach**
  - Develop a set of (base) models
  - Aggregate their predictions
- **Various implementations**
  - How to farm base models
  - How to integrate base model predictions



Why Ensemble Learning Raises Predictive Accuracy

- **A vast amount of empirical evidence shows that ensemble learning (i.e., combining forecasts of multiple base models) raises predictive accuracy**
- **What factors explain this success of ensemble learning?**

Bias-Variance Trade-Off and Ensemble Learning

Ensemble learning can reduce both, bias and variance

- **Many high-bias models combined can learn complex patterns**
- **Averaging over forecasts reduces variance**
- **Popular ensemble learning algorithms**
  - Bagging-type algorithms target variance reduction
  - Boosting-type algorithms primarily target bias reduction

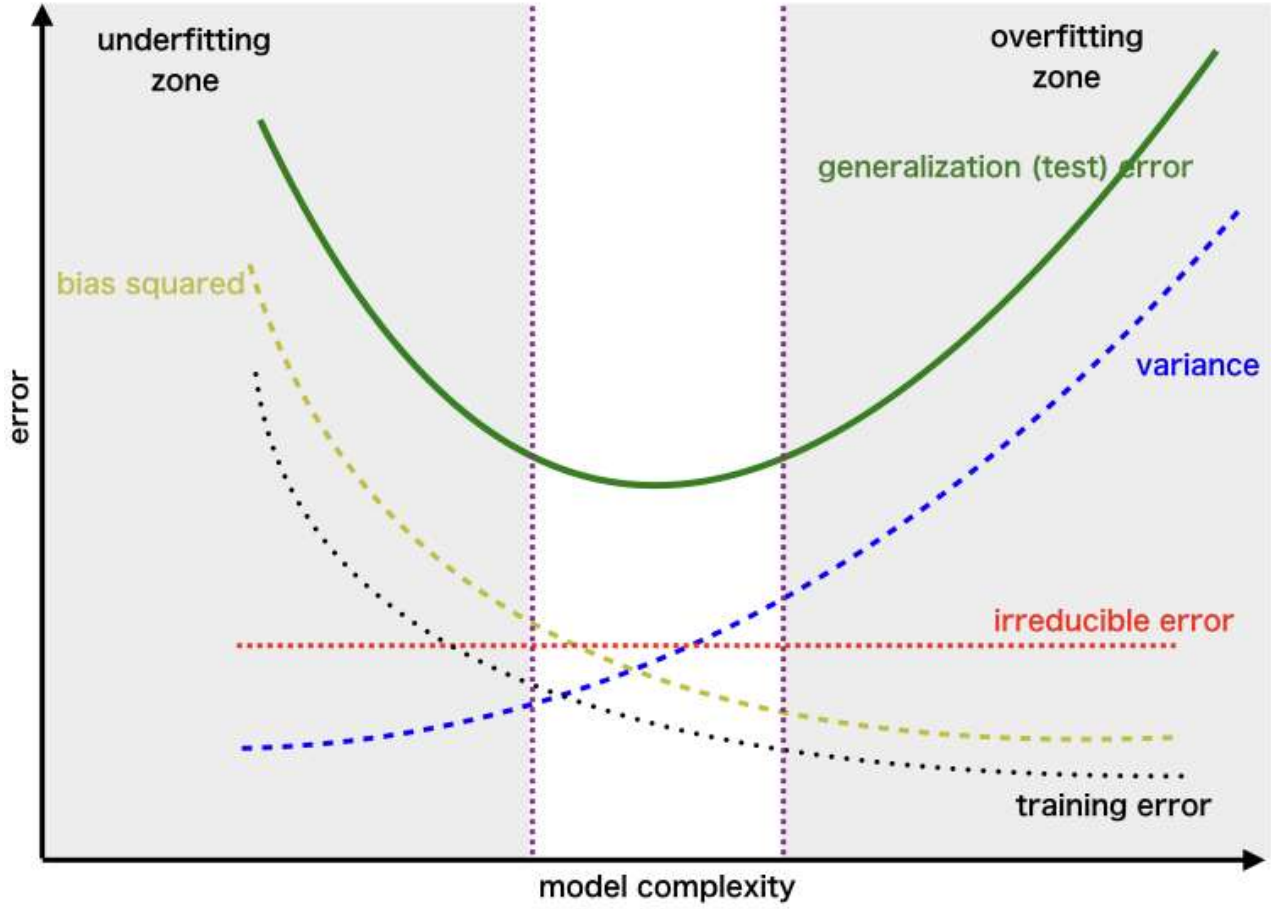
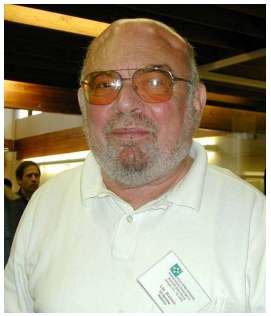


Image source: G. Papachristoudis (2019)
https://towardsdatascience.com/the-bias-variance-tradeoff-8818f41e39e9

Implementing Ensemble Learning

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

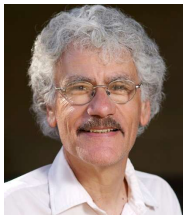
- **Ensemble learning is predominantly executed using trees**
- **Bagging approach**
  - Relies on bootstrap sampling: derive base models from different training sets
  - Independence among base models facilitates parallel training
  - Simple average for base model forecast combination
  - Most popular/powerful representative: Random Forest
- **Boosting approach**
  - 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

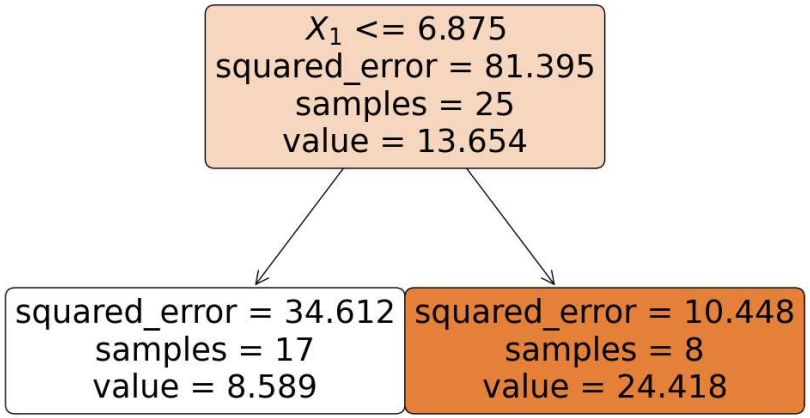


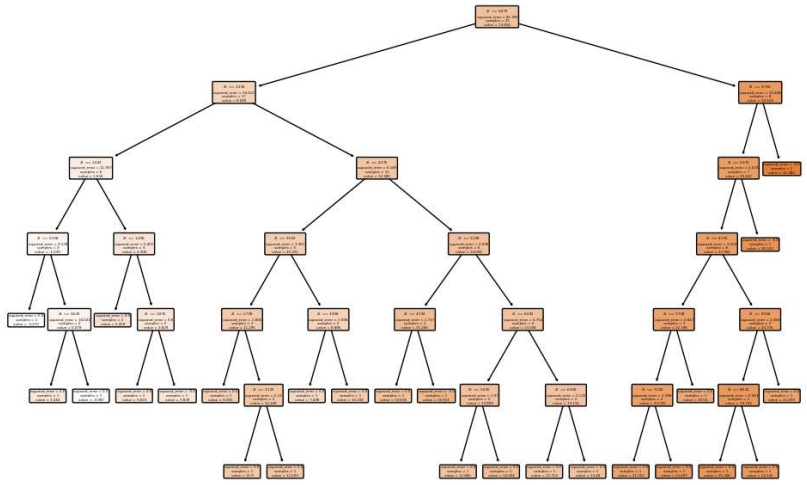
J. Friedman

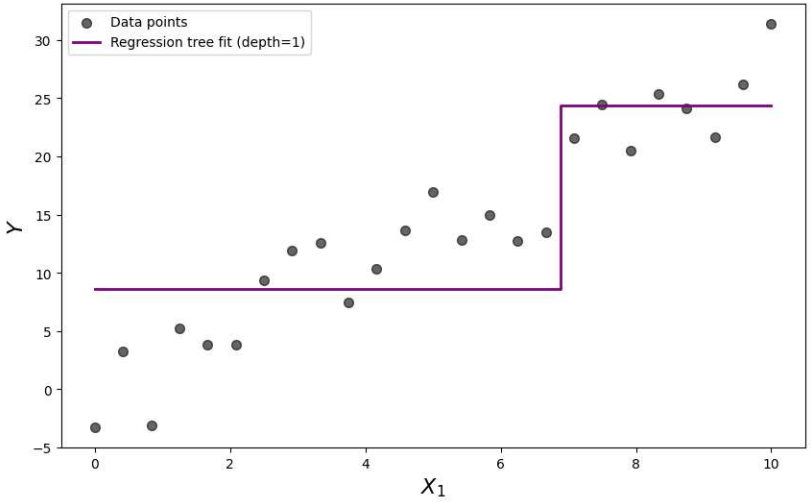
A Tree-Learning Primer

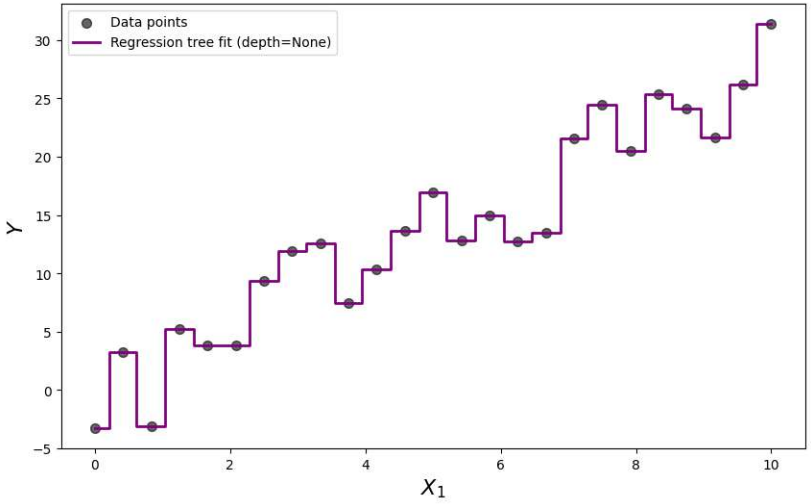
Tree growing algorithm, split criteria, and tree pruning

Recap: Simple and Complex Trees









Regression Example: Leasing Business

Resale price forecasting use case

- **Lessor's pricing problem: Decide on the leasing rate**
  - Leasing rate must cover all relevant costs including depreciation
  - Item's residual value is unknown when signing the contract

- **We can use regression to *model* residual value/resale price**
  - Attribute values are observed before signing the contract
  - Resale prices are unobservable until after the contract expires and the item is resold

i	PRODUCT	LIST PRICE [$]	AGE [month]	CLIENT INDUSTRY	…	**RESALE PRICE [$]**
1	Dell XPS 15'	2,500	36	Mining		347
2	Dell XPS 15'	2,500	24	Health		416
3	Dell XPS 17'	3,000	36	Manufacturing		538
4	HP Envy 17'	1,300	24	Office	…	121
5	HP EliteBook 850	1,900	36	Manufacturing		172
…	…					…
n	Lenovo Yoga 11'	799	12	Office	…	88

Regression Trees in a Nutshell

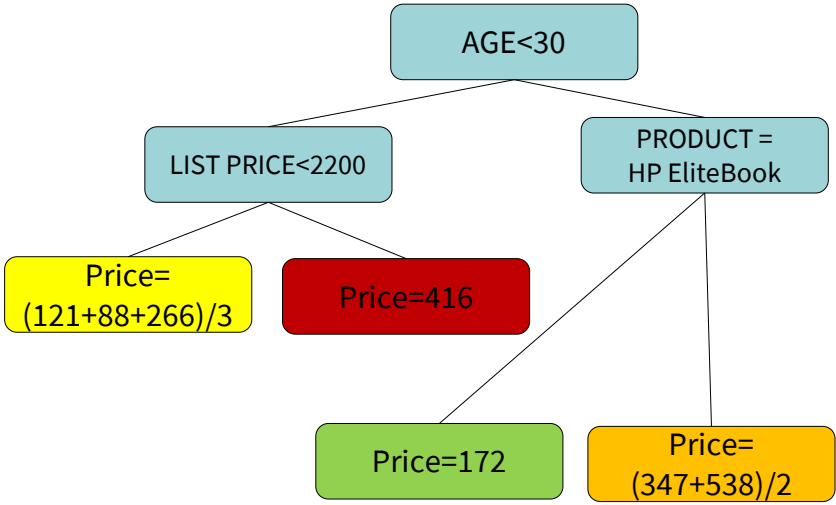
Training

Find the structure of the tree using the training data.

- Split on which feature?
- Split on which threshold?
- When stop splitting?

Training data incl. Y

PRODUCT	LIST PRICE	AGE	…	RESALE PRICE
Dell XPS 15’	2,500	36	…	347
Dell XPS 15’	2,500	24	…	416
Dell XPS 17’	3,000	36	…	538
HP Envy 17’	1,300	24	…	121
HP EliteBook	1,900	36	…	172
Lenovo Yoga 11’	799	12	…	88
Lenovo Yoga 13’	1,100	12	…	266
…	…	…	…	…



Testing

Identify the right leaf node for a new example.

- Put example down the tree
- Apply decision rules to feature values
- Forecasts equals leaf node prediction

Novel data w/o Y

PRODUCT	LIST PRICE	AGE	…	TREE FORECAST
HP Envy 15’	1,150	12	…	158,33
Lenovo Yoga 13’	1,100	36	…	442,5
Dell XPS 15’	2,500	12	…	416
HP EliteBook	2,100	48	…	172
Lenovo Yoga 14’	2,300	24	…	416
HP EliteBook	1,900	24	…	158,33
…	…	…	…	…

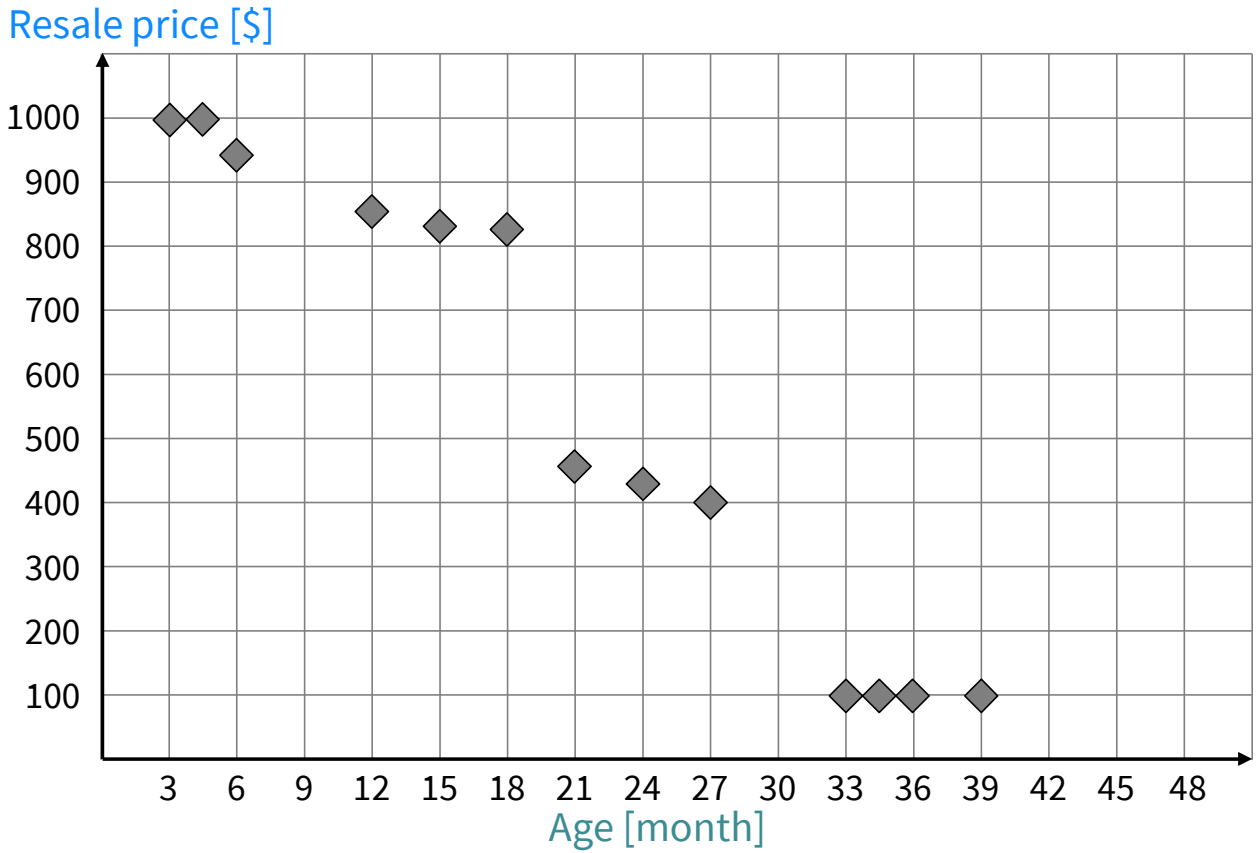
Key Steps in Regression Tree Learning

- **Deciding how to grow a tree, that is on split points**
  - Recursive partitioning approach
    - Find an optimal split partitioning the entire training data into two subgroups (child nodes)
    - For each child node, find an optimal split for the data in that child node
    - Repeat the partitioning of the data until some stopping criterion is met
  - Assessing candidate splits using a loss function
    - Same concept as in linear regression
    - Minimize the sum of squared residuals
- **Deciding when to stop tree growing**
  - Complex trees with many leaves will not forecast accurately (overfitting problem, see later)
  - Tree pruning seeks a balance between fitting the training data well while not letting the tree become too complex

Regression Tree Learning: Finding an Optimal Split

Fully-enumerative search through all possible split points per feature

- **Tree prediction**
  - A tree node contains part of the training set
  - Average target over those data points to obtain prediction
  - Prediction is constant for that node
- **Node splitting**
  - Binary splits
    - Numeric features: $X \leq \text{threshold}$
    - Categorical features: $X = \text{x}$
  - Child nodes must not be empty
    - At most *N*-1 ways to split on a feature
- **Loss function to assess splits**

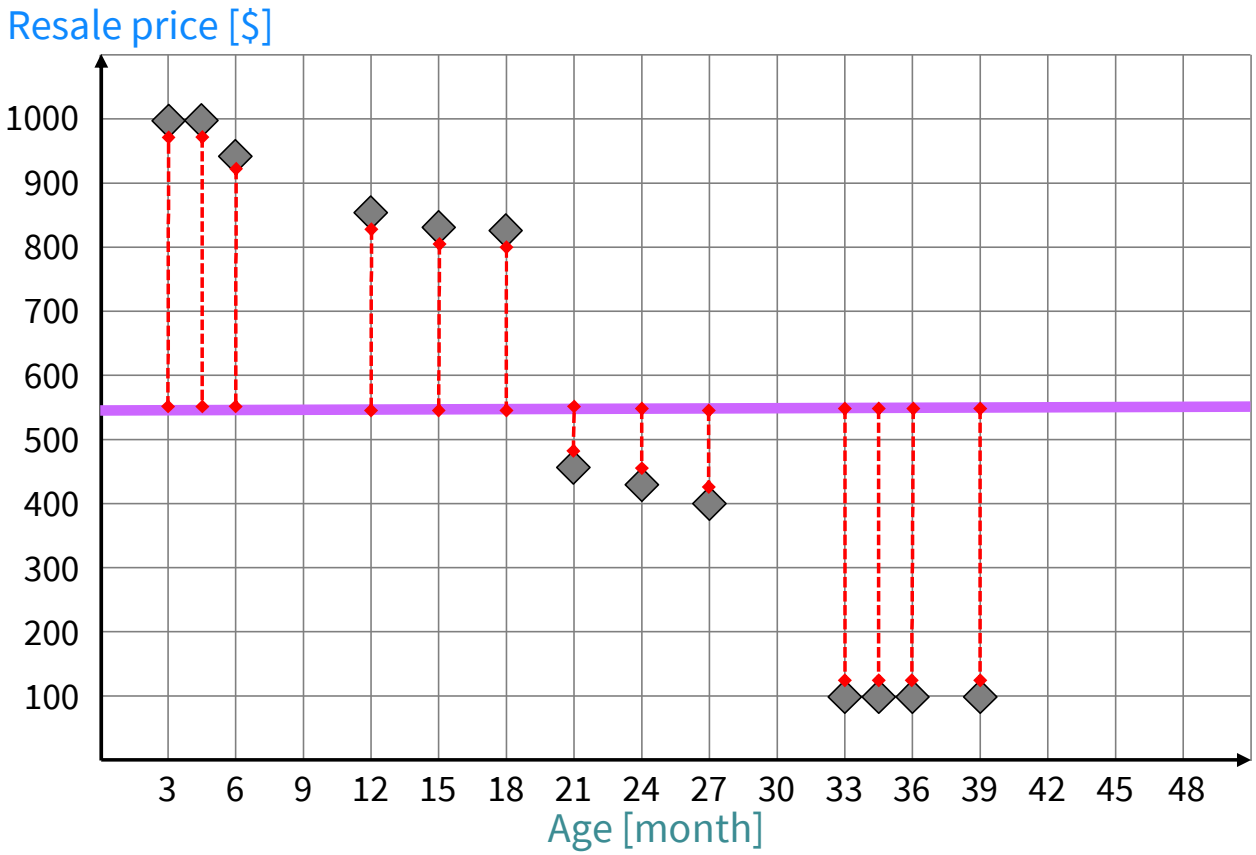


Finding an Optimal Split

Price=550

$$SSR = (1000 - 550)^2 + (1000 - 550)^2 + (950 - 550)^2 + (850 - 550)^2 + \ldots + (100 - 550)^2 + (100 - 550)^2 = 1{,}694{,}375$$

