Machine Learning Days,
EPFL

Ensemble Techniques: “Unity is strength”

Jessica Lanini
EPFL, Lausanne
January, 2020

NOVARTIS | Reimagining Medicine

Machine Learning Days, EPFL

Ensemble Techniques: "Unity is strength"

What to expect from the workshop:

[13:30-14:30] **Theory of ensemble techniques**

- What ensemble techniques are
- Different types of ensemble techniques: overview
- Background: Decision Tree
- Random Forest
- AdaBoost

[14:30- 14:45] **Q&A**

[14:45-15:00] **Introduction to the Mini-Project: Predicting Daily Bike Rentals**

[15-17] **Mini-Project: Predicting Daily Bike Rentals**

- Analysis and Data Preparation
- Definition of useful functions (e.g. cross-validation)
- Models evaluation
- Analysis of features importance
- Robustness Analysis
- Hyperparameters Optimization

NOVARTIS | Reimagining Medicine

What Ensemble Techniques Are

Different types of Ensemble Techniques: Overview

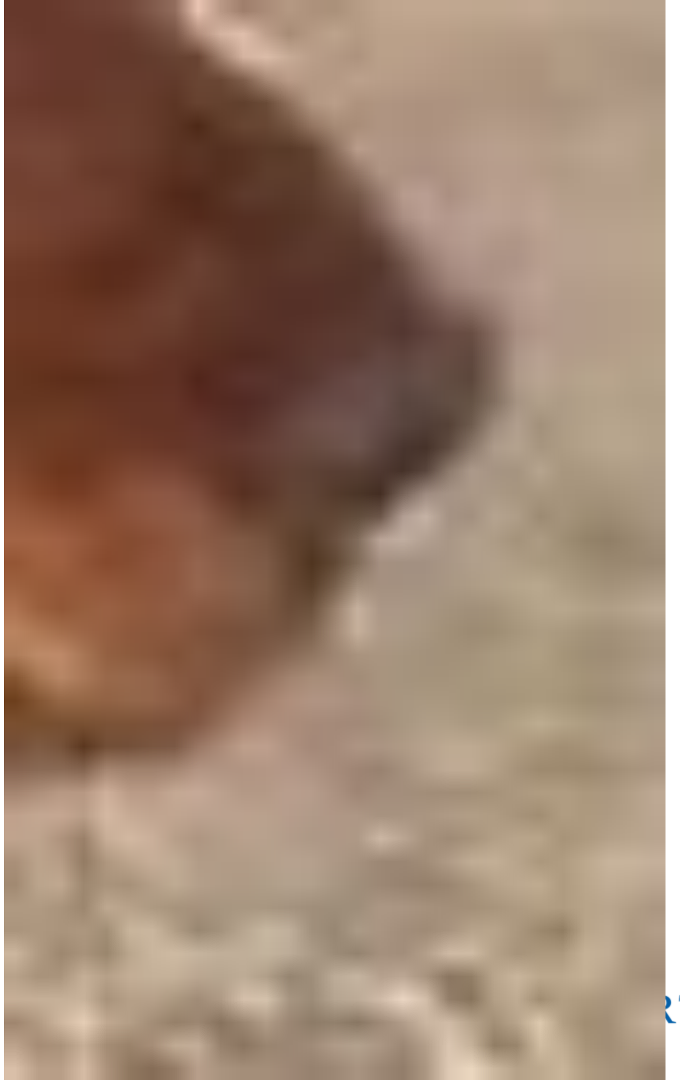
Background: Decision Tree

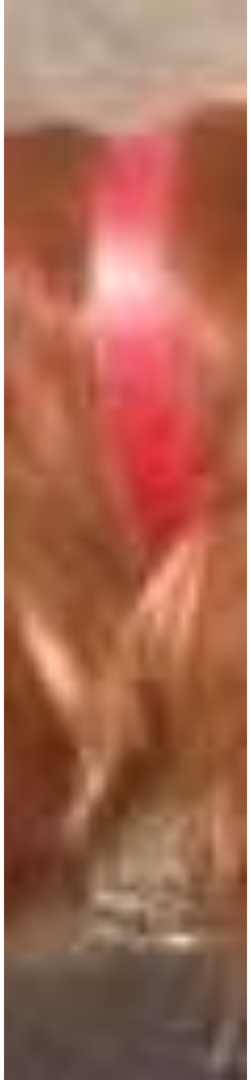
Random Forest

ADA Boost

QUIZ!!!

- What does the image represent?





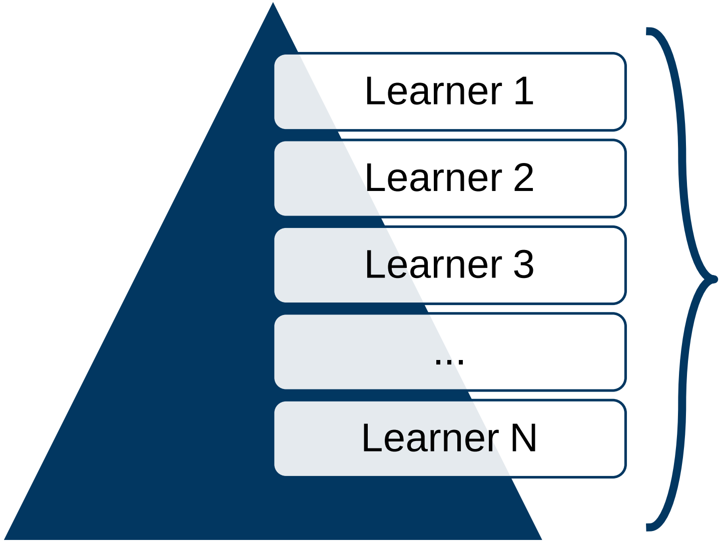
Reimagining Medicine

QUIZ!!!

- What does the image represent?

What Ensemble Techniques Are

- `Techniques that use multiple learning algorithms to obtain better predictive performance than could be obtained from any of the constituent learning algorithms alone.' [Wikipedia]
- Idea:
  - Train multiple `weak' learner
  - Combine their prediction
    – Voting
    – Averaging
    – Weighted Averaging



What Ensemble Techniques Are

Different types of Ensemble Techniques: Overview

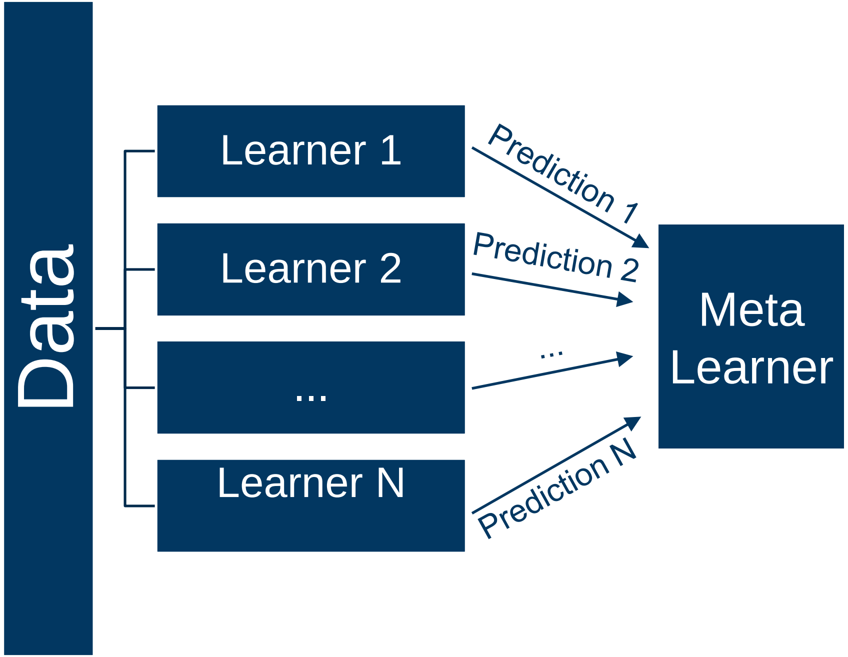
Background: Decision Tree

Random Forest

ADA Boost

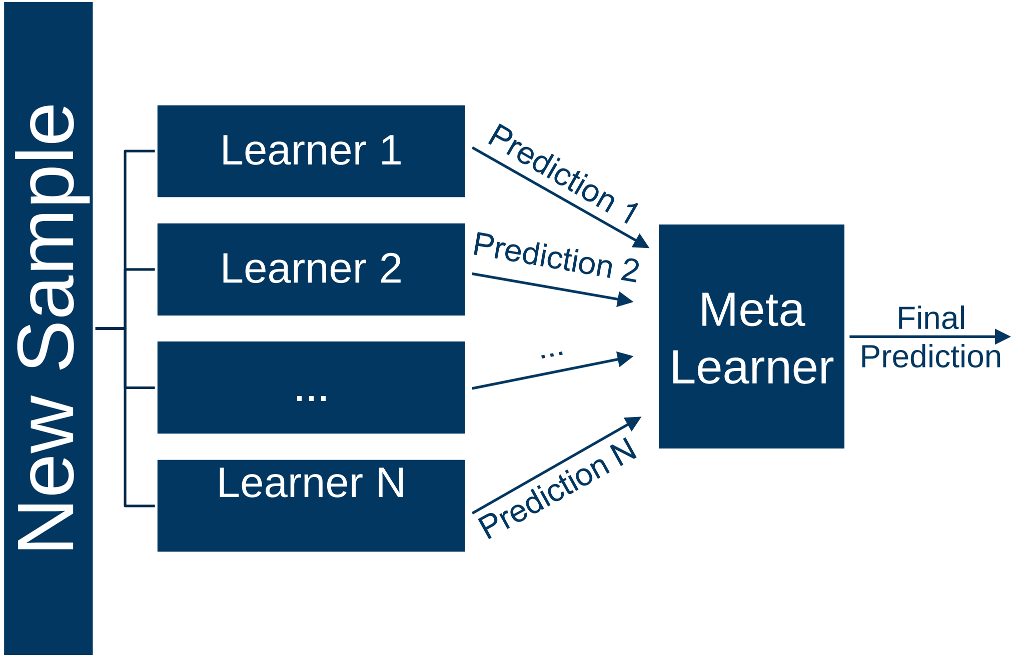
Types of Ensemble Techniques: Overview

- **Stacking**
  - N learners
  - Same training set
  - Meta Learner/Blender
    - –takes previous predictions



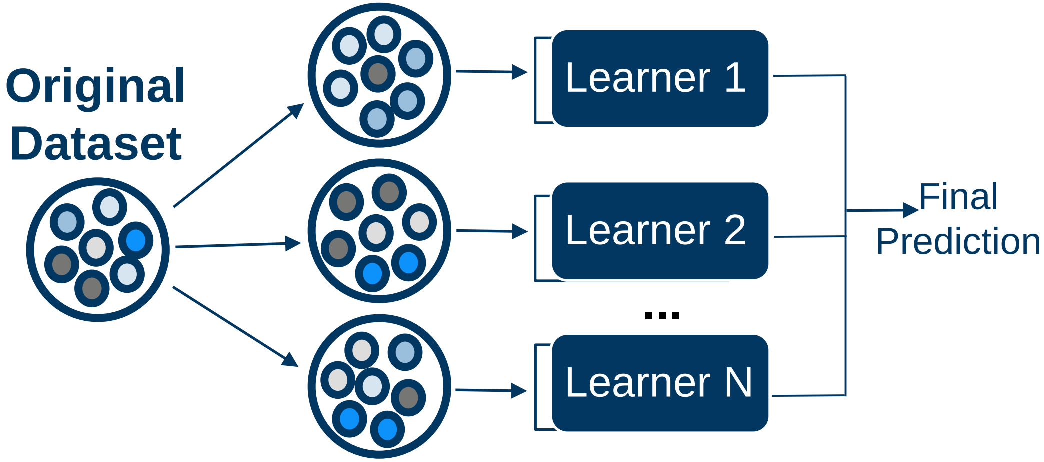
Types of Ensemble Techniques: Overview

- **Stacking**
  - N learners
  - Same training set
  - Meta Learner/Blender
    - takes previous predictions
  - New sample



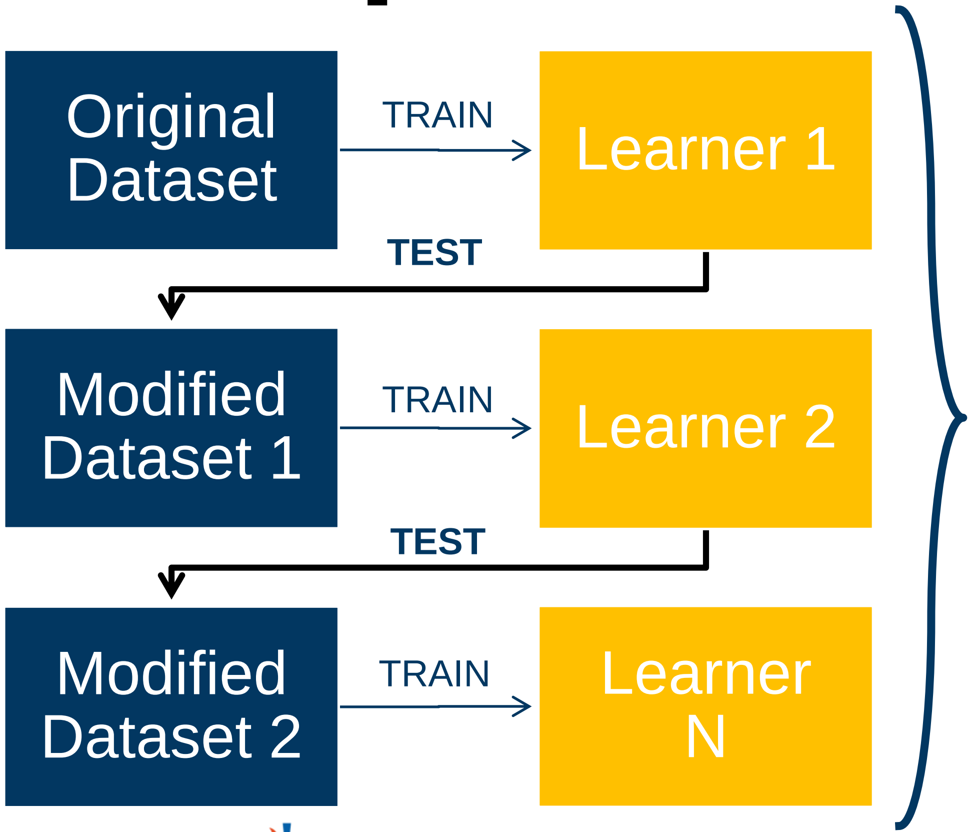
Types of Ensemble Techniques: Overview

- **Bagging**
  - Random sampling with replacement of the training set (bootstrap)
  - Train *n* learners in parallel
  - Combine predictions
  - Reduce variance



