

FIAP

MBA

Agenda

1. Ensemble Methods

1. Definição
2. Tipos

2. Random Forest

1. Bias x Variance
2. Intuição
3. Avaliando uma Random Forest
4. Escolhendo o número de Features
5. Lidando com valores faltantes

Ensemble Methods

Ensemble Methods

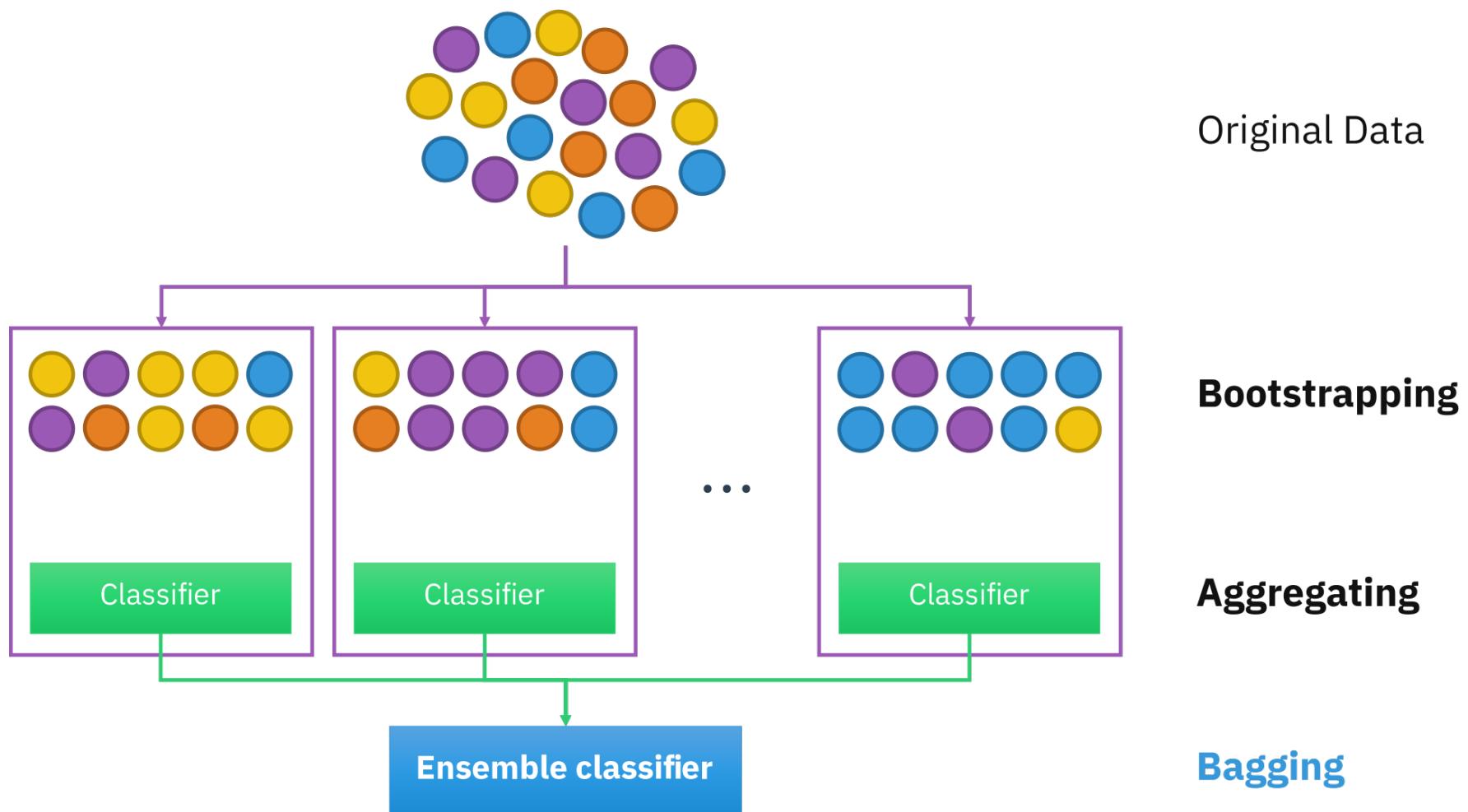
Em machine learning, um método ensemble é um modelo de machine learning que agrupa as previsões de modelos individuais. Visto que eles combinam os resultados de múltiplos modelos, eles são menos propensos a erros e, portanto, tendem a performar melhor.

Os métodos *ensemble* são geralmente classificados em dois tipos. O primeiro tipo combina diferentes modelos de machine learning. São conhecidos como sistemas votantes. O segundo tipo de métodos de *ensemble* combina várias versões do mesmo modelo, como Random Forest, LightGBM e XGBoost.

Aqui, vamos focar no segundo tipo e, dentro dele, vamos iniciar nossos estudos pelo Bagging. No Jupyter, para facilitar o entendimento de Random Forest, eu apresento sistemas votantes.

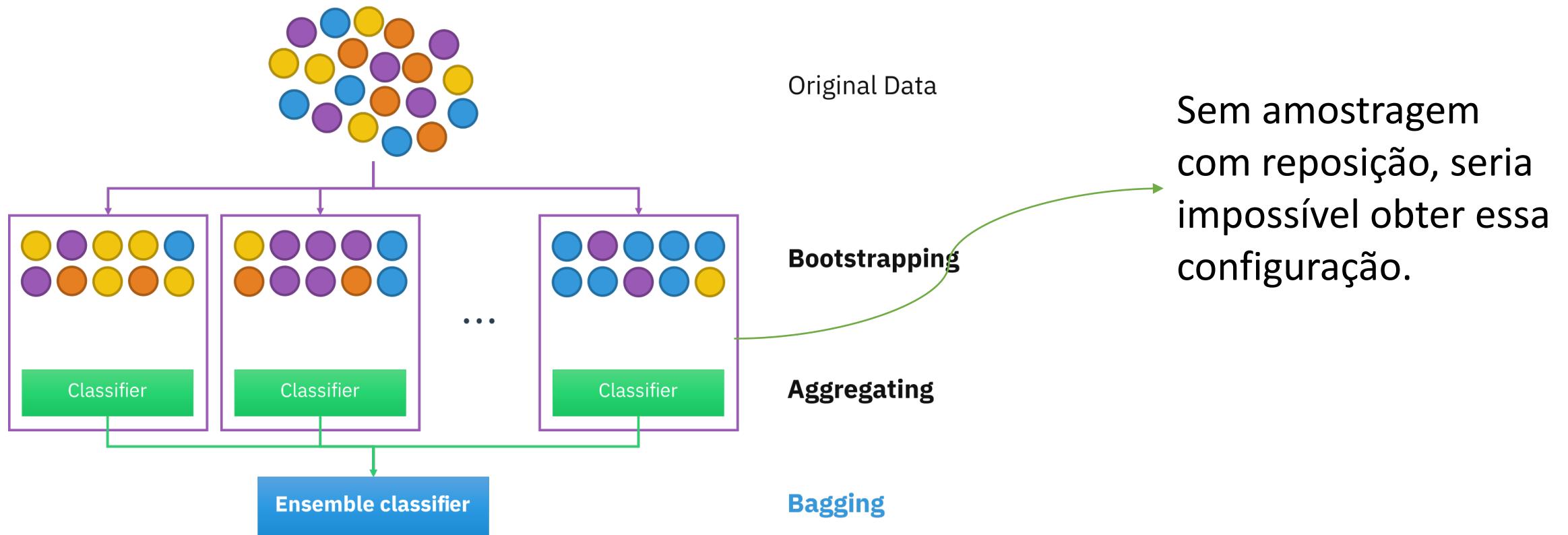
Ensemble Methods

Bagging é uma abreviação para *Bootstrap aggregation*. *Bootstrap* significa amostragem com reposição. Considere a seguinte imagem:



Ensemble Methods

Suponha que você possua um saco (bag) com 20 bolas. Você irá selecionar 10, uma de cada vez. Toda vez que você seleciona uma, você a retorna ao saco. Isso significa que é possível, apesar de extremamente improvável, que você pegue a mesma bola 10 vezes. O mais provável é que você pegue algumas bolas mais que outras, e algumas nunca pegue.



Premissas – Ensemble Methods

E por que isso é importante? Bem, *Bootstrap* funciona por trás das cortinas em Random Forest. Este processo ocorre quando cada árvore de decisão é criada. Se todas as árvores de decisão contivessem os mesmos exemplos, as previsões seriam bem parecidas, fazendo o resultado agregado bem parecido com o resultado de uma árvore individual. Ao invés disso, com Random Forest, as árvores são construídas usando *bootstrap*, usualmente com o mesmo número de amostras no dataset original.

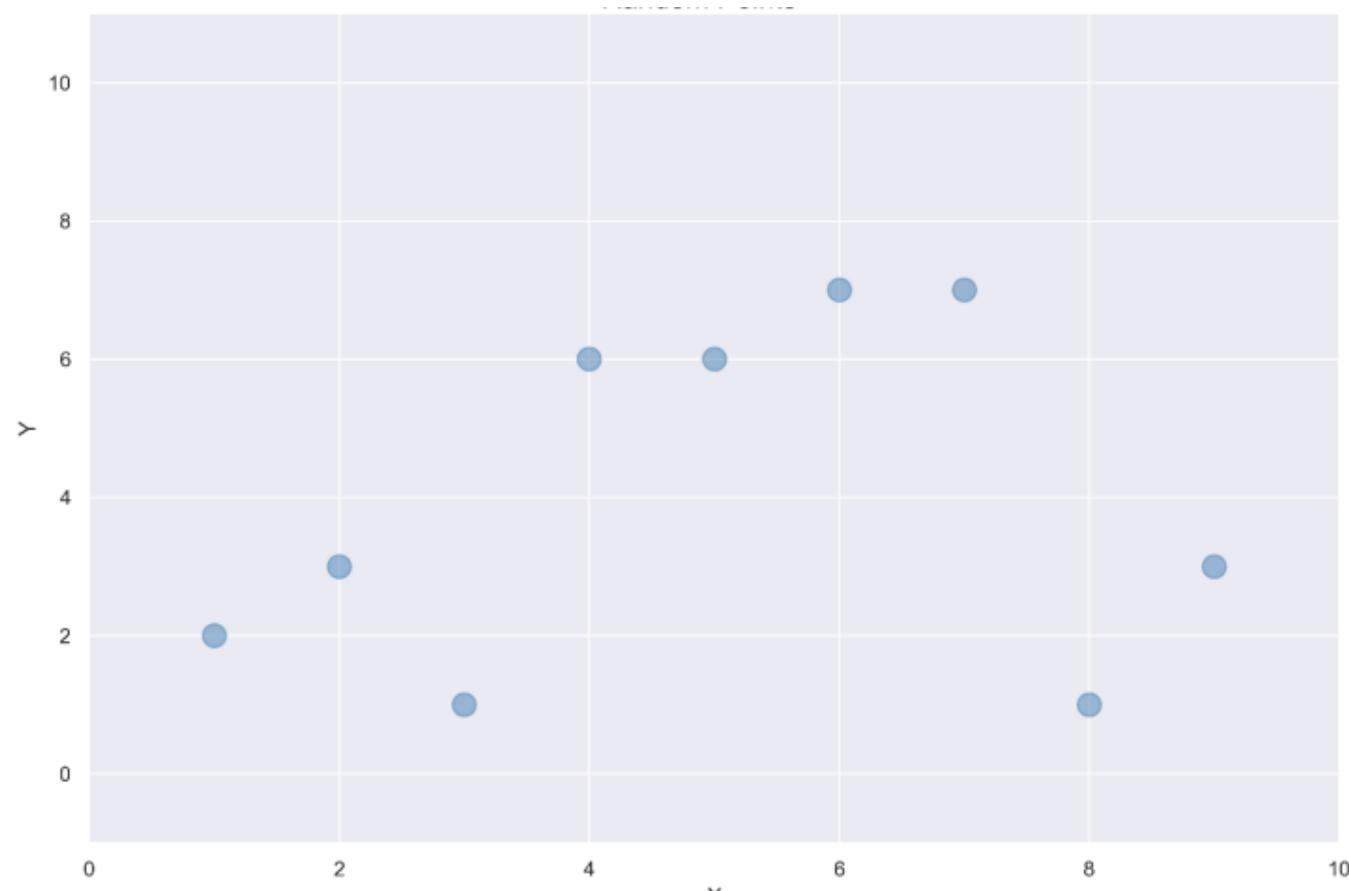
Uma estimativa matemática indica que dois terços das amostras de cada árvore são únicas e um terço incluem duplicatas.

Depois da fase do *bootstrap*, cada árvore de decisão realiza suas previsões individuais. O resultado é uma **floresta de árvores** cujas previsões são agregadas numa única previsão final usando votos majoritários para classificação e média para regressão.