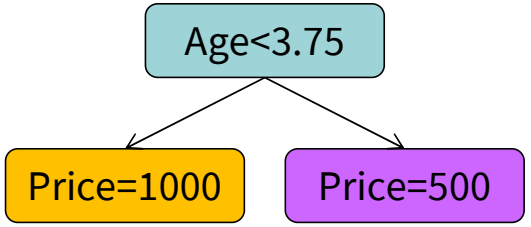
- **Start with the average target**
  - Equivalent to tree root, aka a 'tree' w/o any split
  - Amounts to 550 for the demo data
- **Standard loss for regression: Sum of squared residuals (SSR)**
  - $e_i = (y_i - \hat{y}_i)$
  - $J^{SSR} = \sum_{i=1}^{n} (e_i)^2$



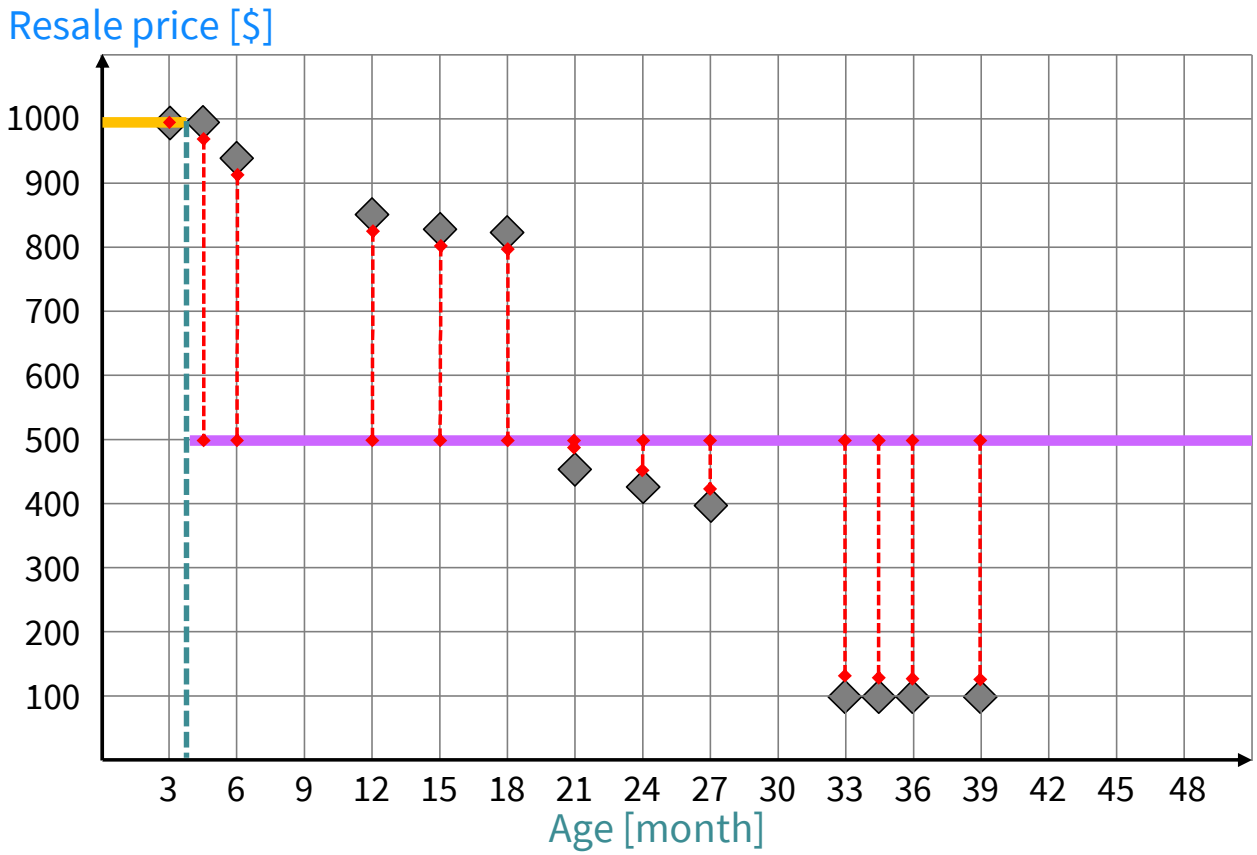
Finding an Optimal Split

Enumerative search strategy

$$SSR = (1000 - 1000)^2 + (1000 - 500)^2 + \ldots + (100 - 500)^2 = 1{,}443{,}073$$



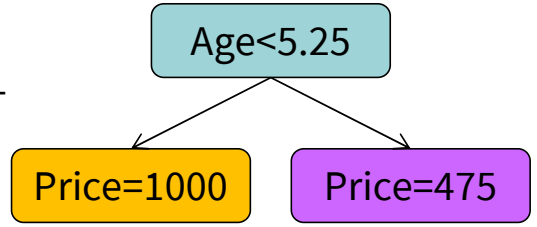
- **Begin with the two data points with smallest age**
- **Assess split that cuts the data between these points**
  - Age values of the two example are, respectively, 3 and 4.5
  - So consider split Age < 3.75
- **Determine tree forecasts for the two leaves**
- **Compute SSR of this split**



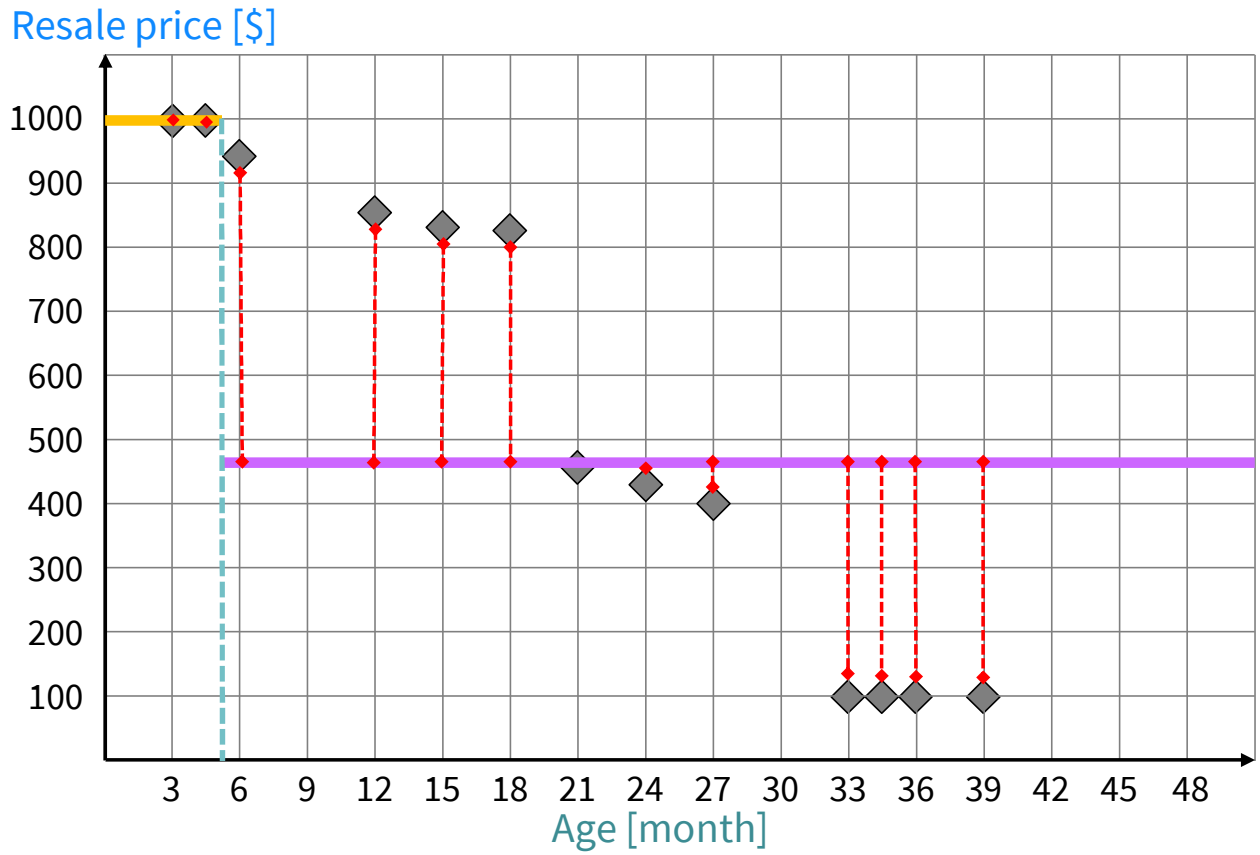
Finding an Optimal Split

Enumerative search strategy

$$SSR = (1000 - 1000)^2 + (1000 - 1000)^2 + (950 - 475)^2 + \cdots + (100 - 475)^2 = 1{,}443{,}073$$



- **Proceed with the next pair of neighboring data points**
- **Assess split that cuts the data between these points**
  - Age values of the two example are, respectively, 4.5 and 6
  - So consider split Age < 5.25
- **Determine tree forecasts for the two leaves**
- **Compute SSR of this split**

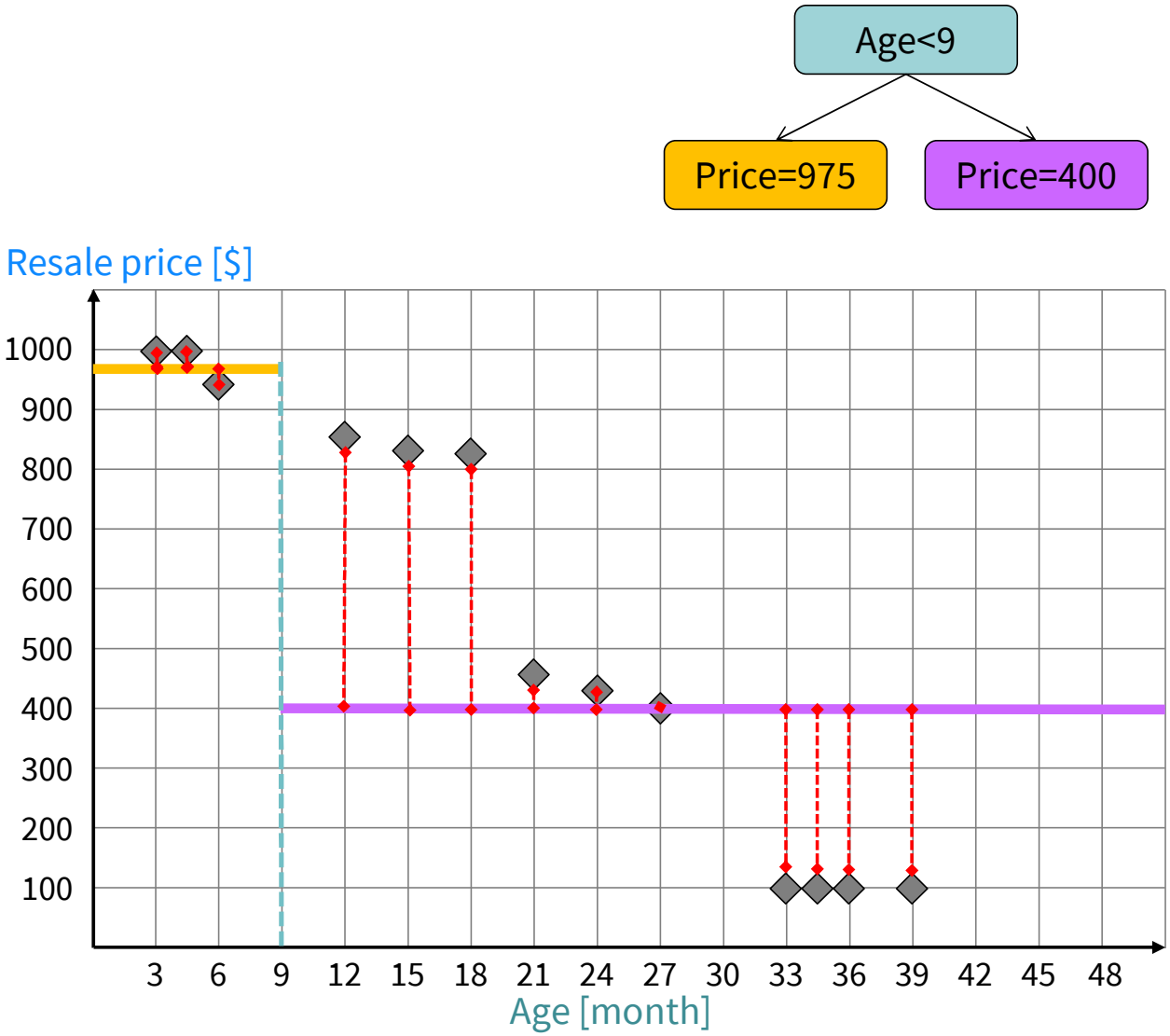


Finding an Optimal Split

Enumerative search strategy

- **Continue in the same way until all candidate splits have been explored**
- **Keep track of the SSR**

$$SSR = (1000 - 975)^2 + \cdots + (950 - 975)^2 + (850 - 400)^2 + \cdots + (100 - 400)^2 = 925{,}479$$

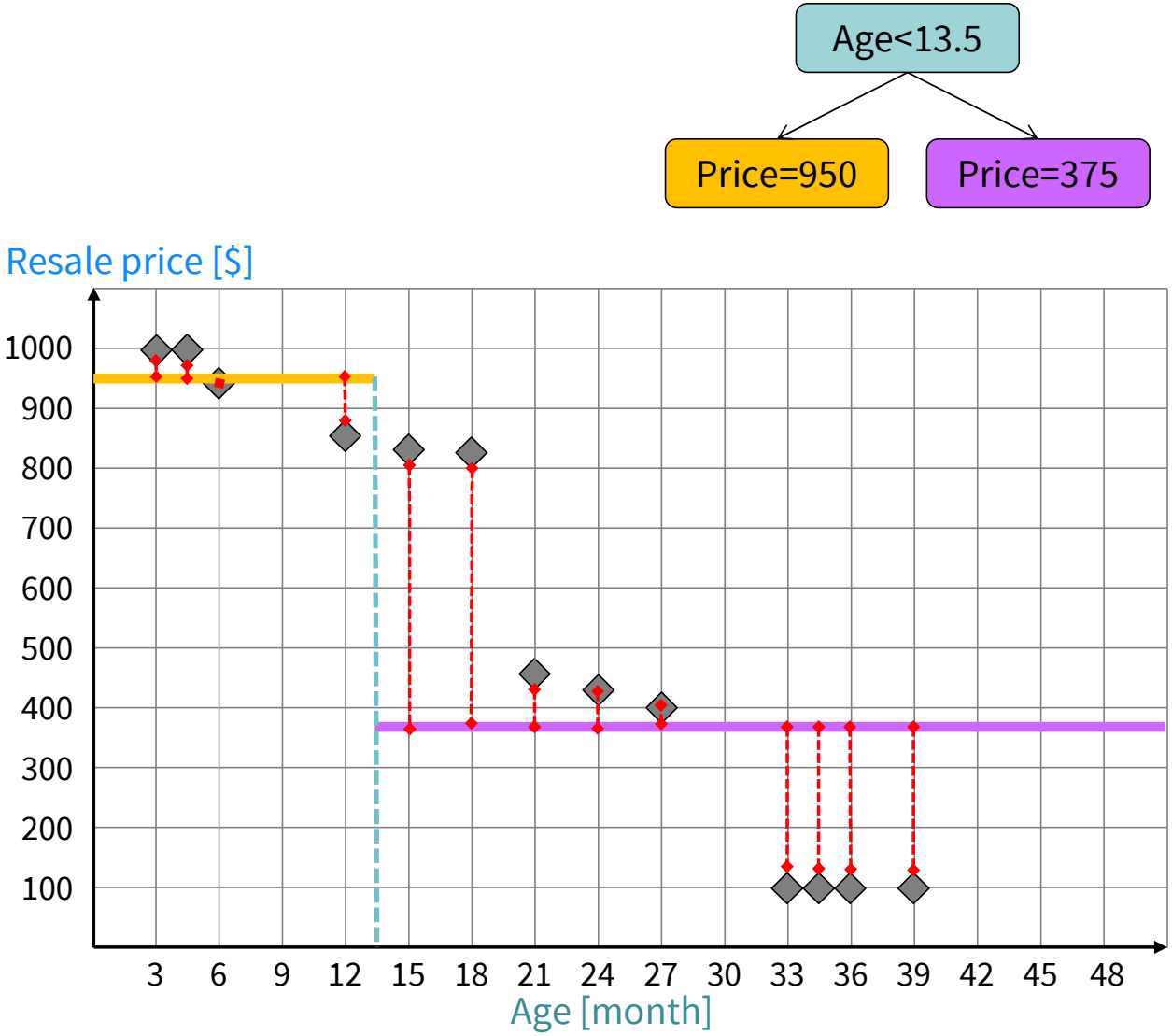


Finding an Optimal Split

Enumerative search strategy

- **Continue in the same way until all candidate splits have been explored**
- **Keep track of the SSR**

$$SSR = (1000 - 950)^2 + \cdots + (850 - 950)^2 + (825 - 375)^2 + \cdots + (100 - 375)^2 = 730{,}972$$

