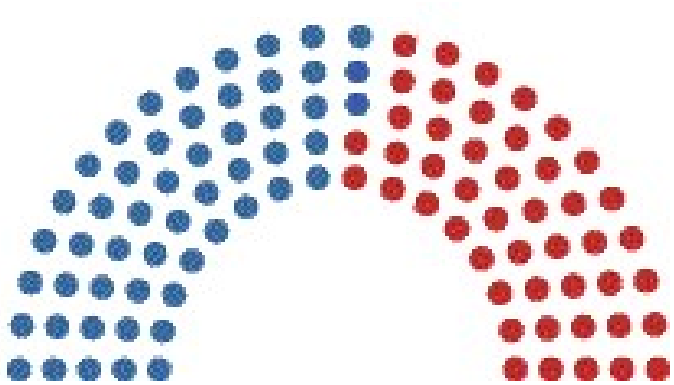
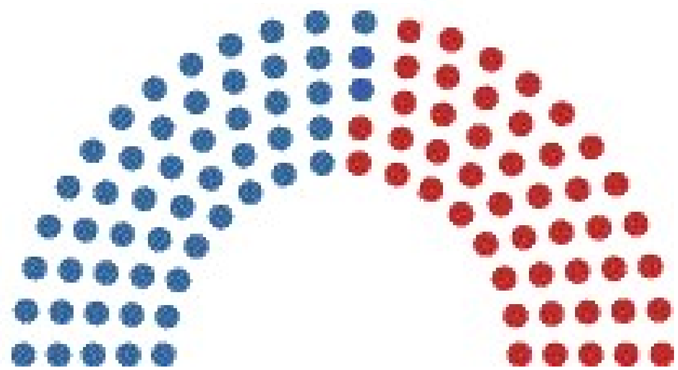
Trees and Forests

Partitioning the Senators With Trees

Partitioning the Senators With Trees

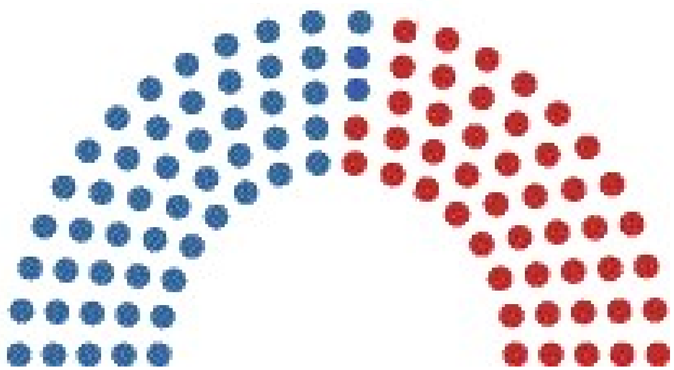


Partitioning the Senators With Trees



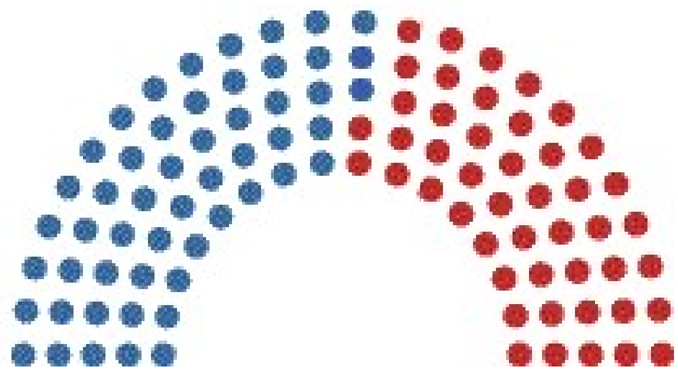
Idea our Senators are defined by their attributes.

Partitioning the Senators With Trees



Idea our Senators are defined by their attributes. Suppose we (optimally) split ('partition') the Senators with respect to x_1, such that we form two subsets of our training data.

