Classificação: Árvores de decisão

Alexandre Checoli Choueiri

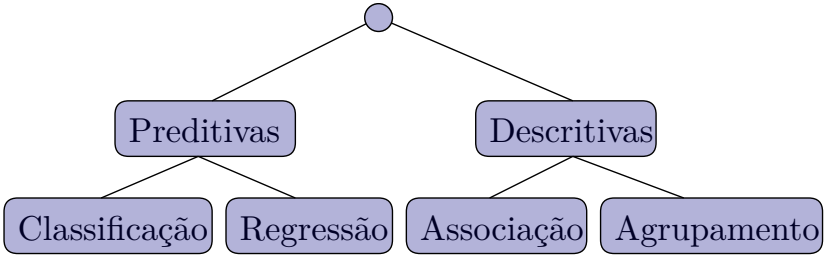
18/12/2022

Conteúdo

1. Aprendizagem
2. Fluxo de atividades
3. Árvores de decisão
4. Indução de árvores de decisão (algoritmo de Hunt)
5. Entropia
6. Métodos de separação dos dados
7. Medidas de desempenho, erros e overfitting
8. Tipos de erros e overfitting do modelo
9. Conclusão

Aprendizagem

As tarefas de mineração de dados podem ser separadas em 2 grandes grupo: tarefas **preditivas** e **descritivas**. Por sua vez, as tarefas preditivas são agrupadas em tarefas de **classificação** e de **regressão**. As tarefas preditivas estão ligadas ao conceito de *aprendizagem supervisionada*.



Definição

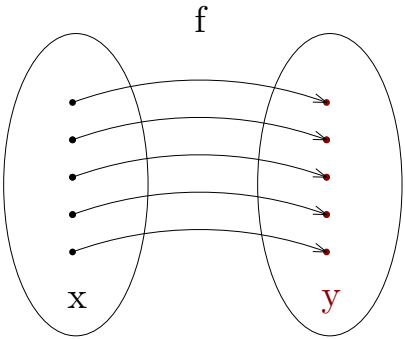
Aprendizado supervisionado: Dado um conjunto de treinamento de N pares de exemplos de entrada e saída:

$$(x_1, y_1), (x_2, y_2), ..., (x_N, y_N)$$

em que cada y_i foi gerado por uma **função desconhecida** $f(x_i) = y_i$, o aprendizado supervisionado busca descobrir uma função h que se aproxime de f.

1 - Aprendizagem

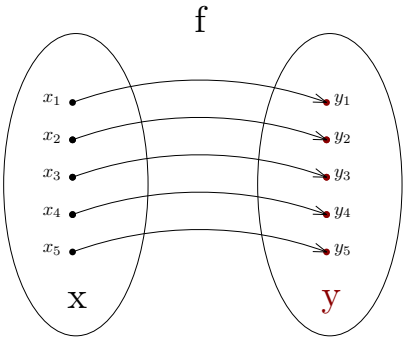
Relembrando as funções



As funções podem ser pensadas como *regras* que processam uma entrada gerando uma saída.

1 - Aprendizagem

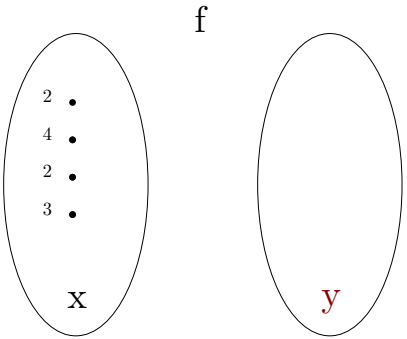
Relembrando as funções



Também como **máquinas**, que recebem um **input** e geram um **output**.

1 - Aprendizagem

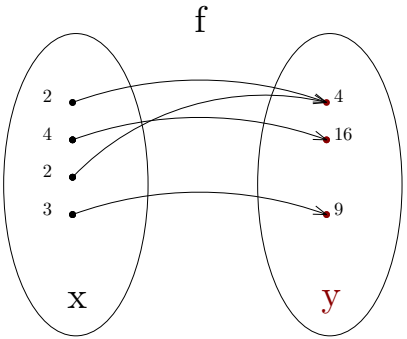
Relembrando as funções



Por exemplo: considerando os conjuntos de entrada acima, com a função $f(x) = x^2$, você consegue definir o conjunto de *output* dessa função?

1 - Aprendizagem

Relembrando as funções



Por exemplo: considerando os conjuntos de entrada acima, com a função $f(x) = x^2$, você consegue definir o conjunto de *output* dessa função?

1 - Aprendizagem

Relembrando as funções

A definição formal de funções é muito abstrata.

1 - Aprendizagem

Relembrando as funções

A definição formal de funções é muito abstrata.

Definição

Função: Uma relação f é chamada *função* desde que $(a, b) \in f$ e $(a, c) \in f$ impliquem $b = c$.[a]

[a] Na forma negativa, uma relação f não é uma função se existem a, b, c com $(a, b) \in f$ e $(a, c) \in f$ e $b \neq c$

1 - Aprendizagem

Relembrando as funções

A definição formal de funções é muito abstrata.

Definição

Função: Uma relação f é chamada *função* desde que $(a, b) \in f$ e $(a, c) \in f$ impliquem $b = c$.[a]

[a] Na forma negativa, uma relação f não é uma função se existem a, b, c com $(a, b) \in f$ e $(a, c) \in f$ e $b \neq c$

Definição

Relação: Uma relação é um conjunto de pares ordenados.

1 - Aprendizagem

Relembrando as funções

A definição formal de funções é muito abstrata.

Definição

Função: Uma relação f é chamada *função* desde que $(a, b) \in f$ e $(a, c) \in f$ impliquem $b = c$.[a]

[a] Na forma negativa, uma relação f não é uma função se existem a, b, c com $(a, b) \in f$ e $(a, c) \in f$ e $b \neq c$

Definição

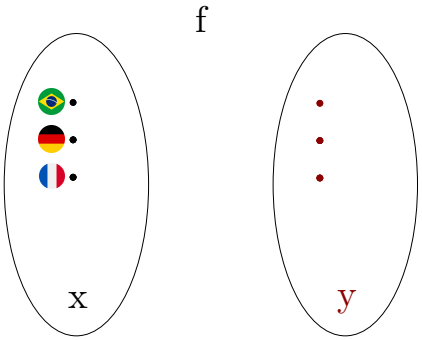
Relação: Uma relação é um conjunto de pares ordenados.

Note que os elementos da relação **nem precisam ser numéricos**!

Você consegue pensar em uma função não numérica?

1 - Aprendizagem

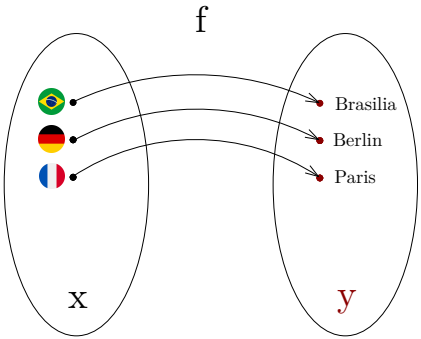
Uma função não numérica



Imagine a função $f(\text{país}) = capital$. Ela leva valores de x_i do domínio das *strings* para o contra-domínio y_i também das strings.

1 - Aprendizagem

Uma função não numérica



Note também que **não existe uma fórmula matemática que define a função**.

Definição

Aprendizado: Dado um conjunto de treinamento de N pares de exemplos de entrada e saída:

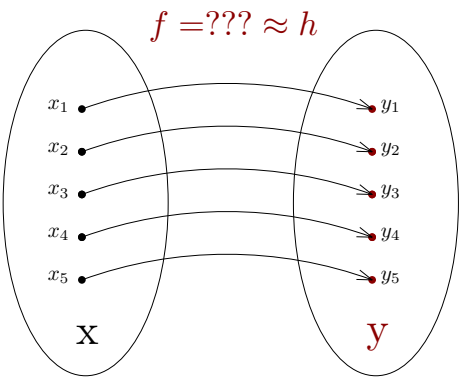
$$(x_1, y_1), (x_2, y_2), ..., (x_N, y_N)$$

em que cada y_i foi gerado por uma função desconhecida $f(x_i) = y_i$, o aprendizado busca descobrir uma função h que se aproxime de f.

No aprendizado, temos então os dois conjuntos, e devemos encontrar a função que os relaciona.

1 - Aprendizagem

Retomando a definição



Não conhecemos a função f, precisamos **aprender** uma estimativa para a mesma (h).

1 - Aprendizagem

Retomando a definição

Conclusão

Para haver o aprendizado, precisamos fornecer tanto o conjunto **X** quanto o conjunto **Y**, de forma que algum algoritmo deve aprender a função que os relaciona.

Exemplos:

1 - Aprendizagem

Retomando a definição

Conclusão

Para haver o aprendizado, precisamos fornecer tanto o conjunto **X** quanto o conjunto **Y**, de forma que algum algoritmo deve aprender a função que os relaciona.

Exemplos:

1. $\begin{cases} X : \text{Metragem quadrada de apartamentos de uma cidade.} \\ Y : \text{Preço de venda dos apartamentos.} \end{cases}$

1 - Aprendizagem

Retomando a definição

Conclusão

Para haver o aprendizado, precisamos fornecer tanto o conjunto **X** quanto o conjunto **Y**, de forma que algum algoritmo deve aprender a função que os relaciona.

Exemplos:

1. $\begin{cases} X : \text{Metragem quadrada de apartamentos de uma cidade.} \\ Y : \text{Preço de venda dos apartamentos.} \end{cases}$
2. $\begin{cases} X : \text{Idade de pessoas e se elas fumam ou não.} \\ Y : \text{Se a pessoa têm câncer ou não.} \end{cases}$

1 - Aprendizagem

Retomando a definição

Conclusão

Para haver o aprendizado, precisamos fornecer tanto o conjunto **X** quanto o conjunto **Y**, de forma que algum algoritmo deve aprender a função que os relaciona.

Exemplos:

1. $\begin{cases} X : \text{Metragem quadrada de apartamentos de uma cidade.} \\ Y : \text{Preço de venda dos apartamentos.} \end{cases}$
2. $\begin{cases} X : \text{Idade de pessoas e se elas fumam ou não.} \\ Y : \text{Se a pessoa têm câncer ou não.} \end{cases}$
3. $\begin{cases} X : \text{Matrizes de pixels (Figuras) de animais.} \\ Y : \text{Classificação do animal (mamífero, inseto, etc...)} \end{cases}$

1 - Aprendizagem

Retomando a definição

Conclusão

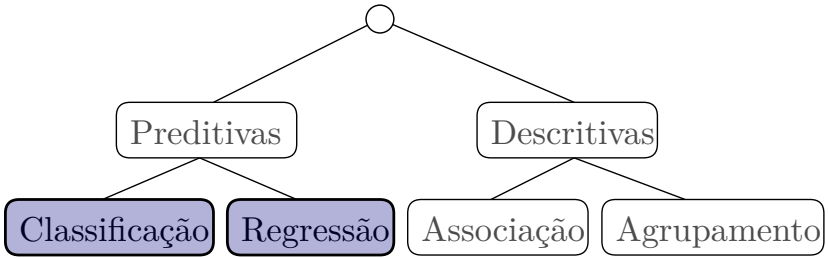
Para haver o aprendizado, precisamos fornecer tanto o conjunto **X** quanto o conjunto **Y**, de forma que algum algoritmo deve aprender a função que os relaciona.

Exemplos:

1. $\begin{cases} X : \text{Metragem quadrada de apartamentos de uma cidade.} \\ Y : \text{Preço de venda dos apartamentos.} \end{cases}$
2. $\begin{cases} X : \text{Idade de pessoas e se elas fumam ou não.} \\ Y : \text{Se a pessoa têm câncer ou não.} \end{cases}$
3. $\begin{cases} X : \text{Matrizes de pixels (Figuras) de animais.} \\ Y : \text{Classificação do animal (mamífero, inseto, etc...)} \end{cases}$
4. $\begin{cases} X : \text{Matrizes de pixels (Figuras) de tomografias.} \\ Y : \text{Se a pessoa têm câncer ou não.} \end{cases}$

Definição

Regressão/Classificação: Quando os valores de **Y** são numéricos, o problema de aprendizagem se chama **regressão**. Já quando os valores de **Y** são classes o problema se chama **classificação**.



Classifique cada um dos exemplos abaixo como uma tarefa de aprendizado do tipo **regressão** ou **classificação**:

1. $\begin{cases} X : \text{Metragem quadrada de apartamentos de uma cidade.} \\ Y : \text{Preço de venda dos apartamentos.} \end{cases}$
2. $\begin{cases} X : \text{Idade de pessoas e se elas fumam ou não.} \\ Y : \text{Se a pessoa têm câncer ou não.} \end{cases}$
3. $\begin{cases} X : \text{Matrizes de pixels (Figuras) de animais.} \\ Y : \text{Classificação do animal (mamífero, inseto, etc...)} \end{cases}$
4. $\begin{cases} X : \text{Matrizes de pixels (Figuras) de tomografias.} \\ Y : \text{Se a pessoa tem câncer ou não.} \end{cases}$

1 - Aprendizagem

Retomando a definição

Classifique cada um dos exemplos abaixo como uma tarefa de aprendizado do tipo **regressão** ou **classificação**:

1. $\begin{cases} X : \text{Metragem quadrada de apartamentos de uma cidade.} \\ Y : \text{Preço de venda dos apartamentos.} \end{cases}$ REGRESSÃO
2. $\begin{cases} X : \text{Idade de pessoas e se elas fumam ou não.} \\ Y : \text{Se a pessoa têm câncer ou não.} \end{cases}$
3. $\begin{cases} X : \text{Matrizes de pixels (Figuras) de animais.} \\ Y : \text{Classificação do animal (mamífero, inseto, etc...)} \end{cases}$
4. $\begin{cases} X : \text{Matrizes de pixels (Figuras) de tomografias.} \\ Y : \text{Se a pessoa tem câncer ou não.} \end{cases}$

1 - Aprendizagem

Retomando a definição

Classifique cada um dos exemplos abaixo como uma tarefa de aprendizado do tipo **regressão** ou **classificação**:

1. $\begin{cases} X : \text{Metragem quadrada de apartamentos de uma cidade.} \\ Y : \text{Preço de venda dos apartamentos.} \end{cases}$ REGRESSÃO
2. $\begin{cases} X : \text{Idade de pessoas e se elas fumam ou não.} \\ Y : \text{Se a pessoa têm câncer ou não.} \end{cases}$ CLASSIFICAÇÃO
3. $\begin{cases} X : \text{Matrizes de pixels (Figuras) de animais.} \\ Y : \text{Classificação do animal (mamífero, inseto, etc...)} \end{cases}$
4. $\begin{cases} X : \text{Matrizes de pixels (Figuras) de tomografias.} \\ Y : \text{Se a pessoa tem câncer ou não.} \end{cases}$

1 - Aprendizagem

Retomando a definição

Classifique cada um dos exemplos abaixo como uma tarefa de aprendizado do tipo **regressão** ou **classificação**:

1. $\begin{cases} X : \text{Metragem quadrada de apartamentos de uma cidade.} \\ Y : \text{Preço de venda dos apartamentos.} \end{cases}$ REGRESSÃO
2. $\begin{cases} X : \text{Idade de pessoas e se elas fumam ou não.} \\ Y : \text{Se a pessoa têm câncer ou não.} \end{cases}$ CLASSIFICAÇÃO
3. $\begin{cases} X : \text{Matrizes de pixels (Figuras) de animais.} \\ Y : \text{Classificação do animal (mamífero, inseto, etc...)} \end{cases}$ CLASSIFICAÇÃO
4. $\begin{cases} X : \text{Matrizes de pixels (Figuras) de tomografias.} \\ Y : \text{Se a pessoa tem câncer ou não.} \end{cases}$

1 - Aprendizagem

Retomando a definição

Classifique cada um dos exemplos abaixo como uma tarefa de aprendizado do tipo **regressão** ou **classificação**:

1. $\begin{cases} X : \text{Metragem quadrada de apartamentos de uma cidade.} \\ Y : \text{Preço de venda dos apartamentos.} \end{cases}$ REGRESSÃO
2. $\begin{cases} X : \text{Idade de pessoas e se elas fumam ou não.} \\ Y : \text{Se a pessoa têm câncer ou não.} \end{cases}$ CLASSIFICAÇÃO
3. $\begin{cases} X : \text{Matrizes de pixels (Figuras) de animais.} \\ Y : \text{Classificação do animal (mamífero, inseto, etc...)} \end{cases}$ CLASSIFICAÇÃO
4. $\begin{cases} X : \text{Matrizes de pixels (Figuras) de tomografias.} \\ Y : \text{Se a pessoa tem câncer ou não.} \end{cases}$ CLASSIFICAÇÃO

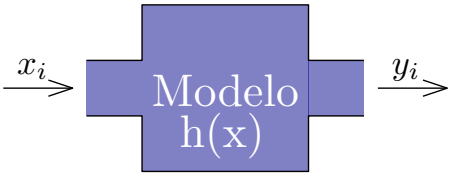
1 - Aprendizagem

O que fazemos depois de estimar a função?