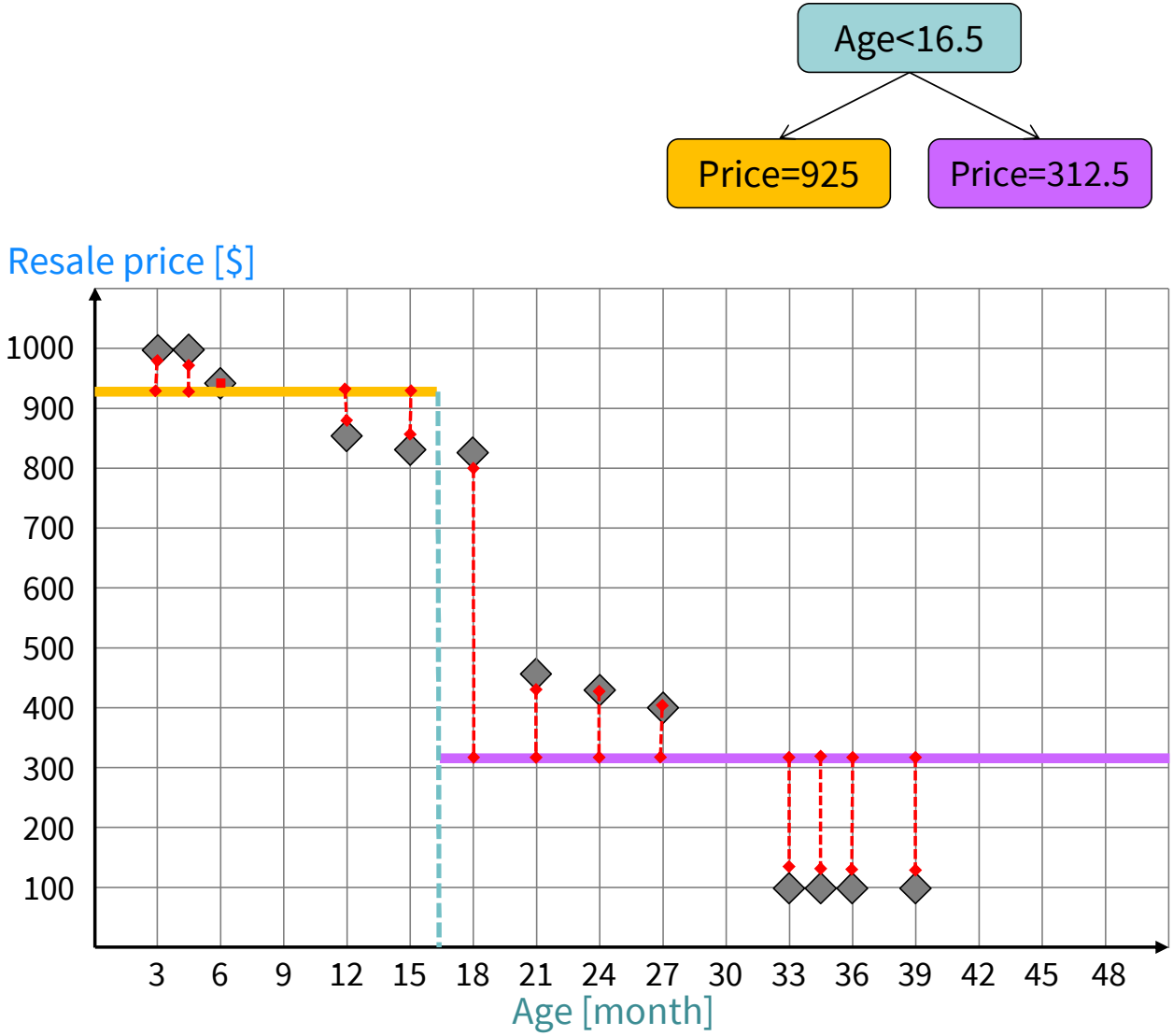


Finding an Optimal Split

Enumerative search strategy

- **Continue in the same way until all candidate splits have been explored**
- **Keep track of the SSR**

$$SSR = (1000 - 925)^2 + \ldots + (825 - 925)^2 + (825 - 312.5)^2 + \cdots + (100 - 312.5)^2 = 510{,}000$$

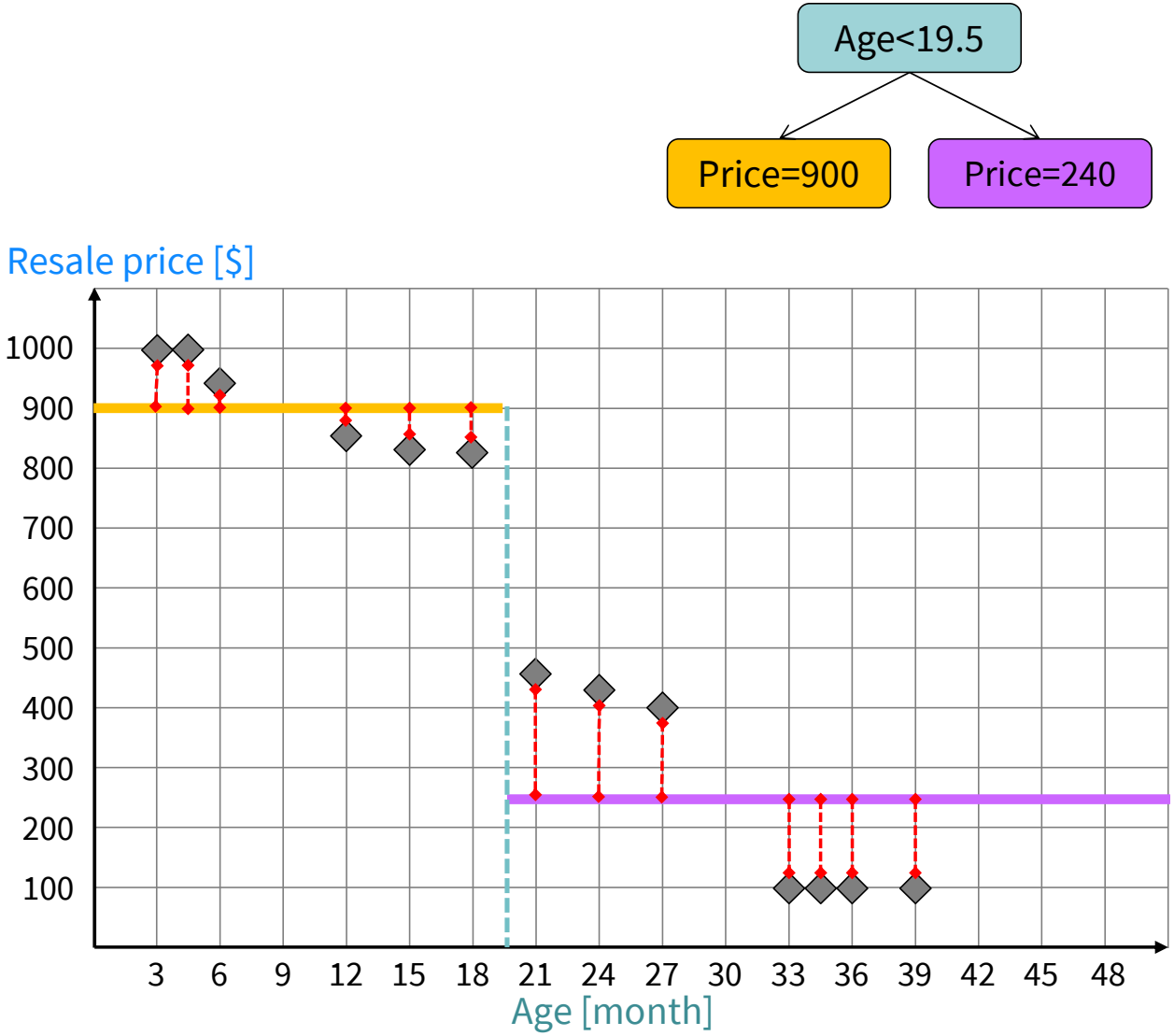


Finding an Optimal Split

Enumerative search strategy

- **Continue in the same way until all candidate splits have been explored**
- **Keep track of the SSR**

$$SSR = (1000 - 900)^2 + \ldots + (825 - 900)^2 + (450 - 240)^2 + \cdots + (100 - 240)^2 = 218{,}155$$

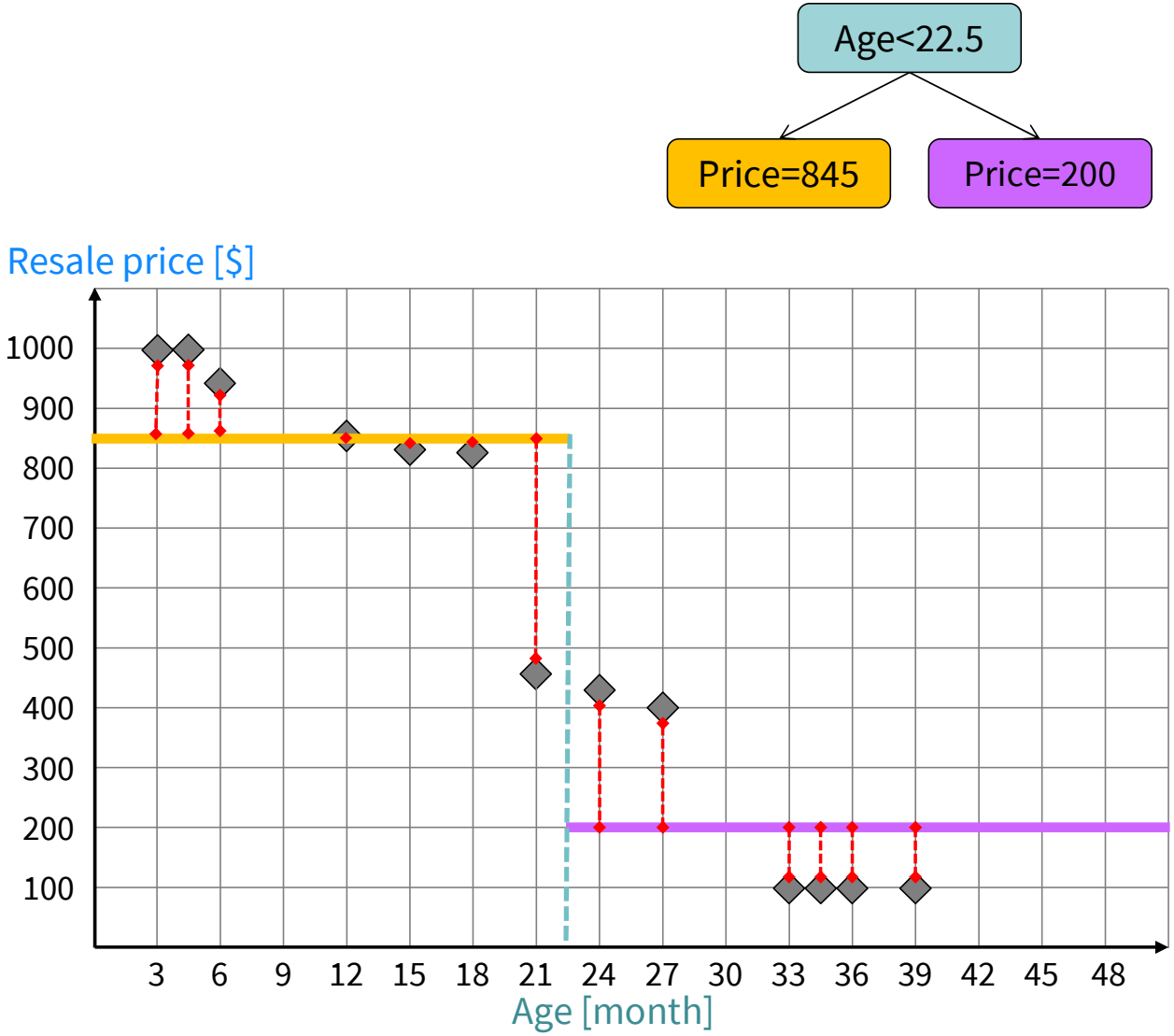


Finding an Optimal Split

Enumerative search strategy

- **Continue in the same way until all candidate splits have been explored**
- **Keep track of the SSR**

$$SSR = (1000 - 845)^2 + \ldots + (450 - 845)^2 + (425 - 200)^2 + \cdots + (100 - 200)^2 = 346{,}414$$

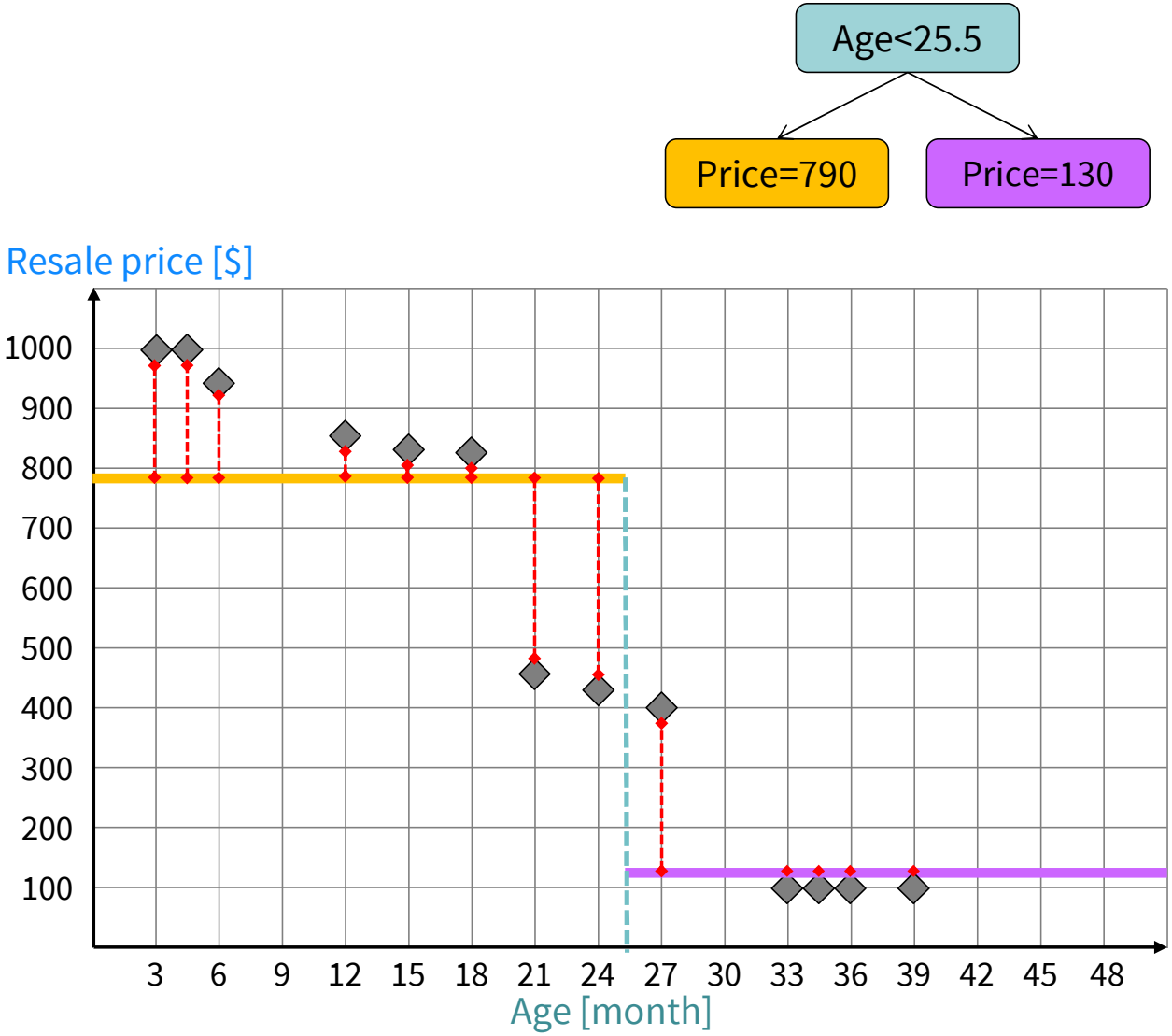


Finding an Optimal Split

Enumerative search strategy

- **Continue in the same way until all candidate splits have been explored**
- **Keep track of the SSR**

$$SSR = (1000 - 790)^2 + \ldots + (425 - 790)^2 + (400 - 130)^2 + \cdots + (100 - 130)^2 = 440{,}672$$

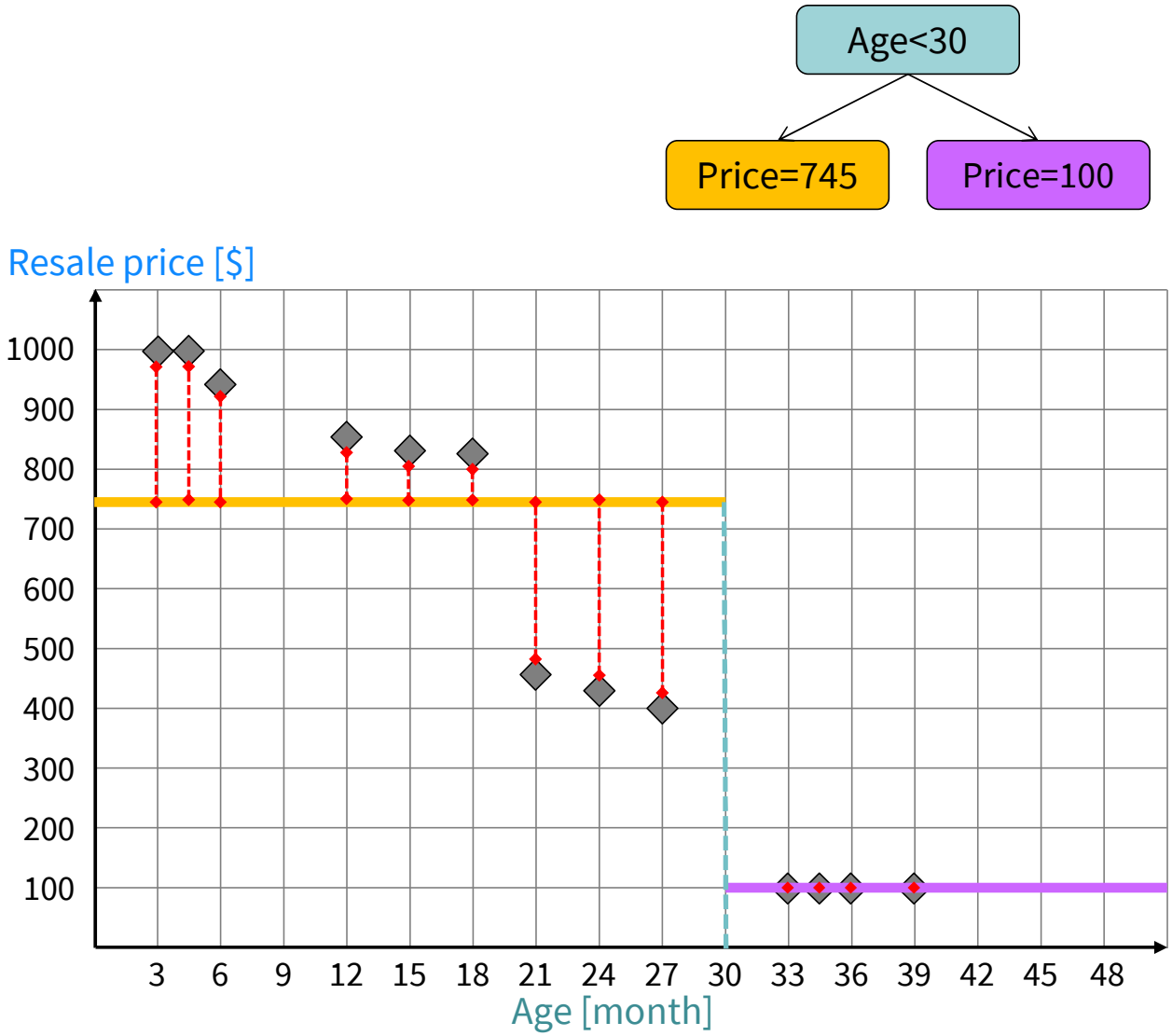


Finding an Optimal Split

Enumerative search strategy

- **Continue in the same way until all candidate splits have been explored**
- **Keep track of the SSR**

$$SSR = (1000 - 745)^2 + \ldots + (400 - 745)^2 + (100 - 100)^2 + \cdots + (100 - 100)^2 = 504{,}306$$

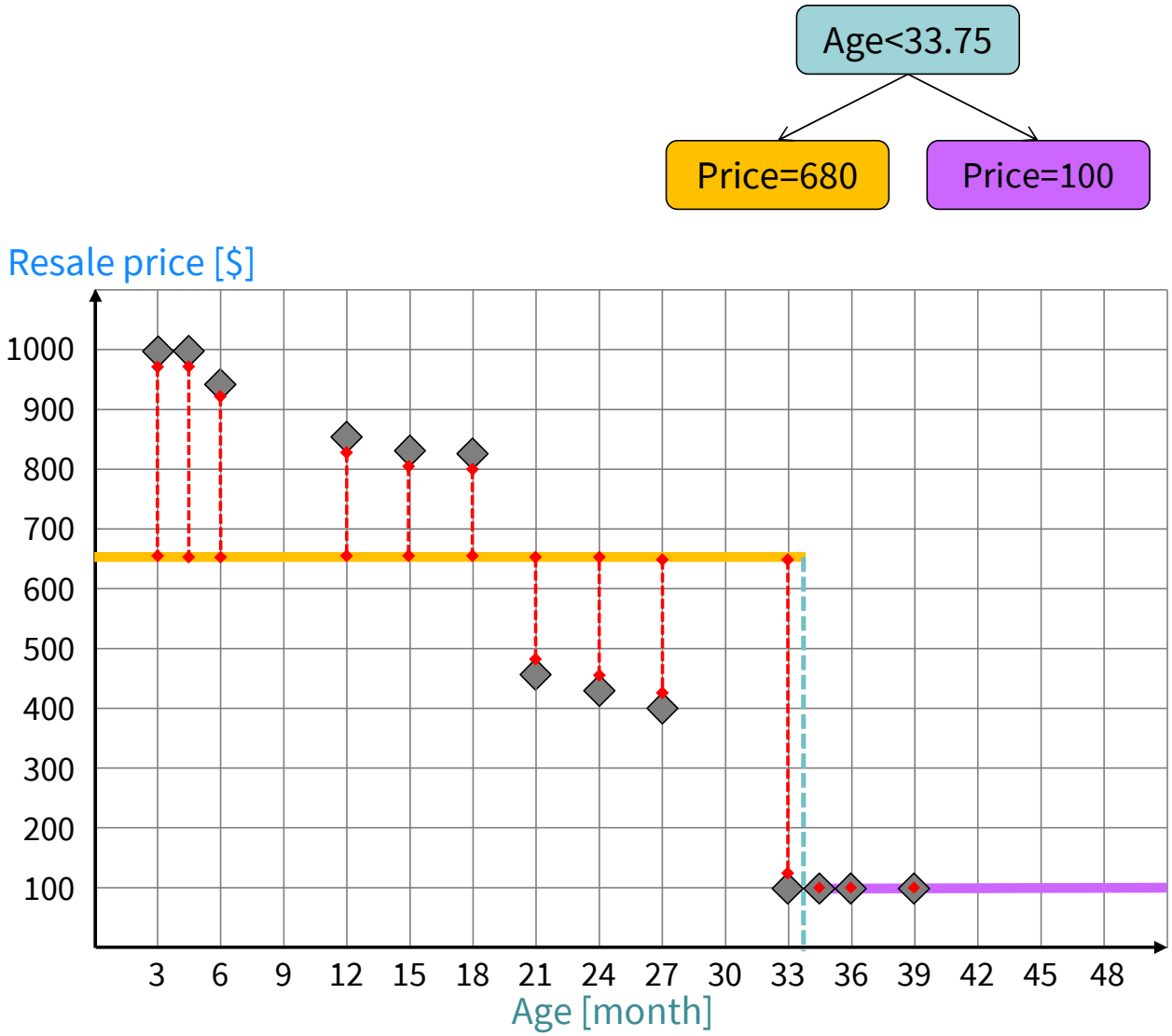


Finding an Optimal Split

Enumerative search strategy

- **Continue in the same way until all candidate splits have been explored**
- **Keep track of the SSR**

$$SSR = (1000 - 680)^2 + \ldots + (100 - 680)^2 + (100 - 100)^2 + \cdots + (100 - 100)^2 = 881{,}313$$

