Machine Learning

Week 4

Decision Trees &
Ensemble Methods

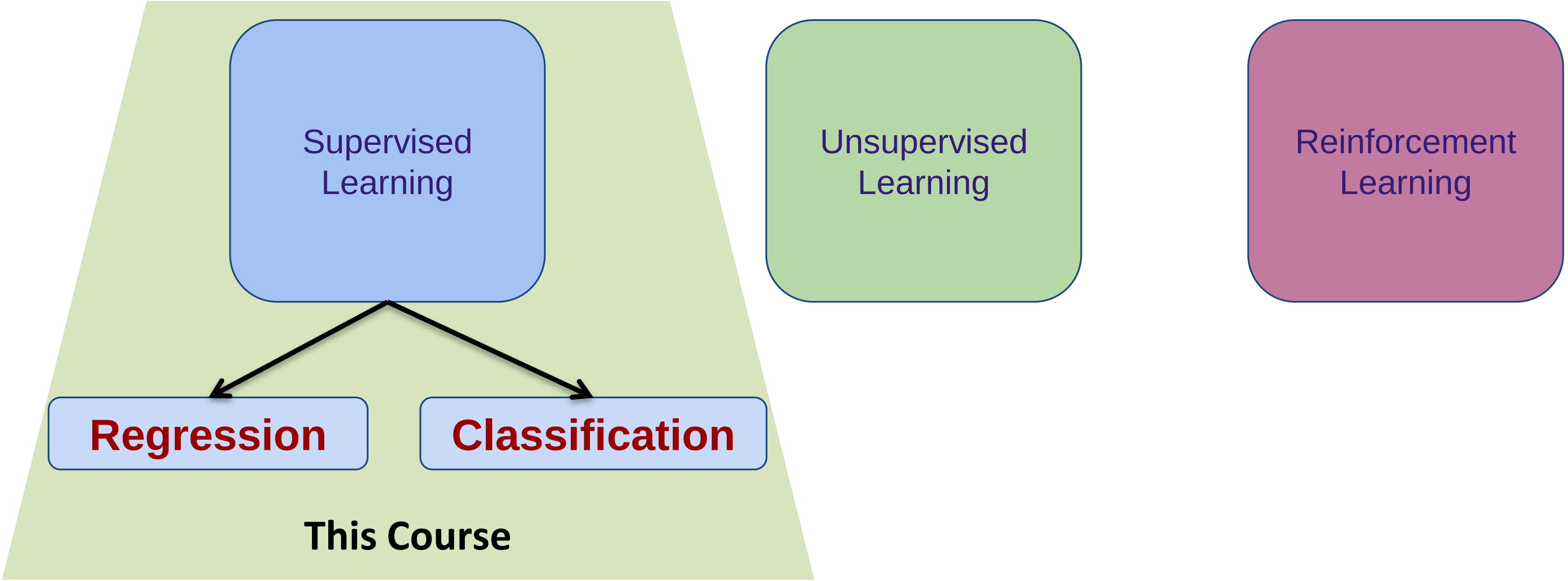
ECE
29/01/2019

Course overview

1. Classification: recall
2. Decision Trees
    1. Decision Trees Intuition
    2. Decision Trees Learning
    3. Decision Trees Prediction
3. Multiclass classification with Decision Trees
4. Overfitting in Decision Trees
5. Ensemble Methods
6. Practical work

Reminder of Machine Learning Types

- Machine learning tasks are typically classified into **three broad categories**

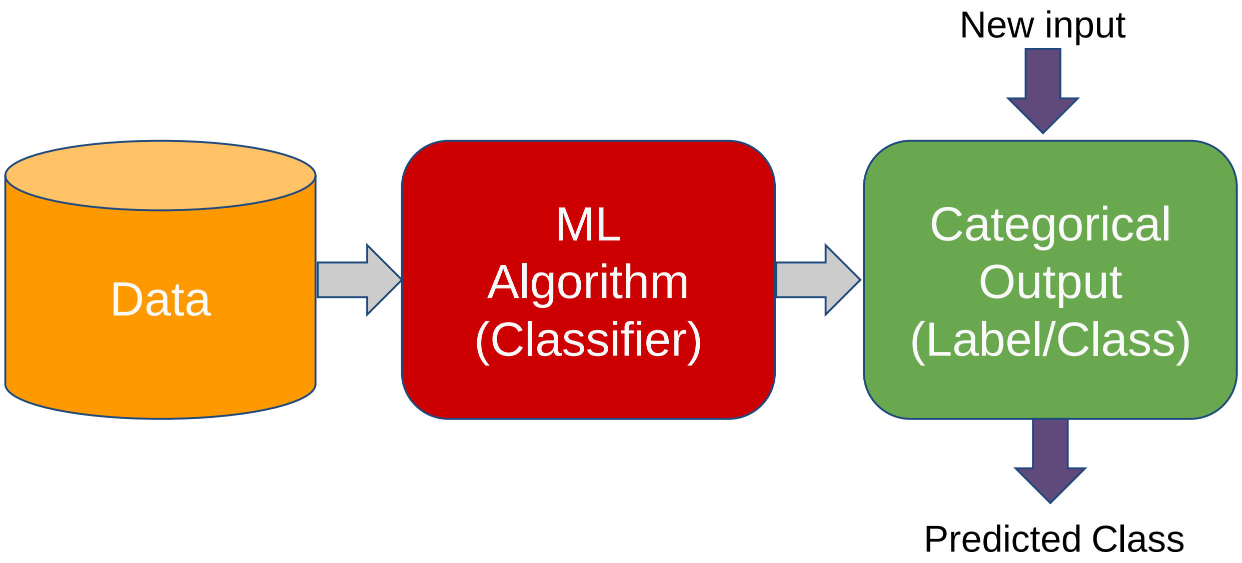


4.1 Classification: Recall

Classification

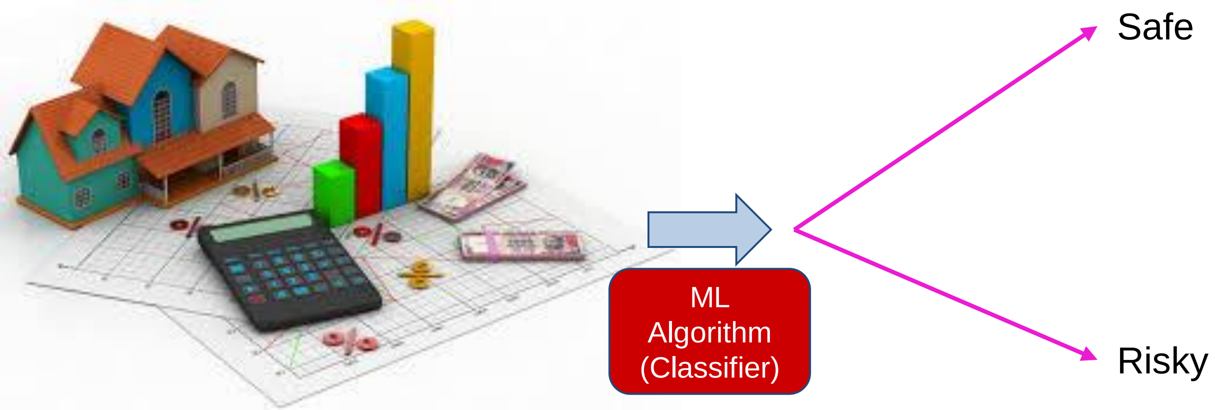
- **Goal:** Inputs are divided into two or more classes, and the ML algorithm must produce a model that assigns unseen inputs to one or more of these classes

- An algorithm that implements classification is known as a **classifier**



Two-class(Binary) Classification

- **Loan demand:** Output y has 2 categories



Input: x

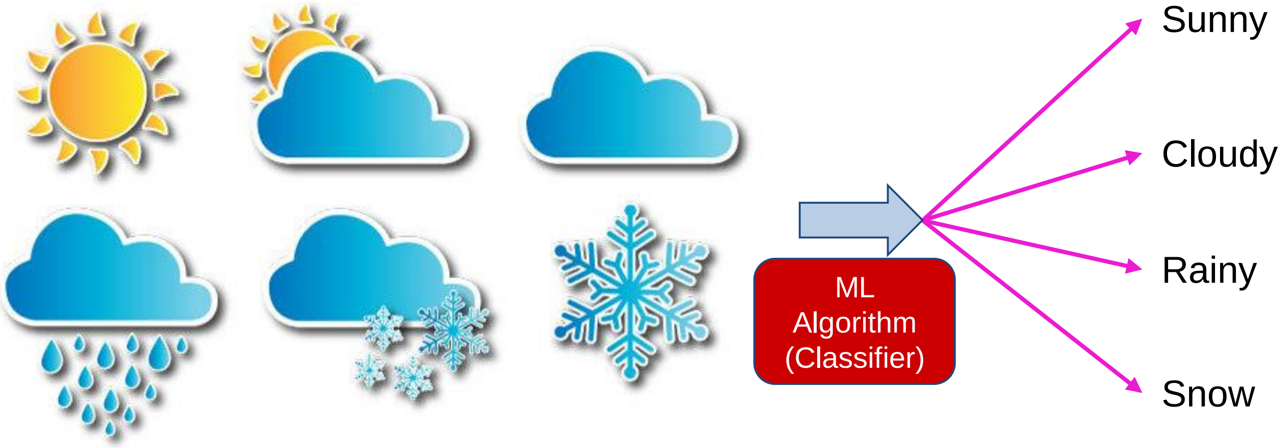
Client's characteristics
(age, Revenue, credit, etc..)

Output: y

Loan safety evaluation

Multi-class Classifier

- **Weather:** Output y has more than 2 categories



Input: x

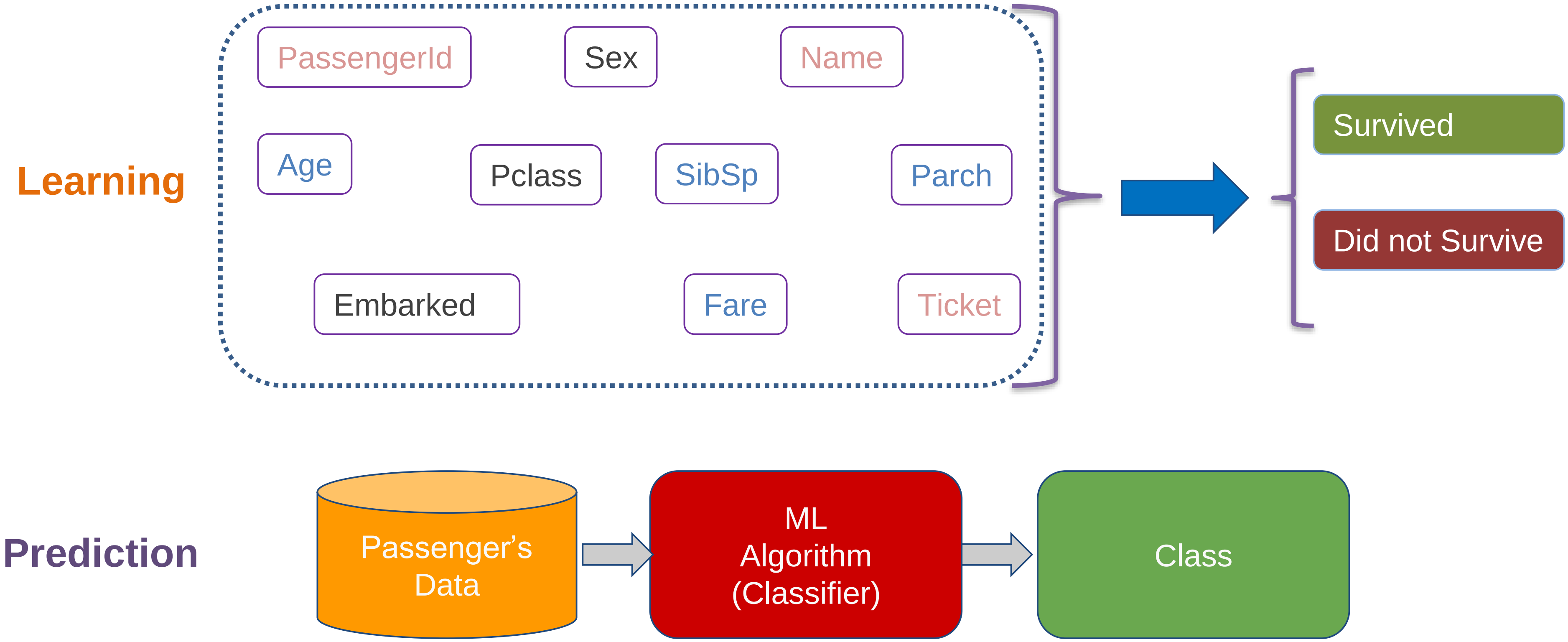
Altitude, region, date, etc…

Output: y

Weather status

Classification Example

- **Titanic survival prediction**: A Binary classification

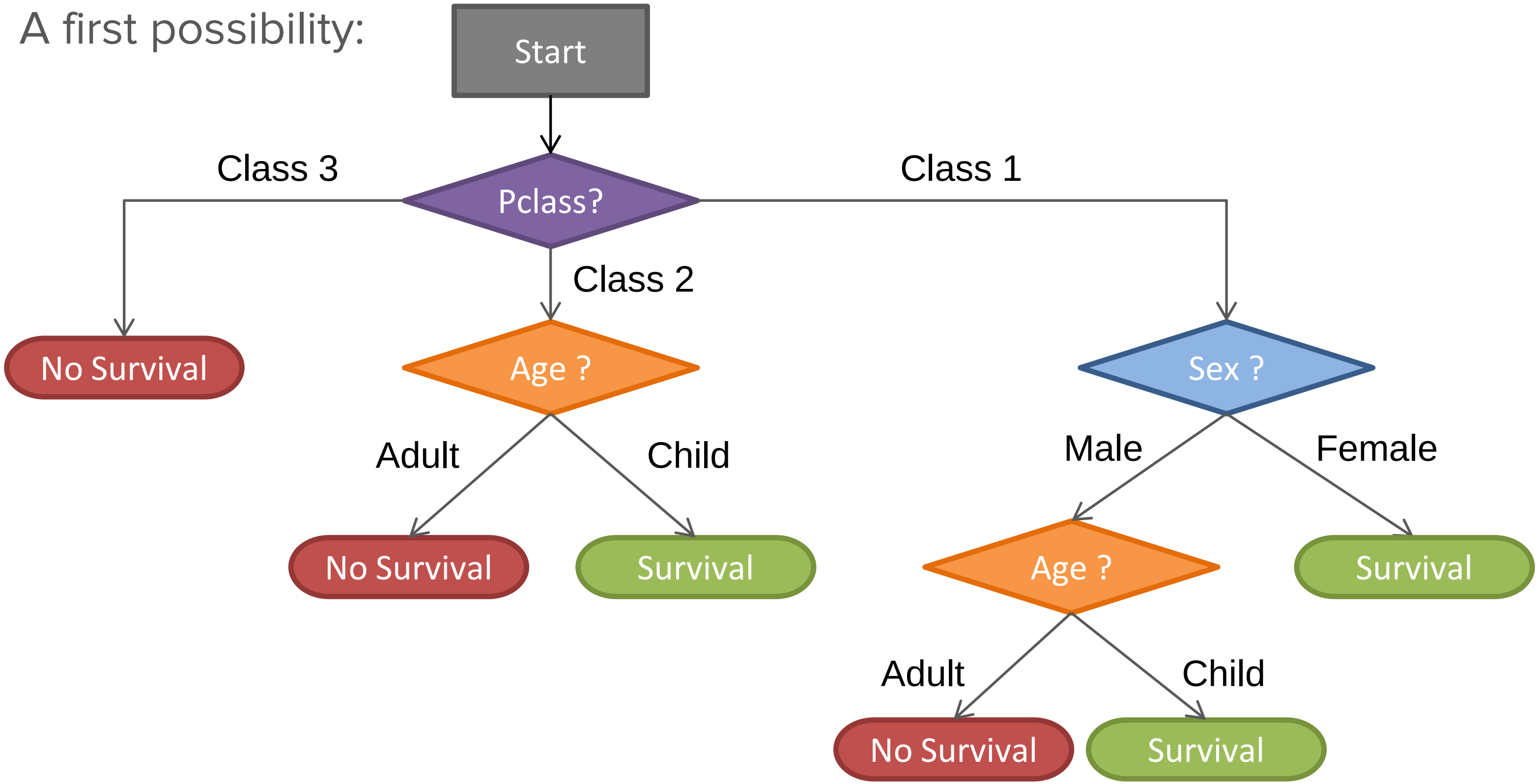


4.2
Decision Trees

4.2.1
Decision Trees: Intuition

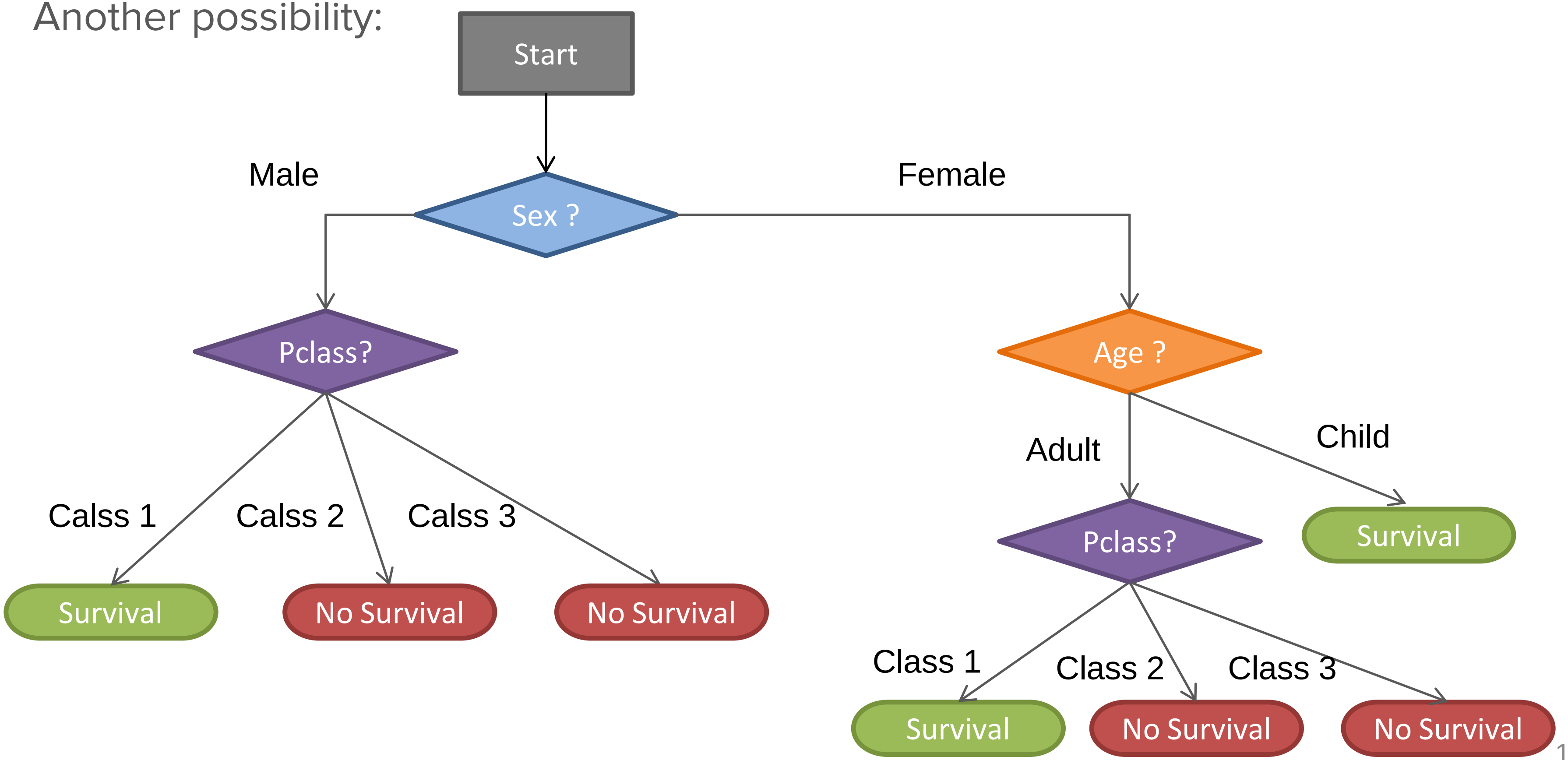
Decision Trees

- A first possibility:



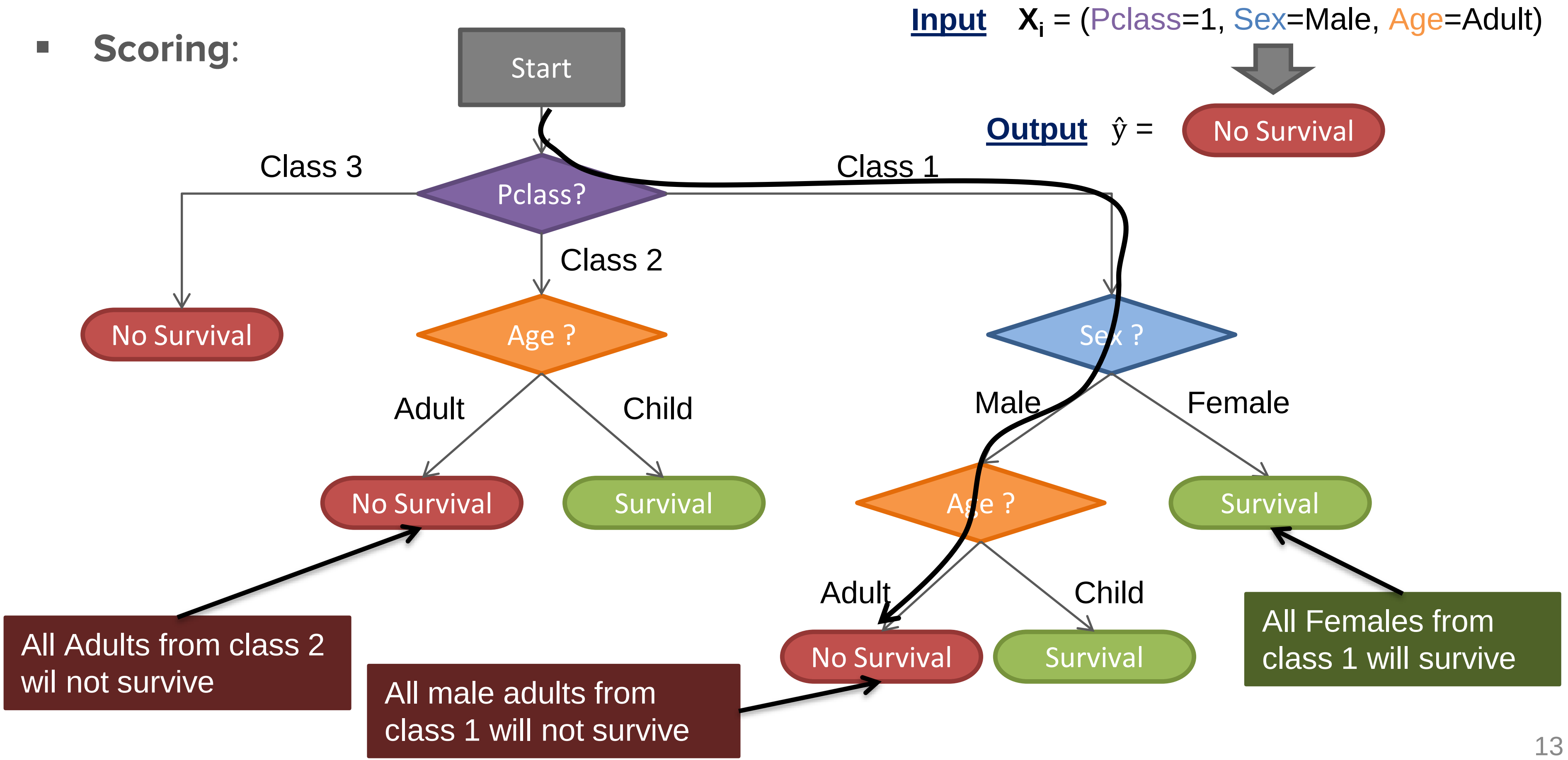
Decision Trees

- Another possibility:



Decision Trees: Intuition

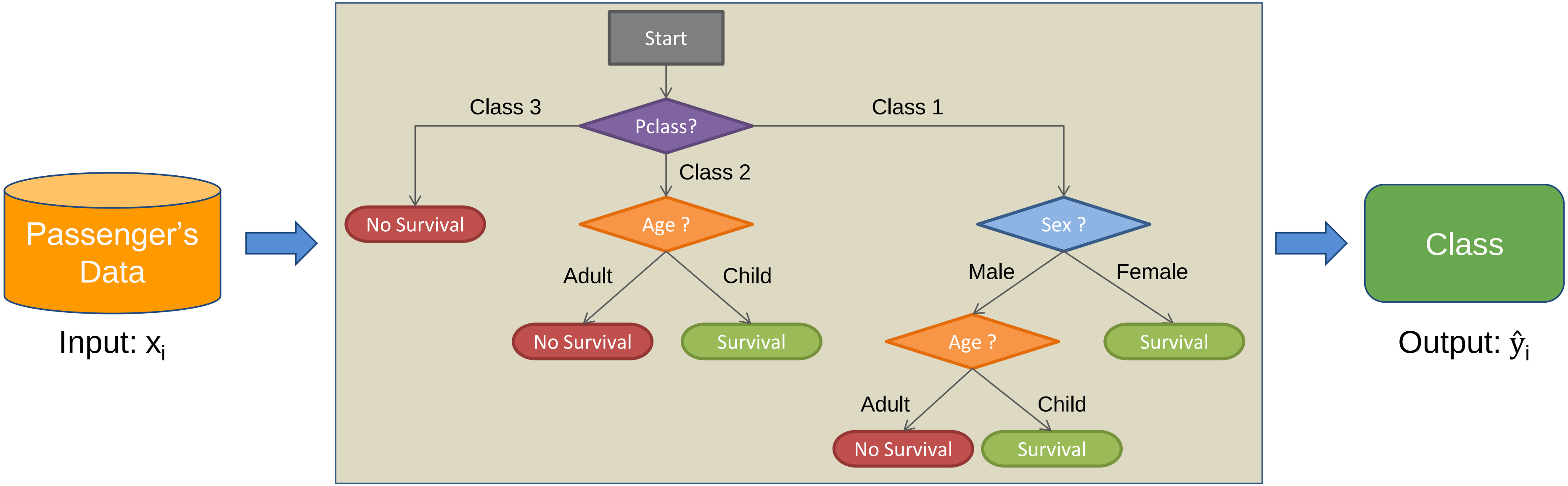
- **Scoring:**



Decision Trees: Model

- **Using a Decision Tree as a Classifier**:

$T(\mathbf{X}_i)$ = Traverse Decision Tree



Input: x_i

Output: $\hat{y}_i$

4.2.2
Decision Trees: Learning

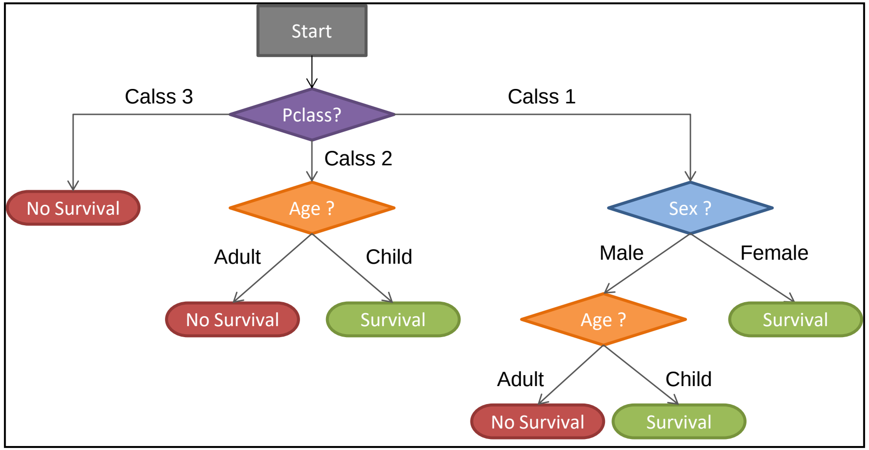
Decision Trees Learning

- **Learning a Decision Tree from Input Data:**

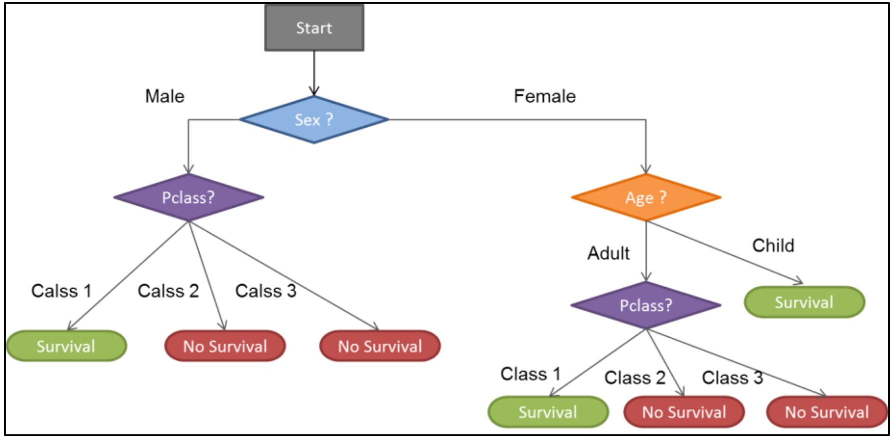
X			y

Pclass	Sex	Age	Survival?
3	male	Adult	No survival
1	female	Adult	Survival
3	female	Adult	No survival
1	female	Adult	Survival
3	male	Adult	No survival
1	male	Adult	No Survival
1	male	Child	survival
3	female	Adult	No survival
2	female	Child	survival
3	female	Child	No survival
1	female	Adult	Survival
3	male	Adult	No survival
3	male	Adult	No survival
3	female	Child	No survival
2	female	Adult	No survival
1	male	Child	survival
3	female	Adult	No survival
2	male	Adult	No survival
2	male	Adult	No survival

$T_1(\mathbf{X})$



$T_2(\mathbf{X})$



Other Trees

Decision Trees Learning

- **Learning a Decision Tree from Input Data:**

X | y

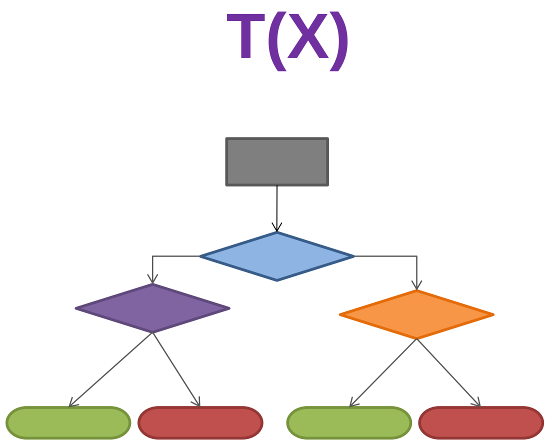
Pclass	Sex	Age	Survival?
3	male	Adult	No survival
1	female	Adult	Survival
3	female	Adult	No survival
1	female	Adult	Survival
3	male	Adult	No survival
1	male	Adult	No Survival
1	male	Child	survival
3	female	Adult	No survival
2	female	Child	survival
3	female	Child	No survival
1	female	Adult	Survival
3	male	Adult	No survival
3	male	Adult	No survival
3	female	Child	No survival
2	female	Adult	No survival
1	male	Child	survival
3	female	Adult	No survival
2	male	Adult	No survival
2	male	Adult	No survival

Training data

(x_i,y_i)

?

By optimizing the **quality metric** on training data



Decision Trees Learning: Classification Error

- Quality metric = **Classification Error**: measures the fraction of mistakes

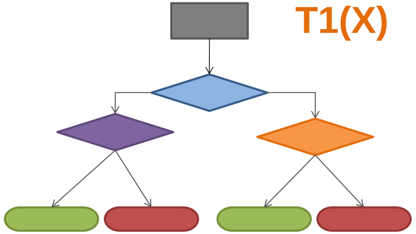
X			y	T1(X)	T2(X)
Pclass	Sex	Age	Survival?	Survival?	Survival?
3	male	Adult	No survival	No survival	No survival
1	female	Adult	Survival	No survival	Survival
3	female	Adult	No survival	No survival	No survival
1	female	Adult	Survival	Survival	Survival
3	male	Adult	No survival	No survival	No survival
1	male	Adult	No Survival	No Survival	No Survival
1	male	Child	Survival	Survival	Survival
3	female	Adult	No survival	No survival	No survival
2	female	Child	Survival	Survival	Survival
3	female	Child	No survival	No survival	No survival
1	female	Adult	Survival	Survival	Survival
3	male	Adult	No survival	Survival	No survival
3	male	Adult	No survival	Survival	No survival
3	female	Child	No survival	No survival	No survival
2	female	Adult	No survival	No survival	No survival
1	male	Child	Survival	Survival	Survival
3	female	Adult	No survival	No survival	Survival
2	male	Adult	No survival	Survival	No survival
2	male	Adult	No survival	No survival	No survival

$$\textbf{ERROR} = \frac{\text{Nb of incorrect predictions}}{\text{Total Nb of samples}}$$

- Best possible value = 0
- Worst possible value = 1

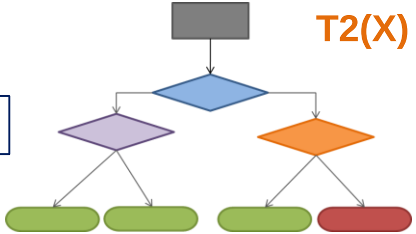
T1(X)

ERROR = 4 / 20 = 0,2



T2(X)

ERROR = 1 / 20 = 0,05



Decision Trees Learning

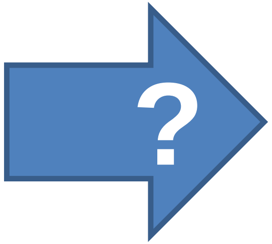
- **Learning a Decision Tree from Input Data: Find the tree with lowest error !!**

X y

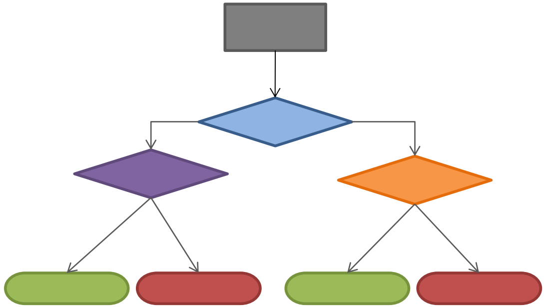
Pclass	Sex	Age	Survival?
3	male	Adult	No survival
1	female	Adult	Survival
3	female	Adult	No survival
1	female	Adult	Survival
3	male	Adult	No survival
1	male	Adult	No Survival
1	male	Child	survival
3	female	Adult	No survival
2	female	Child	survival
3	female	Child	No survival
1	female	Adult	Survival
3	male	Adult	No survival
3	male	Adult	No survival
3	female	Child	No survival
2	female	Adult	No survival
1	male	Child	survival
3	female	Adult	No survival
2	male	Adult	No survival
2	male	Adult	No survival

Training data

(x_i, y_i)



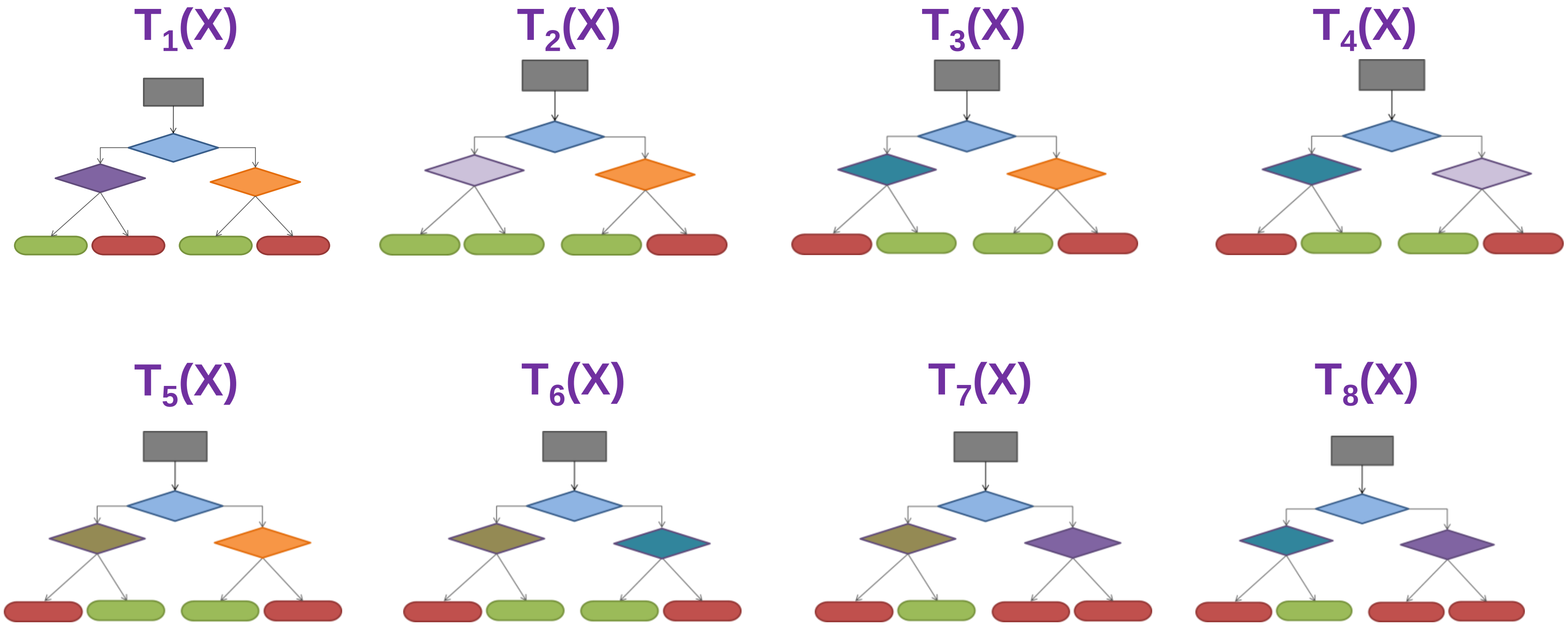
T(X)



By optimizing the **quality metric** on training data **=** By minimizing the **classification error** on training data

How to find the best tree ?

- How to **find the tree with lowest error** ?
  - Exponentially Large Number of possible trees ➔ making decision tree learning hard



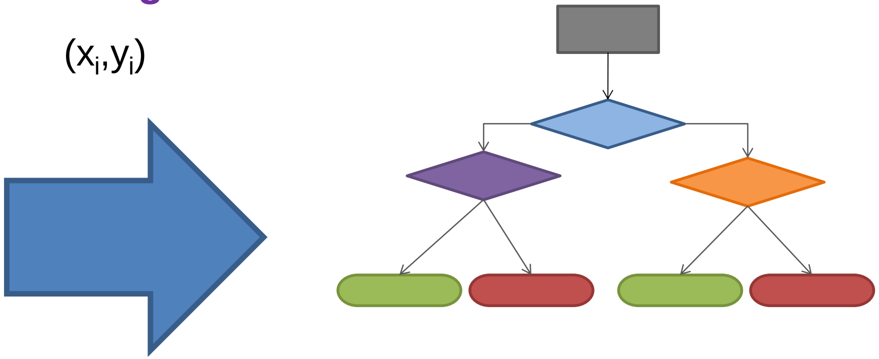
How to find the best tree ?