Random Forest

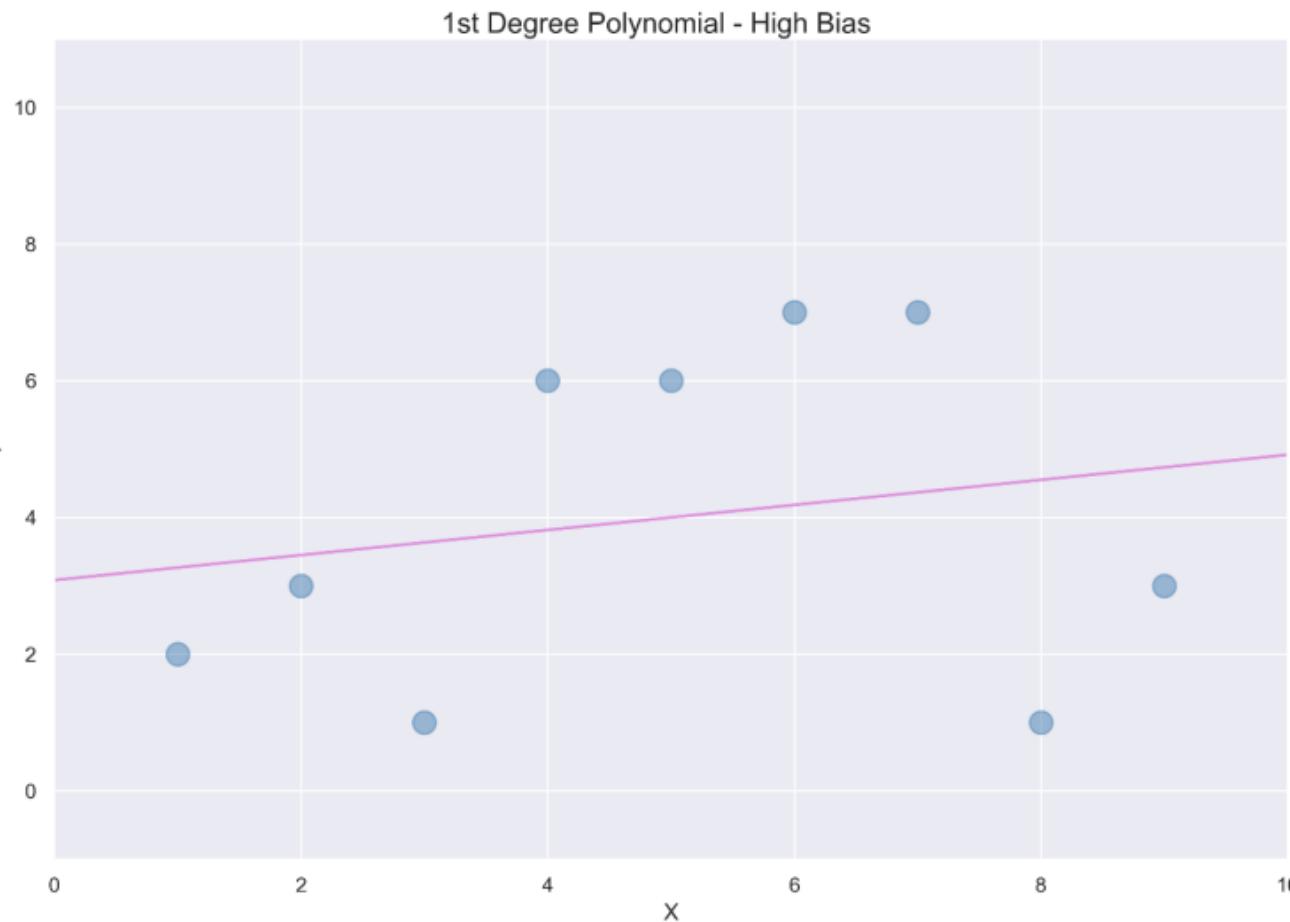
Bias x Variance

Bias e Variance possui relação direta com overfitting. Para entender como ela interfere nos resultados, considere o seguinte conjunto de dados:



Bias x Variance

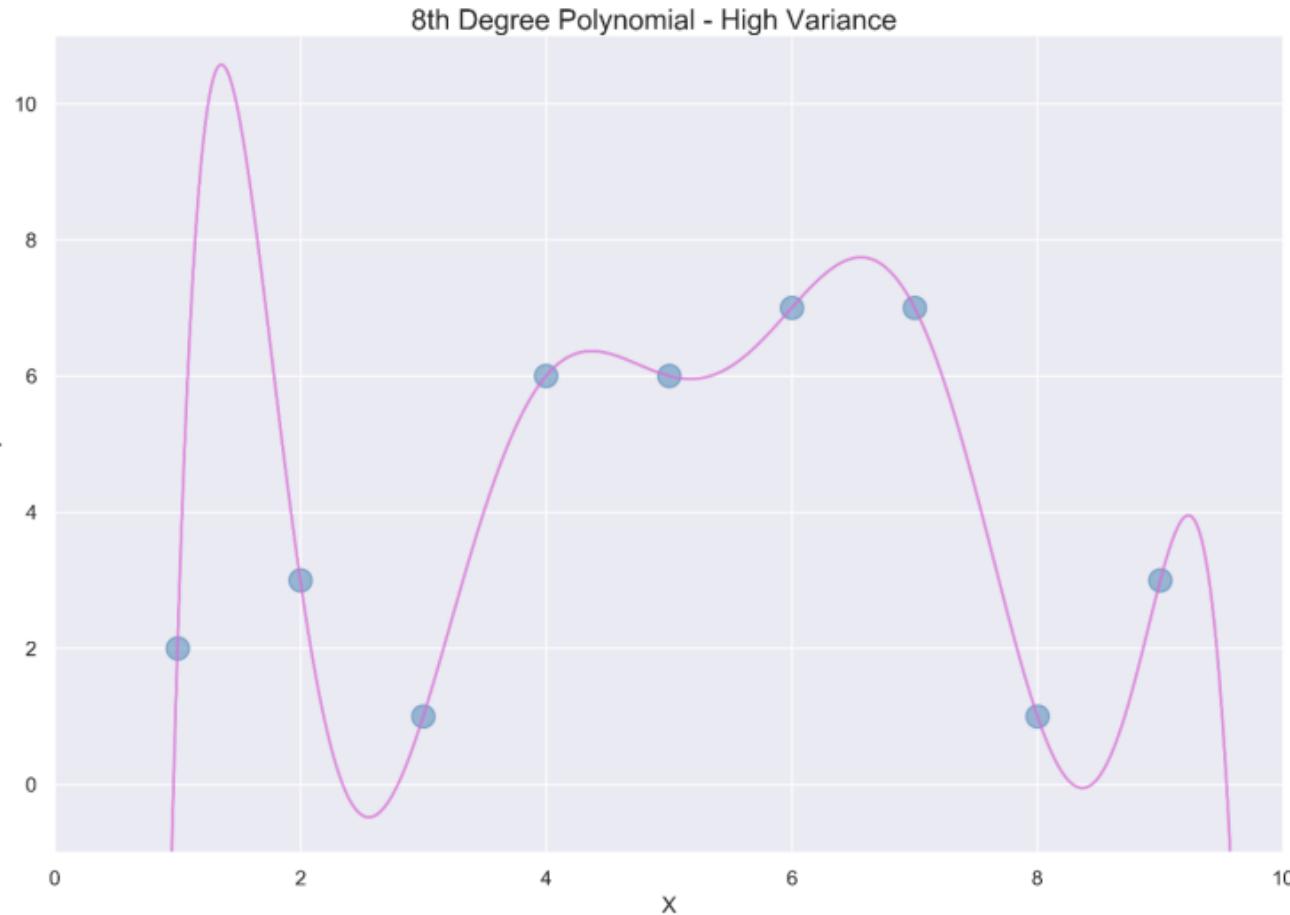
Podemos ajustar uma regressão linear usando MSE como função de custo:



Neste caso, **dizemos que o Bias é alto**, visto que uma simples linha não consegue capturar a complexidade dos dados.

Bias x Variance

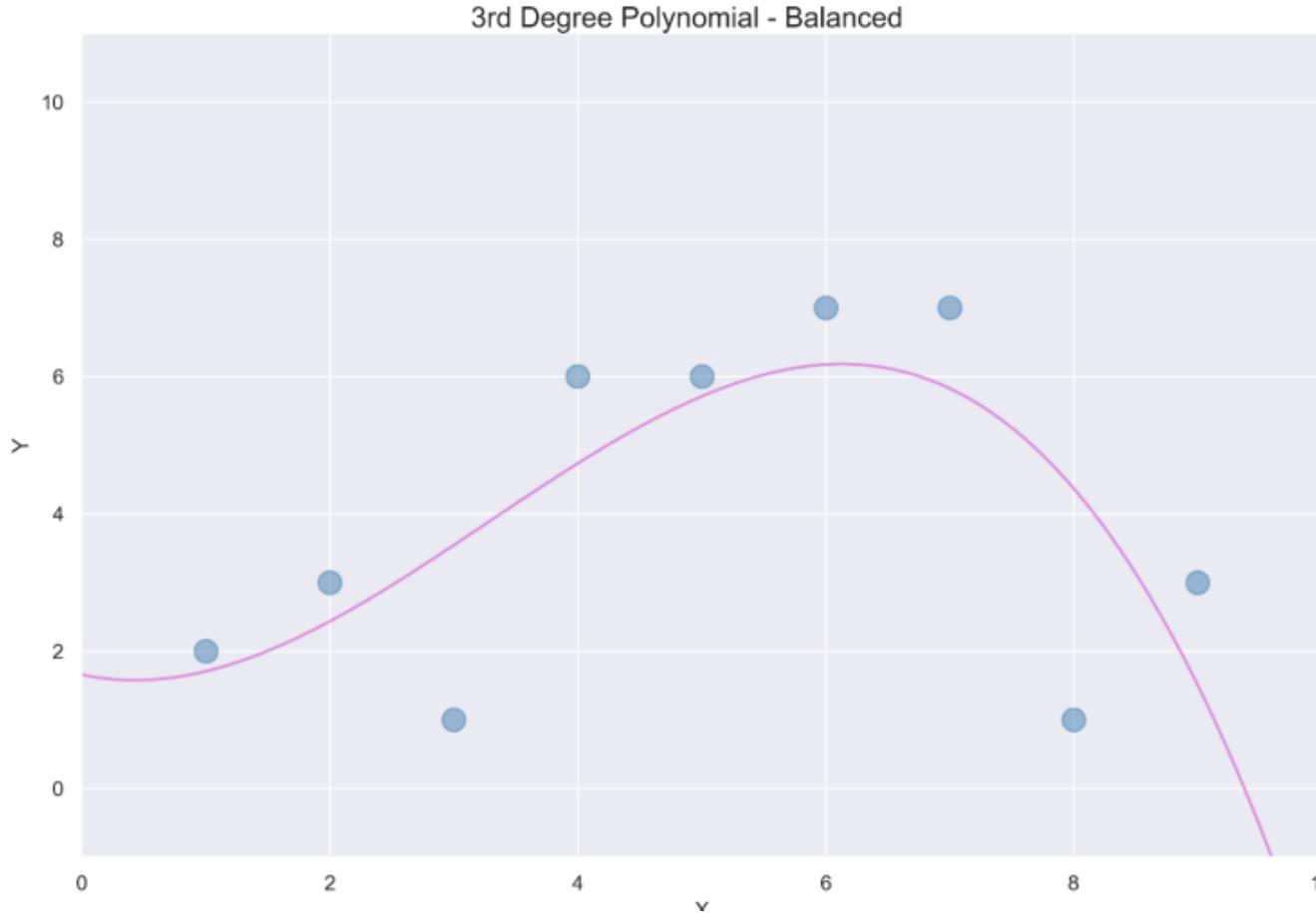
Uma segunda opção é ajustar aos pontos um polinômio de grau 8.



Neste caso, **dizemos que Variance é alto**, visto que o modelo acompanhou todas as mudanças nos dados. Tais modelos tendem a dar overfit nos dados.

Bias x Variance

Uma terceira opção é usar um polinômio de grau 3, como mostrado na figura:



Aqui, temos um bom
balanceamento entre
Bias e Variance.

Intuição

Árvores de decisão, como vimos, são fáceis de construir, fáceis de usar e fáceis de interpretar. Entretanto, "árvores possuem um aspecto que as previne de serem a ferramenta ideal para aprendizado preditivo: **inacurácia.**" (The elements of Statistical Learning)

Na prática, isso significa que elas funcionam muito bem com os dados usados para criá-las, mas não são flexíveis na hora de classificar novas amostras.

Random Forest entra em cena, então, combinando a simplicidade das Árvores de Decisão com flexibilidade, resultando num significativo aumento da acurácia.

Intuição

Para entender o funcionamento da Random Forest, vamos usar esse simples dataset:

Dor no peito	Boa circulação Sanguínea	Artérias bloqueadas	Peso (kg)	Doença cardíaca
N	N	N	57	N
Y	Y	Y	82	Y
Y	Y	N	95	N
Y	N	Y	76	Y

Intuição

O primeiro passo consiste em criar um dataset usando bagging. Para isso, selecionamos aleatoriamente amostras do dataset original. Importante notar que é possível selecionar a mesma amostra mais de uma vez

Dor no peito	Boa circulação Sanguínea	Artérias bloqueadas	Peso (kg)	Doença cardíaca
N	N	N	57	N
Y	Y	Y	82	Y
Y	Y	N	95	N
Y	N	Y	76	Y

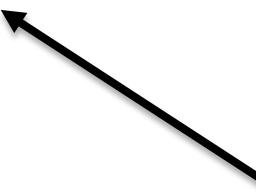
Dor no peito	Boa circulação Sanguínea	Artérias bloqueadas	Peso (kg)	Doença cardíaca
N	N	N	57	N

Intuição

O primeiro passo consiste em criar um dataset usando bagging. Para isso, selecionamos aleatoriamente amostras do dataset original. Importante notar que é possível selecionar a mesma amostra mais de uma vez

Dor no peito	Boa circulação Sanguínea	Artérias bloqueadas	Peso (kg)	Doença cardíaca
N	N	N	57	N
Y	Y	Y	82	Y
Y	Y	N	95	N
Y	N	Y	76	Y

Dor no peito	Boa circulação Sanguínea	Artérias bloqueadas	Peso (kg)	Doença cardíaca
Y	Y	Y	82	Y



Esta é a primeira amostra que selecionamos aleatoriamente

Intuição

O primeiro passo consiste em criar um dataset usando bagging. Para isso, selecionamos aleatoriamente amostras do dataset original. Importante notar que é possível selecionar a mesma amostra mais de uma vez

Dor no peito	Boa circulação Sanguínea	Artérias bloqueadas	Peso (kg)	Doença cardíaca
N	N	N	57	N
Y	Y	Y	82	Y
Y	Y	N	95	N
Y	N	Y	76	Y



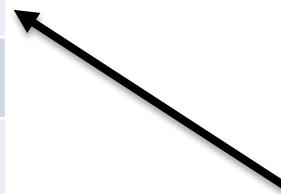
Dor no peito	Boa circulação Sanguínea	Artérias bloqueadas	Peso (kg)	Doença cardíaca
Y	Y	Y	82	Y

Intuição

O primeiro passo consiste em criar um dataset usando bagging. Para isso, selecionamos aleatoriamente amostras do dataset original. Importante notar que é possível selecionar a mesma amostra mais de uma vez

Dor no peito	Boa circulação Sanguínea	Artérias bloqueadas	Peso (kg)	Doença cardíaca
N	N	N	57	N
Y	Y	Y	82	Y
Y	Y	N	95	N
Y	N	Y	76	Y

Dor no peito	Boa circulação Sanguínea	Artérias bloqueadas	Peso (kg)	Doença cardíaca
Y	Y	Y	82	Y

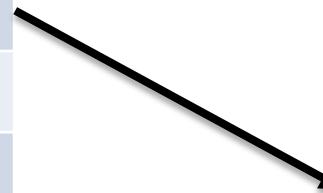


Esta é a segunda amostra que selecionamos aleatoriamente

Intuição

O primeiro passo consiste em criar um dataset usando bagging. Para isso, selecionamos aleatoriamente amostras do dataset original. Importante notar que é possível selecionar a mesma amostra mais de uma vez

Dor no peito	Boa circulação Sanguínea	Artérias bloqueadas	Peso (kg)	Doença cardíaca
N	N	N	57	N
Y	Y	Y	82	Y
Y	Y	N	95	N
Y	N	Y	76	Y



Dor no peito	Boa circulação Sanguínea	Artérias bloqueadas	Peso (kg)	Doença cardíaca
Y	Y	Y	82	Y
N	N	N	57	N

Intuição

O primeiro passo consiste em criar um dataset usando bagging. Para isso, selecionamos aleatoriamente amostras do dataset original. Importante notar que é possível selecionar a mesma amostra mais de uma vez

