Introduction to Machine Learning

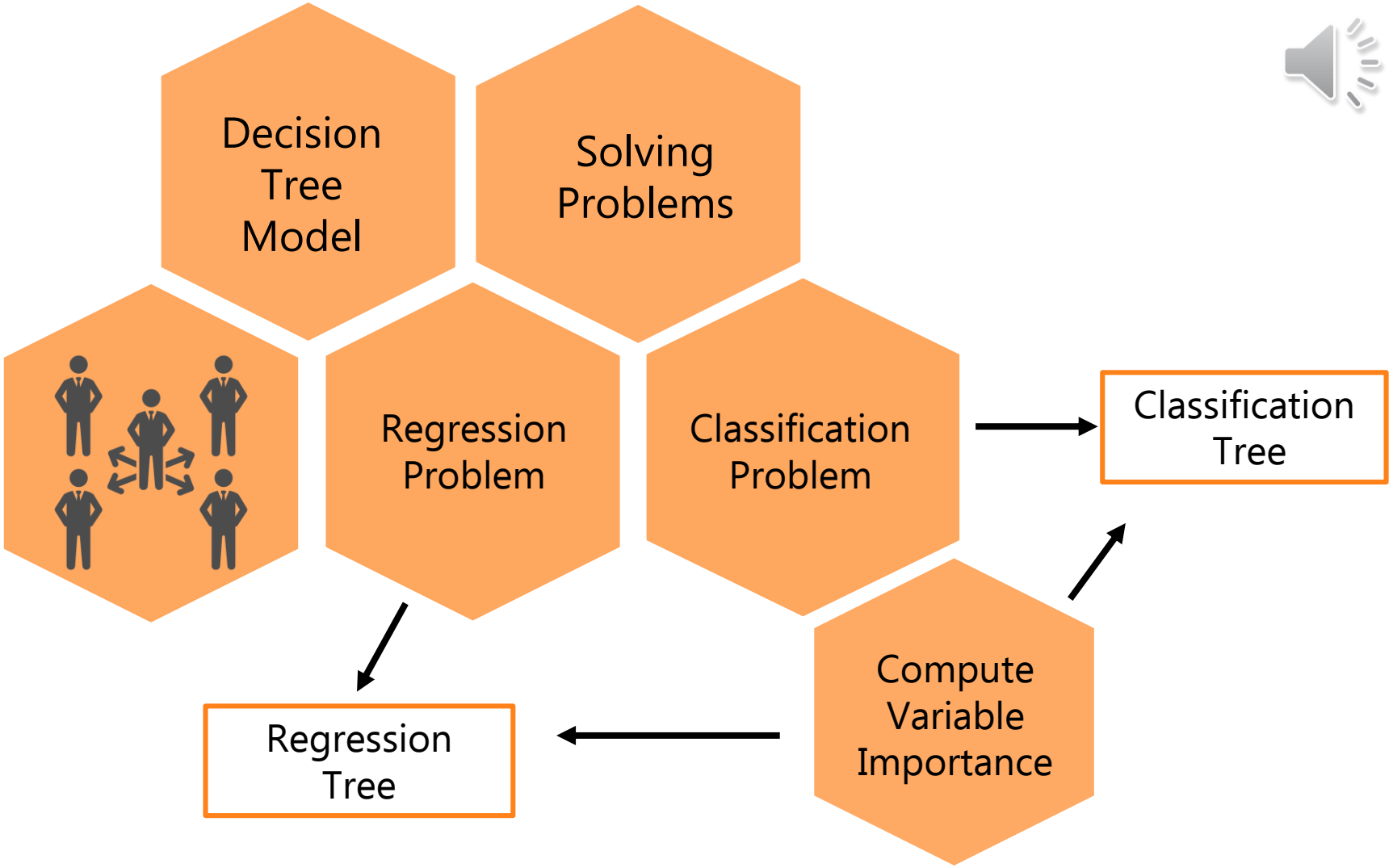
Class

Tree Based Models

Topic

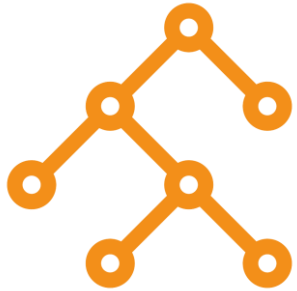
Introduction to Classification Trees

Agenda



Decision Tree: Overview

Solve both regression and classification problems



Decision Tree works is based on a branch of computer science known as **Information Theory**

The classic use case of decision trees is analysis of segments in business data

Decision Tree

Existing Data of a Bank

Customer	Age	Gender	Marital Status	# cr. Cards	Profitability
1	36	M	M	1	P
2	32	M	S	3	U
3	38	M	M	2	P
4	40	M	S	1	U
5	44	M	M	0	P
6	56	F	M	0	P
7	58	F	S	1	U
8	30	F	S	2	P
9	28	F	M	1	U
10	26	F	M	0	U

Profitable

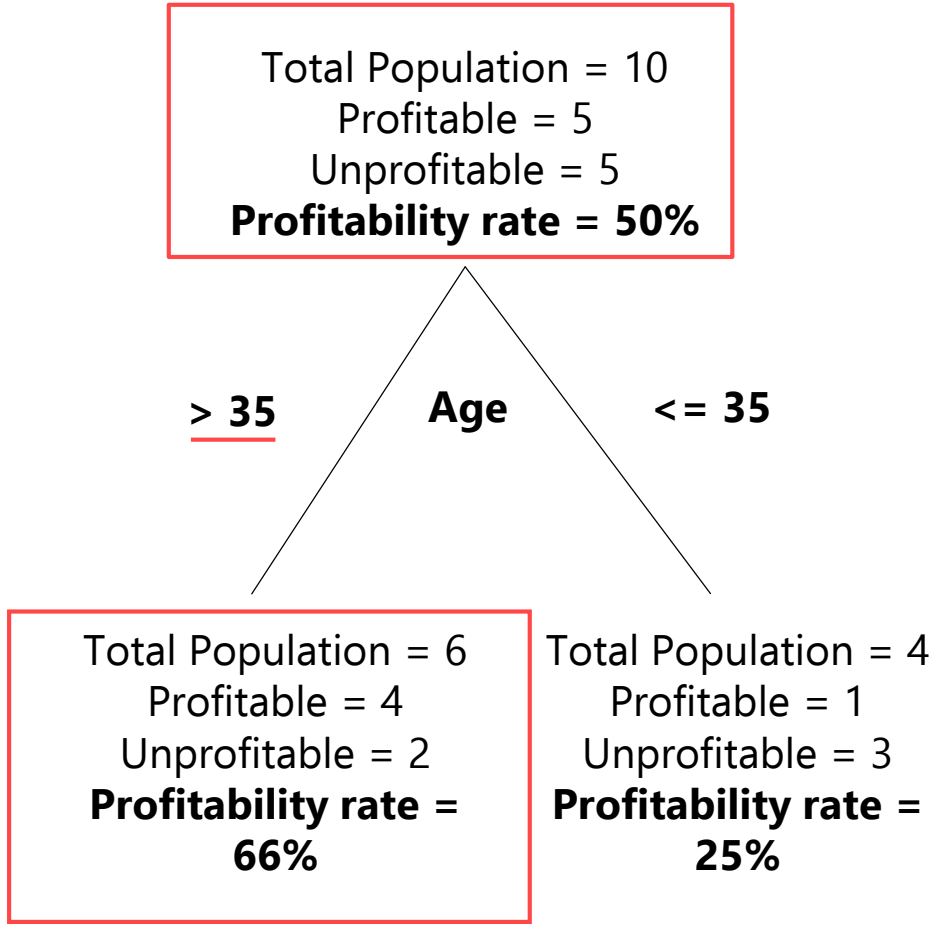
Unprofitable

To build a predictive model classifying customers logistic, Regression Classifier can be used

Decision Tree

Existing Data of a Bank

Customer	Age	Gender	Marital Status	# cr. Cards	Profitability
1	36	M	M	1	P
2	32	M	S	3	U
3	38	M	M	2	P
4	40	M	S	1	U
5	44	M	M	0	P
6	56	F	M	0	P
7	58	F	S	1	U
8	30	F	S	2	P
9	28	F	M	1	U
10	26	F	M	0	U



Decision Tree

Total Population = 10
Profitable = 5
Unprofitable = 5
Profitability rate = 50%

> 35 **Age** **<= 35**

Total Population = 6
Profitable = 4
Unprofitable = 2
Profitability rate = 66%

Total Population = 4
Profitable = 1
Unprofitable = 3
Profitability rate = 25%

The segment of data which is >35 has a higher chance of seeing a profitable customer

Decision Tree

Total Population = 10
Profitable = 5
Unprofitable = 5
Profitability rate = 50%

> 35

Age

<= 35

Total Population = 6
Profitable = 4
Unprofitable = 2
Profitability rate = 66%

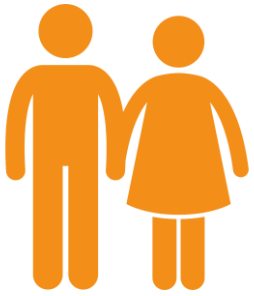
Total Population = 4
Profitable = 1
Unprofitable = 3
Profitability rate = 25%

Profitable

Married

Marital Status

Single

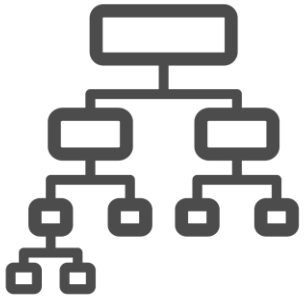


Total Population = 4
Profitable = 4
Unprofitable = 2
Profitability rate = 100%

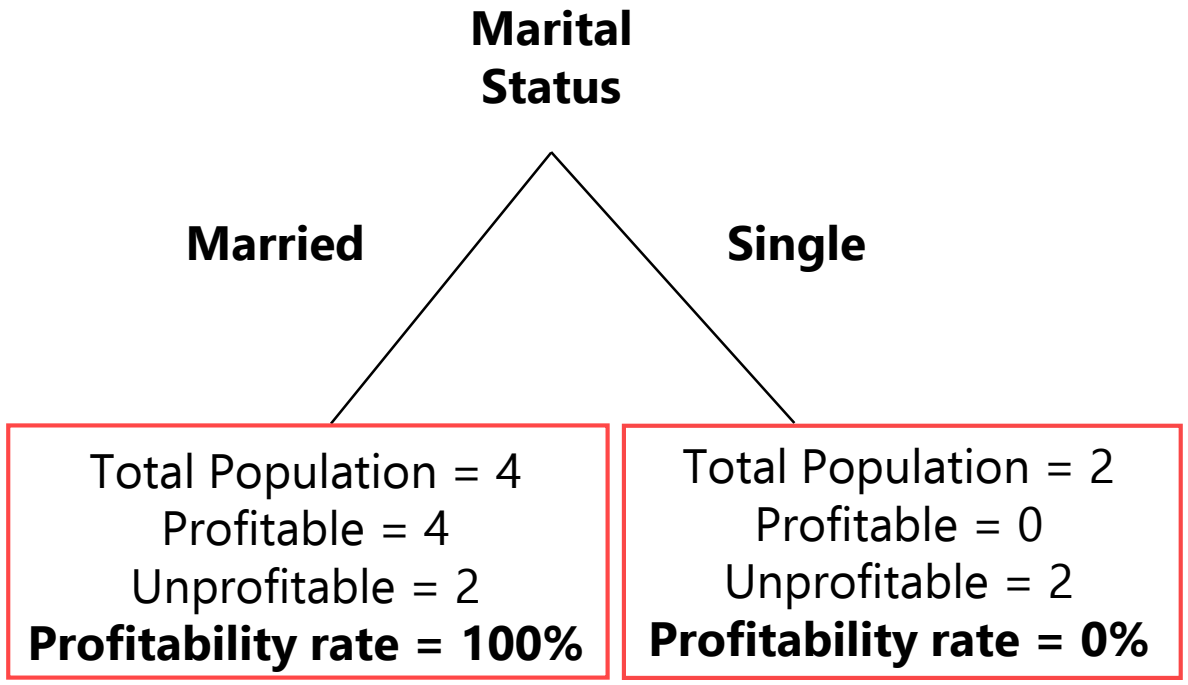
Total Population = 2
Profitable = 0
Unprofitable = 2
Profitability rate = 0%

Decision Tree

Decision tree classifier - Recursively sub-setting data can reveal interesting patterns

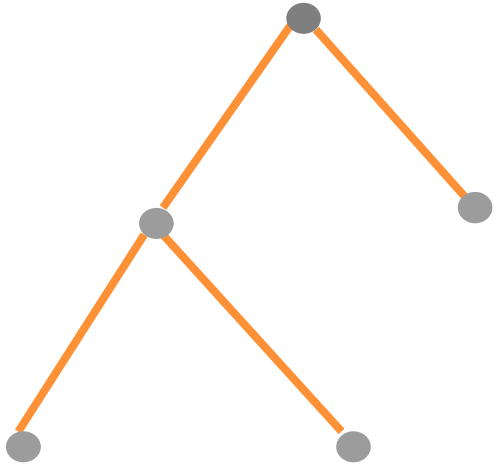


Data needs to be split in such a way so that the subsets of data end up being dominated by one class of the target variable



Decision Tree

Decision Tree splits into 2 parts at each node



Most implementations of a decision trees produce binary splits

Binary Tree

Decision Tree: Algorithm

How to decide which variable should be used to create splits?

Understand the intuition behind creating splits

The intuition will be formalized by introducing purity metrics

Decision Tree: Algorithm

Previous Example

Total Population = 10
Profitable = 5
Unprofitable = 5
Profitability rate = 50%

> 35 **Age** **<= 35**

Total Population = 6
Profitable = 4
Unprofitable = 2
Profitability rate = 66%

Total Population = 4
Profitable = 1
Unprofitable = 3
Profitability rate = 25%

Total Population = 10
Profitable = 5
Unprofitable = 5
Profitability rate = 50%

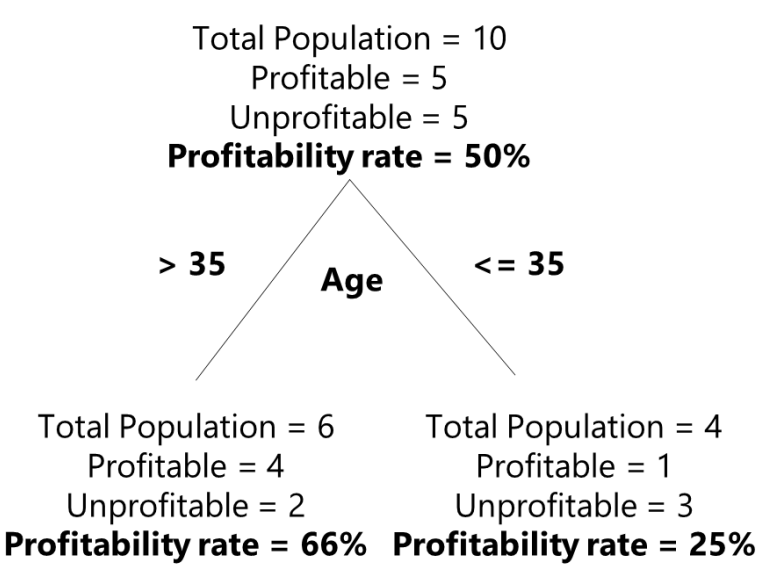
Male **Gender** **Female**

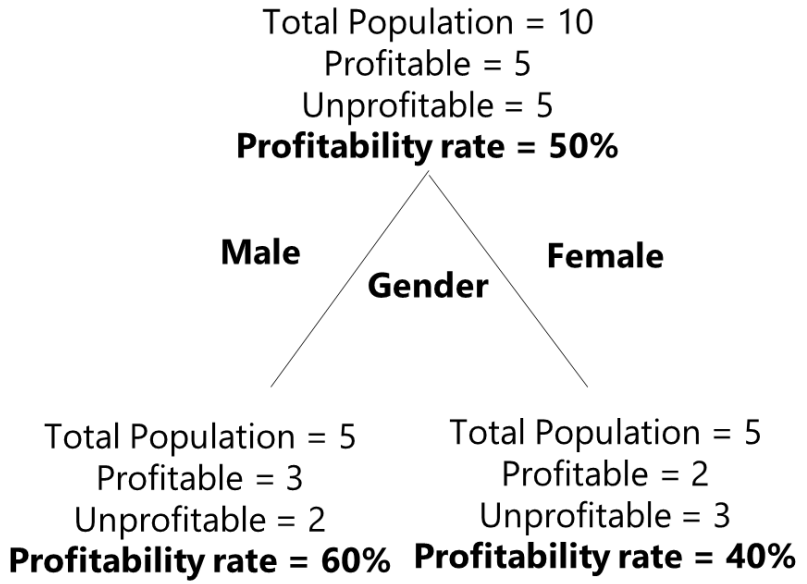
Total Population = 5
Profitable = 3
Unprofitable = 2
Profitability rate = 60%