- **Simple (greedy) algorithm: Finds a « Good » tree**

Pclass	Sex	Age	Survival?
3	male	Adult	No survival
1	female	Adult	Survival
3	female	Adult	No survival
1	female	Adult	Survival
3	male	Adult	No survival
1	male	Adult	No Survival
1	male	Child	survival
3	female	Adult	No survival
2	female	Child	survival
3	female	Child	No survival
1	female	Adult	Survival
3	male	Adult	No survival
3	male	Adult	No survival
3	female	Child	No survival
2	female	Adult	No survival
1	male	Child	survival
3	female	Adult	No survival
2	male	Adult	No survival
2	male	Adult	No survival

Training data

(x_i,y_i)



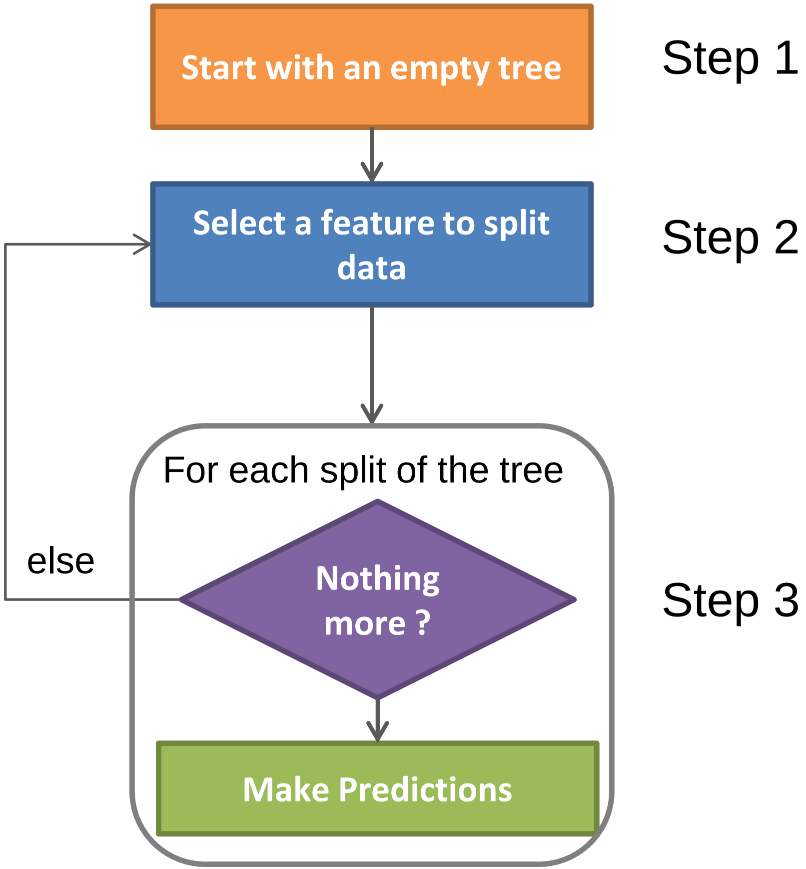
T(X)

Approximately minimize the **classification error** on training data

Greedy Decision Tree Learning

Greedy Decision Tree Learning

- **Algorithm:**



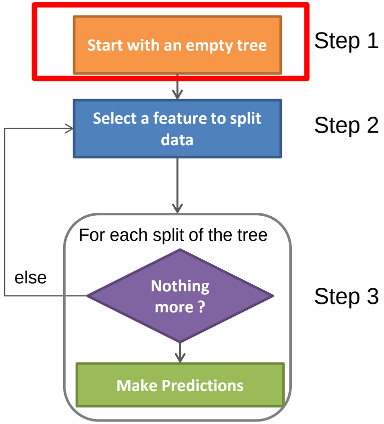
Greedy DT Learning: step 1

- **Start with all Data: Root Node**

Pclass	Sex	Age	Survival?
3	male	Adult	No survival
1	female	Adult	Survival
3	female	Adult	No survival
1	female	Adult	Survival
3	male	Adult	No survival
1	male	Adult	No Survival
1	male	Child	survival
3	female	Adult	No survival
2	female	Child	survival
3	female	Child	No survival
1	female	Adult	Survival
3	male	Adult	No survival
3	male	Adult	No survival
3	female	Child	No survival
2	female	Adult	No survival
1	male	Child	survival
3	female	Adult	No survival
2	male	Adult	No survival
2	male	Adult	No survival

Assume N = 120 & 3 features

Survival status : Survival or No survival



Root node

66	54

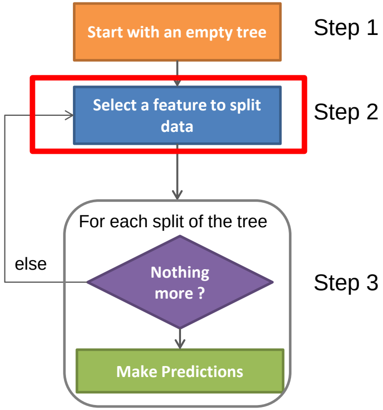
Nb of passengers that **didn't survive**

Nb of passengers that **survived**

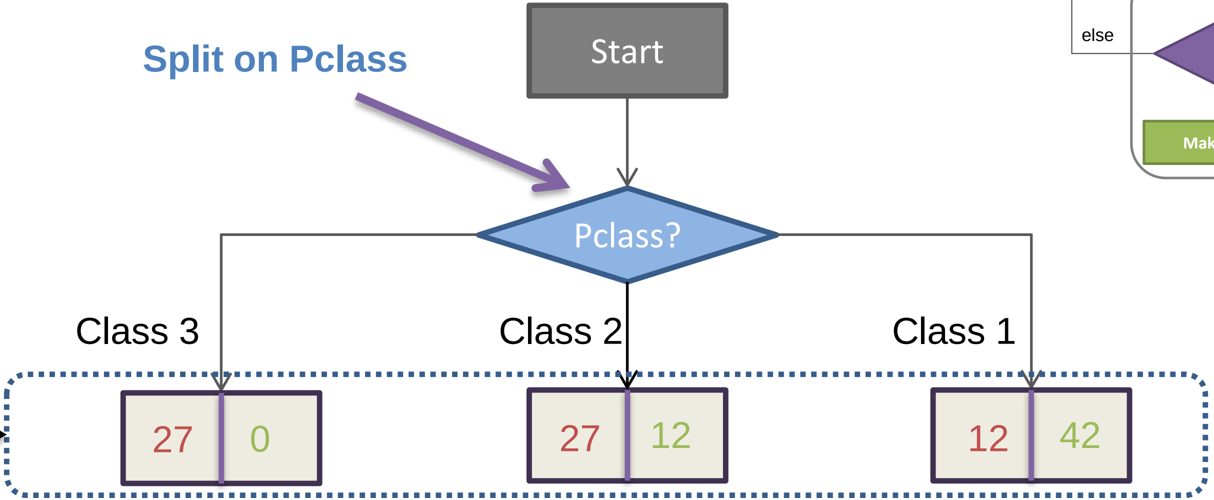
Greedy DT Learning: step 2

- Decision stump: **Single level Tree**

Survival status :
Survival or No survival



Intermediate nodes



Subset of data with Pclass= 3

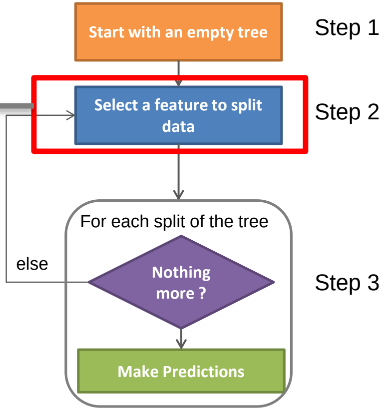
Subset of data with Pclass= 2

Subset of data with Pclass= 1

Greedy DT Learning: step 2

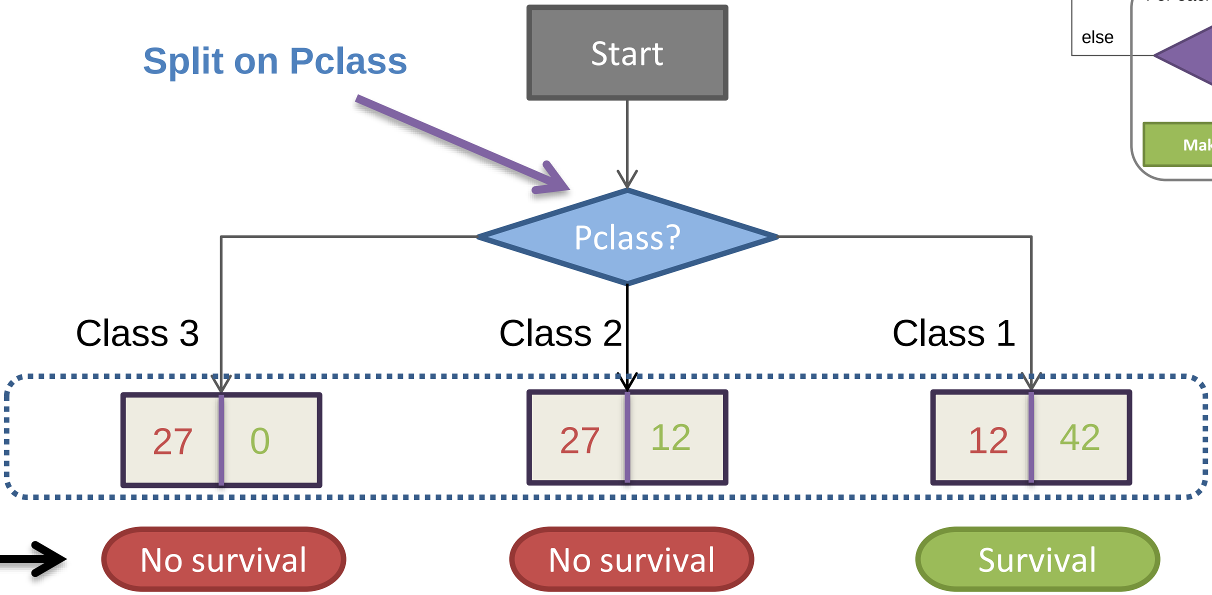
- **Making predictions with a decision stump**

How??



Survival status :
Survival or No survival

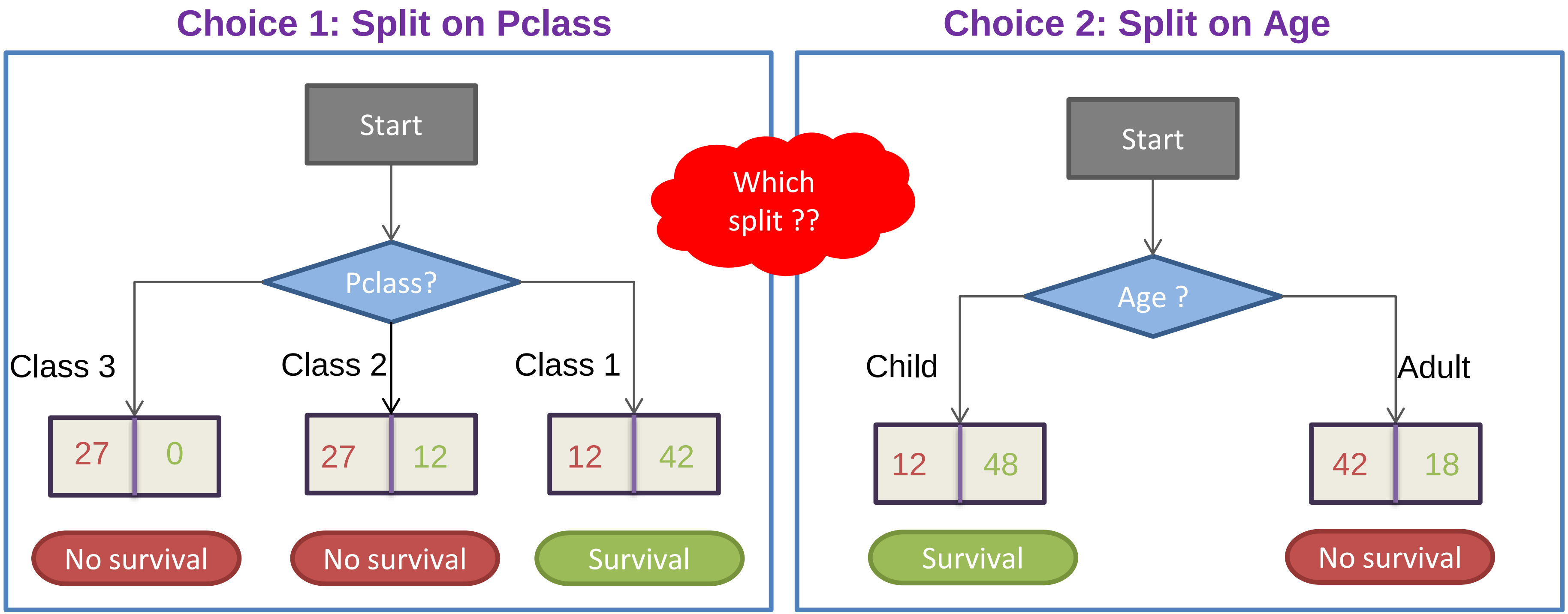
Split on Pclass



For each intermediate node, set
ŷ = majority value

Greedy DT Learning: step 2

- **Selecting best feature to split on**

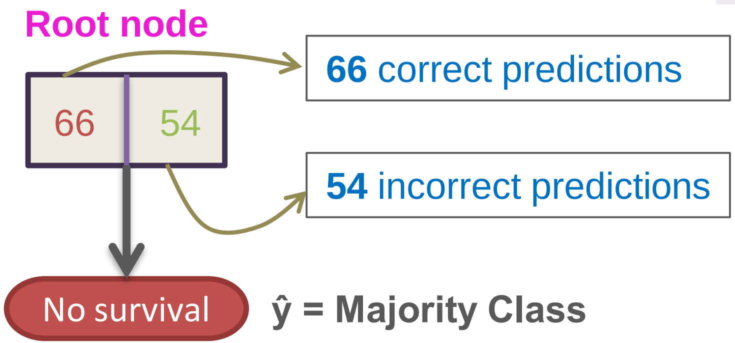


Greedy DT Learning: step 2

- Selecting best feature to split on: **Measuring effectiveness of a split**
  - By calculating the **classification error** of the actual decision stump !
    - Step 1: ŷ = Class of majority of data in node
    - Step 2: Calculate the classification error of predicting ŷ for that data

$$\text{Error} = \frac{\text{Nb of incorrect predictions}}{\text{Total Nb of samples}}$$

Tree	Classification error
Root node	0,45

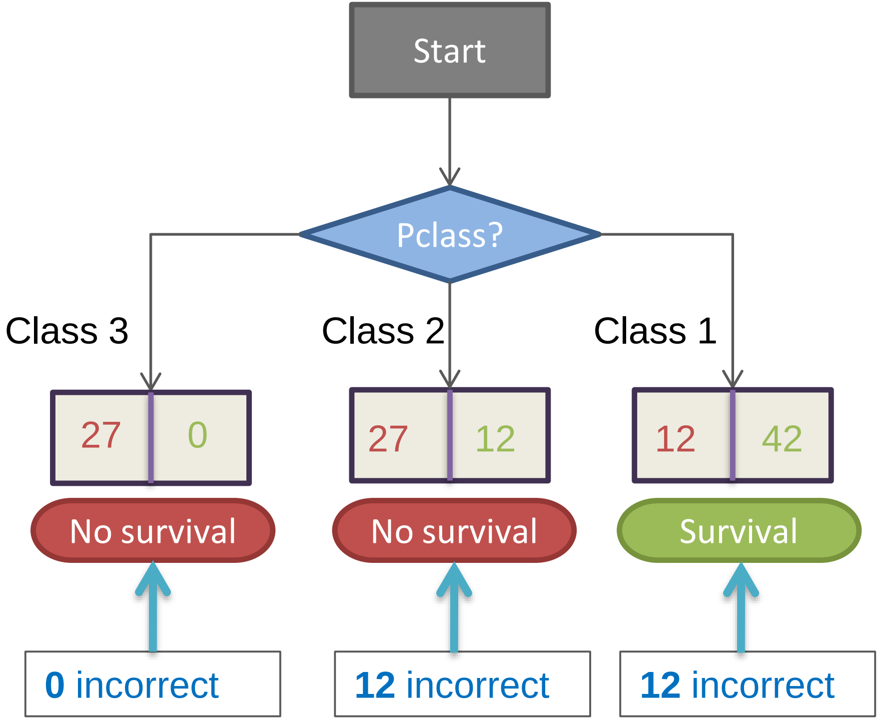


ERROR = 54 / 120 = 0,45

Greedy DT Learning: step 2

- Selecting best feature to split on: **Measuring effectiveness of a split**

Choice 1: Split on Pclass



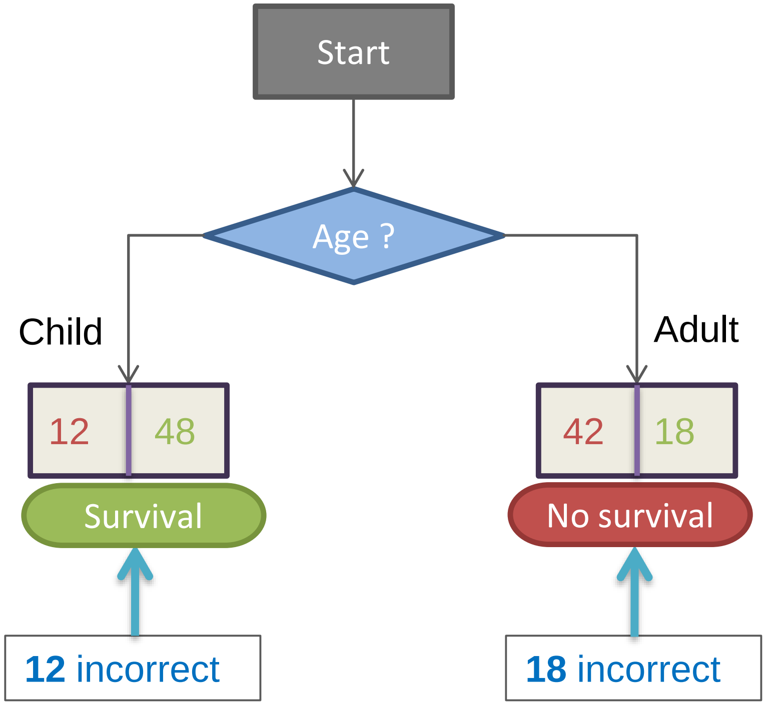
Tree	Classification error
Root node	0,45
Split on Pclass	0,2