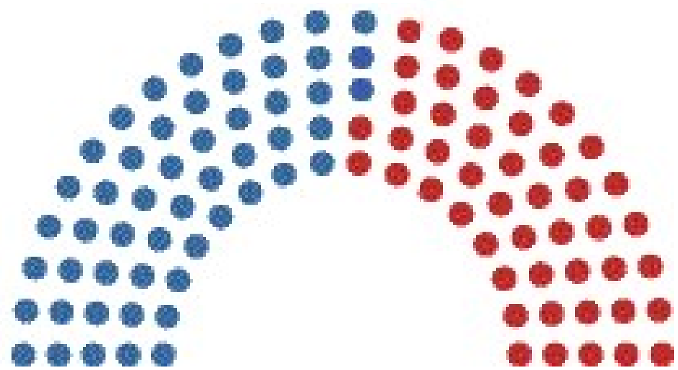
Partitioning the Senators With Trees



Idea our Senators are defined by their attributes. Suppose we (optimally) split ('partition') the Senators with respect to x_1, such that we form two subsets of our training data.

e.g suppose that Republicans generally use 'guns' more than Democrats,

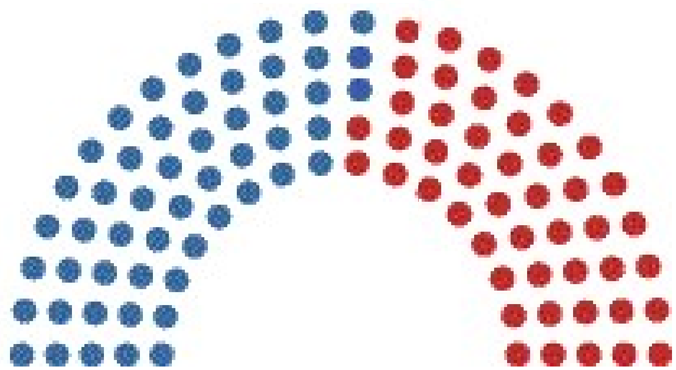
Partitioning the Senators With Trees



Idea our Senators are defined by their attributes. Suppose we (optimally) split ('partition') the Senators with respect to x_1, such that we form two subsets of our training data.

e.g suppose that Republicans generally use 'guns' more than Democrats, such that grabbing all the observations for which $x_{guns} > 0.621$ captures, say, 80% of the Republicans in our data.

Partitioning the Senators With Trees

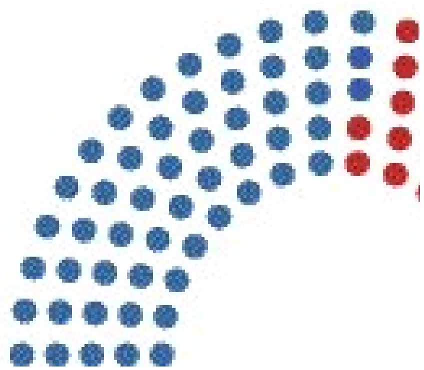


Idea our Senators are defined by their attributes. Suppose we (optimally) split ('partition') the Senators with respect to x_1, such that we form two subsets of our training data.

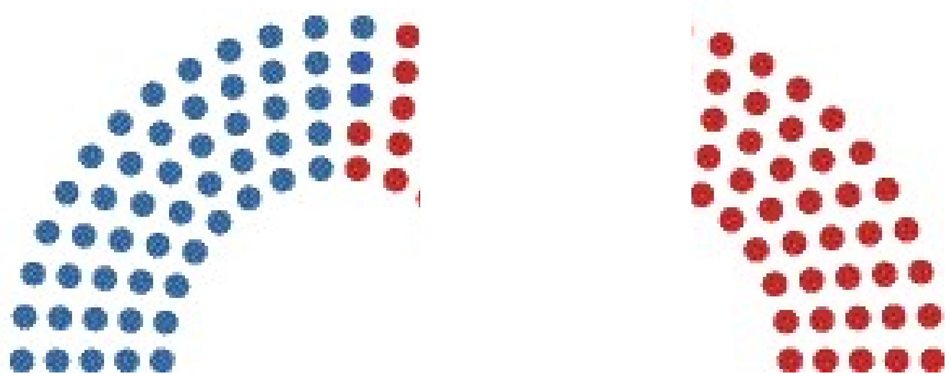
e.g suppose that Republicans generally use 'guns' more than Democrats, such that grabbing all the observations for which $x_{guns} > 0.621$ captures, say, 80% of the Republicans in our data.

