Aprendizado de Máquina

Prof. Aran Morales, UNISUL – Anima Educação

Apostila – Analises Preditiva - Classificação

Roteiro

1. **Classificação e modelos preditivos**: conceitos;

2. **Aprendizagem supervisionada:** abordagem simbólica: arvores de decisão e floresta randômica (Random Forest);

3. **Aprendizagem supervisionada:** abordagem estatística: Naive Bayes, k-vizinhos mais próximos, regressão logística.

4. **Aprendizagem supervisionada:** abordagem biológica: Redes Neurais Artificiais;

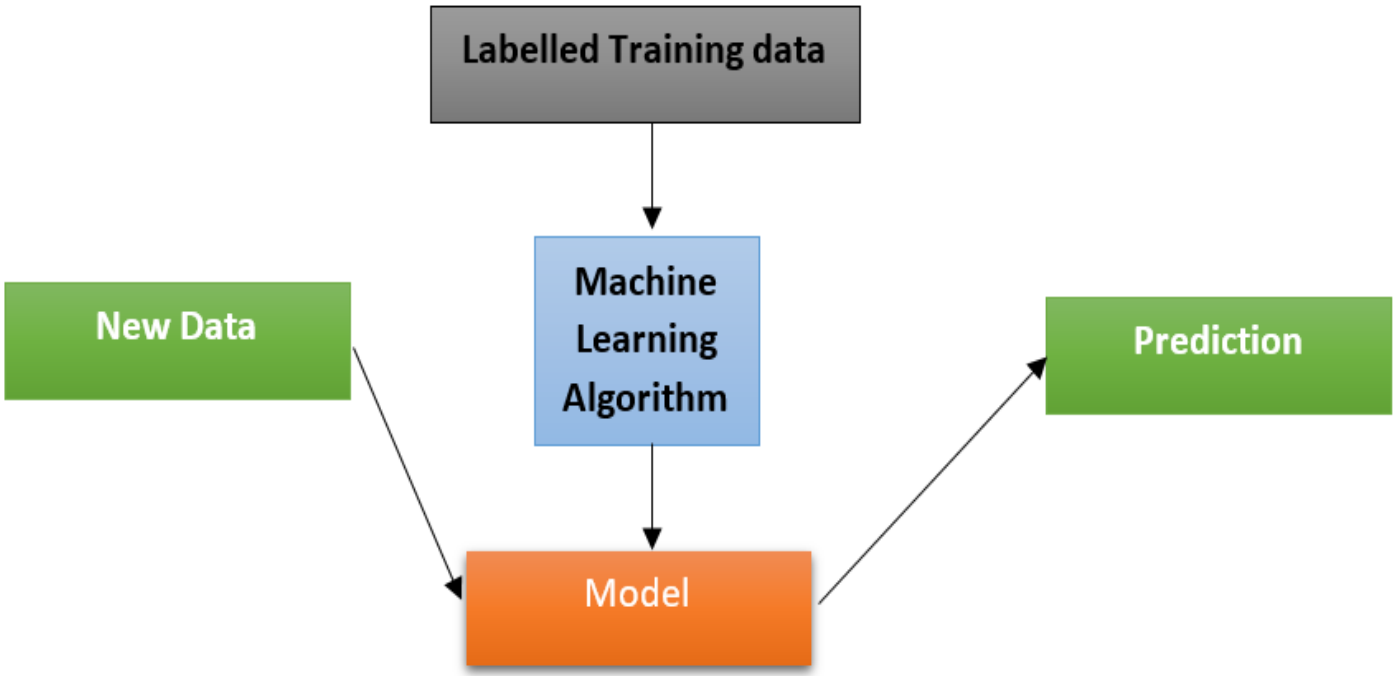
5. **Avaliação de modelos de classificação:** matriz de confusão, Curva ROC (AUC);

1. Classificação e Modelos Preditivos

Modelos Preditivos: Classificação

Aprendizagem Supervisionada

O objetivo da aprendizagem supervisionada é aprender um modelo de dados de treinamento rotulado que nos permite fazer previsões sobre dados futuros.



Modelos Preditivos: Classificação

O objetivo da classificação, é analisar os dados e desenvolver uma **descrição** ou **modelo** para descobrir um **relacionamento** entre os **atributos previsores** e o **atributo meta**.

A tarefa da classificação, é caracterizada por uma **boa definição das classes**, adquirida em um conjunto de exemplos **pre-classificados** (dados de treino).

Modelos Preditivos: Classificação

Training Data

1. Um **conjunto de treino** com **exemplos rotulados** é usado para treinar o classificador.

2. Um algoritmo de aprendizagem é executado para induzir um classificador a partir do conjunto de treino

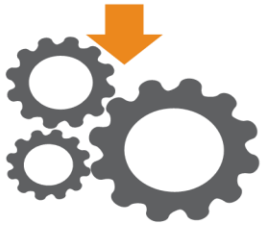
Train the Machine Learning Algorithm

Evaluate

Model

3. Uma vez construído o classificador, este pode ser usado para classificar futuros exemplos

Input Data



Machine Learning Algorithm

Prediction

Modelos Preditivos: Classificação

Introdução: Definição, objetivos, tarefas e características da classificação;

Abordagem Simbólica: classificação baseado na IA simbólica (heurística): árvore de decisão, teoria da informação, algoritmos ID3 e C4.5;

Abordagem Estatística: Classificadores Bayesianos (Naive Bayes), K-Vizinhos mais próximos (k-Nearest Neighbor);

Abordagem Biológica: classificador baseado na IA biológica: redes neurais e algoritmos genéticos.

2. Abordagem simbólica: árvores de decisão

Árvores de Decisão

São um **método de aprendizagem supervisionado** que constrói árvores de classificação a partir de exemplos.

Algoritmos : ID3, C4.5, (Quinlan), CART (Breiman)

Os métodos baseados em árvores, dividem o espaço de entrada em **regiões disjuntas** para construir uma **fronteira de decisão**.

As regiões são escolhidas baseadas em técnicas **heurísticas** onde a cada passo os algoritmos selecionam a variável que provê a **melhor separação de classes.**

Árvores de Decisão

Cada **vértice** (nodo) corresponde a um **atributo**, e cada **aresta** da árvore a um **valor possível** do atributo.

Uma **folha** da árvore corresponde ao valor esperado da **decisão** segundo os dados de treino utilizados.