Types of Ensemble Techniques: Overview

- **Boosting**
  - Weak learners are trained in series
  - Train set changes according to the error of the previous learner
  - Final prediction depends on the learner’s amount of saying
  - Reduce bias



What Ensemble Techniques Are

Different types of Ensemble Techniques: Overview

Background: Decision Tree

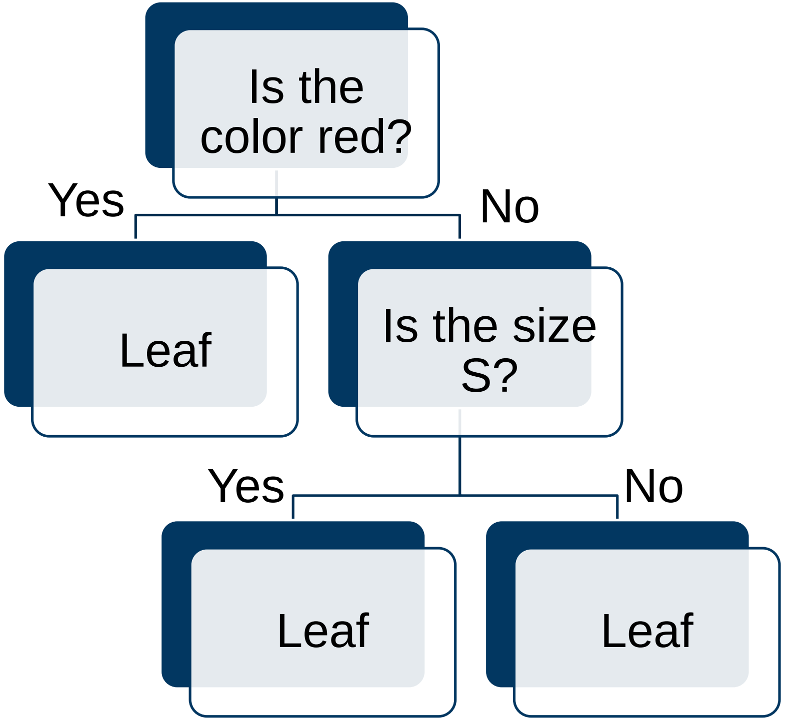
Random Forest

ADA Boost

NOVARTIS | Reimagining Medicine

Decision Tree

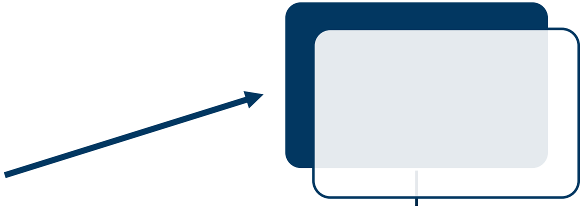
- Non-parametric supervised learning method
- Different types:
  - ID3
  - C4.5
  - C5.0
  - CART
    - –Classification and Regression Tree
- Procedure to decide **which** question to ask and **when**



Decision Tree

Color	Size	Discount	Label
Red	M	50%	Shirt
Yellow	S	20%	Jeans
Red	S	50%	Shirt
Red	M	40%	Shirt
Green	L	50%	Skirt

Decision Tree



Color	Size	Discount	Label
Red	M	50%	Shirt
Yellow	S	20%	Jeans
Red	S	50%	Shirt
Red	M	40%	Shirt
Green	L	50%	Skirt

Decision Tree

- All nodes receive a list of samples
  - Root receives the entire training set
- True/False question about one of the features

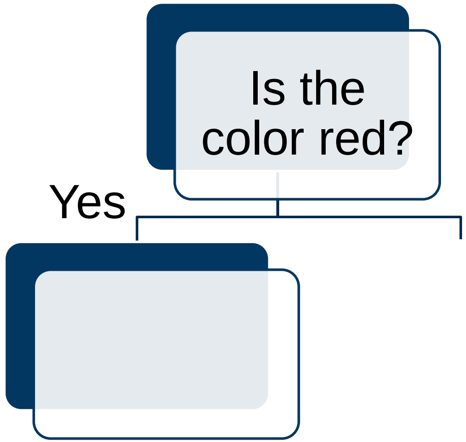
Decision Tree

Color	Size	Discount	Label
Red	M	50%	Shirt
Yellow	S	20%	Jeans
Red	S	50%	Shirt
Red	M	40%	Shirt
Green	L	50%	Skirt

Is the color red?

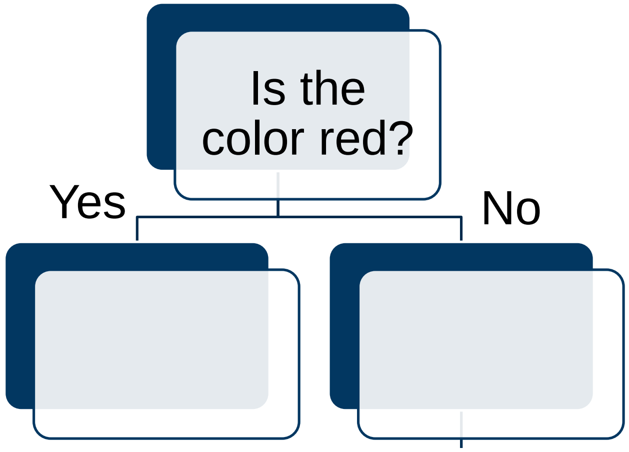
Decision Tree

Color	Size	Discount	Label
Red	M	50%	Shirt
Yellow	S	20%	Jeans
Red	S	50%	Shirt
Red	M	40%	Shirt
Green	L	50%	Skirt



Decision Tree

Color	Size	Discount	Label
Red	M	50%	Shirt
Yellow	S	20%	Jeans
Red	S	50%	Shirt
Red	M	40%	Shirt
Green	L	50%	Skirt

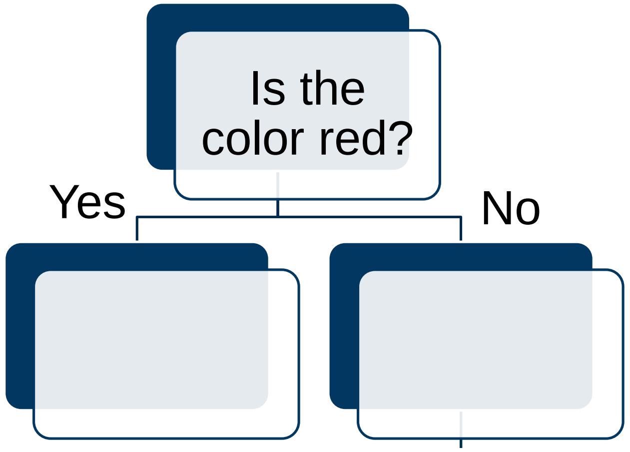


Decision Tree

- All nodes receive a list of samples
  - Root receives the entire training set
- True/False question about one of the features
- The data are split according to the answer at each sample

Decision Tree

Color	Size	Discount	Label
Red	M	50%	Shirt
Red	S	50%	Shirt
Red	M	40%	Shirt



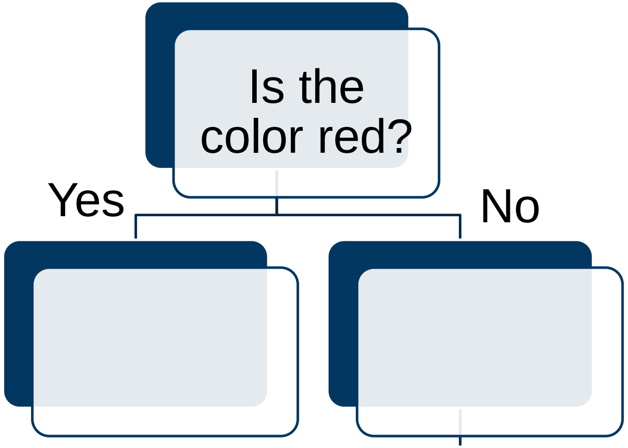
Color	Size	Discount	Label
Yellow	S	20%	Jeans
Green	L	50%	Skirt

Decision Tree

- All nodes receive a list of samples
  - Root receives the entire training set
- True/False question about one of the features
- The data are split according to the answer at each sample
- Goal: unmix label

Decision Tree

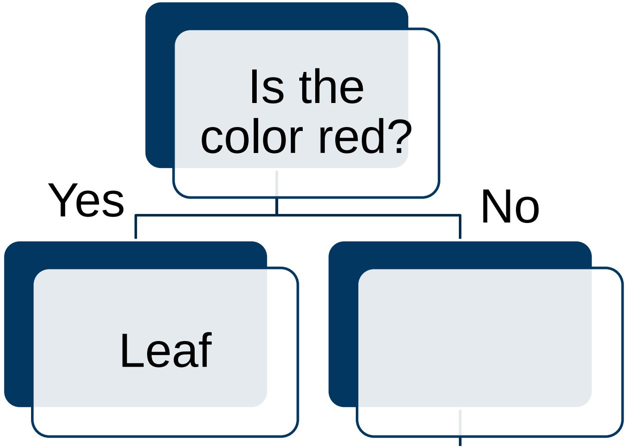
Color	Size	Discount	Label
Red	M	50%	Shirt
Red	S	50%	Shirt
Red	M	40%	Shirt



Color	Size	Discount	Label
Yellow	S	20%	Jeans
Green	L	50%	Skirt

Decision Tree

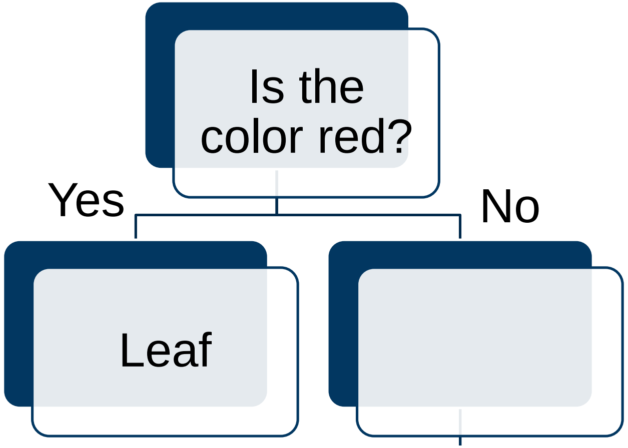
Color	Size	Discount	Label
Red	M	50%	Shirt
Red	S	50%	Shirt
Red	M	40%	Shirt



Color	Size	Discount	Label
Yellow	S	20%	Jeans
Green	L	50%	Skirt

Decision Tree

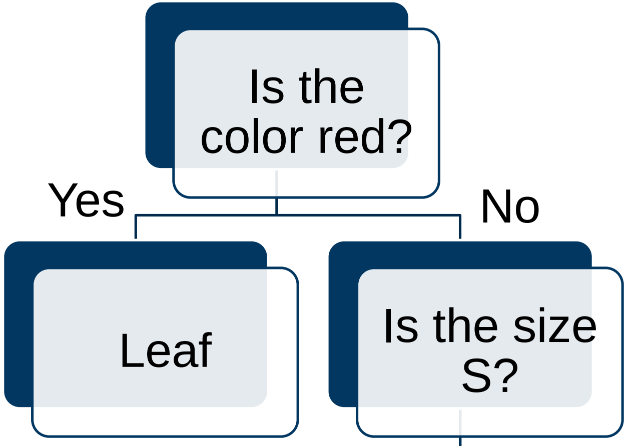
Color	Size	Discount	Label
Red	M	50%	Shirt
Red	S	50%	Shirt
Red	M	40%	Shirt



Color	Size	Discount	Label
Yellow	S	20%	Jeans
Green	L	50%	Skirt

Decision Tree

Color	Size	Discount	Label
Red	M	50%	Shirt
Red	S	50%	Shirt
Red	M	40%	Shirt

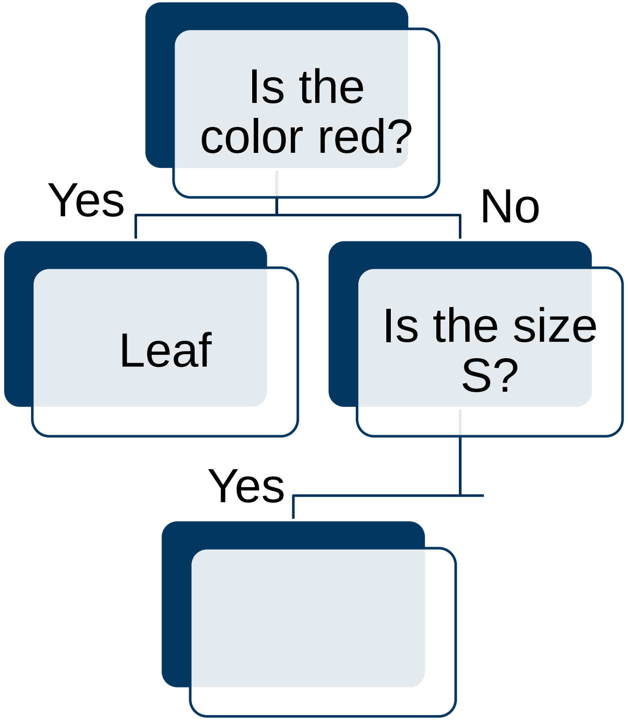


Color	Size	Discount	Label
Yellow	S	20%	Jeans
Green	L	50%	Skirt

Decision Tree

Color	Size	Discount	Label
Red	M	50%	Shirt
Red	S	50%	Shirt
Red	M	40%	Shirt

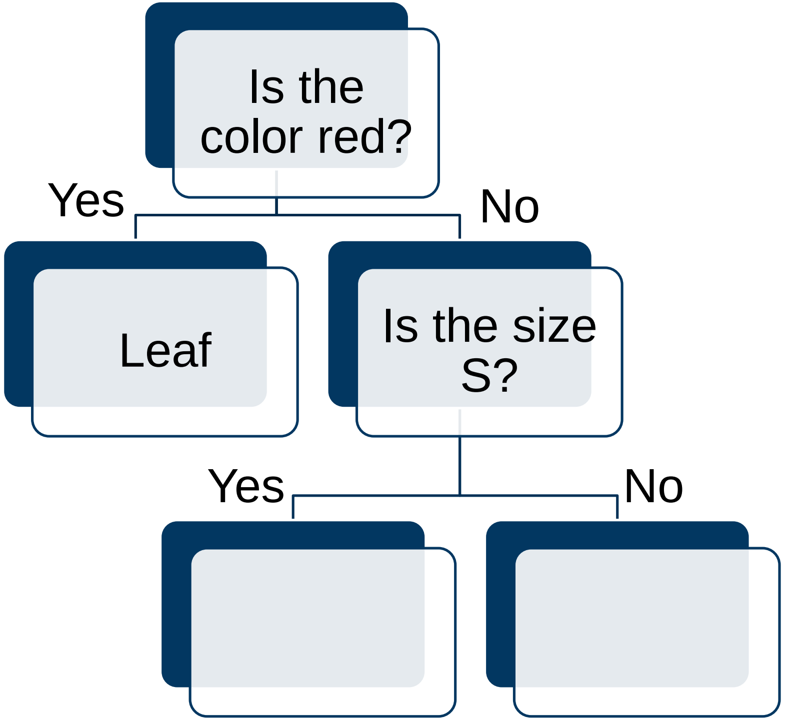
Color	Size	Discount	Label
Yellow	S	20%	Jeans
Green	L	50%	Skirt