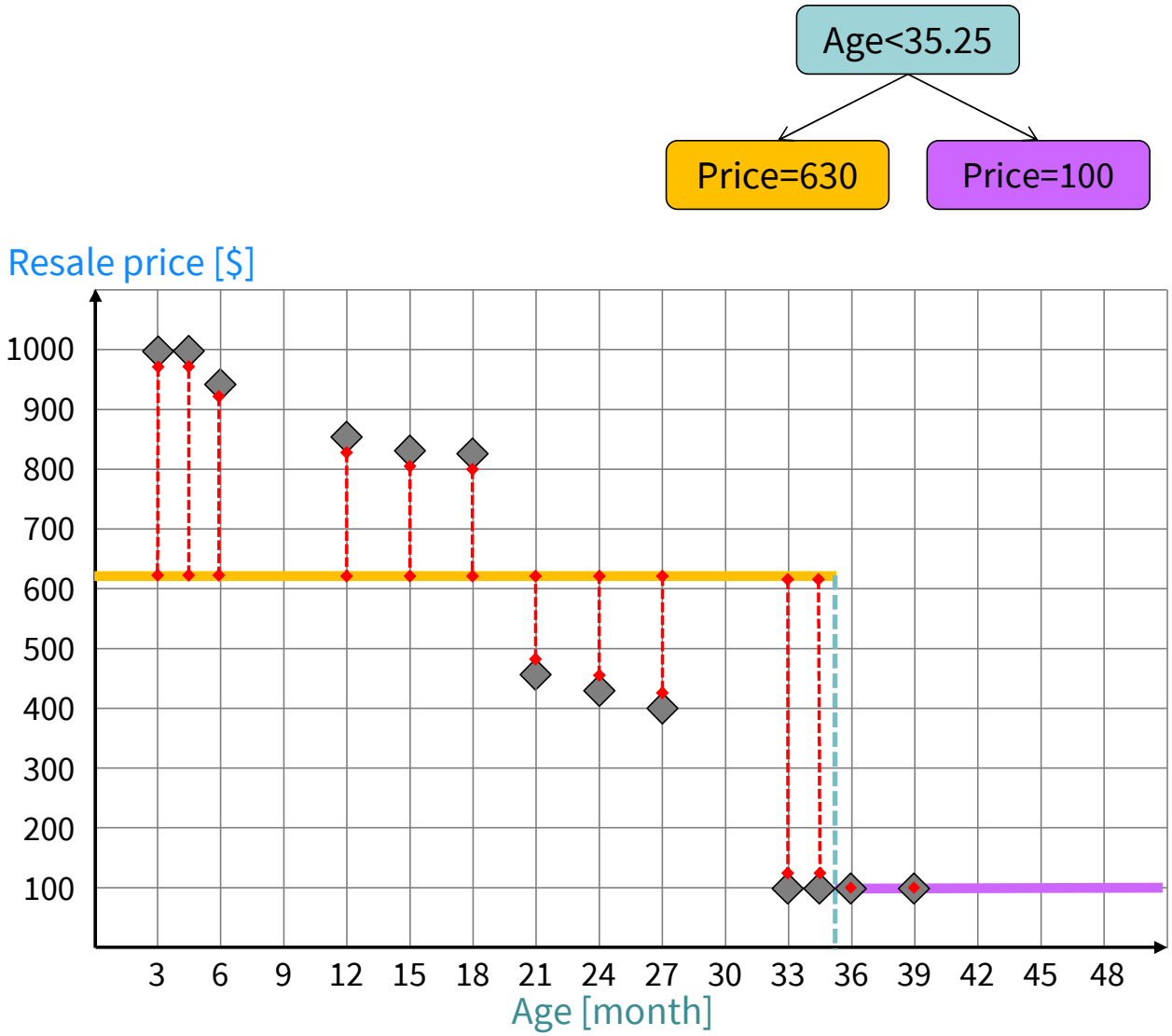


Finding an Optimal Split

Enumerative search strategy

- **Continue in the same way until all candidate splits have been explored**
- **Keep track of the SSR**

$$SSR = (1000 - 630)^2 + \ldots + (100 - 630)^2 + (100 - 100)^2 + (100 - 100)^2 = 1{,}189{,}773$$

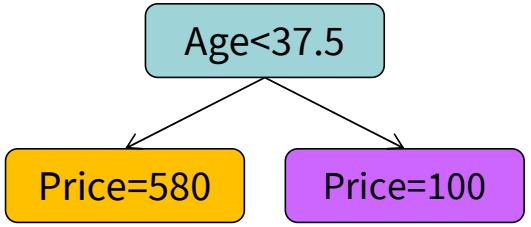


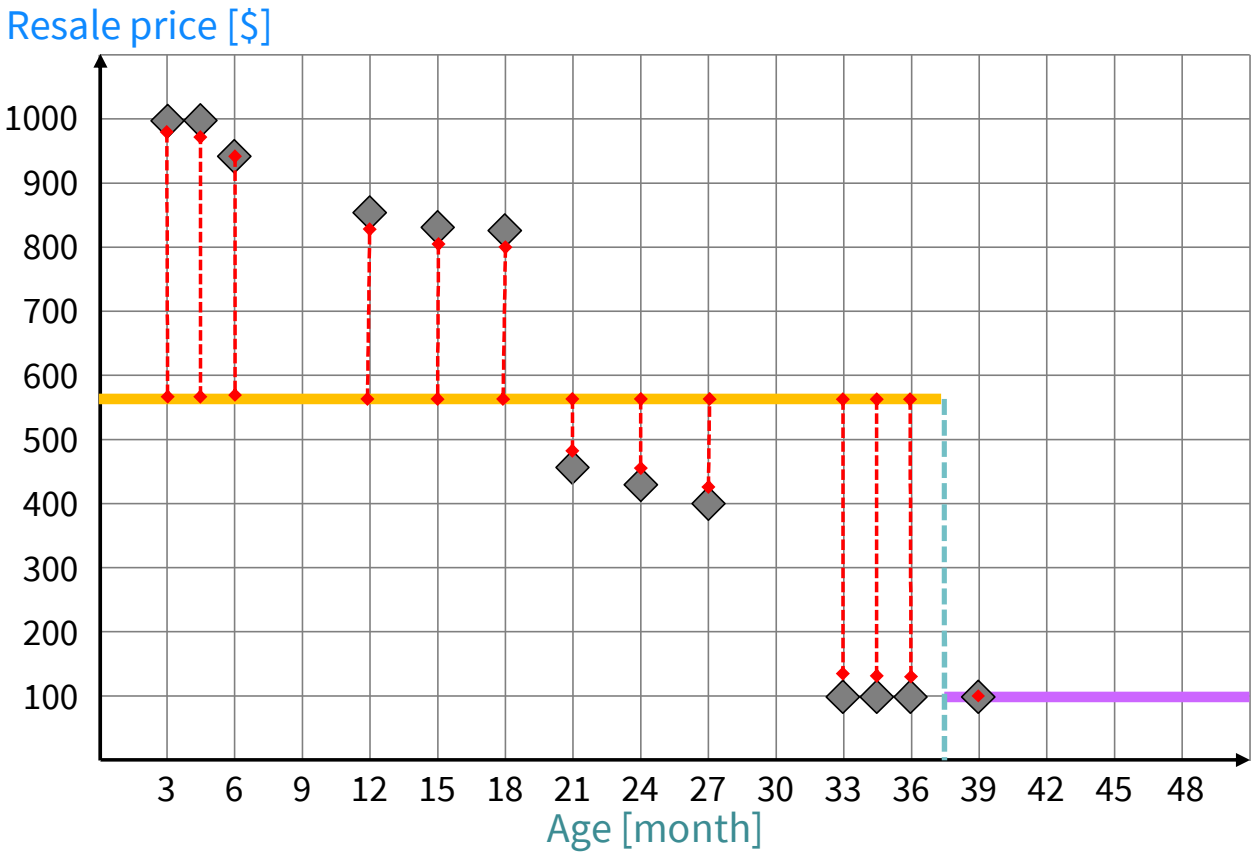
Finding an Optimal Split

Enumerative search strategy

- **Continue in the same way until all candidate splits have been explored**
- **Keep track of the SSR**

$$SSR = (1000 - 580)^2 + \ldots + (100 - 580)^2 + (100 - 100)^2 = 1{,}446{,}823$$



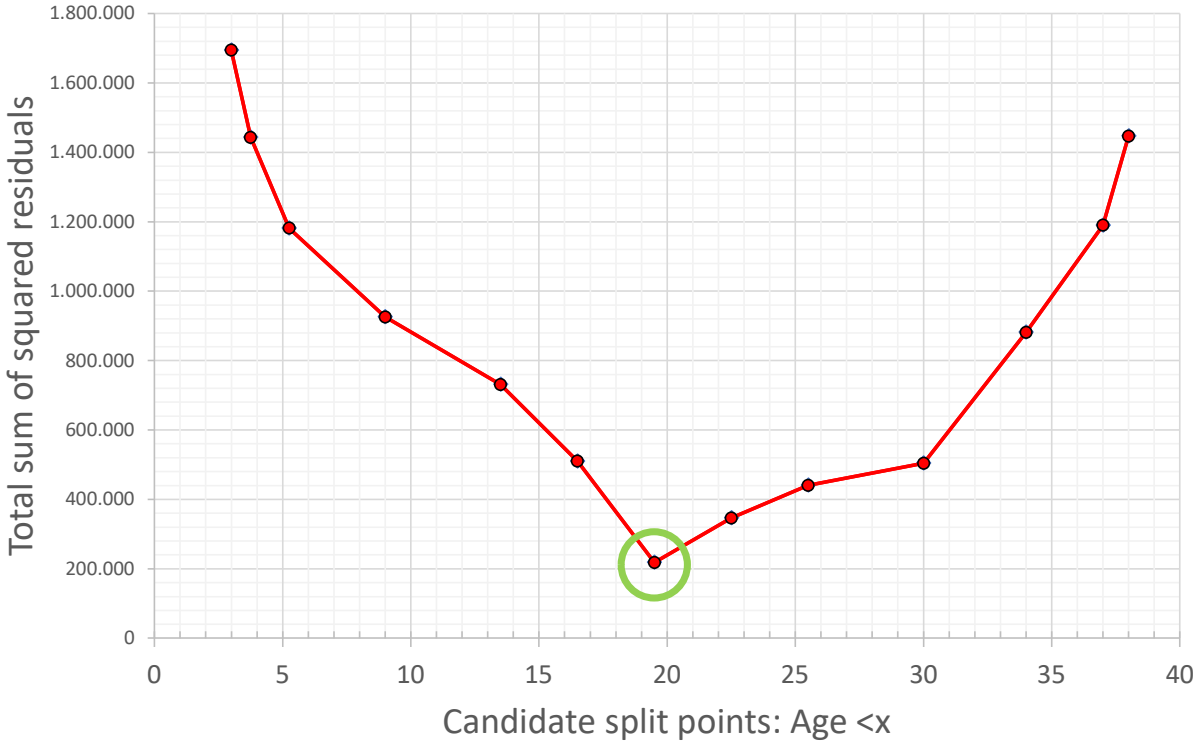


Finding an Optimal Split

Select the overall best split among all candidate splits

- **We obtain a list of candidate split points on feature age**
- **And the corresponding SSR**

Candidate split Age<x	SSR
3	1.694.375
3,75	1.443.073
5,25	1.181.591
9	925.479
13,5	730.972
16,5	510.000
19,5	218.155
22,5	346.414
25,5	440.672
30	504.306
34	881.313
37	1.189.773
38	1.446.823



Regression Tree Learning Continued

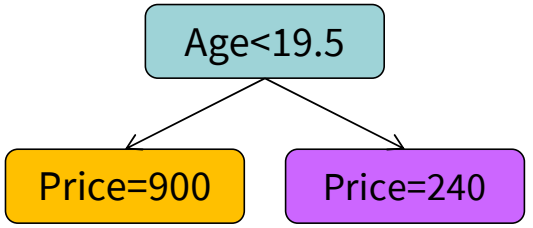
Having found our first split, we search for the next best split in each child

Age [months]	Resale price [$]
3	1000
4,5	1000
6	950
12	850
15	825
18	825
21	450
24	425
27	400
33	100
34,5	100
36	100
39	100

Find best split for these examples

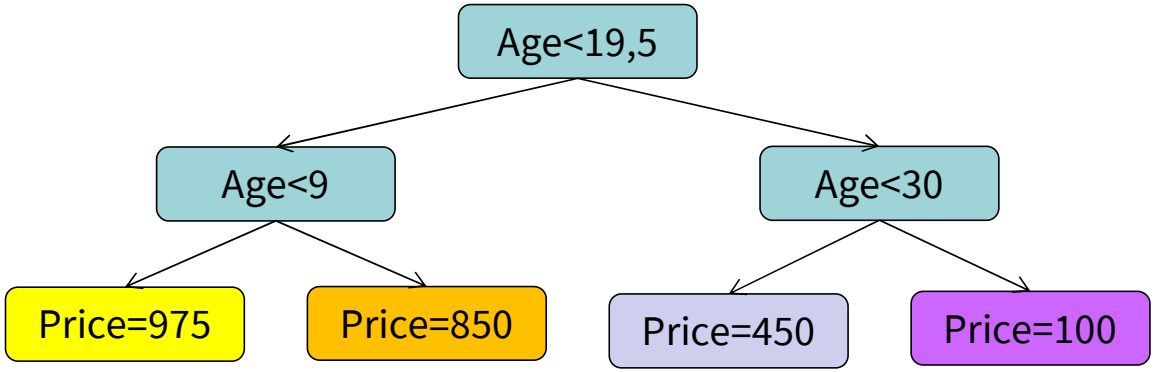
Find best split for these examples

Following exactly the same logic as before. That is, test all candidate splits and pick the one that has minimal SSR.



Regression Tree with Four Leaf Nodes

Age [months]	Resale price [$]
3	1000
4,5	1000
6	950
12	850
15	825
18	825
21	450
24	425
27	400
33	100
34,5	100
36	100
39	100



Notice how a tree with k=4 leaf nodes partitions the training set into *k=4* subgroups.

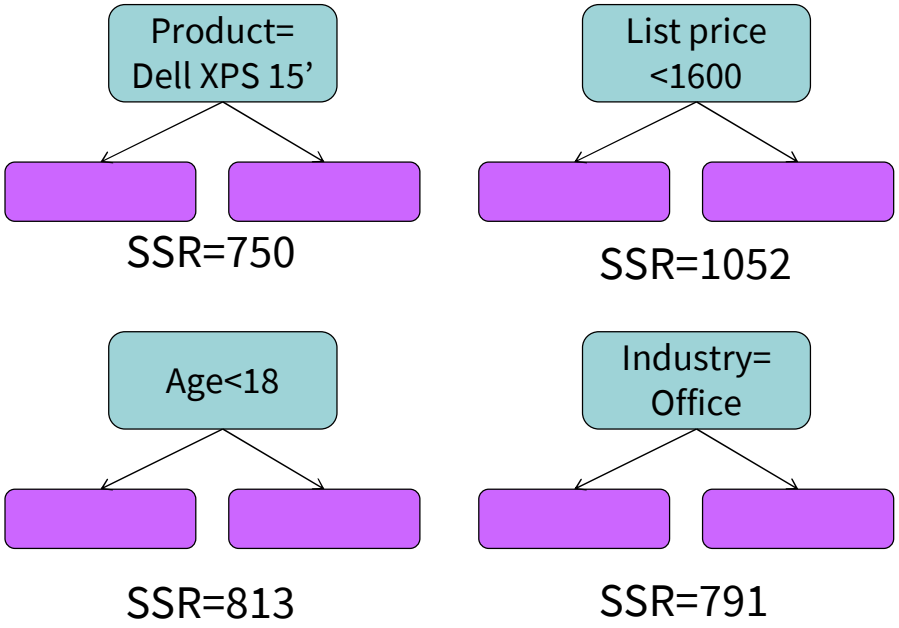
Also recall that each leaf node outputs an estimate of the target variable. This estimate is simply the average value of the target variable computed among the training set examples that fall into the leaf node.

When we process new data and generate forecasts, determine to which leaf node a new example belongs and then take that leaf node's output value as forecast. Consequently, a tree with *k=4* leaf nodes can forecast no more than *k=4* distinct values. In other words, no matter what new data we put down the above tree, the forecasted resale price of the tree will always be a value in (975, 850, 450, 100).