Tree, stage 1

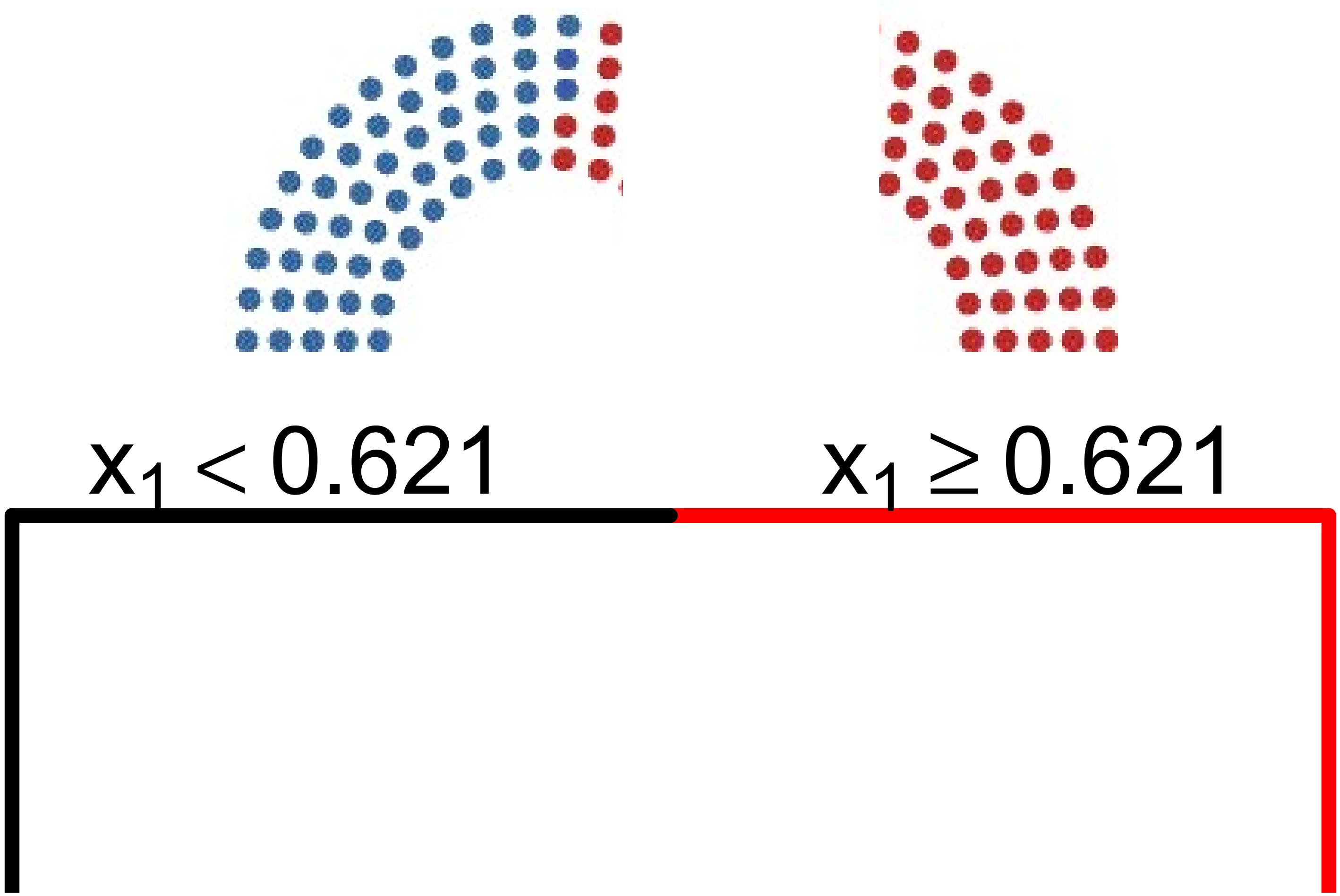
Tree, stage 1



Tree, stage 1



Tree, stage 1



Now, x_2 . . .

Now, x_2 ...

$\rightarrow$ we now have two subsets of our training set: a bunch of Republicans (classified correctly based on x_1 alone)

Now, x_2 ...

$\rightarrow$ we now have two subsets of our training set: a bunch of Republicans (classified correctly based on x_1 alone)

Now, x_2 . . .

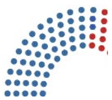
$\longrightarrow$ we now have two subsets of our training set: a bunch of Republicans (classified correctly based on x_1 alone)

and a subset that's still a mix of Republicans and Democrats.

Now, x_2 . . .

$\rightarrow$ we now have two subsets of our training set: a bunch of Republicans (classified correctly based on x_1 alone)

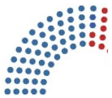
and a subset that's still a mix of Republicans and Democrats.



Now, x_2 . . .

→ we now have two subsets of our training set: a bunch of Republicans (classified correctly based on x_1 alone)

and a subset that's still a mix of Republicans and Democrats.

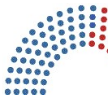


btw The set of Senators we've assigned to Republicans based on their x_1 values are called a leaf.

Now, x_2 ...

$\rightarrow$ we now have two subsets of our training set: a bunch of Republicans (classified correctly based on x_1 alone)

and a subset that's still a mix of Republicans and Democrats.



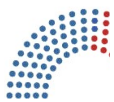
btw The set of Senators we've assigned to Republicans based on their x_1 values are called a leaf.

now suppose we take the mixed group remaining ('internal node') and split them based on x_2, which is their use of '`equality`'.

Now, x_2 . . .