Total Population = 5
Profitable = 2
Unprofitable = 3
Profitability rate = 40%

Both splits can be compared to understand which split is better

Decision Tree: Algorithm

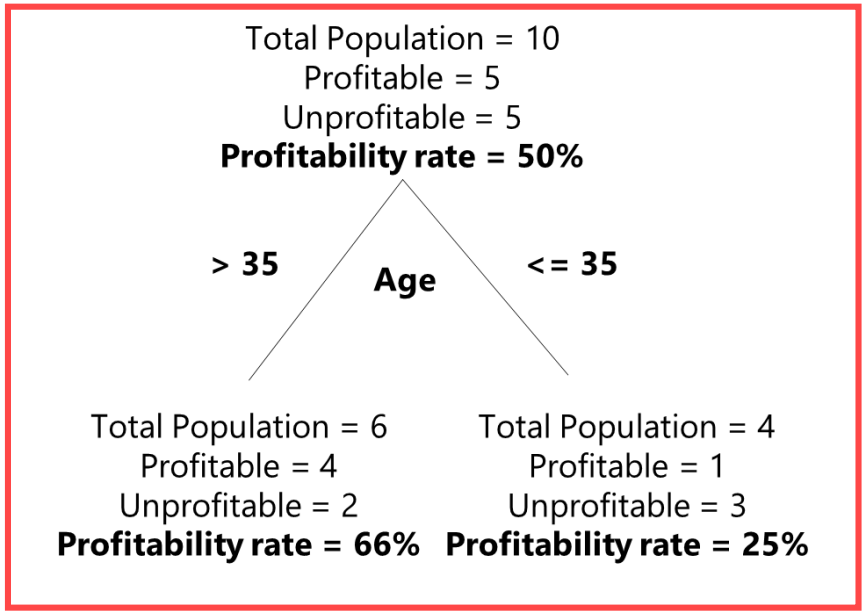


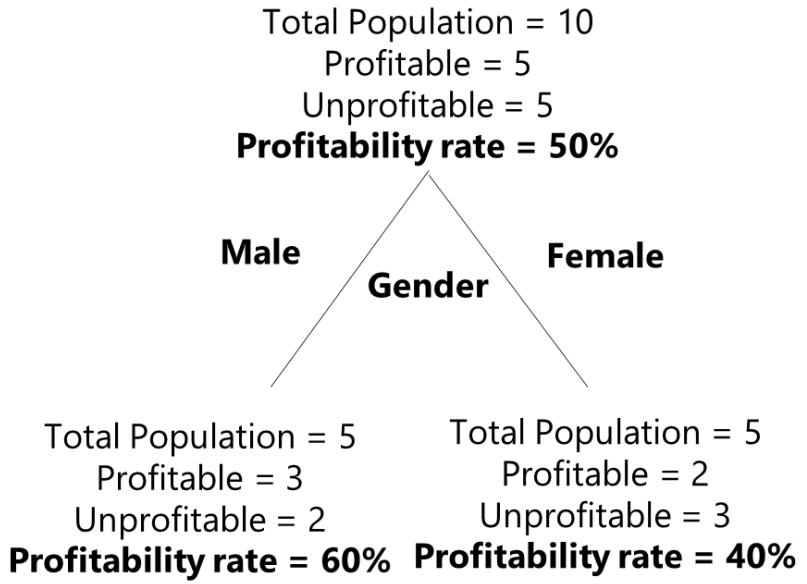


Which variable produces better splits?

Age or Gender?

Decision Tree: Algorithm





Good split in context of classification problem

Split produced by variable age are better than the splits produced by variable gender

Greater the **class imbalance**, better the split

Decision Tree: Algorithm

Class imbalance can be measured by computing Gini or Entropy

$$Gini = 1 - \sum p_i^2$$

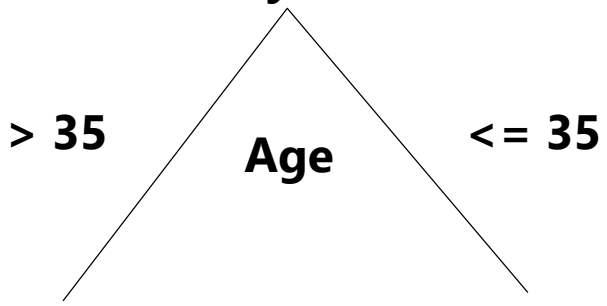
$$Entropy = -\sum p_i log_2 p_i$$

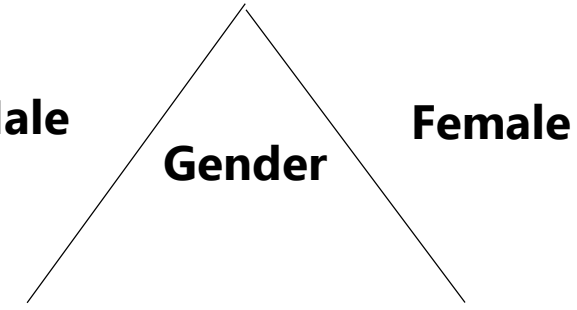
Decision Tree: Algorithm

Total Population = 10
Profitable = 5
Unprofitable = 5
Profitability rate = 50%

$$Gini = 1 - \sum p_i^2$$

Total Population = 10
Profitable = 5
Unprofitable = 5
Profitability rate = 50%





Total Population = 6
Profitable = 4
Unprofitable = 2
Profitability rate = 66%

$$1 - \left[\left(\frac{4}{6}\right)^2 + \left(\frac{2}{6}\right)^2\right]$$

0.44

Total Population = 4
Profitable = 1
Unprofitable = 3
Profitability rate = 25%

$$1 - \left[\left(\frac{1}{4}\right)^2 + \left(\frac{3}{4}\right)^2\right]$$

0.375

Total Population = 5
Profitable = 3
Unprofitable = 2
Profitability rate = 60%

$$1 - \left[\left(\frac{3}{5}\right)^2 + \left(\frac{2}{5}\right)^2\right]$$

0.48

Total Population = 5
Profitable = 2
Unprofitable = 3
Profitability rate = 40%

$$1 - \left[\left(\frac{2}{5}\right)^2 + \left(\frac{3}{5}\right)^2\right]$$

0.48

Decision Tree: Algorithm

Total Population = 10
Profitable = 5
Unprofitable = 5
Profitability rate = 50%

$$Gini = 1 - \sum p_i^2$$

> 35 — **Age** — **<= 35**

Total Population = 6
Profitable = 4
Unprofitable = 2
Profitability rate = 66%

Total Population = 4
Profitable = 1
Unprofitable = 3
Profitability rate = 25%

$(\frac{6}{10}) * 0.44 + (\frac{4}{10}) * 0.375$

0.41

Total Population = 10
Profitable = 5
Unprofitable = 5
Profitability rate = 50%

Male — **Gender** — **Female**

Total Population = 5
Profitable = 3
Unprofitable = 2
Profitability rate = 60%

Total Population = 5
Profitable = 2
Unprofitable = 3
Profitability rate = 40%

$(\frac{5}{10}) * 0.48 + (\frac{5}{10}) * 0.48$

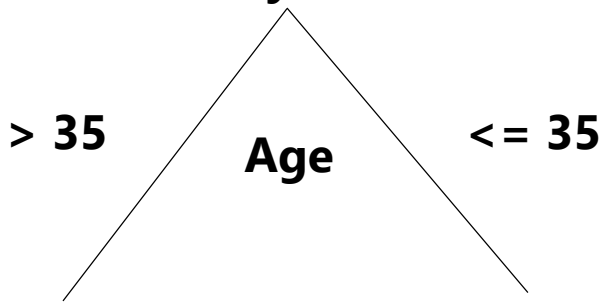
0.48

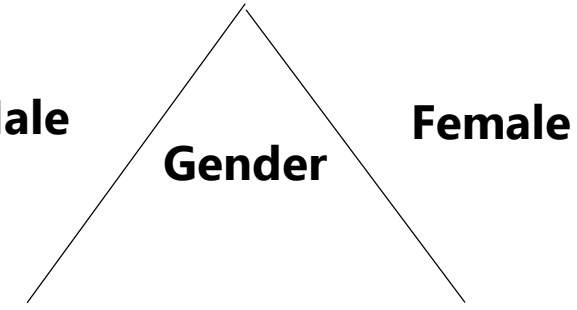
Decision Tree: Algorithm

Total Population = 10
Profitable = 5
Unprofitable = 5
Profitability rate = 50%

$$Entropy = -\sum p_i log_2 p_i$$

Total Population = 10
Profitable = 5
Unprofitable = 5
Profitability rate = 50%





Total Population = 6
Profitable = 4
Unprofitable = 2
Profitability rate = 66%

$$-[\left(\frac{4}{6}\right) * \log_2\left(\frac{4}{6}\right) + \left(\frac{2}{6}\right) * \log_2\left(\frac{2}{6}\right)]$$

0.91

Total Population = 4
Profitable = 1
Unprofitable = 3
Profitability rate = 25%

$$-[\left(\frac{1}{4}\right) * \log_2\left(\frac{1}{4}\right) + \left(\frac{3}{4}\right) * \log_2\left(\frac{3}{4}\right)]$$

0.81

Total Population = 5
Profitable = 3
Unprofitable = 2
Profitability rate = 60%

$$-[\left(\frac{3}{5}\right) * \log_2\left(\frac{3}{5}\right) + \left(\frac{2}{5}\right) * \log_2\left(\frac{2}{5}\right)]$$

0.97

Total Population = 5
Profitable = 2
Unprofitable = 3
Profitability rate = 40%

$$-[\left(\frac{2}{5}\right) * \log_2\left(\frac{2}{5}\right) + \left(\frac{3}{5}\right) * \log_2\left(\frac{3}{5}\right)]$$

0.97

Decision Tree: Algorithm

Total Population = 10
Profitable = 5
Unprofitable = 5
Profitability rate = 50%

$$Entropy = -\sum p_i log_2 p_i$$

Total Population = 10
Profitable = 5
Unprofitable = 5
Profitability rate = 50%

> 35 **Age** **<= 35**

Male **Gender** **Female**

Total Population = 6
Profitable = 4
Unprofitable = 2
Profitability rate = 66%

Total Population = 4
Profitable = 1
Unprofitable = 3
Profitability rate = 25%

Total Population = 5
Profitable = 3
Unprofitable = 2
Profitability rate = 60%

Total Population = 5
Profitable = 2
Unprofitable = 3
Profitability rate = 40%

$\left(\frac{6}{10}\right) * 0.91 + \left(\frac{4}{10}\right) * 0.81$

0.87

$\left(\frac{5}{10}\right) * 0.97 + \left(\frac{5}{10}\right) * 0.97$

0.97

Decision Tree: Algorithm Overview

For each split the purity metric is computed

Choose the lowest variable which results in lowest value of purity metric

Continue doing these till some **stopping criteria** is met

Decision Tree: Algorithm Overview

Stopping Criteria

Depth of tree

Specifying the levels of the tree

Improvement in purity metric

Specifying the minimum change in purity metric from one split to another

Value in terminal node

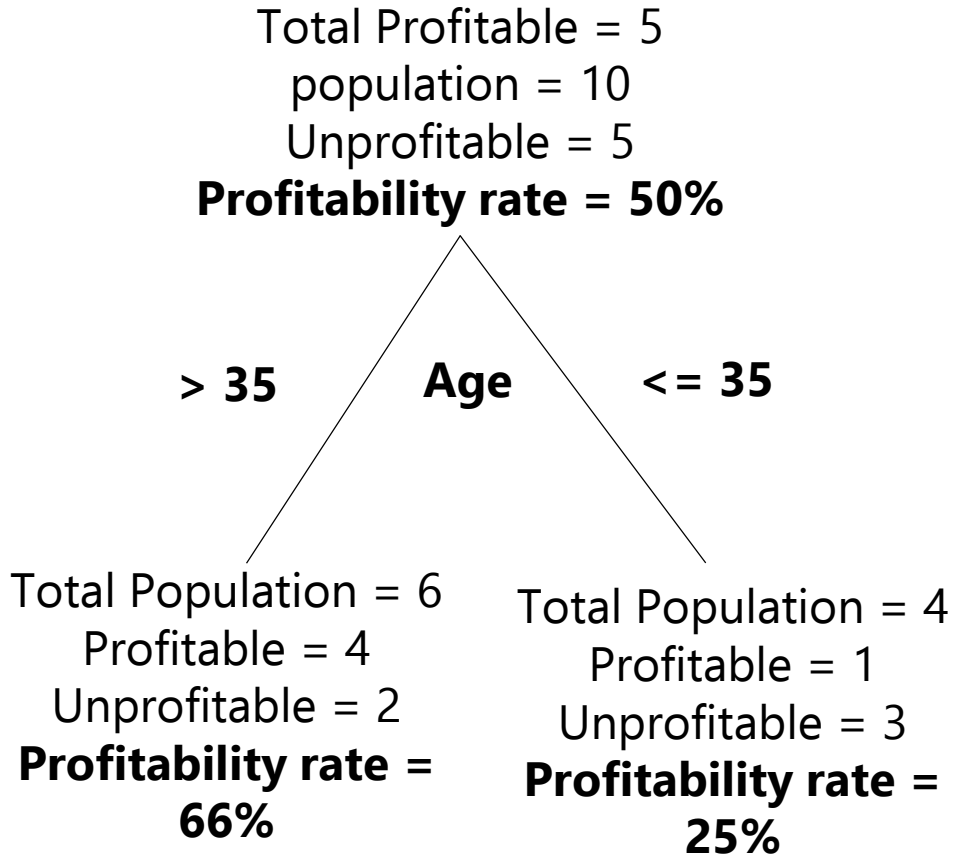
Specifying the number of value in the terminal node

Decision Tree: Prediction

Use decision tree classifier as prediction

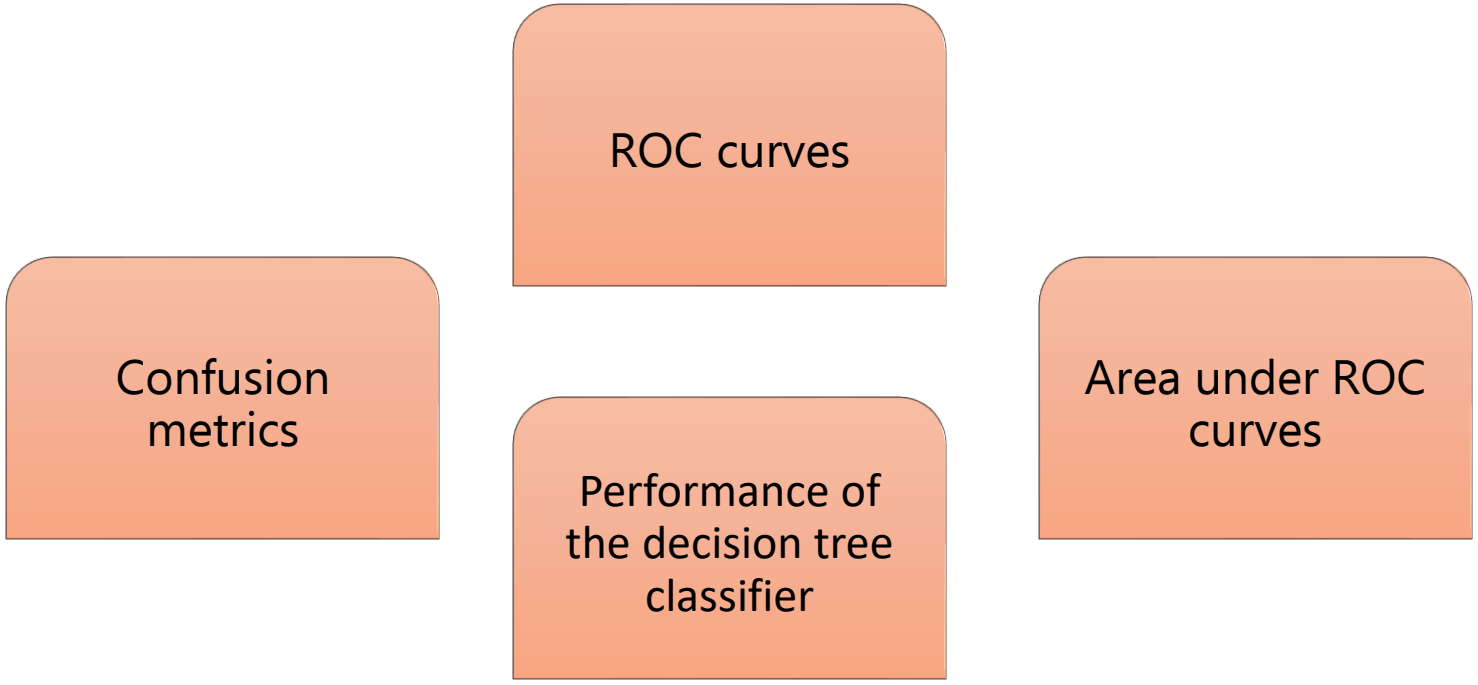
Available data – 20 year old person

Prediction – 25% Chance of him being profitable



Decision Tree: Performance Metrics

Decision tree classifier output probabilities



For multiclass problems, accuracy is used as a performance measure

Decision Tree: Parameters and Hyperparameters

Parameters of a decision tree

Data

Purity metric- Gini or Entropy?