ERROR = 24 / 120 = 0,2

Greedy DT Learning: step 2

- Selecting best feature to split on: **Measuring effectiveness of a split**

Choice 2: Split on Age



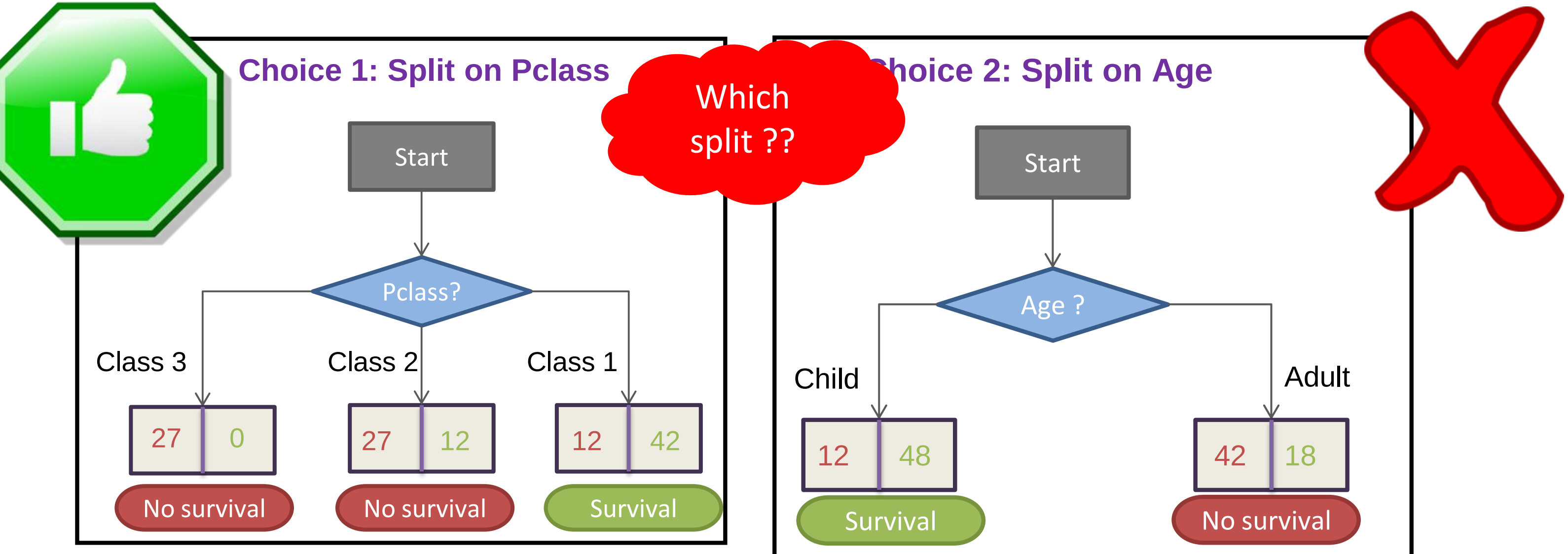
Tree	Classification error
Root node	0,45
Split on Pclass	0,2
Split on Age	0,25

ERROR = 30 / 120 = 0,25

Greedy DT Learning: step 2

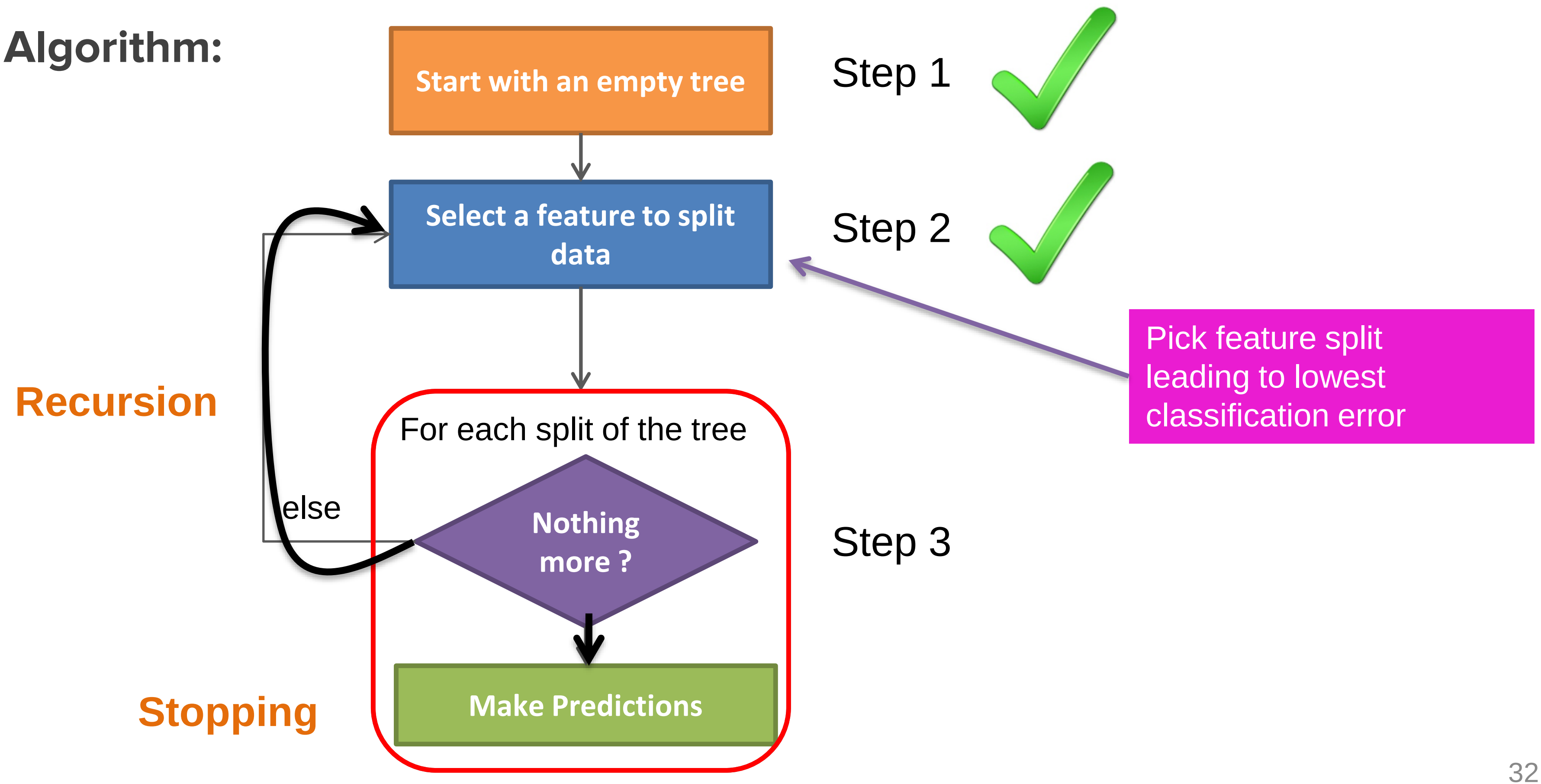
- **Selecting best feature to split on**

Tree	**Classification error**
Root node	0,45
Split on Pclass	0,2
Split on Age	0,25



Greedy Decision Tree Learning

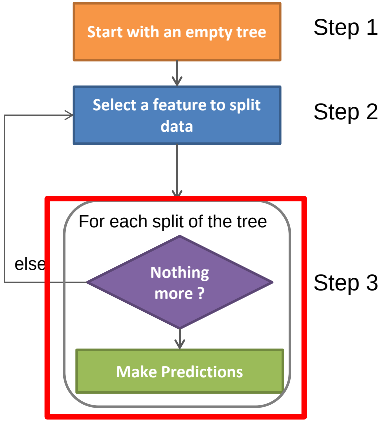
- **Algorithm:**

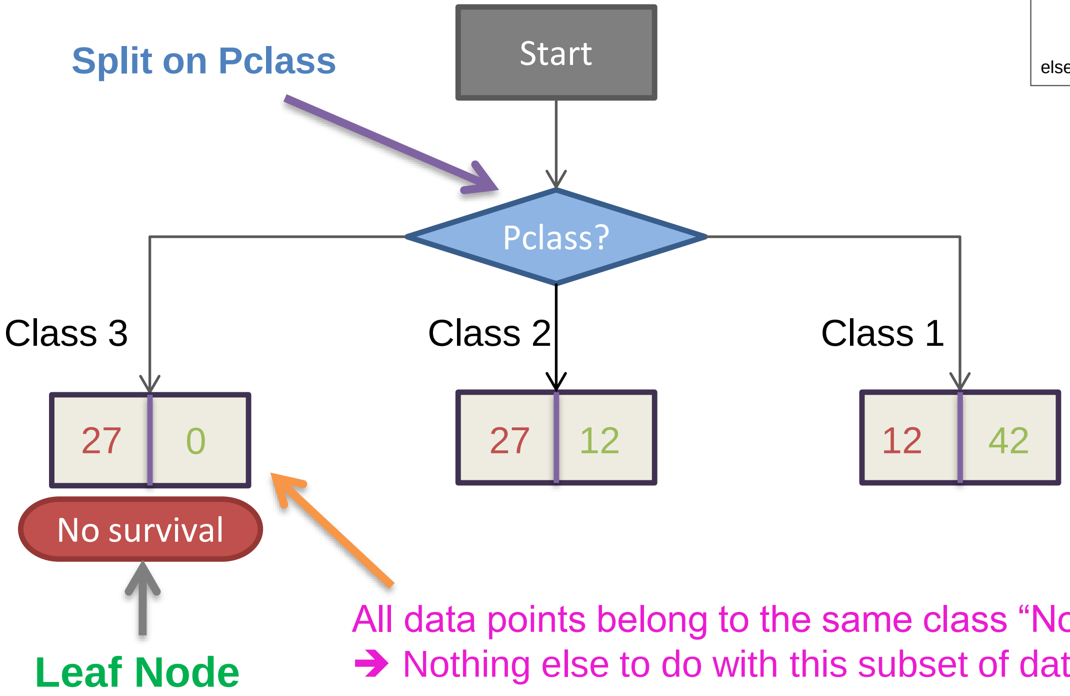


Greedy DT Learning: step 3

- **Recursion & Stopping:**
  - We have learned a decision stump: what's next ?

Survival status :
Survival or No survival

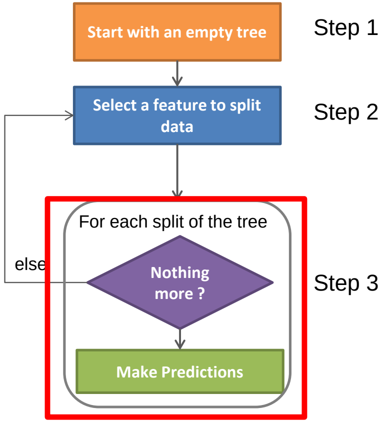




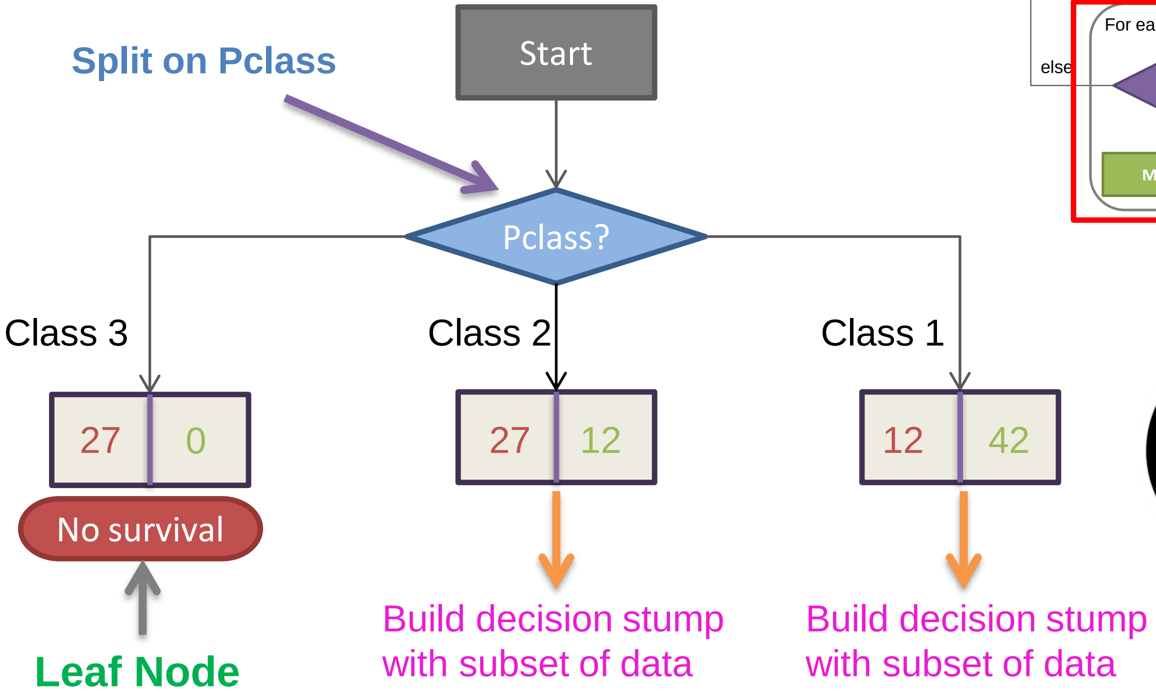
All data points belong to the same class "No survival"
➔ Nothing else to do with this subset of data

Greedy DT Learning: step 3

- **Recursion & Stopping**
  - We have learned a decision stump: what’s next ?

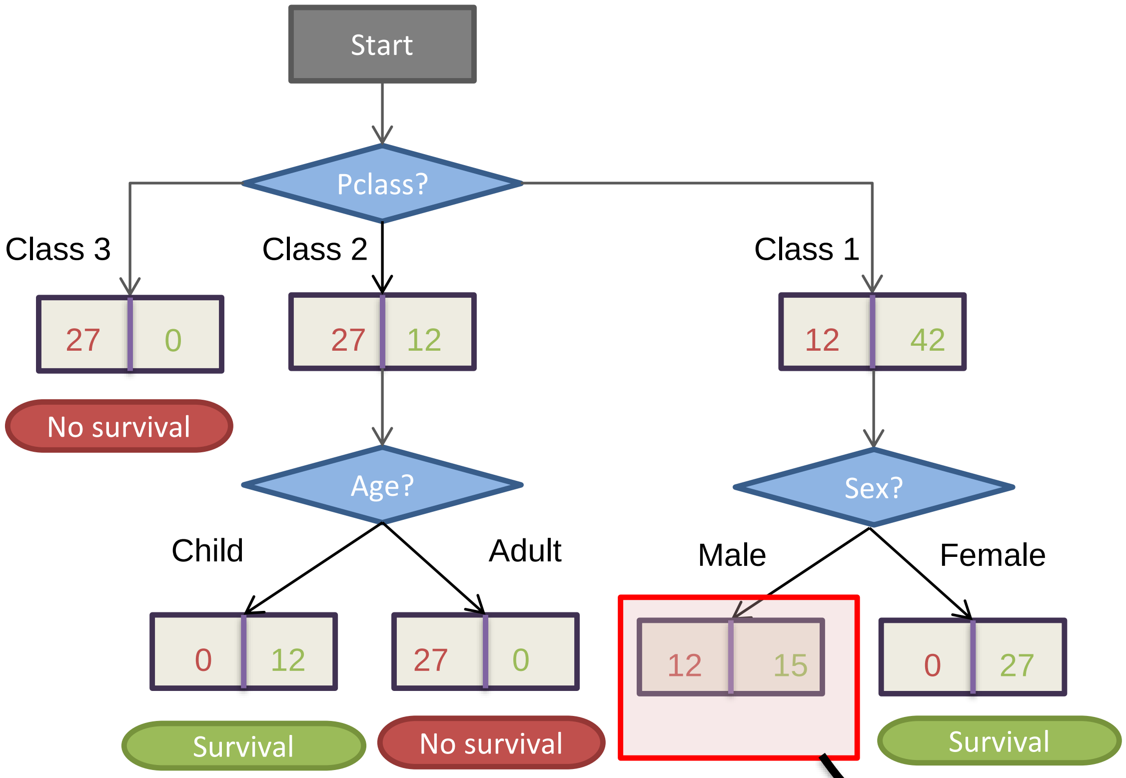


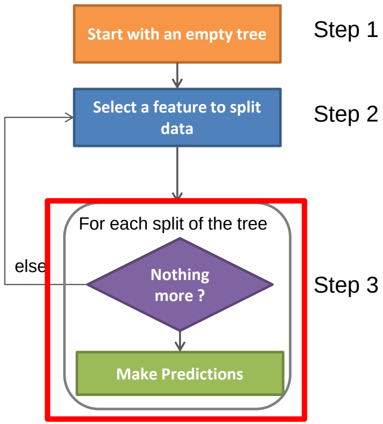
Survival status :
Survival or No survival



Greedy DT Learning: step 3

- **Recursion & Stopping: Second Level**

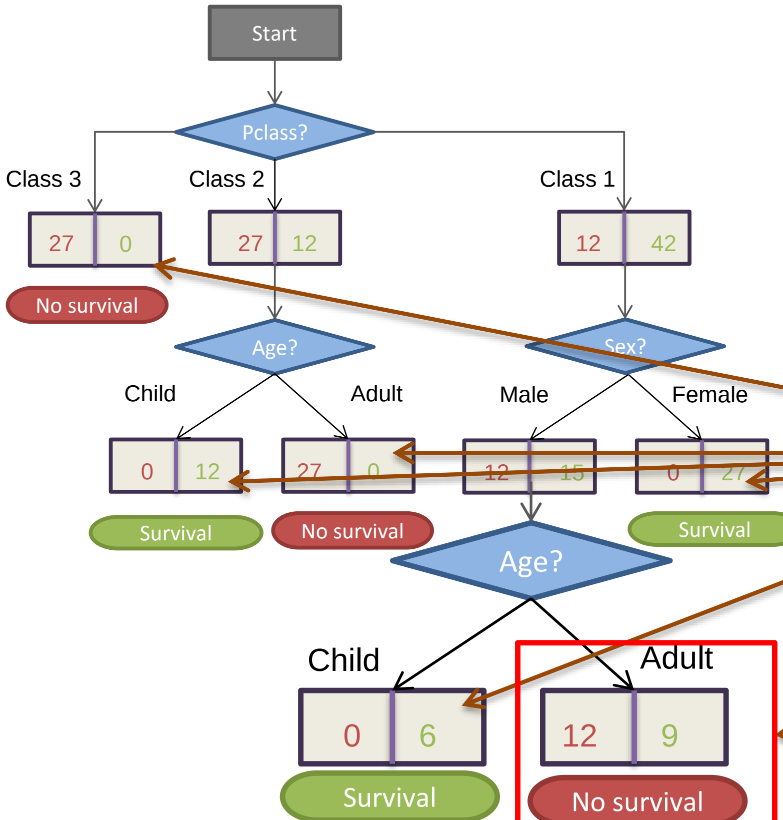




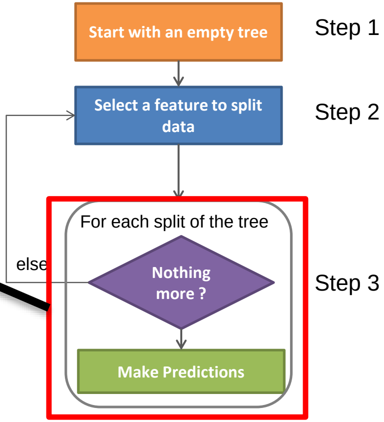
Build another decision stump for these data points

Greedy DT Learning: step 3

- **Recursion & Stopping: Final Decision Tree**



When do we STOP ???