E o que fazemos quando estimamos o valor da função verdadeira $f(x)$ por uma função $h(x)$?

Definição

Modelo: A função estimada h também é chamada de modelo (de regressão ou classificação). Com um modelo estimado, podemos usá-lo para prever valores de y_i desconhecidos.



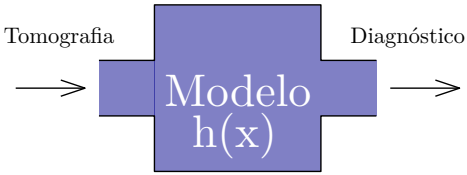
1 - Aprendizagem

O que fazemos depois de estimar a função?

E o que fazemos quando estimamos o valor da função verdadeira $f(x)$ por uma função $h(x)$?

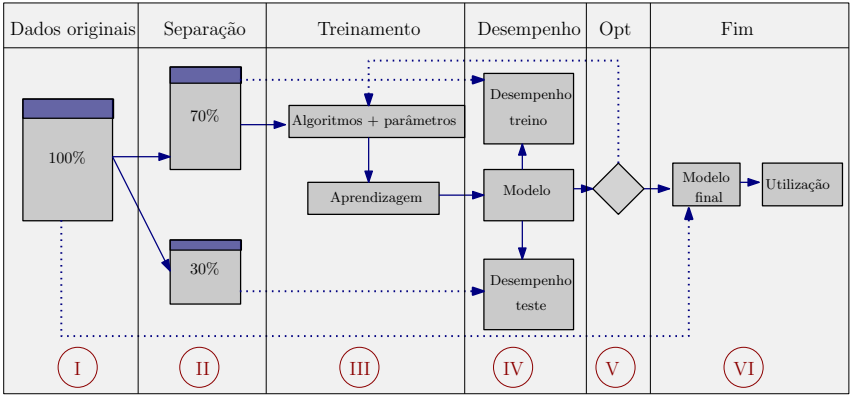
Definição

Modelo: A função estimada h também é chamada de modelo (de regressão ou classificação). Com um modelo estimado, podemos usá-lo para prever valores de y_i desconhecidos.



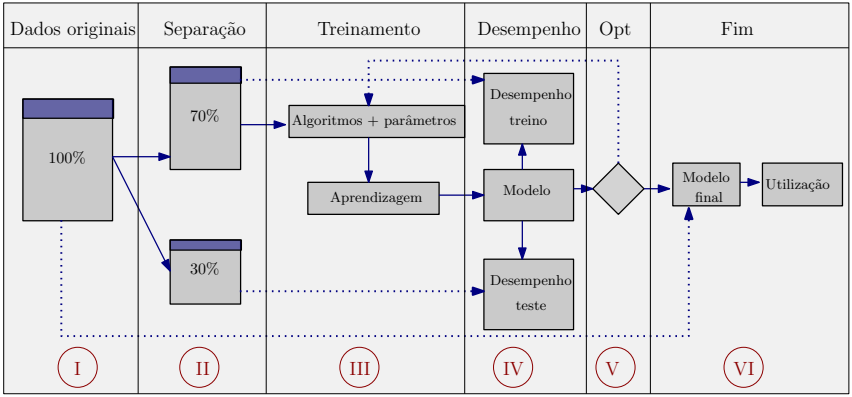
Fluxo de atividades

2 - Fluxo das atividades



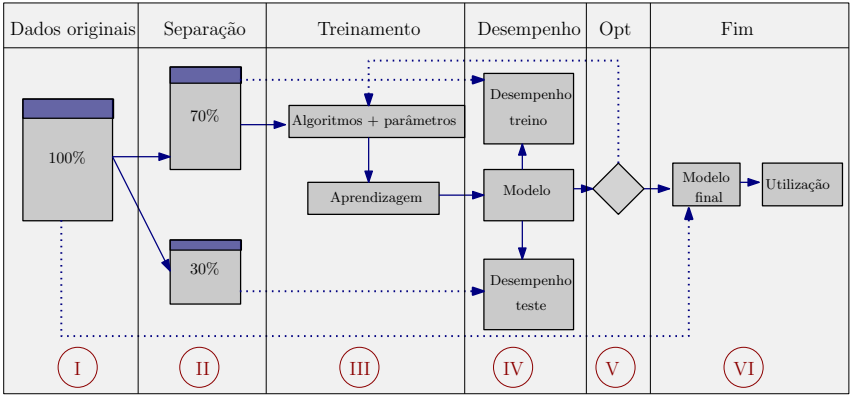
De forma geral, o processo completo para gerarmos e utilizarmos um modelo de classificação/regressão se dá como na Figura acima.

2 - Fluxo das atividades



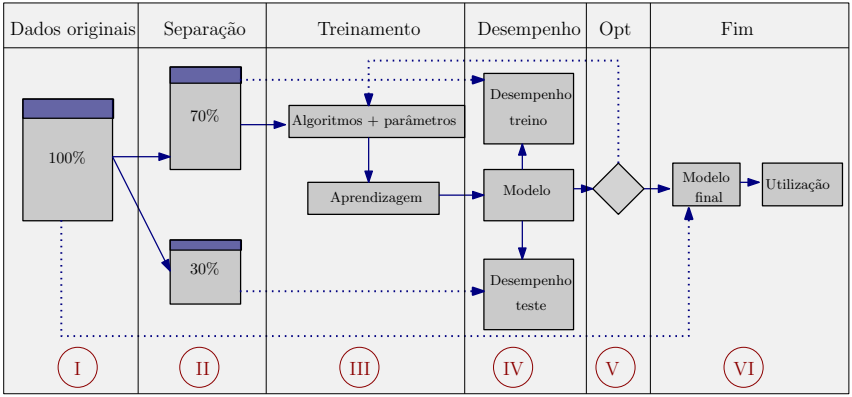
Inicialmente (I), buscamos um conjunto de dados na forma de pares entrada/saída (x_i, y_i) . Logo em seguida, utilizamos um método para realizar a separação dos dados em conjuntos de **teste** e **treino** (II).

2 - Fluxo das atividades



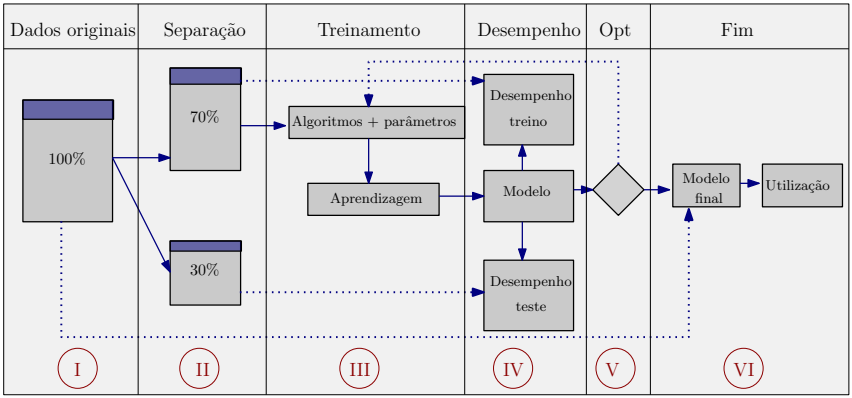
O conjunto de **treino** é utilizado para que um algoritmo de aprendizagem estime a relação entre as entradas x_i e as saídas y_i (III). Note que diferentes algoritmos podem executar essa tarefa (redes neurais, árvores de classificação, etc...).

2 - Fluxo das atividades



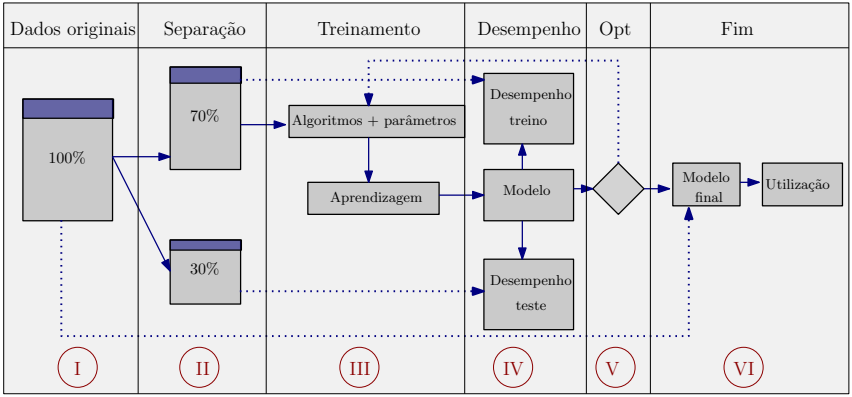
Cada algoritmo possui parâmetros diferentes, que produzirão uma função estimada h (modelo) diferente, o que sugere que um mesmo conjunto de dados pode gerar **diferentes modelos**.

2 - Fluxo das atividades



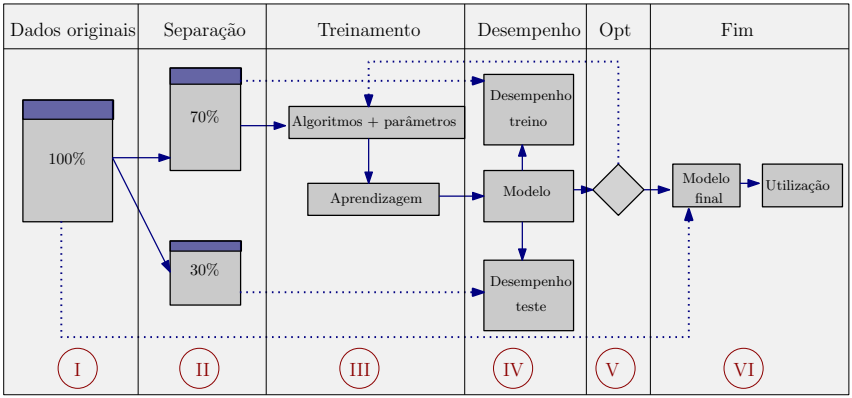
Com o modelo estimado, verificamos a acurácia/desempenho do mesmo (IV). A acurácia pode ser mensurada fornecendo conjuntos de dados em que sabemos o valor alvo (y_i), porém "escondemos" esses valores e pedimos para o modelo realizar a estimativa.

2 - Fluxo das atividades



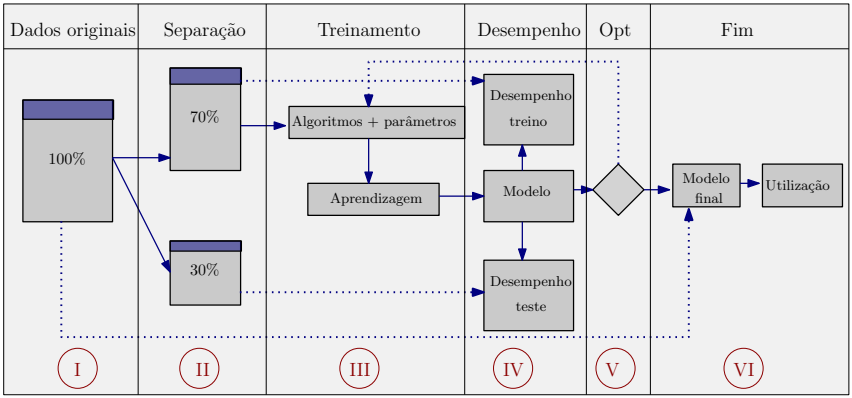
Em seguida, podemos calcular uma taxa de acertos/erros, comparando as estimativas realizadas e os valores ocorridos.

2 - Fluxo das atividades



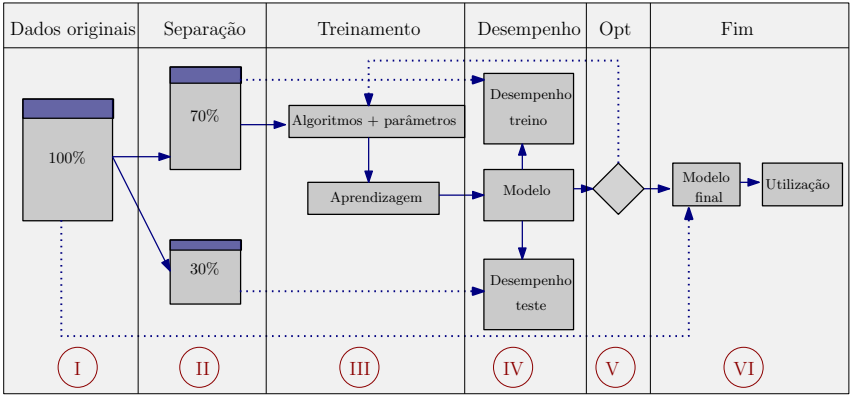
Fazemos isso usando tanto o conjunto de teste quanto o conjunto de treino (IV). Dessa forma, decidimos se o modelo tem um desempenho satisfatório ou não (evitando o *overfitting* do modelo aos dados).