Depth of the tree

These parameters are estimated using cross validation

At the model level of decision tree rules are decided for predicting probabilities or classes

Recap

- Decision Tree Overview
- Decision Tree Algorithms – Gini and Entropy
- Decision Tree Performance Metrics
- Decision Tree Parameter and Hyperparameter

Introduction to Machine Learning

Class

Tree Based Models


Topic

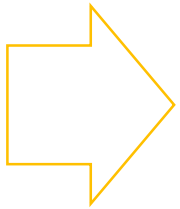
Introduction to Regression Tree

Decision Tree: Regression

Decision Tree can be used to do regression tasks

When the target variable is continuous decision tree regressor can be used

Prediction



Mean value of the target variable

Decision Tree: Regression

Example

Country	Rim	Tires	Type	Price
Japan	R14	195/60	Small	11.95
Japan	R15	205/60	Medium	24.76
Germany	R15	205/60	Medium	26.9
Germany	R14	175/60	Compact	18.9
Germany	R14	195/60	Compact	24.65
Germany	R15	225/60	Medium	33.2
USA	R14	185/75	Medium	13.15
USA	R14	205/75	Large	20.225
USA	R14	205/75	Large	16.145
USA	R15	205/70	Medium	23.04

Build a decision tree model to predict price

Price is a continuous variable

Regression tree

Recursively subset the data

Decision Tree: Regression

Example

Country	Rim	Tires	Type	Price
Japan	R14	195/60	Small	11.95
Japan	R15	205/60	Medium	24.76
Germany	R15	205/60	Medium	26.9
Germany	R14	175/60	Compact	18.9
Germany	R14	195/60	Compact	24.65
Germany	R15	225/60	Medium	33.2
USA	R14	185/75	Medium	13.15
USA	R14	205/75	Large	20.225
USA	R14	205/75	Large	16.145
USA	R15	205/70	Medium	23.04

Total Population =10
Average price = 21.9

R14

Yes

Total Population = 6
Average Price= 17.50

Price
11.95
18.9
24.65
13.15
20.22
16.14

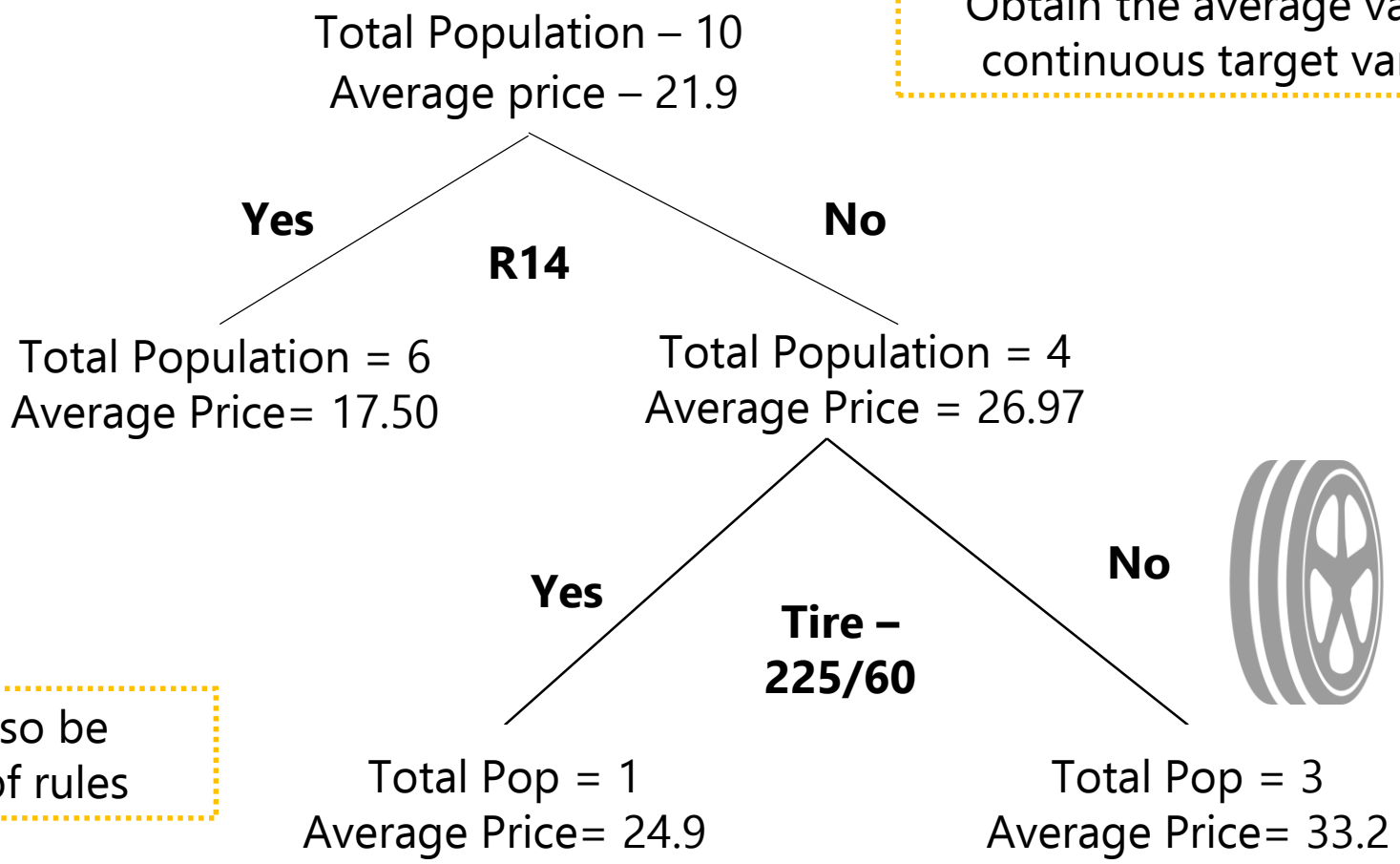
No

Total Population = 4
Average Price = 26.97

Price
24.76
26.90
33.20
23.04

Decision Tree: Regression

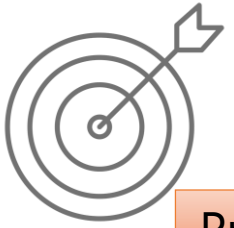
Obtain the average value of continuous target variable



A regression tree can also be summarised as a series of rules

Purity Metrics

How does a regression tree algorithm pick up which variable to split on?



Predictions need to be accurate

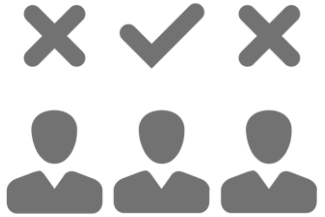
The prediction is the average value of target variable in decision node

Higher the accuracy of prediction, the better the split is

Mean Squared Error (MSE) or Residual Sum of Square (RSS) as a proxy of accuracy in each node

Purity Metrics

Country	Rim	Tires	Type	Price
Japan	R14	195/60	Small	11.95
Japan	R15	205/60	Medium	24.76
Germany	R15	205/60	Medium	26.9
Germany	R14	175/60	Compact	18.9
Germany	R14	195/60	Compact	24.65
Germany	R15	225/60	Medium	33.2
USA	R14	185/75	Medium	13.15
USA	R14	205/75	Large	20.225
USA	R14	205/75	Large	16.145
USA	R15	205/70	Medium	23.04



MSE or RSS helps in deciding which variable to choose for a split

Purity Metrics

Country	Rim	Tires	Type	Price
Japan	R14	195/60	Small	11.95
Japan	R15	205/60	Medium	24.76
Germany	R15	205/60	Medium	26.9
Germany	R14	175/60	Compact	18.9
Germany	R14	195/60	Compact	24.65
Germany	R15	225/60	Medium	33.2
USA	R14	185/75	Medium	13.15
USA	R14	205/75	Large	20.225
USA	R14	205/75	Large	16.145
USA	R15	205/70	Medium	23.04

Total Population =10

Average price = 21.9

Yes | **R14** | **No**

Total Population = 6
Average Price= 17.50

Price
11.95
18.9
24.65
13.15
20.22
16.14

Total Population = 4
Average Price = 26.97

Price
24.76
26.90
33.20
23.04

Purity Metrics

Country	Rim	Tires	Type	Price
Japan	R14	195/60	Small	11.95
Japan	R15	205/60	Medium	24.76
Germany	R15	205/60	Medium	26.9
Germany	R14	175/60	Compact	18.9
Germany	R14	195/60	Compact	24.65
Germany	R15	225/60	Medium	33.2
USA	R14	185/75	Medium	13.15
USA	R14	205/75	Large	20.225
USA	R14	205/75	Large	16.145
USA	R15	205/70	Medium	23.04

Total Population =10

Average price = 21.9

Yes — **Country - Germany** — **No**

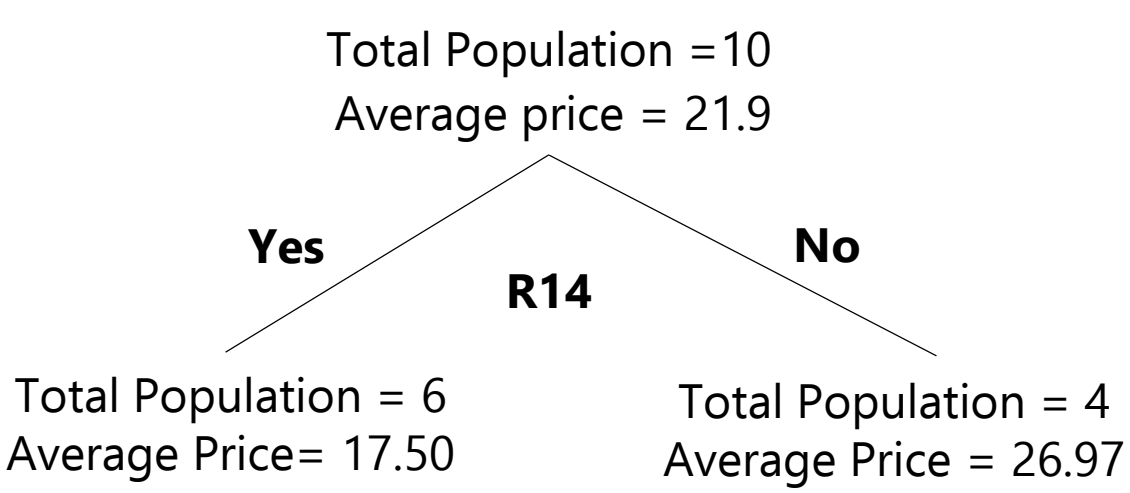
Total Population = 4
Average Price= 25.91

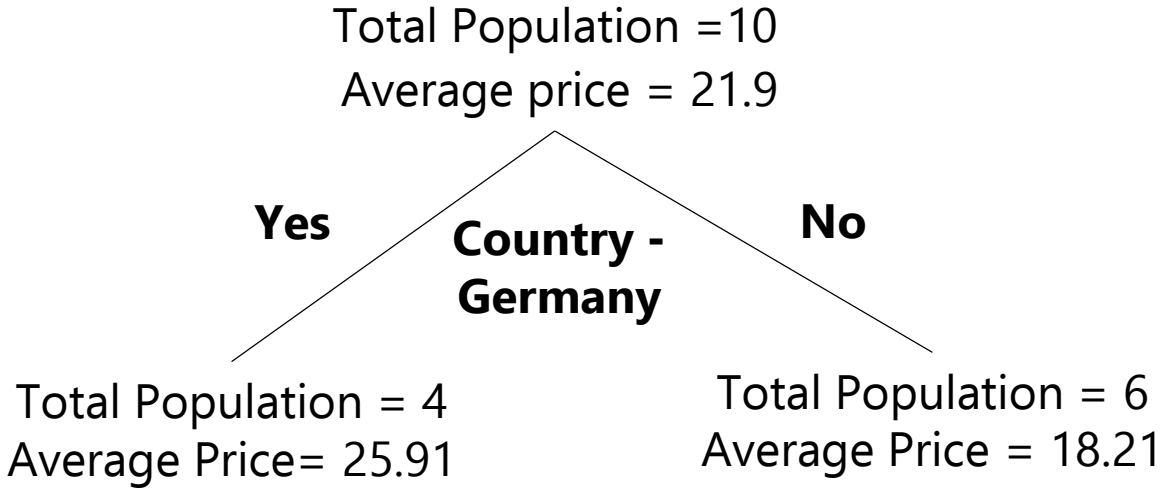
Price
26.90
18.90
26.45
33.20

Total Population = 6
Average Price = 18.21

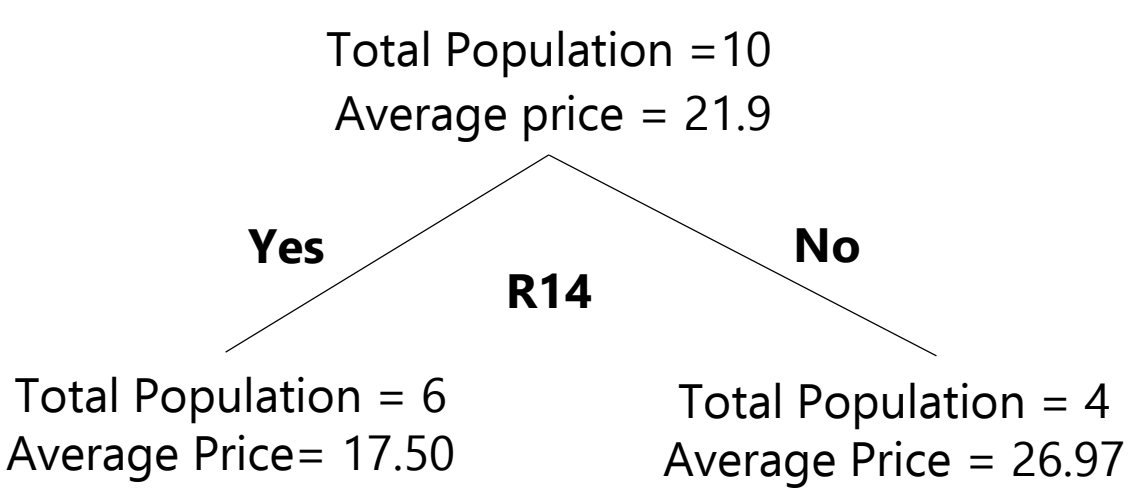
Price
11.95
24.76
13.15
20.22
16.14
23.04

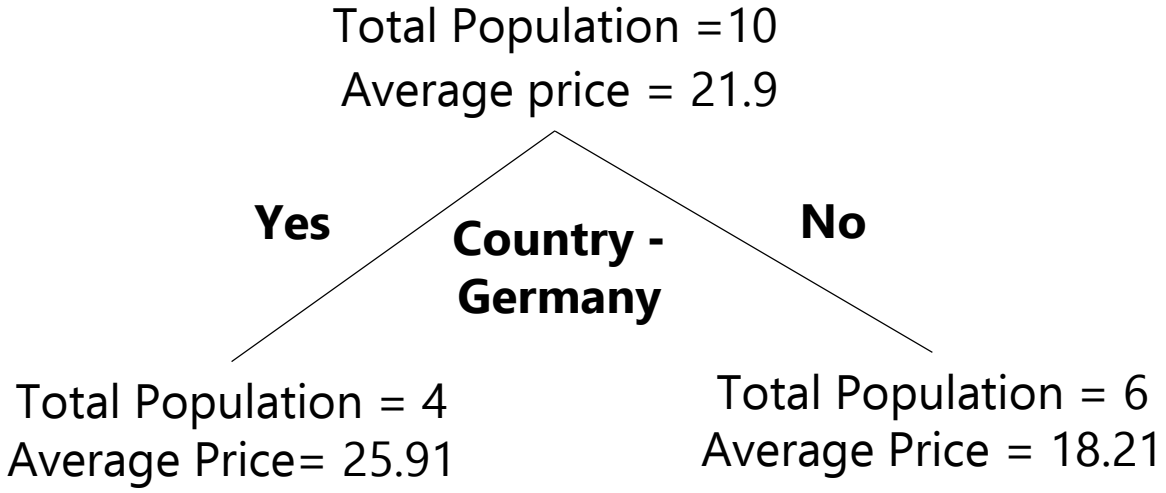
Purity Metric





Purity Metric





Use Mean Squared Error (MSE) or Residual Sum of Square (RSS)

$$MSE = \frac{1}{n}\sum(y_i - \mu)^2$$

MSE is just the average of RSS

Nothing but variance in the values of target in variable in a node

Purity Metric

Total Population =10
Average price = 21.9

Yes — **R14** — **No**

Total Population = 6
Average Price= 17.50

Price	Pred.
11.95	17.50
18.9	17.50
24.65	17.50
13.15	17.50
20.22	17.50
16.14	17.50