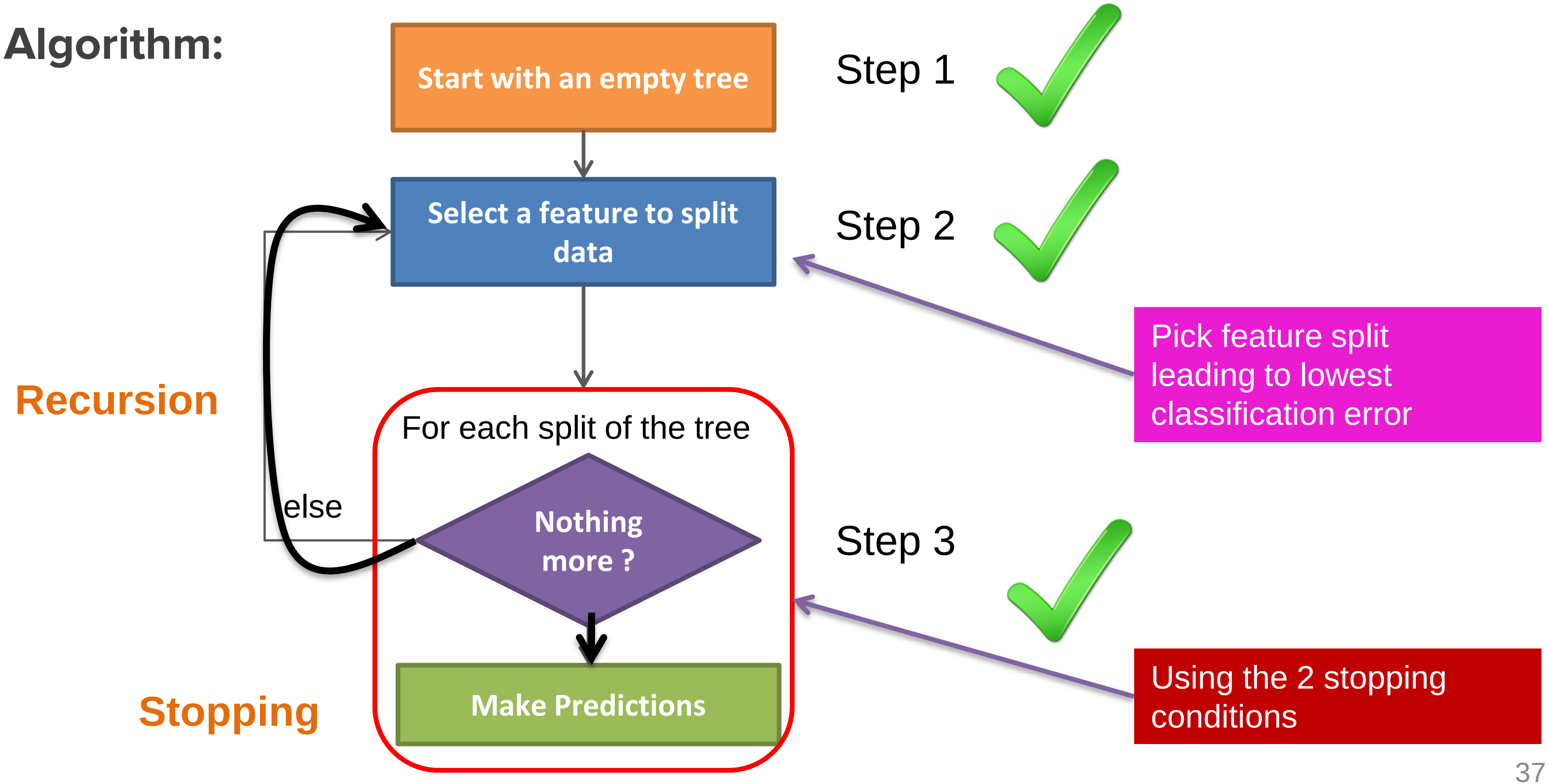


All data in these nodes have the **same class** !!
➔ No more splitting
➔ **STOP**

Already split on **all other features** !!
➔ No more splitting
➔ **STOP**

Greedy Decision Tree Learning

- **Algorithm:**

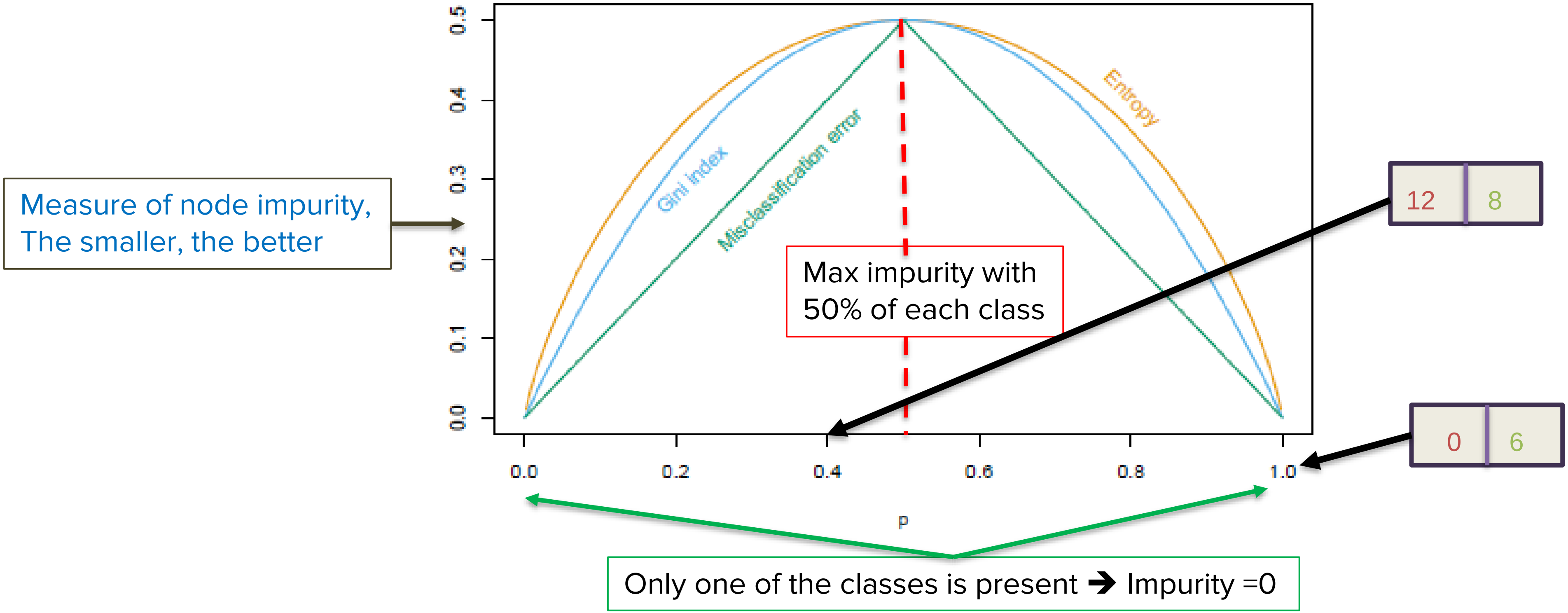


Measuring the effectiveness of a split

- So far, we have used the **Classification error** to choose the best split.
- Two other measures are also possible: **Gini index** and **Entropy**
- These are all measures of node impurity that we want to minimize
- For two classes, if p is the proportion in the second class, these measures are
  - $Classification\ error = 1 - \max(p, 1 - p)$
  - $Gini\ Index = 2p(1 - p)$
  - $Entropy = -plog(p) - (1 - p)\log(1 - p)$
- Gini index and Entropy are more used in practice (differentiable)

Measuring the effectiveness of a split

- P= Fraction of one of the two classes (Survival)



Credit: The Elements of Statistical Learning

Decision Trees Learning: Features with real values

Features with real values

- **How to deal with real valued features ?**

Pclass	Sex	Age	Survival?
3	male	22	No survival
1	female	38	Survival
3	female	26	No survival
1	female	35	Survival
3	male	35	No survival
1	male	54	No Survival
1	male	2	survival
3	female	27	No survival
2	female	14	survival
3	female	4	No survival
1	female	58	Survival
3	male	20	No survival
3	male	39	No survival
3	female	14	No survival
2	female	55	No survival
1	male	2	survival
3	female	31	No survival
2	male	35	No survival
2	male	34	No survival

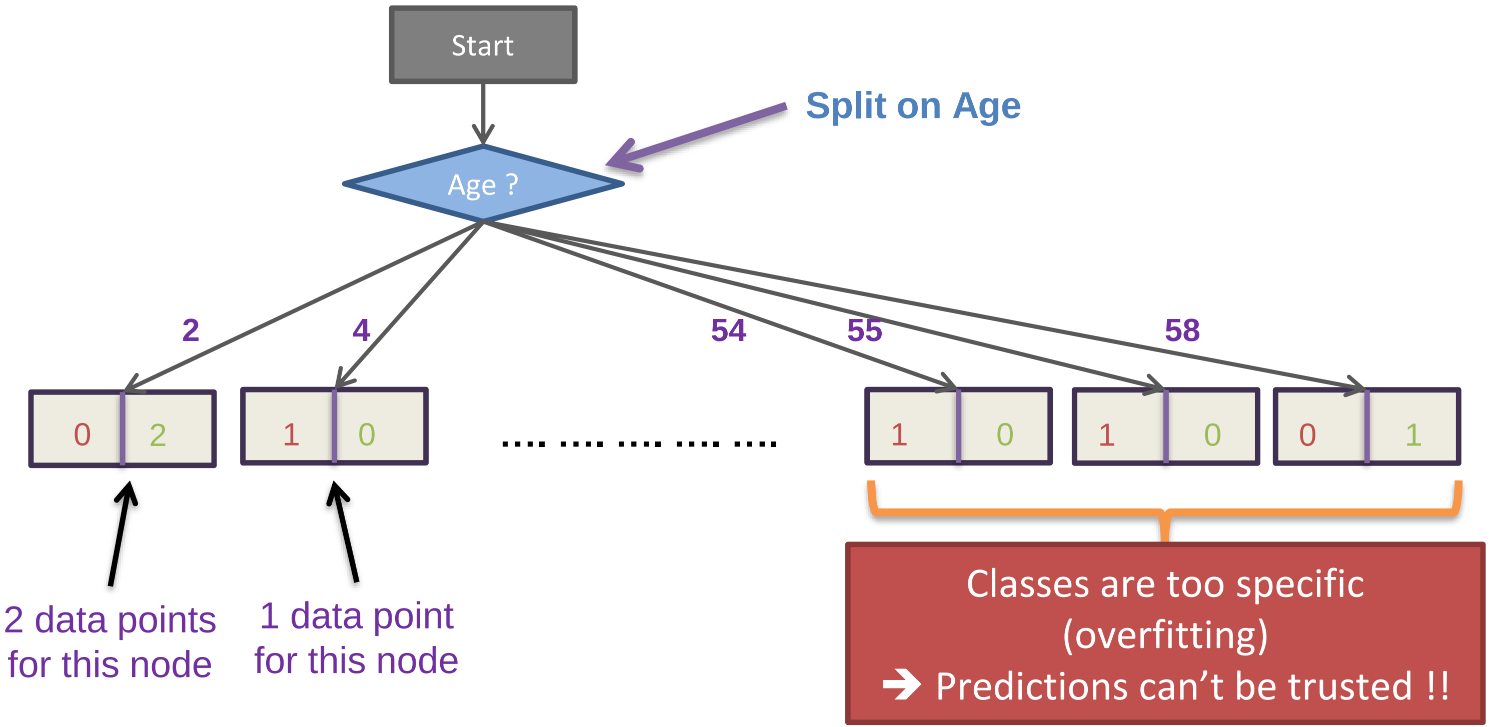
Survival status : **Survival or No survival**

The Age feature has real values (not categorical)