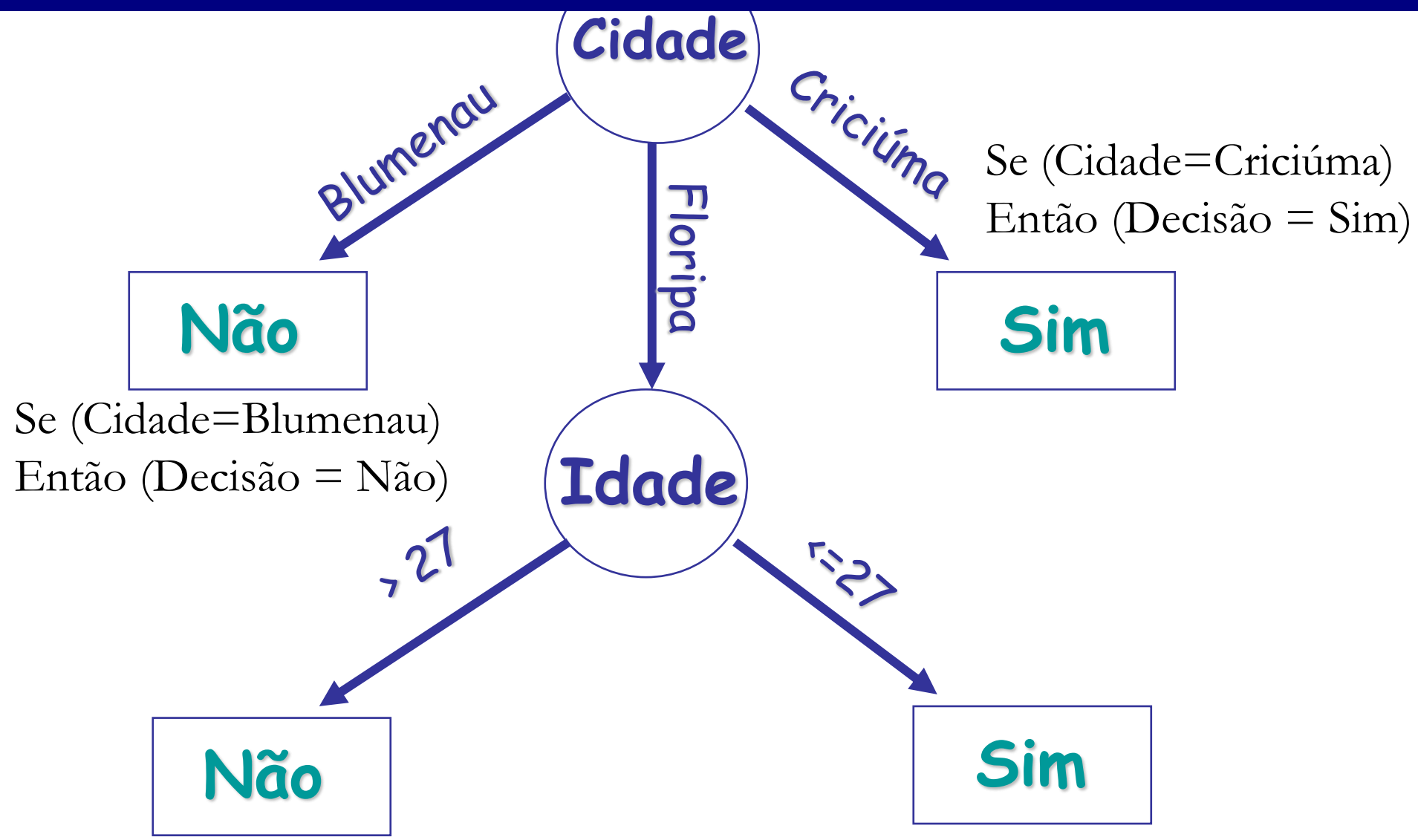
Cada **vértice** é **associado** ao **atributo** mais **informativo** que ainda não tenha sido considerado.

Para medir o **nível de informação** de um atributo se utiliza o conceito de **entropia da Teoria da Informação.**

Árvores de Decisão

ID	Sexo	Cidade	Idade	Comprou Produto X ?
1	M	Floripa	25	S
2	M	Criciuma	21	S
3	F	Floripa	23	S
4	F	Criciuma	34	S
5	F	Floripa	30	N
6	M	Blumenau	21	N
7	M	Blumenau	20	N
8	F	Blumenau	18	N
9	F	Floripa	34	N
10	M	Floripa	55	N