Regression Tree Learning: Extension to Multivariate Settings

Find the best split per feature for all features and pick the best overall split

- **Real-word data sets comprise many features**
- **This does not alter our approach to find splits**
- **Find best split for each feature and then select the best overall split**

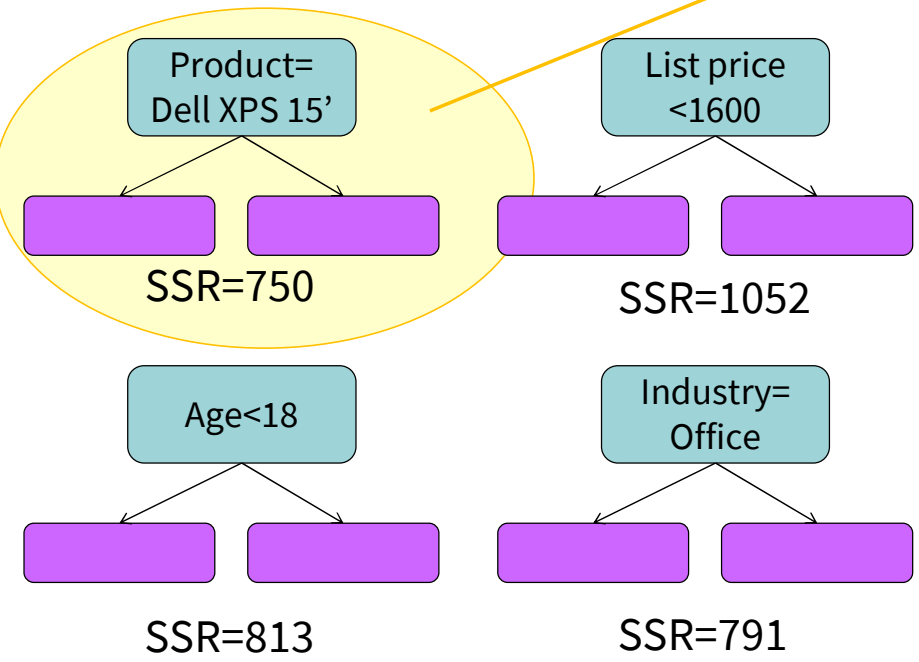


Product	List price [$]	Age [month]	Industry	…	Observed resale price [$]
Dell XPS 15’	2,500	36	Mining	…	347
Dell XPS 15’	2,500	24	Health	…	416
Dell XPS 17’	3,000	36	Manufacturing	…	538
HP Envy 17’	1,300	24	Office	…	121
HP EliteBook 850	1,900	36	Manufacturing	…	172
Lenovo Yoga 11’	799	12	Office	…	88
Lenovo Yoga 13’	1,100	12	Office	…	266
…	…	…	…	…	…

Regression Tree Learning: Extension to Multivariate Settings

Find the best split per feature for all features and pick the best overall split

- **Real-word data sets comprise many features**
- **This does not alter our approach to find splits**
- **Find best split for each feature and then select**



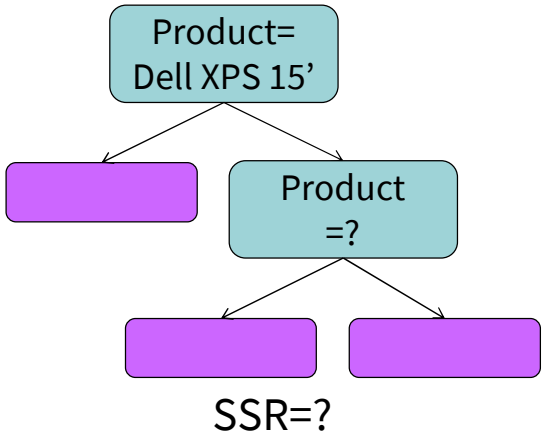
Product	Li				
Dell XPS 15'					
Dell XPS 15'	2,500	24	Health	…	416
Dell XPS 17'	3,000	36	Manufacturing	…	538
HP Envy 17'	1,300	24	Office	…	121
HP EliteBook 850	1,900	36	Manufacturing	…	172
Lenovo Yoga 11'	799	12	Office	…	88
Lenovo Yoga 13'	1,100	12	Office	…	266
…	…	…	…	…	…

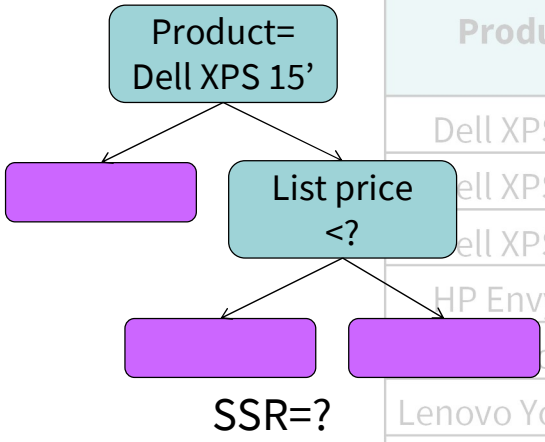
Assume we found these thresholds to give the respectively best splits (i.e., lowest SSR) per feature. Next, we compare the SSR across features. Assume the corresponding SSR values are as illustrated. Then we select the best overall split on feature PRODUCT. Next, we repeat the whole process of finding an optimal split for each of the resulting child nodes.

Regression Tree Learning: Extension to Multivariate Settings

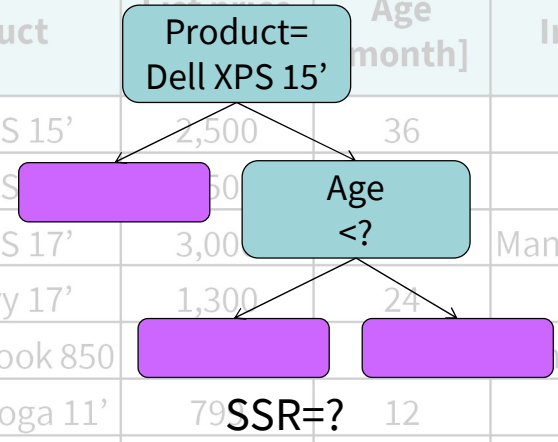
Find the best split per feature for all features and pick the best overall split

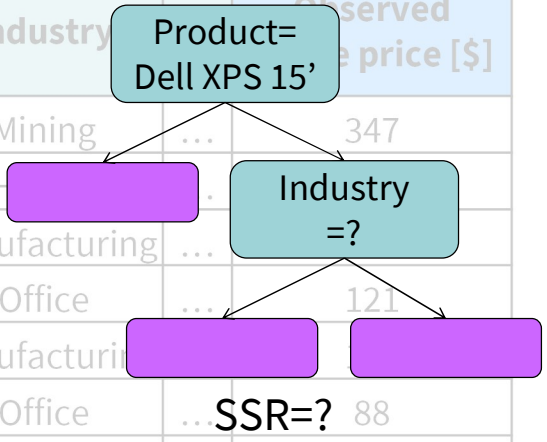
- **Real-word data sets comprise many features**
- **This does not alter our approach to find splits**
- **Find best split for each feature and then select the best overall split**





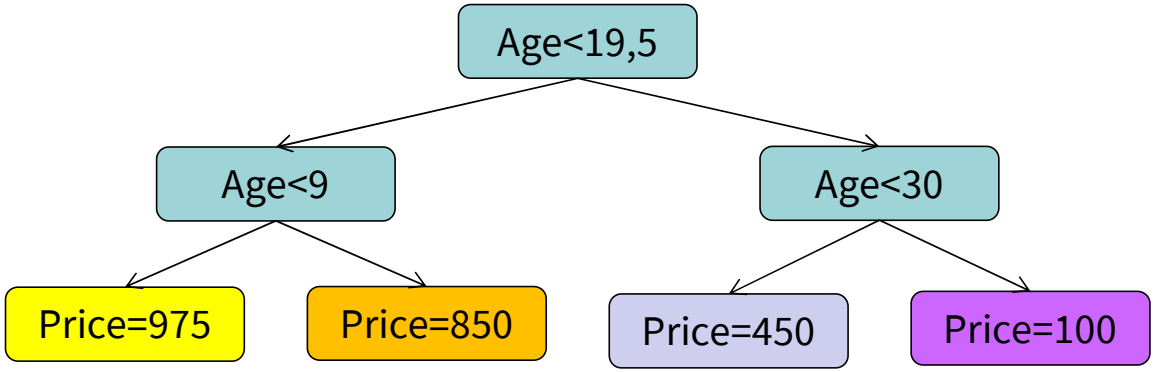
Product	List price	Age [month]	Industry	…	Observed price [$]
Dell XPS 15’	2,500	36	Mining	…	347
ell XPS	50		H	.	
ell XPS 17’	3,00		Manufacturing	…	
HP Envy 17’	1,300	24	Office	…	121
ook 850			ufacturi		
Lenovo Yoga 11’	79	12	Office	…	88
Lenovo Yoga 13’	1,100	12	Office	…	266
…	…	…	…	…	…





When to Stop Tree Growing

Age [months]	Resale price [$]
3	1000
4,5	1000
6	950
12	850
15	825
18	825
21	450
24	425
27	400
33	100
34,5	100
36	100
39	100



We can continue growing the tree until

- Each leaf node has only one data point
- Further splits do no longer improve SSR

But would that be a good idea?

Tree Depth and Overfitting

Deep trees are prone to overfitting

- **Overfitting problem occurs with (too) complex models**
- **Number of nodes in a tree captures its complexity**
  - Whether it can accommodate complex structure in data
  - Nonlinear patters, interactions, etc.

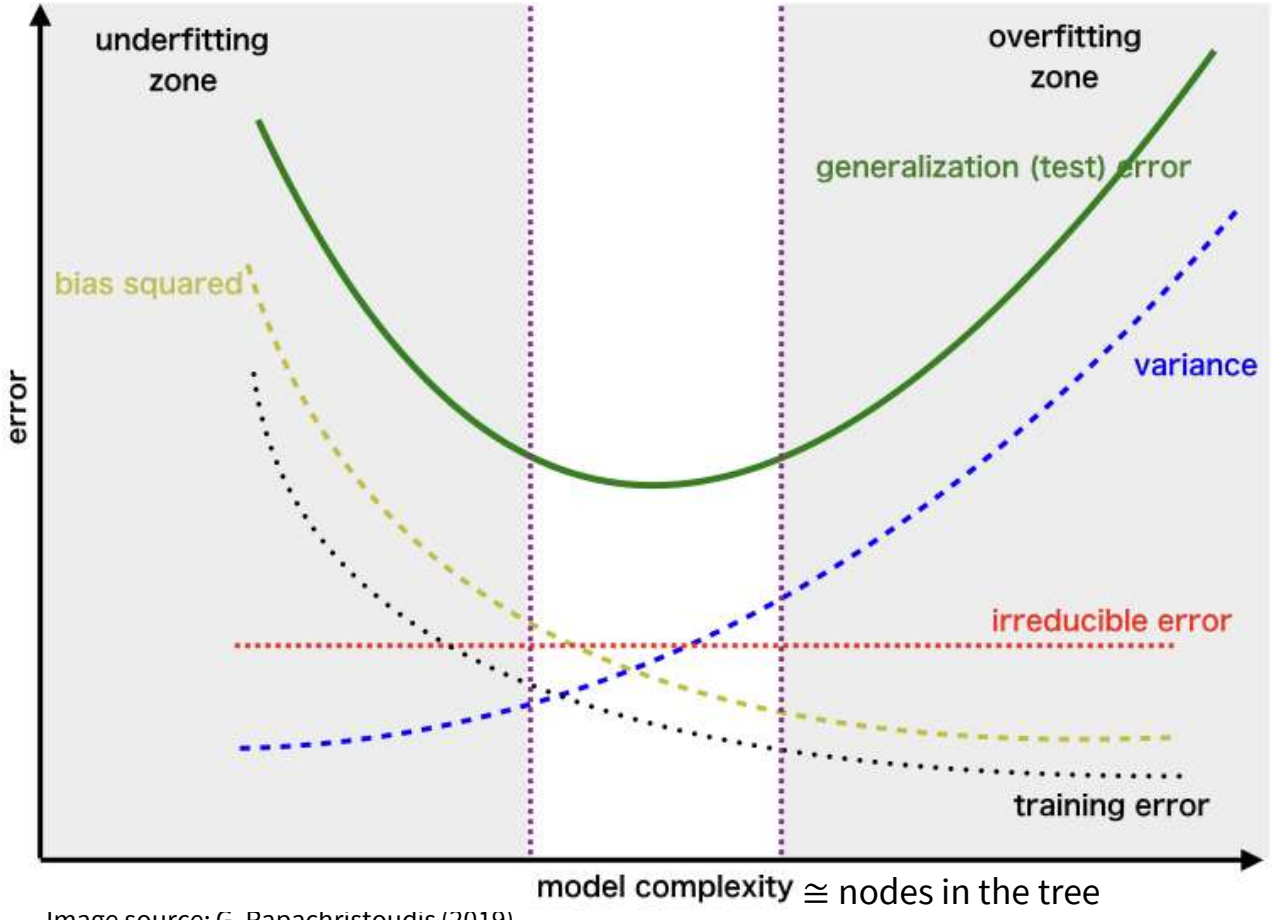
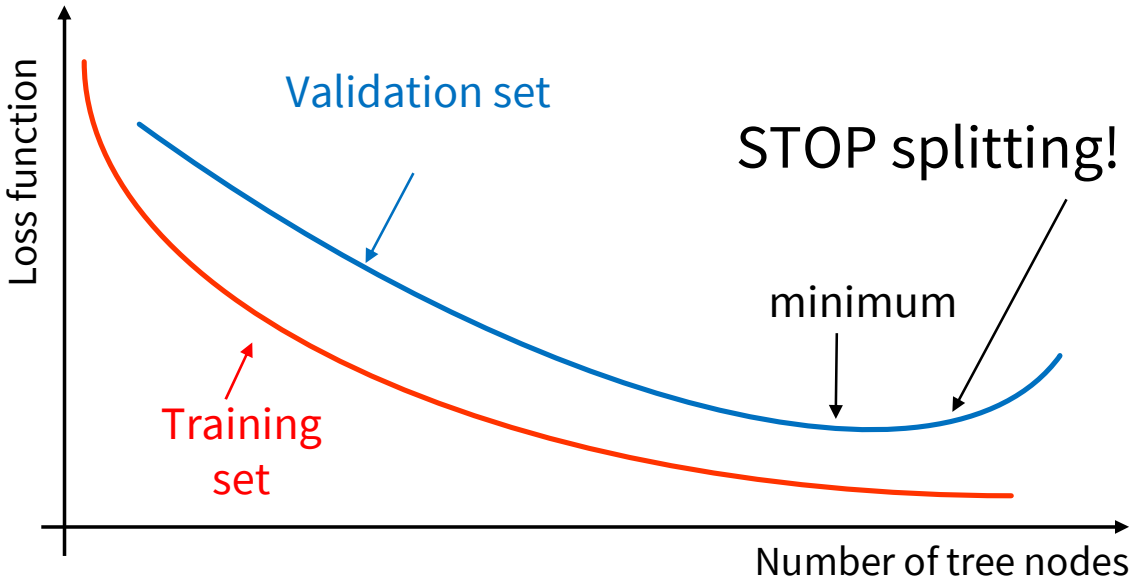


Image source: G. Papachristoudis (2019)
https://towardsdatascience.com/the-bias-variance-tradeoff-8818f41e39e9

Tree Pruning to Reduce Complexity and Protect Against Overfitting

Stop growing the tree when its validation error starts to increase

- **Pre-Pruning**
  - Through algorithmic hyperparameters
  - Max. depth, min. data points per node, …
- **Post Pruning**
  - Grow full tree and merge nodes ex post
  - Based on, e.g., increase in SSR
- **Early stopping**
  - Monitor tree performance on separate validation set drawn from the training data
  - Stop tree growing when validation error increases
    - Evolution of validation error follows from bias-variance trade-off
    - Shallow tree → high bias → high error / deep tree → high variance → high error



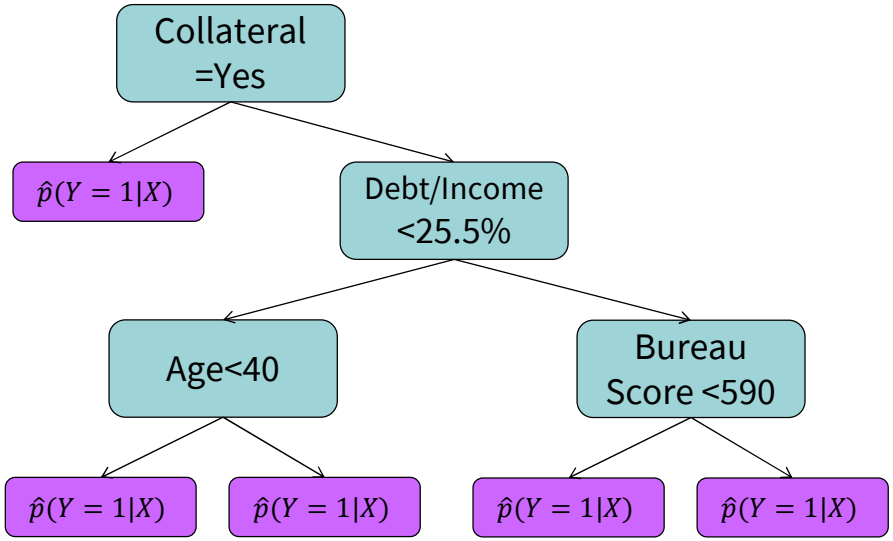
Regression Tree Learning in Pseudo-Code

```
Start from root node
For each feature
  Find best split
    Nominal variables: consider splits X=a, X=b, ...
    Numeric variables: consider splits X < a
    Assess quality of in terms of reducing SSR (or other loss function)
Compare best splits per feature across features
  SSR (best split at age) vs. SSR (best split at list price) vs. …
Select best overall split
Repeat above steps for each child node
Continue tree growing until stopping criterion or tree has
reached its full size
```

Classification Tree Learning Follows the Same Approach

Two revisions: need of a loss function to guiding splitting & predictions become probabilities

- **New loss function: information gain assesses node impurity**
- **Leaf nodes predict class membership probability**
  - Relative frequency of the target variable in a leaf

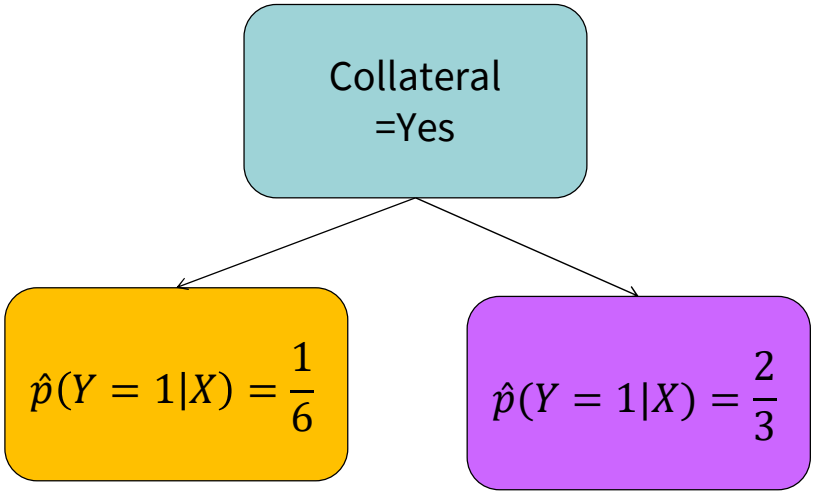


Bureau score	Collateral	Debt/ Income	Years at address	Age	…	Default
650	Yes	20%	2	<21	…	**No**
280	No	43%	0	21-29	…	**Yes**
750	Yes	27%	8	30-39	…	**No**
600	Yes	18%	4	40-50	…	**No**
575	No	33%	12	>50	…	**No**
715	Yes	24%	1	21-29	…	**No**
580	No	28%	6	40-50	…	**Yes**
410	Yes	29%	4	21-29	…	**No**
800	Yes	34%	10	40-50	…	**Yes**

Classification Tree Learning Follows the Same Approach

Two revisions: need of a loss function to guiding splitting & predictions become probabilities

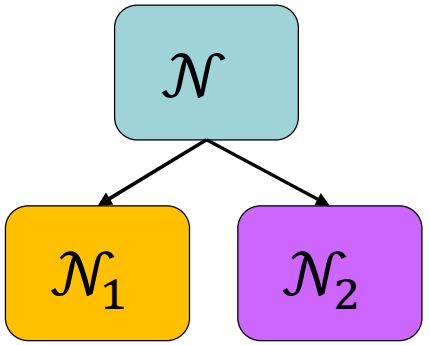
- **Classification requires probabilistic predictions**
  - Leaf nodes estimate class membership probabilities
  - Approximated as relative frequency of the target variable in a leaf
- **New loss function: information gain (IG) assesses node impurity**



Bureau score	Collateral	Debt/ Income	Years at address	Age	…	Default
650	Yes	20%	2	<21	…	**No**
280	No	43%	0	21-29	…	**Yes**
750	Yes	27%	8	30-39	…	**No**
600	Yes	18%	4	40-50	…	**No**
575	No	33%	12	>50	…	**No**
715	Yes	24%	1	21-29	…	**No**
580	No	28%	6	40-50	…	**Yes**
410	Yes	29%	4	21-29	…	**No**
800	Yes	34%	10	40-50	…	**Yes**

Information Gain (IG)

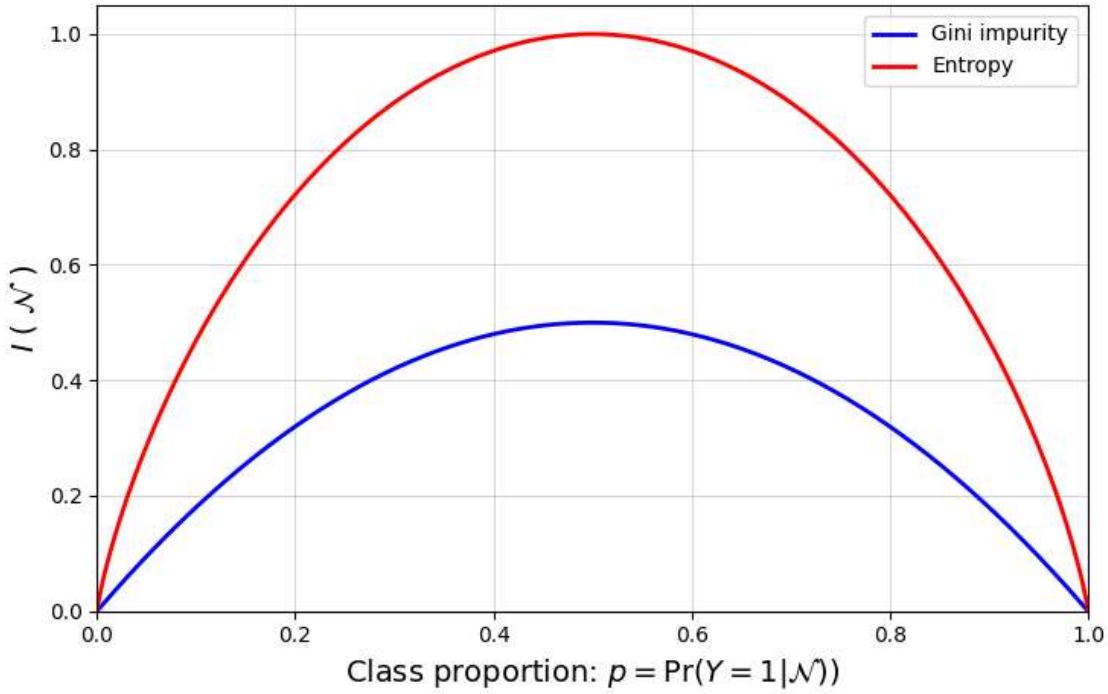
Reduction of node impurity due to splitting