Features with real values

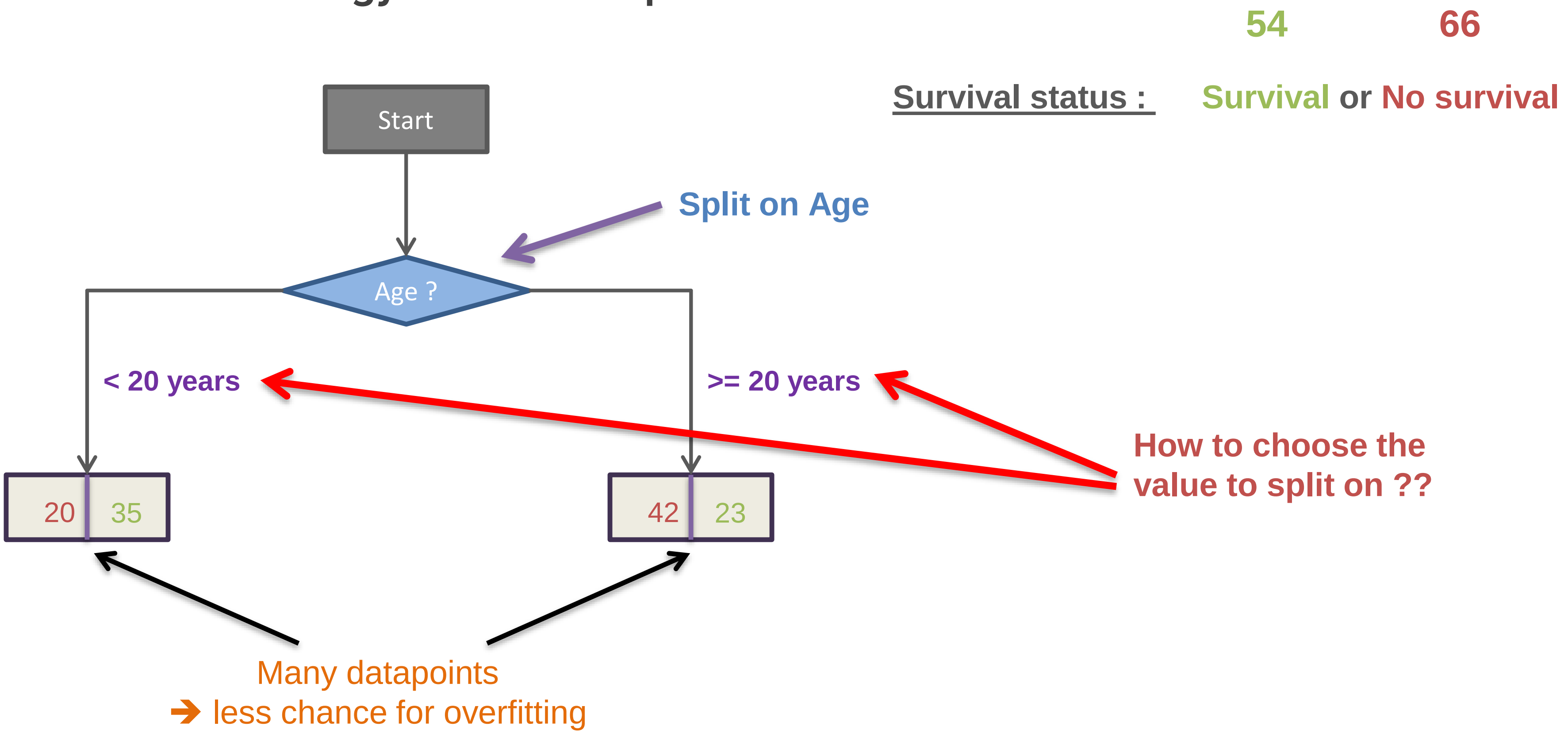
- **Split on each numeric value ?**

Survival status : **Survival** or **No survival**



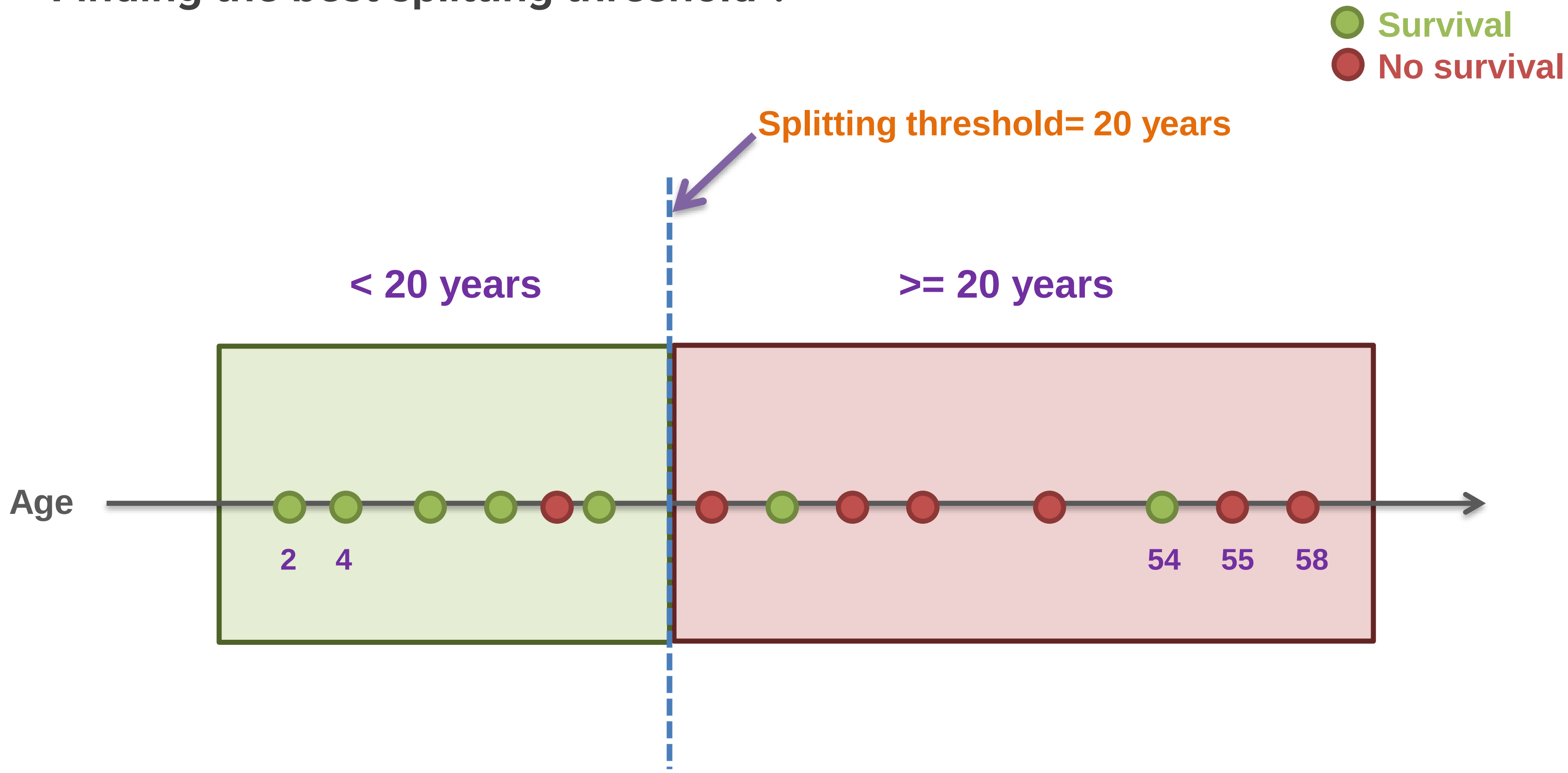
Features with real values

- **A better strategy: Threshold split**



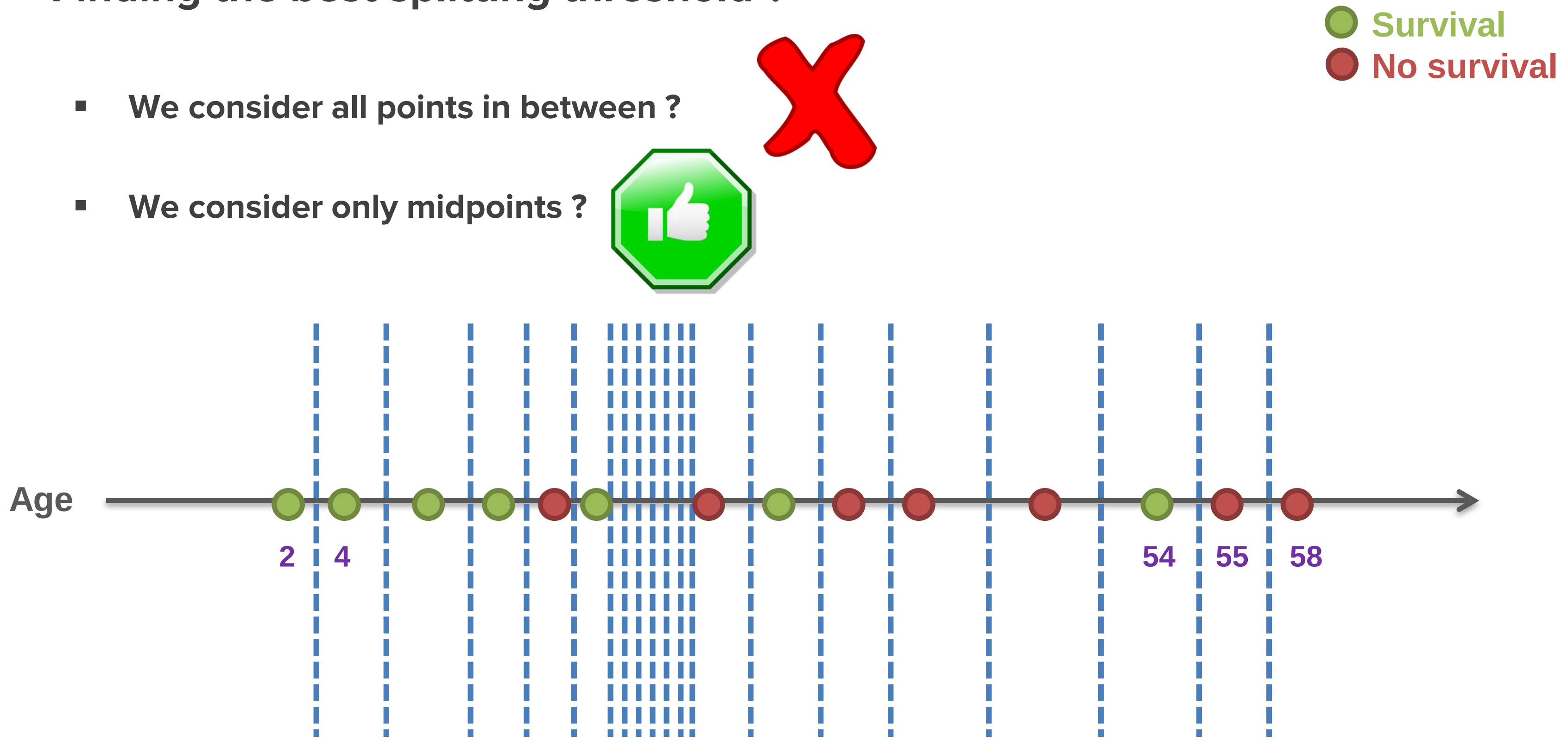
Features with real values

- **Finding the best splitting threshold ?**



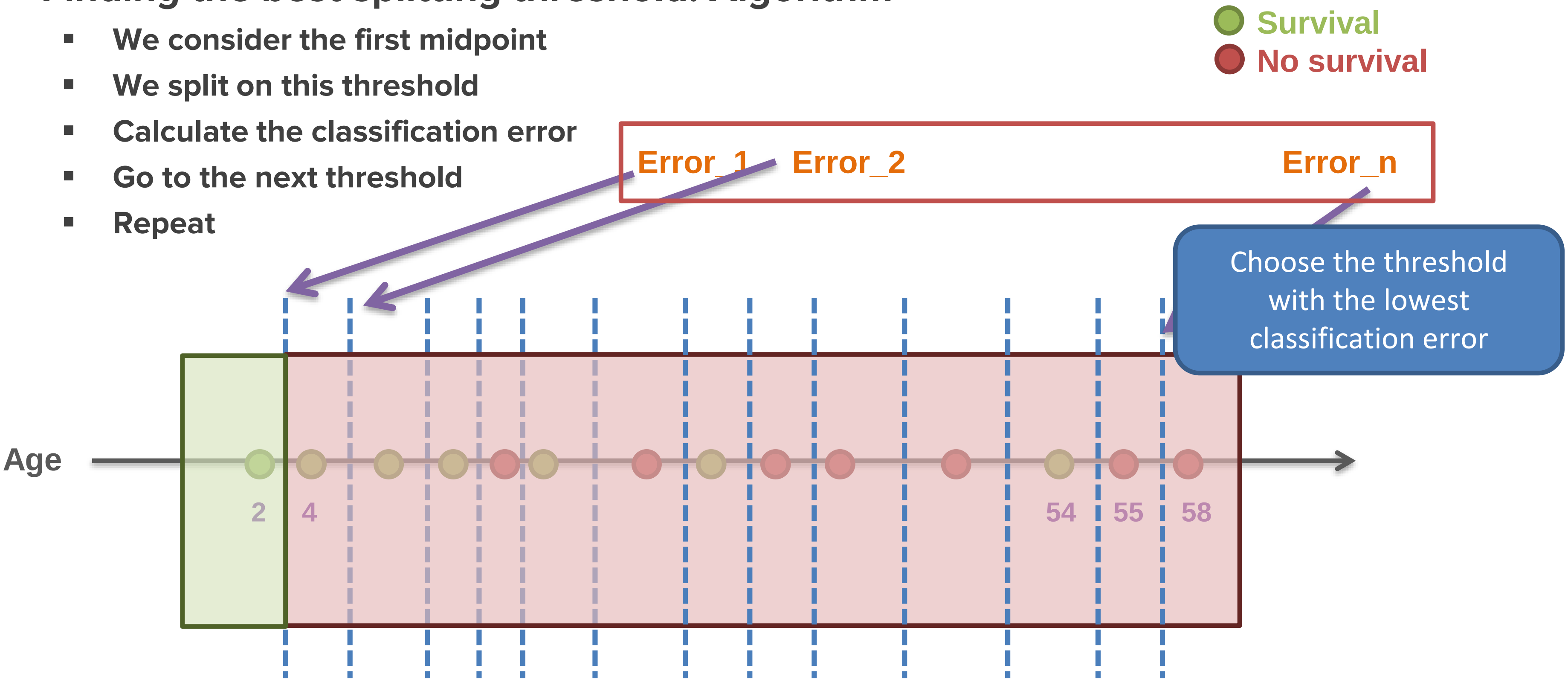
Features with real values

- **Finding the best splitting threshold ?**
  - **We consider all points in between ?**
  - **We consider only midpoints ?**



Features with real values

- **Finding the best splitting threshold: Algorithm**
  - We consider the first midpoint
  - We split on this threshold
  - Calculate the classification error
  - Go to the next threshold
  - Repeat

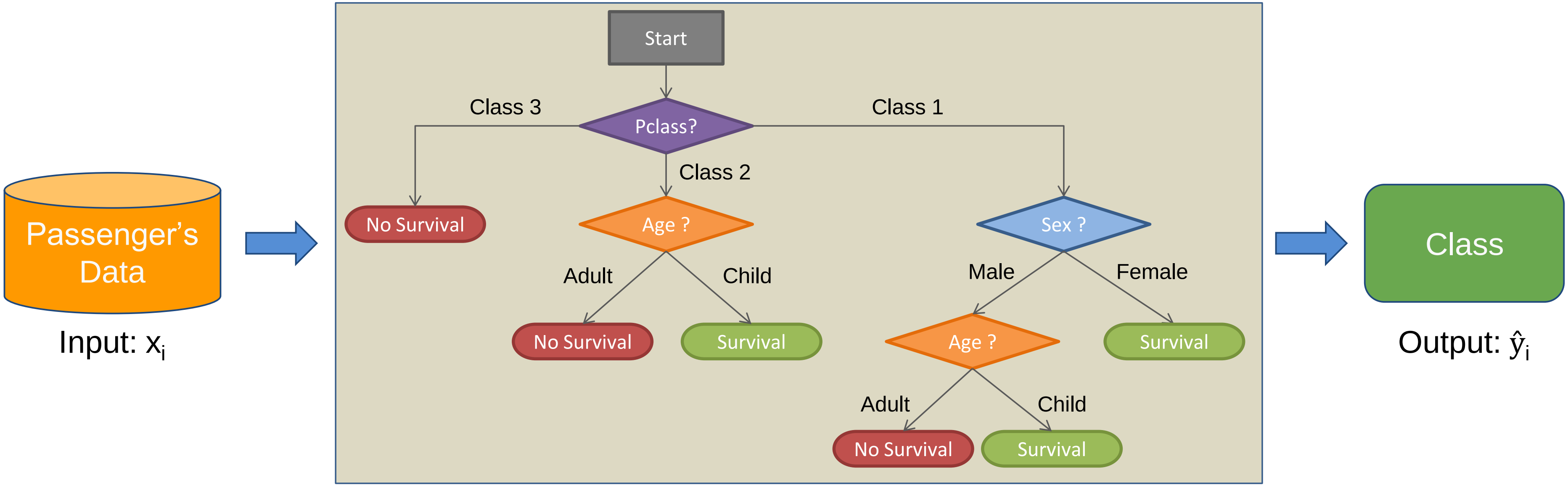


4.2.3
Prediction with Decision Trees

Decision Trees Prediction

- **Using a Decision Tree as a Classifier:**

$T(\mathbf{X}_i)$ = Traverse Decision Tree

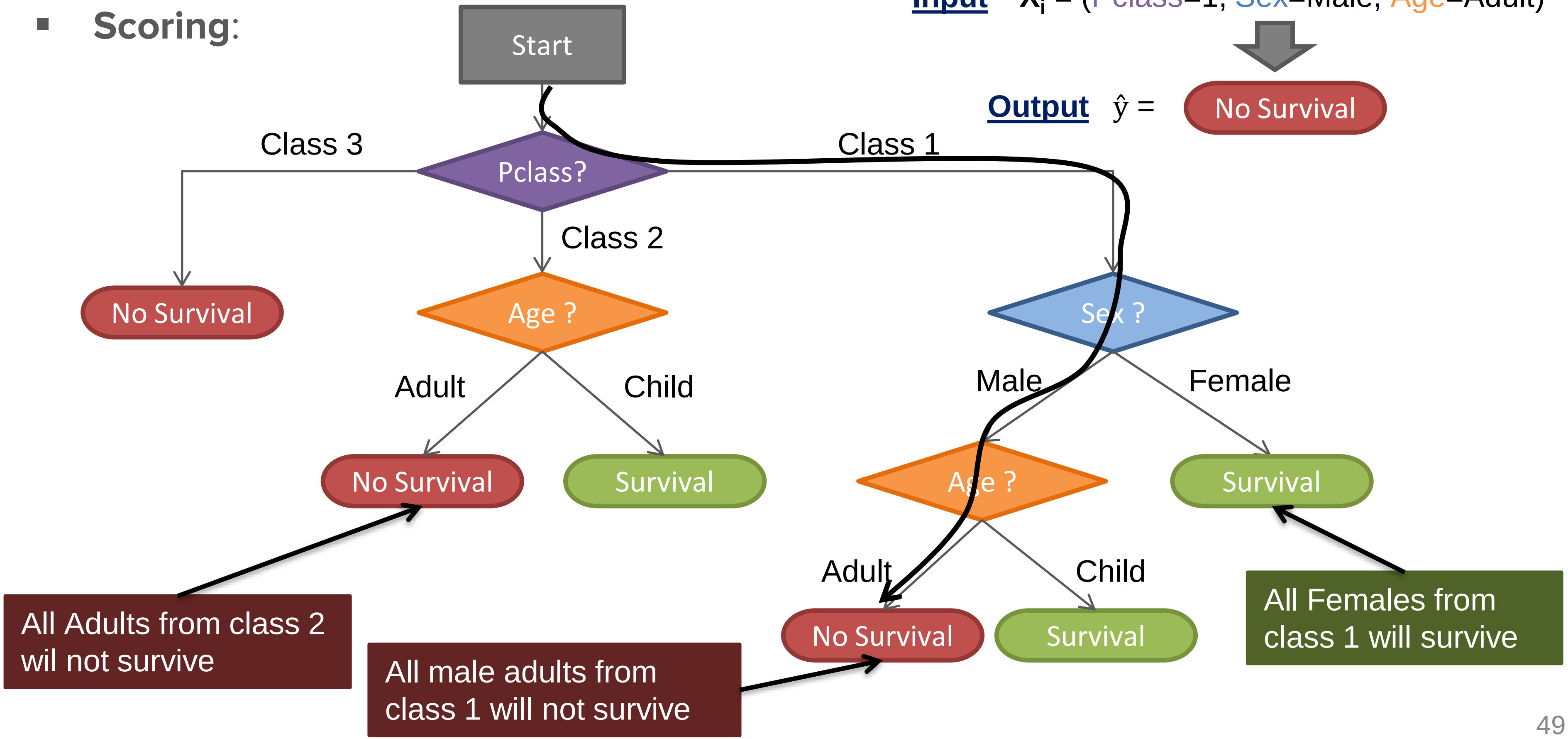


Decision Trees Prediction

- **Scoring:**

Input $\mathbf{X}_i$ = (Pclass=1, Sex=Male, Age=Adult)

Output $\hat{y}$ = No Survival



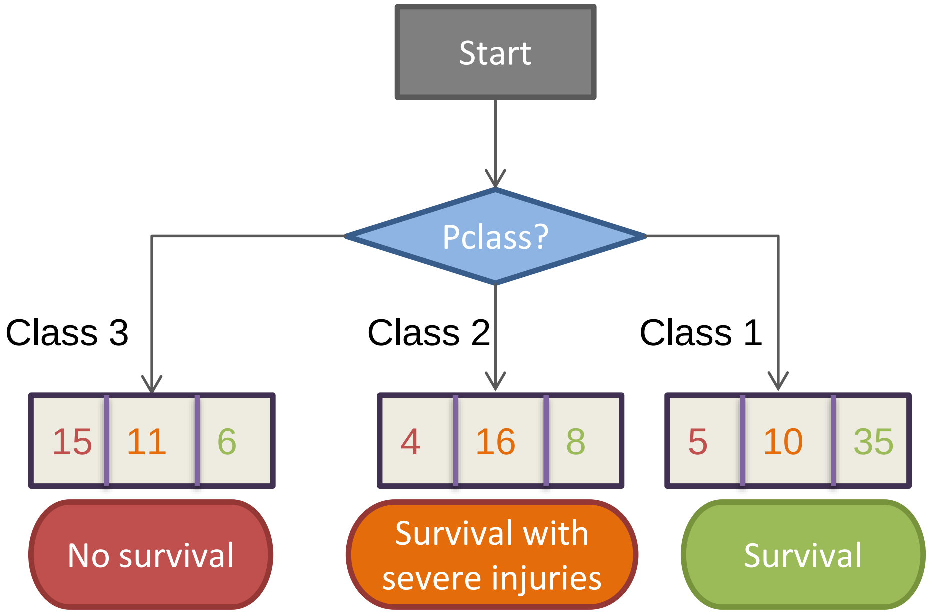
4.3
Multiclass Classification

Multiclass Classification

- **Multiclass Decision stump**

Pclass	Sex	Survival?
3	male	No survival
1	female	Survival with severe injuries
3	female	No survival
1	female	Survival
3	male	No survival
1	male	No Survival
1	male	survival
3	female	Survival with severe injuries
2	female	survival
3	female	No survival
1	female	Survival with severe injuries
3	male	No survival
3	female	No survival
....	....	....

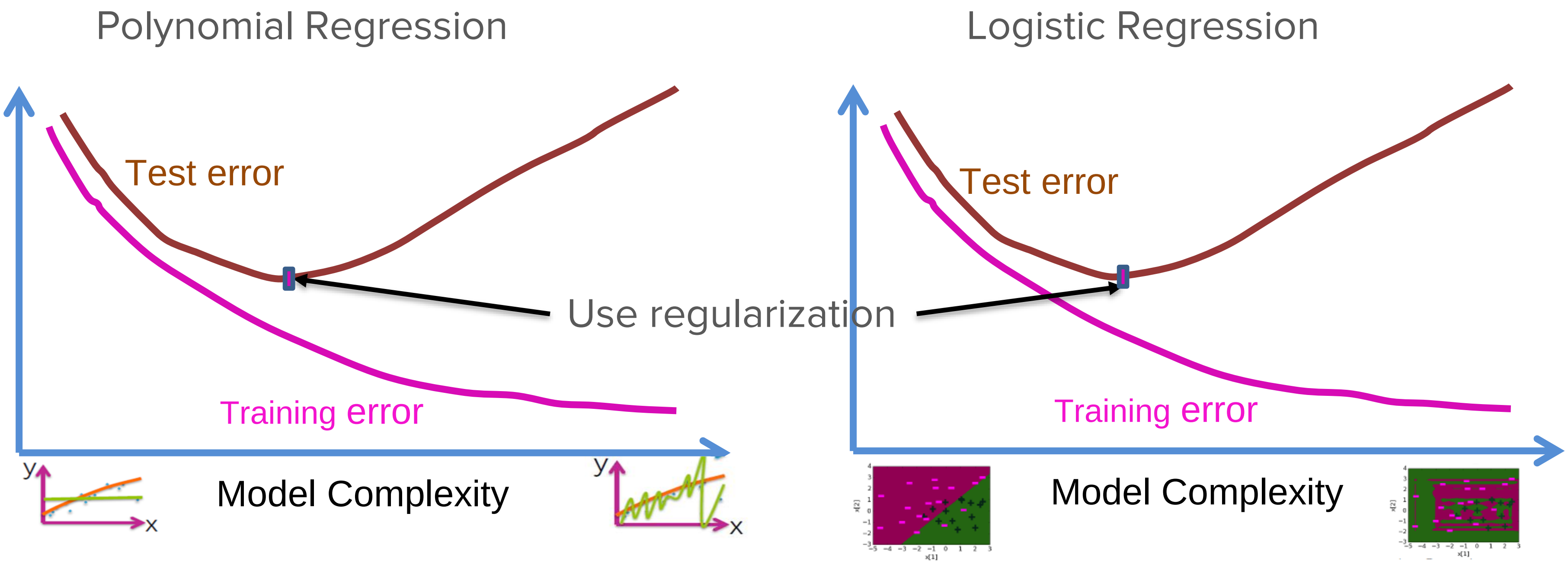
Survival status : Survival
or Survival with severe injuries
or No survival



For each intermediate node, set ŷ = majority value

4.4
Overfitting in decision trees

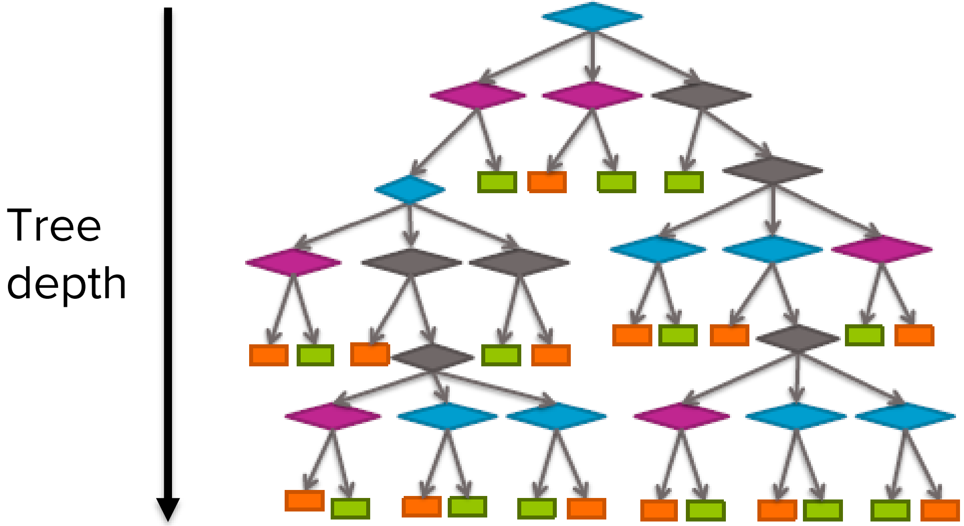
Overfitting review

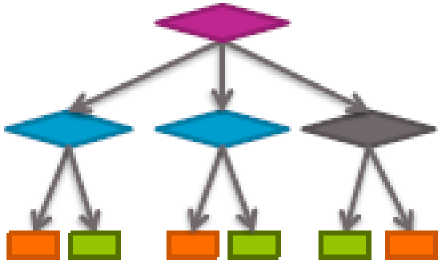


- What about decision trees ?