Dor no peito	Boa circulação Sanguínea	Artérias bloqueadas	Peso (kg)	Doença cardíaca
N	N	N	57	N
Y	Y	Y	82	Y
Y	Y	N	95	N
Y	N	Y	76	Y

Dor no peito	Boa circulação Sanguínea	Artérias bloqueadas	Peso (kg)	Doença cardíaca



Esta é a terceira amostra que selecionamos aleatoriamente

Intuição

O primeiro passo consiste em criar um dataset usando bagging. Para isso, selecionamos aleatoriamente amostras do dataset original. Importante notar que é possível selecionar a mesma amostra mais de uma vez

Dor no peito	Boa circulação Sanguínea	Artérias bloqueadas	Peso (kg)	Doença cardíaca
N	N	N	57	N
Y	Y	Y	82	Y
Y	Y	N	95	N
Y	N	Y	76	Y



Dor no peito	Boa circulação Sanguínea	Artérias bloqueadas	Peso (kg)	Doença cardíaca
Y	Y	Y	82	Y
N	N	N	57	N
Y	N	Y	76	Y

Intuição

O primeiro passo consiste em criar um dataset usando bagging. Para isso, selecionamos aleatoriamente amostras do dataset original. Importante notar que é possível selecionar a mesma amostra mais de uma vez

Dor no peito	Boa circulação Sanguínea	Artérias bloqueadas	Peso (kg)	Doença cardíaca
N	N	N	57	N
Y	Y	Y	82	Y
Y	Y	N	95	N
Y	N	Y	76	Y

Dor no peito	Boa circulação Sanguínea	Artérias bloqueadas	Peso (kg)	Doença cardíaca



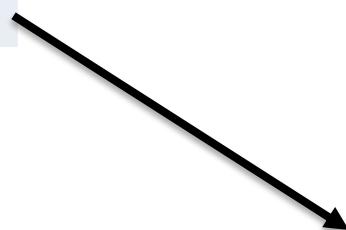
Esta é a quarta amostra que selecionamos aleatoriamente

Intuição

O primeiro passo consiste em criar um dataset usando bagging. Para isso, selecionamos aleatoriamente amostras do dataset original. Importante notar que é possível selecionar a mesma amostra mais de uma vez

Dor no peito	Boa circulação Sanguínea	Artérias bloqueadas	Peso (kg)	Doença cardíaca
N	N	N	57	N
Y	Y	Y	82	Y
Y	Y	N	95	N
Y	N	Y	76	Y

Dor no peito	Boa circulação Sanguínea	Artérias bloqueadas	Peso (kg)	Doença cardíaca
Y	Y	Y	82	Y
N	N	N	57	N
Y	N	Y	76	Y
Y	N	Y	76	Y



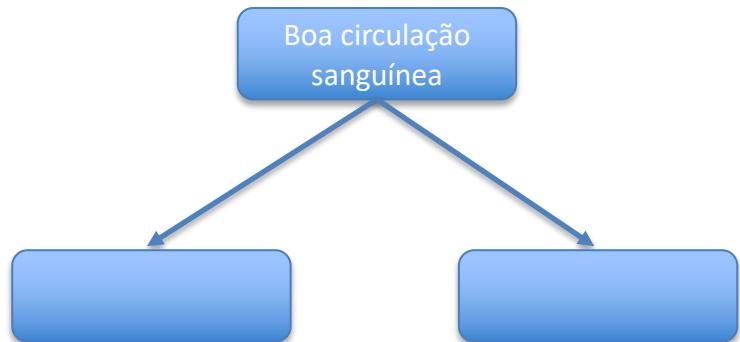
Intuição

O passo seguinte consiste em criar uma Árvore de Decisão usando o dataset criado com bagging, mas usando apenas um subconjunto aleatório de features a cada passo. Neste exemplo, usaremos 2 features (mais adiante falarei sobre como escolher o número de features a se considerar).

Dor no peito	Boa circulação Sanguínea	Artérias bloqueadas	Peso (kg)	Doença cardíaca
Y	Y	Y	82	Y
N	N	N	57	N
Y	N	Y	76	Y
Y	N	Y	76	Y

Intuição

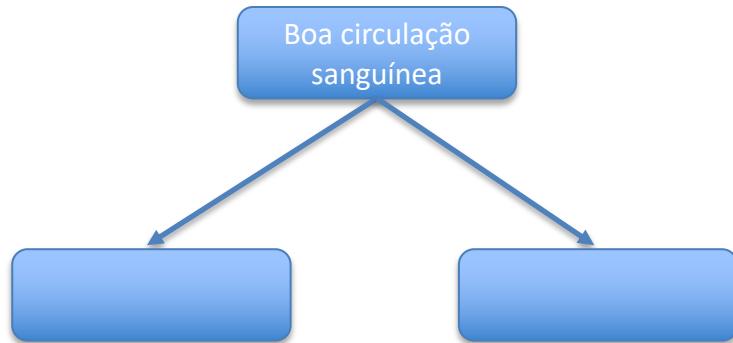
Vamos considerar Boa circulação sanguínea e Artérias bloqueadas como candidatos a nó raiz. A título de exemplo, Boa circulação sanguínea foi escolhido como nó raiz.



Dor no peito	Boa circulação Sanguínea	Artérias bloqueadas	Peso (kg)	Doença cardíaca
Y	Y	Y	82	Y
N	N	N	57	N
Y	N	Y	76	Y
Y	N	Y	76	Y

Intuição

Vamos considerar Boa circulação sanguínea e Artérias bloqueadas como candidatos a nó raiz. A título de exemplo, Boa circulação sanguínea foi escolhido como nó raiz.



Visto que Boa circulação Sanguínea já foi usada, vou marcar ela para focarmos nas features que ainda restam.

Dor no peito	Boa circulação Sanguínea	Artérias bloqueadas	Peso (kg)	Doença cardíaca
Y	Y	Y	82	Y
N	N	N	57	N
Y	N	Y	76	Y
Y	N	Y	76	Y

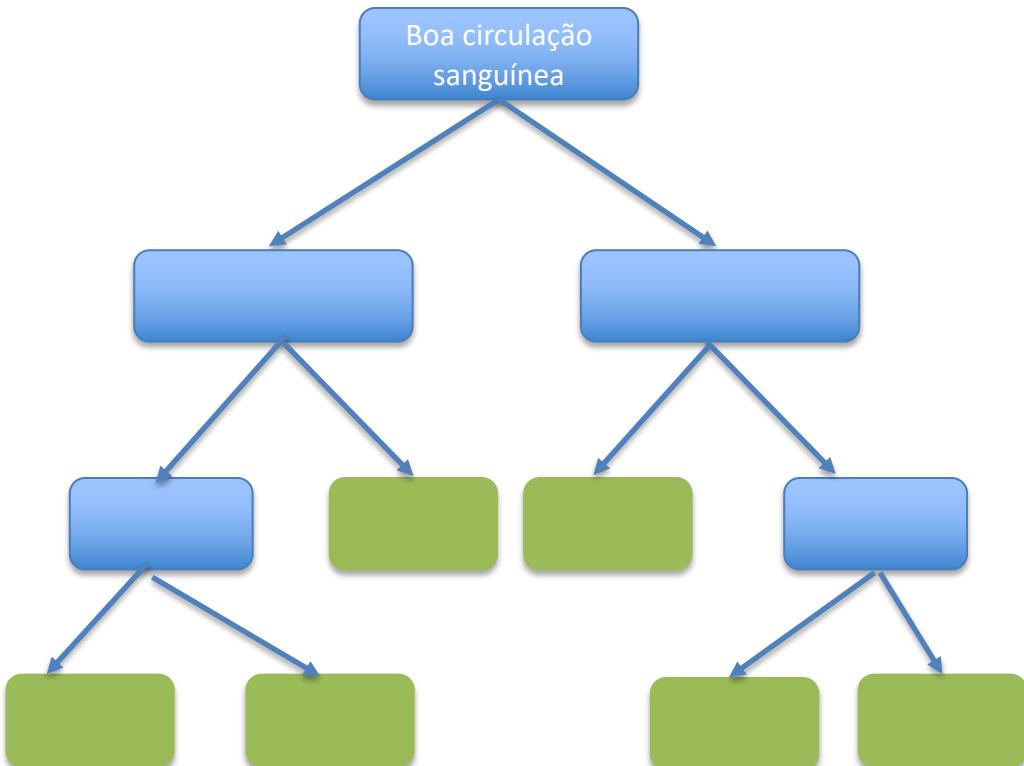
Intuição

Agora, precisamos determinar como separar o nó da esquerda. Para isso, vamos considerar que as 2 features selecionadas como candidatas a nó raiz tenham sido Dor no Peito e Peso



Intuição

E repetimos o processo até criar uma árvore como abaixo, sempre considerando apenas duas features a cada passo



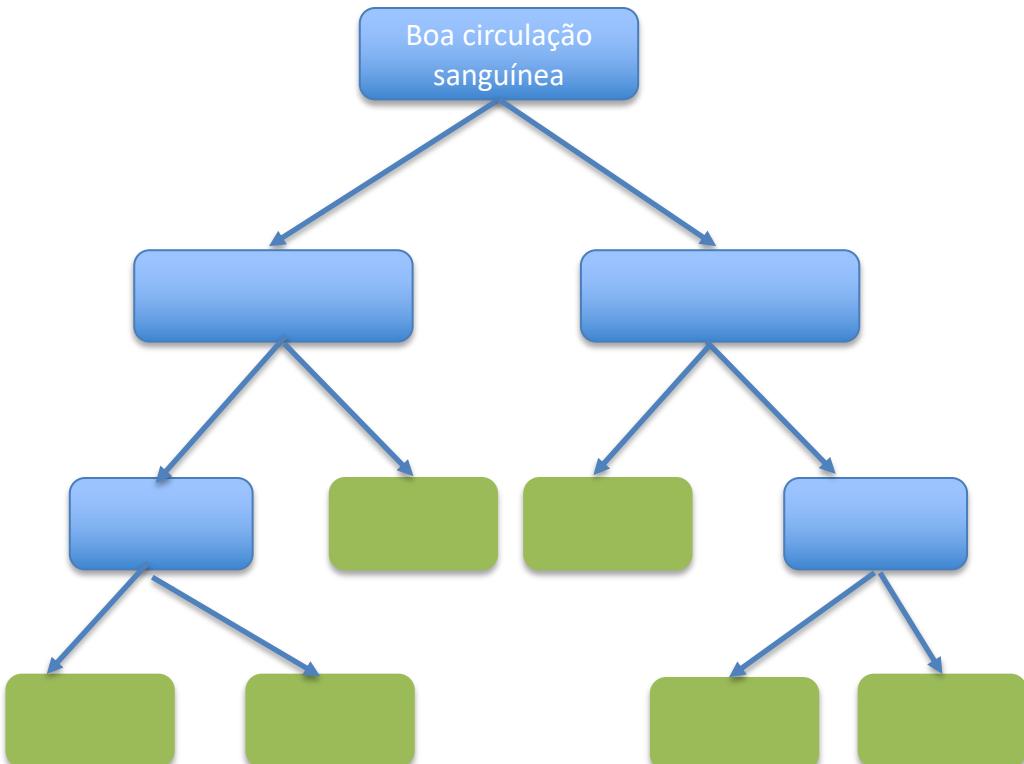
Dor no peito	Boa circulação Sanguínea	Artérias bloqueadas	Peso (kg)	Doença cardíaca
Y	Y	Y	82	Y
N	N	N	57	N
Y	N	Y	76	Y
Y	N	Y	76	Y

Intuição

Recapitulando:

Construímos uma árvore:

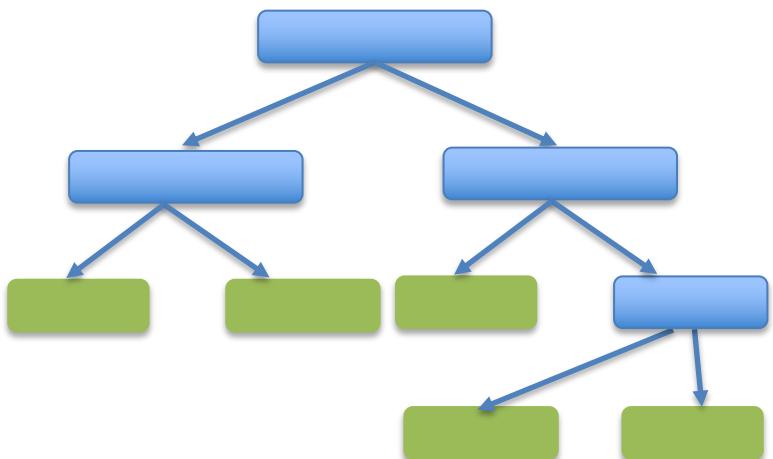
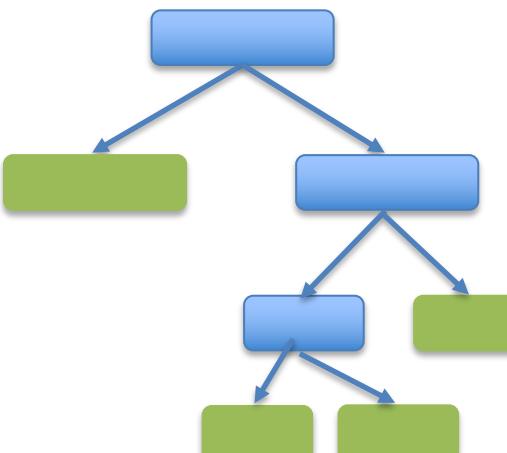
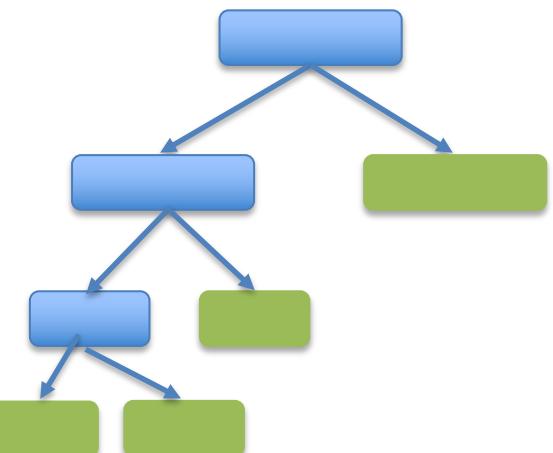
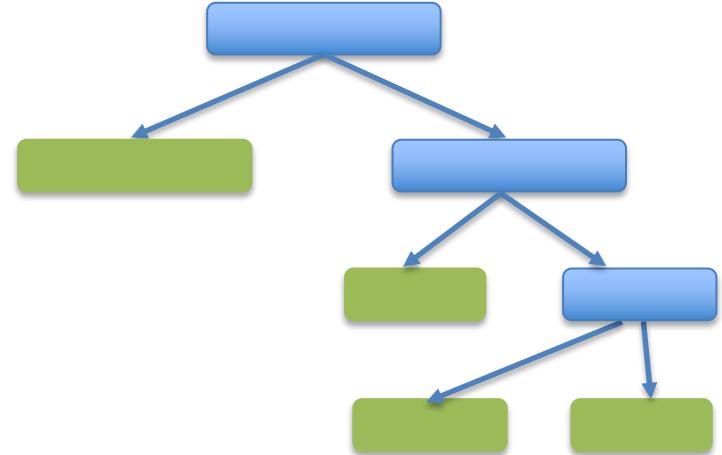
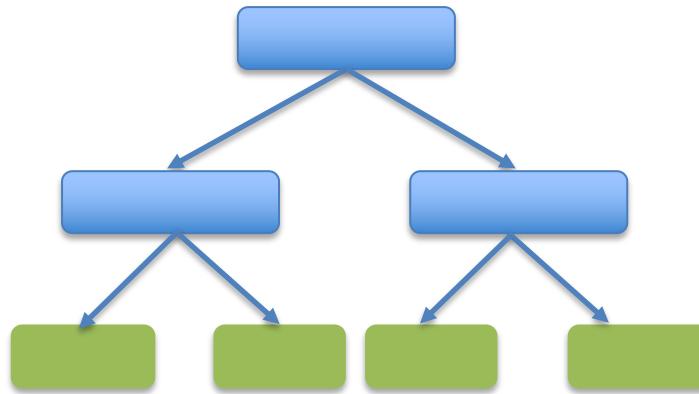
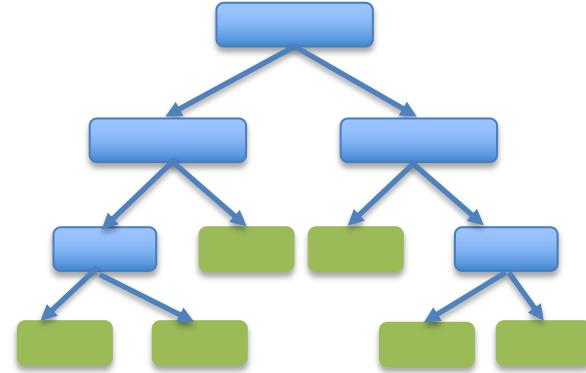
1. Usando um dataset criado com bagging
2. Considerando apenas um subconjunto de variáveis a cada passo