Decision Tree

Color	Size	Discount	Label
Red	M	50%	Shirt
Red	S	50%	Shirt
Red	M	40%	Shirt

Color	Size	Discount	Label
Yellow	S	20%	Jeans
Green	L	50%	Skirt



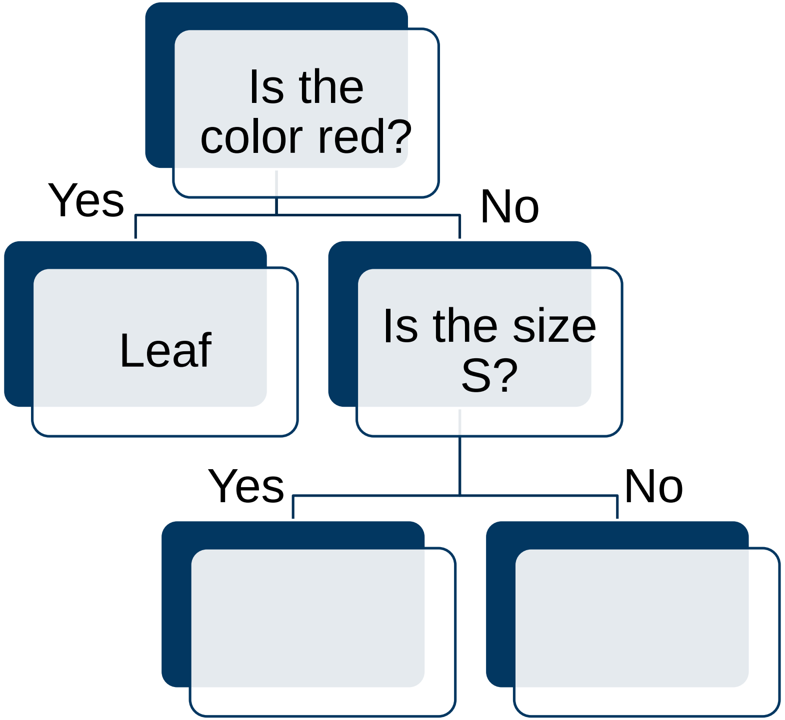
NOVARTIS | Reimagining Medicine

Decision Tree

Color	Size	Discount	Label
Red	M	50%	Shirt
Red	S	50%	Shirt
Red	M	40%	Shirt

Color	Size	Discount	Label
Yellow	S	20%	Jeans

Color	Size	Discount	Label
Green	L	50%	Skirt

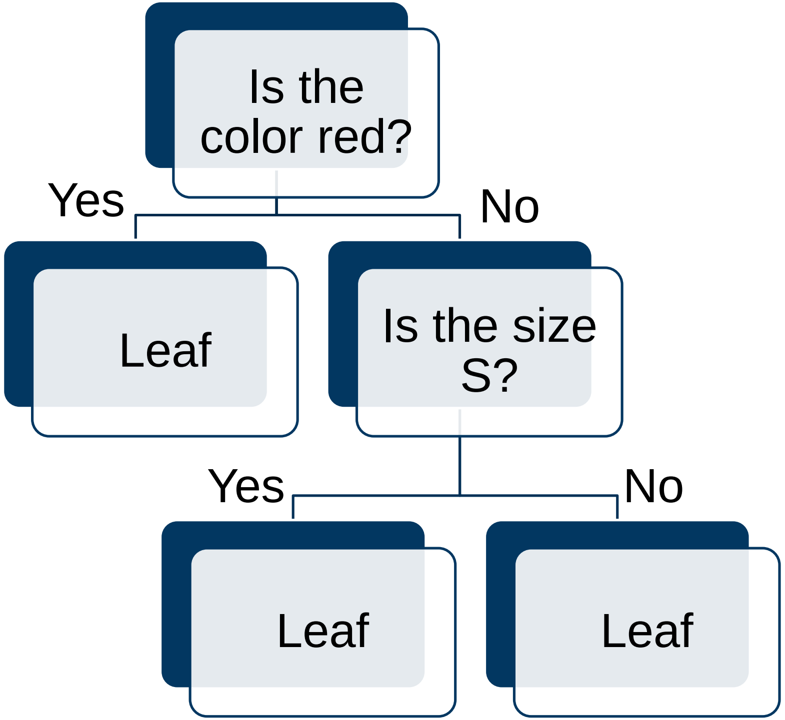


Decision Tree

Color	Size	Discount	Label
Red	M	50%	Shirt
Red	S	50%	Shirt
Red	M	40%	Shirt

Color	Size	Discount	Label
Yellow	S	20%	Jeans

Color	Size	Discount	Label
Green	L	50%	Skirt



Decision Tree

Color	Size	Discount	Label
Red	M	50%	Shirt
Yellow	S	20%	Jeans
Red	S	50%	Shirt
Red	M	40%	Shirt
Green	L	50%	Skirt

Decision Tree

- **Which** question to ask and **when**

Color	Size	Discount	Label
Red	M	50%	Shirt
Yellow	S	20%	Jeans
Red	S	50%	Shirt
Red	M	40%	Shirt
Green	L	50%	Skirt

Decision Tree

- **Which** question to ask and **when**
  - Gini impurity (or Entropy)
  - Information Gain

Color	Size	Discount	Label
Red	M	50%	Shirt
Yellow	S	20%	Jeans
Red	S	50%	Shirt
Red	M	40%	Shirt
Green	L	50%	Skirt

Decision Tree

- **Which** question to ask and **when**
  - Gini impurity (or Entropy)
  - Information Gain

Color	Size	Discount	Label
Red	M	50%	Shirt
Yellow	S	20%	Jeans
Red	S	50%	Shirt
Red	M	40%	Shirt
Green	L	50%	Skirt

Decision Tree

- **Gini Impurity**
  - quantifies the amount of uncertainty

Color	Size	Discount	Label
Red	M	50%	Shirt
Yellow	S	20%	Jeans
Red	S	50%	Shirt
Red	M	40%	Shirt
Green	L	50%	Skirt

Decision Tree

- **Gini Impurity**
  - quantifies the amount of uncertainty
  - [0,1]

Color	Size	Discount	Label
Red	M	50%	Shirt
Yellow	S	20%	Jeans
Red	S	50%	Shirt
Red	M	40%	Shirt
Green	L	50%	Skirt

Decision Tree

- **Gini Impurity**
  - quantifies the amount of uncertainty
  - [0,1]

$$G = \sum_{i=1}^{C} p(i) * (1 - p(i))$$

Color	Size	Discount	Label
Red	M	50%	Shirt
Yellow	S	20%	Jeans
Red	S	50%	Shirt
Red	M	40%	Shirt
Green	L	50%	Skirt

Decision Tree

- **Gini Impurity**
  - quantifies the amount of uncertainty
  - [0,1]

$$G = \sum_{i=1}^{C} p(i) * (1 - p(i))$$

Color	Size	Discount	Label
Red	M	50%	Shirt
Yellow	S	20%	Jeans
Red	S	50%	Shirt
Red	M	40%	Shirt
Green	L	50%	Skirt

$$G_1 = \overbrace{\frac{3}{5} \cdot (1 - \frac{3}{5})}^{\text{Shirt}} + \frac{1}{5} \cdot (1 - \frac{1}{5}) + \frac{1}{5} \cdot (1 - \frac{1}{5})$$

$$= \frac{14}{25}$$

NOVARTIS | Reimagining Medicine

Decision Tree

- **Gini Impurity**
  - quantifies the amount of uncertainty
  - [0,1]

$$G = \sum_{i=1}^{C} p(i) * (1 - p(i))$$

Color	Size	Discount	Label
Red	M	50%	Shirt
Yellow	S	20%	Jeans
Red	S	50%	Shirt
Red	M	40%	Shirt
Green	L	50%	Skirt

$$G_1 = \frac{3}{5} \cdot (1 - \frac{3}{5}) + \overbrace{\frac{1}{5} \cdot (1 - \frac{1}{5})}^{\text{Jeans}} + \frac{1}{5} \cdot (1 - \frac{1}{5})$$

$$= \frac{14}{25}$$

Decision Tree

- **Gini Impurity**
  - quantifies the amount of uncertainty
  - [0,1]

$$G = \sum_{i=1}^{C} p(i) * (1 - p(i))$$

Color	Size	Discount	Label
Red	M	50%	Shirt
Yellow	S	20%	Jeans
Red	S	50%	Shirt
Red	M	40%	Shirt
Green	L	50%	Skirt

$$G_1 = \frac{3}{5} \cdot (1 - \frac{3}{5}) + \frac{1}{5} \cdot (1 - \frac{1}{5}) + \overbrace{\frac{1}{5} \cdot (1 - \frac{1}{5})}^{\text{Skirt}}$$

$$= \frac{14}{25}$$

NOVARTIS | Reimagining Medicine

Decision Tree

- **Gini Impurity**
  - quantifies the amount of uncertainty
  - [0,1]

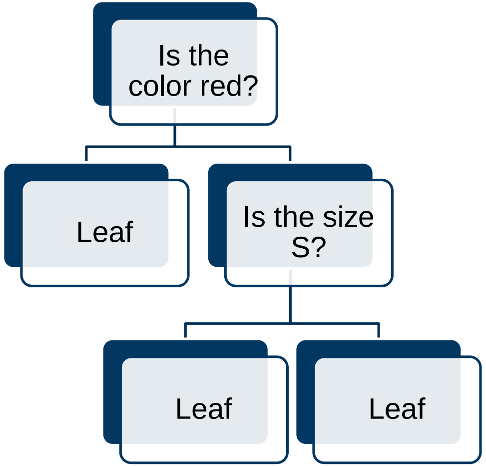
$$G = \sum_{i=1}^{C} p(i) * (1 - p(i))$$

Color	Size	Discount	Label
Red	M	50%	Shirt
Yellow	S	20%	Jeans
Red	S	50%	Shirt
Red	M	40%	Shirt
Green	L	50%	Skirt

$$\boxed{G_1 = \frac{14}{25}}$$

Decision Tree

- **Gini Impurity**

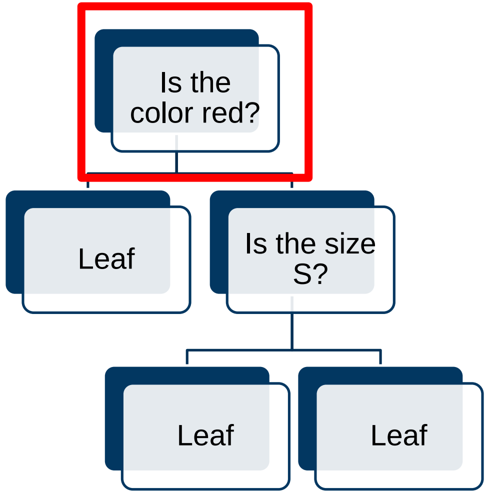


$$\boxed{G_1 = \frac{14}{25}}$$

Color	Size	Discount	Label
Red	M	50%	Shirt
Yellow	S	20%	Jeans
Red	S	50%	Shirt
Red	M	40%	Shirt
Green	L	50%	Skirt

Decision Tree

- **Gini Impurity**



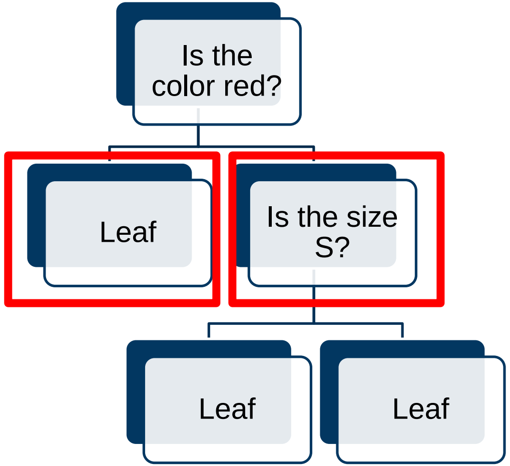
$$G_1 = \frac{14}{25}$$

Color	Size	Discount	Label
Red	M	50%	Shirt
Yellow	S	20%	Jeans
Red	S	50%	Shirt
Red	M	40%	Shirt
Green	L	50%	Skirt

Is the color Red?

Decision Tree

- **Gini Impurity**



$$G_1 = \frac{14}{25}$$

Color	Size	Discount	Label
Red	M	50%	Shirt
Yellow	S	20%	Jeans
Red	S	50%	Shirt
Red	M	40%	Shirt
Green	L	50%	Skirt

Is the color Red?