Model Complexity in Decision Trees

- Which tree is simpler?



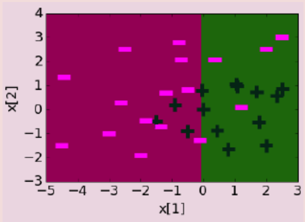
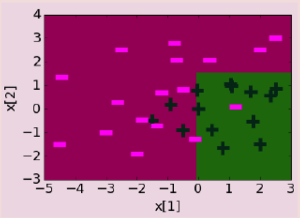
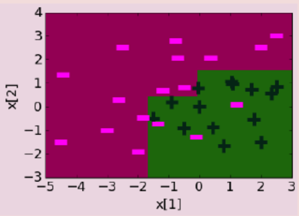
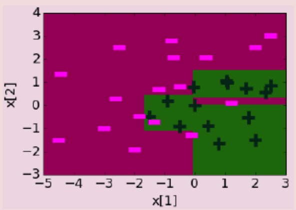
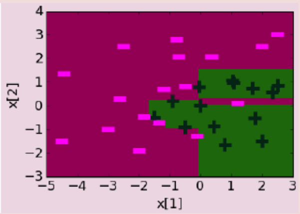


Simpler

- Tree depth is an indicator of model complexity

Overfitting in Decision Trees

- What happens when we increase depth?

Tree depth	depth = 1	depth = 2	depth = 3	depth = 5	depth = 10
Training error	0.23	0.13	0.1	0.033	**0.00**
Decision boundary					

- More depth = More complexity = Risk of overfitting

→ Implement **Early Stopping** before the tree becomes too complex

Early stopping to prevent overfitting

- Control how to grow the tree using the [following parameters]()

`sklearn.tree`.DecisionTreeClassifier

class `sklearn.tree.` **`DecisionTreeClassifier`** (*criterion='gini', splitter='best', max_depth=None, min_samples_split=2, min_samples_leaf=1, min_weight_fraction_leaf=0.0, max_features=None, random_state=None, max_leaf_nodes=None, min_impurity_split=1e-07, class_weight=None, presort=False*) [source]

- Max_depth: The maximum depth of the tree
- min_samples_split: minimum number of samples required to split an internal node
- min_samples_leaf: Minimum number of samples required to be at a leaf node

- *min_weight_fraction_leaf, max_leaf_nodes, min_impurity_split* are also helpful but less used in practice

4.5
Ensemble Methods

Ensemble Methods

- **Goal:** Combine the predictions of several base estimators (ex. Decision trees) in order to improve generalizability / robustness over a single estimator

Two families of ensemble methods are usually distinguished:

- **Bagging (Averaging methods)**: the driving principle is to build several estimators on different **subsets** of the data. Prediction proceeds with **majority vote** (averaging)
  - Example: Random Forest

- **Boosting methods**: base estimators are **built sequentially** and one tries to reduce the error of the previous one. Prediction proceeds with **weighted vote**.
  - Example: Adaboost

- These methods apply also for Classification and for Regression

Bagging

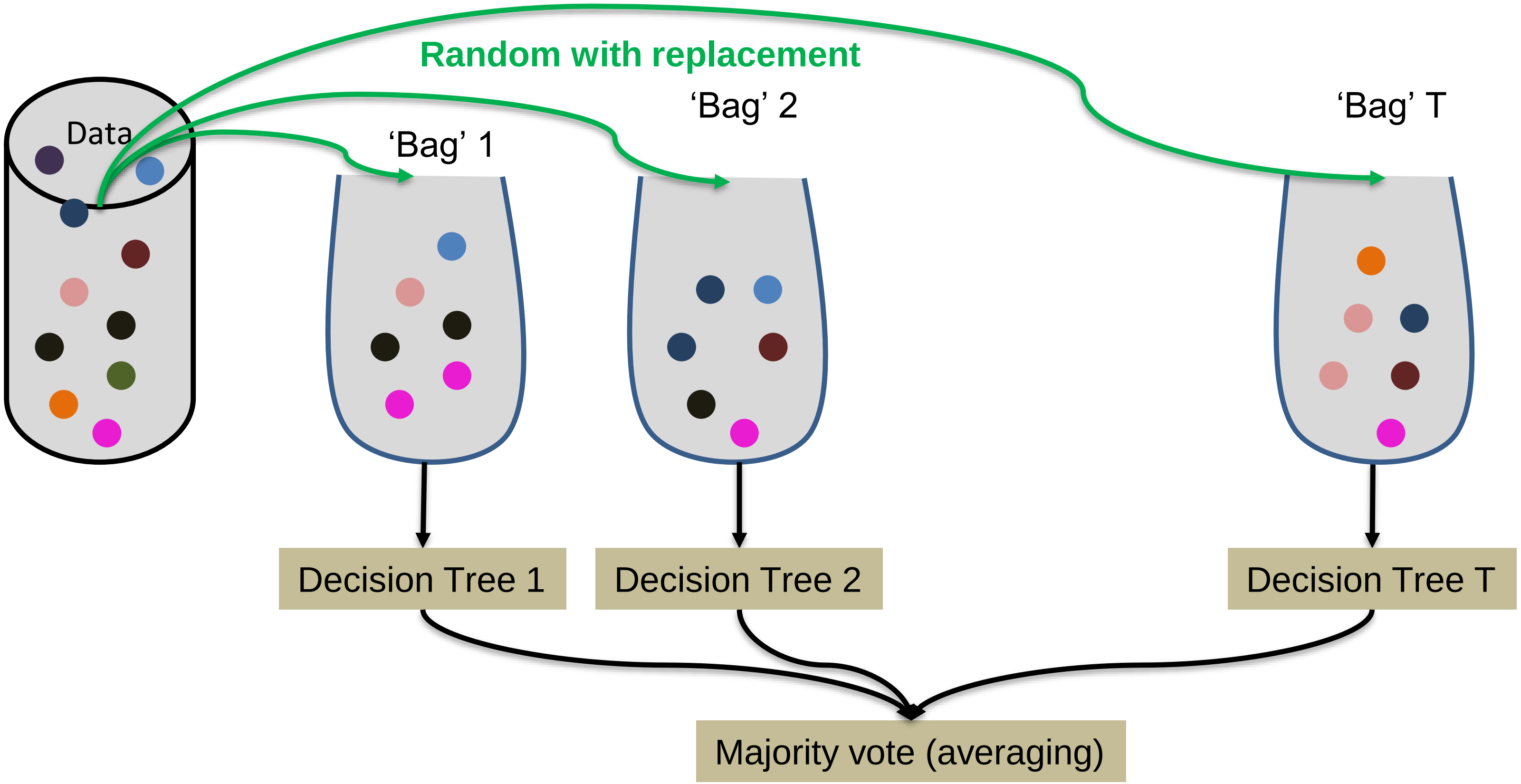
- Each tree in the ensemble is built from a sub-sample drawn with replacement (i.e., a **bootstrap sample**) from the training set.
  - A bootstrap simple of size s: Draw s points with **replacement** at random from the training set. (So some of the data is repeated, but it's ok!)
  - Usually, s = 60%

- To predict a new observation x, use the majority vote of the trees on x (**averaging**)

- Bootstrapping samples + averaging outputs = **Bagging**

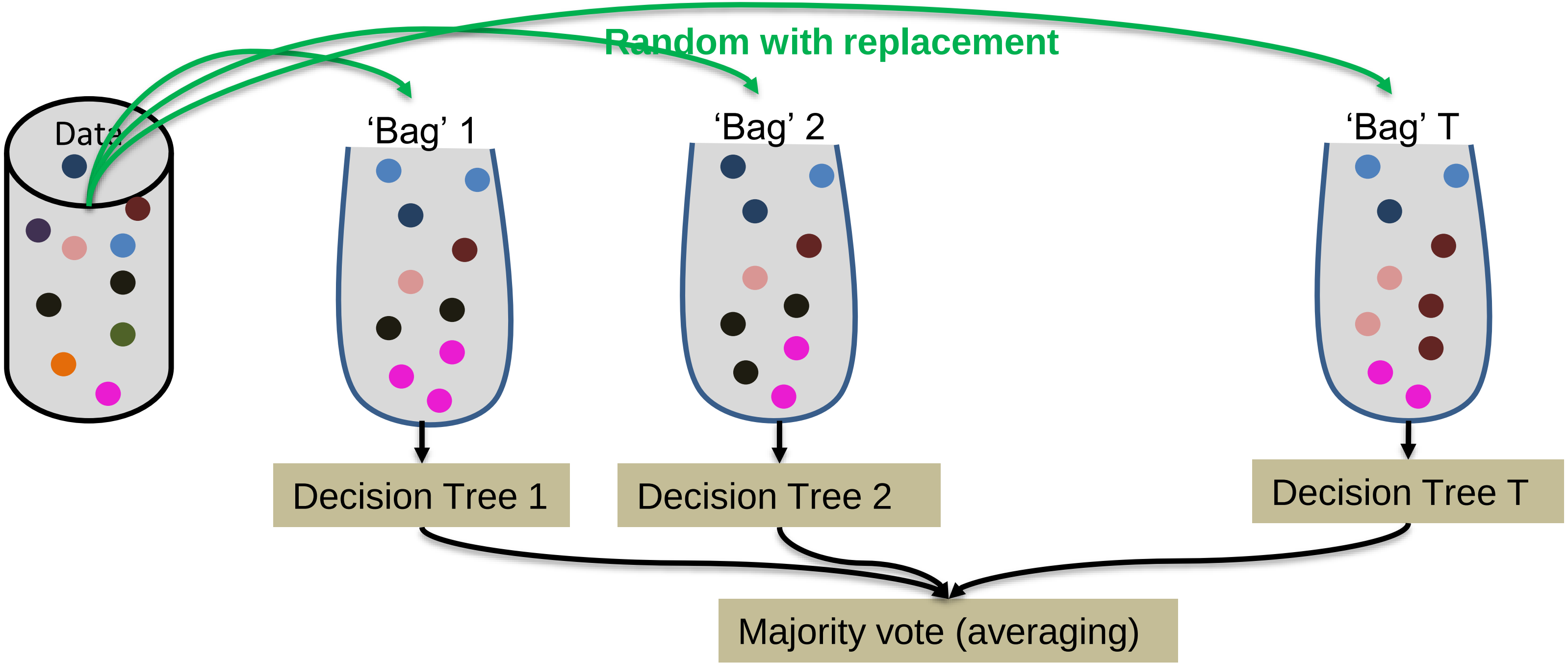
- Bagging works with other classification algorithms, also apply for regression
  - Bagging Classifer
  - Bagging Regressor

Bagging


Random with replacement
Data
'Bag' 1
'Bag' 2
'Bag' T
Decision Tree 1
Decision Tree 2
Decision Tree T
Majority vote (averaging)

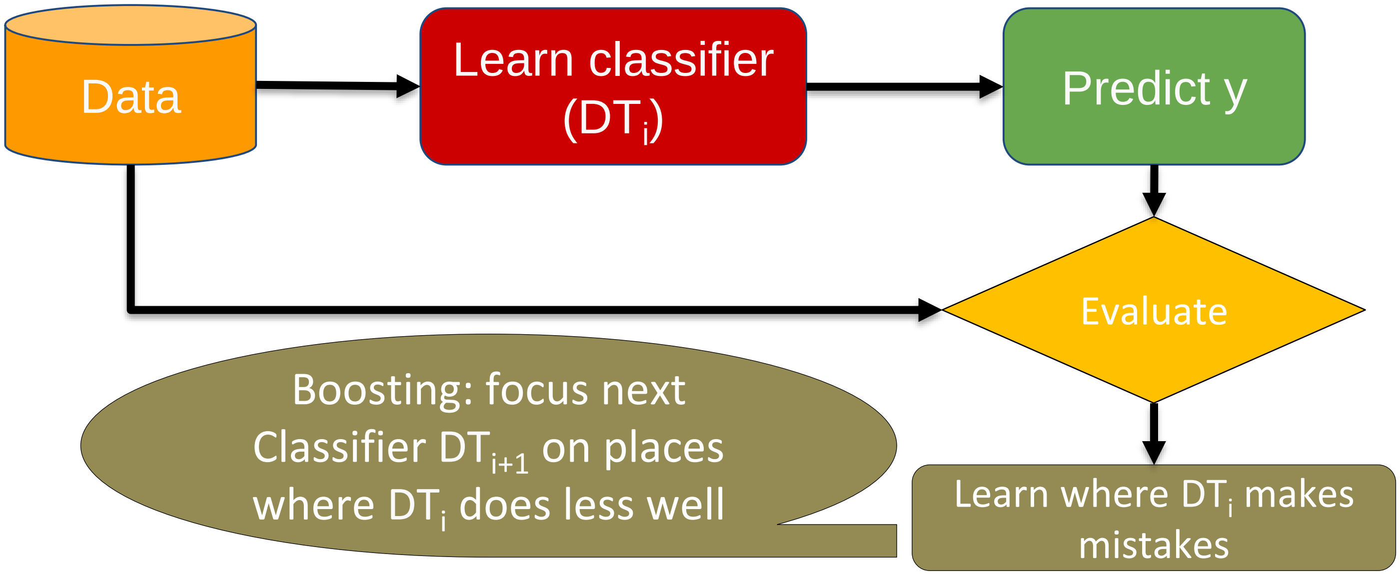
Random Forest

- Random forest is a special case of bagging where:
  - The sub-sample size is always the same as the original input sample size
  - When splitting, pick the best split among a random subset of the features.



Boosting

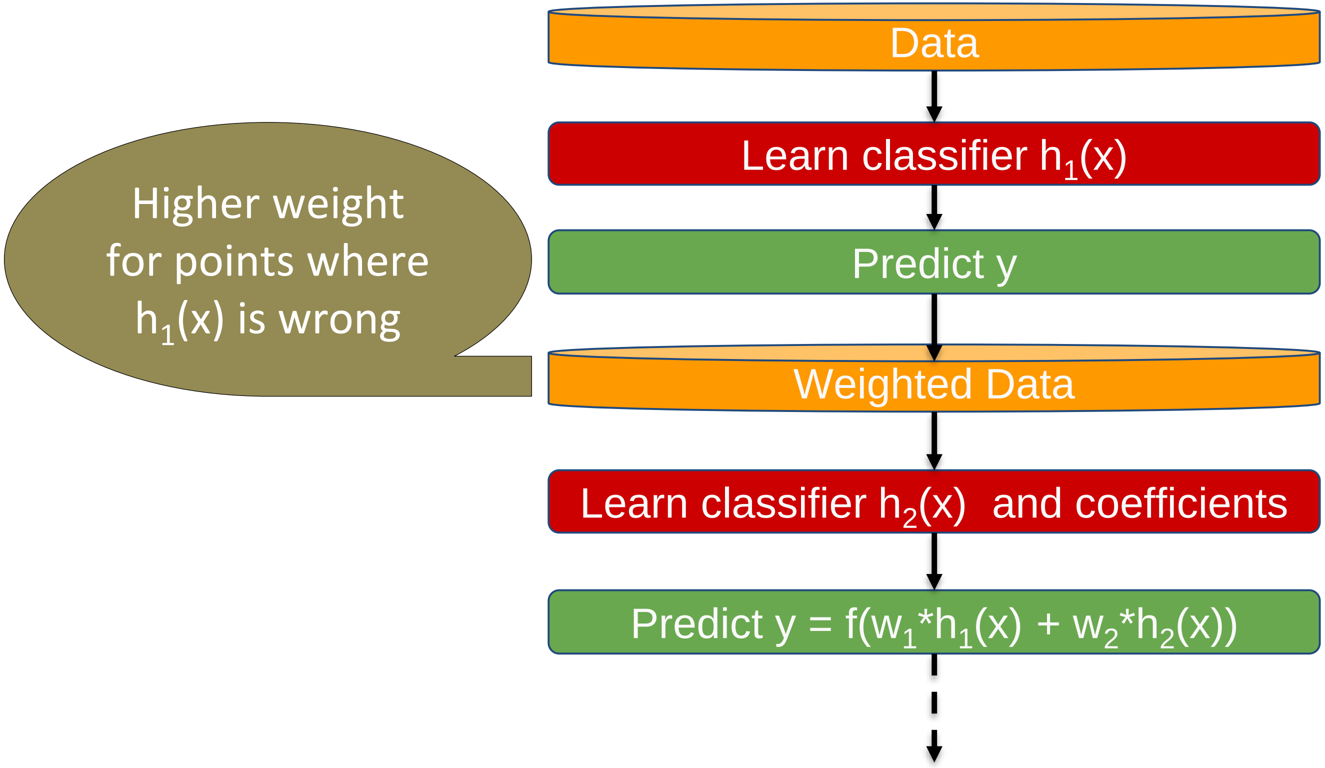
- Goal: turn a “weak” learning algorithm into a “strong” one



- Boosting = Focus learning on “hard” points.

Boosting in general

- Learning from weighted data



AdaBoost

- Adaboost is a boosting algorithm developed in 1999 by Freund & Schapire

- Start same weight for all points: $\alpha^i = 1/m$
- For t = 1,…,T
  - Learn h_t (x) with data weights α^i
  - Compute h_t (x)'s coefficient w_t
  - Update data weights α^i
  - Normalize data weights α^i

- Final model predicts by:
  - $\hat{y} = \text{sign}\ (\sum_{t=1}^{T} w_t * ht(x))$ → Two classes {+1, -1}

AdaBoost

- Adaboost is a boosting algorithm developed in 1999 by Freund & Schapire

- Start same weight for all points: $\alpha^i = 1/m$
- For t = 1,…,T
  - Learn h_t (x) with data weights α^i
  - Compute h_t (x)'s coefficient w_t
  - Update data weights α^i
  - Normalize data weights α^i

$$w_t = \frac{1}{2}\ln\left(\frac{1 - weigted\ error(ht(x))}{weigted\ error(ht(x))}\right)$$

$$\alpha^i \leftarrow \begin{cases} \alpha^i * e^{-wt}, if\, ht(x^i) = yi \\ \alpha^i * e^{wt}, if\, ht(x^i) \neq y^i \end{cases}$$

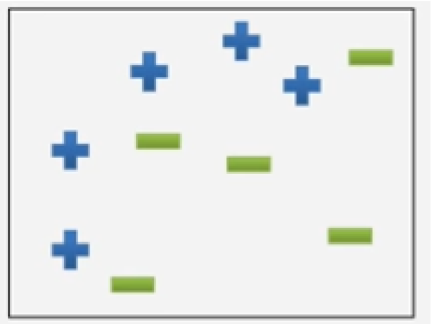
$$\alpha^i \leftarrow \frac{\alpha^i}{\sum_{j=1}^{m} \alpha^j}$$

- Final model predicts by:
  - $\hat{y} = \mathrm{sign}\left(\sum_{t=1}^{T} w_t * ht(x)\right)$

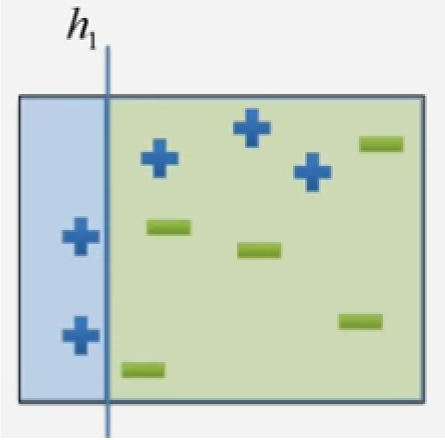
Optional

AdaBoost Example

- Our weak classifiers are only allowed to be lines that are either horizontal or vertical.