$\rightarrow$ we now have two subsets of our training set: a bunch of Republicans (classified correctly based on x_1 alone)

and a subset that's still a mix of Republicans and Democrats.



btw The set of Senators we've assigned to Republicans based on their x_1 values are called a leaf.

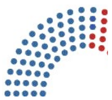
now suppose we take the mixed group remaining ('internal node') and split them based on x_2, which is their use of '`equality`'.

and it turns out that the (remaining) Republicans tend to use this less.

Now, x_2 . . .

$\rightarrow$ we now have two subsets of our training set: a bunch of Republicans (classified correctly based on x_1 alone)

and a subset that's still a mix of Republicans and Democrats.



btw The set of Senators we've assigned to Republicans based on their x_1 values are called a leaf.

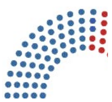
now suppose we take the mixed group remaining ('internal node') and split them based on x_2, which is their use of '`equality`'.

and it turns out that the (remaining) Republicans tend to use this less. So,

Now, x_2 . . .

$\rightarrow$ we now have two subsets of our training set: a bunch of Republicans (classified correctly based on x_1 alone)

and a subset that's still a mix of Republicans and Democrats.



btw The set of Senators we've assigned to Republicans based on their x_1 values are called a leaf.

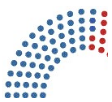
now suppose we take the mixed group remaining ('internal node') and split them based on x_2, which is their use of '`equality`'.

and it turns out that the (remaining) Republicans tend to use this less. So, when we partition according to, say, $x_2 \leq 0.56$ this enables us to perfectly divide this remaining subset into Democrats and Republicans.