2 - Fluxo das atividades



Se o modelo não estiver performando bem (V), voltamos à etapa de treinamento (III) e alteramos os parâmetros do algoritmo escolhido, ou mesmo o próprio algoritmo de aprendizado. Essa etapa é chamada de *otimização dos hiperparâmetros*.

2 - Fluxo das atividades



Quando encontramos o conjunto de parâmetros que gere um modelo satisfatório, realizamos o treinamento usando **todo o conjunto de dados**. Esse modelo pode ser utilizado para realizar as previsões/classificações.

2 - Fluxo das atividades

Para que todo esse processo ocorra, precisamos então conseguir responder às seguintes questões:

1. Qual algoritmo de aprendizado usar, e quais são os seus parâmetros?

2. Como os dados originais serão separados em treino e teste?

3. Qual medida de desempenho será usada para avaliar a performance do modelo?

2 - Fluxo das atividades

Para que todo esse processo ocorra, precisamos então conseguir responder às seguintes questões:

1. Qual algoritmo de aprendizado usar, e quais são os seus parâmetros?
   1.1 Algoritmo de Hunt para indução de árvores de decisão.
2. Como os dados originais serão separados em treino e teste?

3. Qual medida de desempenho será usada para avaliar a performance do modelo?

2 - Fluxo das atividades

Para que todo esse processo ocorra, precisamos então conseguir responder às seguintes questões:

1. Qual algoritmo de aprendizado usar, e quais são os seus parâmetros?
   1.1 Algoritmo de Hunt para indução de árvores de decisão.
2. Como os dados originais serão separados em treino e teste?
   2.1 *Holdout* e *cross-validation*.
3. Qual medida de desempenho será usada para avaliar a performance do modelo?

2 - Fluxo das atividades

Para que todo esse processo ocorra, precisamos então conseguir responder às seguintes questões:

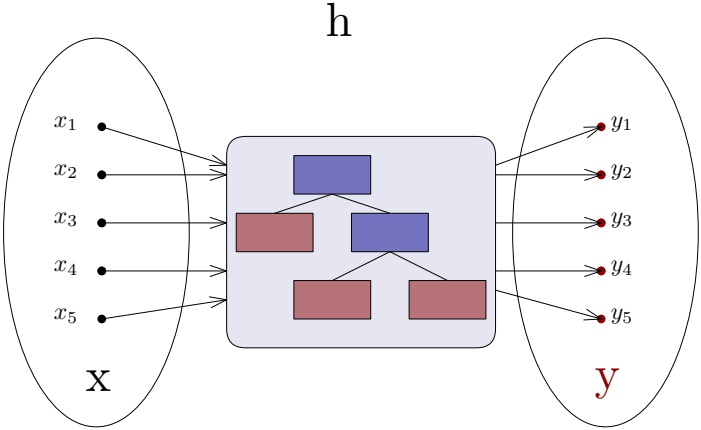
1. Qual algoritmo de aprendizado usar, e quais são os seus parâmetros?
   1.1 Algoritmo de Hunt para indução de árvores de decisão.
2. Como os dados originais serão separados em treino e teste?
   2.1 *Holdout* e *cross-validation*.
3. Qual medida de desempenho será usada para avaliar a performance do modelo?
   3.1 Matriz de confusão/ precisão.

Árvores de decisão

3 - Árvores de decisão

Representando funções por árvores de decisão

A árvore de decisão (ou classificação) é uma **forma de representar a função estimada** h (lembre que uma função não precisa ser representada por uma fórmula numérica).



3 - Árvores de decisão

Representando funções por árvores de decisão

Considere um banco de dados referente aos clientes que chegam para comer em um restaurante que serve 3 tipos de comida, italiana, tailandesa e japonesa. Os atributos (colunas) desse banco de dados são os seguintes:

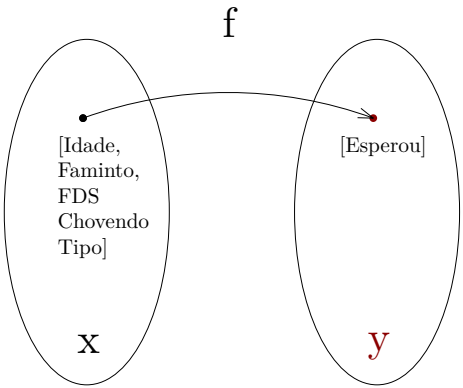
- **Idade (Numérico):** Inteiro representando a idade do cliente.
- **Faminto (Booleano):** Se o cliente chegou ao restaurante faminto ou não.
- **Fim de semana (Booleano):** Se é fim de semana ou não.
- **Chovendo (Booleano):** Se está chovendo ao não.
- **Tipo (Italiano/Tailandesa/Japonesa):** O tipo de comida que o cliente quer comer.
- **Esperou (Booleano):** Se o cliente teve que esperar para ser atendido ou não.

3 - Árvores de decisão

Representação de funções

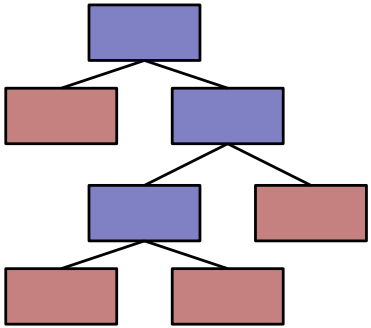
Podemos tentar descobrir se existe uma função que relaciona as características do cliente com o atributo **Esperou**.

Idade	Faminto	FDS	Chovendo	Tipo	Esperou
20	SIM	SIM	NÃO	TAI	SIM
35	NÃO	NÃO	SIM	ITA	NÃO
20	SIM	NÃO	NÃO	JAP	SIM
64	SIM	NÃO	NÃO	TAI	SIM
64	NÃO	NÃO	NÃO	ITA	NÃO



3 - Árvores de decisão

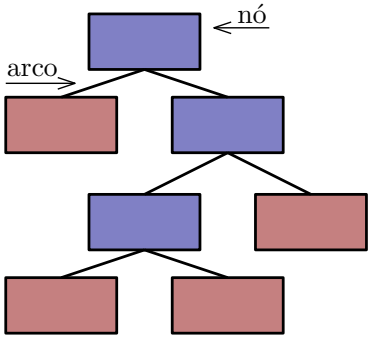
Representação de funções



Uma árvore de decisão é composta por nós e arcos, formando um grafo acíclico direcionado.

3 - Árvores de decisão

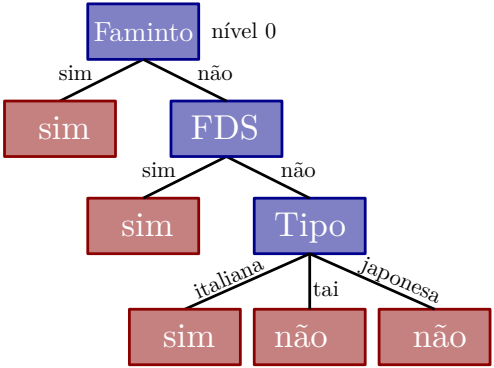
Representação de funções



Uma árvore de decisão é composta por nós e arcos, formando um grafo acíclico direcionado.

3 - Árvores de decisão

Representação de funções

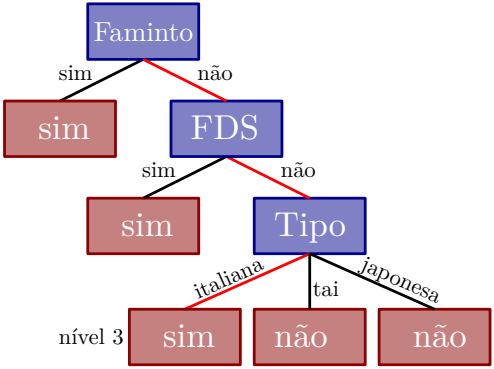


O nível de um nó na árvore é dado pela distancia [1] entre esse nó e a raiz (primeiro nó).

[1] A distância entre 2 nós é dada pelo número de arcos entre os nós

3 - Árvores de decisão

Representação de funções

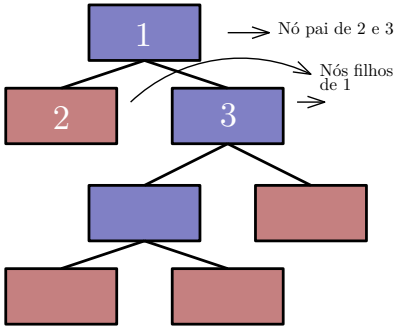


O nível de um nó na árvore é dado pela distancia [2] entre esse nó e a raiz (primeiro nó).

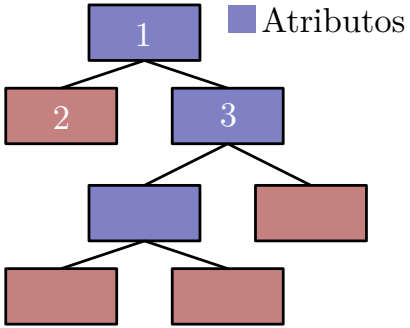
[2] A distância entre 2 nós é dada pelo número de arcos entre os nós

3 - Árvores de decisão

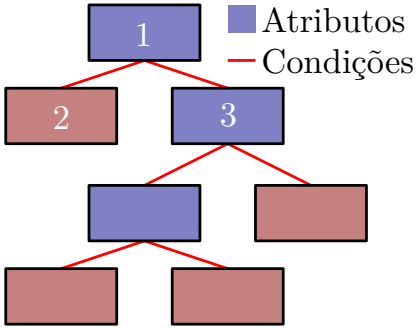
Representação de funções



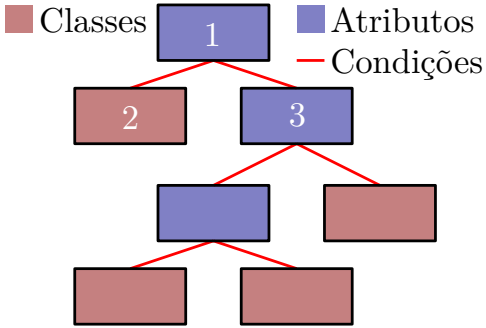
Um nó que possui sub-nós em um nível abaixo do seu, é chamado de nó-pai dos sub-nós, e os sub-nós por sua vez são chamados de nós filhos.



Se um nó da árvore é um nó não-folha, ou seja, possui filhos, então ele representa um **atributo** do conjunto (uma das colunas no banco de dados).



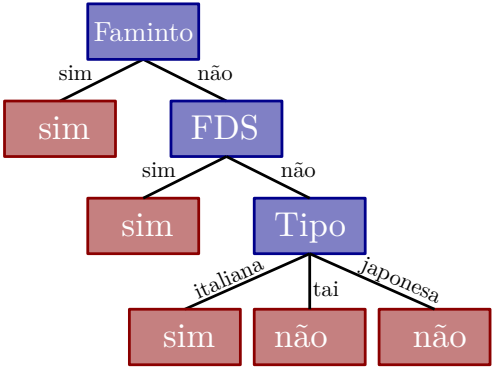
Os arcos da árvore representam **condições** que devem ser aplicadas aos nós.



As classes são mostradas nos nós folhas.

3 - Árvores de decisão

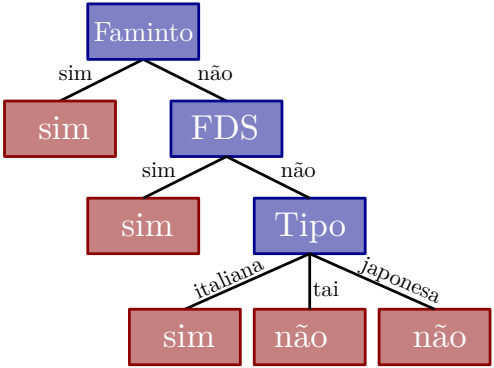
Representação de funções



A árvore acima poderia ser usada para classificar clientes do restaurante em uma das duas classes (SIM/NÃO) considerando o atributo *Espera*.

3 - Árvores de decisão

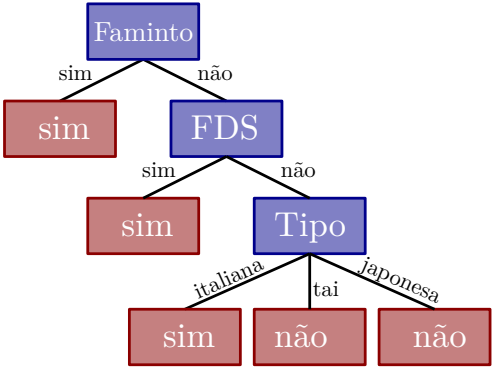
Representação de funções



Para usarmos a árvore para classificar um registro desconhecido X, seguimos o caminho partindo do nó raiz, aplicando todas as condições até chegar a um nó folha. A classificação de X é a classificação mostrada na folha.

3 - Árvores de decisão

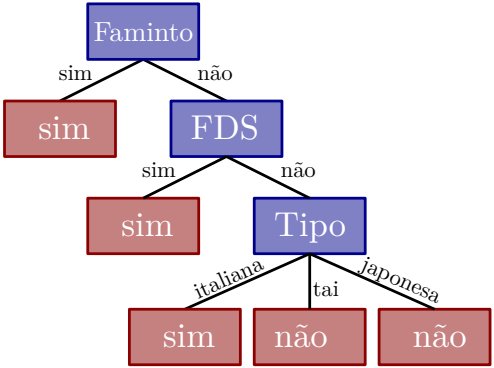
Representação de funções



Supondo que um novo cliente chega ao restaurante com o seguinte vetor de características X:

3 - Árvores de decisão

Representação de funções



X = [Idade,Faminto,FimDeSemana,Chovendo,Tipo,Esperou]
X = [45,NAO,SIM,NAO,JAPONESA,???]

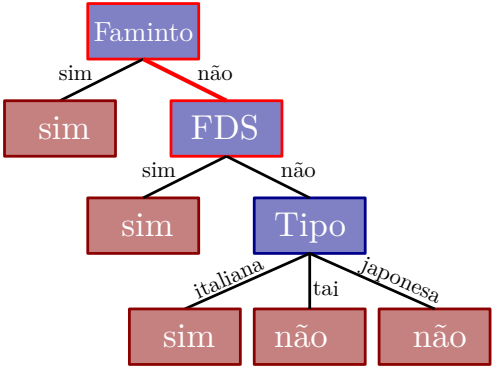
A primeira condição (na árvore) se refere ao atributo Faminto. De forma que olhamos esse atributo no vetor X, seguindo o caminho após aplicar a condição da árvore.

X = [Idade,Faminto,FimDeSemana,Chovendo,Tipo,Esperou]

X = [45,NAO,SIM,NAO,JAPONESA]

3 - Árvores de decisão

Representação de funções



Em X, Faminto = NÃO, de forma que seguimos o caminho da direita na árvore.

