FIAP

Algoritmos de Classificação

Dheny R. Fernandes

Agenda

1. **Decision Trees**
    1. Como construir uma árvore
    2. Gini
    3. Exemplo
2. **Formas de Validação**
    1. Holdout
    2. K-fold
3. **KNN**
    1. Intuição
    2. Características
    3. Medida de Distância
    4. Exemplo
    5. Sensibilidade a escala
    6. Implementação
4. **Métricas de Avaliação**
    1. Matriz de confusão
    2. Acurácia
    3. Precision
    4. Recall
    5. F1-Score

Decision Trees

Decision Trees

Classification and Regression Trees, ou CART, é um termo usado para se referir aos algoritmos de Árvore de Decisão que podem ser usados para modelar problemas tanto de classificação quanto de regressão.

O Algoritmo CART é o fundamento para importantes algoritmos como *Bagged Decision Trees, Boosted Decision Trees* e *Random Forest*.

Aqui, vamos entender de modo geral as Decision Trees, visto que já estudamos profundamente as Regression Trees

Decision Trees

Para começar, vamos entender como construir uma árvore de decisão. Para tanto, suponha que eu possua os seguintes dados:

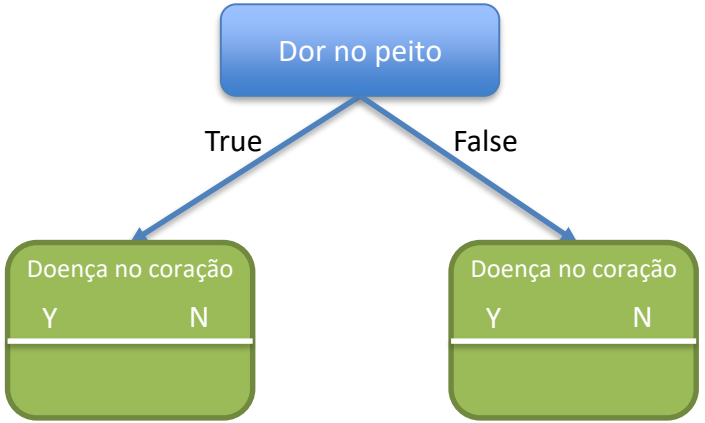
Dor no peito	Boa circulação Sanguínea	Artérias bloqueadas	Doença cardíaca
N	N	N	N
Y	Y	Y	Y
Y	Y	N	N
Y	N	?	Y
...	...	...	...

Assim como nas regression trees, a primeira coisa a ser feita é avaliar cada feature como uma possível candidata a ser o nó raiz.

Decision Trees

Vamos verificar como Dor no peito prediz doença cardíaca:

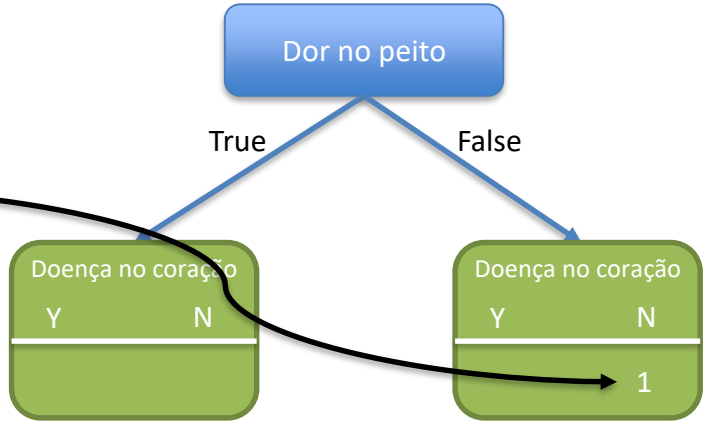
Dor no peito	Boa circulação Sanguínea	Artérias bloqueadas	Doença cardíaca
N	N	N	N
Y	Y	Y	Y
Y	Y	N	N
Y	N	?	Y
...	...	...	...



Decision Trees

Agora contamos os valores para cada combinação:

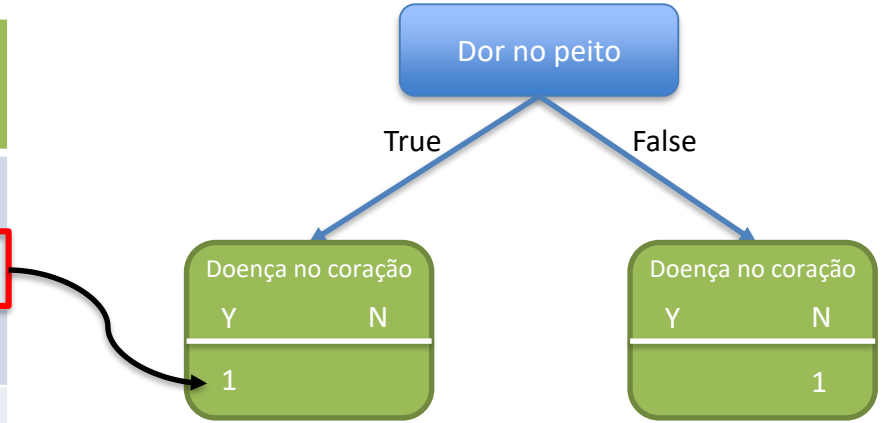
Dor no peito	Boa circulação Sanguínea	Artérias bloqueadas	Doença cardíaca
N	N	N	N
Y	Y	Y	Y
Y	Y	N	N
Y	N	?	Y
...	...	...	...



Decision Trees

Agora contamos os valores para cada combinação:

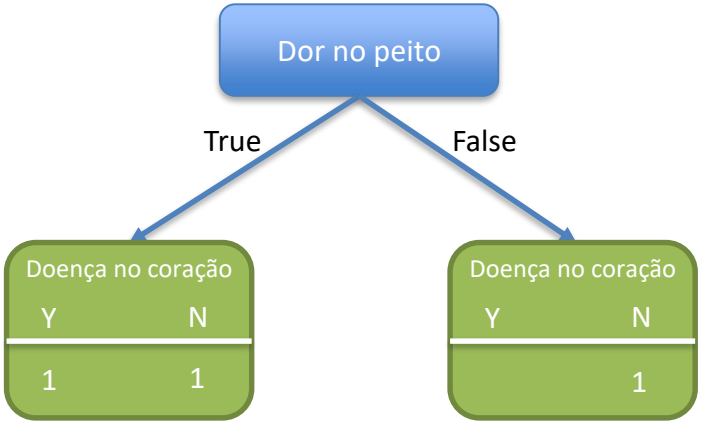
Dor no peito	Boa circulação Sanguínea	Artérias bloqueadas	Doença cardíaca
N	N	N	N
Y	Y	Y	Y
Y	Y	N	N
Y	N	?	Y
...	...	...	...



Decision Trees

Agora contamos os valores para cada combinação:

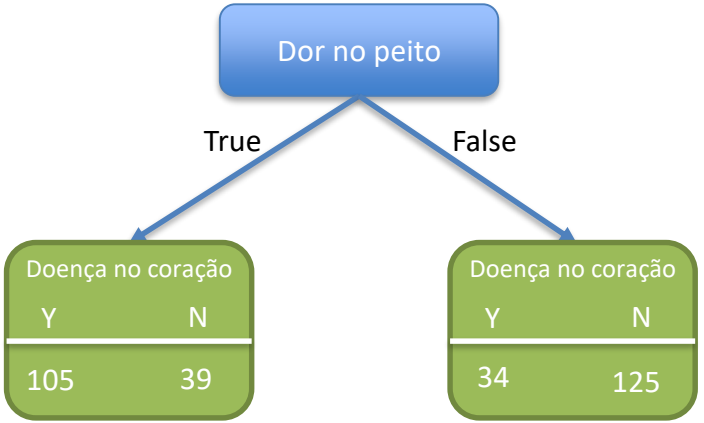
Dor no peito	Boa circulação Sanguínea	Artérias bloqueadas	Doença cardíaca
N	N	N	N
Y	Y	Y	Y
Y	Y	N	N
Y	N	?	Y
...	...	...	...



Decision Trees

E calculamos até obter o valor relativo a todas as linhas do data set:

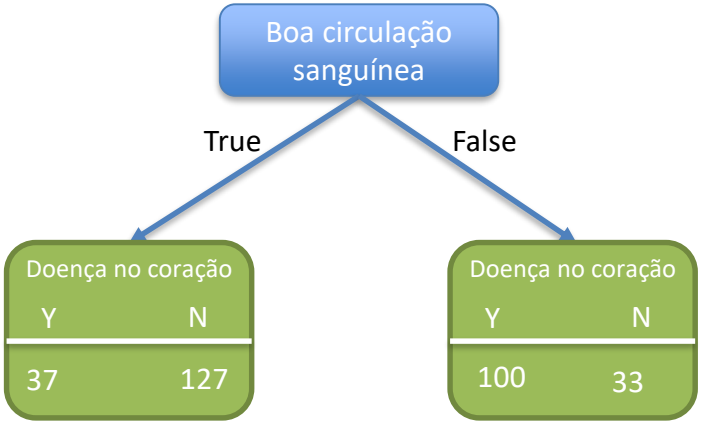
Dor no peito	Boa circulação Sanguínea	Artérias bloqueadas	Doença cardíaca
N	N	N	N
Y	Y	Y	Y
Y	Y	N	N
Y	N	?	Y
...	...	...	...