Total Population = 4
Average Price = 26.97

Price	Pred.
24.76	26.97
26.90	26..97
33.20	26.97
23.04	26.97

Total Population =10
Average price = 21.9

Yes — **Country - Germany** — **No**

Total Population = 4
Average Price= 25.91

Price	Pred.
26.90	25.91
18.90	25.91
26.45	25.91
33.20	25.91

Total Population = 6
Average Price = 18.21

Price	Pred.
11.95	18.21
24.76	18.21
13.15	18.21
20.22	18.21
16.14	18.21
23.04	18.21

MSE tries to find out how accurate a prediction is in each node

Purity Metric

Total Population =10
Average price = 21.9

$$MSE = \frac{1}{n}\sum(y_i - \mu)^2$$

R14

Yes

Total Population = 6
Average Price= 17.50

Price	Pred.
11.95	17.50
18.9	17.50
24.65	17.50
13.15	17.50
20.22	17.50
16.14	17.50

$$\frac{1}{6}(11.95 - 17.50)^2 + (18.90 - 17.50)^2 + .. + (16.14 - 17.50)^2$$

No

Total Population = 4
Average Price = 26.97

Price	Pred.
24.76	26.97
26.90	26..97
33.20	26.97
23.04	26.97

$$\frac{1}{4}(24.76 - 26.97)^2 + (26.90 - 26.97)^2 + .. + (23.04 - 26.97)^2$$

Total Population =10
Average price = 21.9

Country - Germany

Yes

Total Population = 4
Average Price= 25.91

Price	Pred.
26.90	25.91
18.90	25.91
26.45	25.91
33.20	25.91

No

Total Population = 6
Average Price = 18.21

Price	Pred.
11.95	18.21
24.76	18.21
13.15	18.21
20.22	18.21
16.14	18.21
23.04	18.21

Purity Metric

Total Population =10
Average price = 21.9

Yes | **R14** | **No**

Total Population = 6
Average Price= 17.50

MSE – 18.67

Total Population = 4
Average Price = 26.97

MSE – 14.78

$$MSE = \frac{1}{n}\sum(y_i - \mu)^2$$

$$\frac{6}{10} * 18.67 + \frac{4}{10} * 14.78 \quad = 17.114$$

Total Population =10
Average price = 21.9

Yes | **Country - Germany** | **No**

Total Population = 4
Average Price= 25.91

MSE – 26.21

Total Population = 6
Average Price = 18.21

MSE – 23.22

$$\frac{4}{10} * 26.21 + \frac{6}{10} * 23.22 \quad = 24.416$$

Rim is better than country at producing more accurate predictions

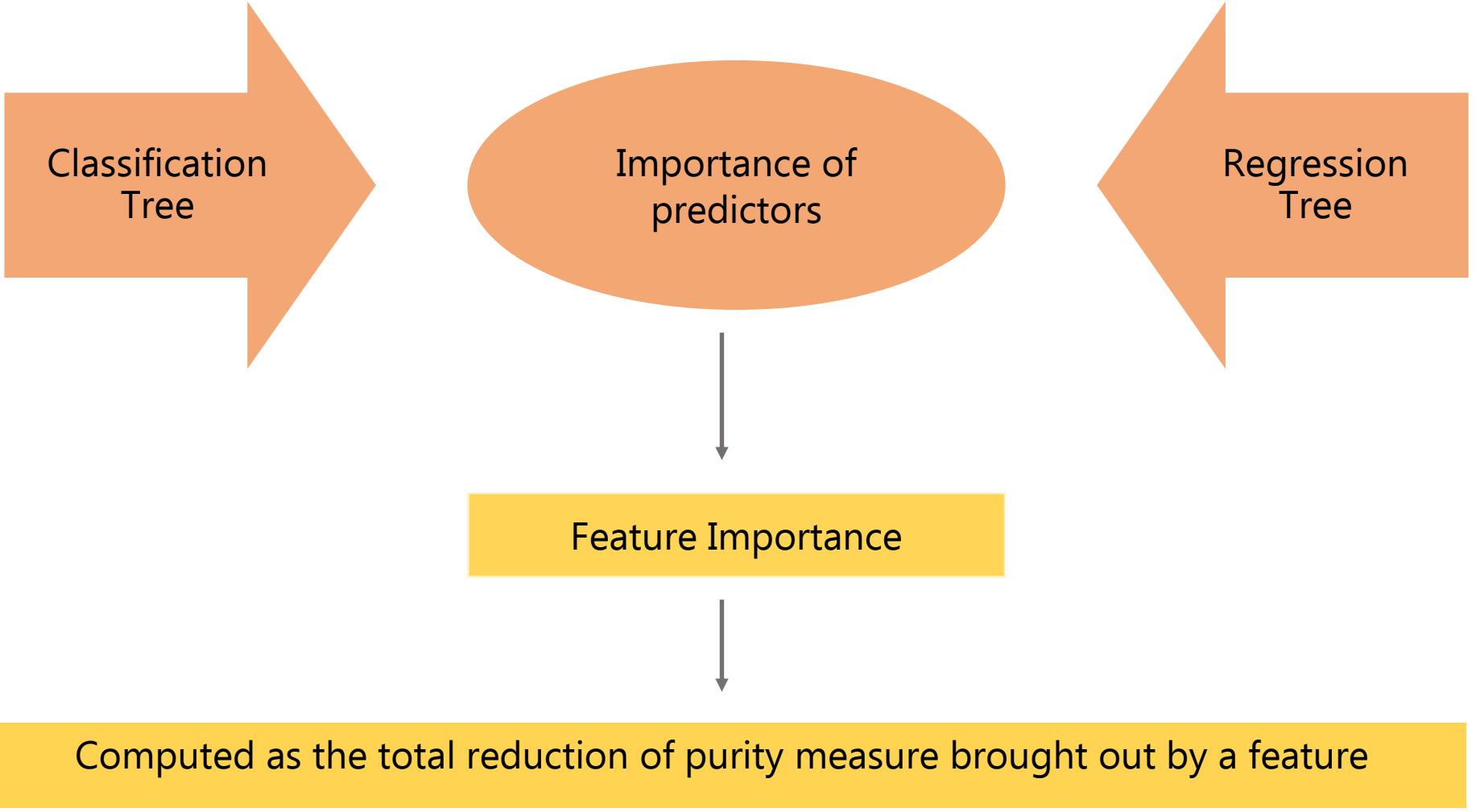
Hyperparameters

Regression Tree

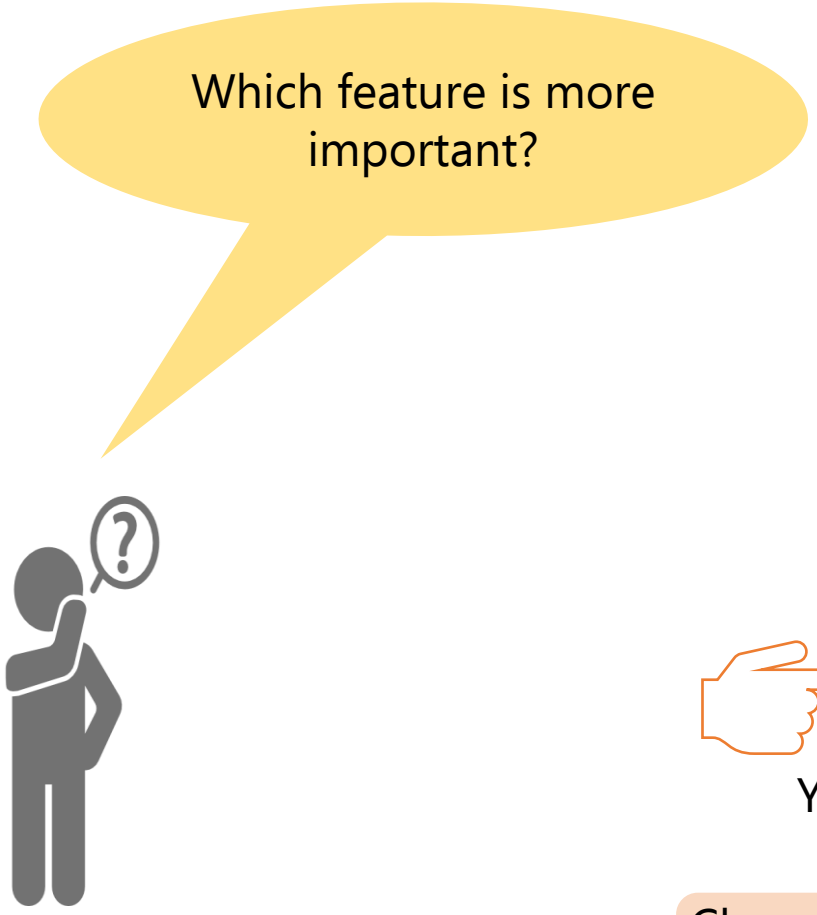
- Depth of tree
- Number of observations in terminal node

Grid search procedure to compute the appropriate values of these hyperparameters

Feature Importance



Feature Importance



Class A = 25%
Class B = 75%
N = 1000
Feature A

Yes

No

Feature A causes the 1st split

Class A = 20%
Class B = 80%
N = 400
Feature B

Class A = 45%
Class B = 55%

Yes

No

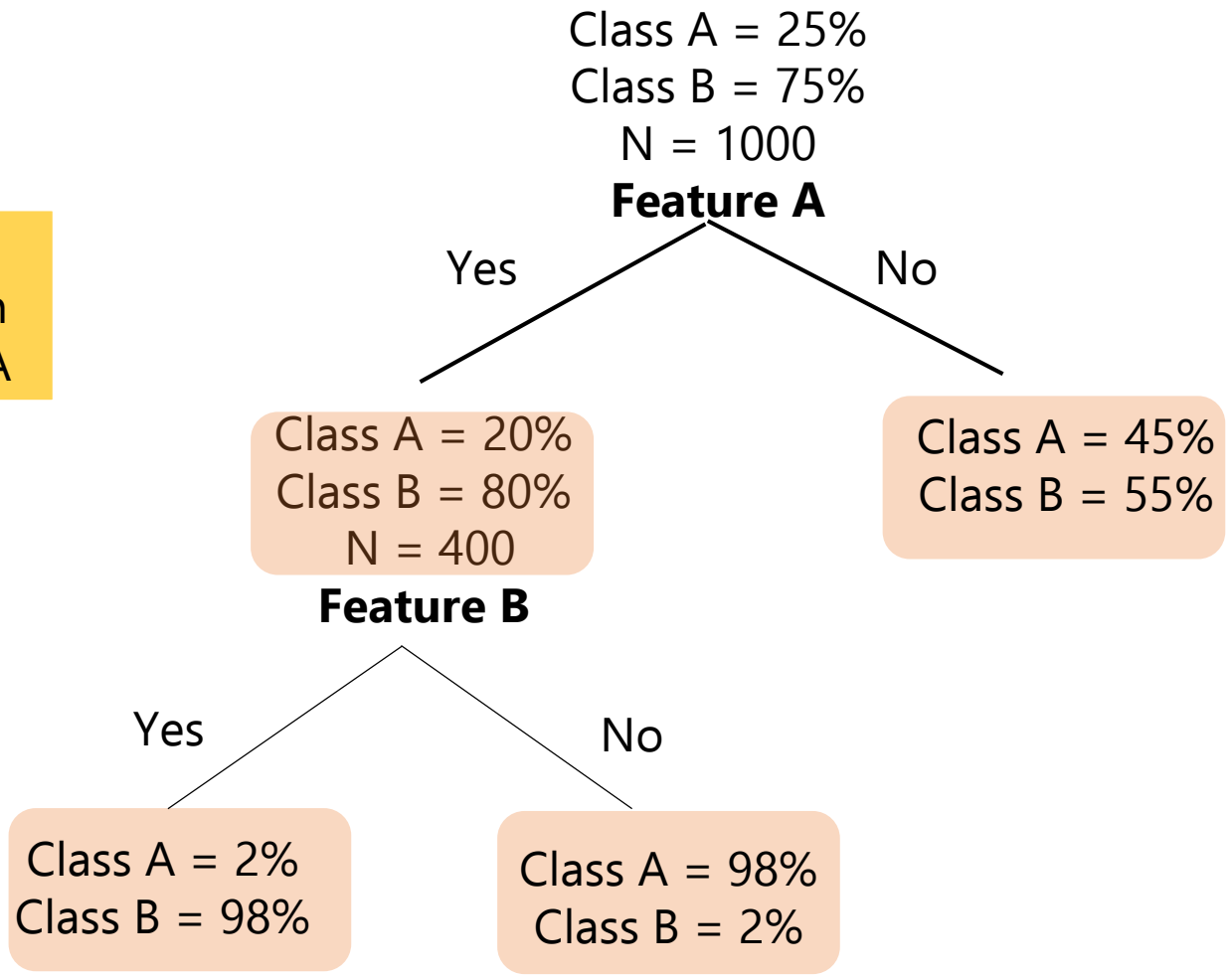
Feature B creates greater class imbalance

Class A = 2%
Class B = 98%

Class A = 98%
Class B = 2%

Feature Importance

Proportion of classes are **more disproportionate** in Feature B than in Feature A

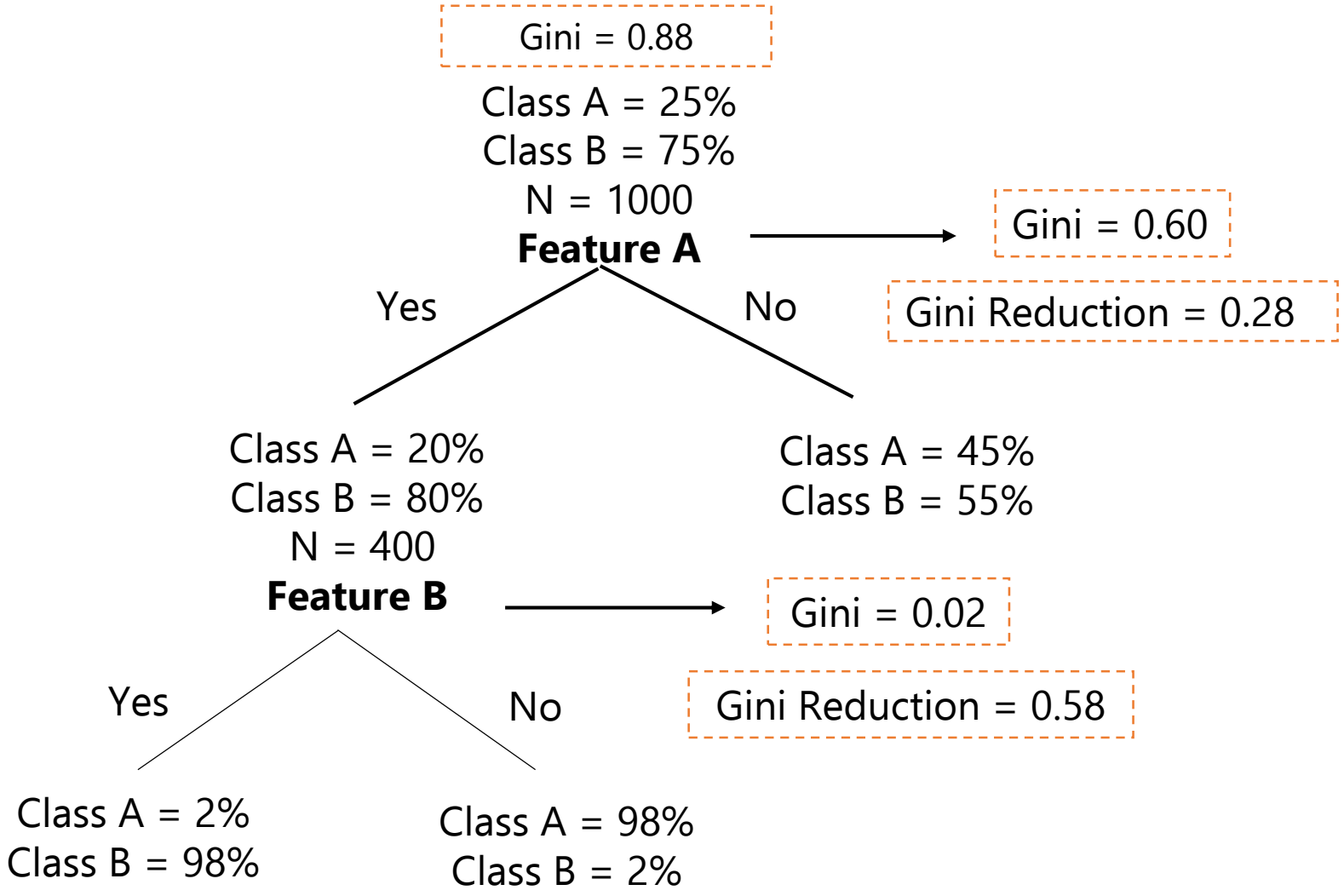


Feature Importance

In Variable Importance both the sequence of the split and the purity of a node should be considered