X = [Idade,Faminto,FimDeSemana,Chovendo,Tipo,Esperou]

X = [45,NAO,SIM,NAO,JAPONESA]

3 - Árvores de decisão

Representação de funções

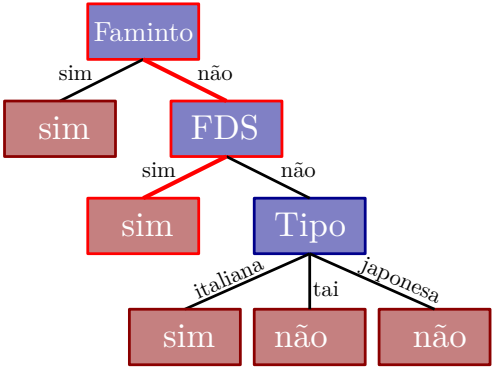
Em seguida verificamos o atributo FimDeSemana. Em X FimDeSemana = SIM, de forma que seguimos o caminho da esquerda na árvore.

X = [Idade,Faminto,FimDeSemana,Chovendo,Tipo,Esperou]

X = [45,NAO,SIM,NAO,JAPONESA]

3 - Árvores de decisão

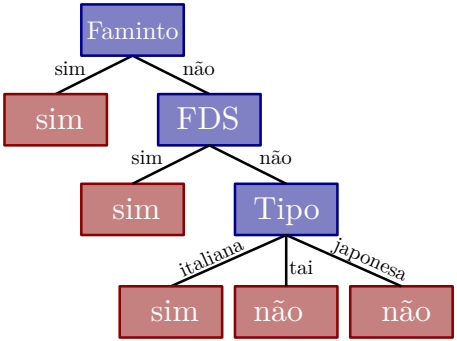
Representação de funções



No fim do processo classificamos o cliente com vetor de atributos com Esperou = SIM.

3 - Árvores de decisão

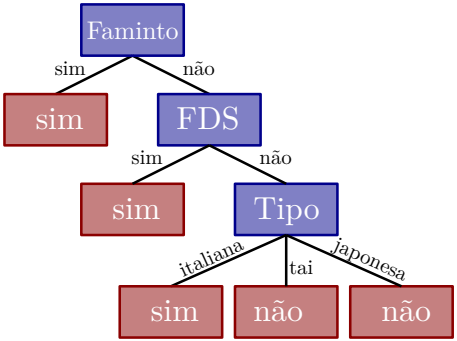
Representação de funções



Classifique os seguintes vetores X:
[Idade,Faminto,FimDeSemana,Chovendo,Tipo,Esperou]
X_1 = [45, NAO, SIM, NAO, JAPONESA,??]
X_2 = [102,SIM, SIM, NAO, JAPONESA,??]
X_3 = [25, NAO, NAO, NAO, TAILANDESA,??]

3 - Árvores de decisão

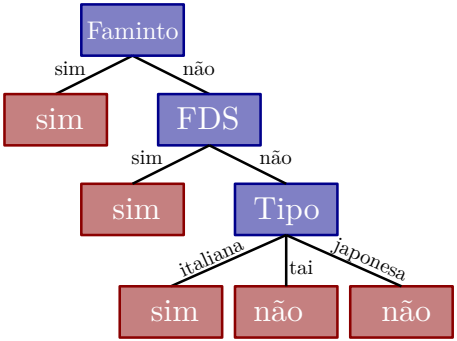
Representação de funções



Classifique os seguintes vetores X:
[Idade,Faminto,FimDeSemana,Chovendo,Tipo,Esperou]
X_1 = [45, NAO, SIM, NAO, JAPONESA,??] SIM
X_2 = [102,SIM, SIM, NAO, JAPONESA,??]
X_3 = [25, NAO, NAO, NAO, TAILANDESA,??]

3 - Árvores de decisão

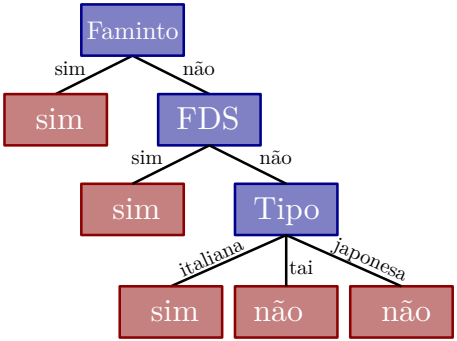
Representação de funções



Classifique os seguintes vetores X:
[Idade,Faminto,FimDeSemana,Chovendo,Tipo,Esperou]
X_1 = [45, NAO, SIM, NAO, JAPONESA,??] SIM
X_2 = [102,SIM, SIM, NAO, JAPONESA,??] SIM
X_3 = [25, NAO, NAO, NAO, TAILANDESA,??]

3 - Árvores de decisão

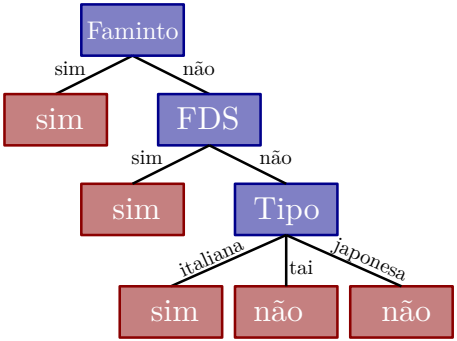
Representação de funções



Classifique os seguintes vetores X:
[Idade,Faminto,FimDeSemana,Chovendo,Tipo,Esperou]
X_1 = [45, NAO, SIM, NAO, JAPONESA,??] SIM
X_2 = [102,SIM, SIM, NAO, JAPONESA,??] SIM
X_3 = [25, NAO, NAO, NAO, TAILANDESA,??] NÃO

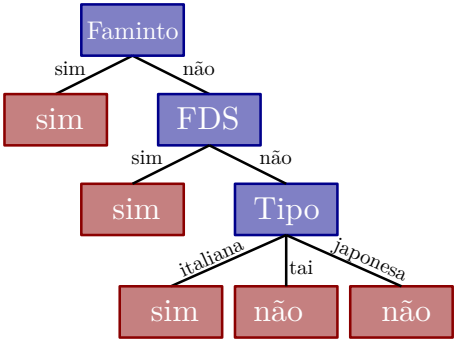
3 - Árvores de decisão

Representação de funções



Podemos usar as árvores para inferir diversas características do sistema. Considere as seguintes alterações que os donos do restaurante estão pensando em fazer para diminuir a espera dos clientes:

1. Criar um atendimento prioritário para idosos.
2. Contratar um chefe novo para preparar as comidas tailandesas.



O que você sugeriria que o restaurante fizesse para reduzir a espera dos clientes (com base no modelo)?

1. Nos fins de semana, contratar mais funcionários.
2. Nos dias de semana contratar um novo chefe para a cozinha italiana.
3. Servir um aperitivo para os clientes logo na chegada.

CONCLUSÕES

CONCLUSÕES

1. A árvore mostra quais atributos mantém uma maior relação com o atributo alvo, ou seja, quais variáveis melhor descrevem (ou influenciam) na variável *target*.

CONCLUSÕES

1. A árvore mostra quais atributos mantém uma maior relação com o atributo alvo, ou seja, quais variáveis melhor descrevem (ou influenciam) na variável *target*.
2. Atributos que **não aparecem** na árvore **não são importantes para a classificação**.

CONCLUSÕES

1. A árvore mostra quais atributos mantém uma maior relação com o atributo alvo, ou seja, quais variáveis melhor descrevem (ou influenciam) na variável *target*.
2. Atributos que **não aparecem** na árvore **não são importantes para a classificação**.
3. O **nível** ao qual o atributo aparece na árvore indica a sua importância na classificação. No caso acima, Faminto (nível 0) é mais importante do que FDS (nível 1) que é mais importante do que TIPO (nível 2).

Atenção

Acabamos de ver como podemos representar uma função estimada por meio de uma árvore de classificação, não vimos **como a árvore é gerada!**

Indução de árvores de decisão (algoritmo de Hunt)

4 - Indução de árvores de decisão

O algoritmo de Hunt

4 - Indução de árvores de decisão

O algoritmo de Hunt

- As árvores de decisão devem ser geradas a partir dos dados. Existem diversos algoritmos que geram árvores de decisão (C4.5, ID3 e CART). O primeiro algoritmo criado foi o **algoritmo de Hunt**, sendo que a sua lógica é usada nos outros, portanto entenderemos o seu comportamento.

4 - Indução de árvores de decisão

O algoritmo de Hunt

- As árvores de decisão devem ser geradas a partir dos dados. Existem diversos algoritmos que geram árvores de decisão (C4.5, ID3 e CART). O primeiro algoritmo criado foi o **algoritmo de Hunt**, sendo que a sua lógica é usada nos outros, portanto entenderemos o seu comportamento.
- O algoritmo funciona de forma a dividir o banco de dados em conjuntos disjuntos de acordo com algum atributo, criando partições menores dos dados, que vão sendo novamente particionados, até um critério de parada.

4 - Indução de árvores de decisão

O algoritmo de Hunt

- As árvores de decisão devem ser geradas a partir dos dados. Existem diversos algoritmos que geram árvores de decisão (C4.5, ID3 e CART). O primeiro algoritmo criado foi o **algoritmo de Hunt**, sendo que a sua lógica é usada nos outros, portanto entenderemos o seu comportamento.
- O algoritmo funciona de forma a dividir o banco de dados em conjuntos disjuntos de acordo com algum atributo, criando partições menores dos dados, que vão sendo novamente particionados, até um critério de parada.
- Cada atributo selecionado para realizar uma partição é um **nó não folha** da árvore. Ao chegar em um nó folha, o número de registros em cada classe é contado, e a classificação é feita com a classe com maior frequência de registros.

Idade	Faminto	FDS	Chovendo	Tipo	Esperou
20	SIM	SIM	NÃO	TAI	SIM
35	NÃO	NÃO	SIM	ITA	NÃO
20	SIM	NÃO	NÃO	JAP	SIM
64	SIM	NÃO	NÃO	TAI	SIM
64	NÃO	NÃO	NÃO	ITA	NÃO

Considere o banco de dados de espera dos clientes. Podemos escolher um atributo qualquer para realizar a separação dos dados.

Idade	Faminto	FDS	Chovendo	Tipo	Esperou
20	SIM	SIM	NÃO	TAI	SIM
35	NÃO	NÃO	SIM	ITA	NÃO
20	SIM	NÃO	NÃO	JAP	SIM
64	SIM	NÃO	NÃO	TAI	SIM
64	NÃO	NÃO	NÃO	ITA	NÃO

Escolhendo o atributo **FDS**. Se separarmos o banco de dados por **FDS**, quantos registros recaem na classe SIM e quantos na classe NÃO (considerando o atributo *target* Esperou)?

4 - Indução de árvores de decisão

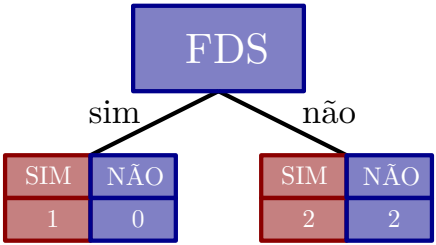
O algoritmo de Hunt

Idade	Faminto	FDS	Chovendo	Tipo	Esperou
20	SIM	SIM	NÃO	TAI	SIM
35	NÃO	NÃO	SIM	ITA	NÃO
20	SIM	NÃO	NÃO	JAP	SIM
64	SIM	NÃO	NÃO	TAI	SIM
64	NÃO	NÃO	NÃO	ITA	NÃO

Temos que:

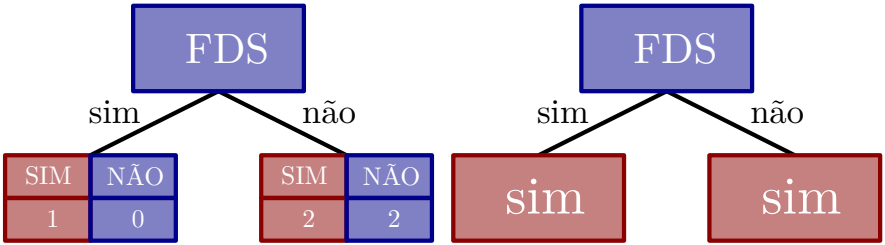
$$\begin{cases} \text{FDS} = \text{SIM: 1 SIM, 0 NÃO} \\ \text{FDS} = \text{NÃO: 2 SIM, 2 NÃO} \end{cases}$$

Representando a separação dos dados em uma árvore, temos o seguinte:



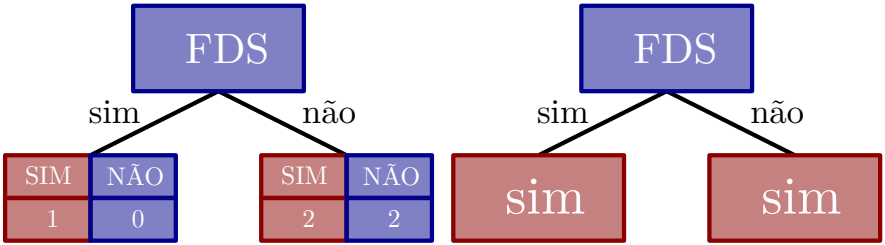
A classificação dos nós é dada pela **classe de maior frequência** nos nós folha.

Representando a separação dos dados em uma árvore, temos o seguinte:



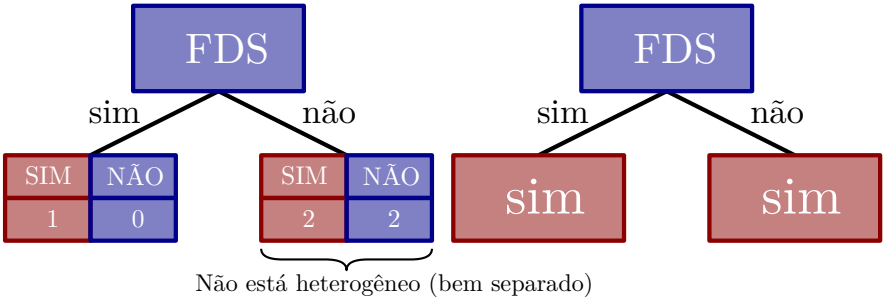
Como mostrado acima, o nó da esquerda = SIM e o nó da direita poderia ser tanto SIM quanto NÃO.

Representando a separação dos dados em uma árvore, temos o seguinte:



Note que podemos ter uma ideia da **qualidade** da separação do banco de dados pelo número de elementos em cada nó folha (classes). Quanto mais **heterogênea** a separação no nó folha, melhor foi a escolha do atributo separador.

Representando a separação dos dados em uma árvore, temos o seguinte:



Note que podemos ter uma ideia da **qualidade** da separação do banco de dados pelo número de elementos em cada nó folha (classes). Quanto mais **heterogênea** a separação no nó folha, melhor foi a escolha do atributo separador.

4 - Indução de árvores de decisão

O algoritmo de Hunt

Idade	Faminto	FDS	Chovendo	Tipo	Esperou
20	SIM	SIM	NÃO	TAI	SIM
35	NÃO	NÃO	SIM	ITA	NÃO
20	SIM	NÃO	NÃO	JAP	SIM
64	SIM	NÃO	NÃO	TAI	SIM
64	NÃO	NÃO	NÃO	ITA	NÃO

Existe algum atributo que separa os dados de forma mais heterogênea?