Árvores de Decisão

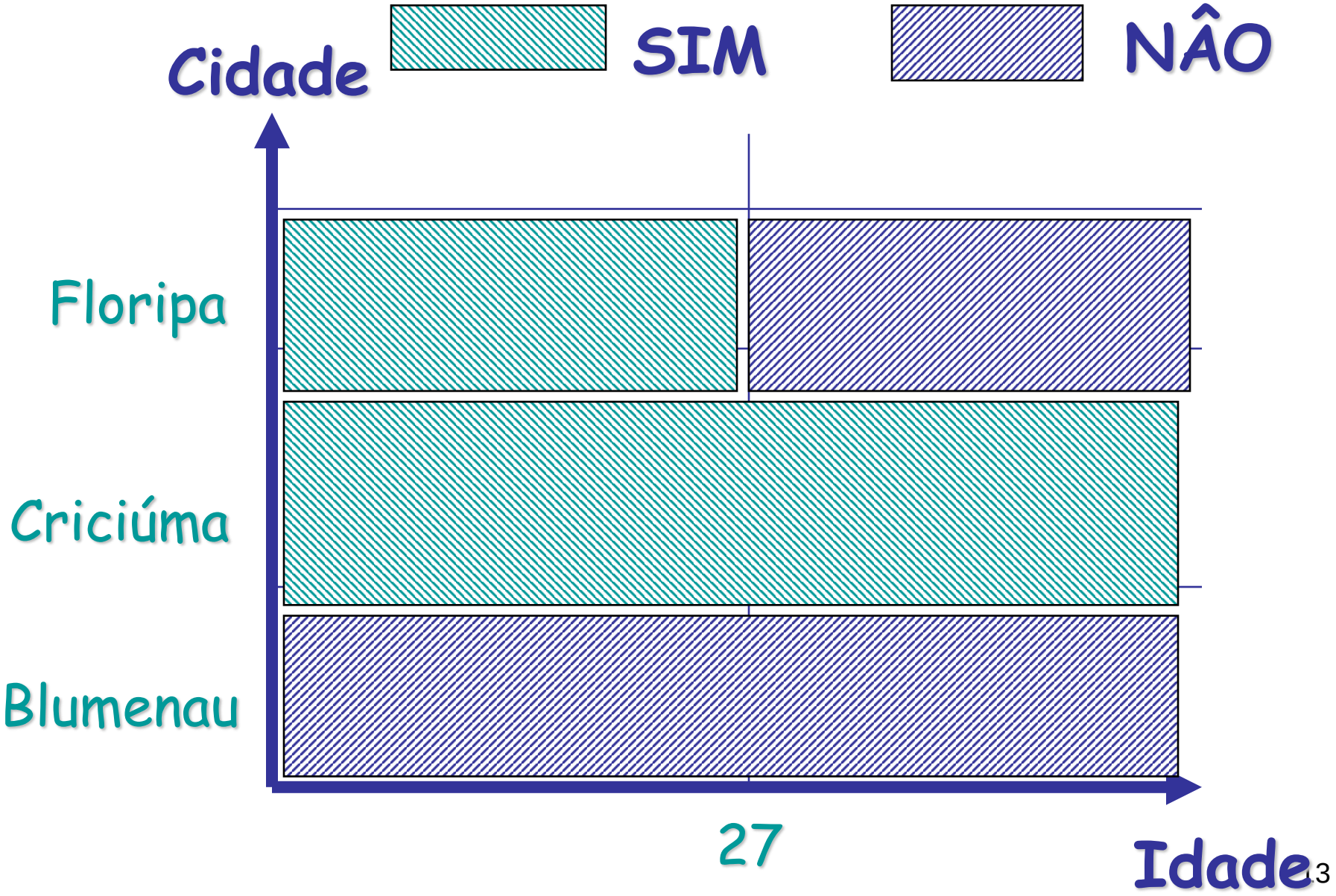


Se (Cidade=Criciúma)
Então (Decisão = Sim)

Se (Cidade=Blumenau)
Então (Decisão = Não)

Se (Cidade=Floripa e Idade <= 27) Então (Decisão = Sim)
Se (Cidade=Floripa e Idade > 27) Então (Decisão = Não)

Árvores de Decisão



Algoritmo ID3

ID3, é um algoritmo que construí uma **árvore de decisão** sob as seguintes premissas:

Cada **vértice** (nodo) corresponde a um **atributo**, e cada **aresta** da árvore a um **valor possível** do atributo.

Uma **folha** da árvore corresponde ao valor esperado da **decisão** segundo os dados de treino utilizados.

A **explicação** de uma determinada decisão está na **trajetória** da raiz a folha representativa desta decisão.

Algoritmo ID3

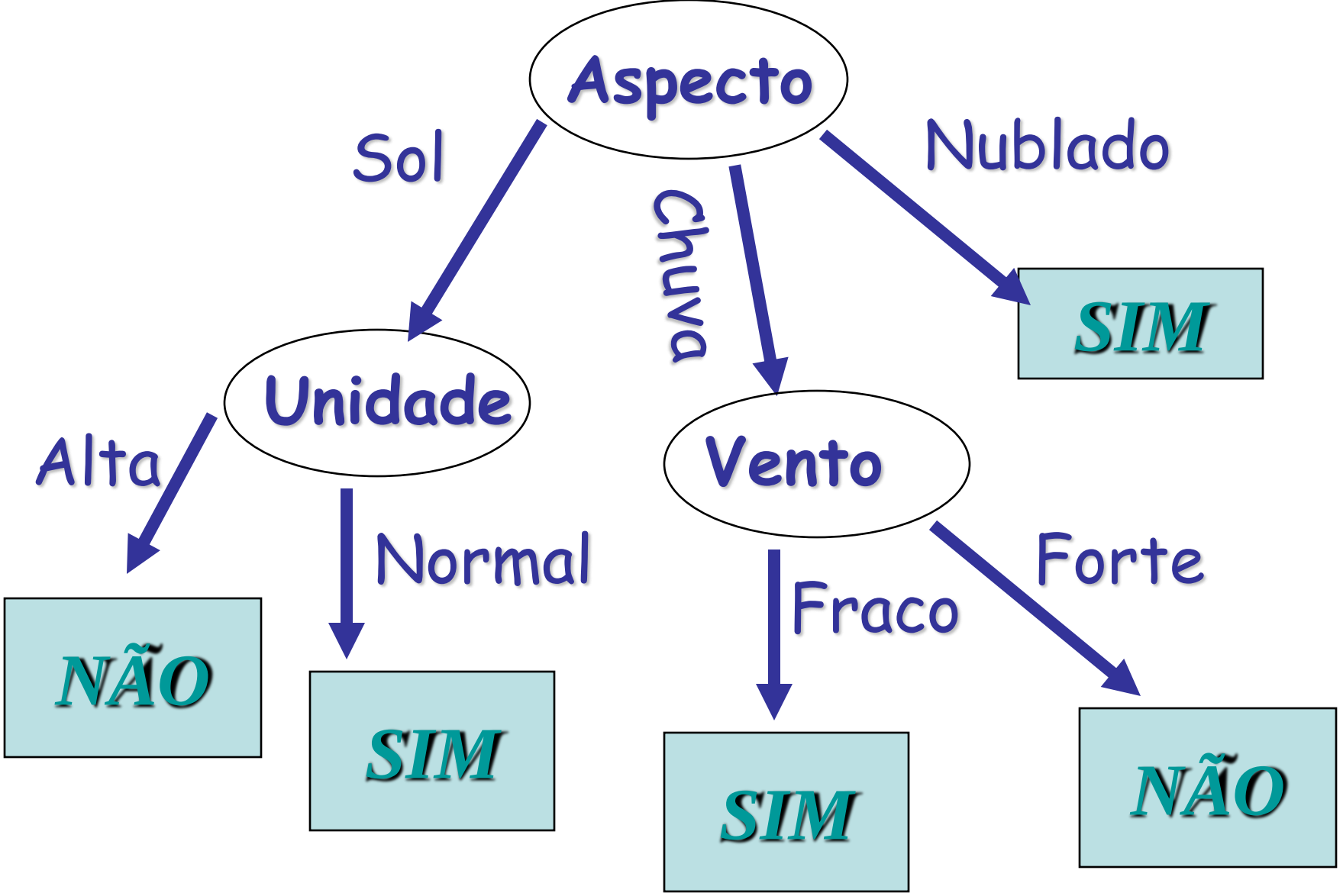
Cada **vértice** é **associado** ao **atributo** mais **informativo** que ainda não tenha sido considerado.

Para medir o **nível de informação** de um atributo se utiliza o conceito de **entropia da Teoria da Informação.**

Menor o valor da entropia, menor a incerteza e mais utilidade tem o atributo para a classificação.