Color	Size	Discount	Label
Red	M	50%	Shirt
Red	S	50%	Shirt
Red	M	40%	Shirt

Color	Size	Discount	Label
Yellow	S	20%	Jeans
Green	L	50%	Skirt

Decision Tree

- **Gini Impurity**

$$G_1 = \frac{14}{25}$$

Color	Size	Discount	Label
Red	M	50%	Shirt
Yellow	S	20%	Jeans
Red	S	50%	Shirt
Red	M	40%	Shirt
Green	L	50%	Skirt

Is the color Red?

$$G_{2,1} = 1 \cdot (1 - 1) = 0$$

$$G_{2,2} = [\frac{1}{2} \cdot (1 - \frac{1}{2})] \cdot 2 = \frac{1}{2}$$

Color	Size	Discount	Label
Red	M	50%	Shirt
Red	S	50%	Shirt
Red	M	40%	Shirt

Color	Size	Discount	Label
Yellow	S	20%	Jeans
Green	L	50%	Skirt

Decision Tree

- **Which** question to ask and **when**
  - Gini impurity (or Entropy)
  - Information Gain

Color	Size	Discount	Label
Red	M	50%	Shirt
Yellow	S	20%	Jeans
Red	S	50%	Shirt
Red	M	40%	Shirt
Green	L	50%	Skirt

Decision Tree

- **Which** question to ask and **when**
  - Gini impurity (or Entropy)
  - Information Gain

Color	Size	Discount	Label
Red	M	50%	Shirt
Yellow	S	20%	Jeans
Red	S	50%	Shirt
Red	M	40%	Shirt
Green	L	50%	Skirt

Decision Tree

- **Information Gain**

Decision Tree

- **Information Gain**
  - How much a question reduce the uncertainties

Decision Tree

- **Information Gain**
  - How much a question reduce the uncertainties

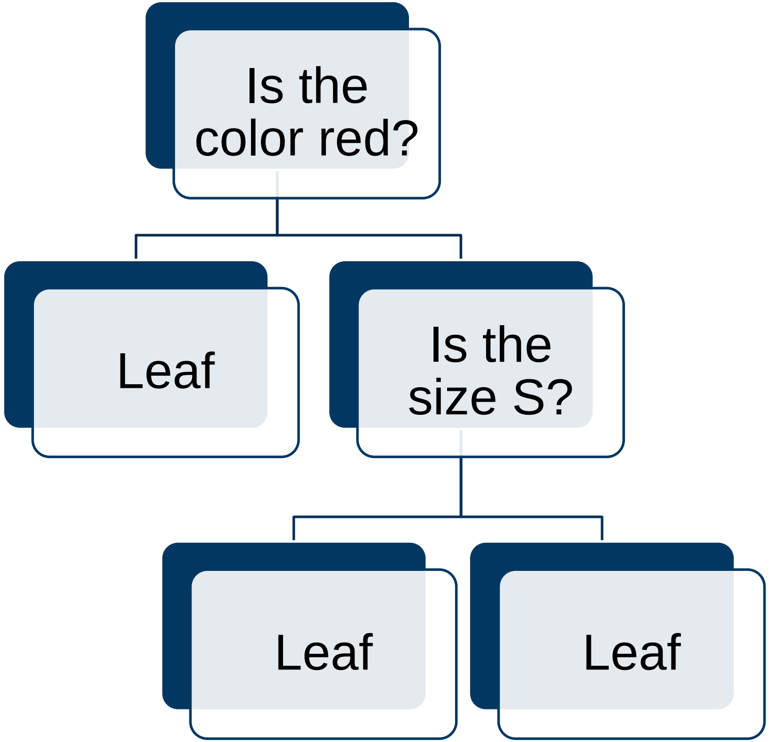
$$IG_i^j = G_i - G_j$$

Reimagining Medicine

Decision Tree

- **Information Gain**
  - How much a question reduce the uncertainties

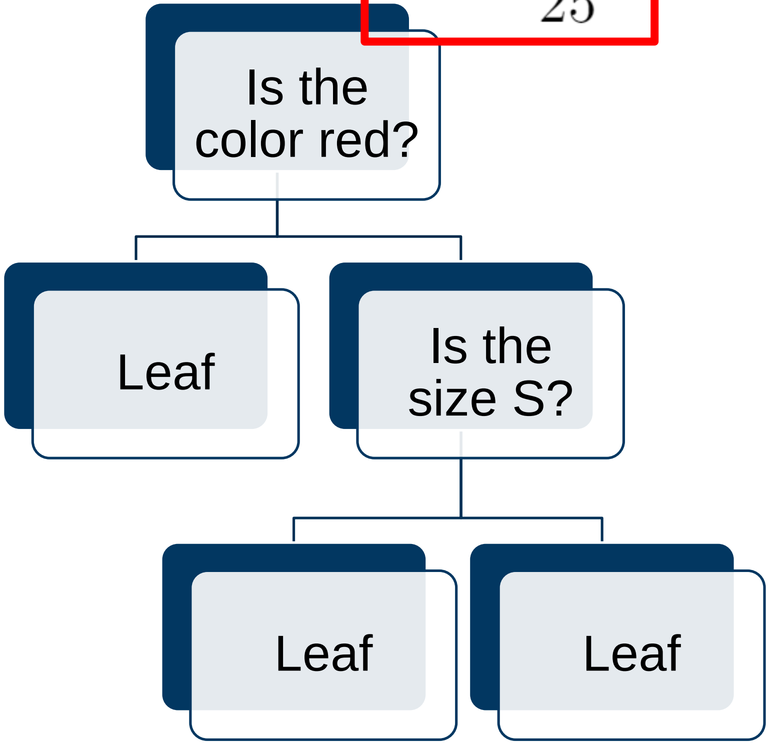
$$IG_i^j = G_i - G_j$$



Decision Tree

- **Information Gain**
  - How much a question reduce the uncertainties

$$\boxed{G_1 = \frac{14}{25}}$$

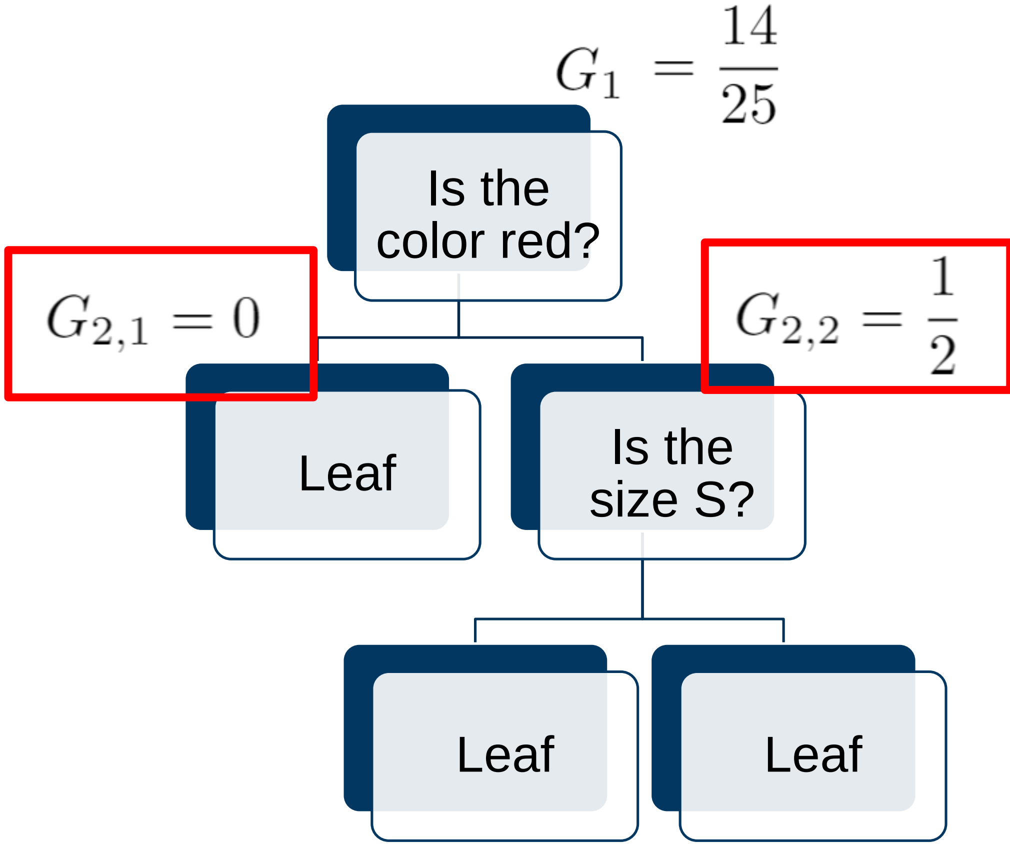


$$IG_i^j = \boxed{G_i} - G_j$$

Decision Tree

- **Information Gain**
  - How much a question reduce the uncertainties

$$IG_i^j = G_i - \boxed{G_j}$$



Decision Tree

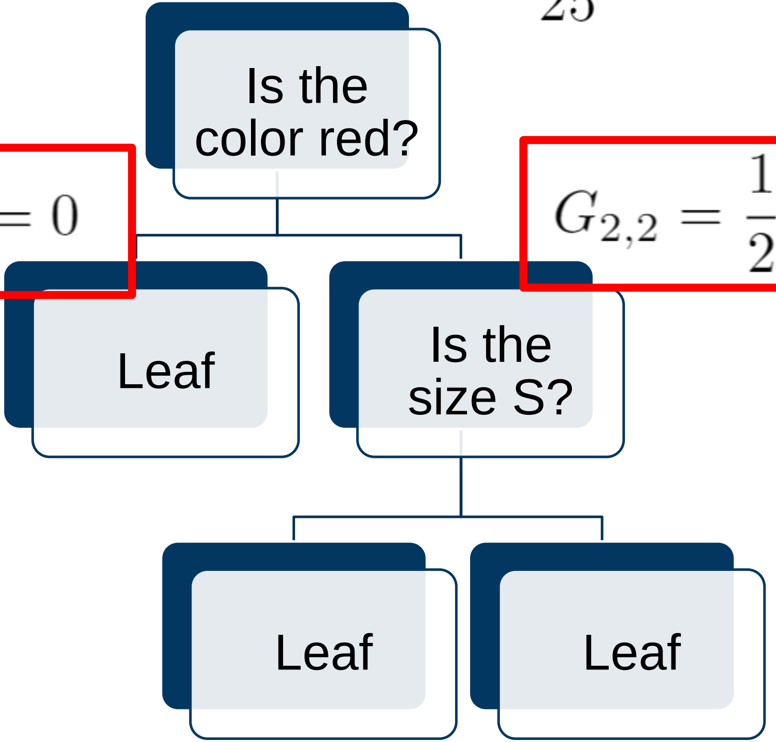
- **Information Gain**
  - How much a question reduce the uncertainties

$$IG_i^j = G_i - \boxed{G_j}$$

$$G_1 = \frac{14}{25}$$

$$\boxed{G_{2,1} = 0}$$

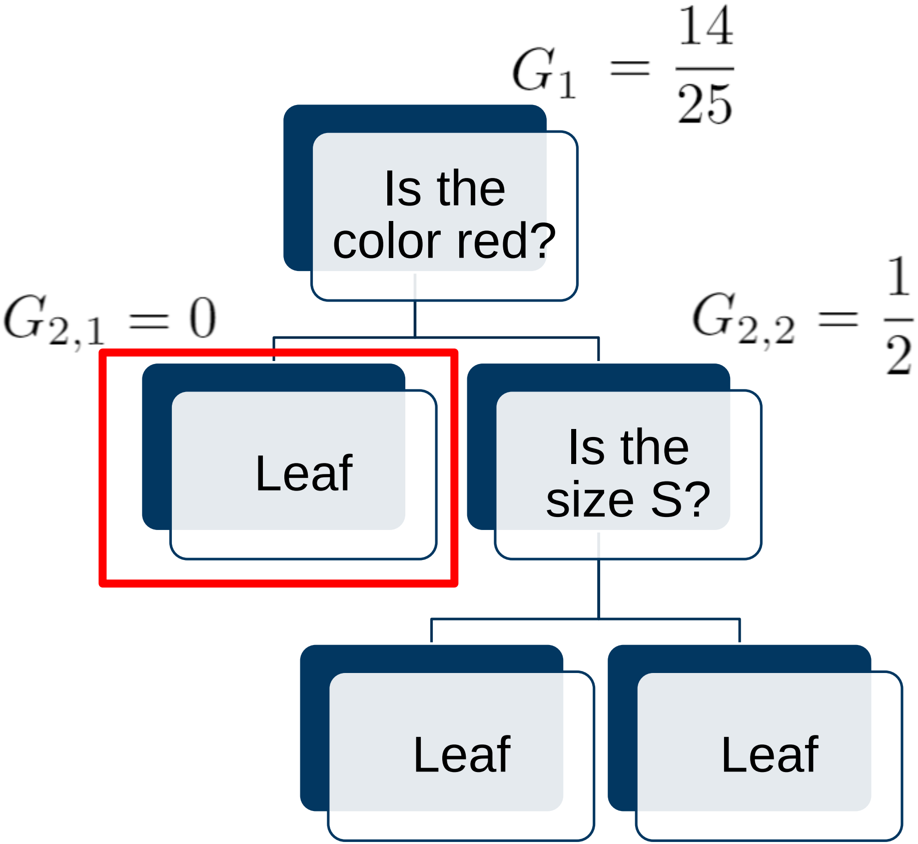
$$\boxed{G_{2,2} = \frac{1}{2}}$$



Decision Tree

- **Information Gain**
  - How much a question reduce the uncertainties

Color	Size	Discount	Label
Red	M	50%	Shirt
Red	S	50%	Shirt
Red	M	40%	Shirt

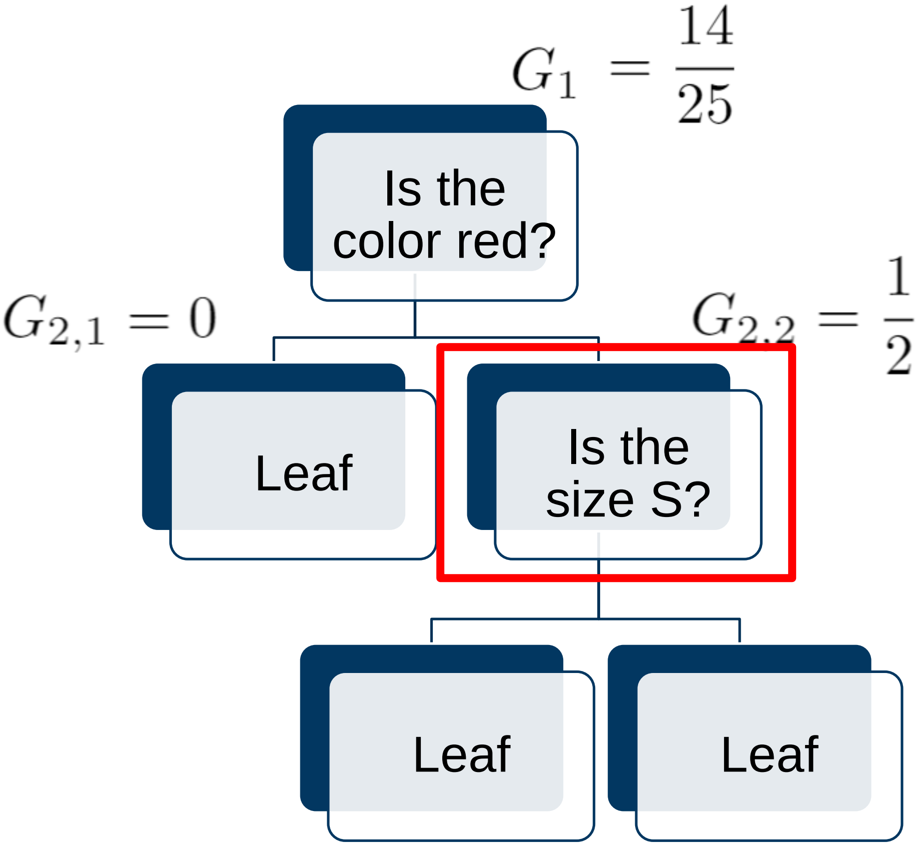


Decision Tree

- **Information Gain**
  - How much a question reduce the uncertainties

Color	Size	Discount	Label
Red	M	50%	Shirt
Red	S	50%	Shirt
Red	M	40%	Shirt

Color	Size	Discount	Label
Yellow	S	20%	Jeans
Green	L	50%	Skirt



Decision Tree

- **Information Gain**
  - How much a question reduce the uncertainties

Color	Size	Discount	Label
Red	M	50%	Shirt
Red	S	50%	Shirt
Red	M	40%	Shirt

Color	Size	Discount	Label
Yellow	S	20%	Jeans
Green	L	50%	Skirt

$$G_1 = \frac{14}{25}$$

Is the color red?

$$G_{2,1} = 0$$

$$G_{2,2} = \frac{1}{2}$$

Leaf

Is the size S?