Idade	Faminto	FDS	Chovendo	Tipo	Esperou
20	SIM	SIM	NÃO	TAI	SIM
35	NÃO	NÃO	SIM	ITA	NÃO
20	SIM	NÃO	NÃO	JAP	SIM
64	SIM	NÃO	NÃO	TAI	SIM
64	NÃO	NÃO	NÃO	ITA	NÃO

Existe algum atributo que separa os dados de forma mais heterogênea?
Considerando o atributo **Faminto**, temos a seguinte separação:

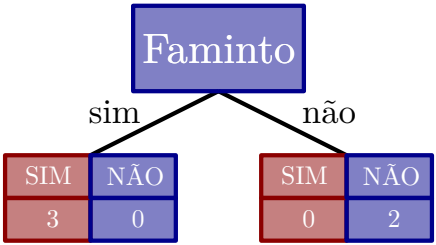
4 - Indução de árvores de decisão

O algoritmo de Hunt

Idade	Faminto	FDS	Chovendo	Tipo	Esperou
20	SIM	SIM	NÃO	TAI	SIM
35	NÃO	NÃO	SIM	ITA	NÃO
20	SIM	NÃO	NÃO	JAP	SIM
64	SIM	NÃO	NÃO	TAI	SIM
64	NÃO	NÃO	NÃO	ITA	NÃO

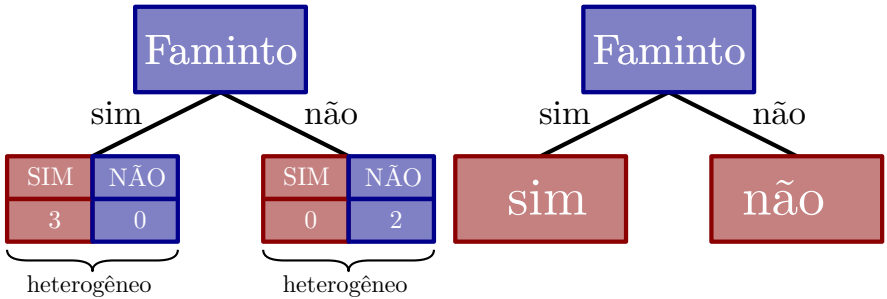
$$\begin{cases} \text{Faminto} = \text{SIM: 3 SIM, 0 NÃO} \\ \text{Faminto} = \text{NÃO: 0 SIM, 2 NÃO} \end{cases}$$

Representando o separação dos dados em uma árvore, temos o seguinte:



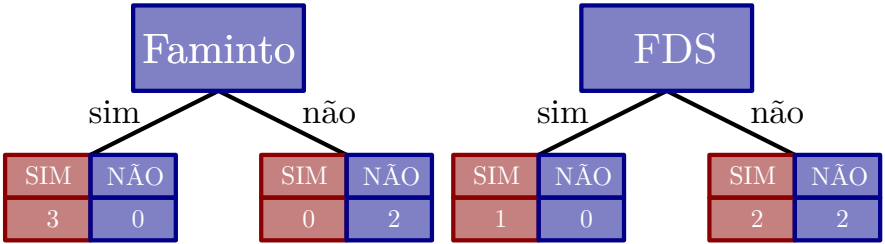
Note que o atributo **Faminto** separou os dados de forma mais heterogênea do que o atributo **FDS**.

Representando o separação dos dados em uma árvore, temos o seguinte:



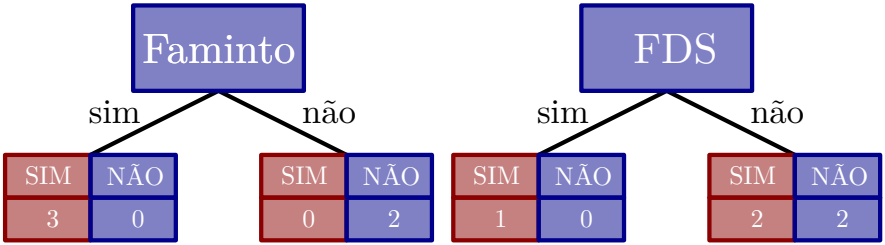
Note o atributo **Faminto** separou os dados de forma mais heterogênea do que o atributo **FDS**.

Comparando as separações dos dados com os dois atributos:



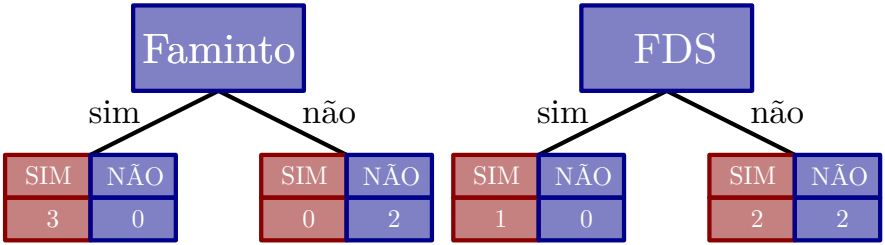
1. Portanto escolhendo o atributo Faminto conseguimos separar melhor os dados, criando dessa forma um **modelo mais preciso** para classificação.

Comparando as separações dos dados com os dois atributos:



1. Portanto escolhendo o atributo Faminto conseguimos separar melhor os dados, criando dessa forma um **modelo mais preciso** para classificação.
2. Se a separação de um nó ainda é muito homogênea, podemos **escolher um novo atributo** com os dados do nó e realizar mais separações.

Comparando as separações dos dados com os dois atributos:



1. Portanto escolhendo o atributo Faminto conseguimos separar melhor os dados, criando dessa forma um **modelo mais preciso** para classificação.
2. Se a separação de um nó ainda é muito homogênea, podemos **escolher um novo atributo** com os dados do nó e realizar mais separações.
3. O **número de nós**, o **tamanho da árvore** e a medida usada para **verificar a qualidade dos nós** são parâmetros dos algoritmos geradores de árvores de classificação.

Conclusão

O algoritmo de Hunt separa os dados sucessivamente escolhendo o **melhor** atributo separador em cada nó. Existem medidas quantificadoras da qualidade da separação dos dados em um nó (que medem a heterogeneidade). Uma delas é **Entropia**.

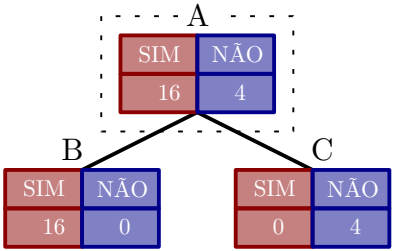
Entropia

5 - Entropia

Mensurando a qualidade da separação nos nós

Considerando um banco de dados de classificação binária (como o banco de espera dos clientes, em que as classes são SIM e NÃO), um atributo para separação perfeita dos dados implica que os nós folha após a separação **manterão somente dados referentes a uma classe**.

Considere um banco de dados com 20 valores, 16 da classe **SIM** e 4 da classe **NÃO**. Se houver um atributo que separe os dados como na Figura, ele é ótimo.



5 - Entropia

Mensurando a qualidade da separação nos nós

Podemos mensurar a qualidade da separação em um nó usando a **Entropia**. A entropia é uma medida de incerteza de uma variável aleatória (estatística), a aquisição de informação reduz a entropia.

5 - Entropia

Mensurando a qualidade da separação nos nós

Podemos mensurar a qualidade da separação em um nó usando a **Entropia**. A entropia é uma medida de incerteza de uma variável aleatória (estatística), a aquisição de informação reduz a entropia.
O cálculo da entropia *(E)* para uma variável aleatória (V) é dada por:

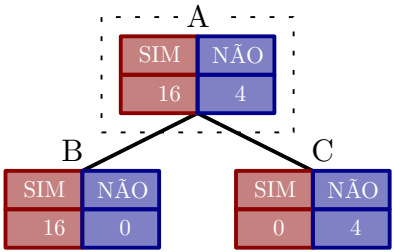
$$E(V) = -\sum_{k} P(v_k).\log_2 P(v_k)$$

em que: $\begin{cases} k : \text{Número de classes em um nó.} \\ P(v_k) : \text{Probabilidade (ou proporção) da classe } v_k \text{ no nó.} \end{cases}$

5 - Entropia

Mensurando a qualidade da separação nos nós

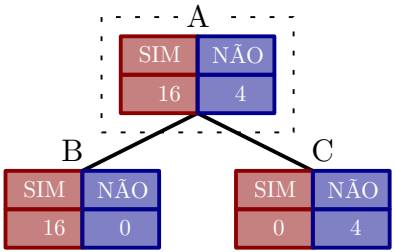
Considerando o nó A:



5 - Entropia

Mensurando a qualidade da separação nos nós

Considerando o nó A:



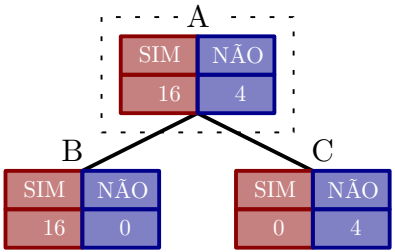
temos que:

$$\begin{cases} k = 2 \text{ (duas classes, SIM e NÃO).} \\ P(SIM) = 16/20 \text{ (proporção de SIM no nó)} \\ P(NAO) = 4/20 \text{ (proporção de NAO no nó)} \end{cases}$$

5 - Entropia

Mensurando a qualidade da separação nos nós

Considerando o nó A:



temos que:

$$\begin{cases} k = 2 \text{ (duas classes, SIM e NÃO).} \\ P(SIM) = 16/20 \text{ (proporção de SIM no nó)} \\ P(NAO) = 4/20 \text{ (proporção de NAO no nó)} \end{cases}$$

$$E(V) = -\sum_k P(v_k).\log_2 P(v_k)$$

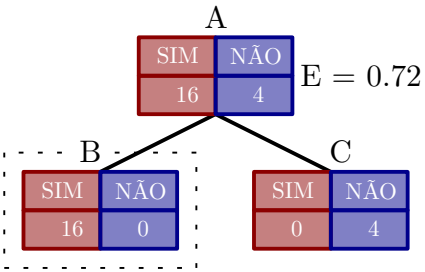
$$E(A) = -\frac{16}{20}\log_2\frac{16}{20} - \frac{4}{20}\log_2\frac{4}{20}$$

$$E(A) \approx 0.72$$

5 - Entropia

Mensurando a qualidade da separação nos nós

Considerando o nó B:



temos que:

$$\begin{cases} k = 2 \text{ (duas classes, SIM e NÃO).} \\ P(SIM) = 16/16 \text{ (proporção de SIM no nó)} \\ P(NAO) = 0/16 \text{ (proporção de NAO no nó)} \end{cases}$$

$$E(V) = -\sum_k P(v_k).\log_2 P(v_k)$$

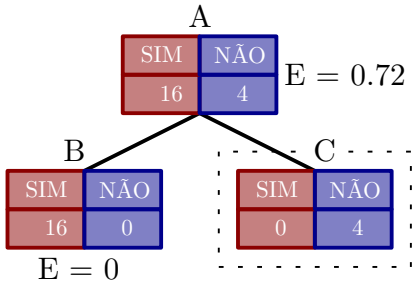
$$E(B) = -\frac{16}{16}\log_2\frac{16}{16} - \frac{0}{16}\log_2\frac{0}{16}$$

$$E(B) = 0$$

5 - Entropia

Mensurando a qualidade da separação nos nós

Considerando o nó C:



temos que:

$$\begin{cases} k = 2 \text{ (duas classes, SIM e NÃO).} \\ P(SIM) = 0/4 \text{ (proporção de SIM no nó)} \\ P(NAO) = 4/4 \text{ (proporção de NAO no nó)} \end{cases}$$

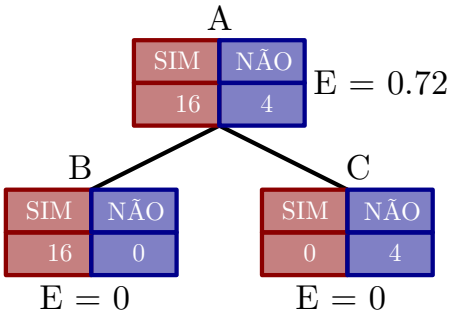
$$E(V) = -\sum_k P(v_k).\log_2 P(v_k)$$

$$E(C) = -\frac{0}{4}\log_2\frac{0}{4} - \frac{4}{4}\log_2\frac{4}{4}$$

$$E(C) = 0$$

5 - Entropia

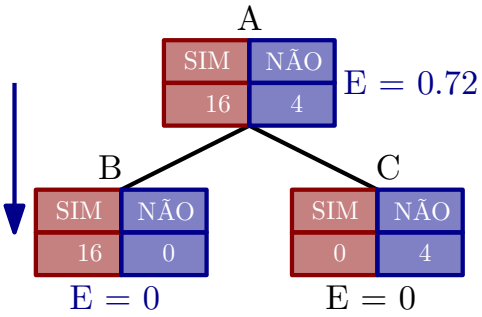
Mensurando a qualidade da separação nos nós



Assim, a **entropia** calculada em todos os nós fica como na Figura acima.

5 - Entropia

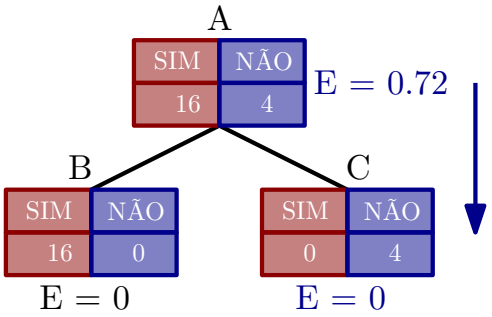
Mensurando a qualidade da separação nos nós



Note que houve uma redução na entropia considerando o nó A e o nó B (0.72 e 0), ou seja, houve um ganho de informação com a separação.

5 - Entropia

Mensurando a qualidade da separação nos nós



A mesma coisa considerando o nó A e o nó C (0.72 e 0), ou seja, houve um ganho de informação com a separação.

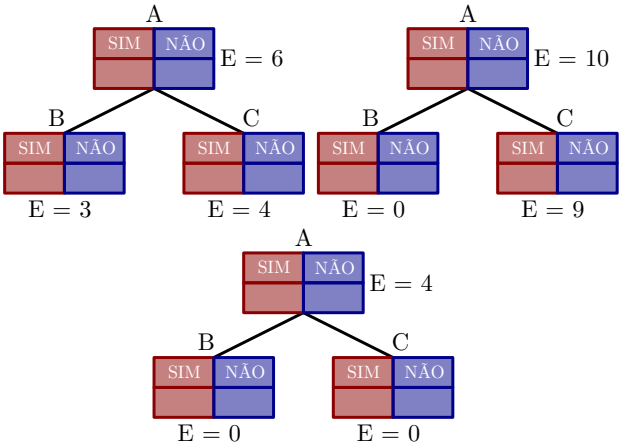
5 - Entropia

Mensurando a qualidade da separação nos nós

1. Com isso, conseguimos ver a diferença de entropia entre **nós individuais**.
2. Porém, para determinarmos se um atributo é melhor do que outro na separação, **devemos levar em consideração a separação em todos os nós filhos**.
3. Considere as separações mostradas abaixo.