Feature A precedes Feature B

Feature B creates greater node purity



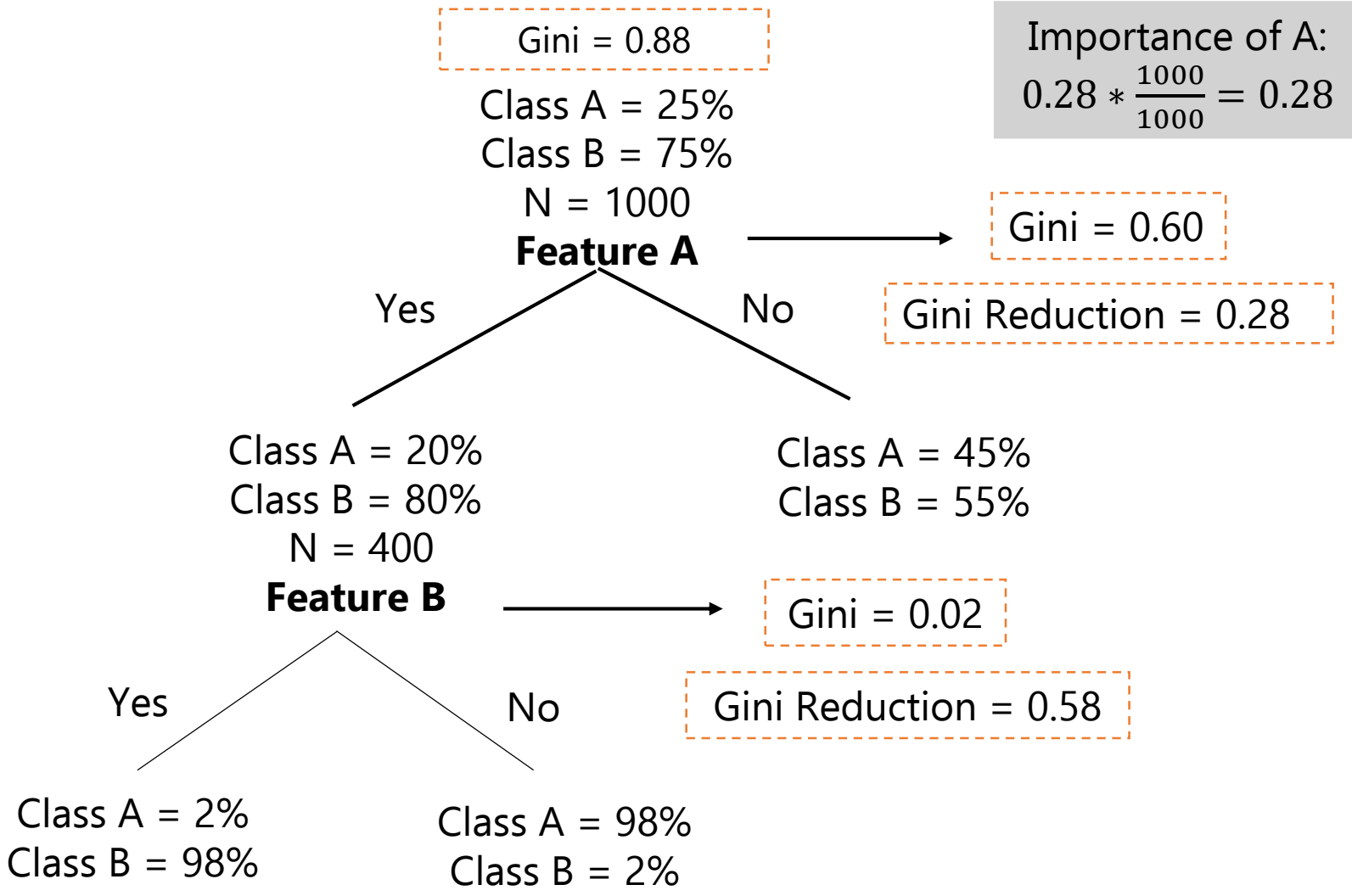
Feature Importance

Importance of A: Decrease in Gini * Proportion of data

Decrease in Gini
Ability of a variable to create class imbalance compared to preceding split

Proportion of data
Sequence in which variable causes the split

More observations will pass through the node caused by an early split

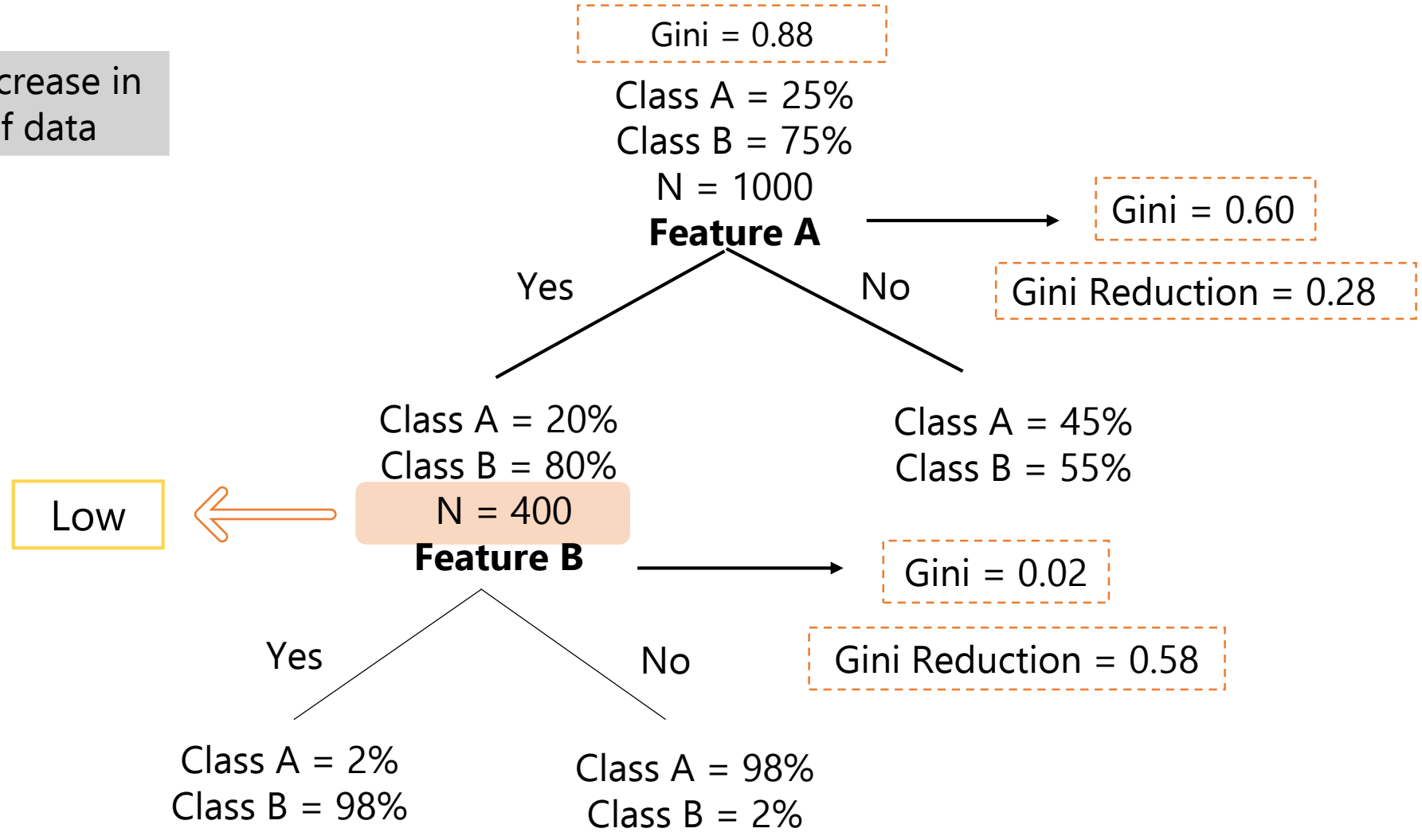


Importance of A:
$0.28 * \frac{1000}{1000} = 0.28$

Feature Importance

Importance of B: Decrease in Gini*Proportion of data

Importance of B:
$0.58 * \frac{400}{1000} = 0.23$



Feature Importance

Weigh the decrease in Mean Squared Error and Residual Sum Square appropriately

Feature Importance

Recap

1. Decision tree – Regression
2. Purity Metric
3. Hyperparameters
4. Feature Importance

Introduction to Machine Learning

Class

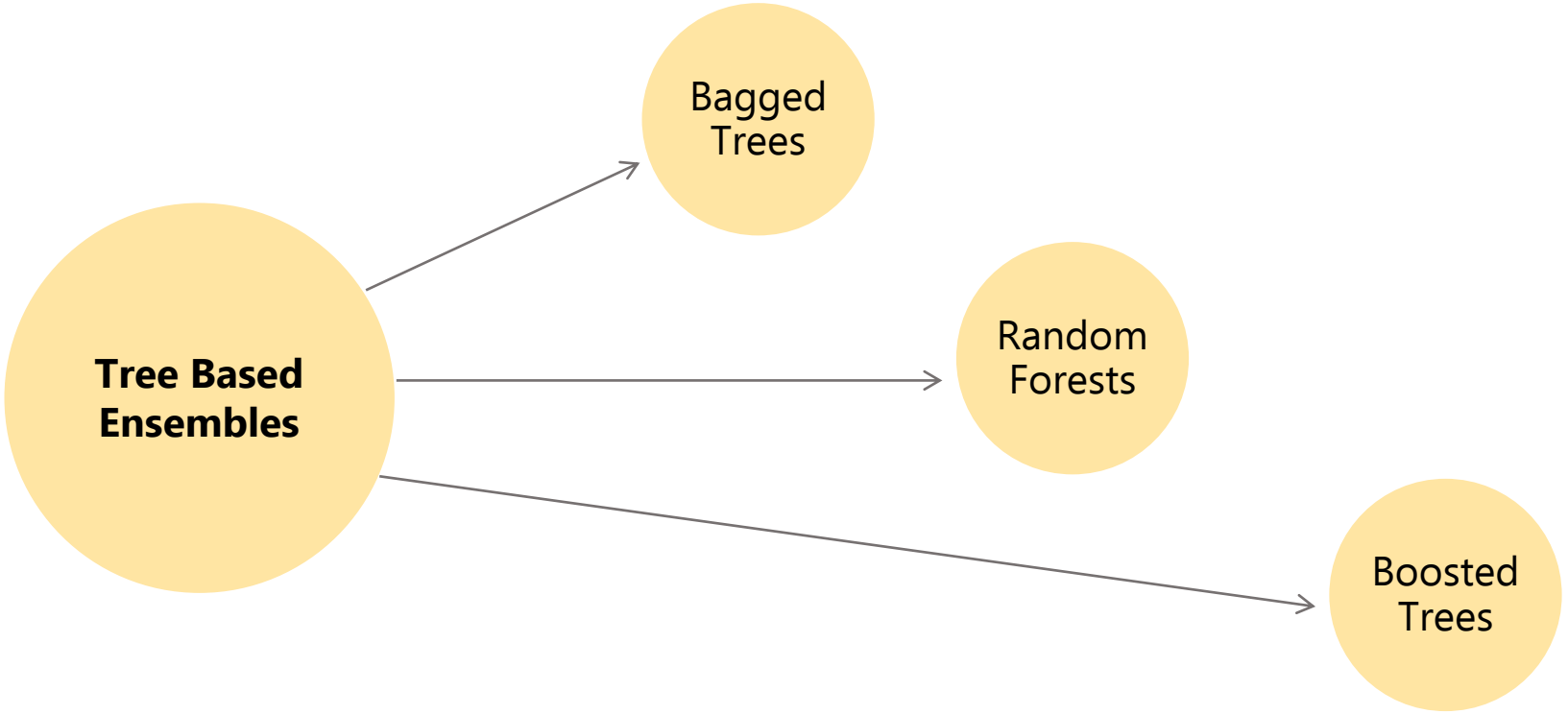
Tree Based Model

Topic

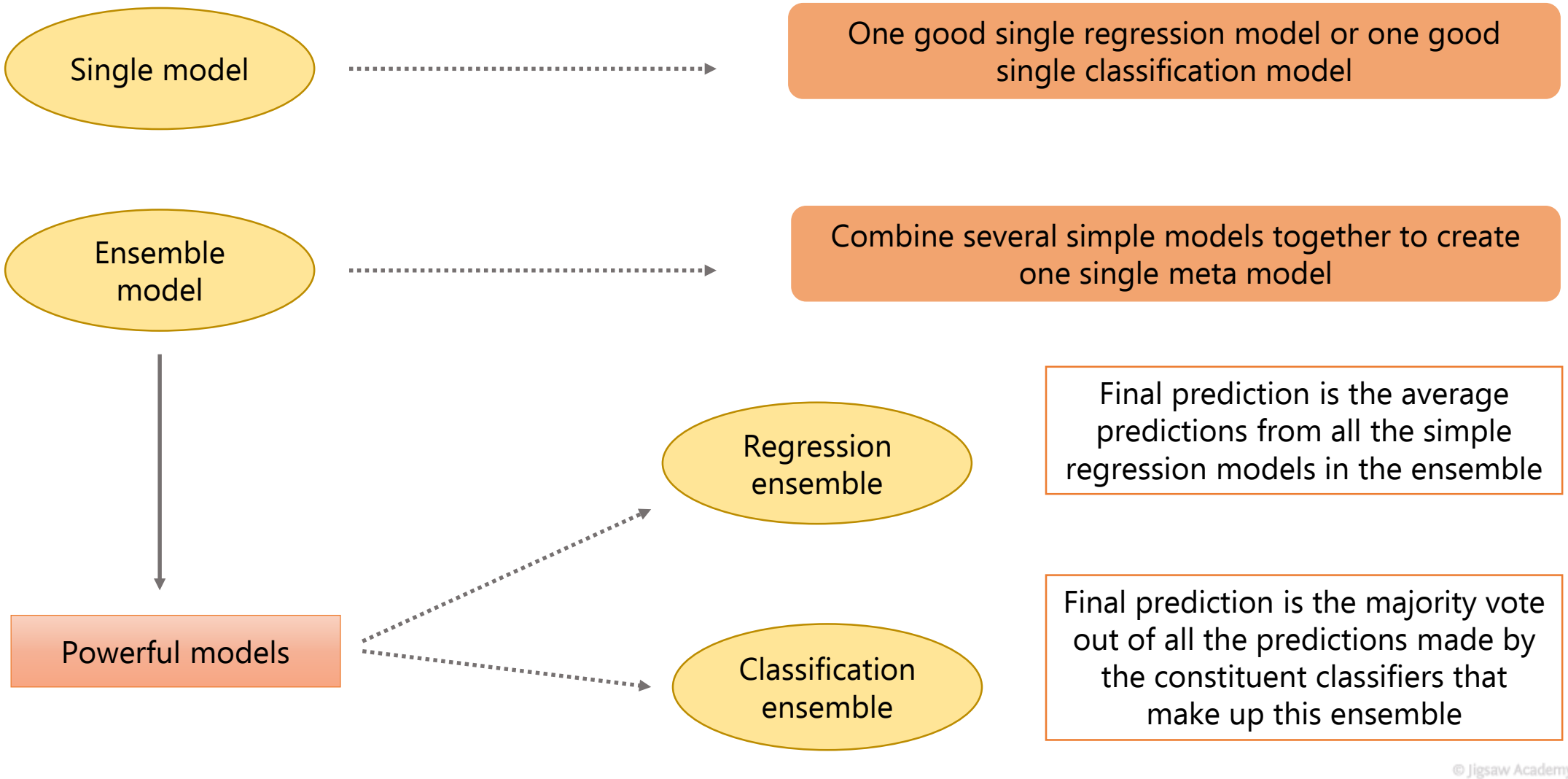
Tree Based Ensembles: Bagged Trees and Random Forests