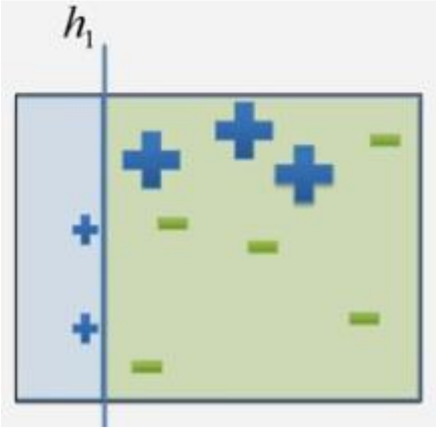
All data points start with equal weights



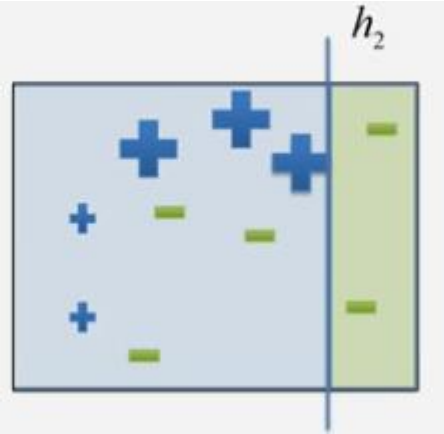
Run the weak learning algorithm, to get a weak classifier

Choose coefficient $w_1 = 0.41$

AdaBoost Example



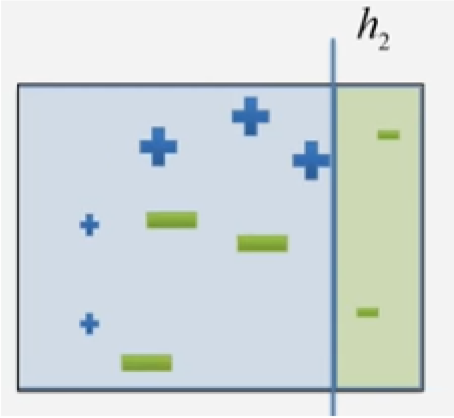
Increase the weights on the misclassified points.
Decrease the weights on the correctly classified points.



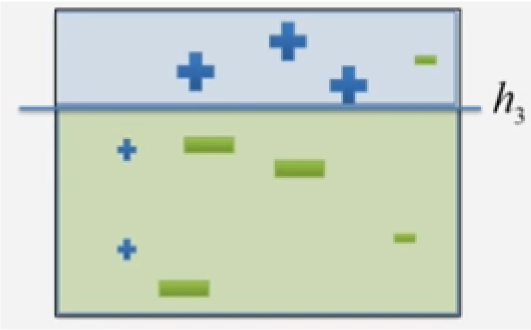
Run the weak learning algorithm, to get a weak classifier for the weighted data

Choose coefficient $w_2 = 0.66$

AdaBoost Example



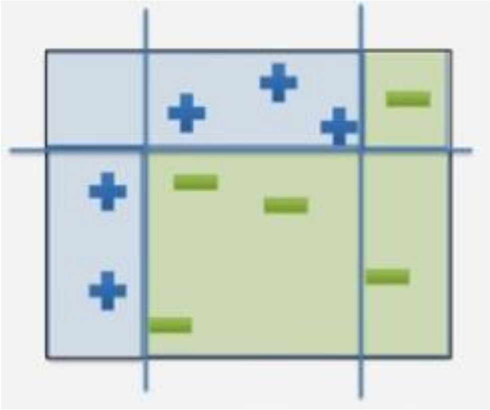
Increase the weights on the misclassified points.
Decrease the weights on the correctly classified points.



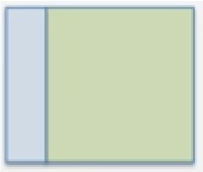
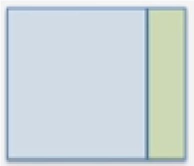
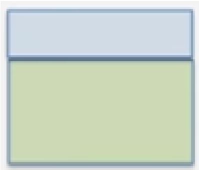
Run the weak learning algorithm, to get a weak classifier for the weighted data

Choose coefficient $w_3 = 0.93$

AdaBoost Example



Combined classifier

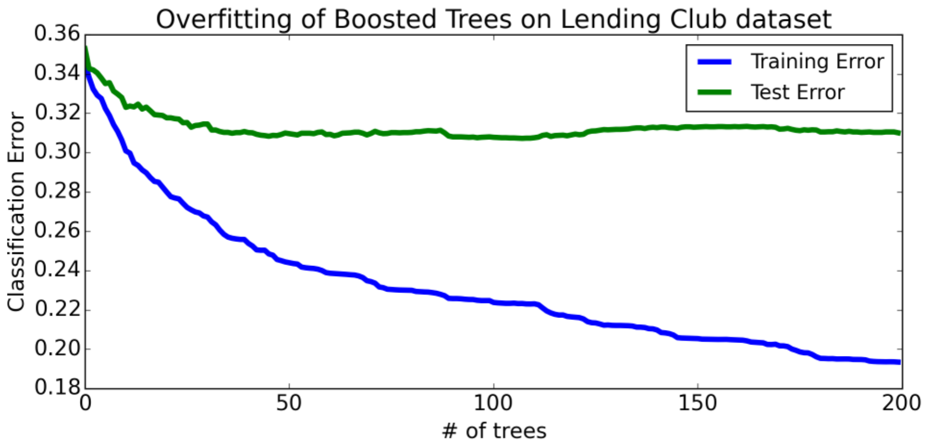
$\hat{y} = \text{sign}\,(0.41 *$  $+0.66 *$  $+0.93 *$  $)$

Credit: Adapted from Freund & Schapire, edx

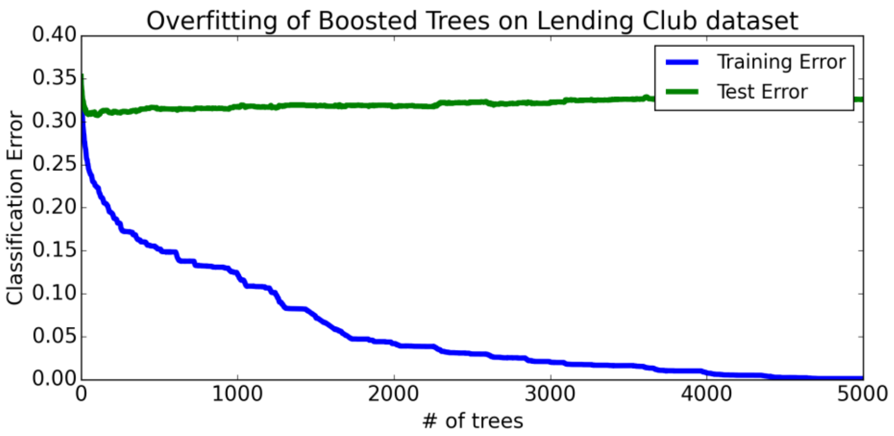
Boosting and overfitting

- Example: [Lending Club dataset]

Boosting tends to be robust to overfitting



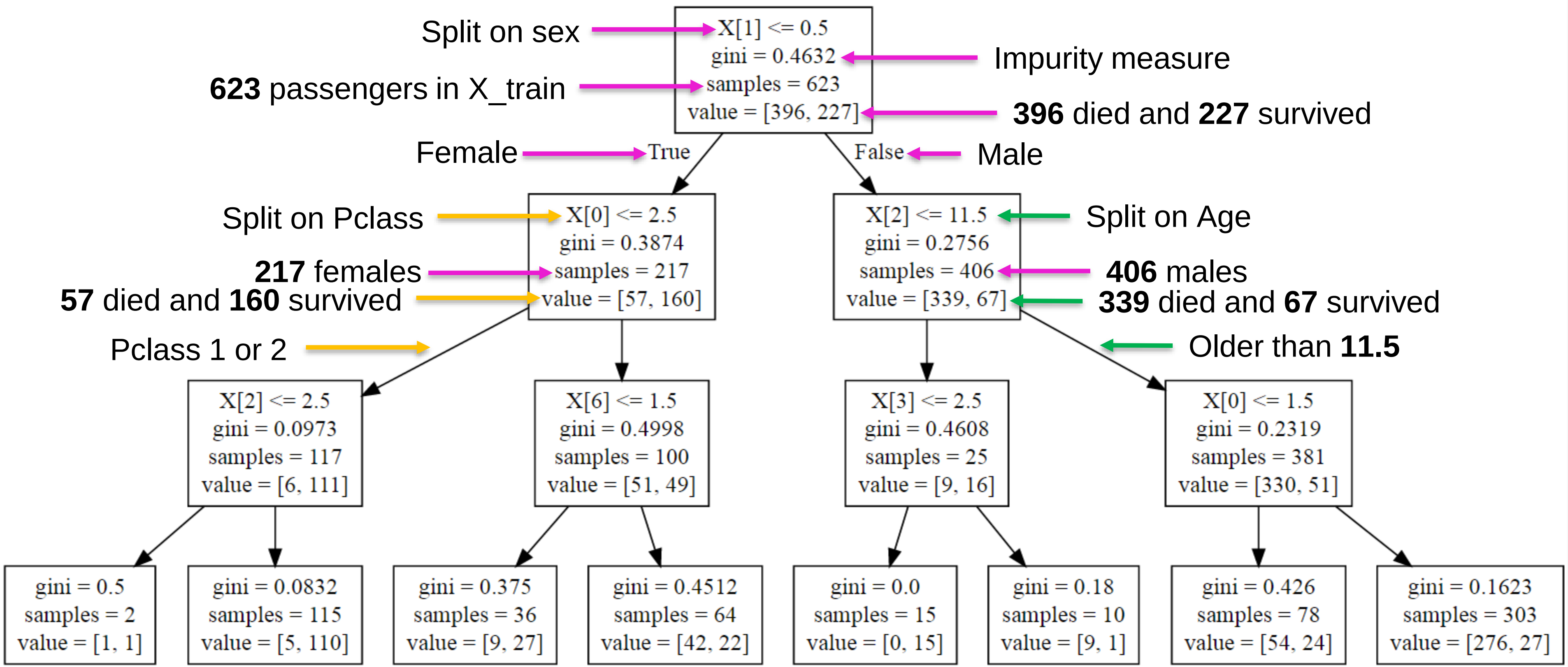
But will eventually overfit with large T



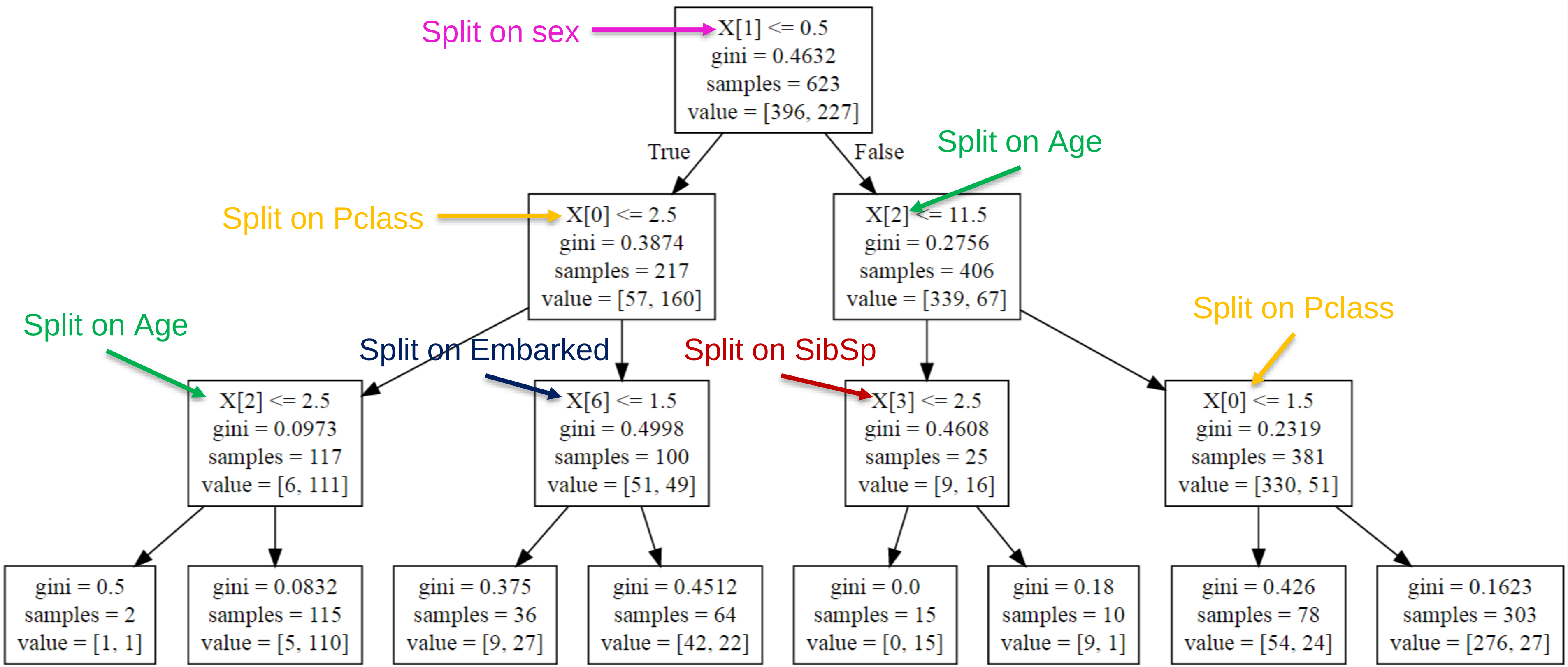
- Use cross validation to choose the value of T

4.6
Practical Work

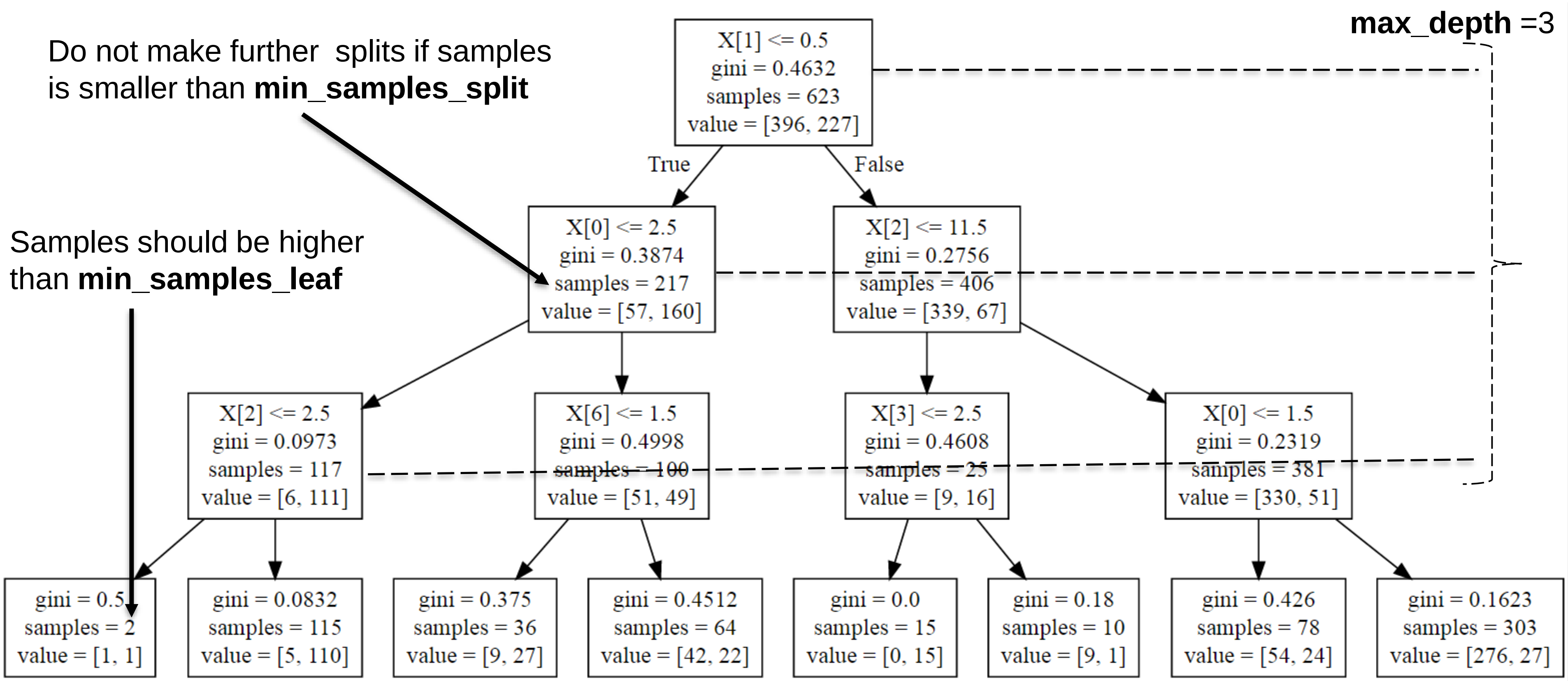
Titanic Decision Tree



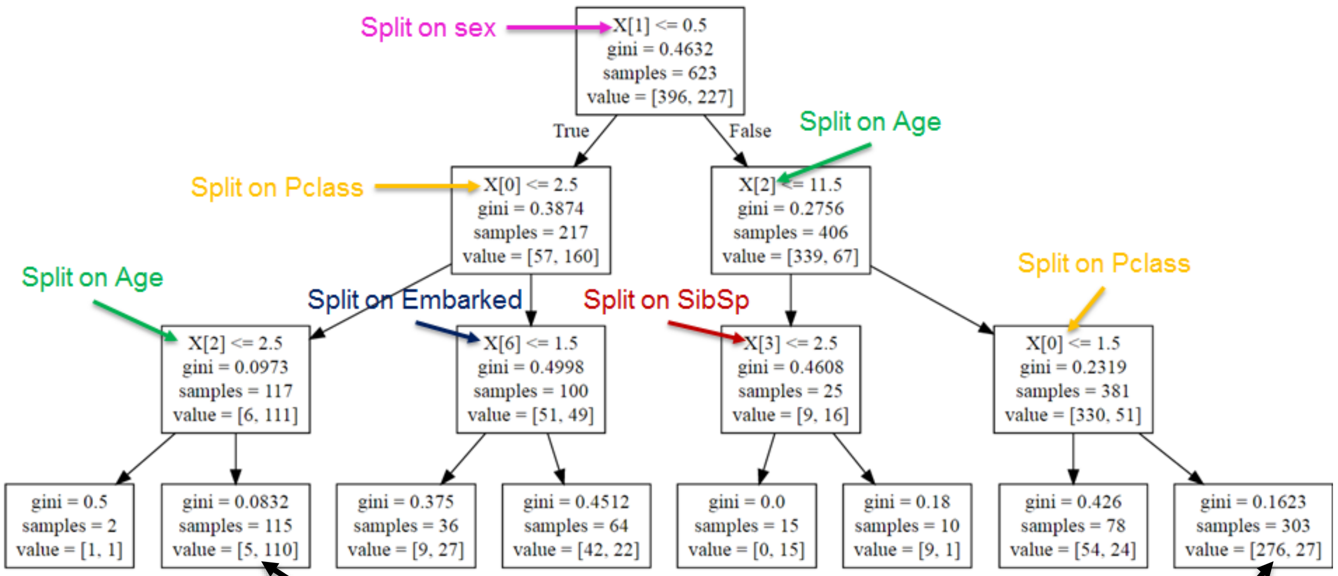
Titanic Decision Tree



Titanic Decision Tree



Titanic Decision Tree



115 females, Pclass **1** or **2**, above age **2.5**: **5** died and **110** survived.
Proba of survival = **110/115 = 95.65** %

303 males, above age **11.5** Pclass **2** or **3**: **276** died and **27** survived.
Proba of survival = **27/303 = 8.91** %

Thank you for your attention