Now, x_2 . . .

→ we now have two subsets of our training set: a bunch of Republicans (classified correctly based on x_1 alone)

and a subset that's still a mix of Republicans and Democrats.



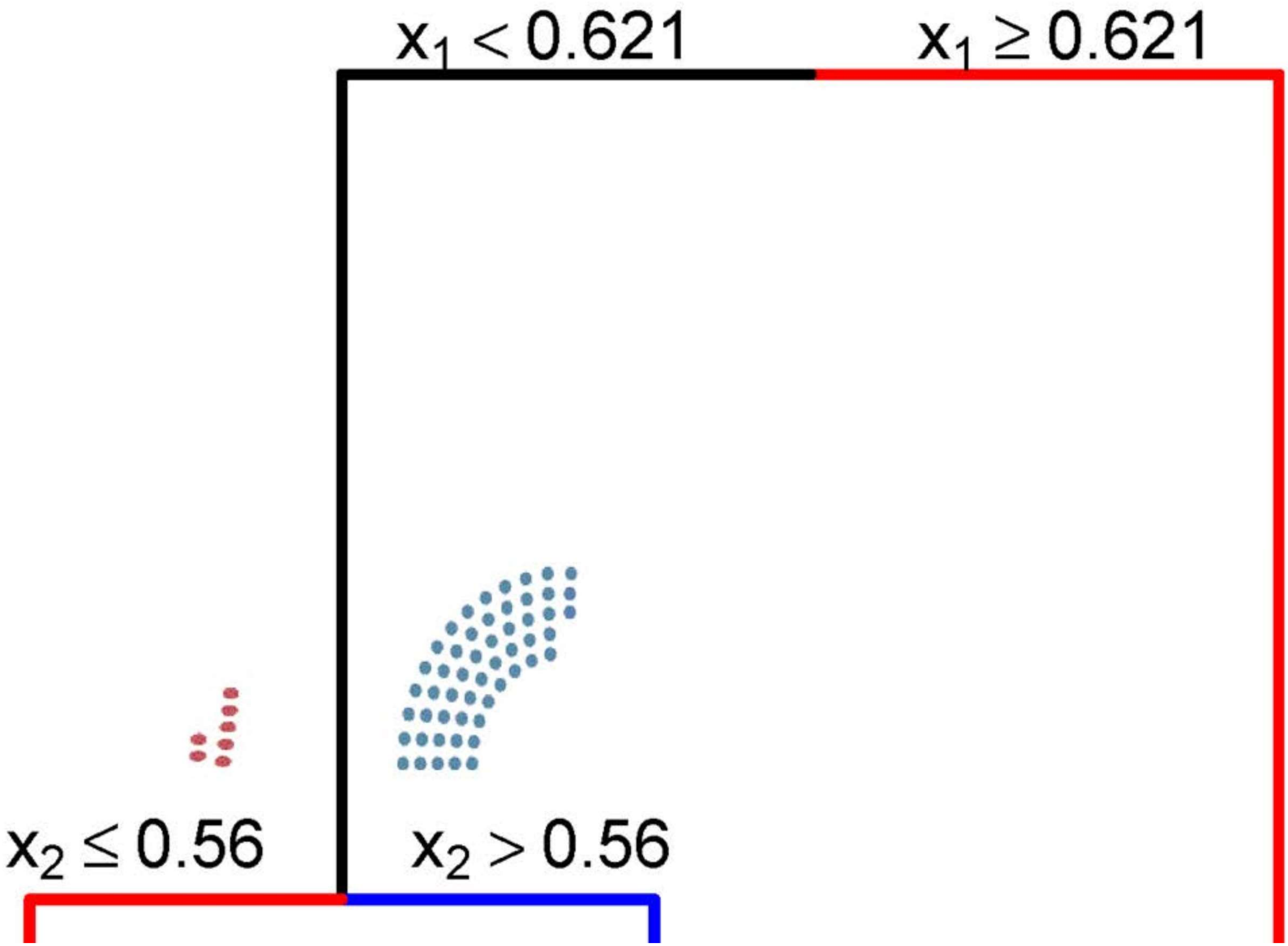
btw The set of Senators we've assigned to Republicans based on their x_1 values are called a leaf.