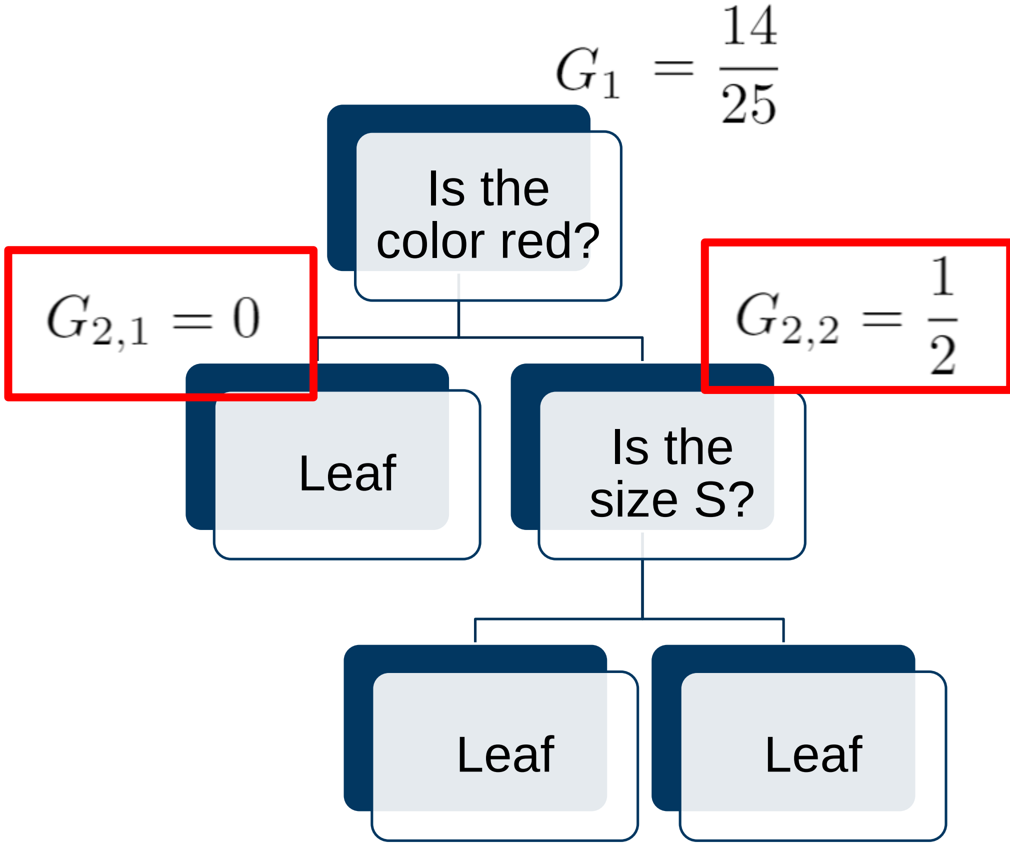
$$G_2 = G_{2,1} \cdot \boxed{\frac{3}{5}} + G_{2,2} \cdot \boxed{\frac{2}{5}} =$$

$$= 0 \cdot \frac{3}{5} + \frac{1}{2} \cdot \frac{2}{5} = \frac{1}{5}$$

Decision Tree

- **Information Gain**
  - How much a question reduce the uncertainties

$$IG_i^j = G_i - \boxed{G_j}$$

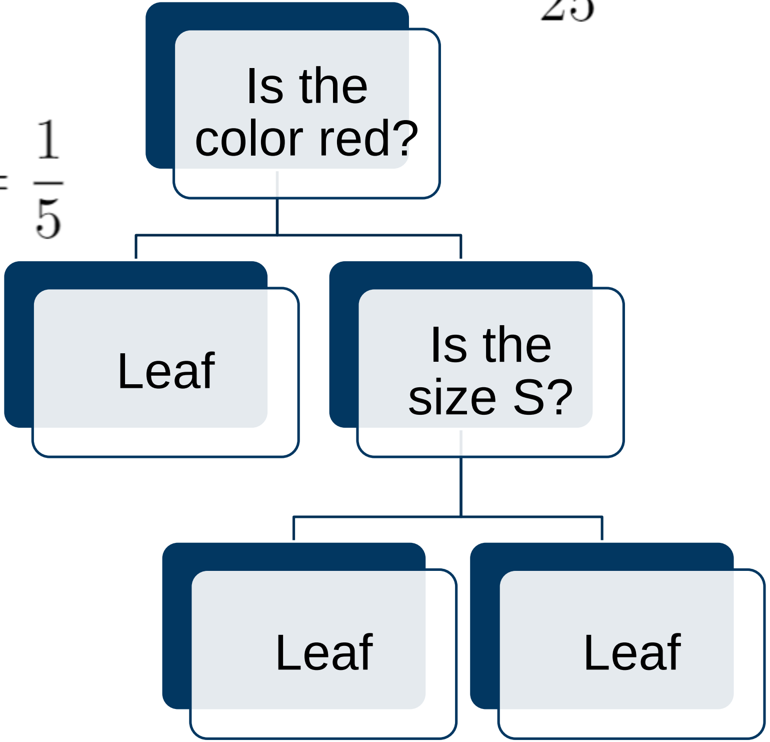


Decision Tree

- **Information Gain**
  - How much a question reduce the uncertainties

$$G_1 = \frac{14}{25}$$

$$G_2 = \frac{1}{5}$$



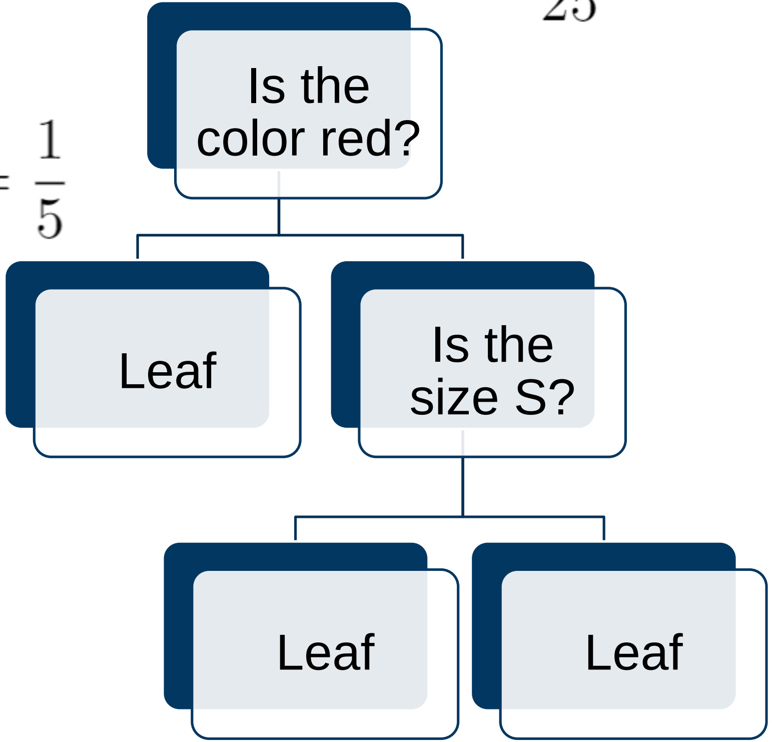
$$IG_i^j = G_i - G_j = \frac{14}{25} - \frac{1}{5} = \frac{9}{25}$$

Decision Tree

- **Information Gain**
  - How much a question reduce the uncertainties

$$G_1 = \frac{14}{25}$$

$$G_2 = \frac{1}{5}$$

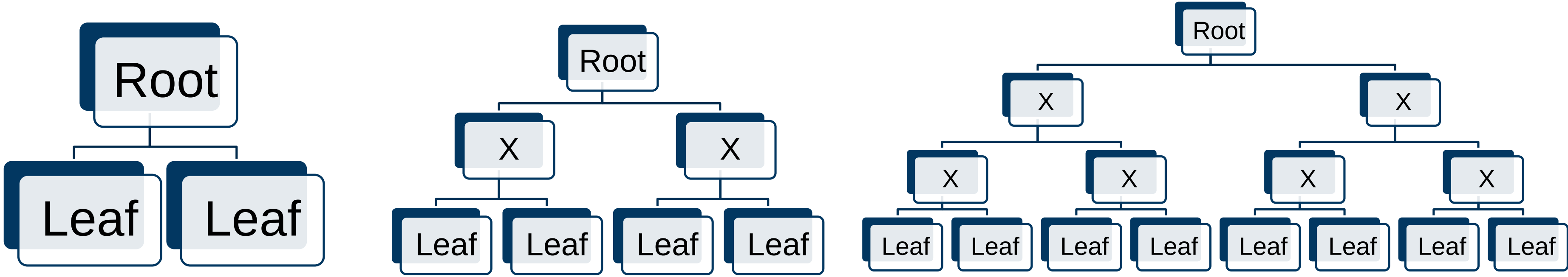


$$IG_i^j = G_i - G_j = \frac{14}{25} - \frac{1}{5} = \frac{9}{25}$$

- NB: the best feature to split is the one that maximizes the IG

Decision Tree

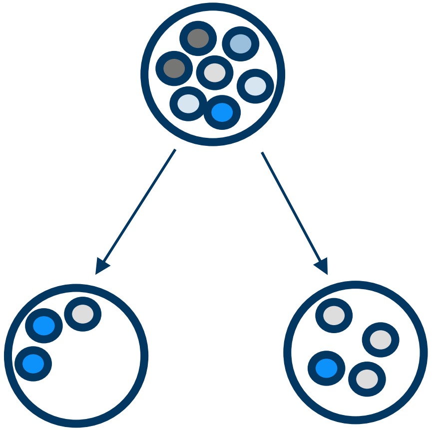
- **Hyperparameters**
  - Max depth



Depth = 1 Depth = 2 Depth = 3

Decision Tree

- **Hyperparameters**
  - Max depth
  - Minimum number of samples to split



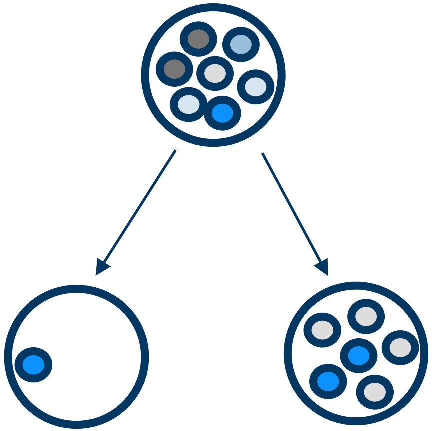
Minimum number of samples to split = 6

No Split!!!!

Minimum number of samples to split = 8

Decision Tree

- **Hyperparameters**
  - Max depth
  - Minimum number of samples to split
  - Minimum number of samples per leaf



Minimum number of samples per leaf = 1

No Split!!!!

Minimum number of samples per leaf = 7

Decision Tree

- **PROS**
  - Simple to understand and interpret
  - No need for complex data preparation
  - Handle numerical and Categorical Data
  - No assumption on the data

Decision Tree

- **PROS**
  - Simple to understand and interpret
  - No need for complex data preparation
  - Handle numerical and Categorical Data
  - No assumption on the data
- **CONS**
  - Sensitive to small data perturbation
  - Not incremental
  - No guarantee to return the globally optimal solution
  - Sensitive to unbalanced dataset
  - Overfitting
    - Pruning
    - Ensemble Methods

Decision Tree

- **Scikit-Learn**
  - What is it?
  - Decision Tree for Classification

```
from sklearn import tree
clf = tree.DecisionTreeClassifier()
clf = clf.fit(X, Y)
clf.predict([[2., 2.]])  or
clf.predict_proba([[2., 2.]])
tree.plot_tree(clf)
```

  - Decision Tree for Regression

```
from sklearn import tree
clf = tree.DecisionTreeRegressor()
clf = clf.fit(X, Y)
clf.predict([[2., 2.]])
```

What Ensemble Techniques Are

Different types of Ensemble Techniques: Overview

Background: Decision Tree

Random Forest

ADA Boost

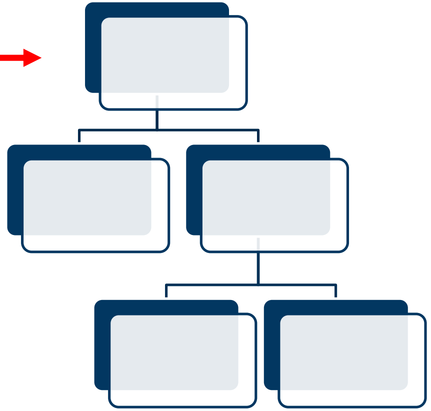
Random Forest

- Boosting method
  - Regression
  - Classification
- `Weak Learner': decision tree
- Ensuring Models Diversity
  – Feature randomness

Random Forest

- Boosting method
  - Regression
  - Classification
- `Weak Learner': decision tree
- Ensuring Models Diversity
  - Feature randomness