Dor no peito	Boa circulação Sanguínea	Artérias bloqueadas	Peso (kg)	Doença cardíaca
Y	Y	Y	82	Y
N	N	N	57	N
Y	N	Y	76	Y
Y	N	Y	76	Y

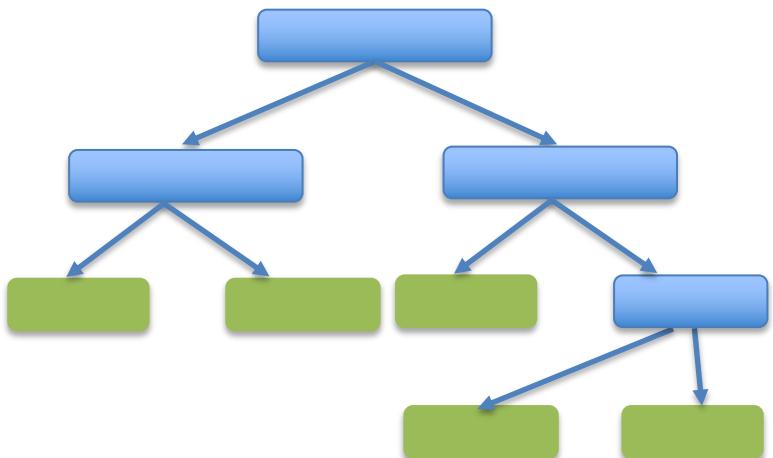
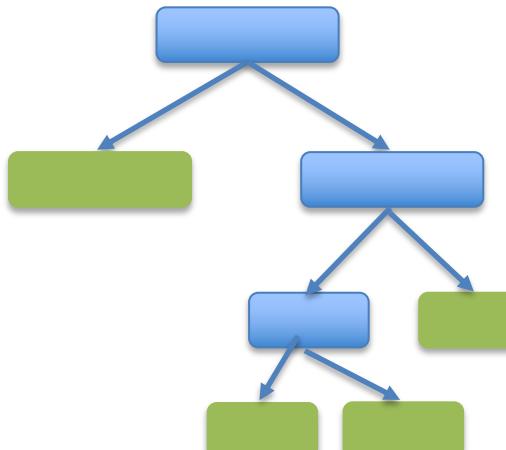
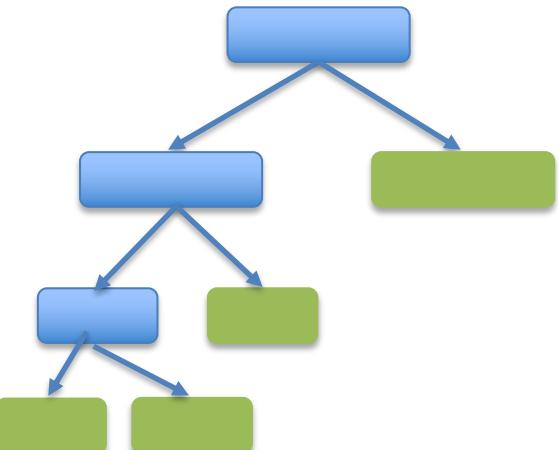
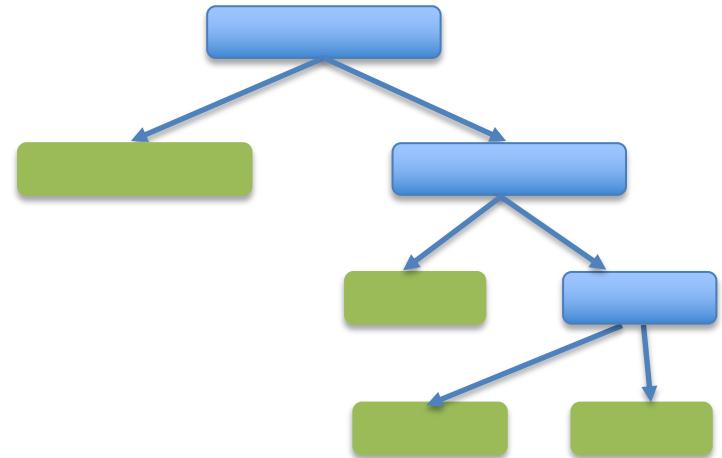
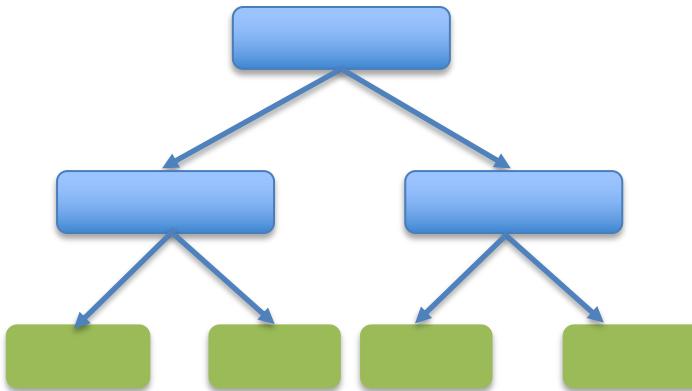
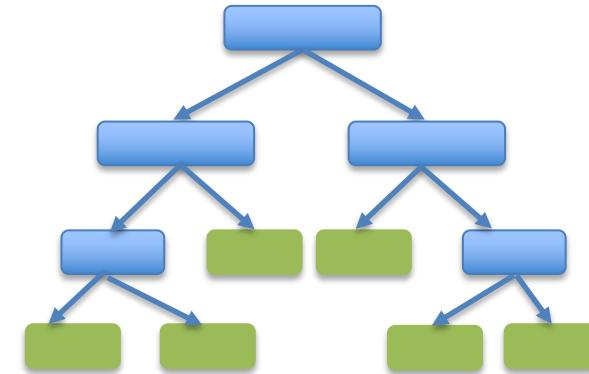
Intuição

Agora, voltamos ao passo 1 e repetimos: criamos um novo dataset com bagging e construímos uma árvore considerando apenas um subconjunto de features a cada passo. Isso nos leva a um conjunto de árvores (ou seja, uma floresta).



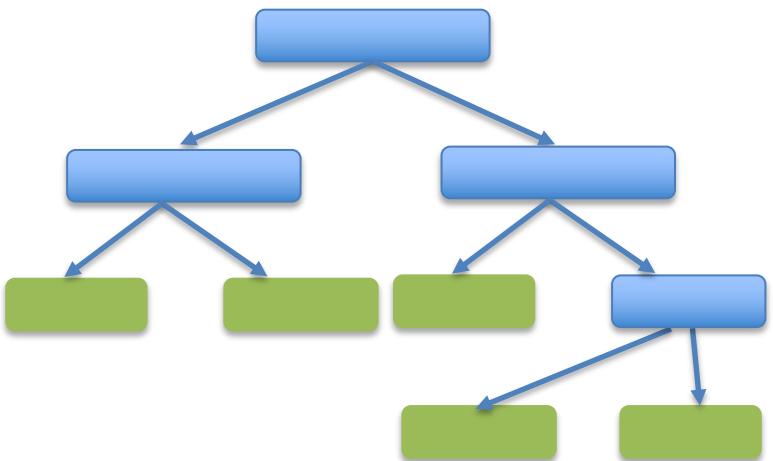
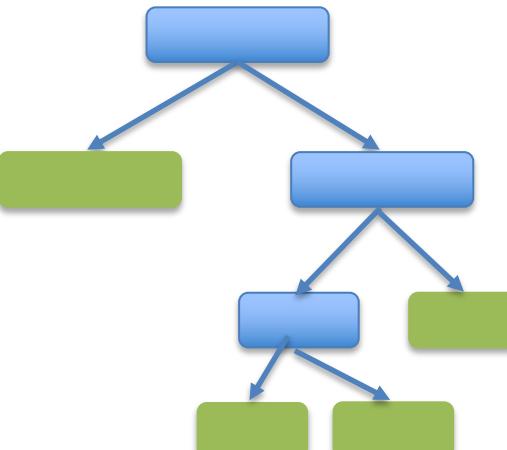
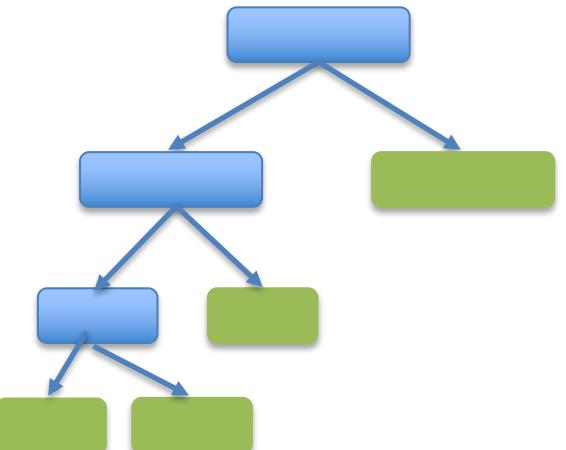
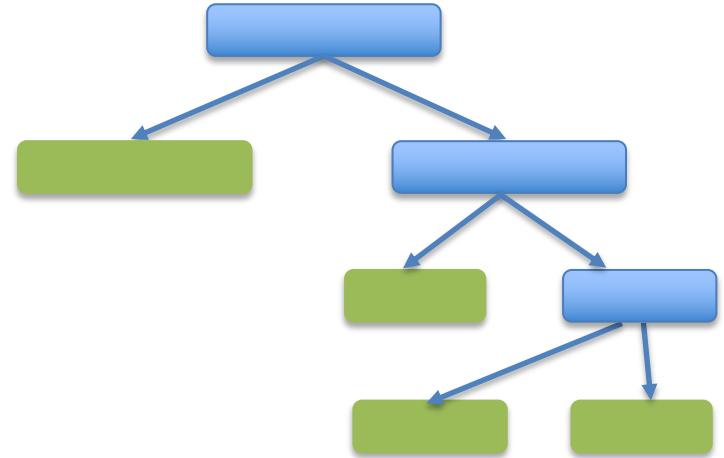
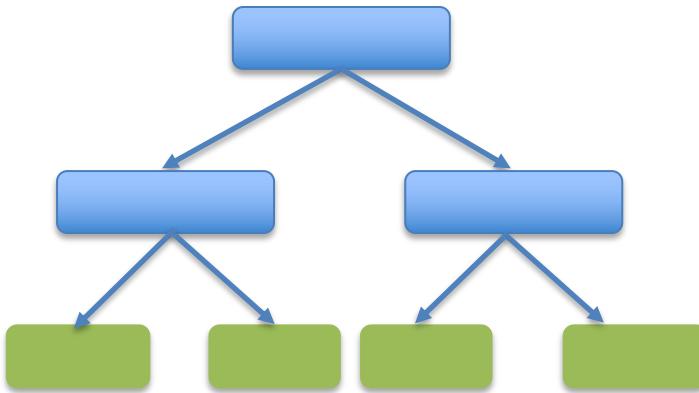
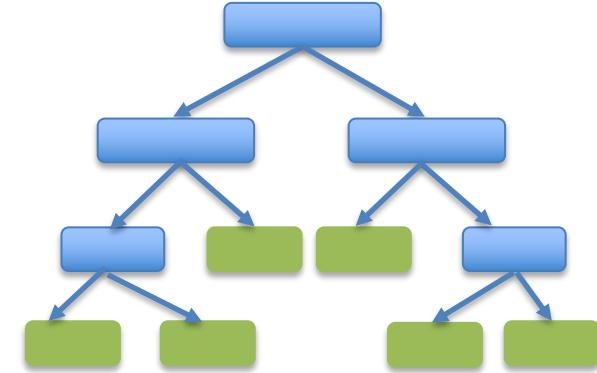
Intuição

Idealmente, fazemos isso milhares de vezes, gerando milhares de árvores. Utilizando amostragem com reposição e considerando apenas um subconjunto de features a cada passo, resulta numa grande variedade de árvores e isto é o que torna Random Forest mais eficaz que as árvores de decisão.



Intuição

Ok, agora que criamos uma Random Forest, como a usamos?