now suppose we take the mixed group remaining ('internal node') and split them based on x_2, which is their use of '`equality`'.

and it turns out that the (remaining) Republicans tend to use this less. So, when we partition according to, say, $x_2 \leq 0.56$ this enables us to perfectly divide this remaining subset into Democrats and Republicans.

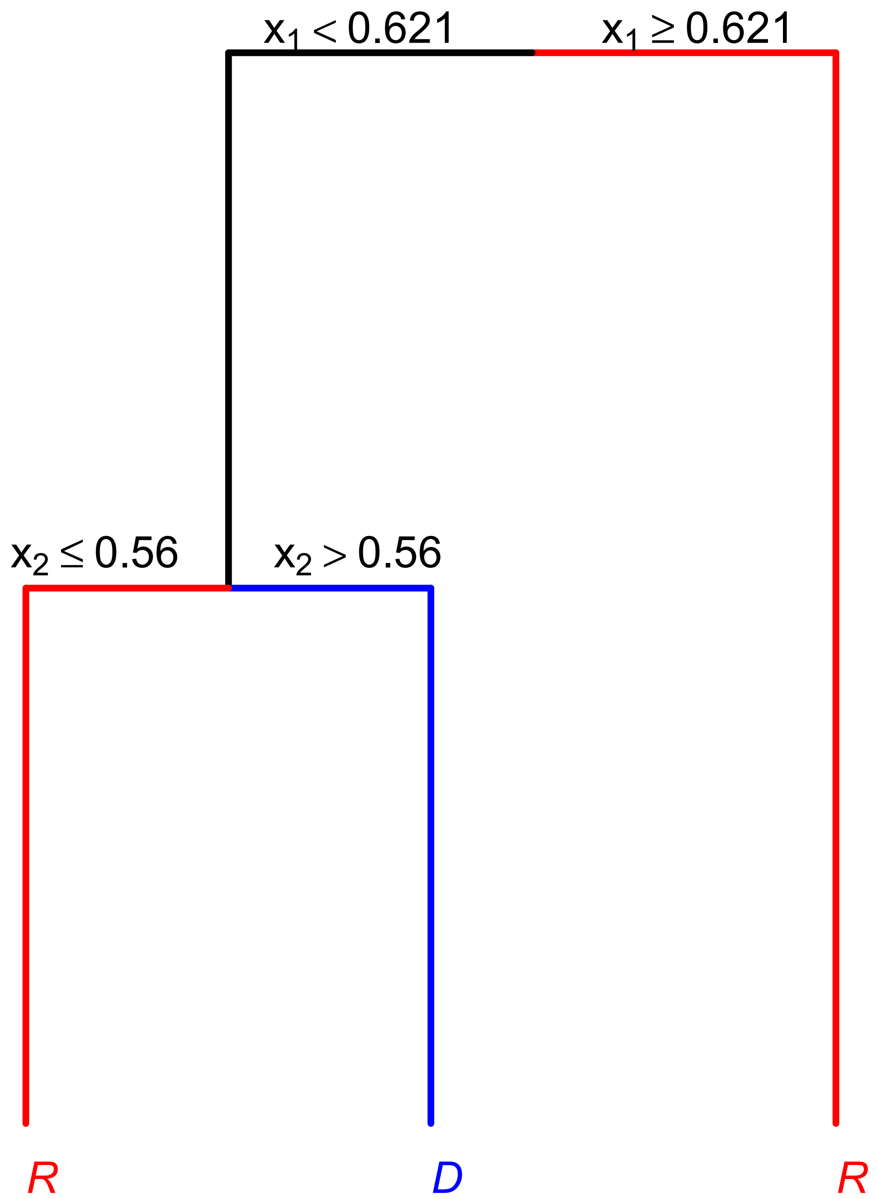
Tree, stage 2

Tree, stage 2



Complete Tree

Complete Tree



Notes

This classifier is known as a tree,

Notes

This classifier is known as a tree, and the recursive partitioning on each derived subset continues until further splits doesn't add 'much' to our classification ability.