Decision Trees

Agora repetimos o processo para Boa circulação sanguínea

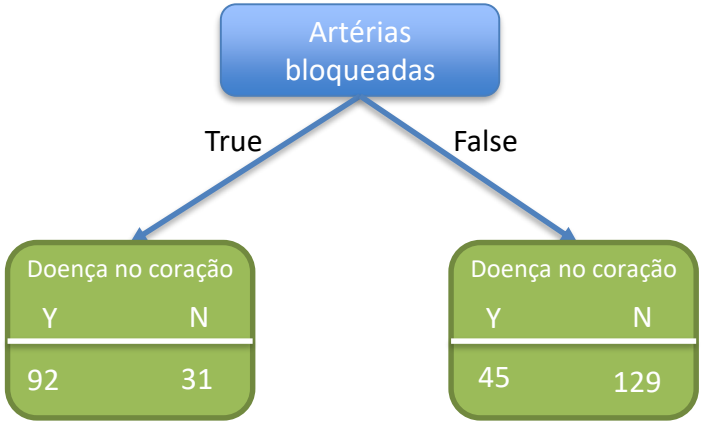
Dor no peito	Boa circulação Sanguínea	Artérias bloqueadas	Doença cardíaca
N	N	N	N
Y	Y	Y	Y
Y	Y	N	N
Y	N	?	Y
...	...	...	...



Decision Trees

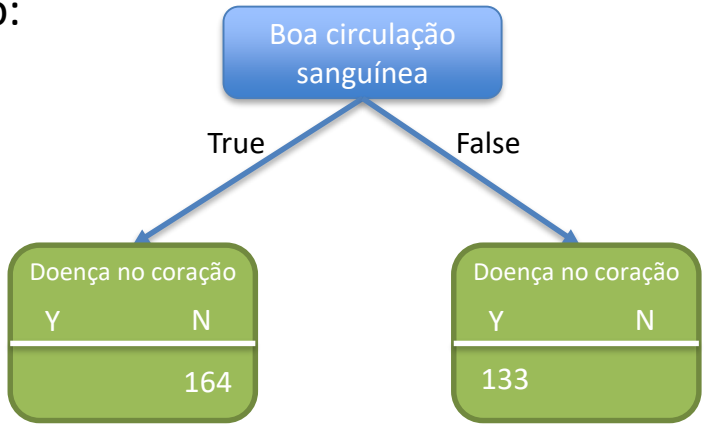
E para Artérias bloqueadas também:

Dor no peito	Boa circulação Sanguínea	Artérias bloqueadas	Doença cardíaca
N	N	N	N
Y	Y	Y	Y
Y	Y	N	N
Y	N	?	Y
...	...	...	...



Importante notar que, como não sabemos o valor para esse paciente, não o levamos em consideração

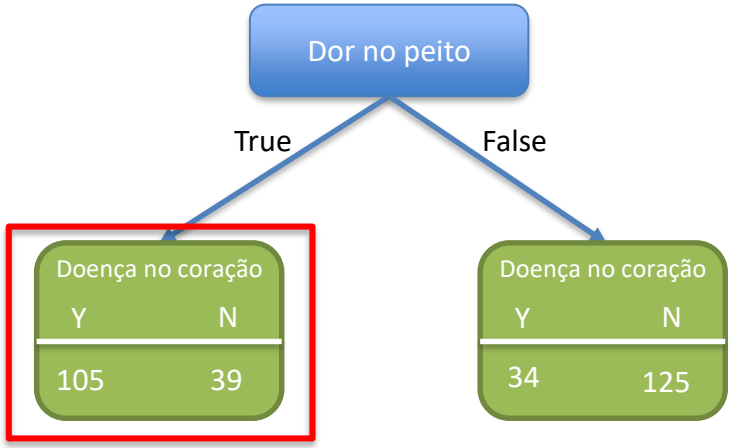
Agora, precisamos determinar quão bem cada feature conseguiu separar os pacientes com e sem doença cardíaca. Olhando os dados, vimos que nenhuma feature conseguiu uma separação perfeita. Exemplo:



Isso significa que todas as features são impuras. Assim, para determinar qual feature deve ser o nó raiz, precisamos de uma maneira de mensurar e comparar impureza.

Aqui, vamos usar a métrica Gini: $Gini = 1 - \sum_{i=1}^{k} p_i^2$

Decision Trees

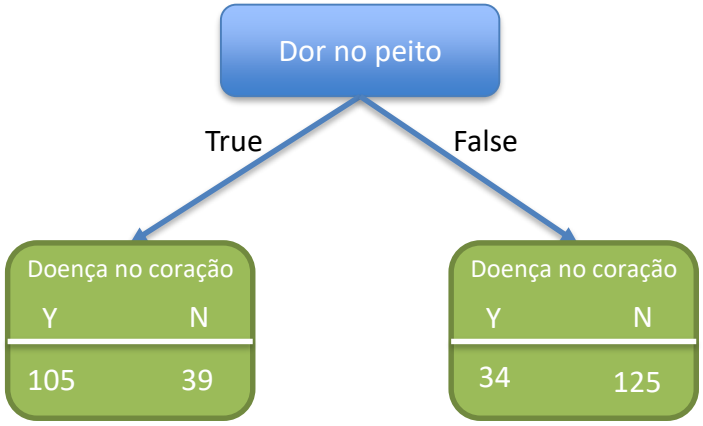


$$Gini = 1 - (p(Y)^2 - p(N)^2 = 1 - \left(\frac{105}{105+39}\right)^2 - \left(\frac{39}{105+39}\right)^2 = 0.395$$

Decision Trees

0.395

$$Gini = 1 - (p(Y)^2 - p(N)^2 = 1 - \left(\frac{34}{34+125}\right)^2 - \left(\frac{125}{34+125}\right)^2 = 0.336$$



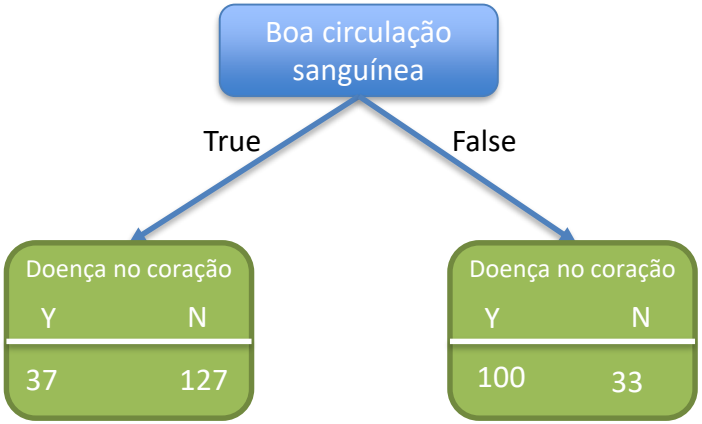
Agora que calculamos o Gini para ambas as folhas, podemos calcular o Gini total do nó Dor no peito para separar pacientes com ou sem doença cardíaca.

Vamos usar a média ponderada pela quantidade de pacientes para determinar o Gini total

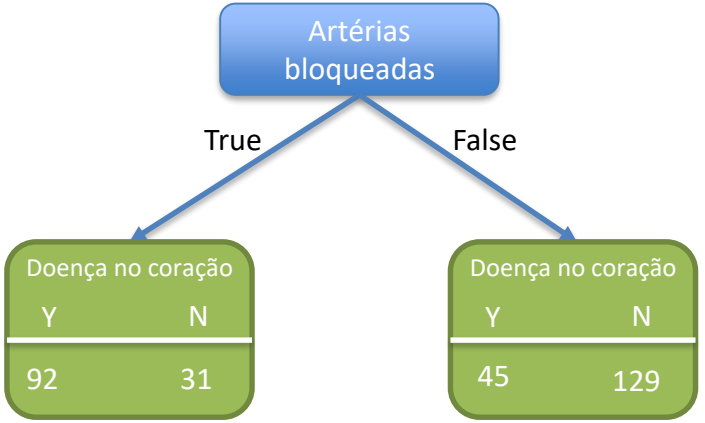
$$GiniTotal = \left(\frac{144}{144+159}\right) \times 0.395 + \left(\frac{159}{144+159}\right) \times 0.336 = 0.364$$

Decision Trees

Repetimos o processo para as outras features:



$Gini = 0.360$

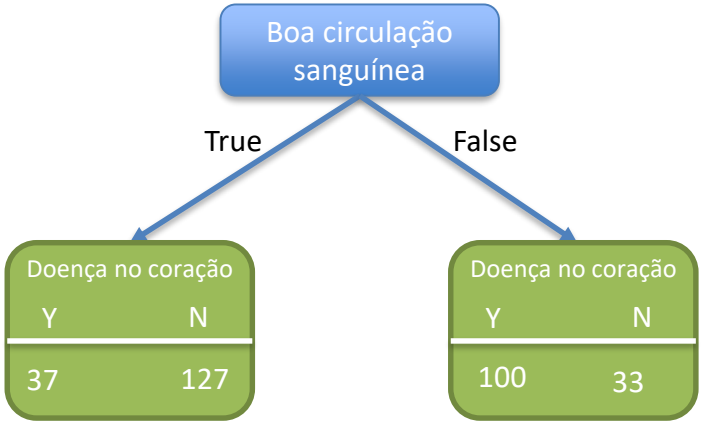


$Gini = 0.381$

Decision Trees

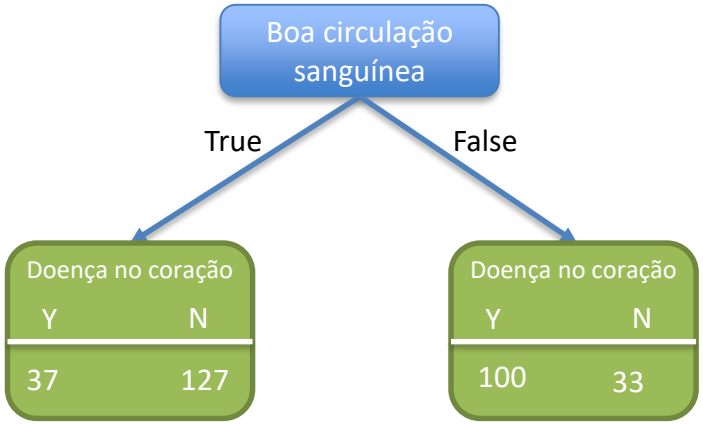
Das métricas, conseguimos observar que Boa circulação sanguínea possui o menor valor de Gini, isto é, é a feature que melhor separa os pacientes entre aqueles que possuem doença cardíaca e aqueles que não possuem.

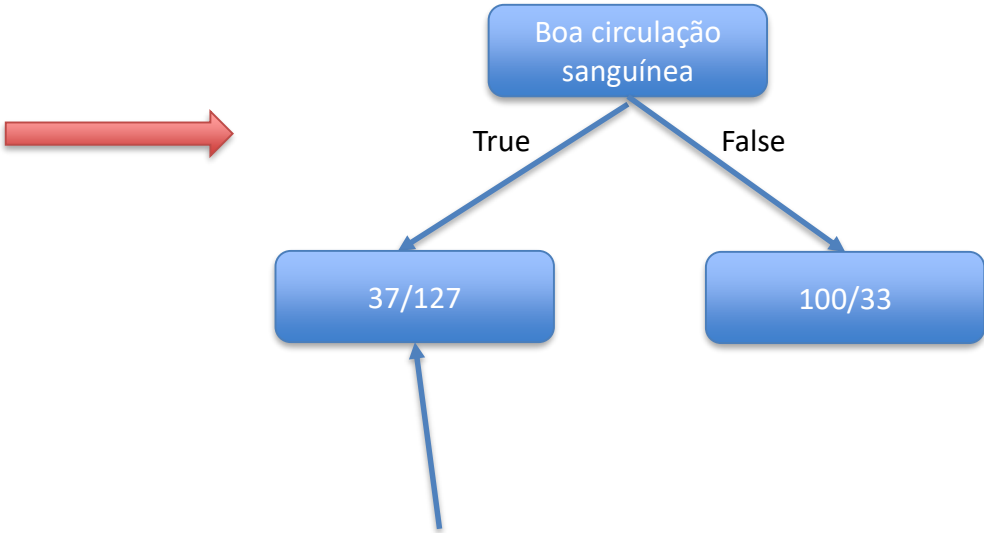
Portanto, ela sera usada como nó raiz:



Decision Trees

Importante notar que os nós folhas de Boa circulação sanguínea agora se comportarão como nós da árvore. Assim, temos que:

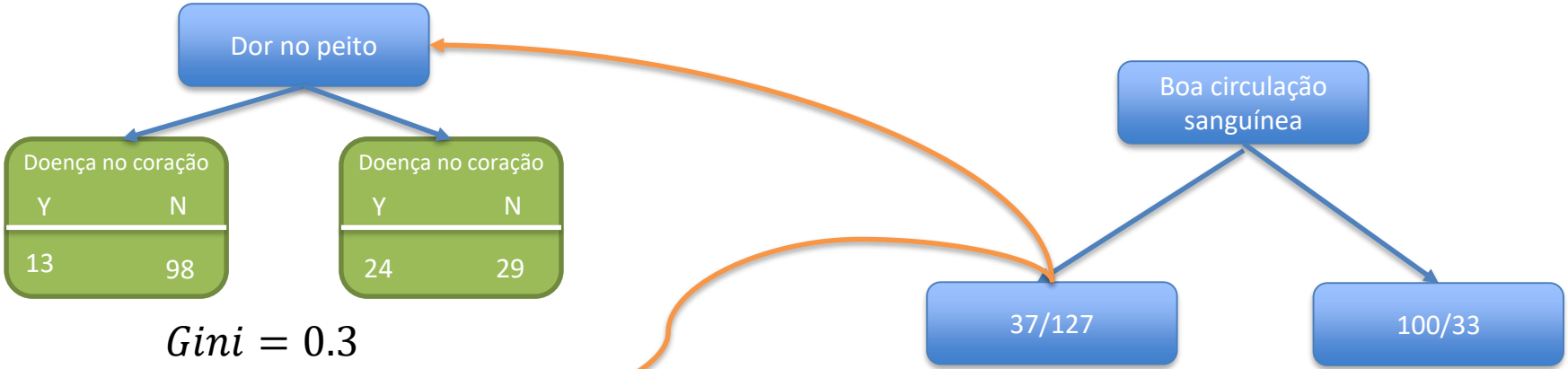




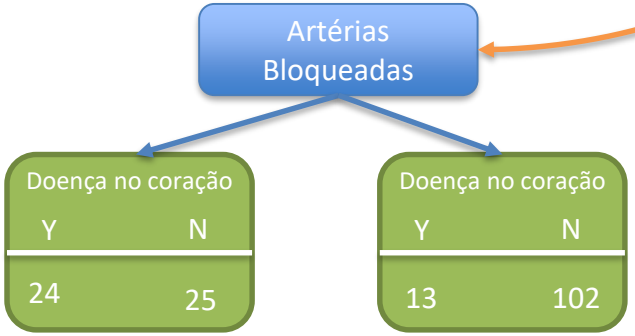
E agora precisamos descobrir como Dor no peito e Artérias bloqueadas separam esse nó.

Decision Trees

Da mesma forma que fizemos antes, separamos os pacientes baseados em Dor no peito e Artérias bloqueadas e calculamos o valor da impureza



$Gini = 0.3$

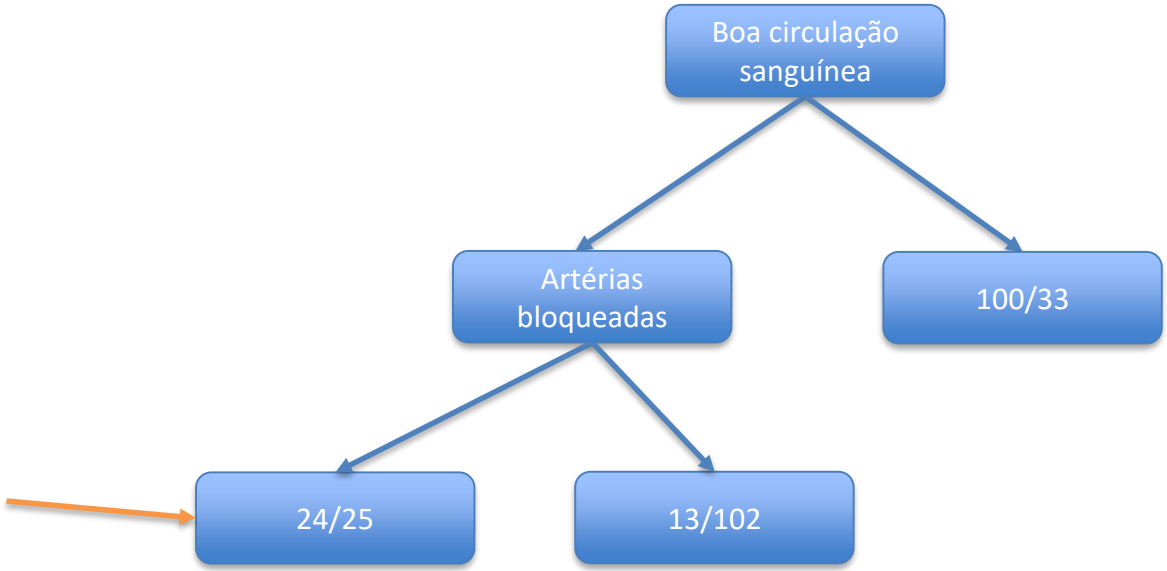


$Gini = 0.29$

Como Artérias bloqueadas possui o menor valor de Gini, vamos usar ela para separar os pacientes