5 - Entropia

Mensurando a qualidade da separação nos nós



5 - Entropia

Mensurando a qualidade da separação nos nós

O ganho de informação deve ser calculado em relação a todos os nós filhos. Usamos então a variação no ganho de informação $\Delta Info$, dada por:

$$\Delta Info = E(\text{nó pai}) - MP(E(\text{nós filhos}))$$

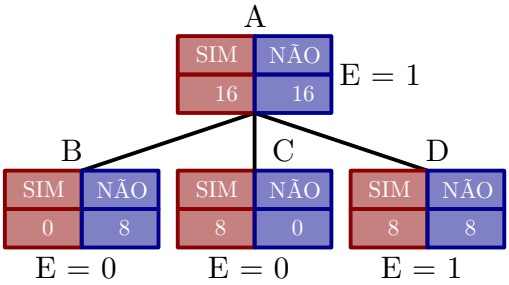
em que MP é a média ponderada pela proporção das frequências nos nós filhos em relação ao nó pai.

Quanto maior o ganho de informação ($\uparrow \Delta Info$) **melhor a separação feita pelo atributo escolhido**.

5 - Entropia

Mensurando a qualidade da separação nos nós

Considerando um atributo que separe os dados da seguinte forma:

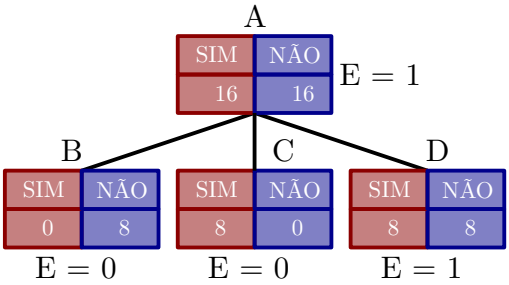


$$\Delta Info = E(\text{nó pai}) - MP(E(\text{nós filhos}))$$

5 - Entropia

Mensurando a qualidade da separação nos nós

Considerando um atributo que separe os dados da seguinte forma:



$$\Delta Info = E(\text{nó pai}) - MP(E(\text{nós filhos}))$$

$$\Delta Info = 1 - (\tfrac{8}{32}0 + \tfrac{8}{32}0 + \tfrac{16}{32}1)$$

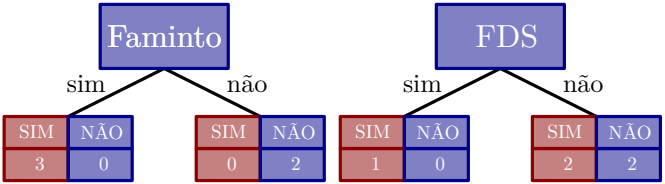
$$\Delta Info = 1 - (0 + 0 + \tfrac{1}{2}) = \tfrac{1}{2}$$

Ou seja, o ganho de informação com a separação é igual a $\frac{1}{2}$

5 - Entropia

Mensurando a qualidade da separação nos nós

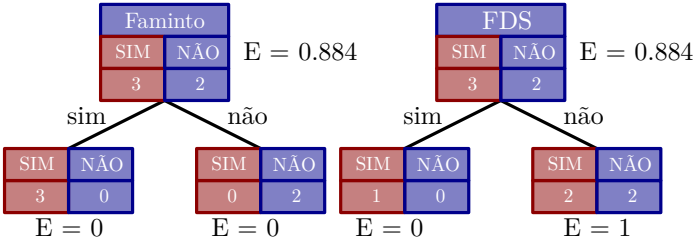
Retornando ao banco de dados do restaurante. **Usando o ganho de informação** $\Delta Info$, conseguimos definir qual dos dois atributos (Faminto ou FDS) separa melhor os dados?



5 - Entropia

Mensurando a qualidade da separação nos nós

Retornando ao banco de dados do restaurante. **Usando o ganho de informação** $\Delta Info$, conseguimos definir qual dos dois atributos (Faminto ou FDS) separa melhor os dados?

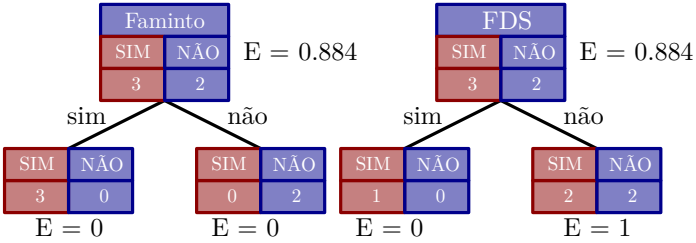


Calculando as entropias para todos os nós.

5 - Entropia

Mensurando a qualidade da separação nos nós

Retornando ao banco de dados do restaurante. **Usando o ganho de informação** $\Delta Info$, conseguimos definir qual dos dois atributos (Faminto ou FDS) separa melhor os dados?

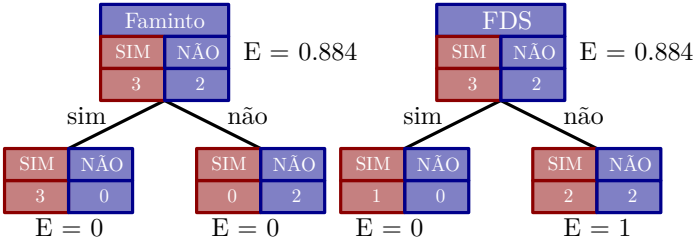


Para o atributo **Faminto**:

5 - Entropia

Mensurando a qualidade da separação nos nós

Retornando ao banco de dados do restaurante. **Usando o ganho de informação** $\Delta Info$, conseguimos definir qual dos dois atributos (Faminto ou FDS) separa melhor os dados?



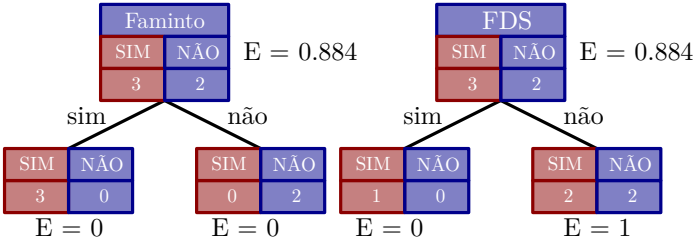
Para o atributo **Faminto**:
$\Delta Info = 0.884 - (\frac{3}{5}0 + \frac{2}{5}0)$
$\Delta Info = 0.884 - (0 + 0) = 0.884$

Ou seja, o ganho de informação com a separação é igual a 0.884

5 - Entropia

Mensurando a qualidade da separação nos nós

Retornando ao banco de dados do restaurante. **Usando o ganho de informação** $\Delta Info$, conseguimos definir qual dos dois atributos (Faminto ou FDS) separa melhor os dados?

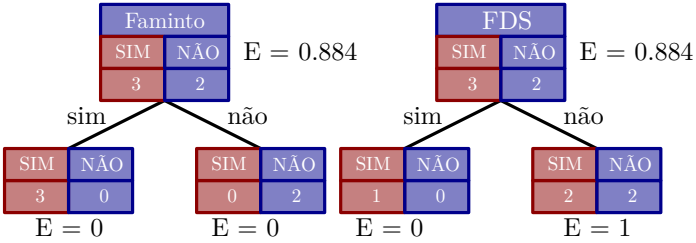


Para o atributo **FDS**:

5 - Entropia

Mensurando a qualidade da separação nos nós

Retornando ao banco de dados do restaurante. **Usando o ganho de informação** $\Delta Info$, conseguimos definir qual dos dois atributos (Faminto ou FDS) separa melhor os dados?



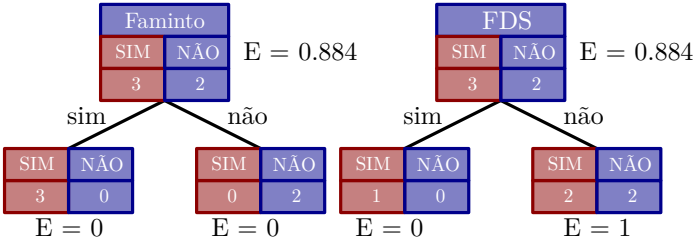
Para o atributo **FDS**:
$\Delta Info = 0.884 - (\frac{1}{5}0 + \frac{4}{5}1)$
$\Delta Info = 0.884 - (0 + 0.8) = 0.084$

Ou seja, o ganho de informação com a separação é igual a 0.0884

5 - Entropia

Mensurando a qualidade da separação nos nós

Retornando ao banco de dados do restaurante. **Usando o ganho de informação** $\Delta Info$, conseguimos definir qual dos dois atributos (Faminto ou FDS) separa melhor os dados?



Conclusão

Como $\Delta Info$ de Faminto (0.884) $>$ $\Delta Info$ FDS (0.084), Faminto é o melhor atributo para separar os dados.

5 - Entropia

Mensurando a qualidade da separação nos nós

Dessa forma, a cada iteração o algoritmo de Hunt calcula o ganho de informação para todos os atributos, e seleciona aquele com maior valor. Esse processo é repetido recursivamente a cada nó filho até que um critério de parada seja atingido, por exemplo:

- Número máximo de nós folha.
- Tamanho máximo da árvore.

Métodos de separação dos dados

6 - Separação dos dados

Como separar os dados?

Uma pergunta que devemos nos fazer ao final da estimação do modelo é:

O modelo estimado é bom? Como medir a sua performance?

Considerando modelos de classificação, uma forma simples de se fazer isso é pela **contagem de acertos e erros**, pela fórmula:

$$Acuracia = \frac{\text{N acertos}}{\text{N total}}$$

6 - Separação dos dados

Como separar os dados?

Idade	Faminto	FDS	Chovendo	Tipo	Esperou
20	SIM	SIM	NÃO	TAI	SIM
35	NÃO	NÃO	SIM	ITA	NÃO
20	SIM	NÃO	NÃO	JAP	SIM
64	SIM	NÃO	NÃO	TAI	SIM
64	NÃO	NÃO	NÃO	ITA	NÃO

Exemplo: Imagine que geramos uma árvore de classificação com os dados de clientes no restaurante. E agora queremos saber se o modelo está realmente classificando clientes de forma satisfatória.

6 - Separação dos dados

Como separar os dados?

Idade	Faminto	FDS	Chovendo	Tipo	Esperou
20	SIM	SIM	NÃO	TAI	SIM
35	NÃO	NÃO	SIM	ITA	NÃO
20	SIM	NÃO	NÃO	JAP	SIM
64	SIM	NÃO	NÃO	TAI	SIM
64	NÃO	NÃO	NÃO	ITA	NÃO
30	SIM	SIM	NÃO	TAI	???
19	SIM	NÃO	SIM	ITA	???
20	NÃO	SIM	NÃO	JAP	???

Poderíamos então **esperar** alguns novos clientes chegarem ao restaurante.

6 - Separação dos dados

Como separar os dados?

Idade	Faminto	FDS	Chovendo	Tipo	Esperou	Classificação
20	SIM	SIM	NÃO	TAI	SIM	
35	NÃO	NÃO	SIM	ITA	NÃO	
20	SIM	NÃO	NÃO	JAP	SIM	
64	SIM	NÃO	NÃO	TAI	SIM	
64	NÃO	NÃO	NÃO	ITA	NÃO	
30	SIM	SIM	NÃO	TAI	???	SIM
19	SIM	NÃO	SIM	ITA	???	SIM
20	NÃO	SIM	NÃO	JAP	???	SIM

E **usar o nosso modelo** para verificar qual seria a classificação.

6 - Separação dos dados

Como separar os dados?

Idade	Faminto	FDS	Chovendo	Tipo	Esperou	Classificação
20	SIM	SIM	NÃO	TAI	SIM	
35	NÃO	NÃO	SIM	ITA	NÃO	
20	SIM	NÃO	NÃO	JAP	SIM	
64	SIM	NÃO	NÃO	TAI	SIM	
64	NÃO	NÃO	NÃO	ITA	NÃO	
30	SIM	SIM	NÃO	TAI	NÃO	SIM
19	SIM	NÃO	SIM	ITA	SIM	SIM
20	NÃO	SIM	NÃO	JAP	SIM	SIM

Finalmente, ao fim do período no restaurante, coletamos o que realmente ocorreu, ou seja, **se os clientes esperaram ou não**.

6 - Separação dos dados

Como separar os dados?

Idade	Faminto	FDS	Chovendo	Tipo	Esperou	Classificação
20	SIM	SIM	NÃO	TAI	SIM	
35	NÃO	NÃO	SIM	ITA	NÃO	
20	SIM	NÃO	NÃO	JAP	SIM	
64	SIM	NÃO	NÃO	TAI	SIM	
64	NÃO	NÃO	NÃO	ITA	NÃO	
30	SIM	SIM	NÃO	TAI	NÃO	SIM
19	SIM	NÃO	SIM	ITA	SIM	SIM
20	NÃO	SIM	NÃO	JAP	SIM	SIM

Como temos o número de acertos do modelo (2) e o número total de classificações realizadas (3), podemos calcular a eficácia desse modelo pela fórmula:

$$Acuracia = \frac{\text{N acertos}}{\text{N total}} = \frac{2}{3} = 0.66$$

6 - Separação dos dados

Como separar os dados?

Problema

O **problema** com essa abordagem, é que se faz necessário **esperar o conjunto de dados aumentar** para avaliar o modelo. Em alguns casos isso pode demorar muito, ou nem ser possível por algum motivo qualquer.

Assim, usamos outra abordagem, em que simulamos esses novos dados com o próprio banco de dados original, simplesmente omitindo alguns registros no conjunto, ou seja, **separando os dados**.

6 - Separação dos dados

Como separar os dados?

Idade	Faminto	FDS	Chovendo	Tipo	Esperou	Classificação
20	SIM	SIM	NÃO	TAI	SIM	
35	NÃO	NÃO	SIM	ITA	NÃO	
20	SIM	NÃO	NÃO	JAP	SIM	
64	SIM	NÃO	NÃO	TAI	SIM	
64	NÃO	NÃO	NÃO	ITA	NÃO	
30	SIM	SIM	NÃO	TAI	NÃO	
19	SIM	NÃO	SIM	ITA	SIM	
20	NÃO	SIM	NÃO	JAP	SIM	

Considere o conjunto de dados completo.

6 - Separação dos dados

Como separar os dados?

Idade	Faminto	FDS	Chovendo	Tipo	Esperou	Classificação
20	SIM	SIM	NÃO	TAI	SIM	
35	NÃO	NÃO	SIM	ITA	NÃO	
20	SIM	NÃO	NÃO	JAP	SIM	
64	SIM	NÃO	NÃO	TAI	SIM	
64	NÃO	NÃO	NÃO	ITA	NÃO	
30	SIM	SIM	NÃO	TAI	NÃO	
19	SIM	NÃO	SIM	ITA	SIM	
20	NÃO	SIM	NÃO	JAP	SIM	

Podemos usar uma parte desse conjunto para treinar o modelo, chamado **conjunto de treino**.