Árvores de Decisão

Dia	Aspecto	Temperatura	Umidade	Vento	Decisão
1	Sol	Quente	Alta	Fraco	N
2	Sol	Quente	Alta	Forte	N
3	Nublado	Quente	Alta	Fraco	S
4	Chuva	Agradável	Alta	Fraco	S
5	Chuva	Fria	Normal	Fraco	S
6	Chuva	Fria	Normal	Forte	N
7	Nublado	Fria	Normal	Forte	S
8	Sol	Agradável	Alta	Fraco	N
9	Sol	Fria	Normal	Fraco	S
10	Chuva	Agradável	Normal	Fraco	S
11	Sol	Agradável	Normal	Forte	S
12	Nublado	Agradável	Alta	Forte	S
13	Nublado	Quente	Normal	Fraco	S
14	Chuva	Agradável	Alta	Forte	N



Qual é a decisão se: Aspecto = Sol, Temperatura = Agradavel;
Umidade = Normal e Vento = Fraco?

Árvores de Decisão: Exemplo ID3

Dia	Aspecto	Temperatura	Umidade	Vento	Decisão
1	Sol	Quente	Alta	Fraco	N
2	Sol	Quente	Alta	Forte	N
3	Nublado	Quente	Alta	Fraco	S
4	Chuva	Agradável	Alta	Fraco	S
5	Chuva	Fria	Normal	Fraco	S
6	Chuva	Fria	Normal	Forte	N
7	Nublado	Fria	Normal	Forte	S
8	Sol	Agradável	Alta	Fraco	N
9	Sol	Fria	Normal	Fraco	S
10	Chuva	Agradável	Normal	Fraco	S
11	Sol	Agradável	Normal	Forte	S
12	Nublado	Agradável	Alta	Forte	S
13	Nublado	Quente	Normal	Fraco	S
14	Chuva	Agradável	Alta	Forte	N

Árvores de Decisão: Exemplo ID3

O número de combinações possíveis são:

Aspecto: sol, nublado, chuva

Temperatura: quente, agradável, frio

Umidade: alta, normal

Vento: fraco, forte

(3 x 3 x 2 x 2 = 36)

Seleção de atributos para construir a árvore de decisão.

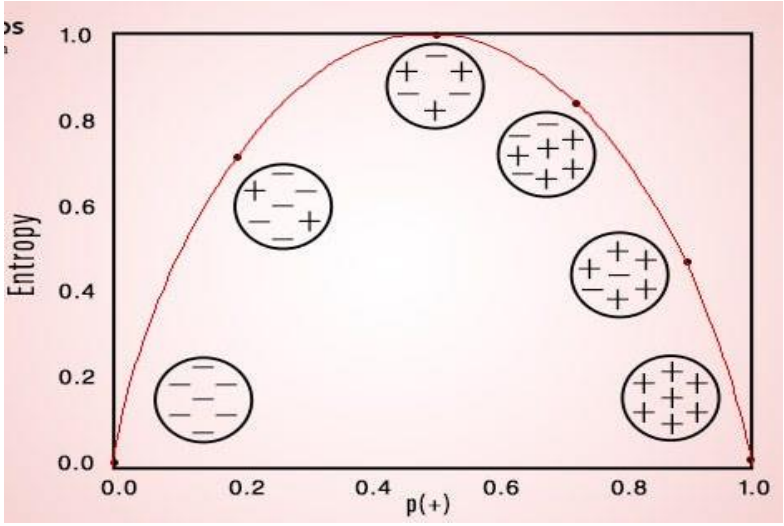
Qual é o atributo previsor mais relevante para prever a classe a qual pertencem os dados ?

Árvores de Decisão: Teoria da Informação

Dada uma distribuição de probabilidade **$P = (p_1, p_2, \ldots, p_n)$,** a Informação contida nesta distribuição, é chamada de **entropia** (função de informação de Shannon), e definida como:

$I(P) = -[p_1 * \log_2(p_1) + \ldots + p_n * \log_2(p_n)]$

Observação: $\log_a N = \log N / \log a$



Exemplos:

P = (0.5;0.5) ➔ I(P) = 1;

P = (0.67;0.33) ➔ I(P) = 0.92;

P = (1.0;0.0) ➔ I(P) = 0;

Observação: mais uniforme é a distribuição de probabilidade, maior é a entropia e por tanto maior a incerteza ou menor a informação.

Árvores de Decisão: Teoria da Informação

"O que é entropia?"

É uma medida da desordem ou a medida da pureza.

Basicamente, **é a medição da impureza ou aleatoriedade nos pontos de dados**.

A entropia é a mais baixa (sem desordem) nos extremos (ambos os extremos) e máxima (alta desordem) no meio do gráfico.

$$\text{Entropy} = -\sum_{i=1}^{n} p_i * \text{Log}_2(p_i)$$

Árvores de Decisão: Teoria da Informação

Se temos um conjunto **T** de registros, distribuídos em **k** classes **(C_1,. . . ,C_k),** a informação necessária para identificar a classe de um elemento de **T** é **Info(T) = I(P)**

Onde **P** é a distribuição de probabilidade das classes **(C_1 , . . . , C_k)**

P = (C_1 / T, . . . , C_k / T)

Isto é, a proporção de elementos pertencentes a classe **i.**

Árvores de Decisão: Exemplo – ID3

Conjunto de registros **T** com 14 registros.

Duas partições **S** e **N**, com probabilidade **9/14** e **5/14**.

Info(T) = I(9/14, 5/14) =

$$-(9/14 \cdot \text{Log}_2 (9/14) + 5/14 \cdot \text{Log}_2 (5/14)) = \mathbf{0.94}$$

Se particionamos **T** sobre a base dos valores do atributo **X** em conjuntos $\mathbf{t_1, \ldots t_n}$, então a informação necessária para identificar a classe de um elemento de **T**, é:

$\mathbf{Info\ (X,T) = Soma_i\ (\ t_i/T) * Info\ (t_i)}$, onde $\mathbf{t_i}$ é o conjunto de Possíveis valores do atributo **X** .