Decision Trees

Esta é a árvore que temos até agora:

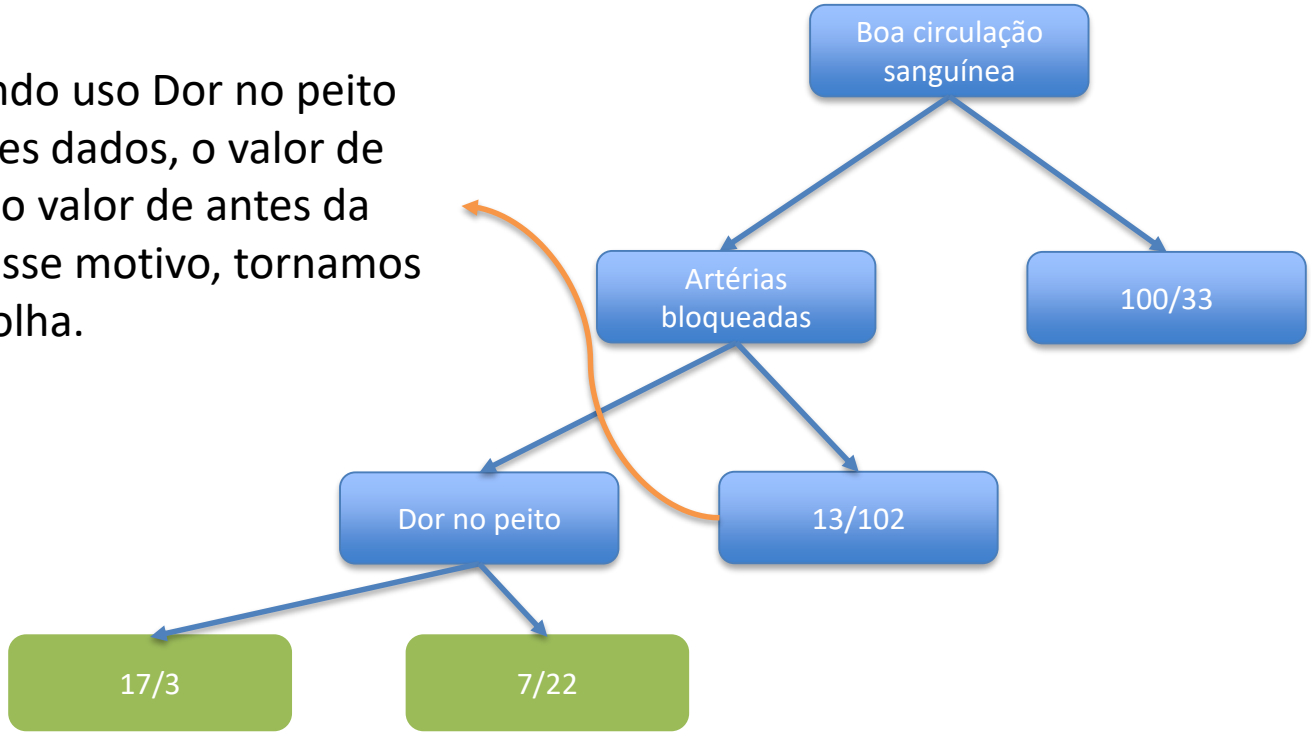


Tudo que restou é a feature Dor no peito, então vamos ver quão bem ela separa essas amostras.

Decision Trees

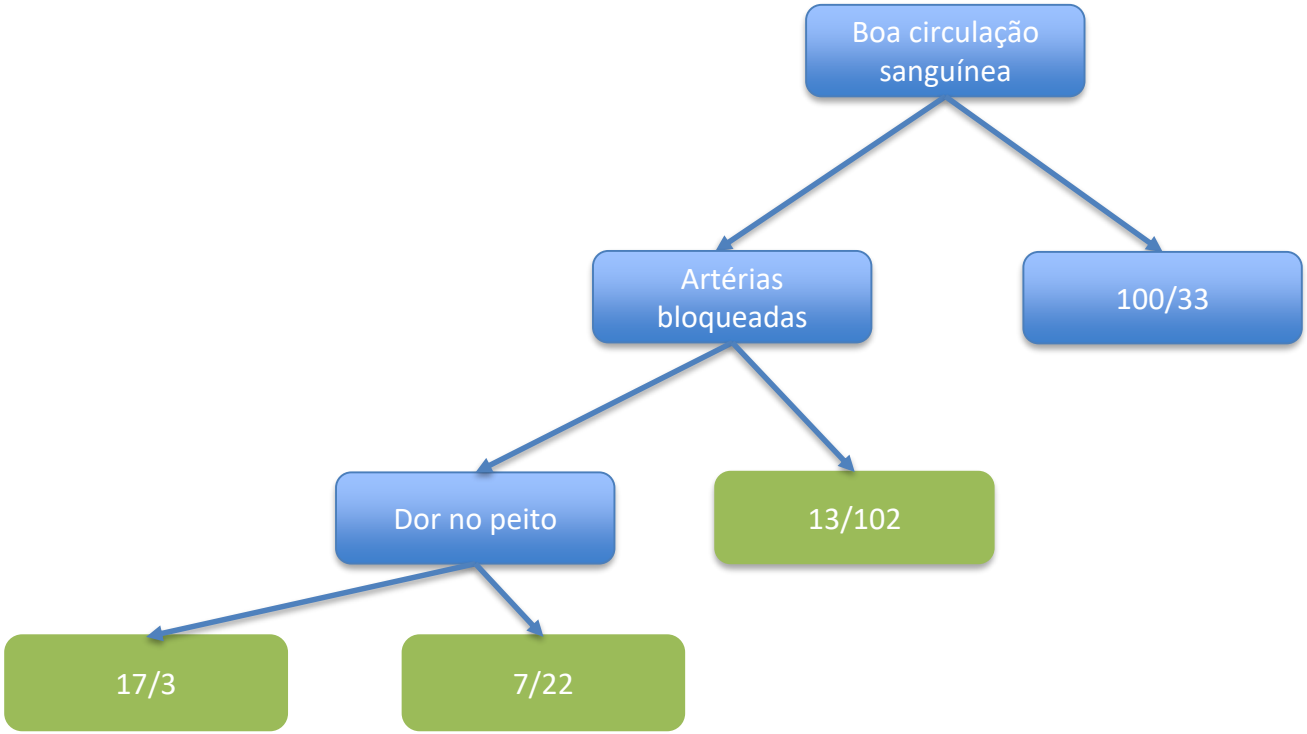
A separação forneceu bons resultados.

Entretanto, quando uso Dor no peito para separar esses dados, o valor de Gini é superior ao valor de antes da separação. Por esse motivo, tornamos esse nó um nó folha.