Notes

This classifier is known as a tree, and the recursive partitioning on each derived subset continues until further splits doesn't add 'much' to our classification ability.

$\rightarrow$ typically not the case that we can (or want to) classify perfectly into the leaves!

Notes

This classifier is known as a tree, and the recursive partitioning on each derived subset continues until further splits doesn't add 'much' to our classification ability.

$\rightarrow$ typically not the case that we can (or want to) classify perfectly into the leaves!

At each node,

Notes

This classifier is known as a tree, and the recursive partitioning on each derived subset continues until further splits doesn't add 'much' to our classification ability.

→ typically not the case that we can (or want to) classify perfectly into the leaves!

At each node, algorithmic tricks allow fast searching over all the variables to find the one that should be used.

Notes

This classifier is known as a tree, and the recursive partitioning on each derived subset continues until further splits doesn't add 'much' to our classification ability.

$\rightarrow$ typically not the case that we can (or want to) classify perfectly into the leaves!

At each node, algorithmic tricks allow fast searching over all the variables to find the one that should be used.

and clearly need a metric for 'best' split in given x:

Notes

This classifier is known as a tree, and the recursive partitioning on each derived subset continues until further splits doesn't add 'much' to our classification ability.

→ typically not the case that we can (or want to) classify perfectly into the leaves!

At each node, algorithmic tricks allow fast searching over all the variables to find the one that should be used.

and clearly need a metric for 'best' split in given x: typically based on how homogenous the resulting subset of the data is

Notes

This classifier is known as a tree, and the recursive partitioning on each derived subset continues until further splits doesn't add 'much' to our classification ability.

→ typically not the case that we can (or want to) classify perfectly into the leaves!

At each node, algorithmic tricks allow fast searching over all the variables to find the one that should be used.

and clearly need a metric for 'best' split in given x: typically based on how homogenous the resulting subset of the data is

e.g. 'Gini impurity' and 'Variance Reduction'

Notes

This classifier is known as a tree, and the recursive partitioning on each derived subset continues until further splits doesn't add 'much' to our classification ability.

→ typically not the case that we can (or want to) classify perfectly into the leaves!

At each node, algorithmic tricks allow fast searching over all the variables to find the one that should be used.

and clearly need a metric for 'best' split in given x: typically based on how homogenous the resulting subset of the data is

e.g. 'Gini impurity' and 'Variance Reduction'

+ Trees are easy to interpret,

Notes

This classifier is known as a tree, and the recursive partitioning on each derived subset continues until further splits doesn't add 'much' to our classification ability.

$\rightarrow$ typically not the case that we can (or want to) classify perfectly into the leaves!

At each node, algorithmic tricks allow fast searching over all the variables to find the one that should be used.

and clearly need a metric for 'best' split in given x: typically based on how homogenous the resulting subset of the data is

e.g. 'Gini impurity' and 'Variance Reduction'

$+$ Trees are easy to interpret, and we can report relative variable importance statistics.

Notes

This classifier is known as a tree, and the recursive partitioning on each derived subset continues until further splits doesn't add 'much' to our classification ability.

→ typically <u>not</u> the case that we can (or want to) classify perfectly into the leaves!

At each node, algorithmic tricks allow fast searching over all the variables to find the one that should be used.

and clearly need a metric for 'best' split in given x: typically based on how homogenous the resulting subset of the data is

e.g. 'Gini impurity' and 'Variance Reduction'

+ Trees are easy to interpret, and we can report relative variable importance statistics.