6 - Separação dos dados

Como separar os dados?

Idade	Faminto	FDS	Chovendo	Tipo	Esperou	Classificação
20	SIM	SIM	NÃO	TAI	SIM	
35	NÃO	NÃO	SIM	ITA	NÃO	
20	SIM	NÃO	NÃO	JAP	SIM	
64	SIM	NÃO	NÃO	TAI	SIM	
64	NÃO	NÃO	NÃO	ITA	NÃO	
30	SIM	SIM	NÃO	TAI	NÃO	SIM
19	SIM	NÃO	SIM	ITA	SIM	NÃO
20	NÃO	SIM	NÃO	JAP	SIM	SIM

Com o modelo estimado, usamos outra parte dos dados (chamada **conjunto de testes**) para estimar os valores, comparando com os dados que realmente ocorreram.

6 - Separação dos dados

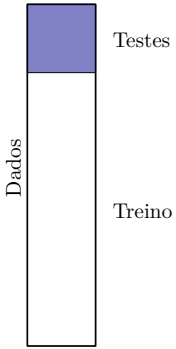
Como separar os dados?

Definição

Conjunto de treino/ testes: O conjunto de treino é a parcela do banco de dados usada para treinar o modelo, ou seja, para que ele estime a função f. O conjunto de testes é a parcela que será usada para verificar a performance do modelo (comparar acertos e erros do modelo). Existem diversas formas de separar o conjunto de dados em treino e teste.

6 - Separação dos dados

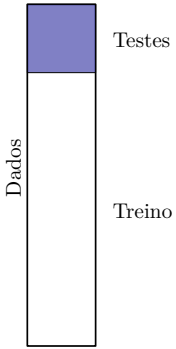
O método holdout



O **método Holdout** usa uma porcentagem para separar os dados em dois conjuntos disjuntos, de **treino** e **testes**. A acurácia do modelo é então calculada usando o conjunto de testes.

6 - Separação dos dados

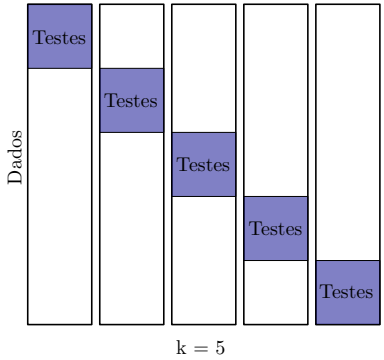
O método holdout



Um problema com essa abordagem é que a amostra de dados contida nos testes, pode não ser representativa de todo o conjunto de dados. Assim, a acurácia do modelo não é confiável.

6 - Separação dos dados

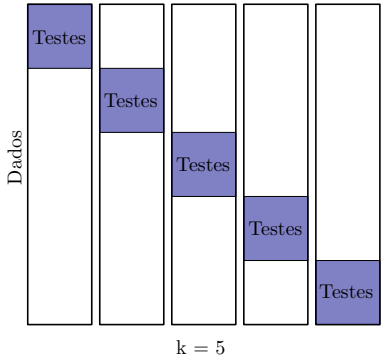
O método Cross-validation (k-fold)



Já o método **cross-validation (k-fold)** separa os dados em k conjuntos disjuntos (no caso acima k = 5).

6 - Separação dos dados

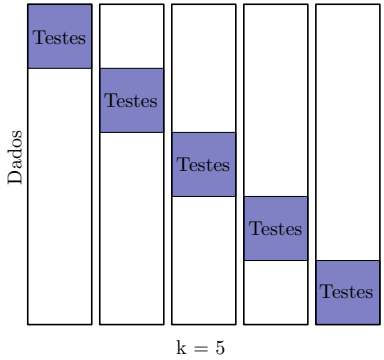
O método Cross-validation (k-fold)



Para cada conjunto uma parcela de testes e uma de treino é separada. Um modelo é estimado em cada conjunto e testado nos testes, de forma que a acurácia final é a média de todas as acurácias dos modelos.

6 - Separação dos dados

O método Cross-validation (k-fold)



Dessa forma, é garantido que todos os dados são usados para treino e também para testes, gerando uma acurácia mais confiável do que a do método *Holdout*.

Medidas de desempenho, erros e overfitting

7 - Medidas de desempenho

Matriz de confusão

- A avaliação do desempenho de um modelo de classificação é baseada na contagem dos registros previstos de forma **correta** e **incorreta** pelo modelo. Esse desempenho pode ser usado tanto no conjunto de testes quanto no conjunto de treino.

7 - Medidas de desempenho

Matriz de confusão

- A avaliação do desempenho de um modelo de classificação é baseada na contagem dos registros previstos de forma **correta** e **incorreta** pelo modelo. Esse desempenho pode ser usado tanto no conjunto de testes quanto no conjunto de treino.
- Somente contando os acertos/erros, no entanto, não nos permite encontrar quais as classes que o modelo tem mais facilidade ou dificuldade de classificar, por esse motivo, podemos tabelar os acertos/erros na chamada **matriz de confusão**.

7 - Medidas de desempenho

Matriz de confusão

EXEMPLO: Considerando o modelo de classificação dos clientes no restaurante. Fizemos a classificação de 13 registros, dos quais 10 de forma correta e 3 de forma errada. Temos a seguinte acurácia:

$$Acuracia = \frac{\text{N acertos}}{\text{N total}} = \frac{10}{13} \approx 0.77$$

Podemos dizer que o modelo tem um boa performance? De forma geral sim, essa é uma acurácia relativamente alta.

7 - Medidas de desempenho

Matriz de confusão

EXEMPLO: Suponha, ainda, que os dados de classificação mais específicos sejam os seguintes (denotando 0 = NÃO e 1 = SIM):

Ocorrido	Previsto
0	0
0	0
0	0
0	0
0	0
0	0
0	0
0	0
0	0
0	0
1	0
1	0
1	0

7 - Medidas de desempenho

Matriz de confusão

EXEMPLO: Suponha, ainda, que os dados de classificação mais específicos sejam os seguintes (denotando 0 = NÃO e 1 = SIM):

Ocorrido	Previsto
0	0
0	0
0	0
0	0
0	0
0	0
0	0
0	0
0	0
0	0
1	0
1	0
1	0

Podemos resumir os dados contidos na tabela em uma matriz (**matriz de confusão**:)

		Classe prevista	
		1	**0**
Classe ocorrida	**1**	??	??
	0	??	??

7 - Medidas de desempenho

Matriz de confusão

EXEMPLO: Suponha, ainda, que os dados de classificação mais específicos sejam os seguintes (denotando 0 = NÃO e 1 = SIM):

Ocorrido	Previsto
0	0
0	0
0	0
0	0
0	0
0	0
0	0
0	0
0	0
0	0
1	0
1	0
1	0

Podemos resumir os dados contidos na tabela em uma matriz (**matriz de confusão**:)

		Classe prevista	
		1	**0**
Classe ocorrida	**1**	??	??
	0	??	??

1. Quantos elementos ocorridos = 1 e o modelo também previu como 1? (1,1).

7 - Medidas de desempenho

Matriz de confusão

EXEMPLO: Suponha, ainda, que os dados de classificação mais específicos sejam os seguintes (denotando 0 = NÃO e 1 = SIM):

Ocorrido	Previsto
0	0
0	0
0	0
0	0
0	0
0	0
0	0
0	0
0	0
0	0
1	0
1	0
1	0

Podemos resumir os dados contidos na tabela em uma matriz (**matriz de confusão**:)

		Classe prevista	
		1	**0**
Classe ocorrida	**1**	0	??
	0	??	??

1. Quantos elementos ocorridos = 1 e o modelo também previu como 1? (1,1).

7 - Medidas de desempenho

Matriz de confusão

EXEMPLO: Suponha, ainda, que os dados de classificação mais específicos sejam os seguintes (denotando 0 = NÃO e 1 = SIM):

Ocorrido	Previsto
0	0
0	0
0	0
0	0
0	0
0	0
0	0
0	0
0	0
0	0
1	0
1	0
1	0

Podemos resumir os dados contidos na tabela em uma matriz (**matriz de confusão**:)

		Classe prevista	
		1	**0**
Classe ocorrida	**1**	0	??
	0	??	??

1. Quantos elementos ocorridos = 1 e o modelo também previu como 1? (1,1).
2. Quantos elementos ocorridos = 1 e o modelo previu como 0? (1,0).

7 - Medidas de desempenho

Matriz de confusão

EXEMPLO: Suponha, ainda, que os dados de classificação mais específicos sejam os seguintes (denotando 0 = NÃO e 1 = SIM):

Ocorrido	Previsto
0	0
0	0
0	0
0	0
0	0
0	0
0	0
0	0
0	0
0	0
1	0
1	0
1	0

Podemos resumir os dados contidos na tabela em uma matriz (**matriz de confusão**:)

		Classe prevista	
		1	**0**
Classe ocorrida	**1**	0	3
	0	??	??

1. Quantos elementos ocorridos = 1 e o modelo também previu como 1? (1,1).
2. Quantos elementos ocorridos = 1 e o modelo previu como 0? (1,0).

7 - Medidas de desempenho

Matriz de confusão

EXEMPLO: Suponha, ainda, que os dados de classificação mais específicos sejam os seguintes (denotando 0 = NÃO e 1 = SIM):

Ocorrido	Previsto
0	0
0	0
0	0
0	0
0	0
0	0
0	0
0	0
0	0
0	0
1	0
1	0
1	0

Podemos resumir os dados contidos na tabela em uma matriz (**matriz de confusão**:)

		Classe prevista	
		1	**0**
Classe ocorrida	**1**	0	3
	0	0	10

1. Quantos elementos ocorridos = 1 e o modelo também previu como 1? (1,1).
2. Quantos elementos ocorridos = 1 e o modelo previu como 0? (1,0).
3. Quantos elementos ocorridos = 0 e o modelo também previu como 0? (0,0).
4. Quantos elementos ocorridos = 0 e o modelo previu como 1? (0,1).

7 - Medidas de desempenho

Matriz de confusão

		Classe prevista	
		1	**0**
Classe ocorrida	**1**	0	3
	0	0	10

Embora a eficácia do modelo de classificação seja boa (77%), pela matriz de confusão conseguimos identificar pontos fortes e fracos do modelo:

1. Ponto forte: boa classificação dos clientes que não esperam.
2. Ponto fraco: classificação ruim de clientes que esperaram.

7 - Medidas de desempenho

Matriz de confusão

		Classe prevista	
		1	**0**
Classe ocorrida	**1**	f_{11}	f_{10}
	0	f_{01}	f_{00}

Ainda, com a matriz de confusão podemos facilmente calcular a **acurácia** e os **erros** do modelo (considerando um modelo de classificação binário com a matriz acima):

7 - Medidas de desempenho

Matriz de confusão

		Classe prevista	
		1	**0**
Classe ocorrida	**1**	f_{11}	f_{10}
	0	f_{01}	f_{00}

Ainda, com a matriz de confusão podemos facilmente calcular a **acurácia** e os **erros** do modelo (considerando um modelo de classificação binário com a matriz acima):

$$Acuracia = \frac{f_{00} + f_{11}}{f_{00} + f_{11} + f_{01} + f_{10}}$$

$$Erro = \frac{f_{10} + f_{01}}{f_{00} + f_{11} + f_{01} + f_{10}}$$

7 - Medidas de desempenho

Matriz de confusão

		Classe prevista	
		1	**0**
Classe ocorrida	**1**	f_{11}	f_{10}
	0	f_{01}	f_{00}

ATENÇÃO: Note que os acertos do modelo sempre estarão na diagonal principal da matriz de confusão, e os erros em todas as outras células.

7 - Medidas de desempenho

Matriz de confusão

EXERCÍCIO: Considere um classificador de imagens que deve discernir entre 3 frutas: maça, banana e laranja. Os dados dos erro de classificação do modelo são mostrados na Tabela:

Ocorrido	Previsto
banana	banana
maça	maça
maça	banana
banana	laranja
laranja	laranja
maça	maça
laranja	laranja
banana	banana
laranja	laranja
maça	maça
banana	banana
banana	banana
maça	maça

Crie a **matriz de confusão** para os dados e calcule a Acurácia e o Erro do modelo.

7 - Medidas de desempenho

Matriz de confusão

1. Também podemos usar a matriz de confusão para calibrar o modelo, quando **existe uma ponderação nos acertos/erros**, ou seja, algum tipo de erro é menos desejado do que outro.
2. Considere um classificador de imagens criado especificamente pelo dep. de defesa norte americano, para classificar pessoas como "BIN-LADEN"(1) OU "NÃO BIN-LADEN"(0).
3. O departamento usou parâmetros diferentes gerando dois modelos. As matrizes de confusão são mostradas abaixo:

7 - Medidas de desempenho

Matriz de confusão

1. Também podemos usar a matriz de confusão para calibrar o modelo, quando **existe uma ponderação nos acertos/erros**, ou seja, algum tipo de erro é menos desejado do que outro.
2. Considere um classificador de imagens criado especificamente pelo dep. de defesa norte americano, para classificar pessoas como "BIN-LADEN"(1) OU "NÃO BIN-LADEN"(0).
3. O departamento usou parâmetros diferentes gerando dois modelos. As matrizes de confusão são mostradas abaixo:

Eficácia = 98%

		Classe prevista	
		1	**0**
Classe ocorrida	**1**	0	2
	0	0	98

Eficácia = 50%

		Classe prevista	
		1	**0**
Classe ocorrida	**1**	2	0
	0	50	48

7 - Medidas de desempenho

Matriz de confusão

Qual o melhor modelo?

Eficácia = 98%

		Classe prevista	
		1	**0**
Classe ocorrida	**1**	0	2
	0	0	98

Eficácia = 50%

		Classe prevista	
		1	**0**
Classe ocorrida	**1**	2	0
	0	50	48

7 - Medidas de desempenho

Matriz de confusão

Qual o melhor modelo?

Note que nesse caso, existe um peso diferente aos erros cometidos pelo modelo. Os erros são:

Eficácia = 98%

		Classe prevista	
		1	**0**
Classe ocorrida	**1**	0	2
	0	0	98

Eficácia = 50%

		Classe prevista	
		1	**0**
Classe ocorrida	**1**	2	0
	0	50	48

7 - Medidas de desempenho

Matriz de confusão

Qual o melhor modelo?

Note que nesse caso, existe um peso diferente aos erros cometidos pelo modelo. Os erros são:

- Erro1: classificar erroneamente uma pessoa qualquer como Bin laden (0,1).
- Erro2: classificar erroneamente Bin Laden como uma pessoa qualquer (1,0).

Eficácia = 98%

		Classe prevista	
		1	**0**
Classe ocorrida	**1**	0	2
	0	0	98

Eficácia = 50%

		Classe prevista	
		1	**0**
Classe ocorrida	**1**	2	0
	0	50	48

7 - Medidas de desempenho

Matriz de confusão

Qual o melhor modelo?