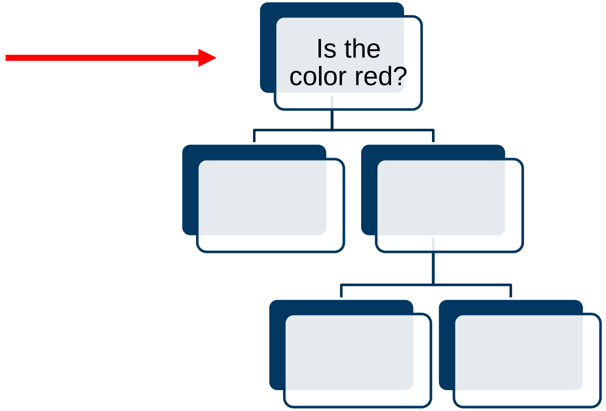
Color	Size	Discount	Label
Red	M	50%	Shirt
Yellow	S	20%	Jeans
Red	S	50%	Shirt
Red	M	40%	Shirt
Green	L	50%	Skirt



Random Forest

- Boosting method
  - Regression
  - Classification
- `Weak Learner’: decision tree
- Ensuring Models Diversity
  - Feature randomness

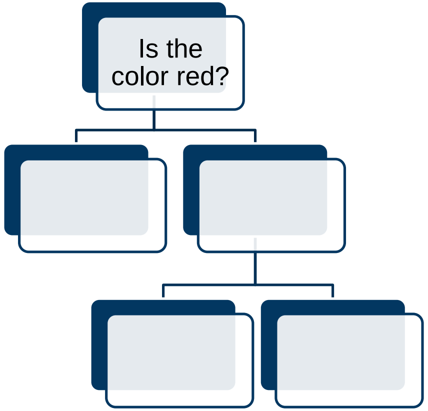
Color	Size	Discount	Label
Red	M	50%	Shirt
Yellow	S	20%	Jeans
Red	S	50%	Shirt
Red	M	40%	Shirt
Green	L	50%	Skirt



Random Forest

- Boosting method
  - Regression
  - Classification
- `Weak Learner': decision tree
- Ensuring Models Diversity
  - Feature randomness

Color	Size	Discount	Label
Red	M	50%	Shirt
Yellow	S	20%	Jeans
Red	S	50%	Shirt
Red	M	40%	Shirt
Green	L	50%	Skirt

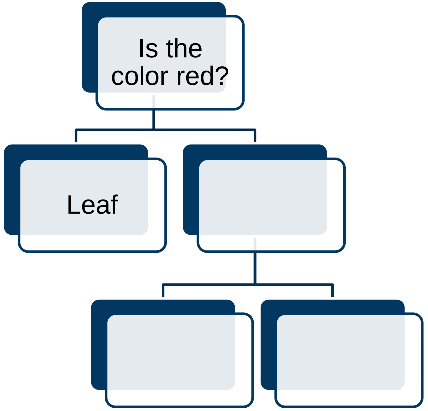


Reimagining Medicine

Random Forest

- Boosting method
  - Regression
  - Classification
- `Weak Learner’: decision tree
- Ensuring Models Diversity
  – Feature randomness

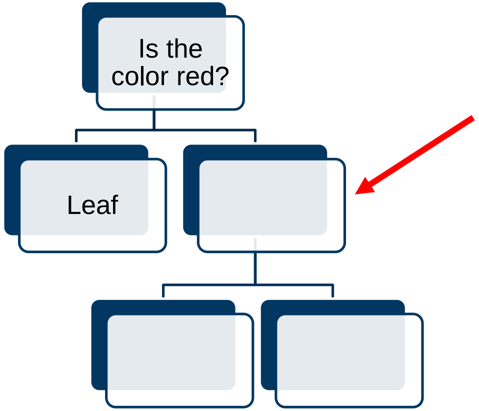
Color	Size	Discount	Label
Red	M	50%	Shirt
Yellow	S	20%	Jeans
Red	S	50%	Shirt
Red	M	40%	Shirt
Green	L	50%	Skirt



Random Forest

- Boosting method
  - Regression
  - Classification
- `Weak Learner': decision tree
- Ensuring Models Diversity
  – Feature randomness

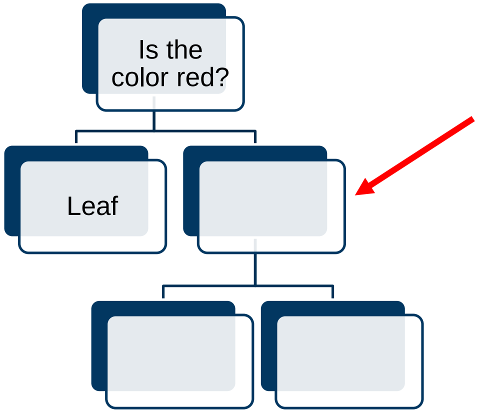
Color	Size	Discount	Label
Red	M	50%	Shirt
Yellow	S	20%	Jeans
Red	S	50%	Shirt
Red	M	40%	Shirt
Green	L	50%	Skirt



Random Forest

- Boosting method
  - Regression
  - Classification
- `Weak Learner': decision tree
- Ensuring Models Diversity
  – Feature randomness

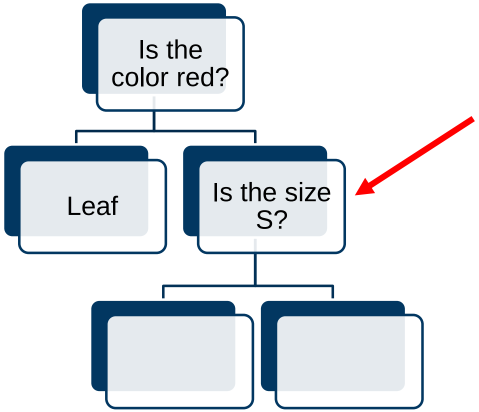
Size	Discount	Label
S	20%	Jeans
L	50%	Skirt



Random Forest

- Boosting method
  - Regression
  - Classification
- `Weak Learner': decision tree
- Ensuring Models Diversity
  - Feature randomness

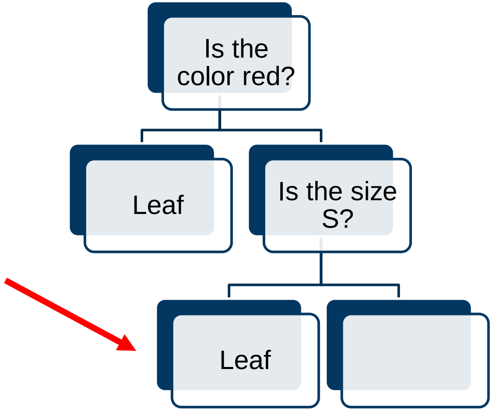
Size	Discount	Label
S	20%	Jeans
L	50%	Skirt



Random Forest

- Boosting method
  - Regression
  - Classification
- `Weak Learner': decision tree
- Ensuring Models Diversity
  – Feature randomness

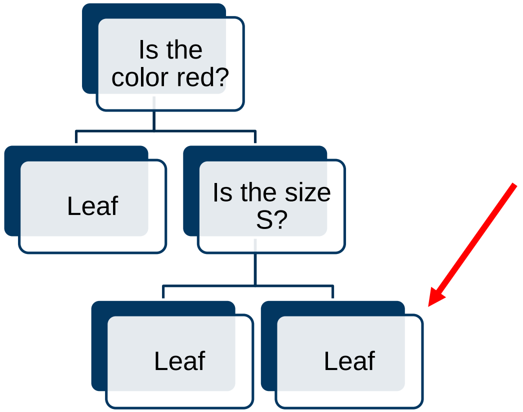
Size	Discount	Label
S	20%	Jeans
L	50%	Skirt



Random Forest

- Boosting method
  - Regression
  - Classification
- `Weak Learner’: decision tree
- Ensuring Models Diversity
  – Feature randomness

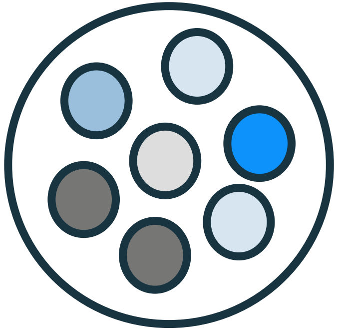
Size	Discount	Label
S	20%	Jeans
L	50%	Skirt



Random Forest

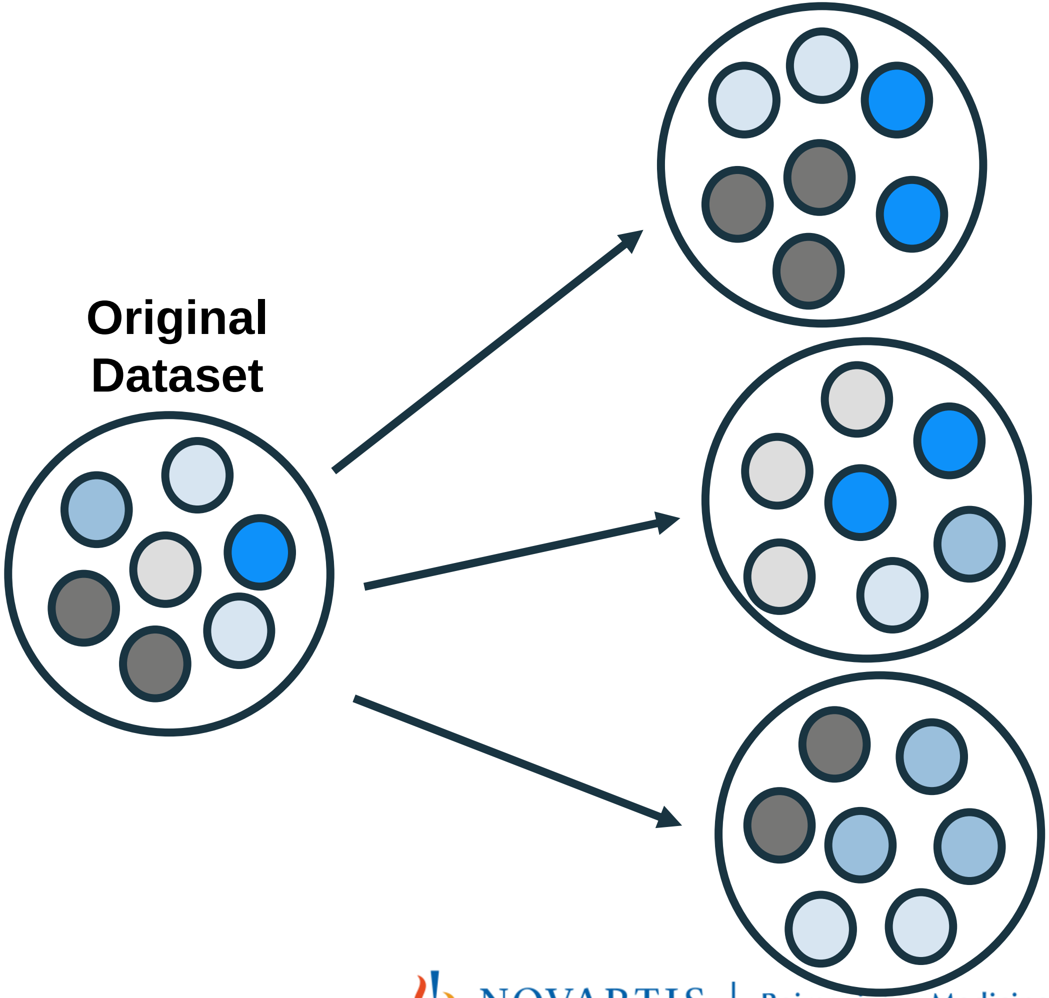
- Boosting method
  - Regression
  - Classification
- `Weak Learner': decision tree
- Ensuring Models Diversity
  - Feature randomness
  - Bagging (Bootstrap Aggregation)

Original Dataset



Random Forest

- Boosting method
  - Regression
  - Classification
- `Weak Learner’: decision tree
- Ensuring Models Diversity
  - Feature randomness
  - Bagging (Bootstrap Aggregation)



NOVARTIS | Reimagining Medicine

Random Forest

- Pseudo-code:
  - Create a “**bootstrapped**” dataset

Color	Size	Discount	Label
Red	M	50%	Shirt
Yellow	S	20%	Jeans
Red	S	50%	Shirt
Red	M	40%	Shirt
Green	L	50%	Skirt

Random Forest

- Pseudo-code:
  - Create a “**bootstrapped**” dataset

	Color	Size	Discount	Label
	Red	M	50%	Shirt
	Yellow	S	20%	Jeans
	Red	S	50%	Shirt
	Red	M	40%	Shirt
x2	Green	L	50%	Skirt

Random Forest

- Pseudo-code:
  - Create a “**bootstrapped**” dataset

	Color	Size	Discount	Label
	Red	M	50%	Shirt
	Yellow	S	20%	Jeans
	Red	S	50%	Shirt
	Red	M	40%	Shirt
x2	Green	L	50%	Skirt

Random Forest

- Pseudo-code:
  - Create a “**bootstrapped**” dataset

Color	Size	Discount	Label
Red	M	50%	Shirt
Red	S	50%	Shirt
Red	M	40%	Shirt
Green	L	50%	Skirt
Green	L	50%	Skirt

	Color	Size	Discount	Label
	Red	M	50%	Shirt
	Yellow	S	20%	Jeans
	Red	S	50%	Shirt
	Red	M	40%	Shirt
x2	Green	L	50%	Skirt

Random Forest

- Pseudo-code:
  - Create a “**bootstrapped**” dataset
    - Randomly select a subset of the features set

Color	Size	Discount	Label
Red	M	50%	Shirt
Red	S	50%	Shirt
Red	M	40%	Shirt
Green	L	50%	Skirt
Green	L	50%	Skirt

Random Forest

- Pseudo-code:
  - Create a “**bootstrapped**” dataset
    - Randomly select a subset of the features set
    - Compute the best feature where to split

Color	Size	Discount	Label
Red	M	50%	Shirt
Red	S	50%	Shirt
Red	M	40%	Shirt
Green	L	50%	Skirt
Green	L	50%	Skirt

Random Forest

- Pseudo-code:
  - Create a “**bootstrapped**” dataset
    - Randomly select a subset of the features set
    - Compute the best feature where to split

Color	Size	Discount	Label
Red	M	50%	Shirt
Red	S	50%	Shirt
Red	M	40%	Shirt
Green	L	50%	Skirt
Green	L	50%	Skirt

Is the color red?

Random Forest

- Pseudo-code:
  - Create a “**bootstrapped**” dataset
    - Randomly select a subset of the features set
    - Compute the best feature where to split
    - Split dataset

Color	Size	Discount	Label
Red	M	50%	Shirt
Red	S	50%	Shirt
Red	M	40%	Shirt

yes

Is the color red?