Agenda

Another category of machine learning models → Ensemble Models

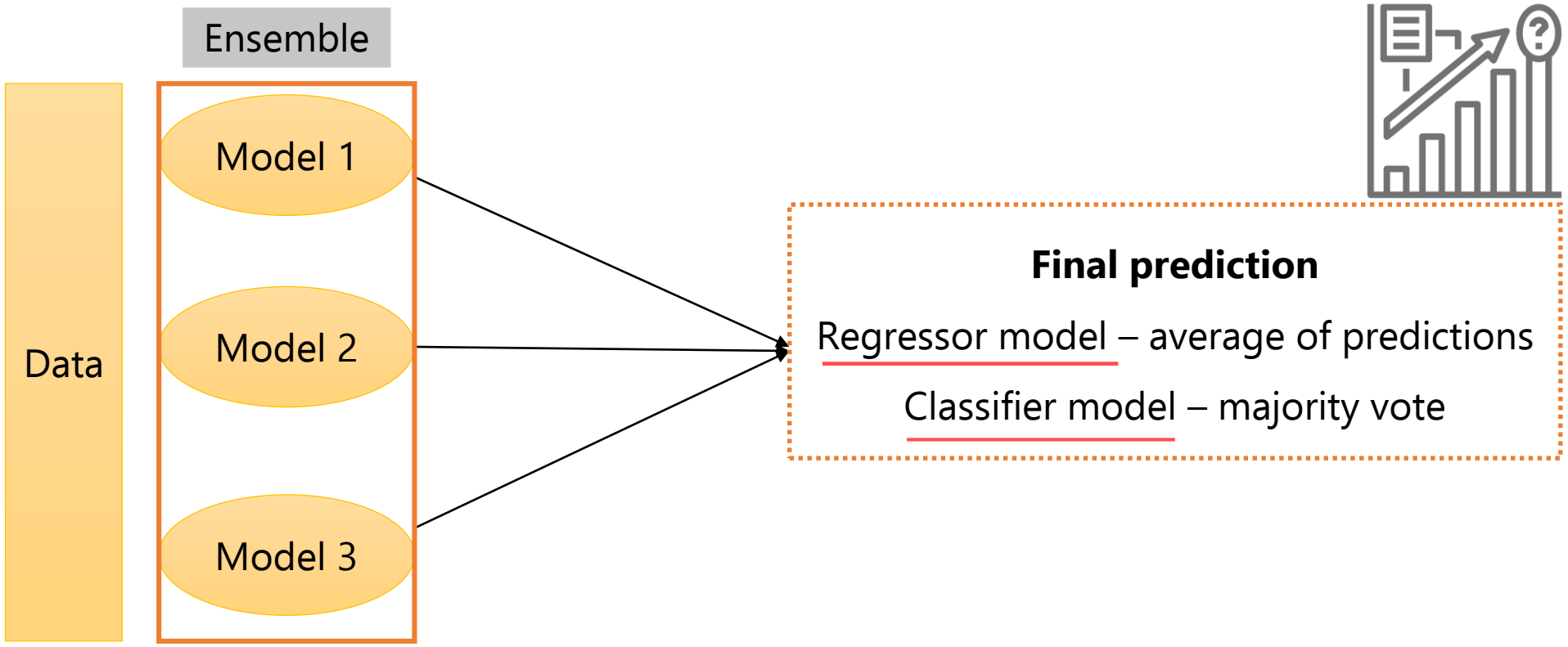


Tree Based Ensembles Overview



Tree Based Ensembles Overview

Schematic working of ensembles



Tree Based Ensembles Overview

Tree based ensemble model

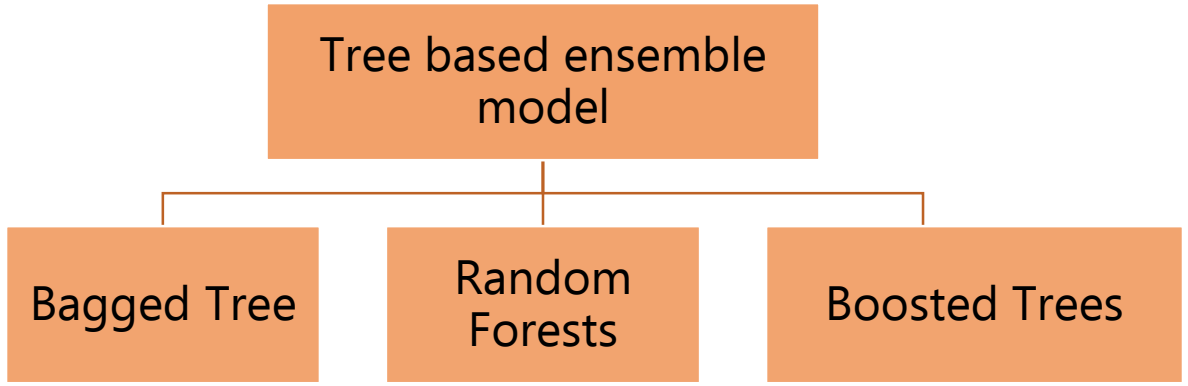
More popular choice

Only a subset of total data is used by each base learner

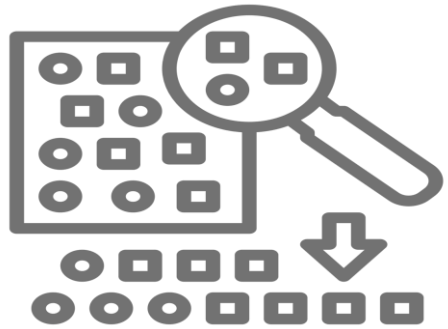
The way this data set is fed into each of the base learners is based on a **data sampling scheme**

Different sampling schemes give rise to different types of tree based ensembles

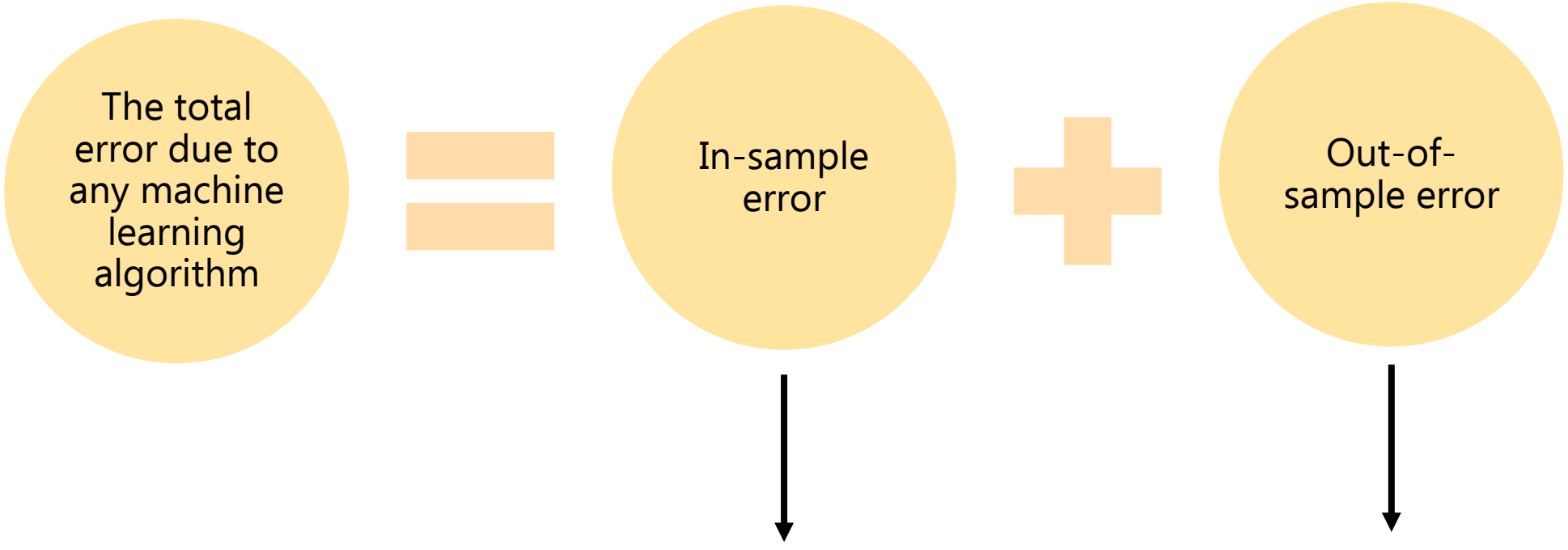
Tree Based Ensemble Models



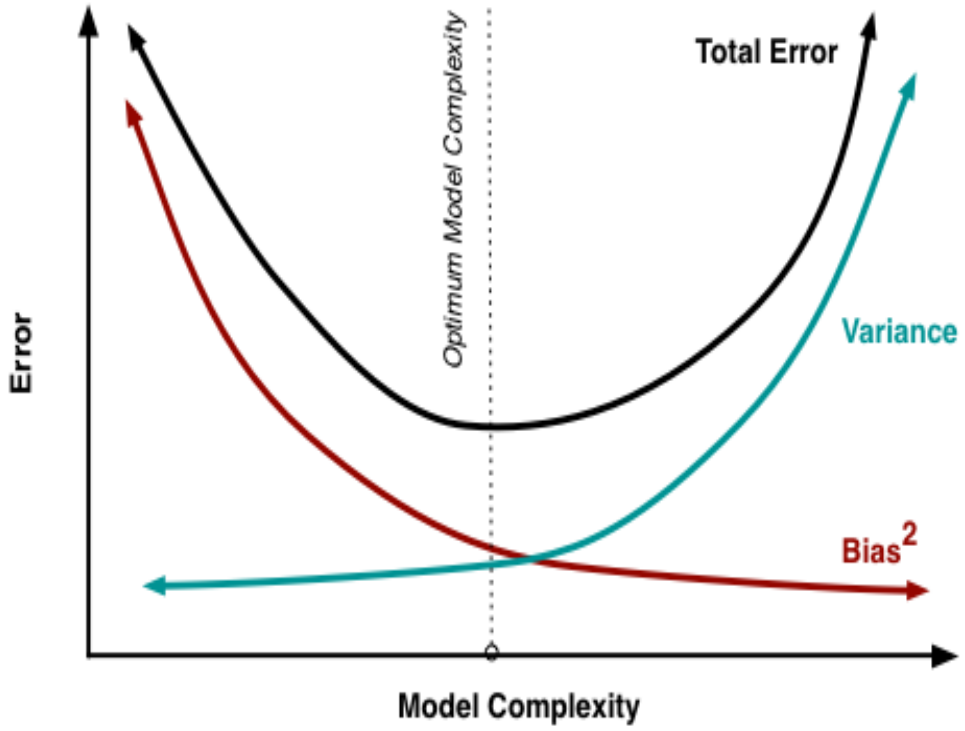
Sampling Scheme	Bootstrap Sampling	Bootstrap Sampling + Feature Sampling	Data Reweighing
Base Learner	Tree	Tree	Tree



Bagged Trees



Bagged Trees



Error = Bias + Variance = In-sample Error + Out-of-sample Error

Trade offs between the model complexity and the error in models

More complicated models have very **low in-sample error but have a high out-of-sample error**

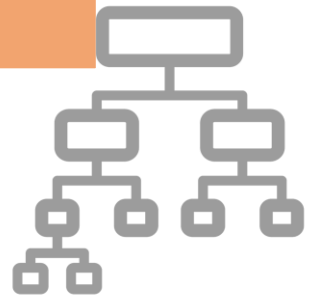
Simpler models have **low out-of-sample error but high in-sample error**

Theoretically, there is a limit to minimum error that can be achieved

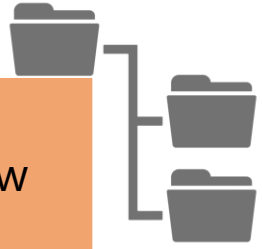
Reduce error further by decreasing in-sample error and out-of-sample error simultaneously

Bagged Trees

Use tree based models as base learners to reduce in-sample error



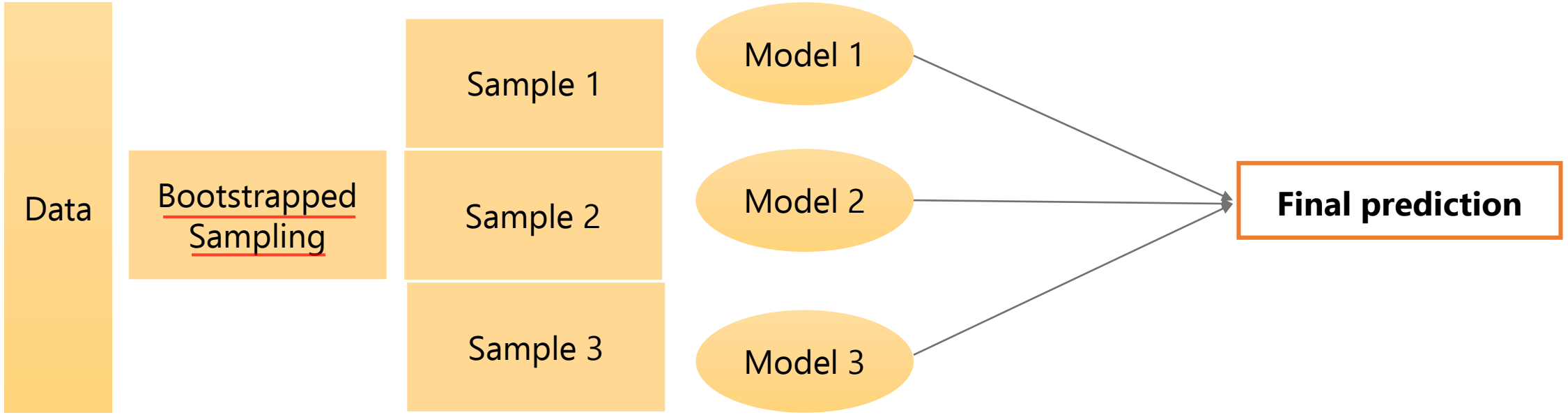
While training a tree based ensemble the constituent tree models are allowed to grow **many levels deep**



The intricacies of training data are captured intimately, thereby reducing the in-sample error

Bagged Trees

In the case of **Bagged trees** each of the unpruned trees are fed bootstrapped samples of original data set



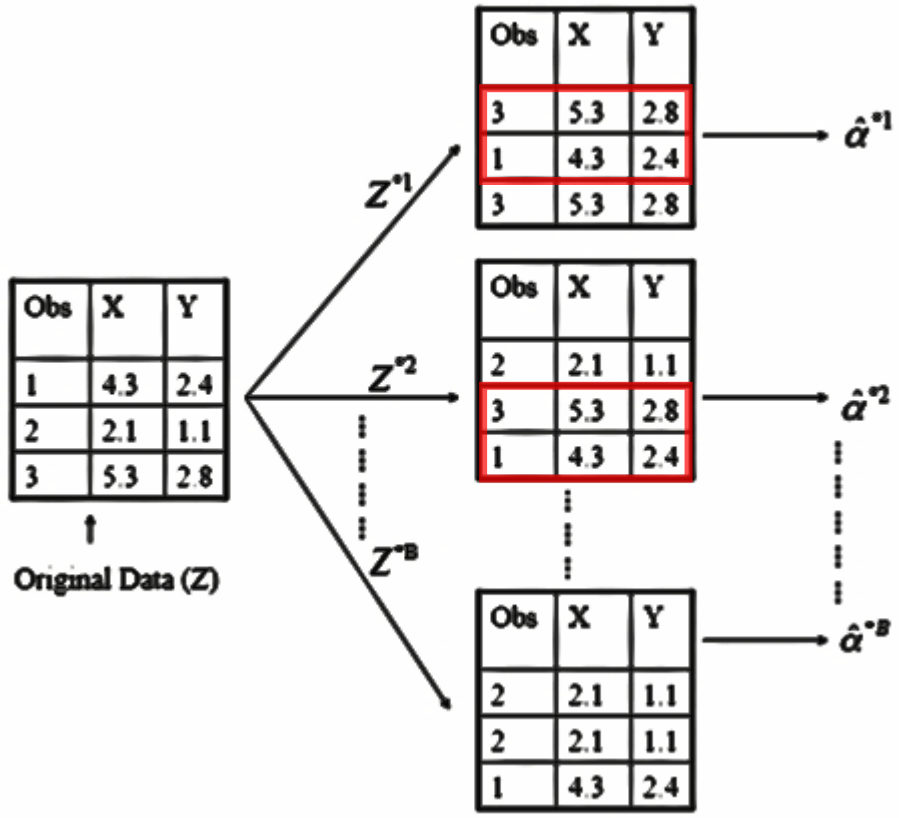
Bootstrapped Sampling

Bootstrapped sampling simply refers to **sampling by replacement**

Blue and red dots are repeated more often in the samples than they are present in the original data