- **Nodes should be pure → only positive or negative examples**
- **Measures of the node impurity**
  - Entropy: $I(\mathcal{N}) = -\sum_c p(y_c|\mathcal{N}) \cdot \log_2\left(p(y_c|\mathcal{N})\right)$
  - Gini impurity: $I(\mathcal{N}) = 1 - \sum_c p(y_c|\mathcal{N})^2$
  - where $c = 1, 2, \ldots \mathcal{C}$ are the possible outcomes
- **Information gain of splitting node $\mathcal{N}$ into child nodes $\mathcal{N}_1$ and $\mathcal{N}_2$**

$$IG(\mathcal{N}) = I(\mathcal{N}) - p_{\mathcal{N}_1} I(\mathcal{N}_1) - p_{\mathcal{N}_2} I(\mathcal{N}_2)$$

  - with $p_{\mathcal{N}_1} = \frac{|\mathcal{N}_1|}{|\mathcal{N}|}$ and $p_{\mathcal{N}_2} = \frac{|\mathcal{N}_2|}{|\mathcal{N}|}$
  - the share of data points in node $\mathcal{N}_1$ and $\mathcal{N}_2$



Implementing Ensemble Learning

Random Forest and Gradient Boosting

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
- **Say this is our original data set**

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

- **Bootstrapping creates as many somewhat different training sets 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

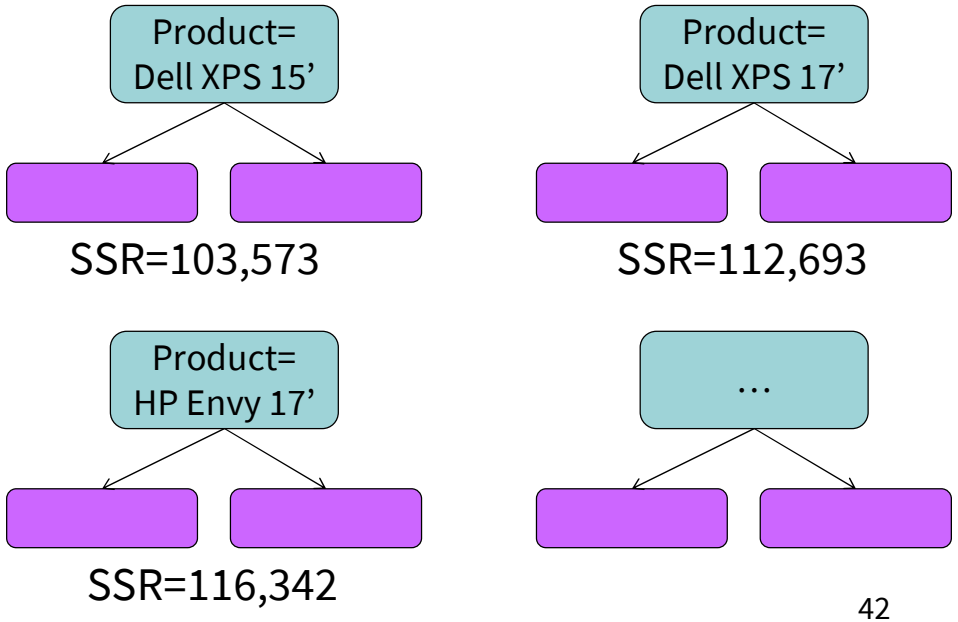
i	Product	List price	Age	Industry	…	Resale price [$]
5	Lenovo Yoga 13’	1,100	12	Office	…	266
3	HP Envy 17’	1,300	24	Office	…	121
4	HP EliteBook 850	1,900	36	Mining	…	172
1	Dell XPS 15’	2,500	36	Mining	…	347
3	HP Envy 17’	1,300	24	Office	…	121

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 overall

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

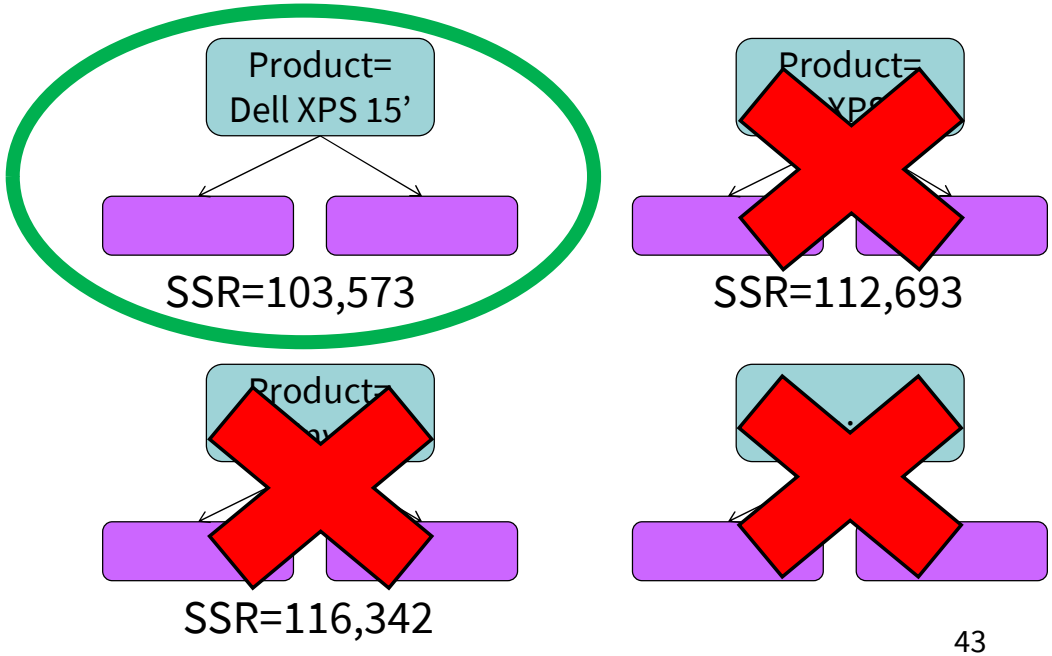


Random Forest Algorithm

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 overall

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

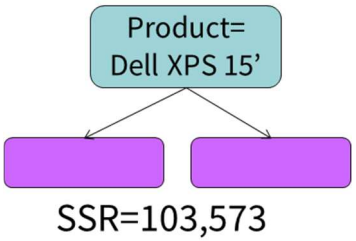


Random Forest Algorithm

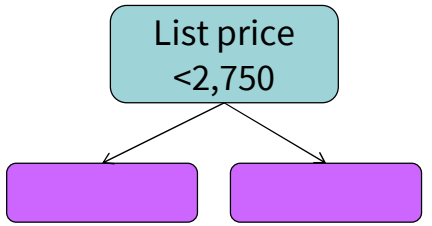
Grow each tree using the random subspace mechanism

- **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 overall

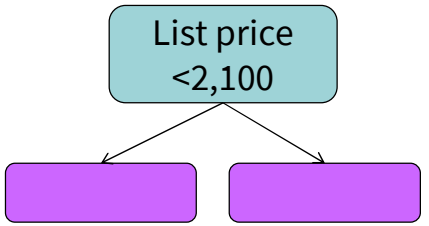
i	Product	List price	Age	Industry	…	Resale price [$]
1	Dell XPS 15’	2,500	36	Mining	…	347
2	Dell XPS 17’	3,000	36	Health	…	538
3	HP Envy 17’	1,300	24	Office	…	121
4	HP EliteBook 850	1,900	36	Mining	…	172
5	Lenovo Yoga 13’	1,100	12	Office	…	266