Random Forest

- Pseudo-code:
  - Create a “**bootstrapped**” dataset
    - Randomly select a subset of the features set
    - Compute the best feature where to split
    - Split dataset

Color	Size	Discount	Label
Red	M	50%	Shirt
Red	S	50%	Shirt
Red	M	40%	Shirt

yes

Is the color red?

no

Color	Size	Discount	Label
Green	L	50%	Skirt
Green	L	50%	Skirt

Random Forest

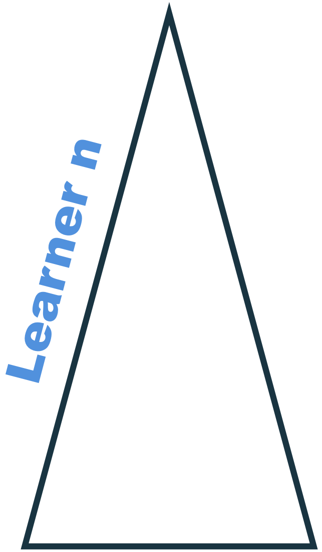
- Pseudo-code:
  - Create a “**bootstrapped**” dataset
    - Randomly select a subset of the features set
    - Compute the best feature where to split
    - Split dataset
    - If not stop condition:

Random Forest

- Pseudo-code:
  - Create a “**bootstrapped**” dataset
    - Randomly select a subset of the features set
    - Compute the best feature where to split
    - Split dataset
    - If not stop condition:
      - Repeat

Random Forest

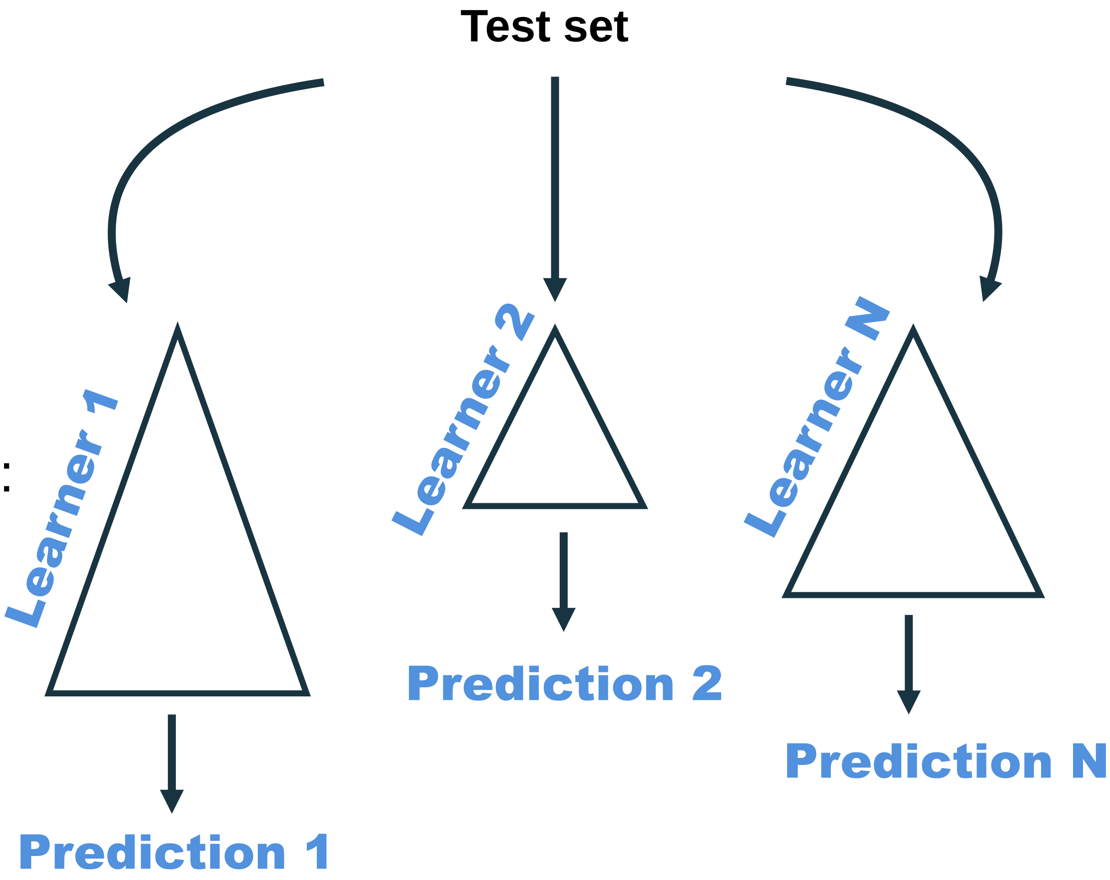
- Pseudo-code:
  - Create a “**bootstrapped**” dataset
    - Randomly select a subset of the features set
    - Compute the best feature where to split
    - Split dataset
    - If not stop condition:
      - Repeat



NOVARTIS | Reimagining Medicine

Random Forest

- Check Performances
  - New samples
    - Averaging
    - Voting
  - Hyperparameter optimisation:
    - Grid search
    - Random search
    - Etc..



Random Forest

- Hyperparameters:
  - Number of learners
  - Max depth
  - Minimum number of samples to split
  - Minimum number of samples per leaf

Random Forest

- **PROS:**
  - Very popular in practice
  - Easy to implement
  - Parallelizes easily
  - Robustness

- **CONS:**
  - Very High number of trees
  - Computation process slower

Random Forest

- Scikit-Learn
  - Random Forest for Classification

```
from sklearn.ensemble import RandomForestClassifier
clf=RandomForestClassifier(n_estimators = 100, random_state=0)
clf = clf.fit(X, Y)
clf.predict([[2., 2.]])  or
clf.predict_proba([[2., 2.]])
```

  - Random Forest for Regression

```
from sklearn.ensemble import RandomForestRegressor
clf=RandomForestRegressor(n_estimators = 100, random_state=0)
clf = clf.fit(X, Y)
clf.predict([[2., 2.]])
```

What Ensemble Techniques Are

Different types of Ensemble Techniques: Overview

Background: Decision Tree

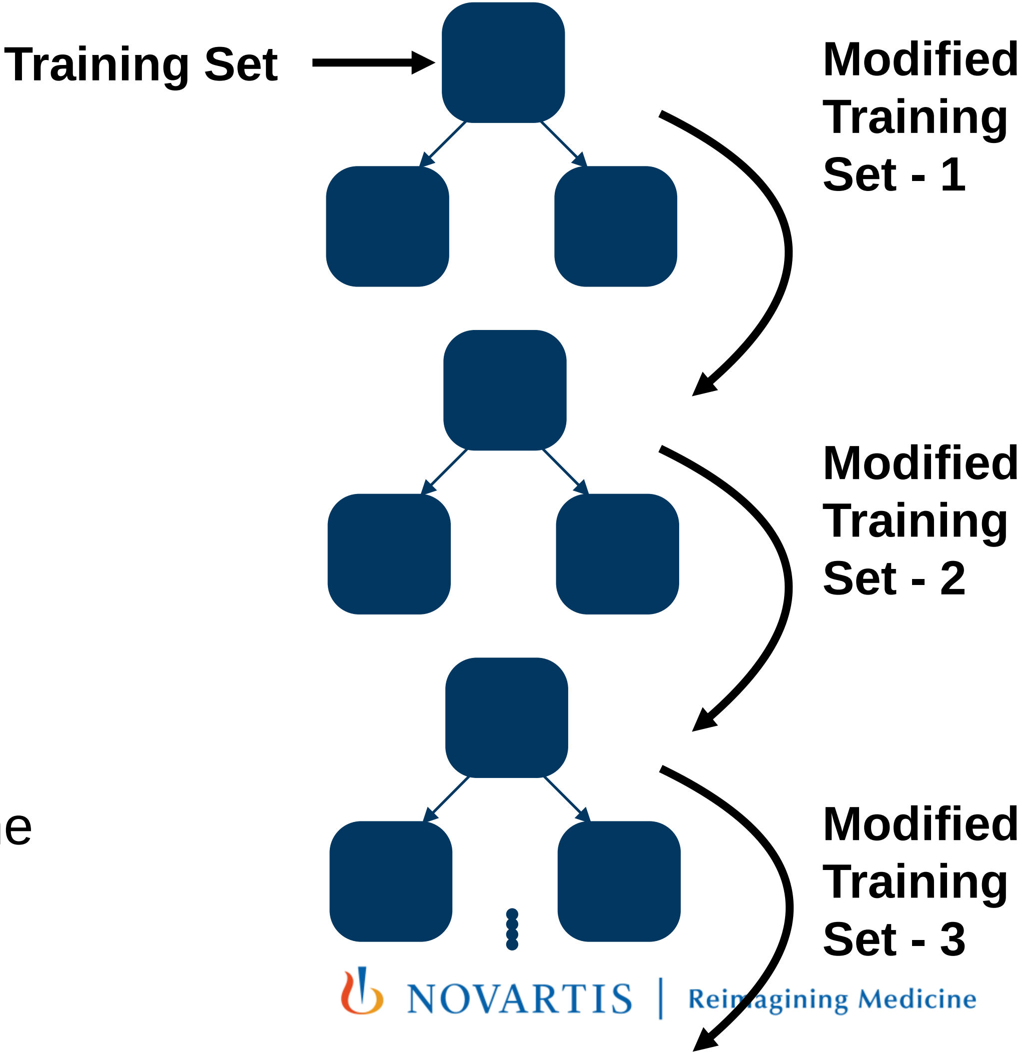
Random Forest

ADA Boost

NOVARTIS | Reimagining Medicine

Adaboost

- Boosting method
  - Regression
  - Classification
- `Weak Learner': stump
- Boosting
  - Weak learners are trained in series
  - Errors made by one stumps influence the input dataset of the next stump



Adaboost

- Pseudocode:

Adaboost

- Pseudocode:
  - Assign to each sample the same weight

Is this a Shirt?

Color	Size	Discount	Label
Red	M	50%	Shirt
Yellow	S	20%	Jeans
Red	S	50%	Shirt
Green	M	40%	Shirt
Red	L	50%	Jeans

Adaboost

- Pseudocode:
  - Assign to each sample the same weight

Is this a Shirt?

Weight	Color	Size	Discount	Label
1/5	Red	M	50%	Shirt
1/5	Yellow	S	20%	Jeans
1/5	Red	S	50%	Shirt
1/5	Green	M	40%	Shirt
1/5	Red	L	50%	Jeans

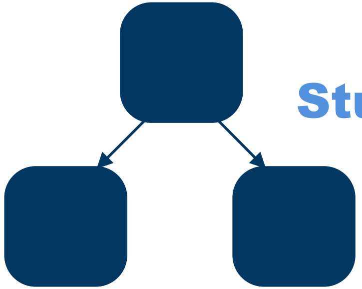
Adaboost

- Pseudocode:
  - Assign to each sample the same weight
  - Select feature that gives the best split

Is this a Shirt?

Weight	Color	Size	Discount	Label
1/5	Red	M	50%	Shirt
1/5	Yellow	S	20%	Jeans
1/5	Red	S	50%	Shirt
1/5	Green	M	40%	Shirt
1/5	Red	L	50%	Jeans

Is it the color red?



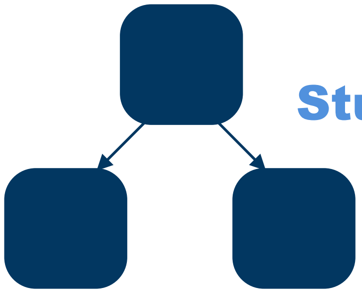
Stump 1

Adaboost

- Pseudocode:
  - Assign to each sample the same weight
  - Select feature that gives the best split
  - Determine how *much to say* the stump has

Weight	Color	Size	Discount	Label
1/5	Red	M	50%	Shirt
1/5	Yellow	S	20%	Jeans
1/5	Red	S	50%	Shirt
1/5	Green	M	40%	Shirt
1/5	Red	L	50%	Jeans

Is it the color red?



Stump 1

Adaboost

- Pseudocode:
  - Assign to each sample the same weight
  - Select feature that gives the best split
  - Determine how *much to say* the stump has
    - Compute the total error

Weight	Color	Size	Discount	Label	
1/5	Red	M	50%	Shirt	yes
1/5	Yellow	S	20%	Jeans	
1/5	Red	S	50%	Shirt	yes
1/5	Green	M	40%	Shirt	
1/5	Red	L	50%	Jeans	yes

Is it the color red?

Total Error = sum of weight of incorrectly classified samples

1 incorrectly classified sample

Adaboost

- Pseudocode:
  - Assign to each sample the same weight
  - Select feature that gives the best split
  - Determine how *much to say* the stump has
    - Compute the total error

Weight	Color	Size	Discount	Label	
1/5	Red	M	50%	Shirt	
1/5	Yellow	S	20%	Jeans	no
1/5	Red	S	50%	Shirt	
1/5	Green	M	40%	Shirt	no
1/5	Red	L	50%	Jeans	

Is it the color red?

Total Error = sum of weight of incorrectly classified samples

1 incorrectly classified sample

Adaboost

- Pseudocode:
  - Assign to each sample the same weight
  - Select feature that gives the best split
  - Determine how *much to say* the stump has
    - Compute the total error

Weight	Color	Size	Discount	Label
1/5	Red	M	50%	Shirt
1/5	Yellow	S	20%	Jeans
1/5	Red	S	50%	Shirt
1/5	Green	M	40%	Shirt
1/5	Red	L	50%	Jeans

Is it the color red?