Decision Trees

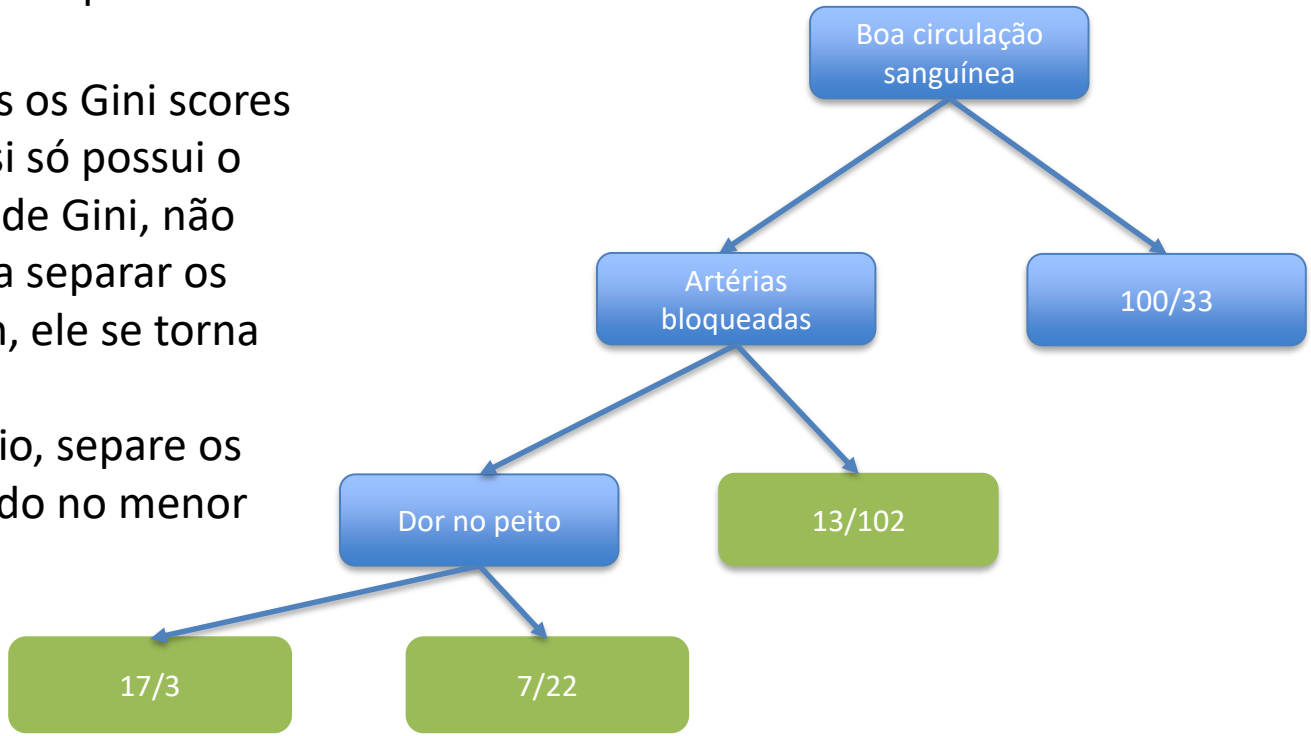
A separação forneceu bons resultados.



Decision Trees

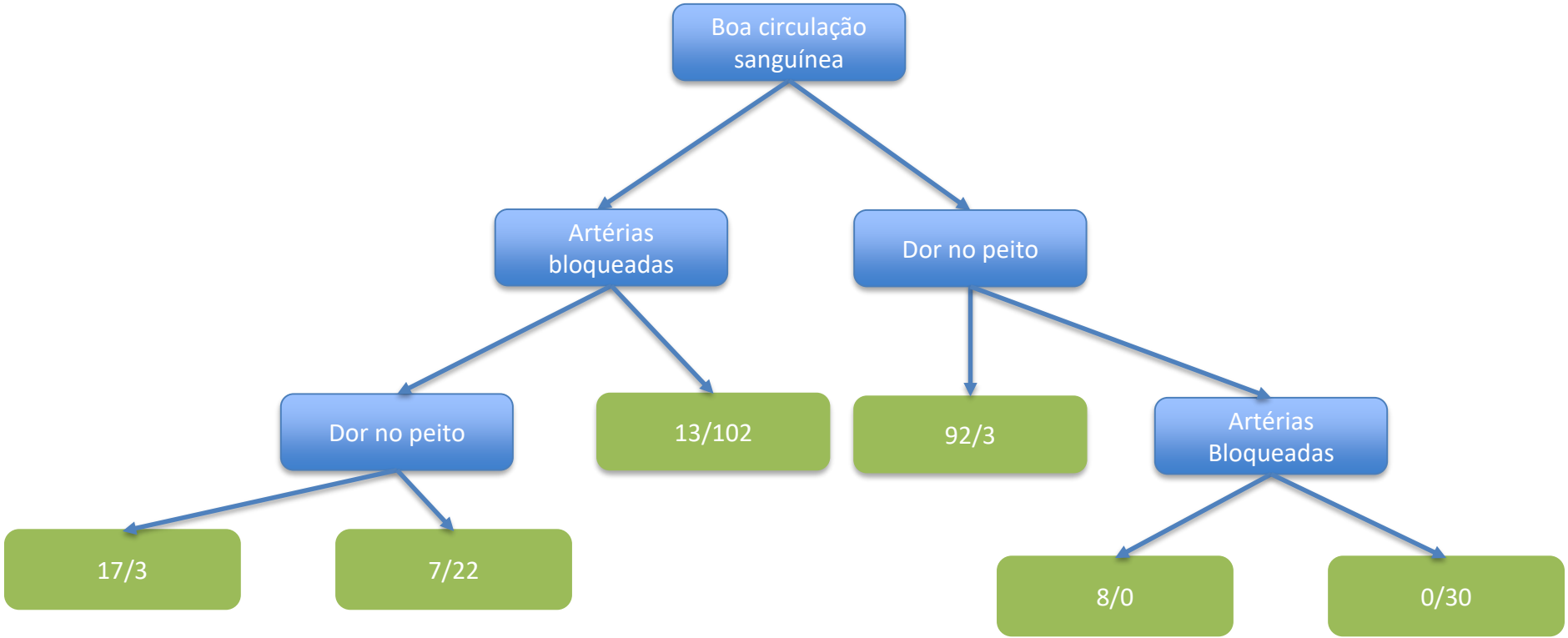
Agora, precisamos trabalhar no lado direito da árvore. Para isso, só repetimos os passos vistos até aqui:

1. Calcule todos os Gini scores
2. Se o nó por si só possui o menor valor de Gini, não há razão para separar os dados. Assim, ele se torna um nó folha
3. Caso contrário, separe os dados baseado no menor valor de Gini



Decision Trees

E obteremos essa árvore



Decision Trees

Uma importante etapa nas árvores de decisão é o ajuste nos hiperparâmetros.

Antes, porém, é importante fazer uma distinção entre parâmetros e hiperparâmetros:

1. **Parâmetros**: são ajustados em tempo de execução do modelo, mais especificamente na etapa de treino.

2. **Hiperparâmetros**: são ajustados antes da etapa de treino. Obviamente, se não valor for selecionado, os valores padrões serão usados.

Aqui, vamos focar em encontrar os melhores hiperparametros.

Decision Trees

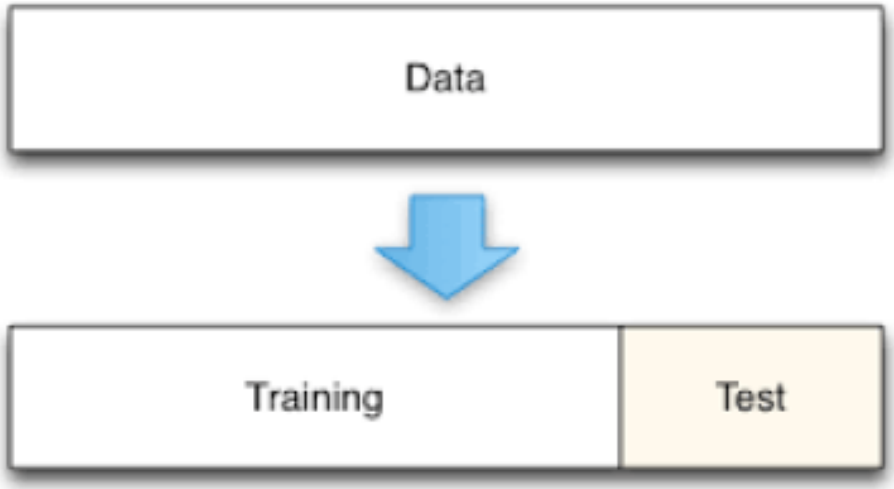
O primeiro hiperparâmetros que vamos entender é o ***max_depth***, que define a profundidade de uma árvore. Por padrão, não há limite para *max_depth*, então podem existir milhões de splits numa árvore, resultando em overfitting. Limitando *max_depth* a pequenos números, a variância é reduzida e o modelo generaliza melhor para novos dados.

O segundo hiperparâmetro é o ***min_samples_leaf***, que provê uma restrição aumentando o número de amostras que uma folha pode ter. Quando não há, restrição, *min_samples_leaf=1*, o que significa que os nós folhas podem conter apenas uma amostra (propenso a overfitting). Quando aumentamos *min_samples_leaf*, reduzimos variância.

Existem vários outros hiperparâmetros a serem analisados e a documentação da scikit-learn provê detalhes de cada um deles.

Formas de Validação

Holdout



K-fold

- The dataset is partitioned into K equal sized samples

KNN

Intuição

Imagine que você esteja tentando predizer qual meu filme favorito. Se você não sabe mais nada a meu respeito, uma abordagem seria perguntar o filme favorito das pessoas que convivem comigo.

Agora imagine que você possua mais informações sobre mim: quais serviçoes de streaming eu assisto, o tipo de filme que mais gosto, meu diretor e ator favoritos e assim por diante.

Na medida em que meu comportamento é influenciado por essas coisas, olhar apenas para os amigos mais chegados dentro todos os que convivem comigo torna-se um preditor melhor do meu filme favorito.