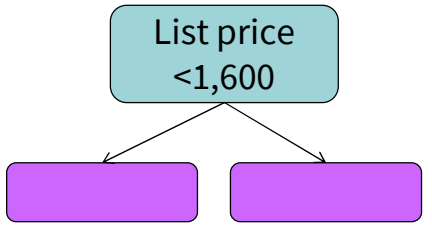
SSR=103,573



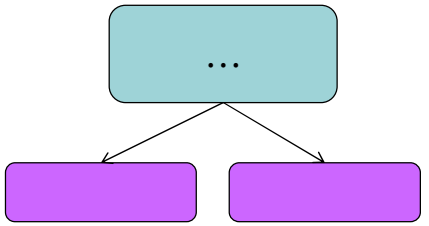
SSR= 112,693



SSR=29,061



SSR=77,533

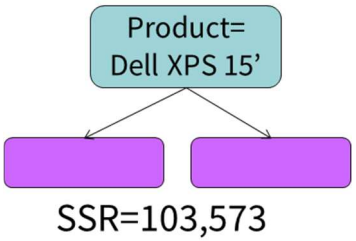


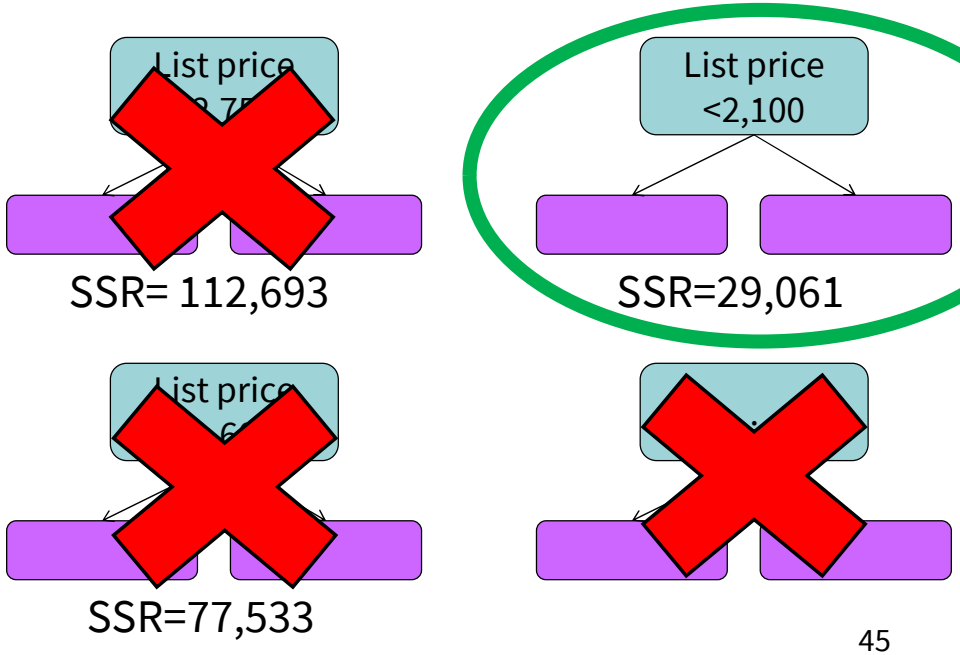
Random Forest Algorithm

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 overall

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



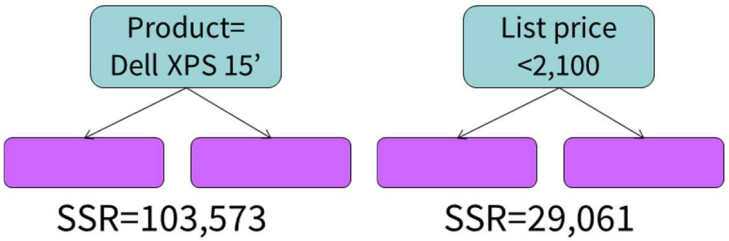


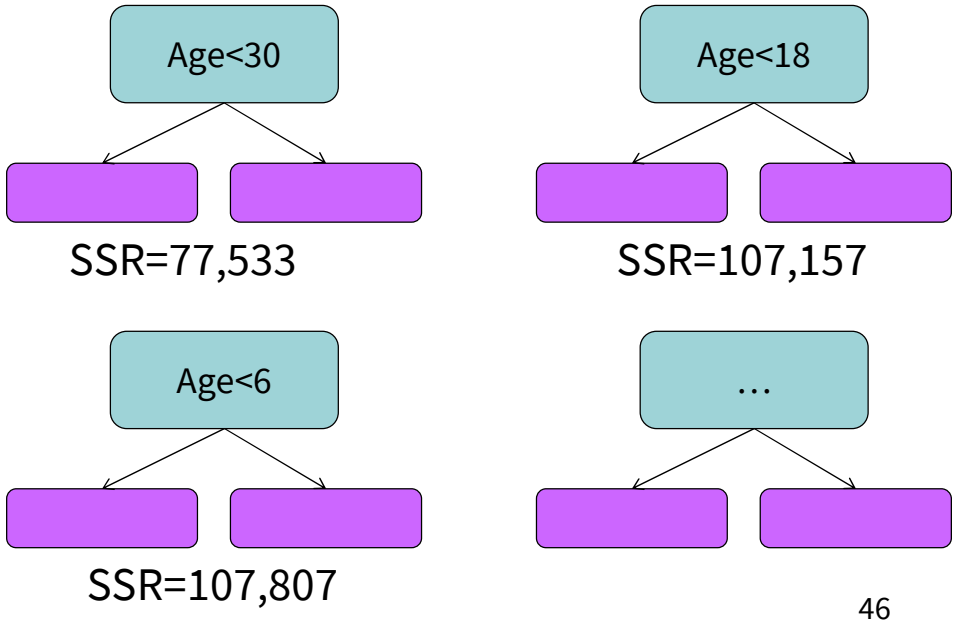
Random Forest Algorithm

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 overall

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



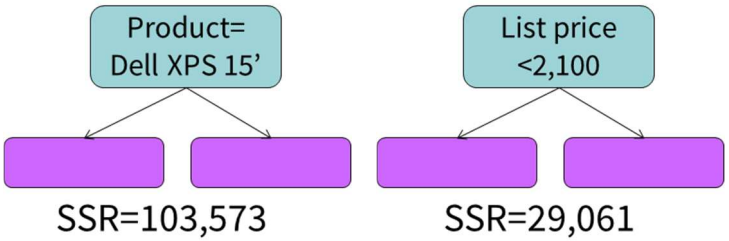


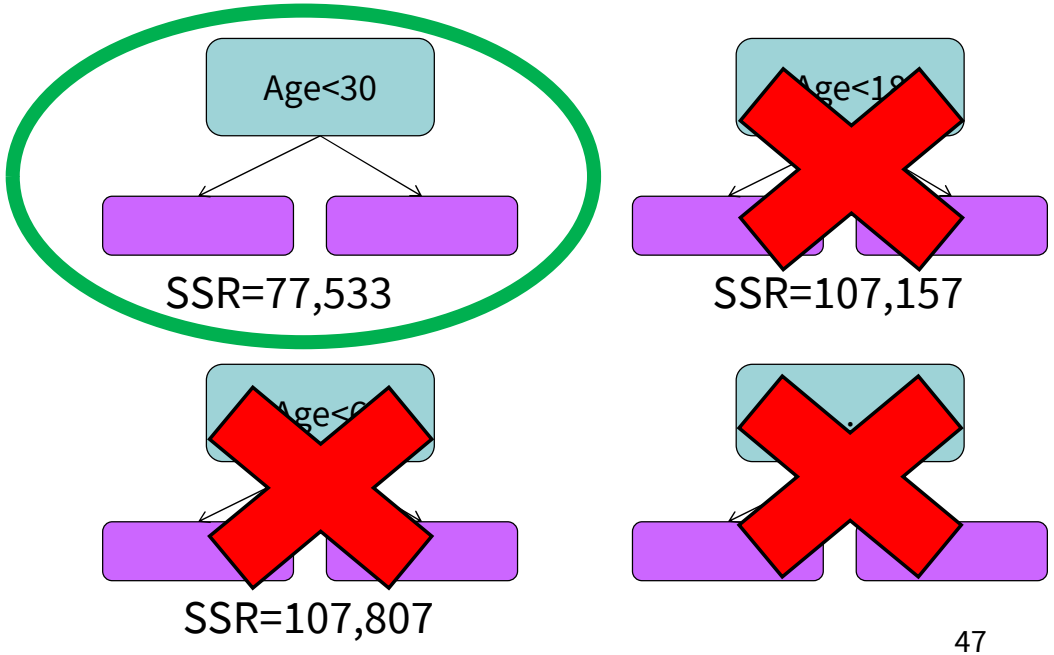
Random Forest Algorithm

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 overall

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



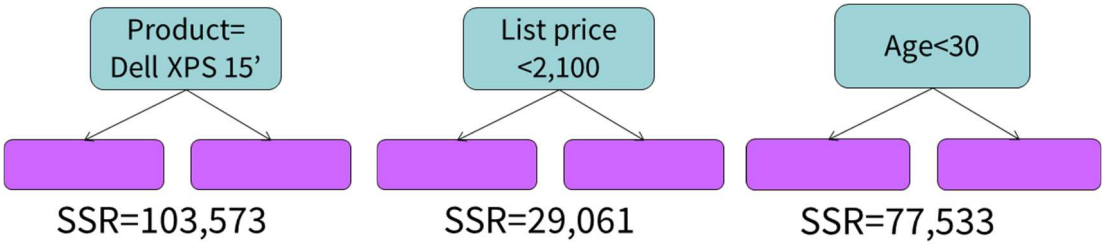


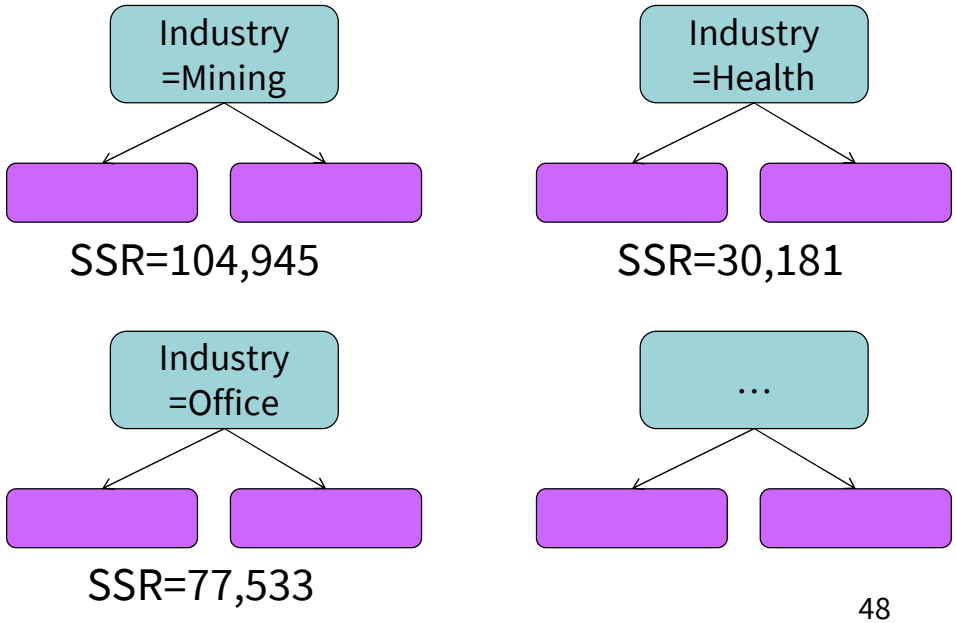
Random Forest Algorithm

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



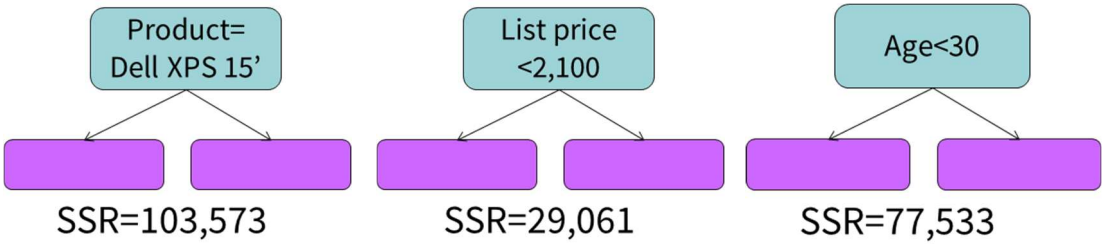


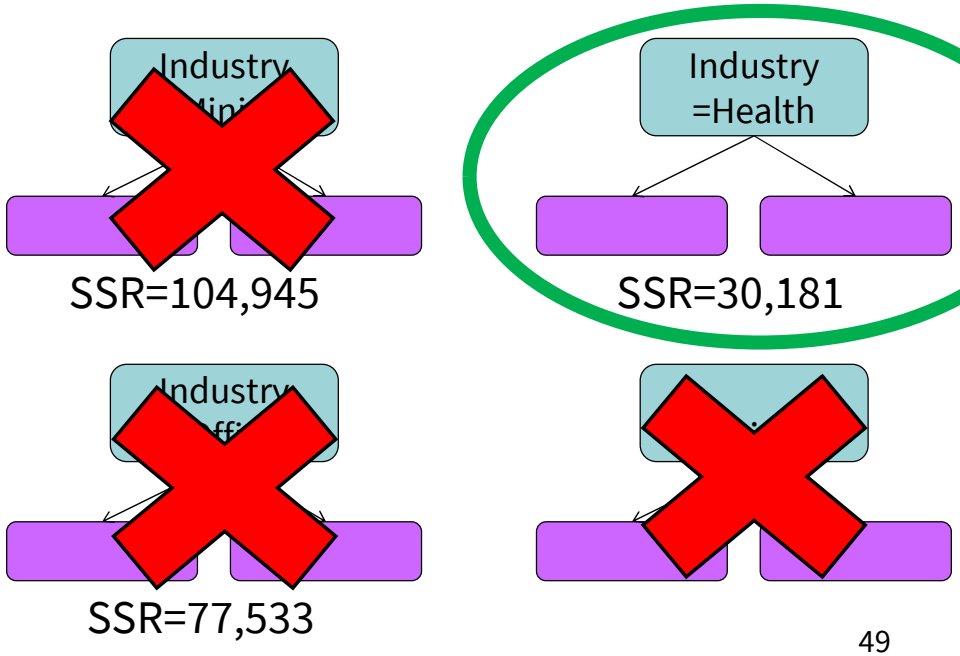
Random Forest Algorithm

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 overall

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



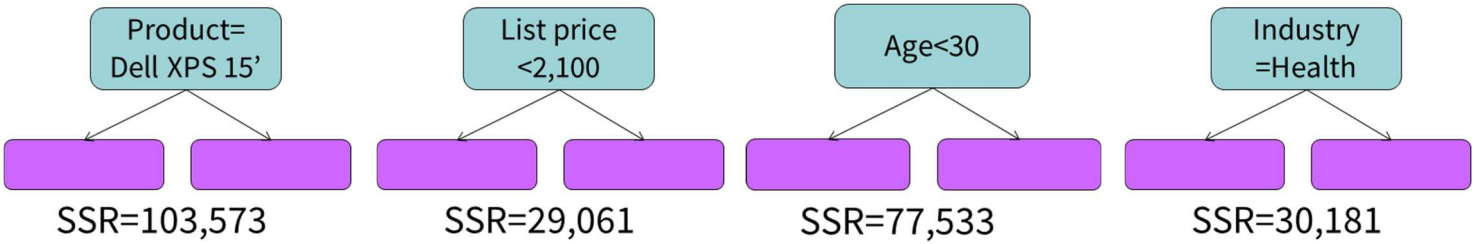


Random Forest Algorithm

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 overall

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

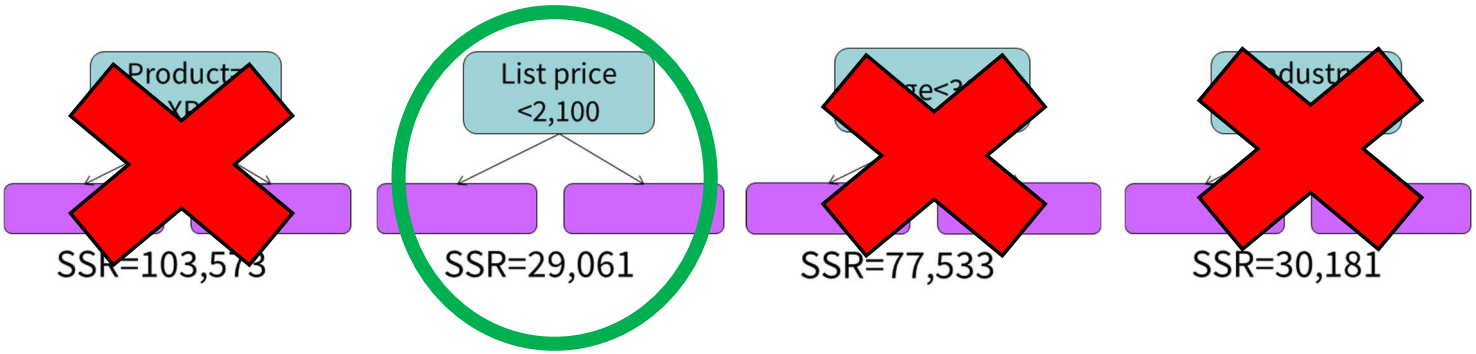


Random Forest Algorithm

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 overall

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

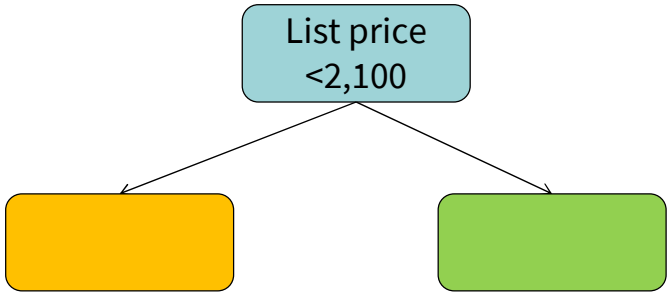


Random Forest Algorithm

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 overall



  - 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 overall
  - 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

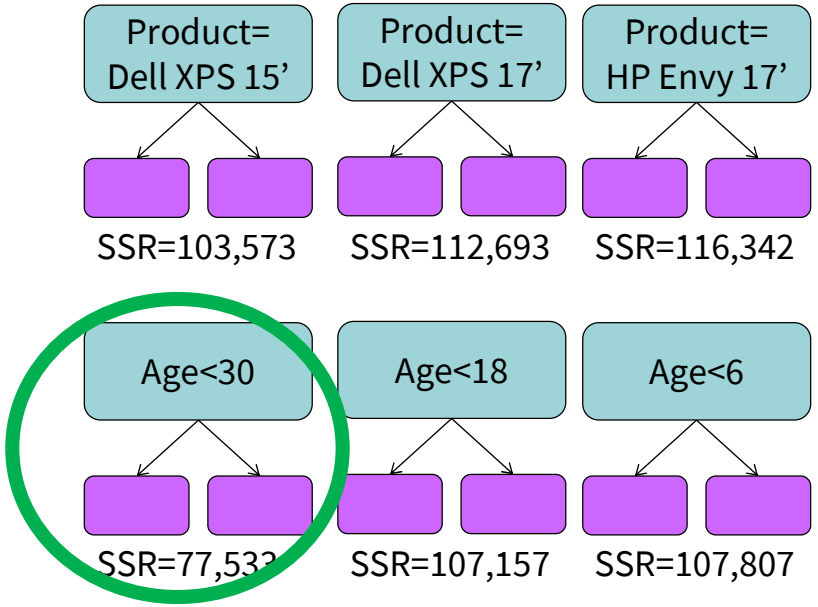
- **Random subspace modifies how we grow a tree**
- **Key decision in tree growing is finding splits**
  - Standard/previous approach
    - Step 1: For each feature, find the best split on that feature
    - Step 2: Select the best split overall
  - Random subspace approach
    - Step 1a: Select a subset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

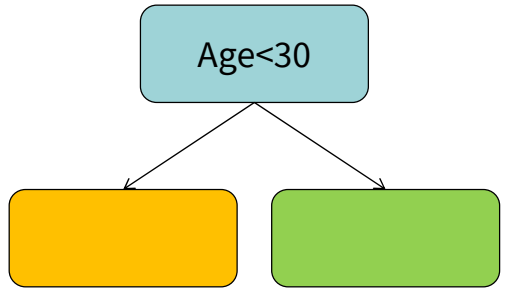
i	Product	List price	Age	Industry	…	Resale price [$]
1	Dell XPS 15’	2,500	36	Mining	…	347
2	Dell XPS 17’	3,000	36	Health	…	538
3	HP Envy 17’	1,300	24	Office	…	121
4	HP EliteBook 850	1,900	36	Mining	…	172
5	Lenovo Yoga 13’	1,100	12	Office	…	266



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



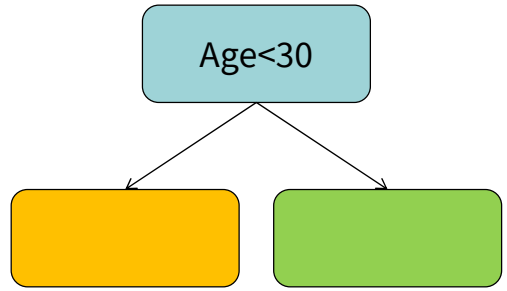
- **Random subspace modifies how we grow a tree**
- **Key decision in tree growing is finding splits**
  - Standard/previous approach
    - Step 1: For each feature, find the best split on that feature
    - Step 2: Select the best split overall
  - Random subspace approach
    - Step 1a: Select a subset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

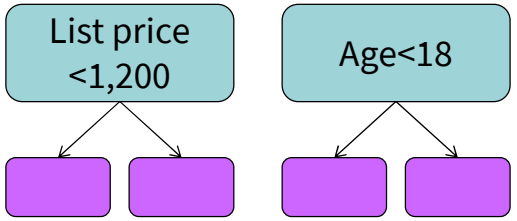
Random Forest Algorithm

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 overall
  - 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





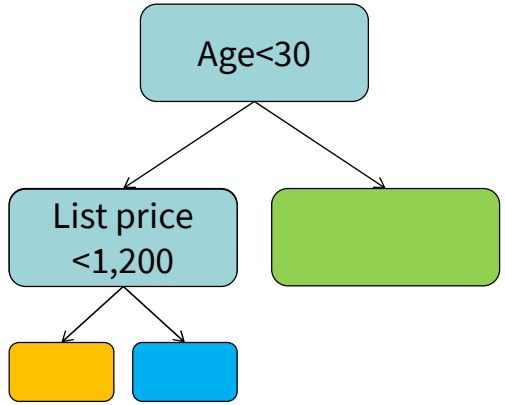
SSR is the same for these splits since we have only two data points left.

Random Forest Algorithm

Grow each tree using the random subspace mechanism

- **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 overall
  - 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

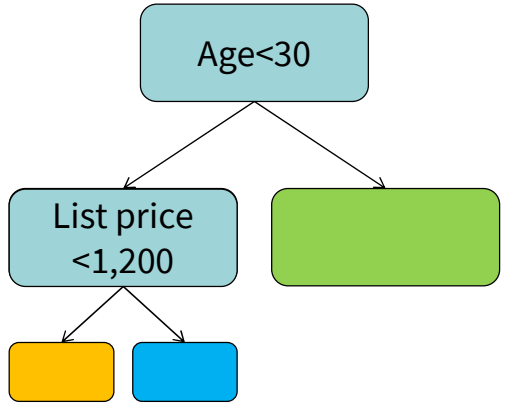


Random Forest Algorithm

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 overall
  - 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

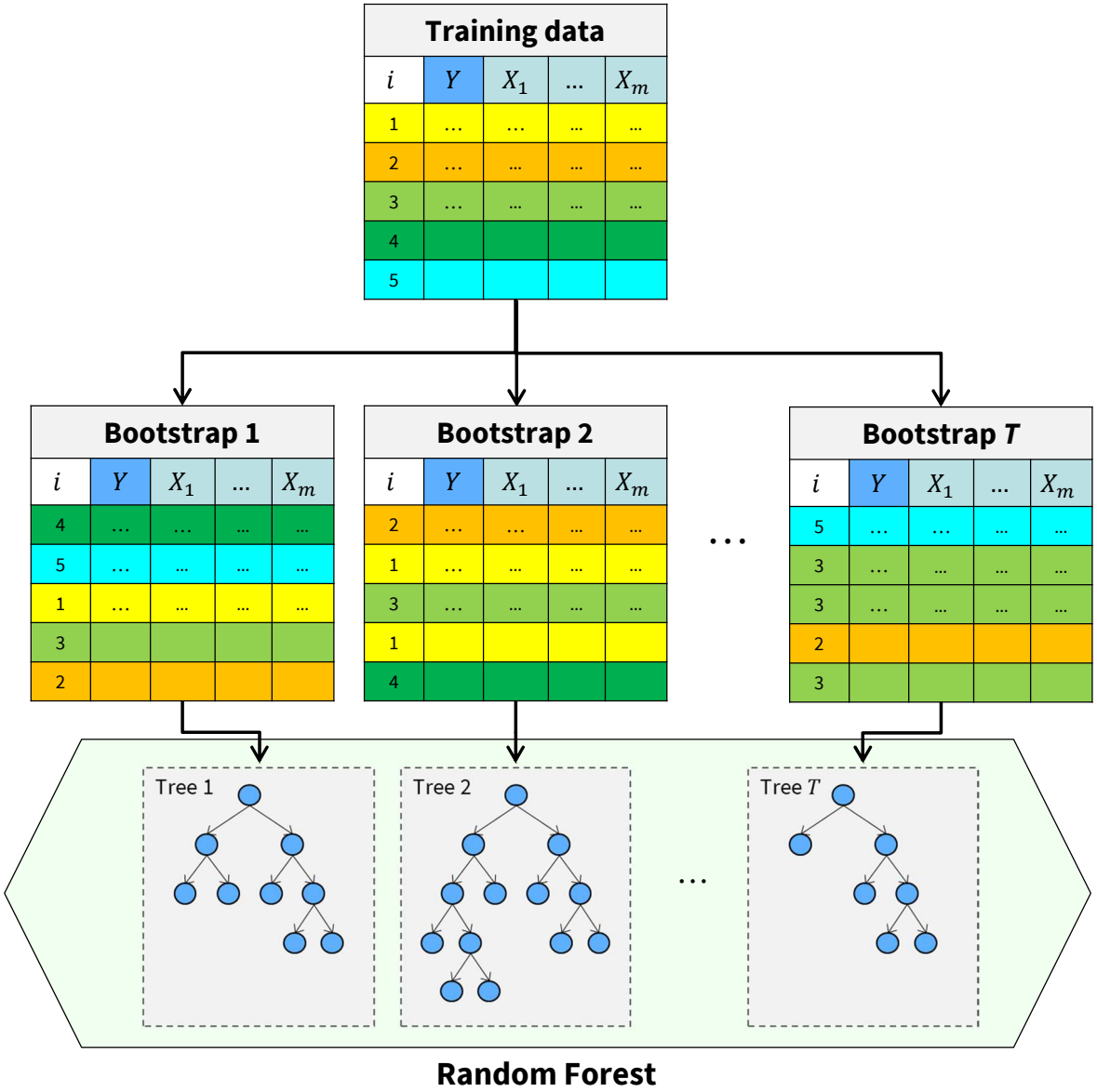
i	Product	List price	Age	Industry	…	Resale price [$]
1	Dell XPS 15’	2,500	36	Mining	…	347
2	Dell XPS 17’	3,000	36	Health	…	538
3	HP Envy 17’	1,300	24	Office	…	121
4	HP EliteBook 850	1,900	36	Mining	…	172
5	Lenovo Yoga 13’	1,100	12	Office	…	266



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



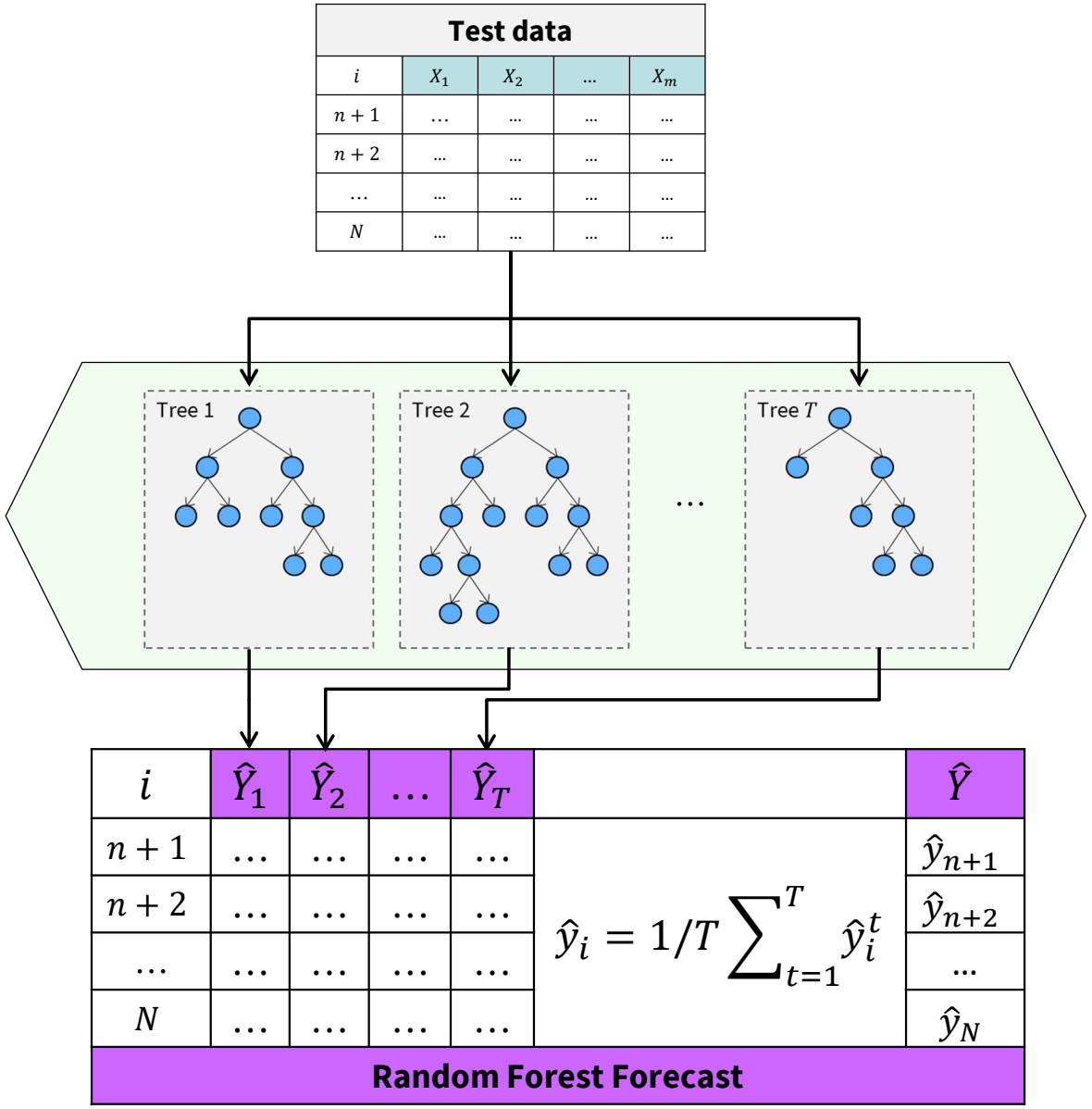
Random Forest in Pseudo-Code

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

Hyperparameter 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 (e.g., tree depth)
- **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 $\sqrt{m}$