Bootstrapped Sampling

Bootstrapped Sampling at the data level

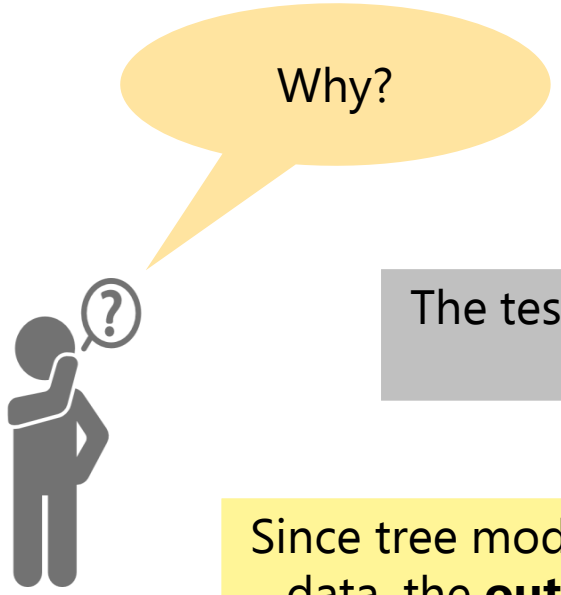


Bootstrapped Sampling

Bootstrapped Sampling helps in reducing out of sample error

If an unpruned decision tree is fit into any data set then the model will have **very high out-of-sample error**

Hypothetically, if unpruned tree model is being fitted on **total population data** the error will be very low

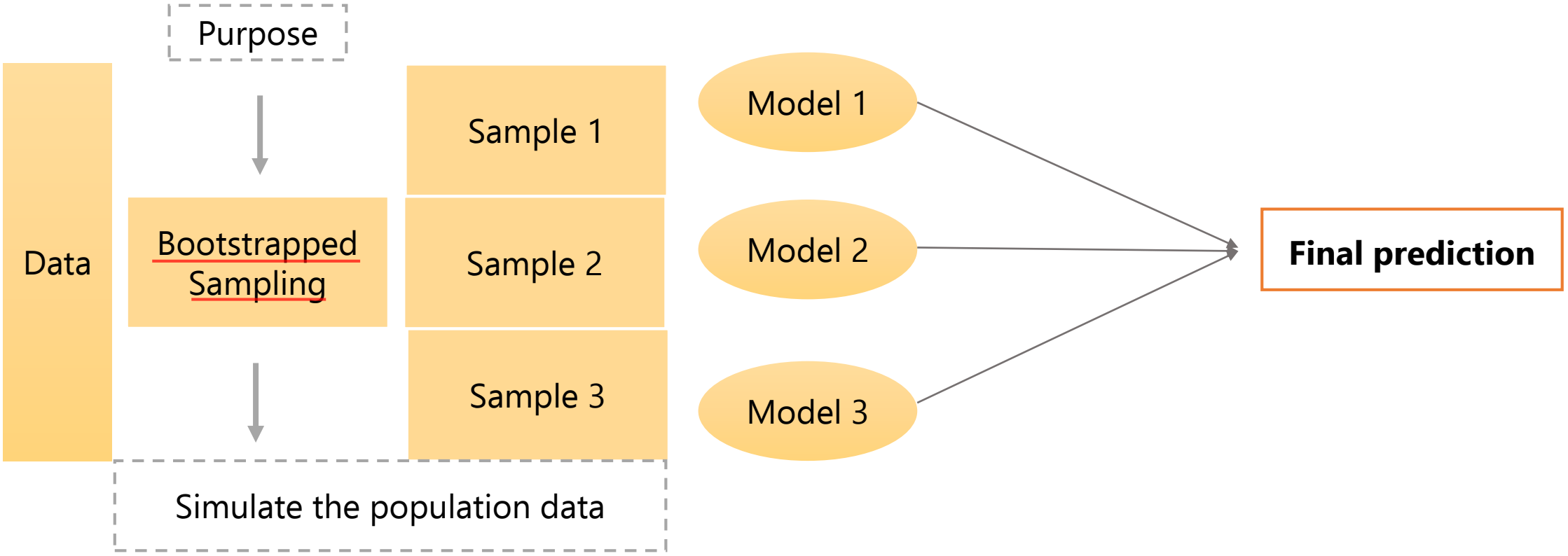


The test data can be very different from training data

Since tree model is being overfitted on training data, the **out-of-sample error will be high**

Low error due to all the variation and variety in data has been already seen by the model as the population data has been used to train the model

Bootstrapped Sampling



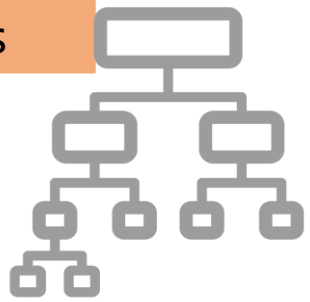
Realistic view of the data generation process

Synthetically generate variation and variety in the training data itself

Bagged Trees

Characteristics of Bagged Model

Using **unpruned decision trees** as base learners

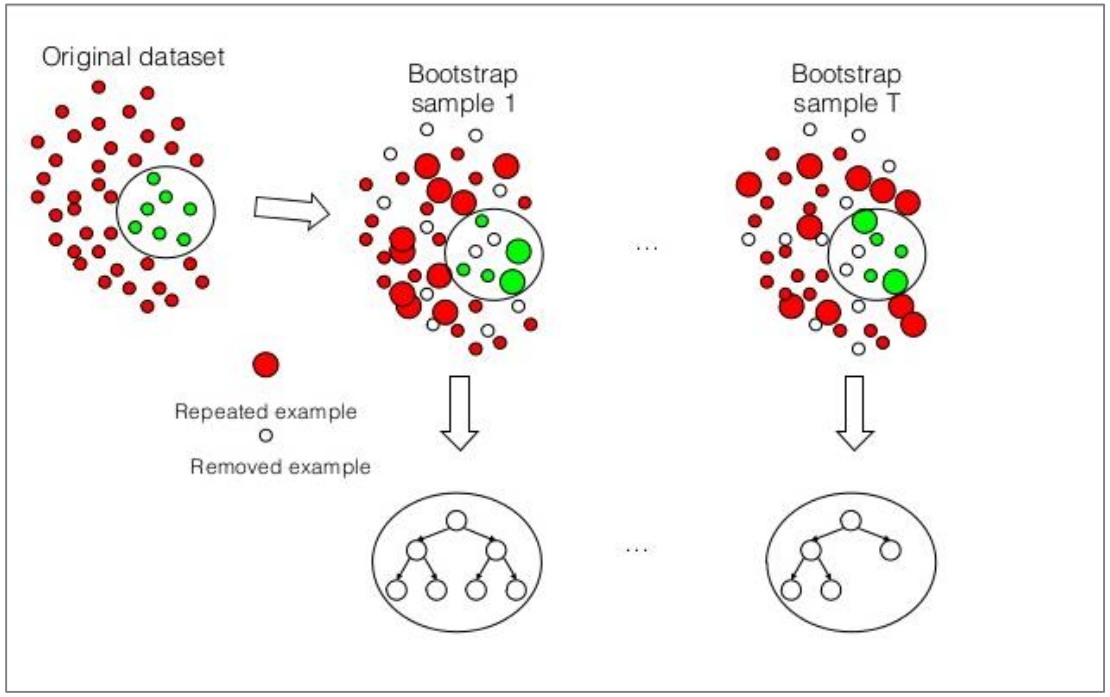


Using **Bootstrapped Sampling** to create samples that are fed to each of the base learners

Peculiarities of Bagged Trees

Bagged tree ensemble is comprised of multiple decision trees

Not interpretable as a linear model or simple decision tree



Qualitative statement on ensemble models

Identify the important predictors by looking at **Variable Importance**

Variable Importance

Variable Importance - Averaging or summing the improvement in **Gini or Entropy for a classification model** and **RSS for a regression model** for all the variables

Bagged tree ensemble model contains many tree models

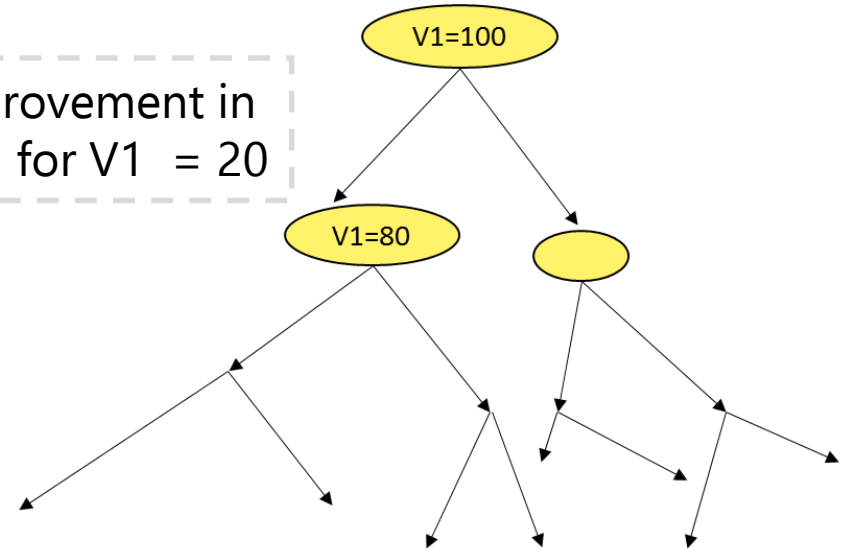
Feature importance of each variable in each of the constituent trees

Tracking the decrease in Gini metric and weighing this decrease appropriately

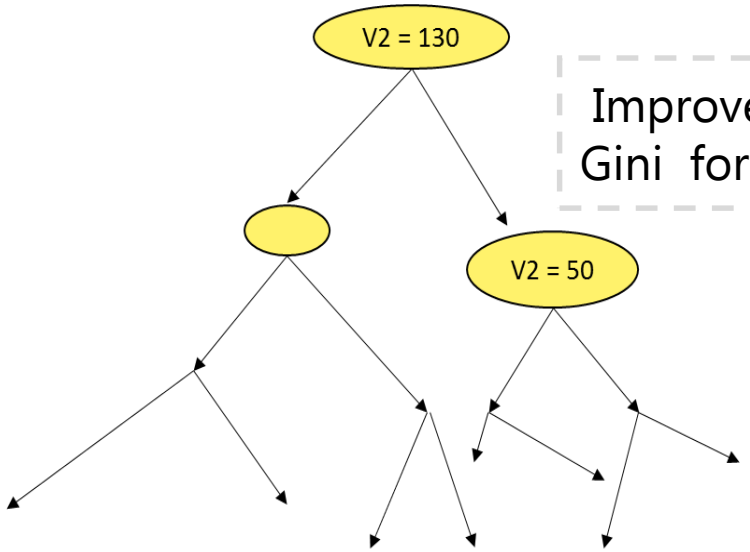
Variable Importance

Example

Improvement in Gini for V1 = 20



Tree 1
Gini Measure for each split



Improvement in Gini for V2 = 80

Tree 2
Gini Measure for each split

Computing variable importance for all the variables used in the split

Variable Importance

Ensemble has N trees

Improvement in Gini/RSS Across Splits

Variable	Tree 1	Tree 2	Tree 3	.....	Tree N
V1	300	30	12	.....	0
V2	600	0	200	.....	150
...	...	...	...	.....	...
V_k	120	450	30	.....	19

Variable	Variable Importance
V1	$\frac{(300 + 30 + 12 + \cdots + 0)}{N}$
V2	$\frac{(600 + 0 + 200 + \cdots + 150)}{N}$
...	....
V_k	$\frac{(120 + 450 + 30 + \cdots + 19)}{N}$

The average values of importance measures per variable will produce a consolidated number

Parameters of Bagged Trees

What could be the user specified parameters while building a bagged tree model?

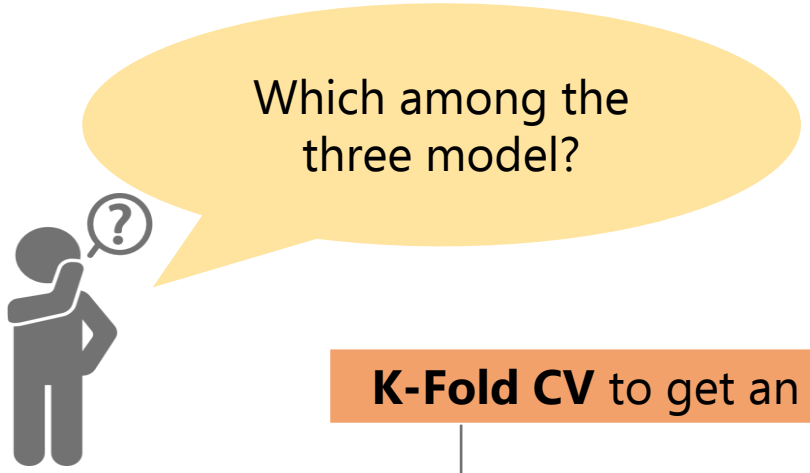
User specified parameters or **Hyperparameters**

- Number of tree used to build an ensemble
- Depth of the tree
- Number of observations per node of a tree

Parameters of Bagged Trees

User specified parameters have an implication

Different ensemble model depending on different parameters



K-Fold CV to get an estimate of out of sample error

Expensive

Out of Bag Error is generally used in most tree based models

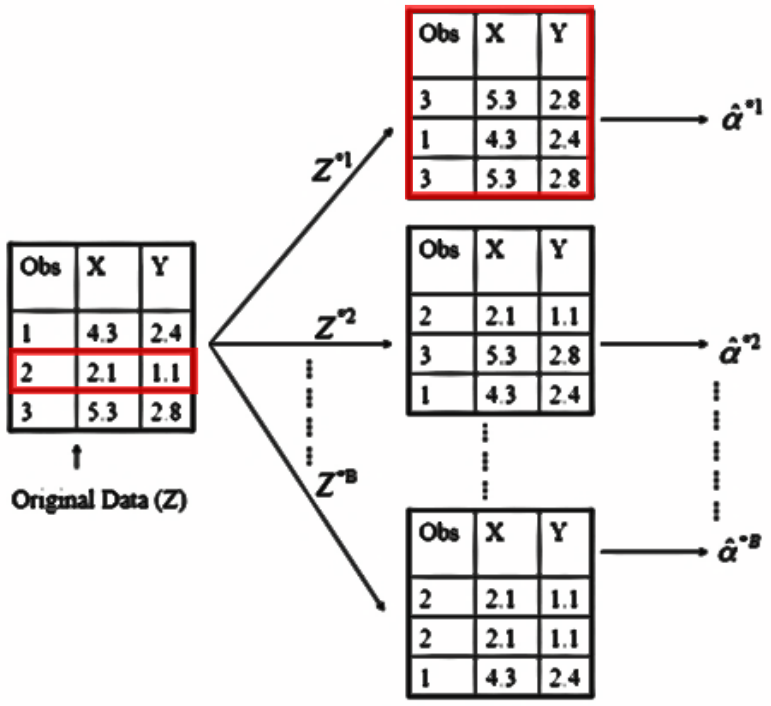
Model 1:
Trees = 100
Depth of Tree = 4

Model 2:
Trees = 150
Depth of Tree = 3

Model 3:
Trees = 500
Depth of Tree = 4

Out Of Bag Error (OOB)