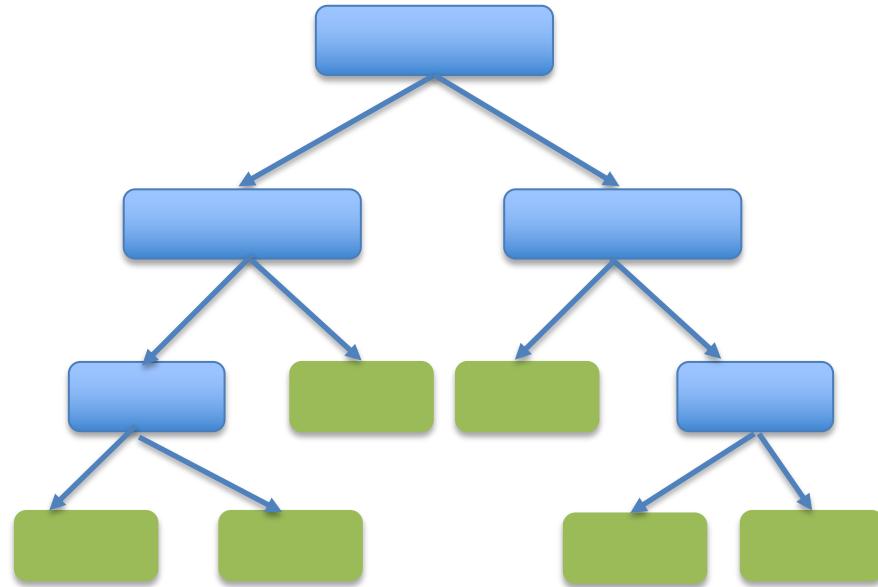
Intuição

Primeiro, recebemos uma nova amostra:

Dor no peito	Boa circulação Sanguínea	Artérias bloqueadas	Peso (kg)	Doença cardíaca
Y	N	N	76	

Intuição

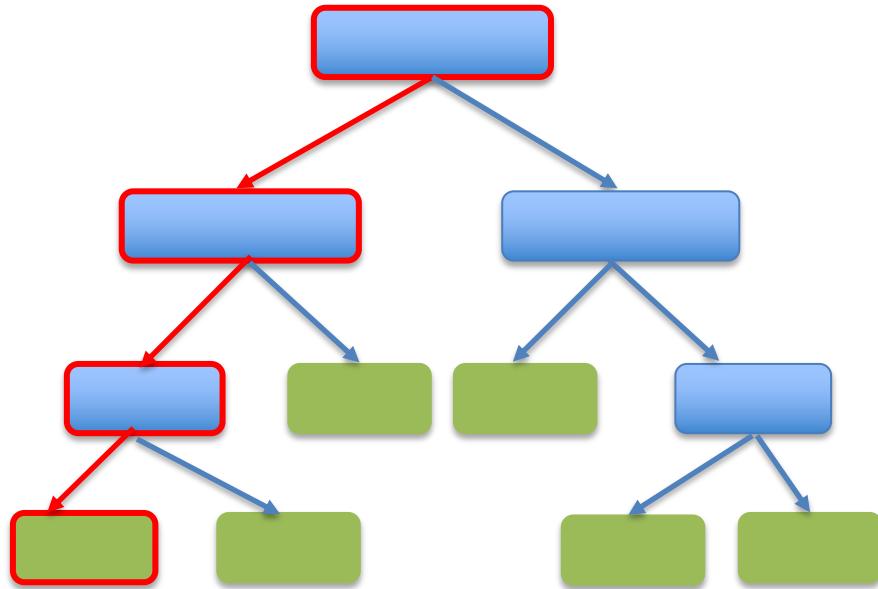
Então, fazemos a predição usando a primeira árvore da nossa floresta:



Dor no peito	Boa circulação Sanguínea	Artérias bloqueadas	Peso (kg)	Doença cardíaca
Y	N	N	76	

Intuição

E a classe predita é "Y".



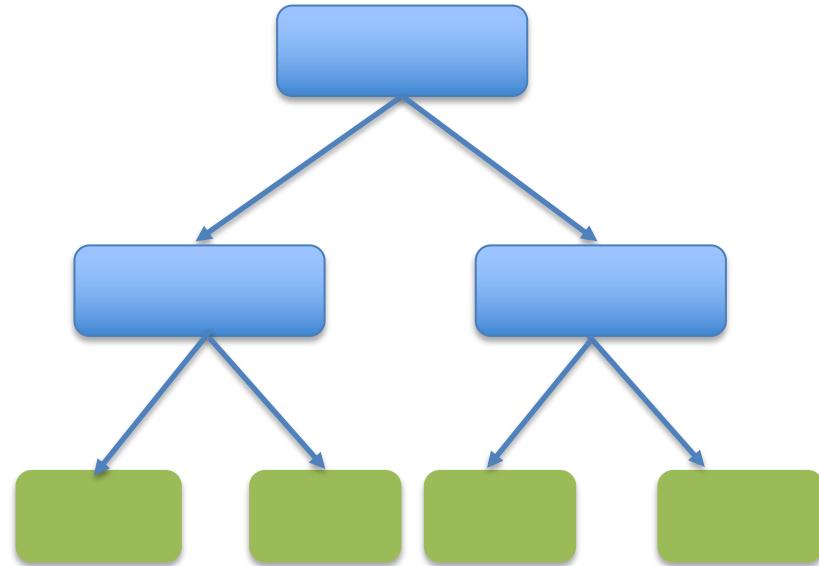
Dor no peito	Boa circulação Sanguínea	Artérias bloqueadas	Peso (kg)	Doença cardíaca
Y	N	N	76	

E guardamos essa informação:

Doença no coração	
Y	N
1	

Intuição

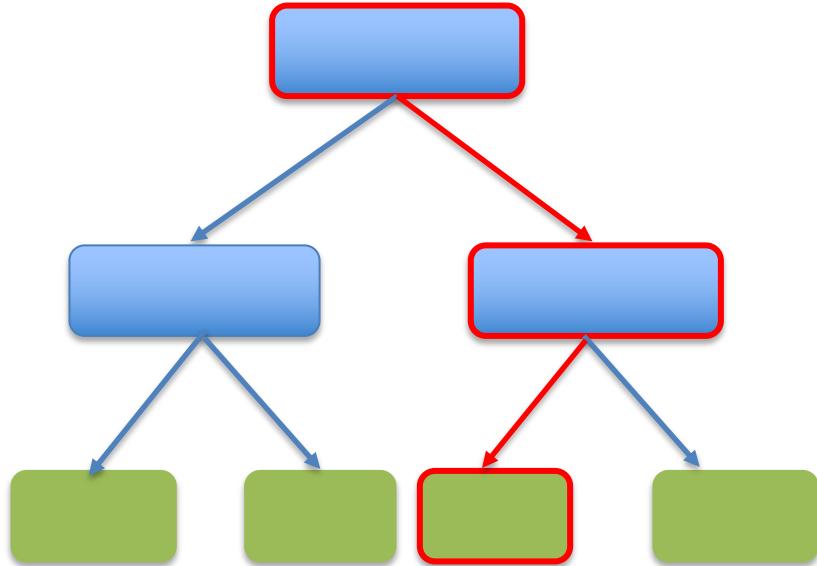
Fazemos a predição, mas agora usando a segunda árvore da nossa floresta:



Dor no peito	Boa circulação Sanguínea	Artérias bloqueadas	Peso (kg)	Doença cardíaca
Y	N	N	76	

Intuição

E a classe predita é "Y":



Dor no peito	Boa circulação Sanguínea	Artérias bloqueadas	Peso (kg)	Doença cardíaca
Y	N	N	76	

E guardamos essa informação:

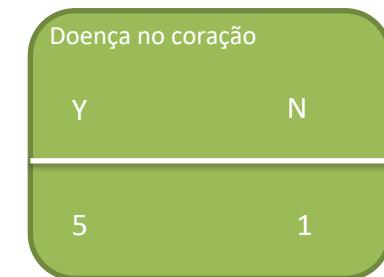
Doença no coração	
Y	N
2	

Intuição

E repetimos o processo para todas as árvores em nossa floresta, obtendo o seguinte placar final:

Dor no peito	Boa circulação Sanguínea	Artérias bloqueadas	Peso (kg)	Doença cardíaca
Y	N	N	76	

Como a classe "Y" recebeu a maioria dos votos, ela é predita para a amostra em questão.

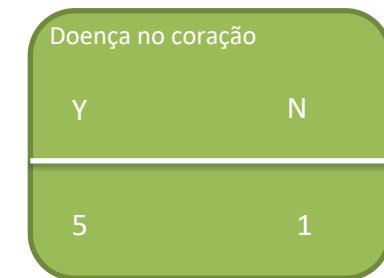


Intuição

E repetimos o processo para todas as árvores em nossa floresta, obtendo o seguinte placar final:

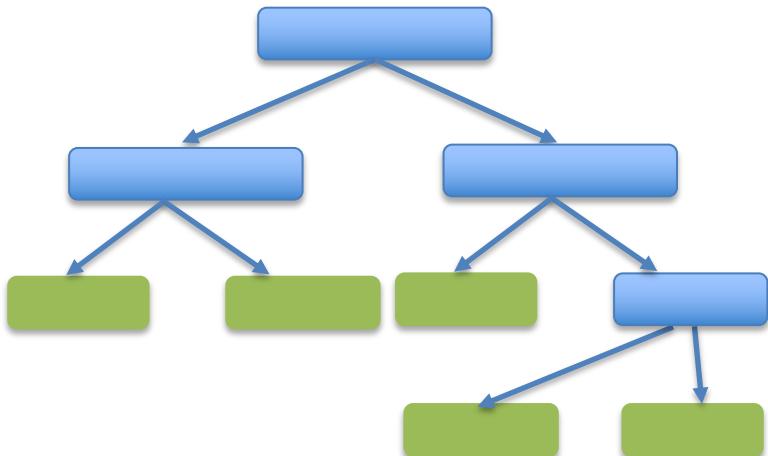
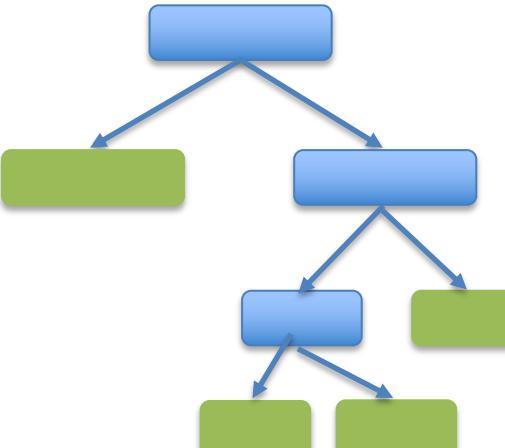
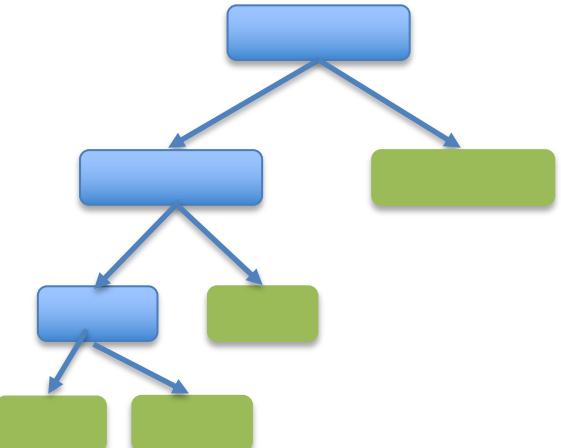
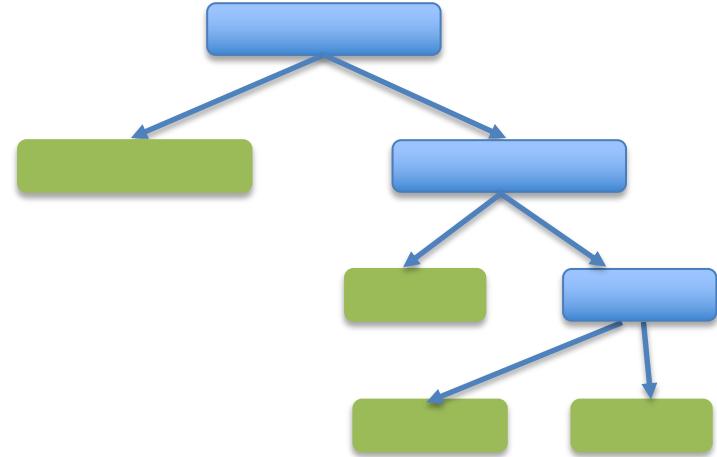
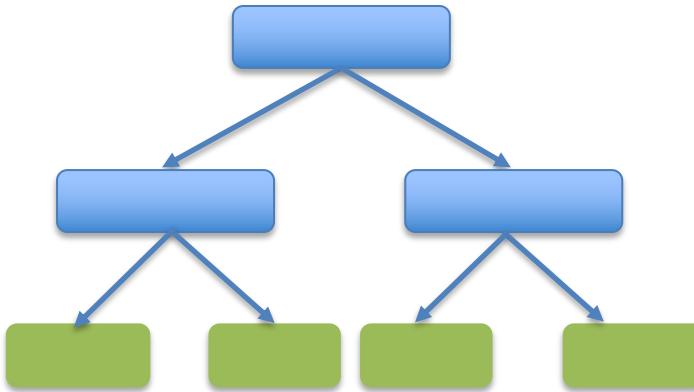
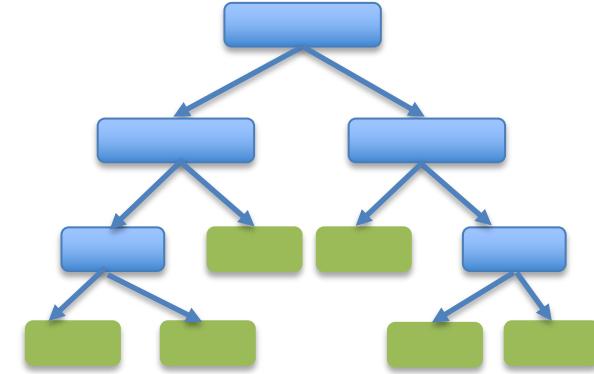
Dor no peito	Boa circulação Sanguínea	Artérias bloqueadas	Peso (kg)	Doença cardíaca
Y	N	N	76	Y

Como a classe "Y" recebeu a maioria dos votos, ela é predita para a amostra em questão.



Avaliando uma Random Forest

Ok, vimos como criar e usar uma Random Forest, mas como sabemos se ele é boa?



Avaliando uma Random Forest

Ao definir o dataset usando bagging, uma amostra não foi incluída, como observamos abaixo:

Dor no peito	Boa circulação Sanguínea	Artérias bloqueadas	Peso (kg)	Doença cardíaca
N	N	N	57	N
Y	Y	Y	82	Y
Y	Y	N	95	N
Y	N	Y	76	Y

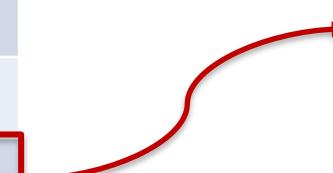
Dor no peito	Boa circulação Sanguínea	Artérias bloqueadas	Peso (kg)	Doença cardíaca
Y	Y	Y	82	Y
N	N	N	57	N
Y	N	Y	76	Y
Y	N	Y	76	Y

Tipicamente, por volta de 1/3 do dataset original não fará parte do dataset criado com bagging.