Esta é a ideia por trás do KNN

Intuição

KNN requer apenas duas coisas para executar:

1. Alguma noção de distância

2. Uma suposição de que pontos próximos uns aos outros são similares

Vamos entender o algoritmo em mais detalhes:

Características

KNN possui três características importantes:

1. Aprendizado baseado em exemplos:
    1. Todas as instâncias de treino são usadas para se fazer uma predição

2. Aprendizado preguiçoso:
    1. Não existe aprendizado do modelo e todo trabalho acontece no tempo de predição

3. Não paramétrico:
    1. O KNN não faz suposições sobre o problema a ser resolvido

Medida de Distância

- Os exemplos correspondem a pontos no espaço n-dimensional;
- Em geral, os vizinhos são definidos em função de uma medida de distância. Podemos usar a distância Euclidiana, Manhattan ou Mahalanobis, pra citar algumas
- As distâncias possuem a seguinte formulação matemática:

$$d(x,y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}$$

Euclidiana

$$d(x,y) = \sum_{i=1}^{n}|x_i - y_i|$$

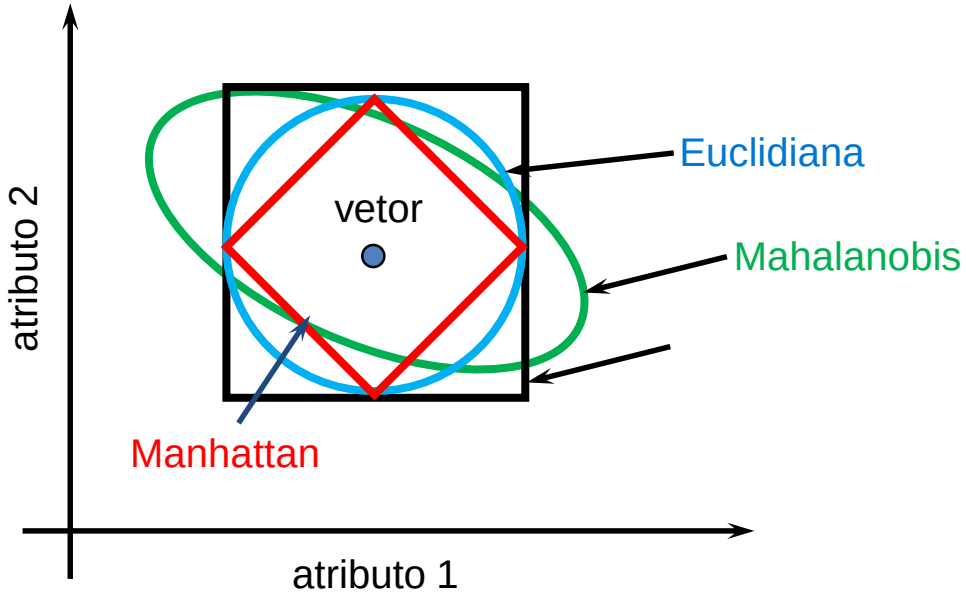
Manhattan

$$d(x,y) = \sqrt{(x-y)^T S^{-1}(x-y)}$$

Mahalanobis

Medida de Distância

Para um espaço de duas características, elas podem ser interpretadas da seguinte maneira:



KNN - Exemplo

Vamos usar a distância Euclidiana para entender o funcionamento do KNN. No exemplo a seguir, eu quero determinar a classe da amostra #11.

#	a1	a2	Classe
1	0.5	1	2
2	2.9	1.9	2
3	1.2	3.1	2
4	0.8	4.7	2
5	2.7	5.4	2
6	8.1	4.7	1
7	8.3	6.6	1
8	6.3	6.7	1
9	8	9.1	1
10	5.4	8.4	1
11	**5**	**7**	**?**

$$d_E(\mathbf{x},\mathbf{y}) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}$$

$$d_E(\#1,\#11) = \sqrt{(0.5-5)^2+(1-7)^2}$$

$$d_E(\#1,\#11) = \sqrt{(-4.5)^2+(-6)^2}$$

$$d_E(\#1,\#11) = \sqrt{20.25+36}$$

$$d_E(\#1,\#11) = \sqrt{56.25}$$

$$d_E(\#1,\#11) = 7.5$$

1. Determine a distância de #11 para todas as demais amostras

KNN - Exemplo

Vamos usar a distância Euclidiana para entender o funcionamento do KNN. No exemplo a seguir, eu quero determinar a classe da amostra #11.

#	a1	a2	Classe
1	0.5	1	2
2	2.9	1.9	2
3	1.2	3.1	2
4	0.8	4.7	2
5	2.7	5.4	2
6	8.1	4.7	1
7	8.3	6.6	1
8	6.3	6.7	1
9	8	9.1	1
10	5.4	8.4	1
11	**5**	**7**	**?**

$$d_E(\mathbf{x},\mathbf{y}) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}$$

$$d_E(\#2,\#11) = \sqrt{(2.9-5)^2+(1.9-7)^2}$$

$$d_E(\#2,\#11) = \sqrt{(-2.1)^2+(-5.1)^2}$$

$$d_E(\#2,\#11) = \sqrt{4.41+26.1}$$

$$d_E(\#2,\#11) = \sqrt{30.51}$$

$$d_E(\#2,\#11) = 5.5$$

KNN - Exemplo

Vamos usar a distância Euclidiana para entender o funcionamento do KNN. No exemplo a seguir, eu quero determinar a classe da amostra #11.

#	a1	a2	Classe
1	0.5	1	2
2	2.9	1.9	2
3	1.2	3.1	2
4	0.8	4.7	2
5	2.7	5.4	2
6	8.1	4.7	1
7	8.3	6.6	1
8	6.3	6.7	1
9	8	9.1	1
10	5.4	8.4	1
11	**5**	**7**	**?**

$$d_E(\mathbf{x},\mathbf{y}) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}$$

$$d_E(\#3,\#11) = 5.4$$

KNN - Exemplo

Vamos usar a distância Euclidiana para entender o funcionamento do KNN. No exemplo a seguir, eu quero determinar a classe da amostra #11.

#	a1	a2	Classe
1	0.5	1	2
2	2.9	1.9	2
3	1.2	3.1	2
4	0.8	4.7	2
5	2.7	5.4	2
6	8.1	4.7	1
7	8.3	6.6	1
8	6.3	6.7	1
9	8	9.1	1
10	5.4	8.4	1
11	**5**	**7**	**?**

$$d_E(\mathbf{x},\mathbf{y}) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}$$

$$d_E(\#4,\#11) = 4.7$$

KNN - Exemplo

Vamos usar a distância Euclidiana para entender o funcionamento do KNN. No exemplo a seguir, eu quero determinar a classe da amostra #11.

#	a1	a2	Classe
1	0.5	1	2
2	2.9	1.9	2
3	1.2	3.1	2
4	0.8	4.7	2
5	2.7	5.4	2
6	8.1	4.7	1
7	8.3	6.6	1
8	6.3	6.7	1
9	8	9.1	1
10	5.4	8.4	1
11	**5**	**7**	**?**

$$d_E(\mathbf{x},\mathbf{y}) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}$$

$$d_E(\#5,\#11) = 2.8$$

KNN - Exemplo

Vamos usar a distância Euclidiana para entender o funcionamento do KNN. No exemplo a seguir, eu quero determinar a classe da amostra #11.

#	a1	a2	Classe
1	0.5	1	2
2	2.9	1.9	2
3	1.2	3.1	2
4	0.8	4.7	2
5	2.7	5.4	2
6	8.1	4.7	1
7	8.3	6.6	1
8	6.3	6.7	1
9	8	9.1	1
10	5.4	8.4	1
11	5	7	2

$$d_E(\mathbf{x},\mathbf{y}) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}$$

$$d_E(\#6,\#11) = 3.8$$

KNN - Exemplo

Vamos usar a distância Euclidiana para entender o funcionamento do KNN. No exemplo a seguir, eu quero determinar a classe da amostra #11.

#	a1	a2	Classe
1	0.5	1	2
2	2.9	1.9	2
3	1.2	3.1	2
4	0.8	4.7	2
5	2.7	5.4	2
6	8.1	4.7	1
7	8.3	6.6	1
8	6.3	6.7	1
9	8	9.1	1
10	5.4	8.4	1
11	**5**	**7**	**?**

$$d_E(\mathbf{x},\mathbf{y}) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}$$

$$d_E(\#7,\#11) = 3.3$$

KNN - Exemplo

Vamos usar a distância Euclidiana para entender o funcionamento do KNN. No exemplo a seguir, eu quero determinar a classe da amostra #11.

#	a1	a2	Classe
1	0.5	1	2
2	2.9	1.9	2
3	1.2	3.1	2
4	0.8	4.7	2
5	2.7	5.4	2
6	8.1	4.7	1
7	8.3	6.6	1
8	6.3	6.7	1
9	8	9.1	1
10	5.4	8.4	1
11	**5**	**7**	**?**

$$d_E(\mathbf{x},\mathbf{y}) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}$$

$$d_E(\#8,\#11) = 1.3$$

KNN - Exemplo

Vamos usar a distância Euclidiana para entender o funcionamento do KNN. No exemplo a seguir, eu quero determinar a classe da amostra #11.