Total Error = sum of weight of incorrectly classified samples → 2/5

Adaboost

- Pseudocode:
  - Assign to each sample the same weight
  - Select feature that gives the best split
  - Determine how *much to say* the stump has
    - Compute the total error

Weight	Color	Size	Discount	Label
1/5	Red	M	50%	Shirt
1/5	Yellow	S	20%	Jeans
1/5	Red	S	50%	Shirt
1/5	Green	M	40%	Shirt
1/5	Red	L	50%	Jeans

Amount of say =

$$Amount of Say = \frac{1}{2} log(\frac{1 - TotalError}{TotalError})$$

Adaboost

- Pseudocode:
  - Assign to each sample the same weight
  - Select feature that gives the best split
  - Determine how *much to say* the stump has
    - Compute the total error

Weight	Color	Size	Discount	Label
1/5	Red	M	50%	Shirt
1/5	Yellow	S	20%	Jeans
1/5	Red	S	50%	Shirt
1/5	Green	M	40%	Shirt
1/5	Red	L	50%	Jeans

Amount of say =

$$AmountofSay = \frac{1}{2}log(\frac{1-\boxed{TotalError}}{\boxed{TotalError}})$$

2/5

Adaboost

- Pseudocode:
  - Assign to each sample the same weight
  - Select feature that gives the best split
  - Determine how *much to say* the stump has
    - Compute the total error

Weight	Color	Size	Discount	Label
1/5	Red	M	50%	Shirt
1/5	Yellow	S	20%	Jeans
1/5	Red	S	50%	Shirt
1/5	Green	M	40%	Shirt
1/5	Red	L	50%	Jeans

Amount of say =

$$AmountofSay = \frac{1}{2}log(\frac{1 - TotalError}{TotalError}) \longrightarrow 0.088$$

Adaboost

- Pseudocode:
  - Assign to each sample the same weight
  - Select feature that gives the best split
  - Determine how *much to say* the stump has
    - Compute the total error
  - Change the sample weights

Weight	Color	Size	Discount	Label
1/5	Red	M	50%	Shirt
1/5	Yellow	S	20%	Jeans
1/5	Red	S	50%	Shirt
1/5	Green	M	40%	Shirt
1/5	Red	L	50%	Jenas

Adaboost

- Pseudocode:
  - Assign to each sample the same weight
  - Select feature that gives the best split
  - Determine how *much to say* the stump has
    - Compute the total error
  - Change the sample weights

Weight	Color	Size	Discount	Label
1/5	Red	M	50%	Shirt
1/5	Yellow	S	20%	Jeans
1/5	Red	S	50%	Shirt
0.218	Green	M	40%	Shirt
0.218	Red	L	50%	Jeans

$$NewSampleWeight = SampleWeight \cdot e^{AmountofSay}$$

NOVARTIS | Reimagining Medicine

Adaboost

- Pseudocode:
  - Assign to each sample the same weight
  - Select feature that gives the best split
  - Determine how *much to say* the stump has
    - Compute the total error
  - Change the sample weights

Weight	Color	Size	Discount	Label
0.183	Red	M	50%	Shirt
0.183	Yellow	S	20%	Jeans
0.183	Red	S	50%	Shirt
0.218	Green	M	40%	Shirt
0.218	Red	L	50%	Jeans

$$NewSampleWeight = SampleWeight \cdot e^{AmountofSay}$$

$$NewSampleWeight = SampleWeight \cdot e^{-AmountofSay}$$

Adaboost

- Pseudocode:
  - Assign to each sample the same weight
  - Select feature that gives the best split
  - Determine how *much to say* the stump has
    - Compute the total error
  - Change the sample weights
  - Normalize the weights

Weight	Color	Size	Discount	Label
0.185	Red	M	50%	Shirt
0.185	Yellow	S	20%	Jeans
0.185	Red	S	50%	Shirt
0.221	Green	M	40%	Shirt
0.221	Red	L	50%	Jeans

Adaboost

- Pseudocode:
  - Assign to each sample the same weight
  - Select feature that gives the best split
  - Determine how *much to say* the stump has
    - Compute the total error
  - Change the sample weights
  - Normalize the weights
  - Create a new dataset considering the sample weight

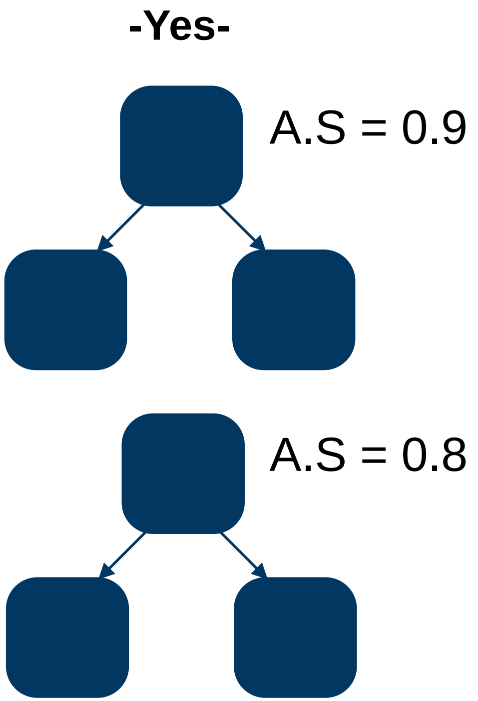
Weight	Color	Size	Discount	Label
0.185	Red	M	50%	Shirt
0.185	Yellow	S	20%	Jeans
0.185	Red	S	50%	Shirt
0.221	Green	M	40%	Shirt
0.221	Red	L	50%	Jeans

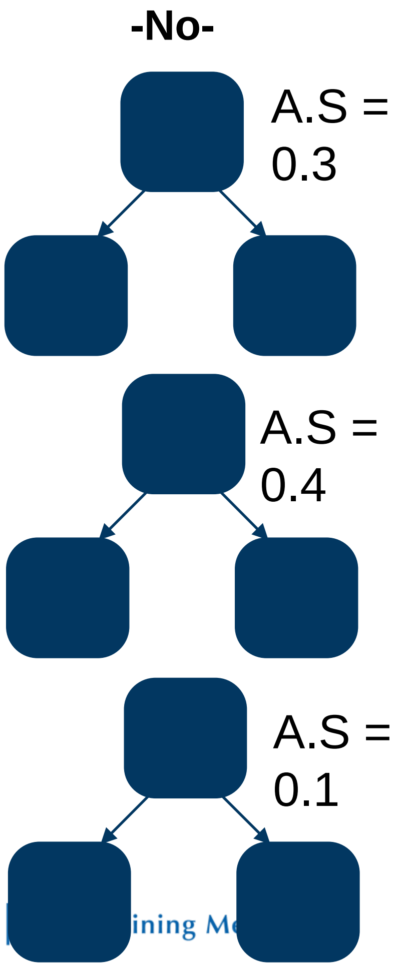
Weight	Color	Size	Discount	Label
0.221	Red	L	50%	Jeans
0.221	Red	L	50%	Jeans
0.221	Green	M	40%	Shirt
0.221	Green	M	40%	Shirt
0.185	Red	M	50%	Shirt

Adaboost

Is the answer yes or not?

- Given a new sample, check the answer of each stump
- Group stumps with the same answer
- Compute the total Amount of Say





-Yes-

Total A.S (“Yes”) = 1.7

Total A.S (“No”) = 0.8

Adaboost

- **PROS:**
  - Easy to implement
  - Improve accuracy of the weak learners
  - Not prone to overfitting

- **CONS:**
  - Sensitive to noise data
  - Affected by outliers

Adaboost

- Scikit-Learn
  - Adaboost for <u>Classification</u>

```
from sklearn.ensemble import AdaBoostClassifier
clf=AdaBoostClassifier(n_estimators = 100, random_state=0)
clf = clf.fit(X, Y)
clf.predict([[2., 2.]])  or
clf.predict_proba([[2., 2.]])
```

  - Adaboost for <u>Regression</u>

```
from sklearn.ensemble import AdaBoostRegressor
clf=AdaBoostRegressor(n_estimators = 100, random_state=0)
clf = clf.fit(X, Y)
clf.predict([[2., 2.]])
```

References

[1] Scikit-learn: Machine Learning in Python, Pedregosa *et al.*, JMLR 12, pp. 2825-2830, 2011.
[2] https://www.kdnuggets.com/2019/01/ensemble-learning-5-main-approaches.html
[3] Géron, Aurélien. *Hands-on machine learning with Scikit-Learn and TensorFlow: concepts, tools, and techniques to build intelligent systems*. " O'Reilly Media, Inc.", 2017.
[4] https://www.datacamp.com/community/tutorials/adaboost-classifier-python
[5] https://julienbeaulieu.gitbook.io/wiki/sciences/machine-learning/decision-trees
[6] https://julienbeaulieu.gitbook.io/wiki/sciences/machine-learning/decision-trees
[7] https://dev.to/nexttech/classification-and-regression-analysis-with-decision-trees-jgp
[8] https://www.kdnuggets.com/2019/01/ensemble-learning-5-main-approaches.html
[9] https://medium.com/ml-research-lab/bagging-ensemble-meta-algorithm-for-reducing-variance-c98fffa5489f
[10] https://www.kdnuggets.com/2019/01/ensemble-learning-5-main-approaches.html

Thank you

NOVARTIS | Reimagining Medicine

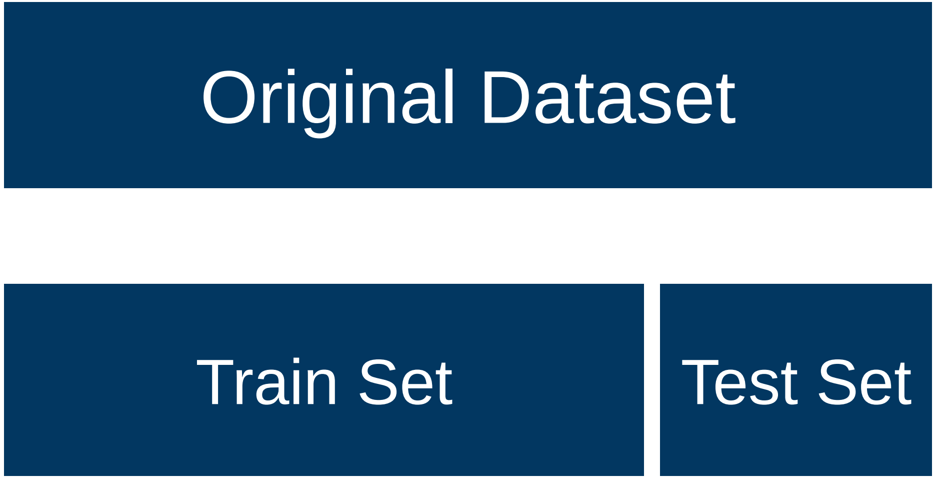
Mini-Project : Predicting Daily Bike Rentals

The project will be structured as follows:

1) Analysis and Data Preparation
2) Definition of useful functions (e.g. for **cross-validation, test/train split, computation of metrics of interest** etc.)
3) Models Evaluation
4) Analysis of features importance
5) Robustness Analysis
6) Hyperparameter Optimization

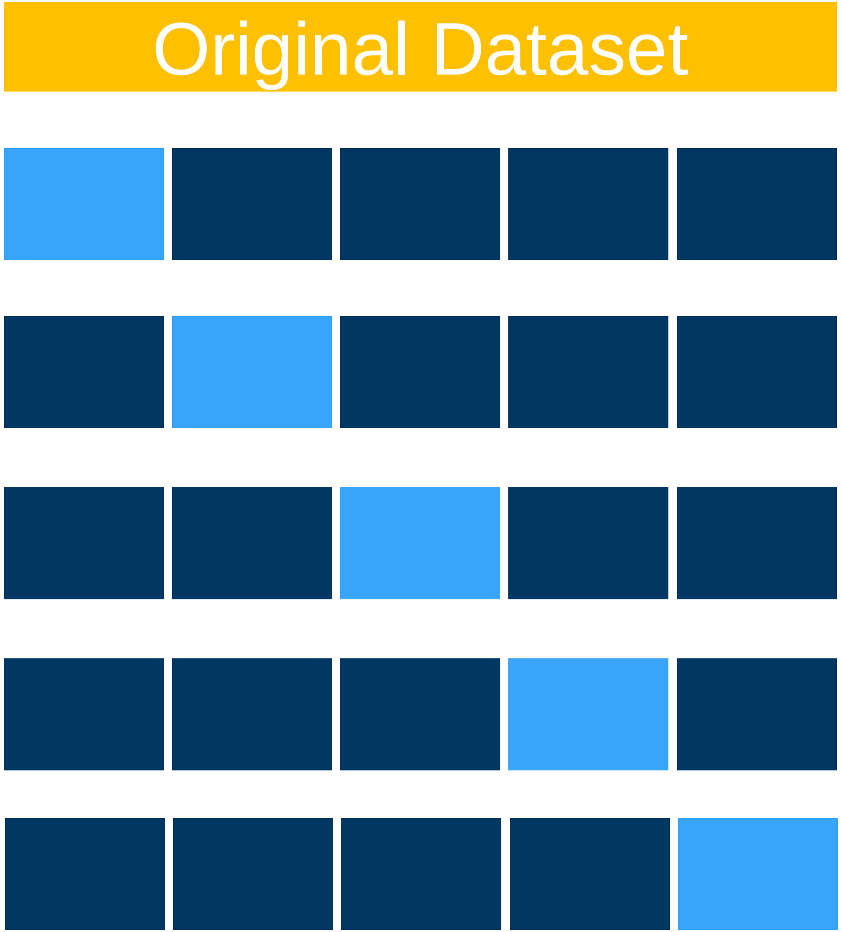
Test/Train split

- Big mistake: train and test the model on the same data
  - The model memorize the data
  - The model may not able to generalize (overfitting)
- Good practice: hold out part of the original data as **test set**
- A common split is 80% - 20%
- Main problem: dependency on the specific train-test split



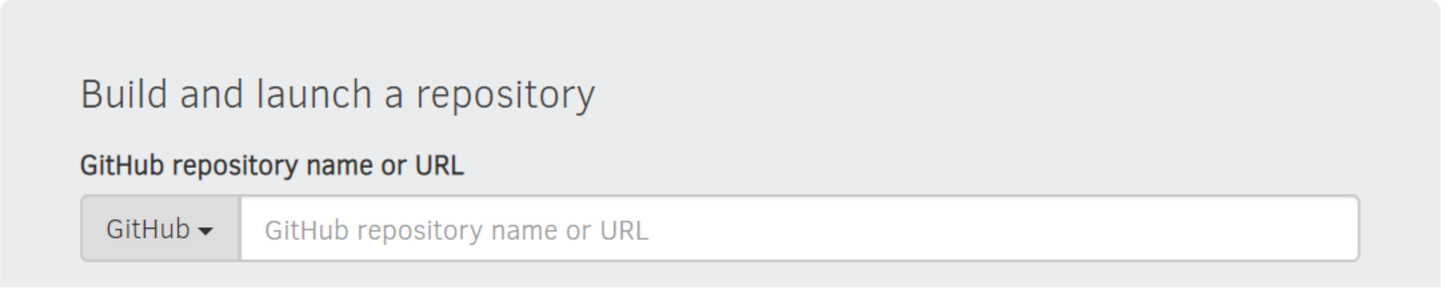
K-fold Cross Validation

- Statistical method to estimate model performances on unseen data
- Used to select and compare different models
- Main idea:
  - Split dataset into k sets
  - Use the k-1 folds to train the model
  - Use the remaining kth fold to validate the model
  - Repeat the procedure for each kth set
  - Average the performances



Setup your environment

- Go to: https://mybinder.org/
- Copy my github repository link (https://github.com/Jlanini/AMLD_2020) in this section:



- Launch!