Avaliando uma Random Forest

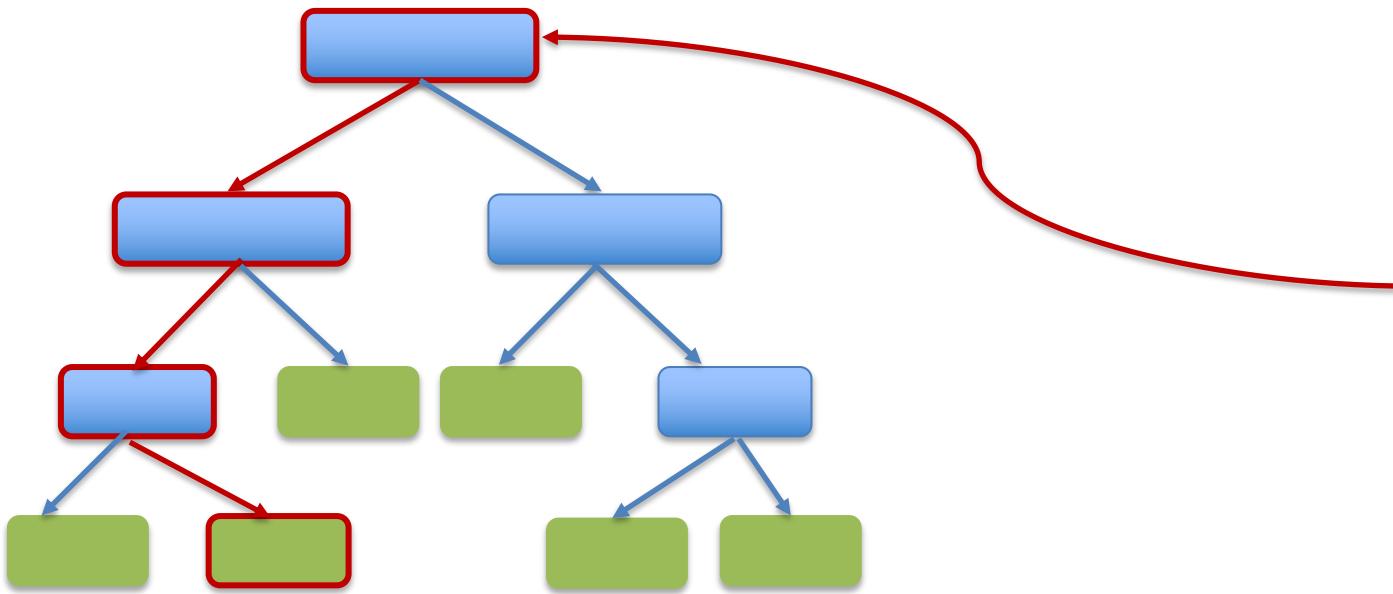
Essa amostra que não entrou no dataset criado com bagging compõe um dataset chamado "out-of-bag"

Dataset original					Dataset Out-of-bag				
Dor no peito	Boa circulação Sanguínea	Artérias bloqueadas	Peso (kg)	Doença cardíaca	Dor no peito	Boa circulação Sanguínea	Artérias bloqueadas	Peso (kg)	Doença cardíaca
N	N	N	57	N	Y	Y	N	95	N
Y	Y	Y	82	Y	Y	Y	N	95	N
Y	Y	N	95	N	Y	Y	N	76	Y
Y	N	Y	76	Y					



Avaliando uma Random Forest

Visto que esse dataset não foi usado para construir essa árvore, podemos usá-la para predizer essa amostra e verificar se ela irá predizer "N" como classe:



Dataset Out-of-bag

Dor no peito	Boa circulação Sanguínea	Artérias bloqueadas	Peso (kg)	Doença cardíaca
Y	Y	N	95	N

Avaliando uma Random Forest

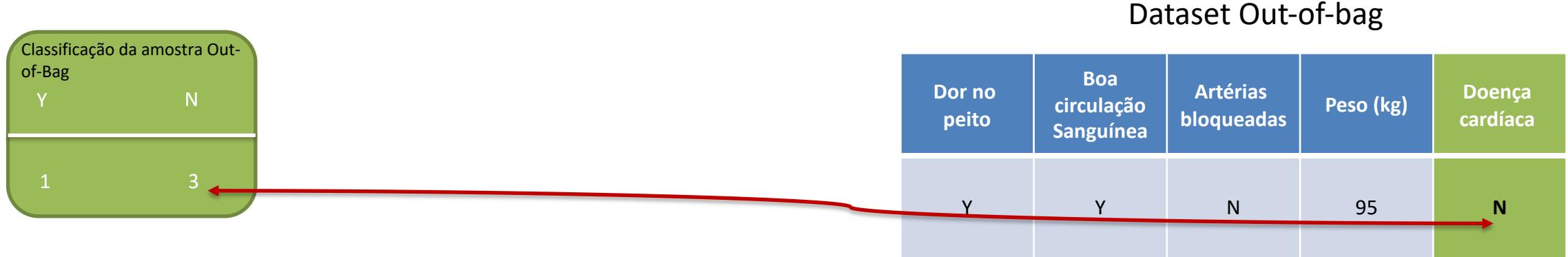
E repetimos o processo para todas as árvores, computando a predição de cada uma delas, como segue:

Classificação da amostra Out-of-Bag	
Y	N
1	3

Dataset Out-of-bag				
Dor no peito	Boa circulação Sanguínea	Artérias bloqueadas	Peso (kg)	Doença cardíaca
Y	Y	N	95	N

Avaliando uma Random Forest

Neste caso, a amostra Out-of-Bag foi corretamente classificada pela Random Forest



Ao repetir o processo para todas as amostras do dataset Out-of-bag, podemos mensurar quão bom está nossa Random Forest observando a proporção delas que foram corretamente classificadas.

Escolhendo o Número de Features

Até agora, nós aprendemos a:

1. Construir uma Random Forest
2. Usar uma Random Forest
3. Estimar a acurácia de uma Random Forest

Vamos agora entender como podemos escolher o número de features a ser considerado ao construir uma árvore do dataset criado com bagging

Escolhendo o Número de Features

Ao construir a primeira árvore usando o dataset criado com bagging, usamos apenas duas features:

Dor no peito	Boa circulação Sanguínea	Artérias bloqueadas	Peso (kg)	Doença cardíaca
Y	Y	Y	82	Y
N	N	N	57	N
Y	N	Y	76	Y
Y	N	Y	76	Y

Escolhendo o Número de Features

Ao construir a primeira árvore usando o dataset criado com bagging, usamos apenas duas features:

Dor no peito	Boa circulação Sanguínea	Artérias bloqueadas	Peso (kg)	Doença cardíaca
Y	Y	Y	82	Y
N	N	N	57	N
Y	N	Y	76	Y
Y	N	Y	76	Y

A ideia é que agora podemos o dataset Out-of-Bag e comparar o resultado de uma árvore criada com duas features com outra usando três (ou mais) features.

E podemos testar várias configurações e escolher a que retorna o melhor resultado

Escolhendo o Número de Features

Em outras palavras:

1. Construímos uma Random Forest
2. Estimamos a acurácia da Random Forest
3. Alteramos o número de features usado a cada passo e voltamos ao ponto 1.

Por padrão, é usado a raiz quadrada do número de features disponíveis, mas há outras opções. (hiperparâmetro `max_features` na Sklearn)

Lidando com valores faltantes

Random Forest considera dois tipos de valores faltantes:

1. No dataset original usado pra criar a Random Forest
2. Valores faltantes numa nova amostra que queremos classificar

1

Dor no peito	Boa circulação Sanguínea	Artérias bloqueadas	Peso (kg)	Doença cardíaca
N	N	N	57	N
Y	Y	Y	82	Y
Y	Y	N	95	N
Y	Y	?	?	N

2

Dor no peito	Boa circulação Sanguínea	Artérias bloqueadas	Peso (kg)	Doença cardíaca
N	N	?	76	

Vamos começar pelo primeiro.

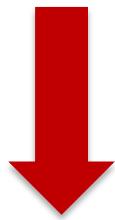
Lidando com valores faltantes

A ideia geral para lidar com valores faltantes nesse contexto é fazer palpite inicial (que pode ser ruim) e ir refinando esse palpite até que ele (espera-se) seja bom o suficiente.

Dor no peito	Boa circulação Sanguínea	Artérias bloqueadas	Peso (kg)	Doença cardíaca
N	N	N	57	N
Y	Y	Y	82	Y
Y	Y	N	95	N
Y	Y	?	?	N

Lidando com valores faltantes

Vemos que a classe da amostra 4 é N. Assim, um palpite inicial seria pegar o valor mais comum de Artérias bloqueadas para quando a classe é N.



Dor no peito	Boa circulação Sanguínea	Artérias bloqueadas	Peso (kg)	Doença cardíaca
N	N	N	57	N
Y	Y	Y	82	Y
Y	Y	N	95	N
Y	Y	?	?	N

Lidando com valores faltantes

Assim, N é nosso palpite inicial para preencher o valor faltante de Artérias Bloqueadas.



Dor no peito	Boa circulação Sanguínea	Artérias bloqueadas	Peso (kg)	Doença cardíaca
N	N	N	57	N
Y	Y	Y	82	Y
Y	Y	N	95	N
Y	Y	N	?	N

Lidando com valores faltantes

Como Peso é numérico, nosso palpite inicial será a mediana dos pacientes que não possuem Doença cardíaca



Dor no peito	Boa circulação Sanguínea	Artérias bloqueadas	Peso (kg)	Doença cardíaca
N	N	N	57	N
Y	Y	Y	82	Y
Y	Y	N	95	N
Y	Y	N	?	N

Lidando com valores faltantes

Como Peso é numérico, nosso palpite inicial será a mediana dos pacientes que não possuem Doença cardíaca



Dor no peito	Boa circulação Sanguínea	Artérias bloqueadas	Peso (kg)	Doença cardíaca
N	N	N	57	N
Y	Y	Y	82	Y
Y	Y	N	95	N
Y	Y	N	76	N

Lidando com valores faltantes

Com o dataset com valores faltantes preenchidos, precisamos refinar agora os palpites. Fazemos isso ao, primeiramente, determinar quais amostras são similares àquela com valor faltante.

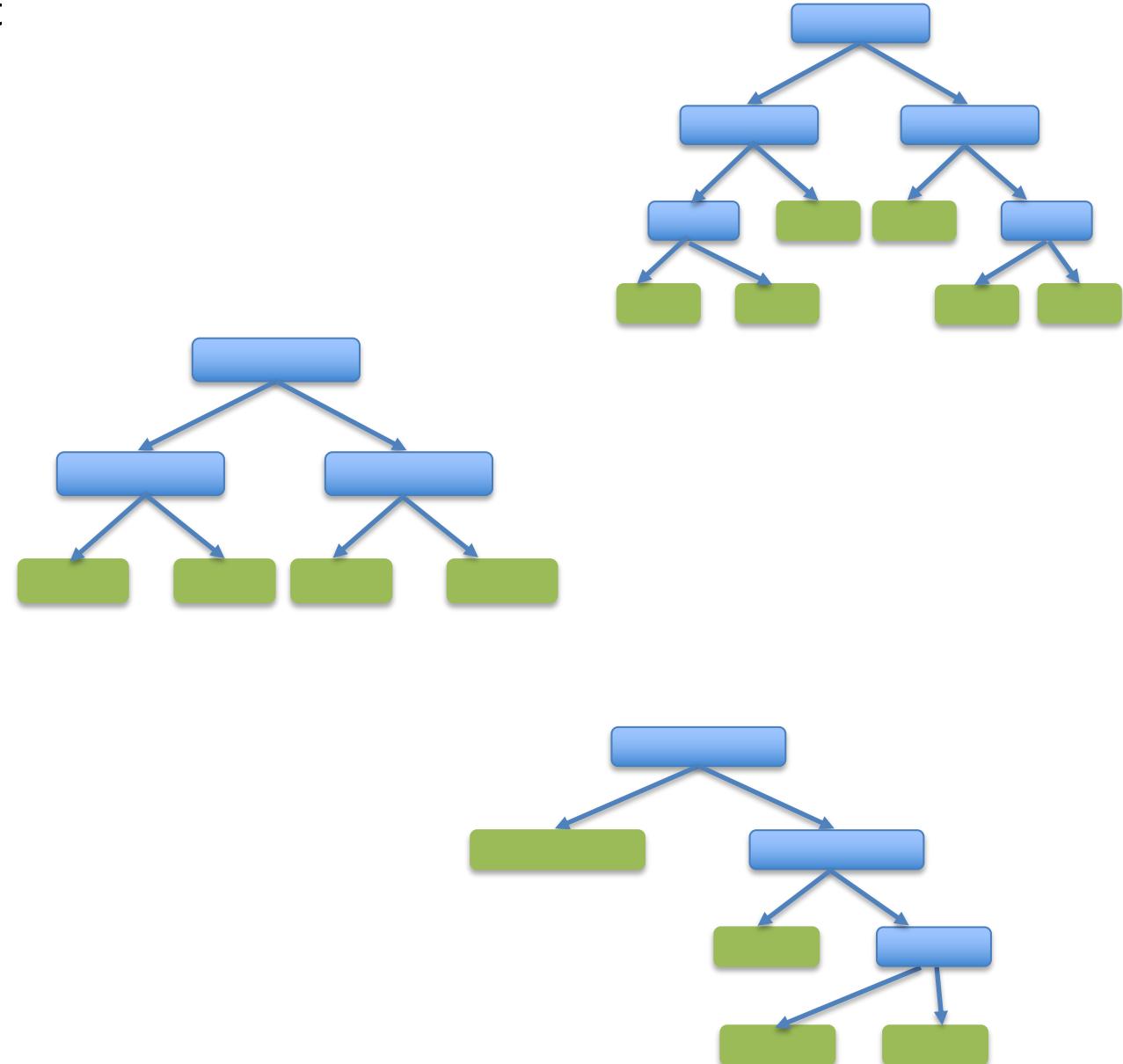
Mas como determinamos similaridade? Vamos entender.