Árvores de Decisão: Exemplo – ID3

Info (Aspecto, T) = sol/T * I (sol) +
nublado/T * I (nublado) + chuva/T * I (chuva) =

Aspecto	FS	FN
Sol	2/5	3/5
Nublado	4/4	0/4
Chuva	3/5	2/5

Info (Aspecto, T) = 5/14 * I (2/5, 3/5) +
4/14 * I (4/4 , 0) + 5/14 * I (3/5,2/5) = 0.693

Árvores de Decisão: Teoria da Informação

Definição: o ganho de informação do atributo **X**, é a diferencia entre a informação necessária para identificar um elemento de **T** e a informação necessária para identificar um elemento de **T** depois que o valor de atributo **X** tenha sido considerado:

Ganho (X,T) = Info (T) - Info (X, T)

Info(T) = 0.94; Info(Aspecto, T) = 0.693

Ganho (Aspecto, T) = 0.94 - 0.693 = 0.247

Com o objetivo de criar árvores de decisão pequenos, para identificar poucas regras, o atributo escolhido para nó da árvore é o **atributo de maior ganho.**

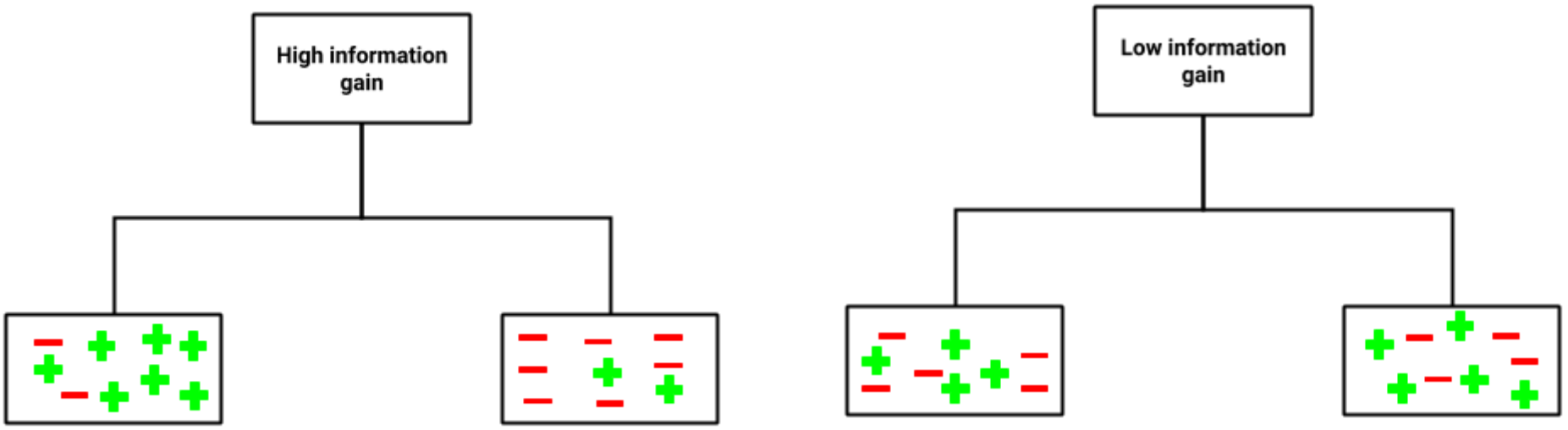
Árvores de Decisão: Teoria da Informação

Ganho de informação é aplicado para quantificar qual recurso fornece informações máximas sobre a classificação com base na noção de entropia, ou seja , **quantificando o tamanho da incerteza, desordem ou impureza** , em geral, **com a intenção de diminuir a quantidade de entropia iniciando do topo (nó raiz) para baixo (nós folhas).**

Árvores de Decisão: Teoria da Informação

O ganho de informação é uma propriedade estatística que mede o quão bem um determinado atributo separa os exemplos de treinamento de acordo com sua classificação alvo.

Na figura abaixo, podemos ver que um atributo com baixo ganho de informação (à direita) divide os dados de forma relativamente uniforme e, como resultado, não nos aproxima de uma decisão. Considerando que, um atributo com alto ganho de informação (à esquerda) divide os dados em grupos com um número ímpar de positivos e negativos e, como resultado, ajuda a separar os dois um do outro



Árvores de Decisão: Exemplo – ID3

Aspecto	FS	FN
Sol	2/5	3/5
Nublado	4/4	0/4
Chuva	3/5	2/5

Info (Sol) = 0.971
Info (Nublado) = 0.0
Info (Chuva) = 0.971

Info (Aspecto) = 0.693
(5/14 * 0.971 + 4/14 * 0.0 + 5/14 * 0.971)
Ganho (Aspecto) = 0.94 - 0.693 = 0.247

Árvores de Decisão: Exemplo – ID3

Temperatura	FS	FN
Quente	2/4	2/4
Agradável	4/6	2/6
Fria	3/4	1/4

Info (Quente) = 1.0

Info (Agradável) = 0.919

Info (Fria) = 0.811

Info (Temperatura) = 0.912

(4/14 * 1.0 + 6/14 * 0.919 + 4/14 * 0.811)

Ganho (Temperatura) = 0.94 - 0.912 = 0.028

Árvores de Decisão: Exemplo – ID3

Umidade	FS	FN
Alta	3/7	4/7
Normal	6/7	1/7

Info (Alta) = 0.984

Info (Normal) = 0.592

Info (Umidade) = 0.636

(7/14 * 0.984 + 7/14 * 0.592)

Ganho (Umidade) = 0.94 - 0.788 = 0.152

Árvores de Decisão: Exemplo – ID3

Vento	FS	FN
Fraco	6/8	2/8
Forte	3/6	3/6

Info (Forte) = 1.0

Info (Fraco) = 0.811

Info (Vento) = 0.892

(6/14 * 1.0 + 8/14 * 0.541)