#	a1	a2	Classe
1	0.5	1	2
2	2.9	1.9	2
3	1.2	3.1	2
4	0.8	4.7	2
5	2.7	5.4	2
6	8.1	4.7	1
7	8.3	6.6	1
8	6.3	6.7	1
9	8	9.1	1
10	5.4	8.4	1
11	**5**	**7**	**?**

$$d_E(\mathbf{x},\mathbf{y}) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}$$

$$d_E(\#9,\#11) = 3.6$$

KNN - Exemplo

Vamos usar a distância Euclidiana para entender o funcionamento do KNN. No exemplo a seguir, eu quero determinar a classe da amostra #11.

#	a1	a2	Classe
1	0.5	1	2
2	2.9	1.9	2
3	1.2	3.1	2
4	0.8	4.7	2
5	2.7	5.4	2
6	8.1	4.7	1
7	8.3	6.6	1
8	6.3	6.7	1
9	8	9.1	1
10	5.4	8.4	1
11	**5**	**7**	**?**

$$d_E(\mathbf{x},\mathbf{y}) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}$$

$$d_E(\#10,\#11) = 1.4$$

KNN - Exemplo

Vamos usar a distância Euclidiana para entender o funcionamento do KNN. No exemplo a seguir, eu quero determinar a classe da amostra #11.

#	a1	a2	Classe	Distância para a instância #11
1	0.5	1	2	7.5
2	2.9	1.9	2	5.5
3	1.2	3.1	2	5.4
4	0.8	4.7	2	4.7
5	2.7	5.4	2	2.8
6	8.1	4.7	1	3.8
7	8.3	6.6	1	3.3
8	6.3	6.7	1	1.3
9	8	9.1	1	3.6
10	5.4	8.4	1	1.4
11	**5**	**7**	**?**	

KNN - Exemplo

Vamos usar a distância Euclidiana para entender o funcionamento do KNN. No exemplo a seguir, eu quero determinar a classe da amostra #11.

#	a1	a2	Classe	Distância para a instância #11
1	0.5	1	2	7.5
2	2.9	1.9	2	5.5
3	1.2	3.1	2	5.4
4	0.8	4.7	2	4.7
5	2.7	5.4	2	2.8
6	8.1	4.7	1	3.8
7	8.3	6.6	1	3.3
8	6.3	6.7	1	1.3
9	8	9.1	1	3.6
10	5.4	8.4	1	1.4
11	**5**	**7**	**?**	

1. Determine a distância de #11 para todas as demais amostras

2. Escolha um valor de k para determinar a classe de #11

KNN - Exemplo

Vamos usar a distância Euclidiana para entender o funcionamento do KNN. No exemplo a seguir, eu quero determinar a classe da amostra #11.

#	a1	a2	Classe	Distância para a instância #11
1	0.5	1	2	7.5
2	2.9	1.9	2	5.5
3	1.2	3.1	2	5.4
4	0.8	4.7	2	4.7
5	2.7	5.4	2	2.8
6	8.1	4.7	1	3.8
7	8.3	6.6	1	3.3
8	6.3	6.7	1	1.3
9	8	9.1	1	3.6
10	5.4	8.4	1	1.4
11	5	7	?	

Para k=1, temos o 1-NN e a classe da instância #11 seria 1.

1. Determine a distância de #11 para todas as demais amostras
2. Escolha um valor de k para determinar a classe de #11

KNN - Exemplo

Vamos usar a distância Euclidiana para entender o funcionamento do KNN. No exemplo a seguir, eu quero determinar a classe da amostra #11.

#	a1	a2	Classe	Distância para a instância #11
1	0.5	1	2	7.5
2	2.9	1.9	2	5.5
3	1.2	3.1	2	5.4
4	0.8	4.7	2	4.7
5	2.7	5.4	2	2.8
6	8.1	4.7	1	3.8
7	8.3	6.6	1	3.3
8	6.3	6.7	1	1.3
9	8	9.1	1	3.6
10	5.4	8.4	1	1.4
11	5	7	?	

Para k=2, temos o 2-NN e a classe da instância #11 seria ?

1. Determine a distância de #11 para todas as demais amostras
2. Escolha um valor de k para determinar a classe de #11

KNN - Exemplo

Vamos usar a distância Euclidiana para entender o funcionamento do KNN. No exemplo a seguir, eu quero determinar a classe da amostra #11.

#	a1	a2	Classe	Distância para a instância #11
1	0.5	1	2	7.5
2	2.9	1.9	2	5.5
3	1.2	3.1	2	5.4
4	0.8	4.7	2	4.7
5	2.7	5.4	2	2.8
6	8.1	4.7	1	3.8
7	8.3	6.6	1	3.3
8	6.3	6.7	1	1.3
9	8	9.1	1	3.6
10	5.4	8.4	1	1.4
11	**5**	**7**	**?**	

Para k=2, temos o 2-NN e a classe da instância #11 seria 1

1. Determine a distância de #11 para todas as demais amostras

2. Escolha um valor de k para determinar a classe de #11

KNN - Exemplo

Vamos usar a distância Euclidiana para entender o funcionamento do KNN. No exemplo a seguir, eu quero determinar a classe da amostra #11.

#	a1	a2	Classe	Distância para a instância #11
1	0.5	1	2	7.5
2	2.9	1.9	2	5.5
3	1.2	3.1	2	5.4
4	0.8	4.7	2	4.7
5	2.7	5.4	2	2.8
6	8.1	4.7	1	3.8
7	8.3	6.6	1	3.3
8	6.3	6.7	1	1.3
9	8	9.1	1	3.6
10	5.4	8.4	1	1.4
11	5	7	?	

Para k=3, temos o 3-NN e a classe da instância #11 seria ?

1. Determine a distância de #11 para todas as demais amostras

2. Escolha um valor de k para determinar a classe de #11

KNN - Exemplo

Vamos usar a distância Euclidiana para entender o funcionamento do KNN. No exemplo a seguir, eu quero determinar a classe da amostra #11.

#	a1	a2	Classe	Distância para a instância #11
1	0.5	1	2	7.5
2	2.9	1.9	2	5.5
3	1.2	3.1	2	5.4
4	0.8	4.7	2	4.7
5	2.7	5.4	2	2.8
6	8.1	4.7	1	3.8
7	8.3	6.6	1	3.3
8	6.3	6.7	1	1.3
9	8	9.1	1	3.6
10	5.4	8.4	1	1.4
11	5	7	?	

Para k=3, temos o 3-NN e a classe da instância #11 seria 1, pois temos 2 vizinho da classe 1 e apenas 1 vizinho da classe 2.

1. Determine a distância de #11 para todas as demais amostras
2. Escolha um valor de k para determinar a classe de #11

KNN - Exemplo

Vamos usar a distância Euclidiana para entender o funcionamento do KNN. No exemplo a seguir, eu quero determinar a classe da amostra #11.

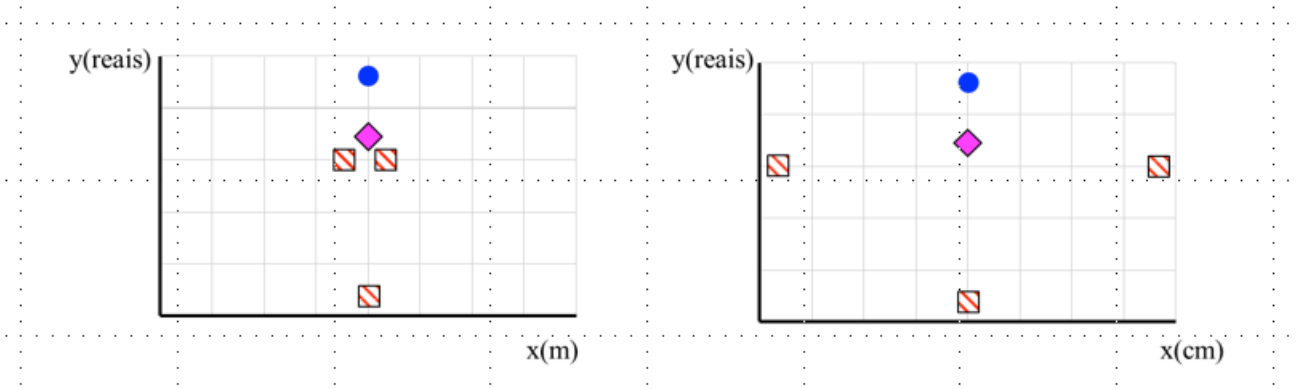
#	a1	a2	Classe	Distância para a instância #11
1	0.5	1	2	7.5
2	2.9	1.9	2	5.5
3	1.2	3.1	2	5.4
4	0.8	4.7	2	4.7
5	2.7	5.4	2	2.8
6	8.1	4.7	1	3.8
7	8.3	6.6	1	3.3
8	6.3	6.7	1	1.3
9	8	9.1	1	3.6
10	5.4	8.4	1	1.4
11	5	7	?	