Dor no peito	Boa circulação Sanguínea	Artérias bloqueadas	Peso (kg)	Doença cardíaca
N	N	N	57	N
Y	Y	Y	82	Y
Y	Y	N	95	N
Y	Y	N	76	N

Lidando com valores faltantes

O primeiro passo é construir uma Random Forest

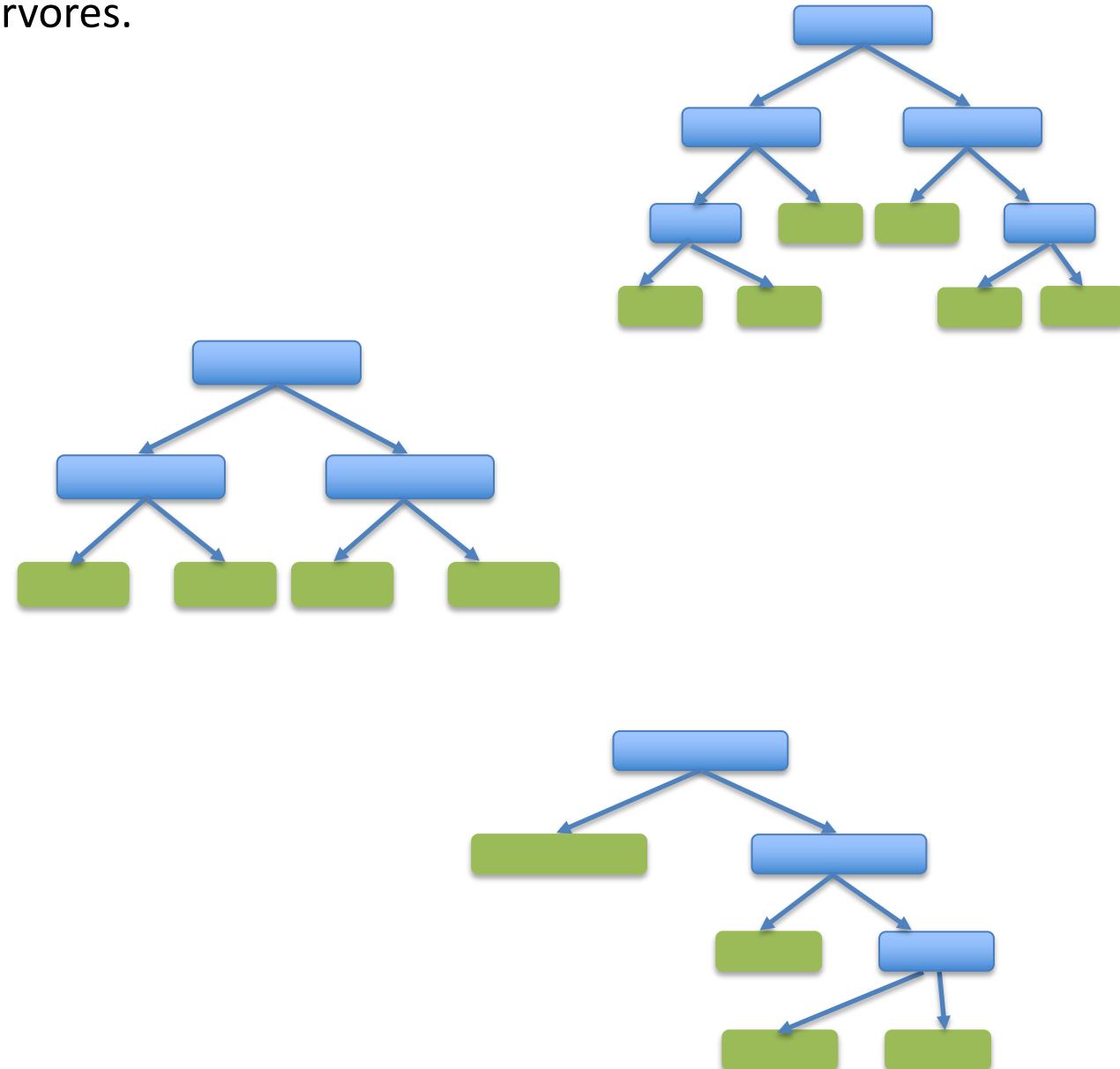
Dor no peito	Boa circulação Sanguínea	Artérias bloqueadas	Peso (kg)	Doença cardíaca
N	N	N	57	N
Y	Y	Y	82	Y
Y	Y	N	95	N
Y	Y	N	76	N



Lidando com valores faltantes

O segundo passo é rodar o dataset em todas as árvores.

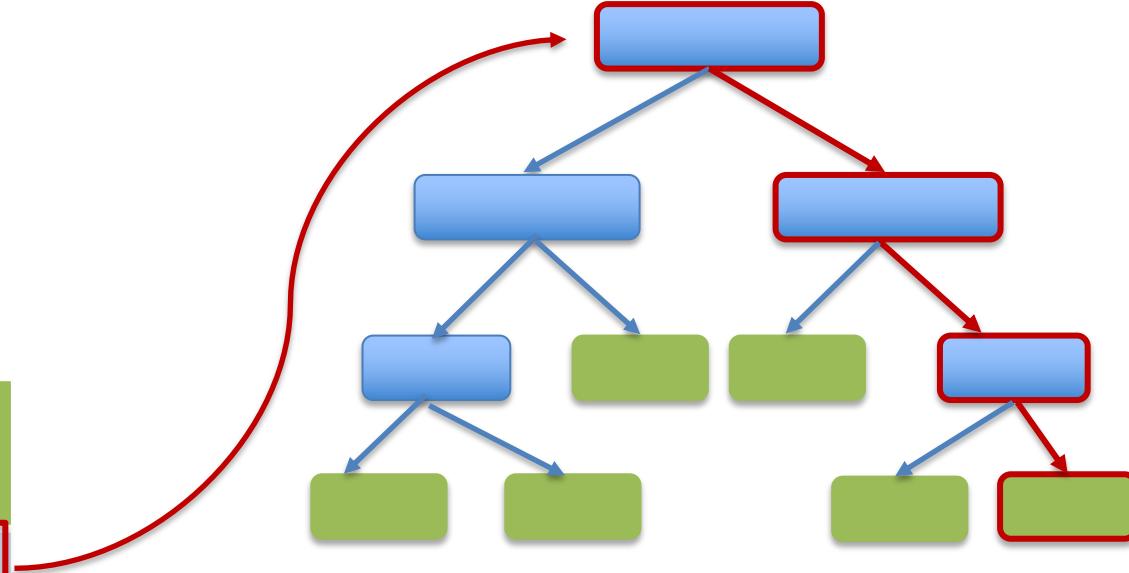
Dor no peito	Boa circulação Sanguínea	Artérias bloqueadas	Peso (kg)	Doença cardíaca
N	N	N	57	N
Y	Y	Y	82	Y
Y	Y	N	95	N
Y	Y	N	76	N



Lidando com valores faltantes

Vamos começar pela primeira e executar para cada amostra:

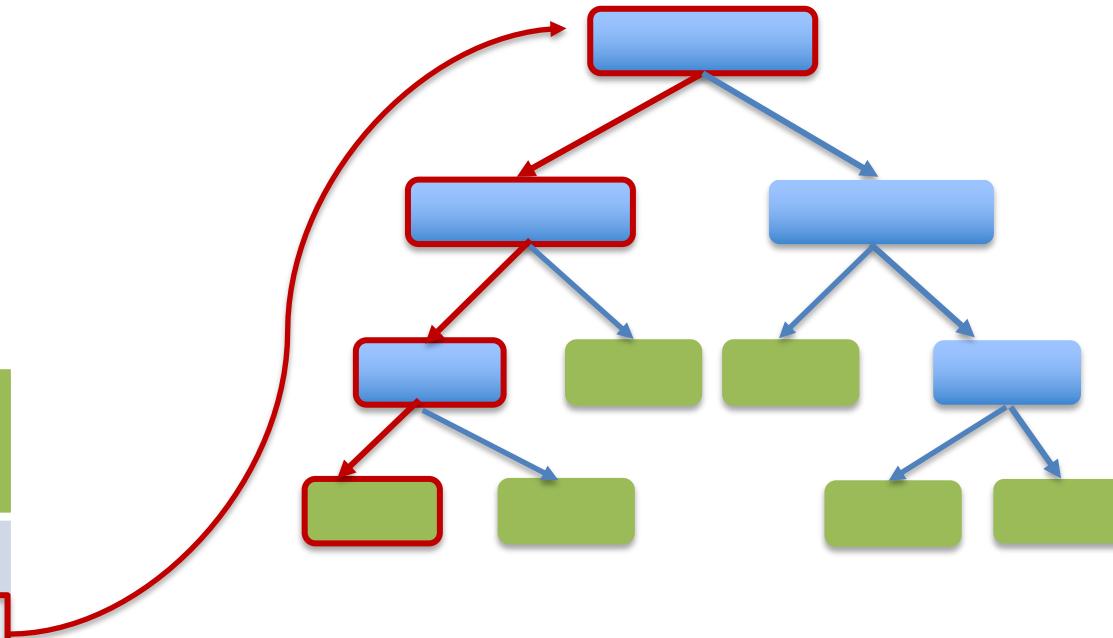
Dor no peito	Boa circulação Sanguínea	Artérias bloqueadas	Peso (kg)	Doença cardíaca
N	N	N	57	N
Y	Y	Y	82	Y
Y	Y	N	95	N
Y	Y	N	76	N



Lidando com valores faltantes

Vamos começar pela primeira e executar para cada amostra:

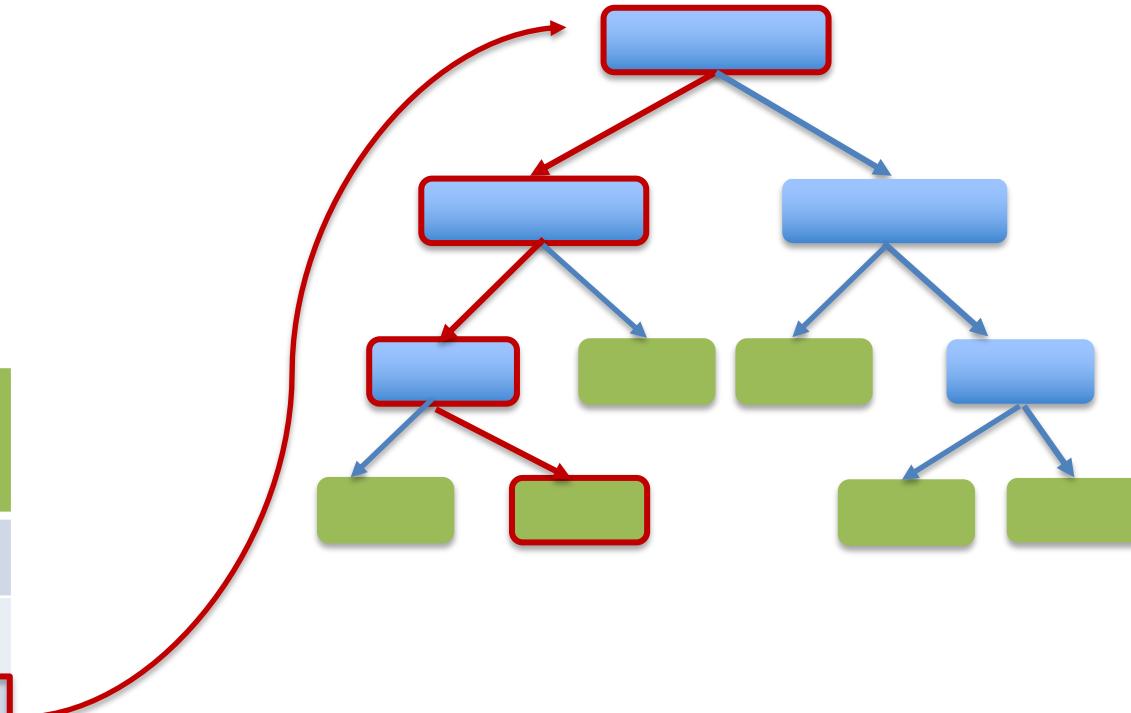
Dor no peito	Boa circulação Sanguínea	Artérias bloqueadas	Peso (kg)	Doença cardíaca
N	N	N	57	N
Y	Y	Y	82	Y
Y	Y	N	95	N
Y	Y	N	76	N



Lidando com valores faltantes

Vamos começar pela primeira e executar para cada amostra:

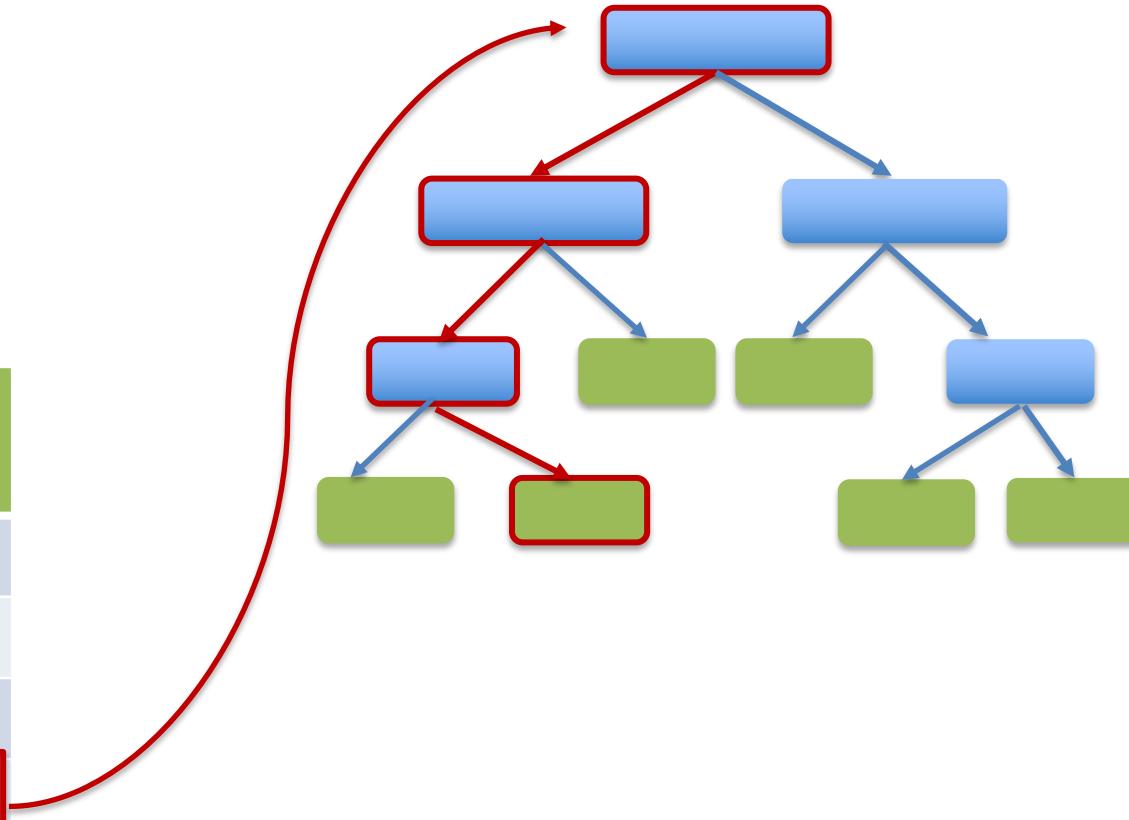
Dor no peito	Boa circulação Sanguínea	Artérias bloqueadas	Peso (kg)	Doença cardíaca
N	N	N	57	N
Y	Y	Y	82	Y
Y	Y	N	95	N
Y	Y	N	76	N