Ganho (Vento) = 0.94 - 0.738 = 0.048

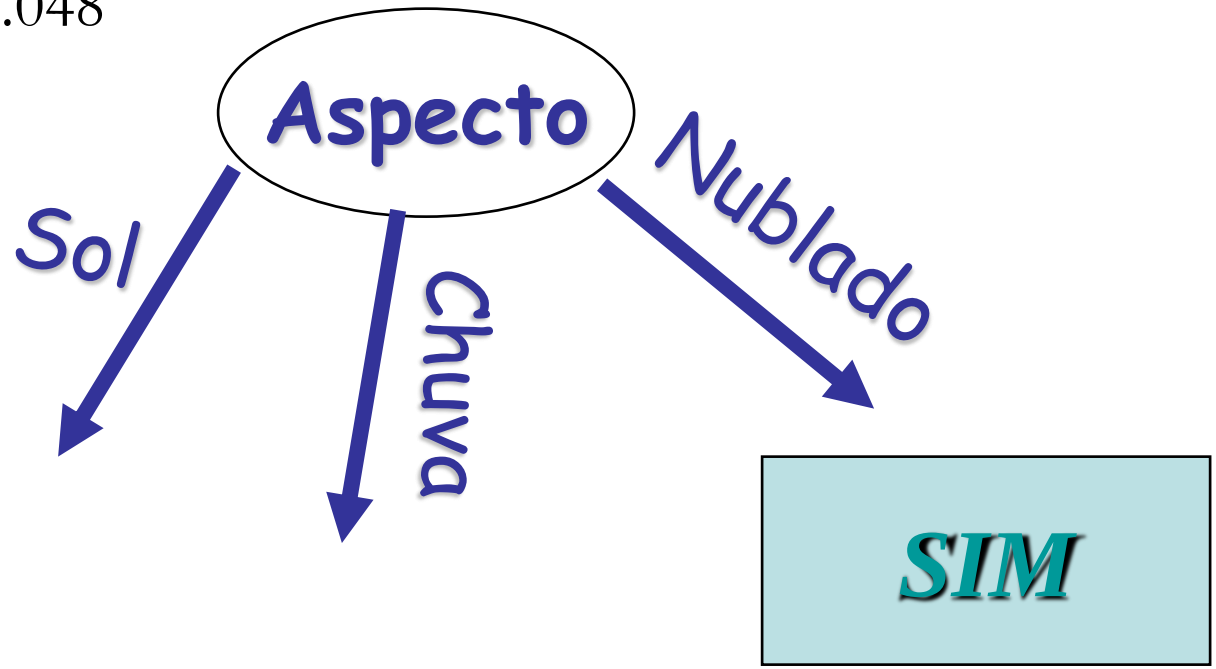
Árvores de Decisão: Exemplo – ID3

Ganho (Aspecto) = 0.247

Ganho (Temperatura) = 0.028

Ganho (Umidade) = 0.152

Ganho (Vento) = 0.048



Árvores de Decisão: Exemplo – ID3

Dia	Aspecto	Temperatura	Umidade	Vento	Decisão
1	Sol	Quente	Alta	Fraco	N
2	Sol	Quente	Alta	Forte	N
3	Nublado	Quente	Alta	Fraco	S
4	Chuva	Agradável	Alta	Fraco	S
5	Chuva	Fria	Normal	Fraco	S
6	Chuva	Fria	Normal	Forte	N
7	Nublado	Fria	Normal	Forte	S
8	Sol	Agradável	Alta	Fraco	N
9	Sol	Fria	Normal	Fraco	S
10	Chuva	Agradável	Normal	Fraco	S
11	Sol	Agradável	Normal	Forte	S
12	Nublado	Agradável	Alta	Forte	S
13	Nublado	Quente	Normal	Fraco	S
14	Chuva	Agradável	Alta	Forte	N

Árvores de Decisão: Exemplo – ID3

Escolhemos Aspecto = Sol

Info (T) = I (2/5; 3/5) = 0.971

Temperatura

Info (Quente) = I (0/2, 2/2) = 0.0
Info (Agradável) = I (1/2, 1/2) = 1.0
Info (Fria) = I (1/1, 0/1) = 0.0

Info(Temperatura) = 0.571
(2/5 * 0.0 + 2/5 * 1.0 + 1/5 * 0.0 = 0.4)

Ganho (Temperatura) = 0.971 - 0.4 = 0.571

Árvores de Decisão: Exemplo – ID3

Umidade

Info (Alta) = I (0/3, 3/3) = 0.0

Info (Normal) = I (2/2, 0/2) = 0.0

Info(Umidade) = 0.0 (3/5 * 0.0 + 2/5 * 0.0 = 0.0)