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**

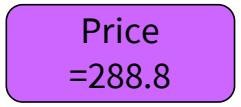
Price =288.8 t=0

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**



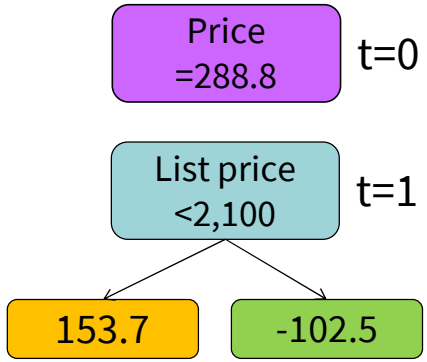
t=0

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**

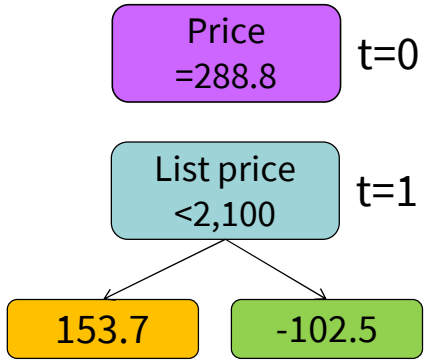


i	Product	List price	Age	Industry	…	Resale price [$]	$\epsilon^{\mathrm{GBM}_0}$
1	Dell XPS 15’	2,500	36	Mining	…	347	58.2
2	Dell XPS 17’	3,000	36	Health	…	538	249.2
3	HP Envy 17’	1,300	24	Office	…	121	-167.8
4	HP EliteBook 850	1,900	36	Mining	…	172	-116.8
5	Lenovo Yoga 13’	1,100	12	Office	…	266	-22.8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

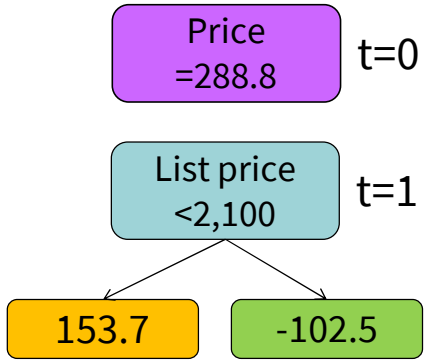


i	Product	List price	Age	Industry	…	Resale price [$]	$\epsilon^{GB}{}_0$	$\hat{Y}^1$
1	Dell XPS 15’	2,500	36	Mining	…	347	58.2	153.7
2	Dell XPS 17’	3,000	36	Health	…	538	249.2	153.7
3	HP Envy 17’	1,300	24	Office	…	121	-167.8	-102.5
4	HP EliteBook 850	1,900	36	Mining	…	172	-116.8	-102.5
5	Lenovo Yoga 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 [$]	$\epsilon^{\mathrm{GBM}_0}$	$\hat{Y}^1$
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:

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

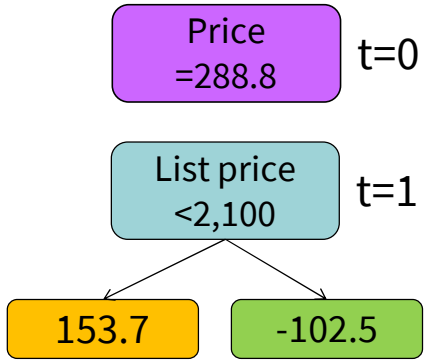
$$\hat{Y}^{\mathrm{GBM}_t} = \hat{Y}^{\mathrm{GBM}_{t-1}} - \eta \frac{\partial J(w)}{\partial w}$$

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 $\widehat{Y}^{\mathrm{GBM}_1}$
  - The residuals $\epsilon^{\mathrm{GBM}_1}$ are guaranteed to be lower than before



i	Product	List price	Age	Industry	…	Resale price [$]	ϵ^{GB_0}	$\widehat{Y}^1$	$\widehat{Y}^{\mathrm{GB}_1} = \widehat{Y}^{\mathrm{GB}_0} + \eta\widehat{Y}^1$	$Y - \widehat{Y}^{\mathrm{GBM}_1}$
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**

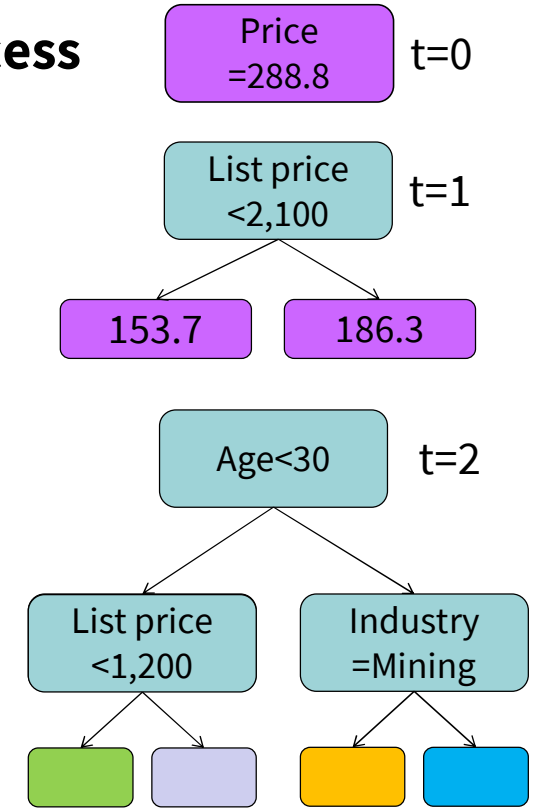
Note: η (eta) is a meta-parameter in (0,1) called the learning rate. We set it to 0.1 in the example.

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 $\hat{Y}^{\mathrm{GB}_2}$
  - 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 [$]	$\epsilon^{\mathrm{GBM}_1}$
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**

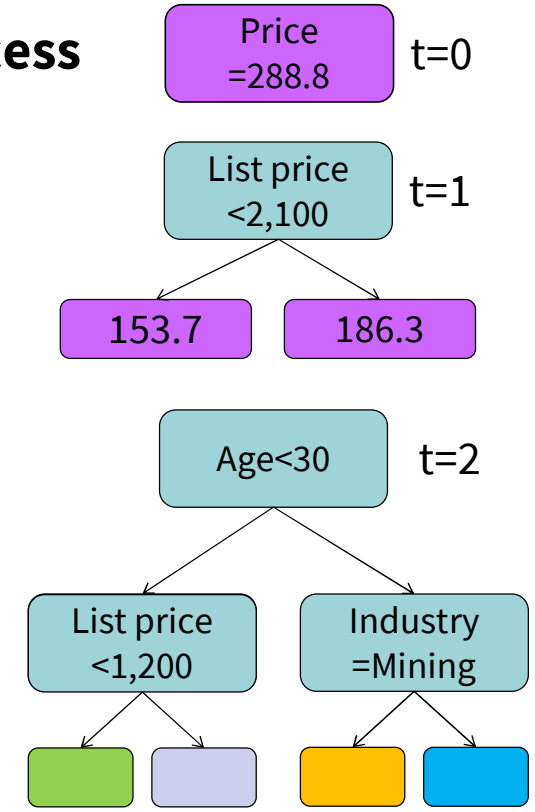


Gradient Boosting Algorithm

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 $\widehat{Y}^{\mathrm{GBM}_2}$
  - 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 [$]	$\epsilon^{\mathrm{GBM}_1}$
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**

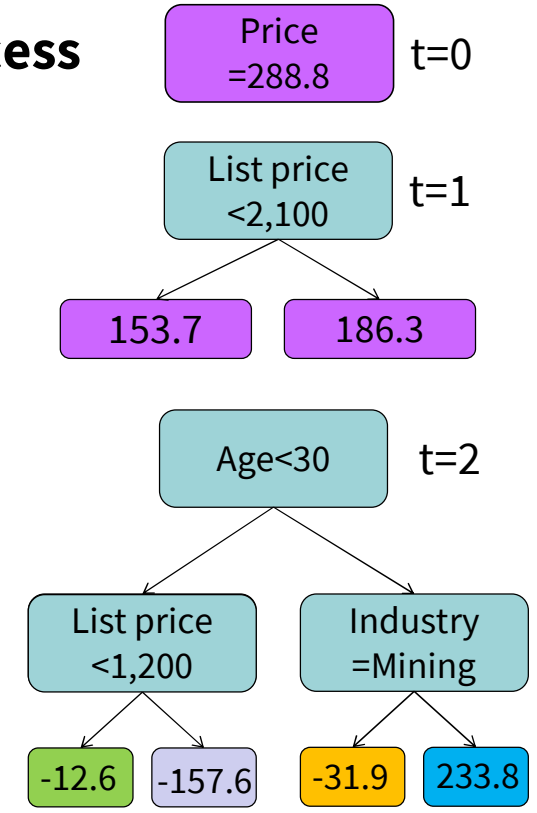


Gradient Boosting Algorithm

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**

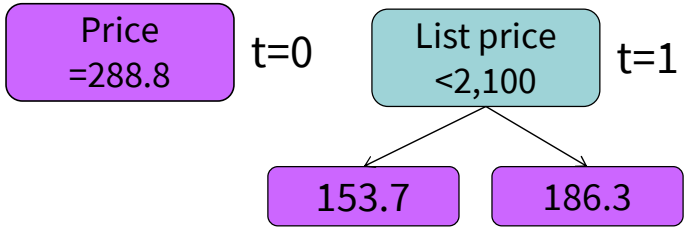
i	Product	List price	Age	Industry	…	Resale price [$]	$\epsilon^{\mathrm{GB}\ 1}$	$\hat{Y}^2$
1	Dell XPS 15’	2,500	36	Mining	…	347	42.8	-31.9
2	Dell XPS 17’	3,000	36	Health	…	538	233.8	233.8
3	HP Envy 17’	1,300	24	Office	…	121	-157.6	-157.6
4	HP EliteBook 850	1,900	36	Mining	…	172	-106.6	31.9
5	Lenovo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

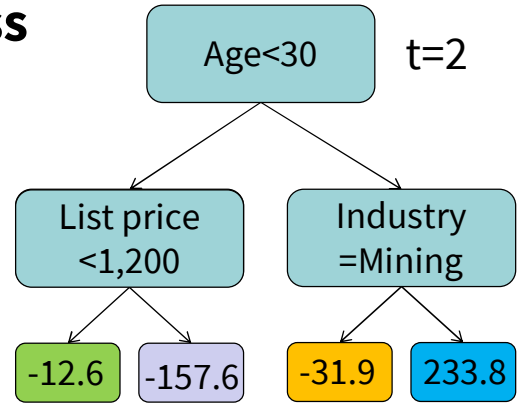


Gradient Boosting Algorithm

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**

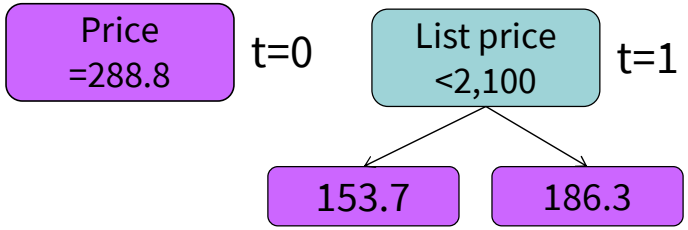
i	Product	List price	Age	Industry	…	Resale price [$]	$\epsilon^{\mathrm{GB}\ 1}$	$\hat{Y}^2$	$\hat{Y}^{\mathrm{GB}\ 2} = \hat{Y}^{\mathrm{GB}\ 1} + \eta\hat{Y}^2$
1	Dell XPS 15’	2,500	36	Mining	…	347	42.8	-31.9	304.2+0.1*31.9=301.0
2	Dell XPS 17’	3,000	36	Health	…	538	233.8	233.8	304.2+0.1*233.8=327.6
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**		

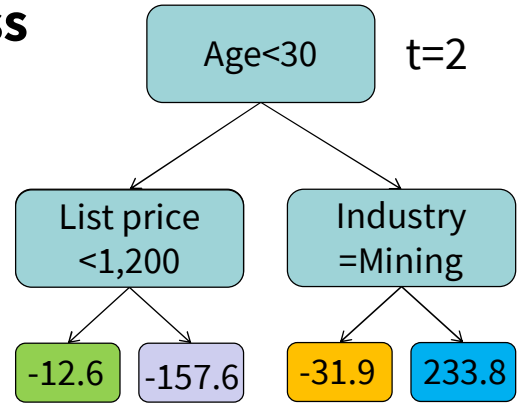
Note: η (eta) is a meta-parameter in (0,1) called the learning rate. We set it to 0.1 in the example.

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 $\hat{Y}^{\mathrm{GBM}_2}$
  - 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 [$]	$\epsilon^{\mathrm{GB}\ 1}$	$\hat{Y}^2$	$\hat{Y}^{\mathrm{GB}\ 2} = \hat{Y}^{\mathrm{GB}\ 1} + \eta\hat{Y}^2$	$Y - \hat{Y}^{\mathrm{GBM}_2}$
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 $\widehat{Y}^{GBM_0}$

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 $\mathcal{D}^t = \{\boldsymbol{X}, \boldsymbol{\epsilon}^{GBM_{t-1}}\}$

Compute output of tree t, $\hat{Y}^t$, on data $\mathcal{D}^t$

Compute new GBM forecast as $\hat{Y}^{GBM_t} = \hat{Y}^{GBM_t - 1} + \eta \hat{Y}^t$

Compute new residual forecast as $\boldsymbol{\epsilon}^{GB\ \ t} = Y - \hat{Y}^{GBM_t}$

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 Algorithm

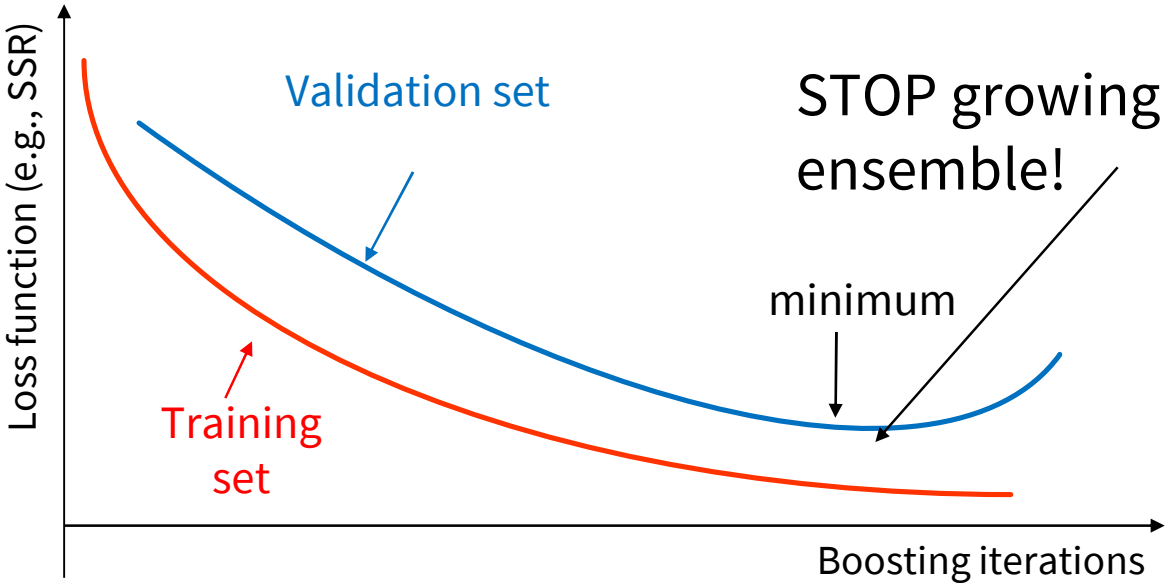
Contemporary libraries add several features and improvements

- **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**

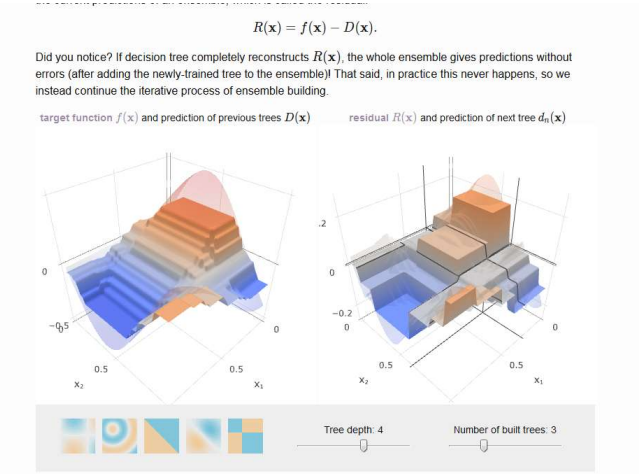
Gradient Boosting Algorithm

Practical use and configurations

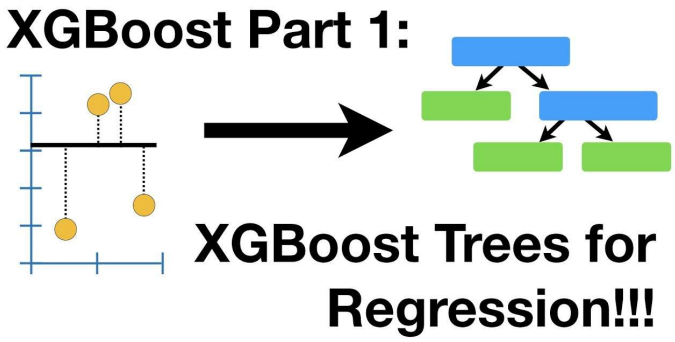
- **GBM repeats the above steps many times**
- **We use validation data to avoid overfitting (just as in ordinary trees)**
- **GBM hyperparameters**
  - Ensemble size *T*, often between 100 and 500
  - Tree depth, often between 3 and 12
  - Learning rate, always between (0,1)
  - Modern SW implementations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



Gradient Boosting Algorithm
Outlook

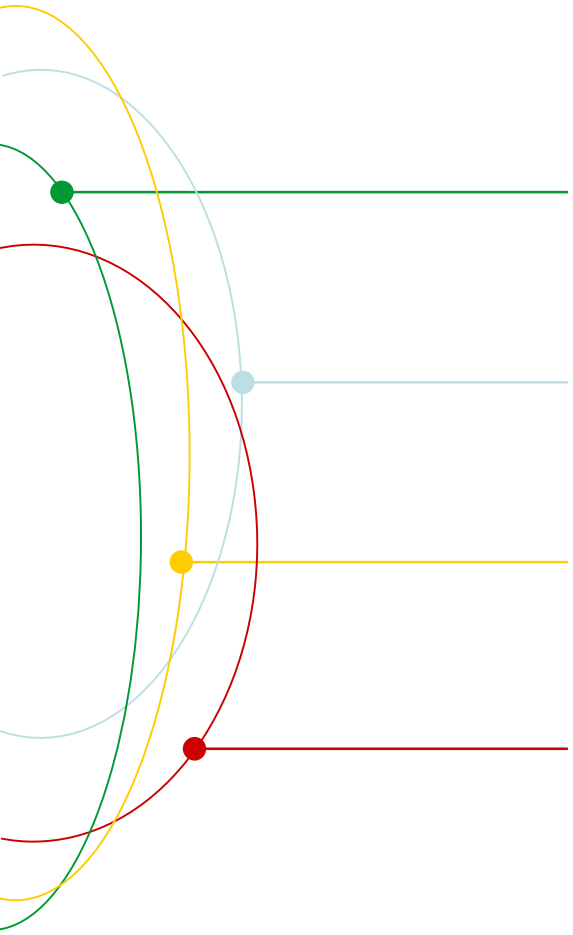


- **State-of-the-art packages:**
  - XGBoost [Chen & Guestrin, 2016]
  - GBMLight [Microsoft]
  - CatBoost [Yandex]
- **Further readings and resources**
  - Gradient boosting explained
    - Visual demo and playground by Alex Rogozhnikov
    - Available at http://arogozhnikov.github.io/2016/06/24/gradient_boosting_explained.html
  - 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



Summary

Summary



Learning goals

- Principles of ensemble learning
- Powerful tree-based ensemble learners

Findings

- Trees partition feature spaces through binary splits
  - Increase purity of child nodes
  - Leaf nodes generate constant predictions
- Ensembles combine base models to raise accuracy
- Bagging & RF combine independent base models
- Boosting grows ensembles incrementally

What next

- Coding session on ML-based credit scoring
- Apply learning algorithms in practice

Literature

- Breiman, L., Friedman, J. H., Olshen, R., & Stone, C. (1984). *Classification and Regression Trees*. Belmont: Wadsworth.
- 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.
- 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.
- Y. Freund, R. E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. Journal of Computer and System Science, 55, 119-139 (1997). Friedman, J., Hastie, T., & Tibshirani, R. (2000). Additive logistic regression: A statistical view of boosting. *The Annals of Statistics, 28*, 337-407.
- Hastie, T., Tibshirani, R., & Friedman, J. H. (2009). *The Elements of Statistical Learning* (2nd ed.). New York: Springer.
- Ho, T. K. (1998). The random subspace method for constructing decision forests. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 20*, 832-844.
- Quinlan, J. R. (1986). Induction of Decision Trees. *Machine Learning,* 1(1), 81-106.

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