Note que nesse caso, existe um peso diferente aos erros cometidos pelo modelo. Os erros são:

- Erro1: classificar erroneamente uma pessoa qualquer como Bin laden (0,1).
- Erro2: classificar erroneamente Bin Laden como uma pessoa qualquer (1,0).

Claramente o **Erro2** é muito mais sério do que o **Erro1**, de forma que, mesmo com uma eficácia menor, o modelo 2 é mais adequado para ser usado pelo dep. de defesa norte-americano.

Eficácia = 98%

		Classe prevista	
		1	**0**
Classe ocorrida	**1**	0	2
	0	0	98

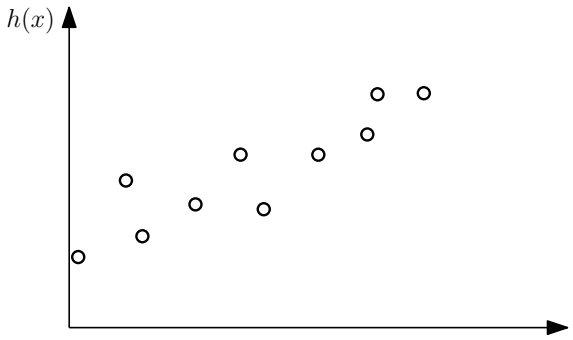
Eficácia = 50%

		Classe prevista	
		1	**0**
Classe ocorrida	**1**	2	0
	0	50	48

Tipos de erros e overfitting do modelo

8 - Tipos de erros e overfitting do modelo

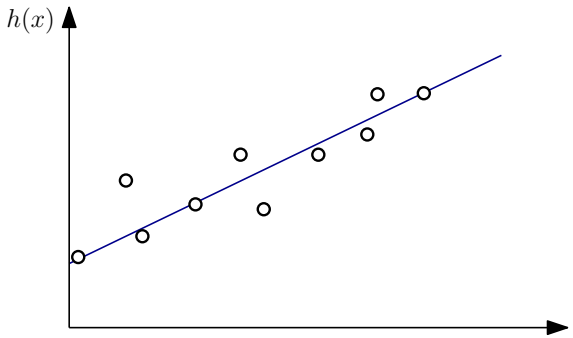
Overfitting do modelo



Considere um problema de aprendizagem supervisionada, em que devemos encontrar uma função h que estime os pares (x_i, y_i) dados pela Figura.

8 - Tipos de erros e overfitting do modelo

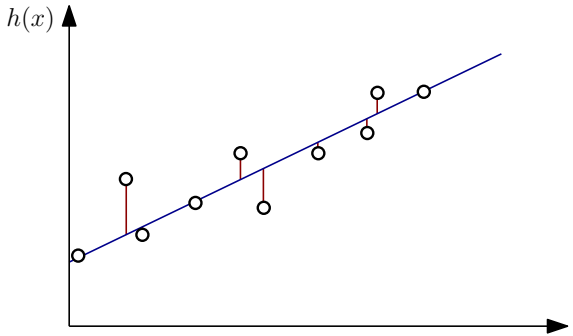
Overfitting do modelo



Uma possibilidade seria ajustar uma reta aos pontos.

8 - Tipos de erros e overfitting do modelo

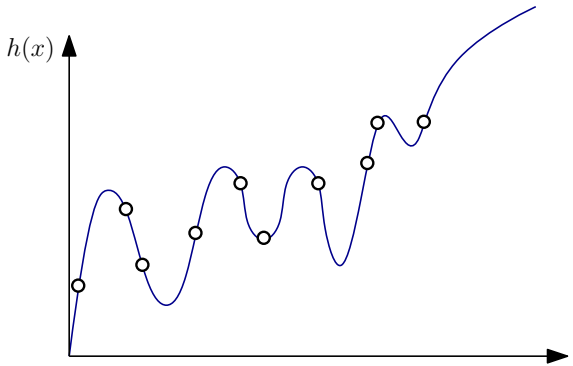
Overfitting do modelo



Associado a essa reta, conseguimos mensurar o erro ao prever os valores do banco de dados. (Esse é um **erro de treino**, pois usamos o próprio conjunto de treino para calcular os erros).

8 - Tipos de erros e overfitting do modelo

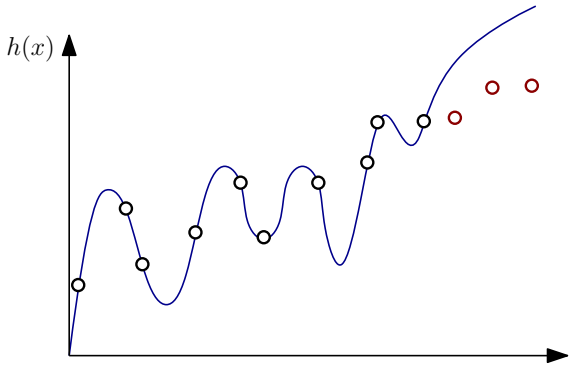
Overfitting do modelo



Na tentativa de criar um modelo mais performático, poderíamos estimar uma função h por um polinômio de grau 5, que ajusta perfeitamente todos os pontos (erro de treino $=$ 0).

8 - Tipos de erros e overfitting do modelo

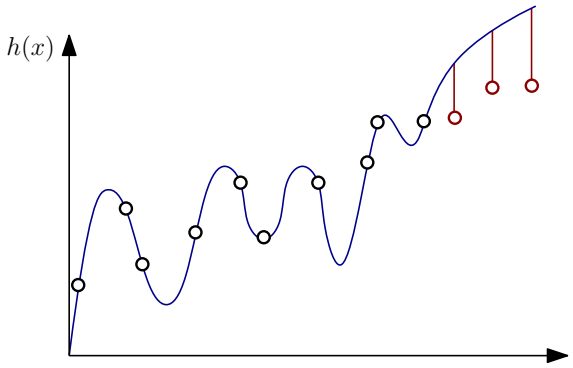
Overfitting do modelo



O objetivo de estimarmos um modelo é usarmos com dados não vistos. Imagine que usemos o modelo para estimar os valores dos novos pontos em vermelho.

8 - Tipos de erros e overfitting do modelo

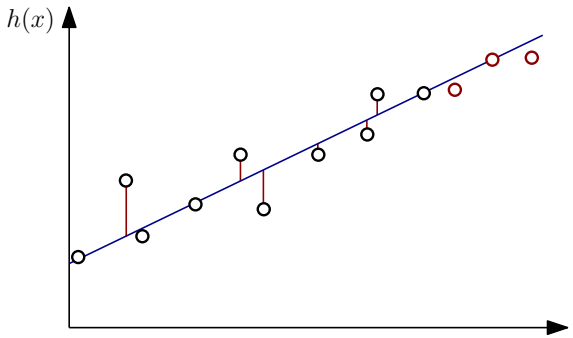
Overfitting do modelo



Podemos aferir o erro que o modelo comete nesses dados que nunca foram vistos. Este erro é chamado de erro de testes.

8 - Tipos de erros e overfitting do modelo

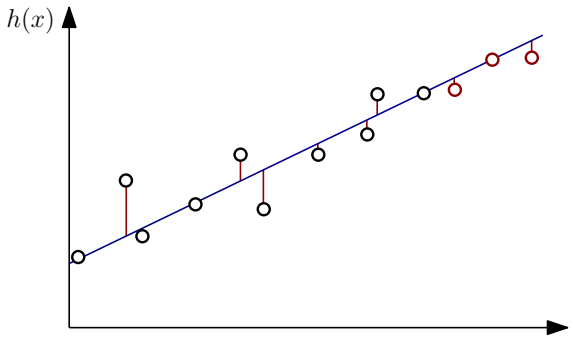
Overfitting do modelo



A mesma coisa com o modelo linear. Qual a performance dele considerando esses novos valores?

8 - Tipos de erros e overfitting do modelo

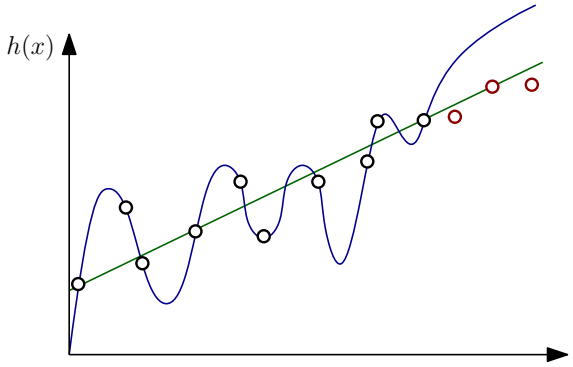
Overfitting do modelo



Aferimos isso calculando o erro de testes.

8 - Tipos de erros e overfitting do modelo

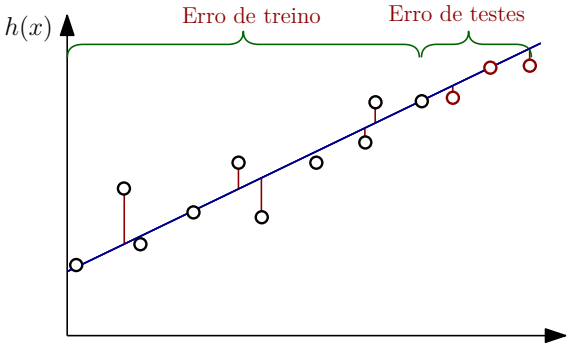
Overfitting do modelo



Observando os dois modelos juntos, qual você diria que têm um menor erro de testes (valores não vistos antes)?

8 - Tipos de erros e overfitting do modelo

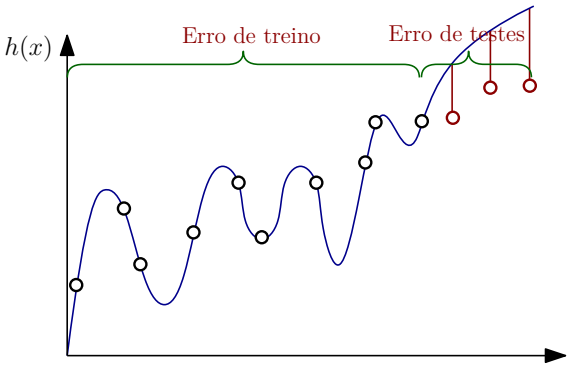
Overfitting do modelo



Para avaliar a performance de um modelo, devemos verificar os dois tipos de erros: (no conjunto de treino e no conjunto de testes).

8 - Tipos de erros e overfitting do modelo

Overfitting do modelo



Para avaliar a performance de um modelo, devemos verificar os dois tipos de erros: (no conjunto de treino e no conjunto de testes).

8 - Tipos de erros e overfitting do modelo

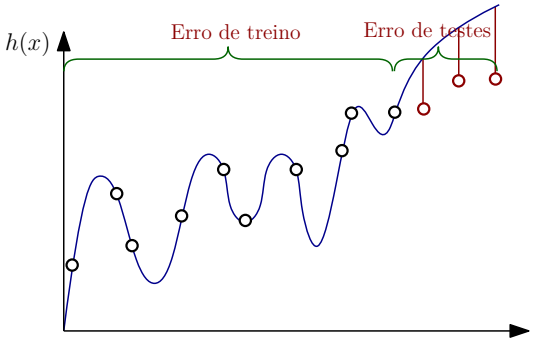
Overfitting do modelo

Os erros cometidos por um modelo de classificação são geralmente divididos em dois grupos: **erro de treinamento** e **erro de generalização (ou testes)**. **Erros de treinamento** se referem aos erros de classificação equivocada do modelo cometido no registro de treinamento, enquanto os **erros de generalização** são os erros do modelo em registros não vistos anteriormente.

Usamos esses dois tipos de erros para identificar uma patologia nos modelos de classificação, chamada de **overfitting** (ou superajuste).

8 - Tipos de erros e overfitting do modelo

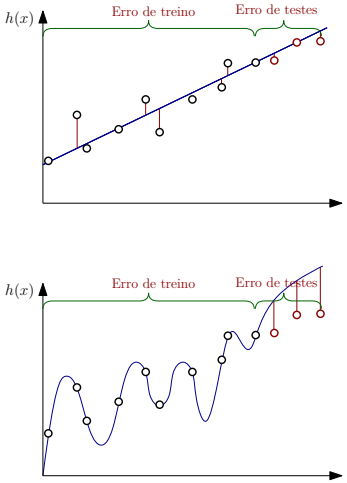
Overfitting do modelo



1. Os dois tipos de erros devem ser avaliados ao escolhermos um modelo.
2. O erro de treino nos diz se a função h descreve bem os dados.
3. O erro de testes nos diz o potencial de generalização da função h ao estimar novos casos.
4. A situação em que um modelo tem baixo erro de treino e alto erro de testes é conhecida como *overfitting* (super ajuste), ou seja, é como se o modelo tivesse "decorado"os dados de treino, de forma que a capacidade de prever novos valores é nula.

8 - Tipos de erros e overfitting do modelo

Overfitting do modelo

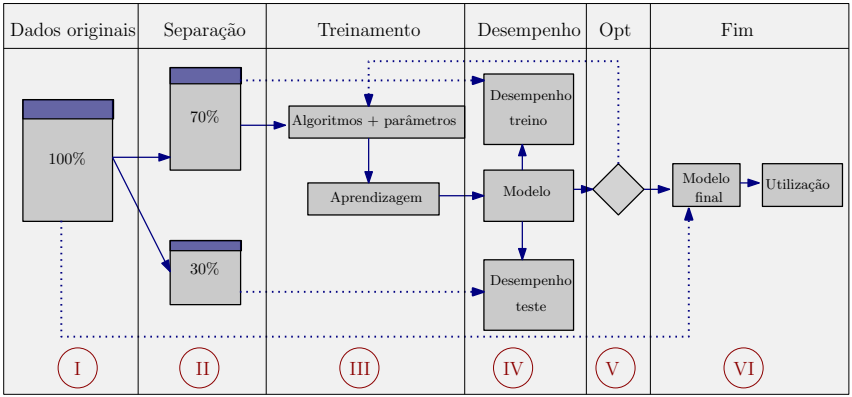


No exemplo, embora o modelo linear tenha um pior desempenho no conjunto de treino, ele performa melhor no conjunto de testes, o que o faz um modelo melhor do que o gerado pelo polinômio de quinto grau.

O modelo mais complexo sofre de **overfitting** dos dados.

Conclusão

9 - Conclusão


Dados originais
Separação
Treinamento
Desempenho
Opt
Fim
100%
70%
30%
Algoritmos + parâmetros
Aprendizagem
Desempenho treino
Modelo
Desempenho teste
Modelo final
Utilização
I
II
III
IV
V
VI

9 - Conclusão

Os seguintes termos devem ser de conhecimento após o fim desta aula:

1. Aprendizado supervisionado.
2. Conjuntos de teste e treino.
3. Árvores de classificação.
4. Otimização de hiperparâmetros.
5. Matriz de confusão.
6. Overfitting.