Ganho (Umidade) = 0.971 - 0.0 = 0.971

Vento

Info (Fraco) = I (1/3, 2/3) = 0.919

Info (Forte) = I (1/2, 1/2) = 1.0

Info(Vento) = 0.951 (3/5 * 0.919 + 2/5 *1.0) = 0.951

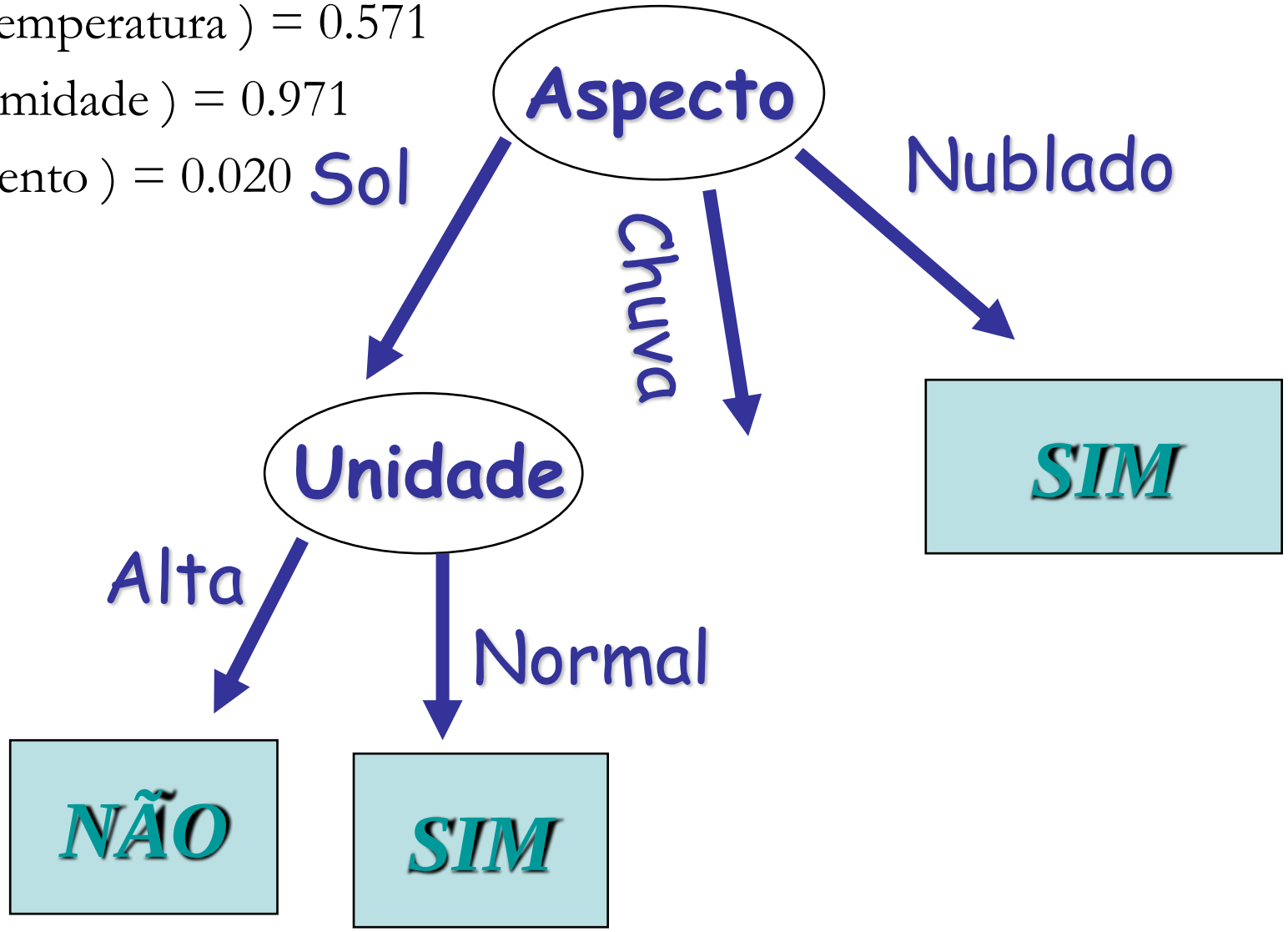
Ganho (Vento) = 0.971 - 0.951 = 0.020

Árvores de Decisão: Exemplo – ID3

Ganho (Temperatura) = 0.571

Ganho (Umidade) = 0.971

Ganho (Vento) = 0.020



Árvores de Decisão: Exemplo – ID3

Dia	Aspecto	Temperatura	Umidade	Vento	Decisão
1	Sol	Quente	Alta	Fraco	N
2	Sol	Quente	Alta	Forte	N
3	Nublado	Quente	Alta	Fraco	S
4	Chuva	Agradável	Alta	Fraco	S
5	Chuva	Fria	Normal	Fraco	S
6	Chuva	Fria	Normal	Forte	N
7	Nublado	Fria	Normal	Forte	S
8	Sol	Agradável	Alta	Fraco	N
9	Sol	Fria	Normal	Fraco	S
10	Chuva	Agradável	Normal	Fraco	S
11	Sol	Agradável	Normal	Forte	S
12	Nublado	Agradável	Alta	Forte	S
13	Nublado	Quente	Normal	Fraco	S
14	Chuva	Agradável	Alta	Forte	N

Árvores de Decisão: Exemplo – ID3

Escolhemos Aspecto = Chuva

Info (T) = I (3/5; 2/5) = 0.971

Temperatura
Info (Quente) = I (0/0, 0/0) = 0;
Info (Agradável) = I (2/3, 1/3) = 0.919
Info (Fria) = I (1/2, 1/2) = 1.0

Info(Temperatura) = 0.95 (3/5 * 0.919 + 2/5 * 1.0 = 0.95)
Ganho (Temperatura) = 0.971 - 0.95 = 0.021

Árvores de Decisão: Exemplo – ID3

Vento

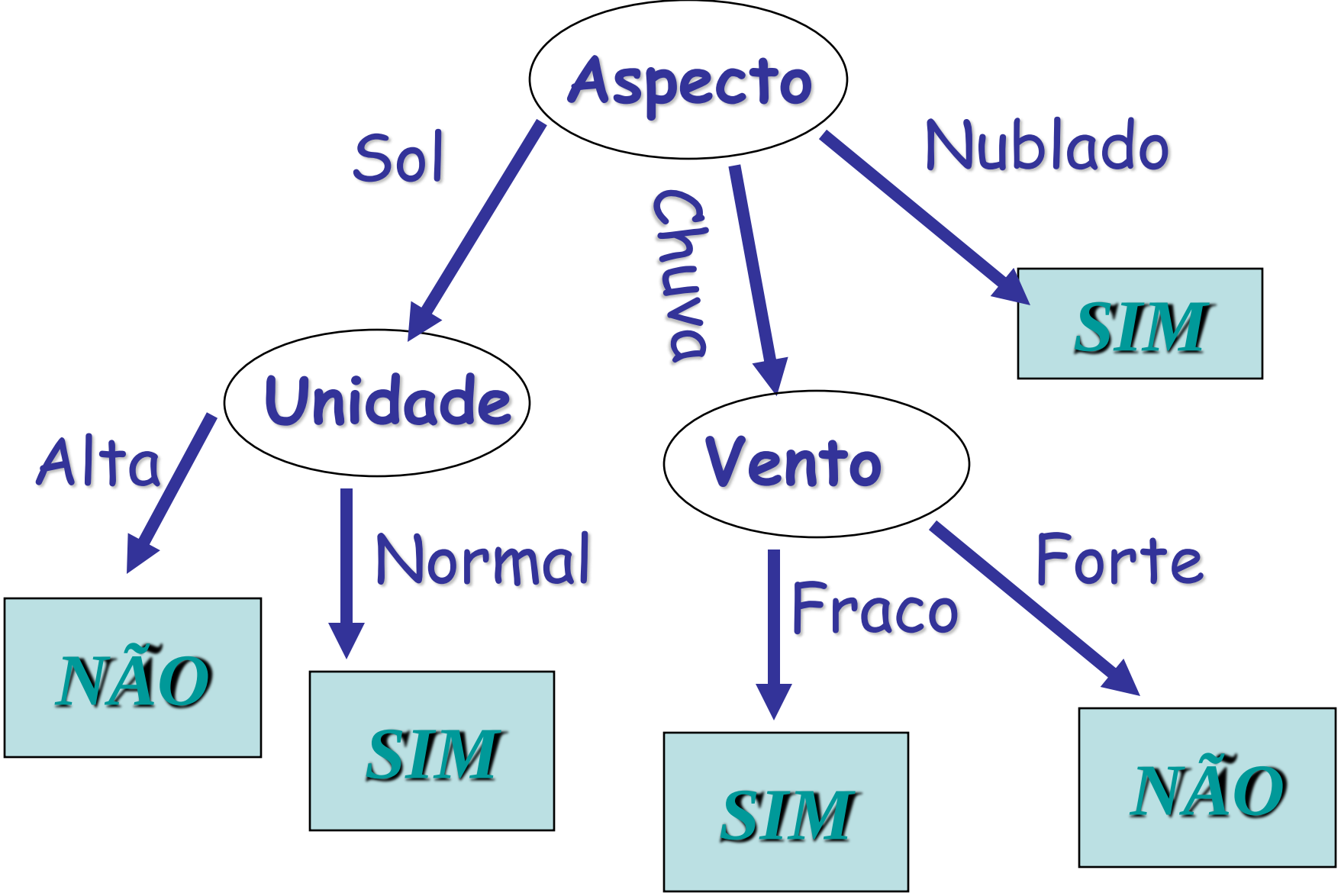
Info (Fraco) = I (3/3, 0/3) = 0.0

Info (Forte) = I (0/2, 2/2) = 0.0

Info(Vento) = 0.0 (3/5 * 0.0 + 2/5 *0. 0)

Ganho (Vento) = 0.971-0.0 = 0.971

Ganho (Temperatura) = 0.021


Aspecto
Sol
Nublado
Chuva
SIM
Unidade
Vento
Alta
Normal
Fraco
Forte
NÃO
SIM
SIM
NÃO

Árvores de Decisão: Exemplo – ID3

Se Aspecto = **Sol** e Umidade = **Alta** Então Jogar = **Não**
Se Aspecto = **Sol** e Umidade = **Normal** Então Jogar = **Sim**
Se Aspecto = **Chuva** e Vento = **Fraco** Então Jogar = **Sim**
Se Aspecto = **Chuva** e Vento = **Forte** Então Jogar = **Não**
Se Aspecto = **Nublado** Então Jogar = **Sim**

Qual será a decisão, se o dia estiver com sol, temperatura fria, umidade alta e vento forte ?