In Bootstrap Sampling, some observation gets left out from the original data



Tree 1

Row 2 is **out of bag**

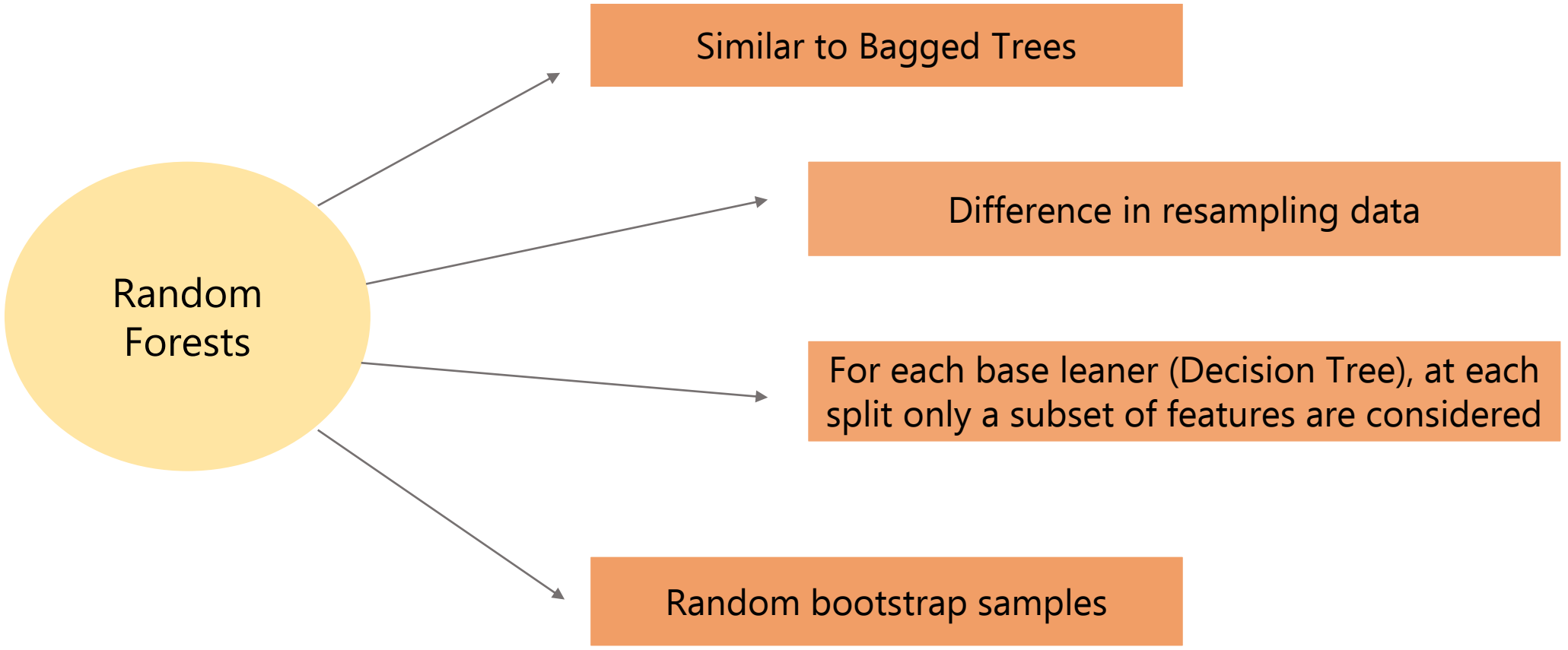
Tree 2

Tree B

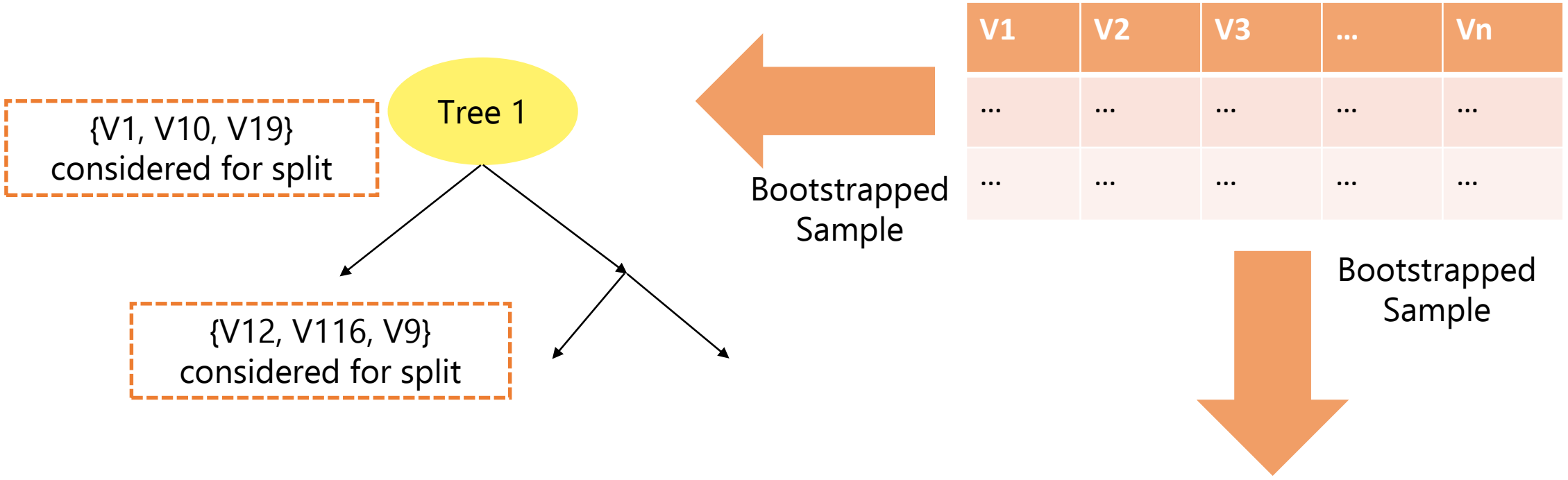
Row 3 is **out of bag**

Average Out of Bag observations in Bootstrapped Sampling is around **33%**

Random Forests



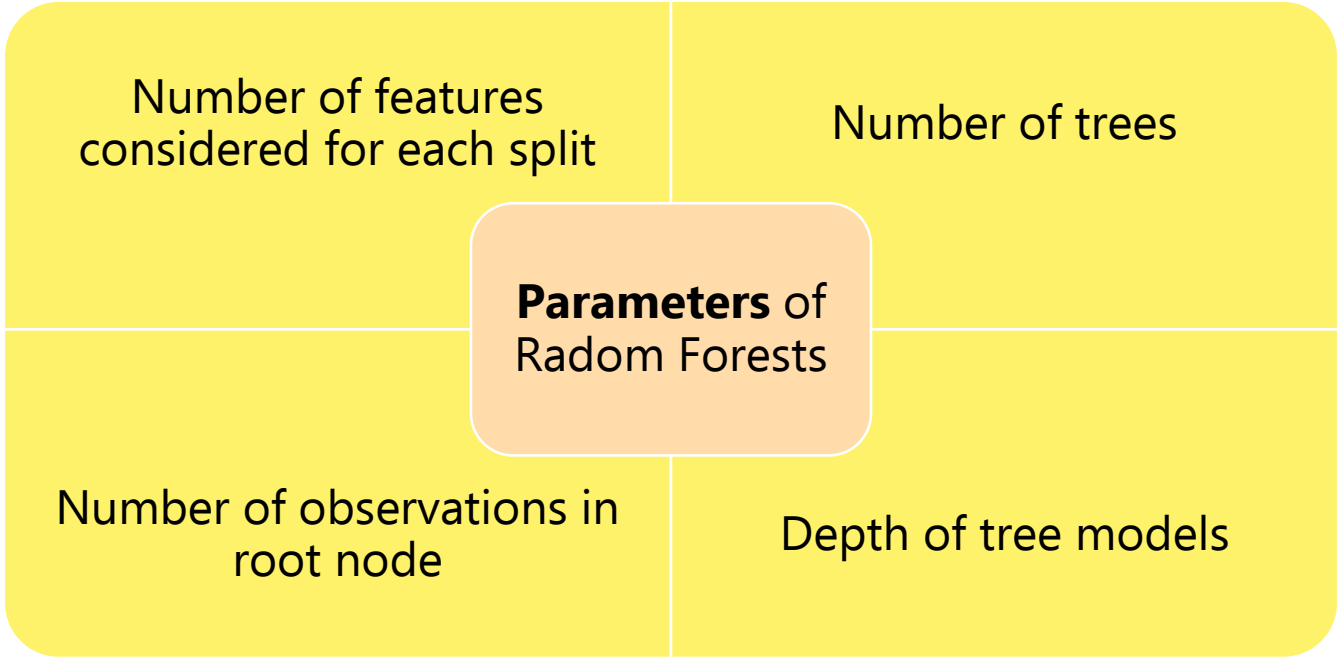
Random Forests



Random Forests

Random Forests uses tree models as base learner

Extract **Variable Importance** or compute **Out-of-bag Error** to get an estimate of out of sample model performance for parameter tuning



Recap

- Tree based ensembles overview
- Tree based ensembles models – Bagged Tree and Random Forests
- Bootstrapped sampling
- Variable importance
- Out of bag error (OOB error)
- Random Forests
- Code Demo

Introduction to Machine Learning

Class

Tree Based Models

Topic

Tree Based Ensembles: Adaboost and Gradient Boosting

Adaboost

Boosting – Another ensemble technique built using decision trees as base learners

Boosted trees works differently than Bagged trees and Random forests

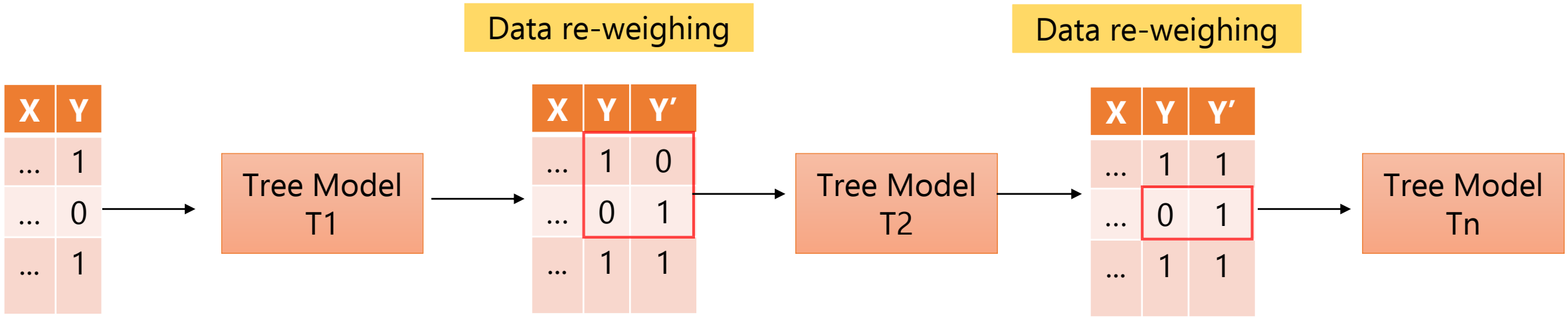
	Boosted Trees	Bagged Trees	Random Forests
Used to build an ensemble	Data re-weighing strategy	Bootstrapped samples	
Depth of tree	Not large (2 to 3 levels)	As many required	

Adaboost is a popular technique

Data Re-weighing Strategy

Classification task using a data set

Wherever the model makes a mistake, that row is given more importance



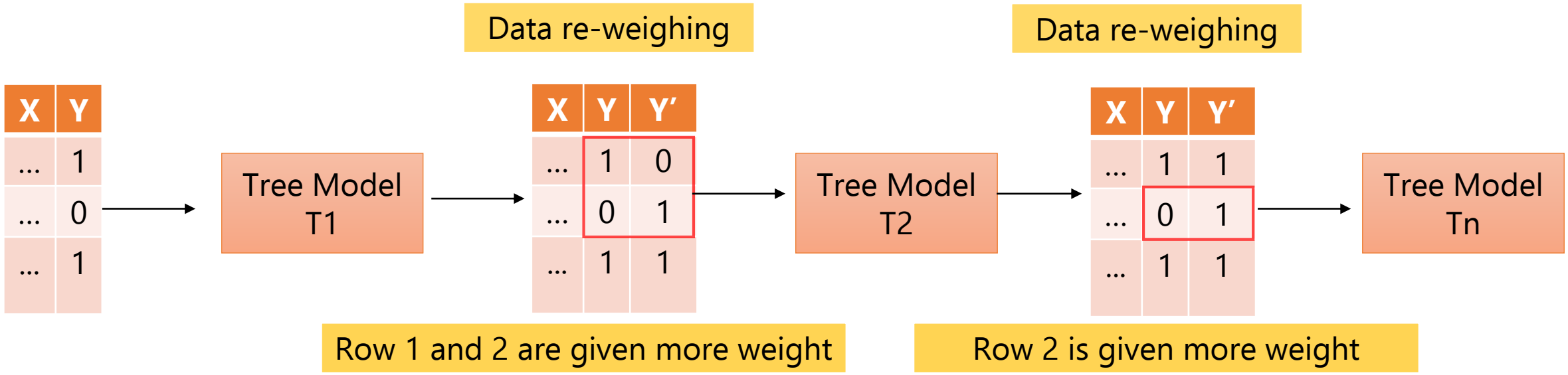
Tree models will be shallow

Final model is a combination of these trees (T1, T2,... Tn)

Data Re-weighing Strategy

Classification task using a data set

Wherever the model makes a mistake, that row is given more importance



Re-weighing strategy - each successive tree pays more attention to the parts of the data that preceding trees have failed to correctly predict

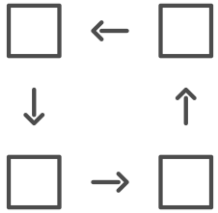
Successive trees try to improve the error rate

Gradient Boosting

Gradient Boosting is another popular boosting technique

Regression task using a dataset

Yet, the mechanics of the discussions are valid in a classification task



Gradient boosting is an iterative algorithm

Gradient Boosting

Step 1

Data set with 1 predictor and 1 target variable

X	Y
30	32
48	62
19	23
22	25

→ T1

Simple tree model

Predictions

X	Y	Y'	Residual
30	32	31	1
48	62	60	2
19	23	24	-1
22	25	26	-1

Error

→ T2

Error – difference between the actual variable and predicted variable

Step 2

X	Y
30	32
48	62
19	23
22	25

→ T1+T2

Combination of 2 tree models

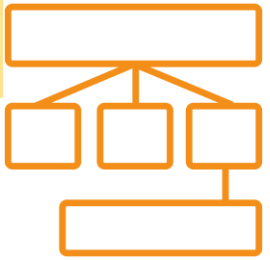
X	Y	Y'	Residual
30	32	92.5	0.9
48	62	61	1
19	23	23.5	-0.5
22	25	25.5	-0.5

→ T3

Gradient Boosting

Keep on repeating this process quite a few times and eventually end up with an ensemble of trees

Gradient boosting is a general ensemble framework



Boosting trees can use other base learner other than decision tree

Partial Dependence Plot

Not as interpretable as simple models like decision trees or linear models

Ensembles provide a list of important predictors by computing variable importance measures

Able to calculate the variables that are important predictors

What is the direction of impact a given predictor has on the dependent variable?

Is the given predictor positively or negatively impacting the dependant variable?

Partial Dependence Plot

Partial Dependence Plot – helps in understanding relationships between a dependent variable and an independent variable

Help in establishing the direction of impact of a predictor on target variable

Depending on which machine learning framework is being used, partial dependence plots for the ensembles may or may not be supported

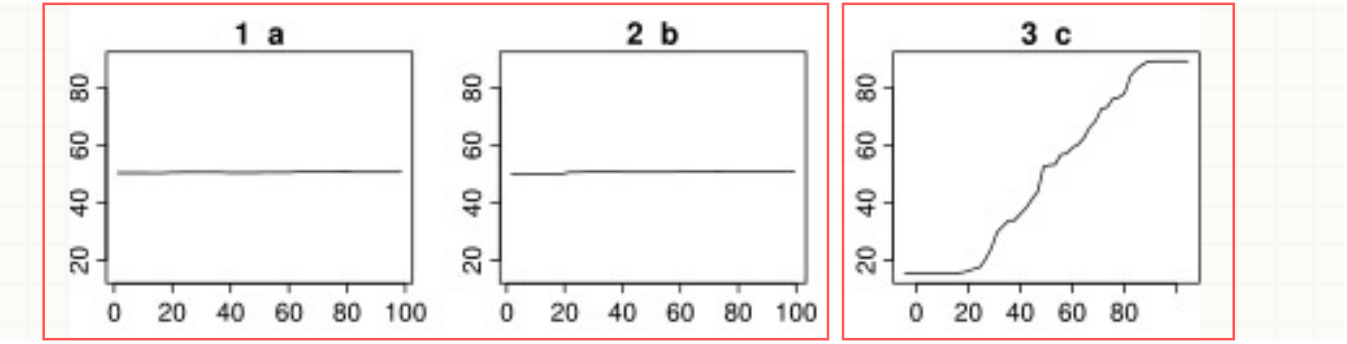
Drawbacks

- Only bivariate relationships can be understood but unearthing interaction effects can be difficult
- Creating partial dependence plot is computationally expensive

Partial Dependence Plot

Partial dependence plots helps identifying the relationship between value of target and the value of a predictor variable after considering the effect of all the other variables

Y axis = values of target variable



3 partial dependence plots

X axis = values of a predictor

Plot 1 and 2 - Value of target variable doesn't change

Plot 3 - Positive relationship between the target variable and the predictor variable

Code Demo

Recap

- Adaboost
- Data re-weighing strategy
- Gradient boosting
- Partial dependence plot
- Code demo