Para k=4, temos o 4-NN e a classe da instância #11 seria ?

1. Determine a distância de #11 para todas as demais amostras
2. Escolha um valor de k para determinar a classe de #11

KNN – Sensibilidade a Escala

Um dos problemas relacionados ao KNN é a sensibilidade em relação à escala. Observe o exemplo abaixo:

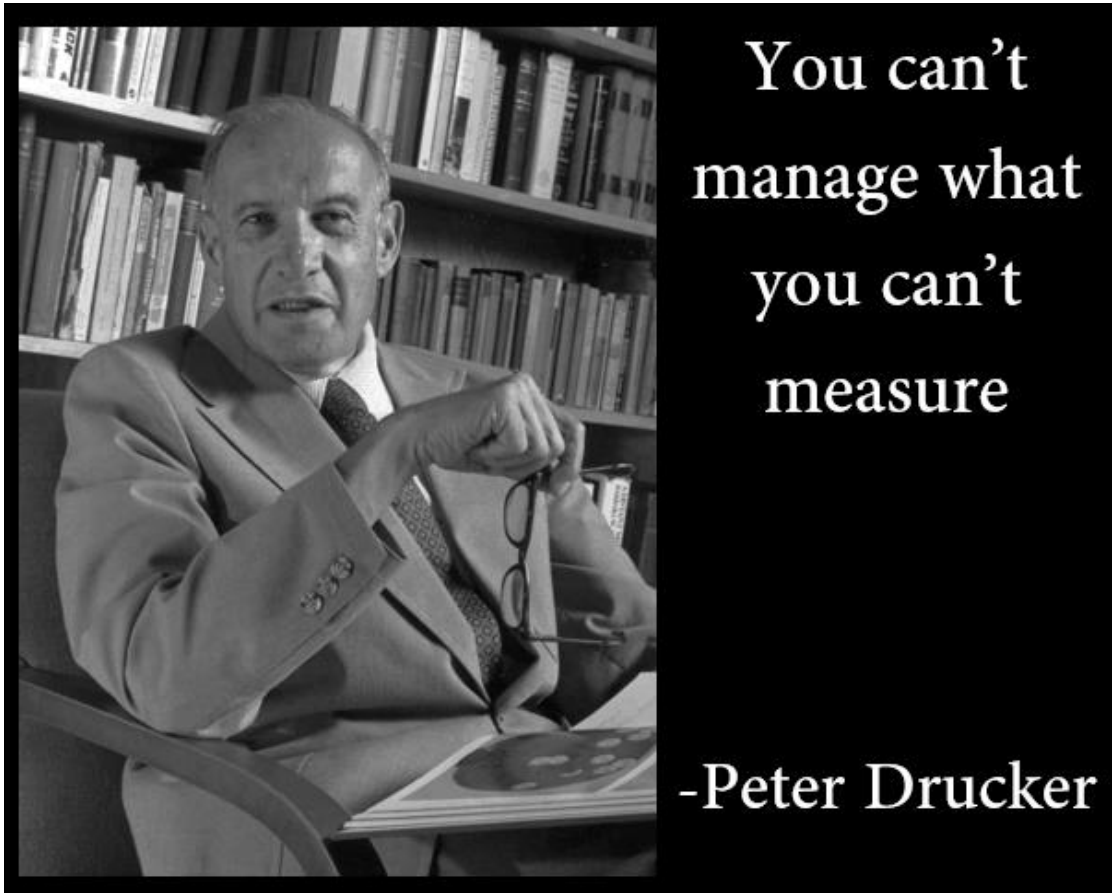


Como podemos resolver?

Em geral, normalização dos dados resolvem o problema (StandardScaler, MinMaxScaler)

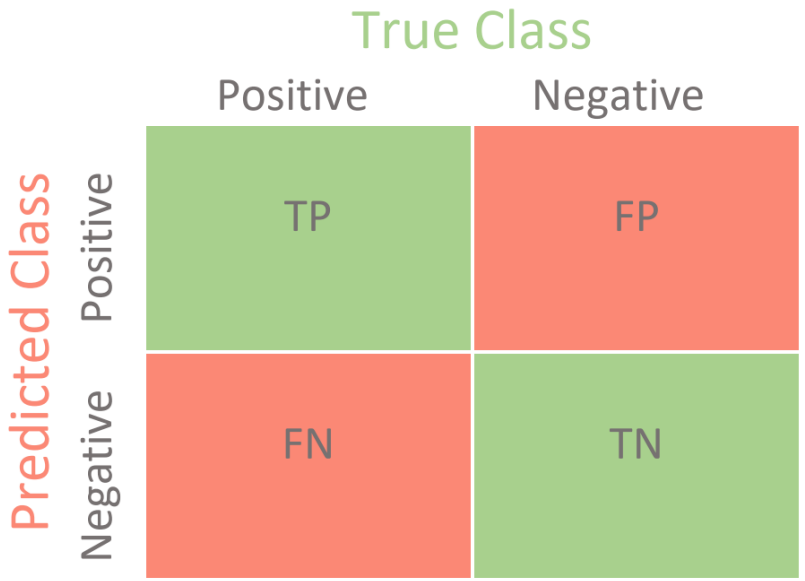
Métricas de Avaliação

Métricas de Avaliação



Métricas de Avaliação

Precisamos de maneiras de mensurar a qualidade de predição de nosso algoritmo. Em classificação, as métricas mais comuns, e as mais simples, são **Acurácia, Precision, Recall e F1 Score**. Elas são obtidas a partir da Matriz de Confusão, apresentada abaixo:



Métricas de Avaliação

A acurácia é dada pela seguinte função:

$$acc = \frac{tp + tn}{(tp + tn + fp + fn)}$$

Precision (acurácia das predições positivas) pode ser calculado da seguinte maneira:

$$P = \frac{tp}{(tp + fp)}$$

Já o Recall (taxa de amostras positivas corretamente encontradas pelo classificador) é calculado da seguinte forma:

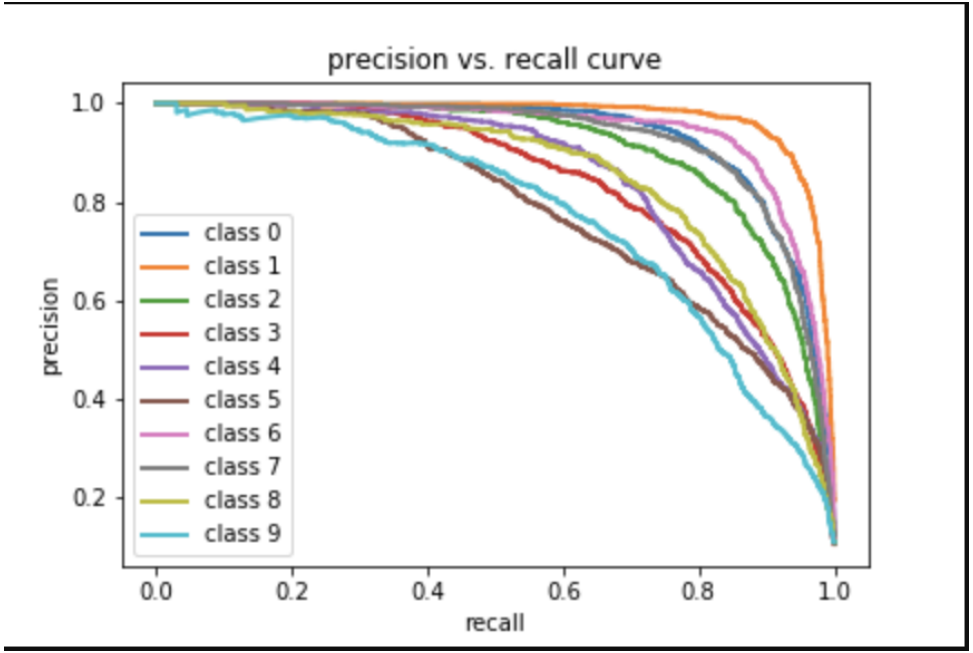
$$R = \frac{tp}{(tp + fn)}$$

Métricas de Avaliação

Precision e **Recall** tendem a ser inversamente proporcionais, como apresentado no gráfico ao lado.

Para minimizar esse problema, existe a métrica **F1-Score**, que leva em consideração tanto Precision quanto Recall.

$$F1 = \frac{2PR}{P + R}$$



Implementação

Obrigado!

profdheny.fernandes@fiap.com.br

/dhenyfernandes


FIAP