But. . .

But. . .

These basic CART (Classification and Regression Trees) approaches show instability in practice:

But. . .

These basic CART (Classification and Regression Trees) approaches show instability in practice:

i.e. minor changes to training data can have large consequences for classification decisions,

But. . .

These basic CART (Classification and Regression Trees) approaches show instability in practice:

i.e. minor changes to training data can have large consequences for classification decisions, because any error is propagated down the tree.

But. . .

These basic CART (Classification and Regression Trees) approaches show instability in practice:

i.e. minor changes to training data can have large consequences for classification decisions, because any error is propagated down the tree.

$\rightarrow$ related to problems of overfitting in the training data.

But...

These basic CART (Classification and Regression Trees) approaches show instability in practice:

i.e. minor changes to training data can have large consequences for classification decisions, because any error is propagated down the tree.

$\rightarrow$ related to problems of overfitting in the training data.

So effort is made to prune the trees back

But. . .

These basic CART (Classification and Regression Trees) approaches show instability in practice:

i.e. minor changes to training data can have large consequences for classification decisions, because any error is propagated down the tree.

$\rightarrow$ related to problems of overfitting in the training data.

So effort is made to prune the trees back—remove less helpful branches.

But. . .

These basic CART (Classification and Regression Trees) approaches show instability in practice:

i.e. minor changes to training data can have large consequences for classification decisions, because any error is propagated down the tree.

$\rightarrow$ related to problems of overfitting in the training data.

So effort is made to prune the trees back—remove less helpful branches.

Or can construct many trees (from slightly different samples of the data) and average over them:

But. . .

These basic CART (Classification and Regression Trees) approaches show instability in practice:

i.e. minor changes to training data can have large consequences for classification decisions, because any error is propagated down the tree.

$\rightarrow$ related to problems of overfitting in the training data.

So effort is made to prune the trees back—remove less helpful branches.

Or can construct many trees (from slightly different samples of the data) and average over them: known as bagging ('bootstrap aggregating').

But. . .

These basic CART (Classification and Regression Trees) approaches show instability in practice:

i.e. minor changes to training data can have large consequences for classification decisions, because any error is propagated down the tree.

$\rightarrow$ related to problems of overfitting in the training data.

So effort is made to prune the trees back—remove less helpful branches.

Or can construct many trees (from slightly different samples of the data) and average over them: known as bagging ('bootstrap aggregating').

$\rightarrow$ random forests combines *many* trees (forest) and

But...

These basic CART (Classification and Regression Trees) approaches show instability in practice:

i.e. minor changes to training data can have large consequences for classification decisions, because any error is propagated down the tree.

$\rightarrow$ related to problems of overfitting in the training data.

So effort is made to prune the trees back—remove less helpful branches.

Or can construct many trees (from slightly different samples of the data) and average over them: known as bagging ('bootstrap aggregating').

$\rightarrow$ random forests combines *many* trees (forest) and at each split a *random sample* of features is considered (rather than all features).

Using Random Forests (in regression context)...

Using Random Forests (in regression context)...

what stems are most important predictors of treaty harshness? (note: $p >> n$)

Using Random Forests (in regression context)...

what stems are most important predictors of treaty harshness? (note: $p >> n$)

use reduction in mean square error when that term is used in a tree

Using Random Forests (in regression context). . .

what stems are most important predictors of treaty harshness? (note: $p >> n$)

use reduction in mean square error when that term is used in a tree (increase in MSE when it is not used)

Using Random Forests (in regression context)...

what stems are most important predictors of treaty harshness? (note: $p >> n$)

use reduction in mean square error when that term is used in a tree (increase in MSE when it is not used)

fairly robust to TDM construction choices

Using Random Forests (in regression context)...

what stems are most important predictors of treaty harshness? (note: $p >> n$)

use reduction in mean square error when that term is used in a tree (increase in MSE when it is not used)

fairly robust to TDM construction choices