Lidando com valores faltantes

Vamos começar pela primeira e executar para cada amostra:

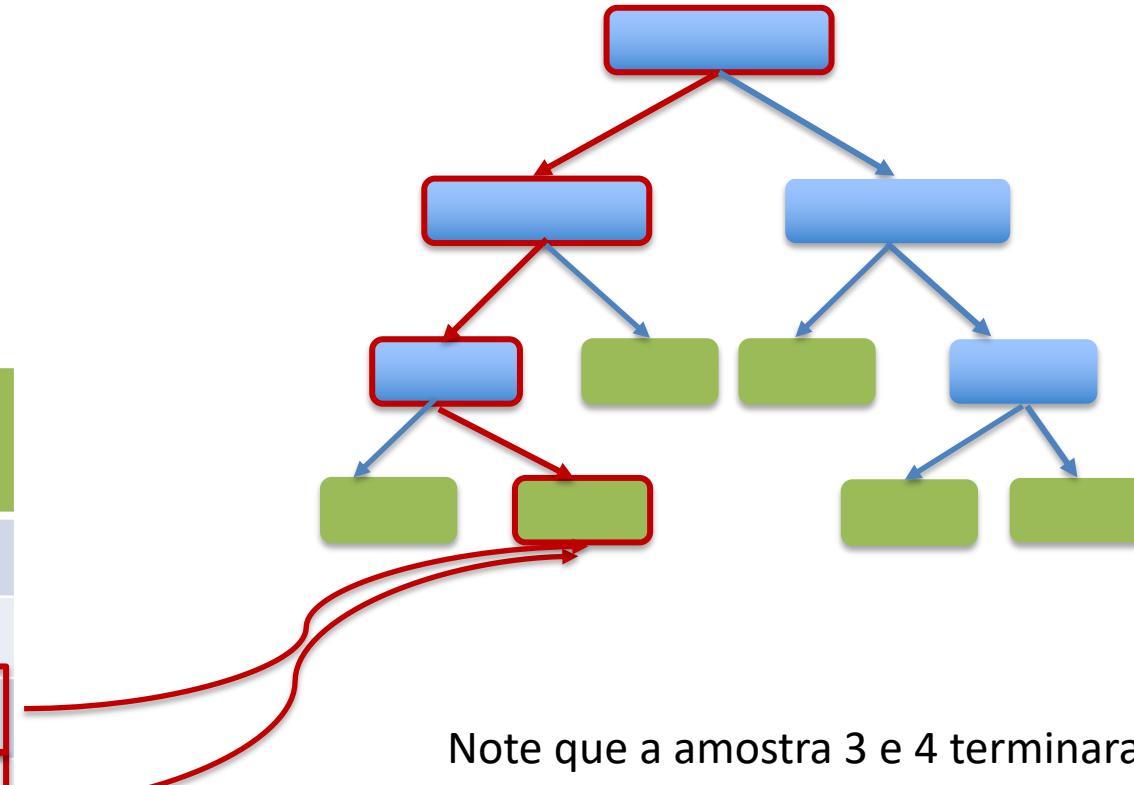
Dor no peito	Boa circulação Sanguínea	Artérias bloqueadas	Peso (kg)	Doença cardíaca
N	N	N	57	N
Y	Y	Y	82	Y
Y	Y	N	95	N
Y	Y	N	76	N



Lidando com valores faltantes

Vamos começar pela primeira e executar para cada amostra:

Dor no peito	Boa circulação Sanguínea	Artérias bloqueadas	Peso (kg)	Doença cardíaca
N	N	N	57	N
Y	Y	Y	82	Y
Y	Y	N	95	N
Y	Y	N	76	N



Note que a amostra 3 e 4 terminaram no mesmo nó folha. Isto significa que elas são similares. Em outras palavras, essa é a maneira que medimos similaridade.

Lidando com valores faltantes

Podemos guardar informações de similaridade entre amostras usando uma Matriz de Proximidade, que possui shape definido pela quantidade de amostras. Quando duas delas acabam no mesmo nó folha, a intersecção é incrementada em uma unidade.

Dor no peito	Boa circulação Sanguínea	Artérias bloqueadas	Peso (kg)	Doença cardíaca
N	N	N	57	N
Y	Y	Y	82	Y
Y	Y	N	95	N
Y	Y	N	76	N

	1	2	3	4
1				
2				
3				1
4			1	

Como nenhum outro par de amostras caiu no mesmo nó folha, essa é a Matriz de Proximidade depois de executar a primeira árvore em todas as amostras.

Lidando com valores faltantes

Repetimos o processo para cada árvore presente na Random Forest, incrementando o valor da célula quando duas ou mais amostras caem no mesmo nó folha. Ao final da execução, teremos a seguinte Matriz de Proximidade

Dor no peito	Boa circulação Sanguínea	Artérias bloqueadas	Peso (kg)	Doença cardíaca
N	N	N	57	N
Y	Y	Y	82	Y
Y	Y	N	95	N
Y	Y	N	76	N

	1	2	3	4
1		2	1	1
2	2		1	1
3	1	1		8
4	1	1	8	

Lidando com valores faltantes

Agora, dividimos cada valor pelo número total de árvores. Neste caso, considerei 10 árvores.

Dor no peito	Boa circulação Sanguínea	Artérias bloqueadas	Peso (kg)	Doença cardíaca
N	N	N	57	N
Y	Y	Y	82	Y
Y	Y	N	95	N
Y	Y	?	?	N

	1	2	3	4
1		0.2	0.1	0.1
2	0.2		0.1	0.1
3	0.1	0.1		0.8
4	0.1	0.1	0.8	

E usamos os valores de proximidade para a amostra 4 para criar um palpite melhor para os dados faltantes.

Lidando com valores faltantes

Para Artérias Bloqueadas, calculamos a frequência ponderada de "Y" e "N" usando os valores de proximidade como pesos.

Dor no peito	Boa circulação Sanguínea	Artérias bloqueadas	Peso (kg)	Doença cardíaca
N	N	N	57	N
Y	Y	Y	82	Y
Y	Y	N	95	N
Y	Y	?	?	N

	1	2	3	4
1		0.2	0.1	0.1
2	0.2		0.1	0.1
3	0.1	0.1		0.8
4	0.1	0.1	0.8	

Frequencia ponderada para "Y" = frequencia de "Y" na feature Artérias bloqueadas (1/3) * peso "Y"

O peso "Y" é dado pela razao entre os valores de proximidade "Y" e todos os valores de proximidade
 $= 0.1/(0.1+0.1+0.8) = 0.1/1 = 0.1$

Assim, a frequencia ponderada para "Y" é $1/3*0.1 = 0.03$

Lidando com valores faltantes

De maneira similar, a frequencia ponderada para "N" é dada por $2/3 * \text{peso } "N"$.

O peso "N" é dado por $(0.1+0.8)/(0.1+0.1+0.8) = 0.9/1 = 0.9$

Dor no peito	Boa circulação Sanguínea	Artérias bloqueadas	Peso (kg)	Doença cardíaca
N	N	N	57	N
Y	Y	Y	82	Y
Y	Y	N	95	N
Y	Y	?	?	N

	1	2	3	4
1		0.2	0.1	0.1
2	0.2		0.1	0.1
3	0.1	0.1		0.8
4	0.1	0.1	0.8	

Assim, a frequencia ponderada para "N" é $2/3 * 0.9 = 0.6$

Lidando com valores faltantes

Como "N" tem um valor ponderado maior que "Y", "N" será o valor a ser preenchido para a coluna Artérias Bloqueadas.

Dor no peito	Boa circulação Sanguínea	Artérias bloqueadas	Peso (kg)	Doença cardíaca
N	N	N	57	N
Y	Y	Y	82	Y
Y	Y	N	95	N
Y	Y	N	?	N

	1	2	3	4
1		0.2	0.1	0.1
2	0.2		0.1	0.1
3	0.1	0.1		0.8
4	0.1	0.1	0.8	

Lidando com valores faltantes

Para peso, usamos os valores de proximidade para calcular a média ponderada. Para isso, multiplicamos cada valor de Peso por seu correspondente valor de proximidade, como abaixo:

Dor no peito	Boa circulação Sanguínea	Artérias bloqueadas	Peso (kg)	Doença cardíaca
N	N	N	57	N
Y	Y	Y	82	Y
Y	Y	N	95	N
Y	Y	N	90	N

	1	2	3	4
1		0.2	0.1	0.1
2	0.2		0.1	0.1
3	0.1	0.1		0.8
4	0.1	0.1	0.8	

$$\text{Peso} = (57 * 0.1 + 82 * 0.1 + 95 * 0.8) = 90 \text{ e preenchemos com esse valor.}$$

Esse processo todo é executado novamente. Ou seja, construímos uma Random Forest, rodamos os dados por todas as árvores, recalculamos os valores de proximidade e recalculamos os valores faltantes.

Isto é feito até que valores faltantes converjam, isto é, não alterem quando forem recalculados.

Lidando com valores faltantes

Agora é hora de lidar com o segundo tipo de valor faltante

1. No dataset original usado pra criar a Random Forest
2. Valores faltantes numa nova amostra que queremos classificar

2

Dor no peito	Boa circulação Sanguínea	Artérias bloqueadas	Peso (kg)	Doença cardíaca
N	N	?	76	

Lidando com valores faltantes

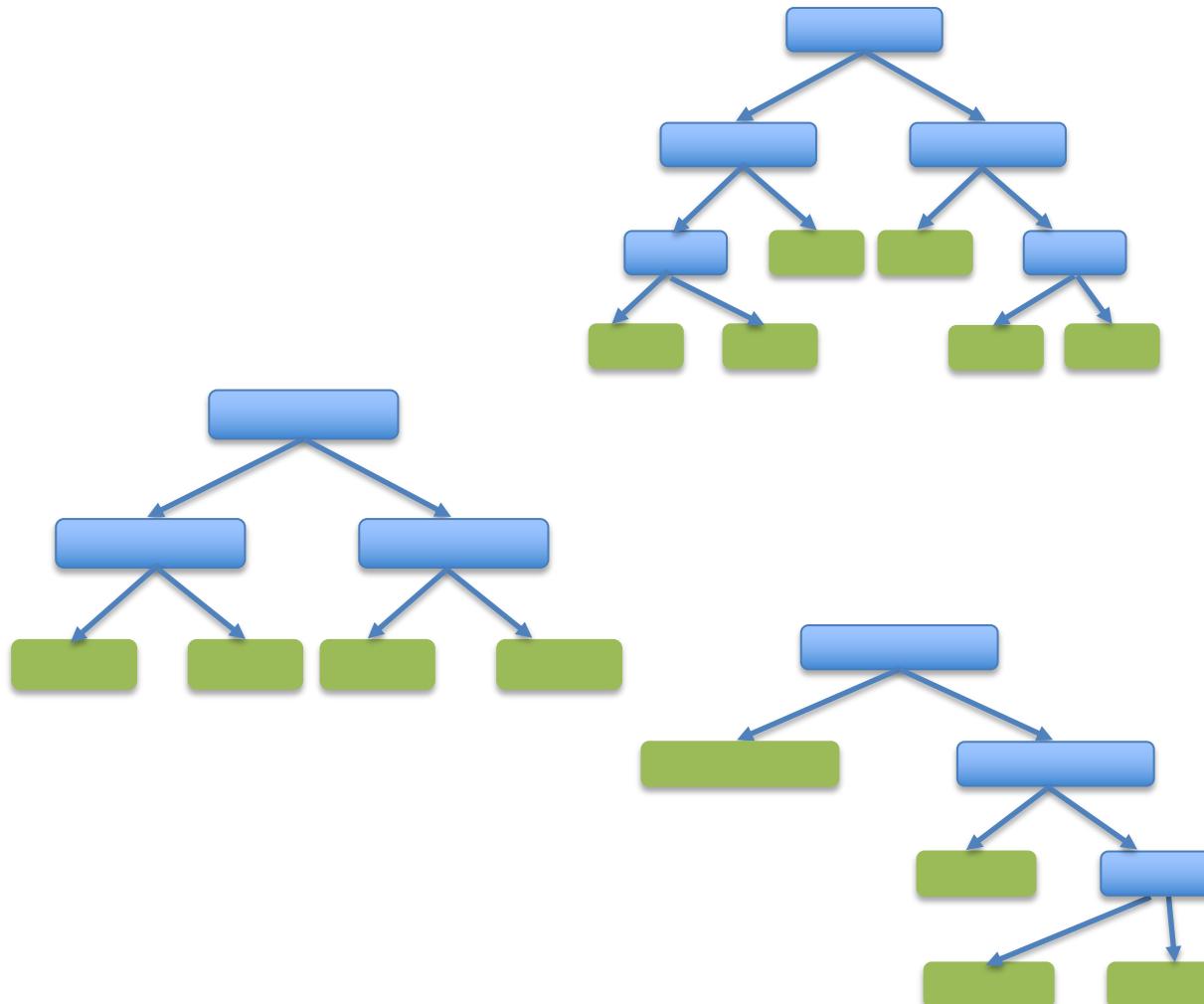
A primeira coisa que precisamos fazer é criar duas cópias desse dado, uma em que a classe seja "Y" e outra que a classe seja "N".

Dor no peito	Boa circulação Sanguínea	Artérias bloqueadas	Peso (kg)	Doença cardíaca
N	N	?	76	Y

Dor no peito	Boa circulação Sanguínea	Artérias bloqueadas	Peso (kg)	Doença cardíaca
N	N	?	76	N

Lidando com valores faltantes

Em seguida, usamos o método iterativo que acabamos de ver pra preencher valores faltantes no dataset, ou seja, rodamos os dois dados na nossa Random Forest e verificamos quais dos dois dados são corretamente classificados a maioria das vezes.



Dor no peito	Boa circulação Sanguínea	Artérias bloqueadas	Peso (kg)	Doença cardíaca
N	N	?	76	Y

Dor no peito	Boa circulação Sanguínea	Artérias bloqueadas	Peso (kg)	Doença cardíaca
N	N	?	76	N

Obrigado!

profdheny.fernandes@fiap.com.br

 /dhenyfernandes

FIAP MBA⁺

Copyright © 2022 | Professor Dheny R. Fernandes

Todos os direitos reservados. Reprodução ou divulgação total ou parcial deste documento, é expressamente proibido sem consentimento formal, por escrito, do professor/autor.

F | A P