Algoritmo C4.5

O **C 4.5** é uma extensão do **ID3**: Construí árvores de decisão, com valores **desconhecidos** para alguns **atributos**.

Trabalha com **atributos** que apresentam **valores contínuos**.
Utiliza o conceito de **poda (pruning)** de árvores.

Quando existem **atributos desconhecidos** para alguma variável, os mesmos são considerado como uma nova categoria.

Quando existem variáveis com **atributos contínuos**, o algoritmo cria intervalos segundo as alterações na variável de decisão.

Árvores de Decisão: Algoritmo C4.5

Variável	Decisão
64	S
65	N
68	S
69	S
70	S
71	N
72	N
73	S
75	S
75	S
80	N
81	S
83	S
85	N

Variável	Decisão
64	S
65	N
68	S
69	S
70	S
71	N
72	N
73	S
75	S
75	S
80	N
81	S
83	S
85	N

64.5

66.5

70.5

72.5

77.5

80.5

84

Árvores de Decisão: Algoritmo C4.5

Exemplo:

Info(71.5) = Info([4,2], [5,3]) =

$$6/14 \,.\, I(4/6, 2/6) + 8/14 \,.\, I(5/8, 3/8) = 0.939$$

Poda da árvore, significa substituir uma **parte da árvore** (sub-árvore) por uma **folha**, com o objetivo de simplificar as regras de decisão.

A poda tem lugar quando o **valor esperado** do **erro da sub-árvore** é **maior que o erro** da **folha** que o fico em seu lugar.

Árvores de Decisão: Algoritmo C4.5 - Prunning

Erro Esperado (Nó) = (N - n + k -1) /(N + k)

onde :

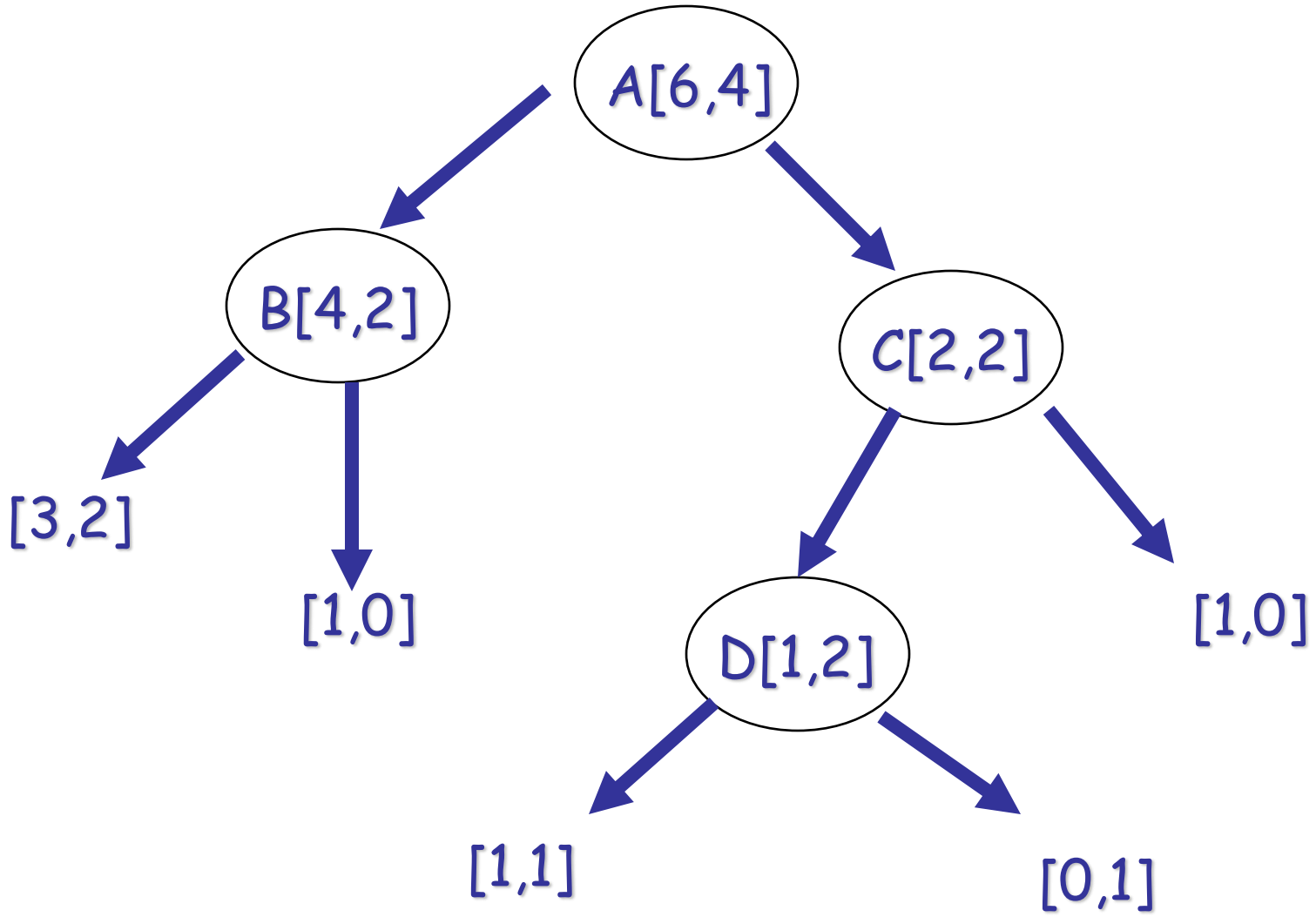
N = Número de exemplos do nó

k = Número de classes

n = Número de exemplos de N pertencentes a classe C (classe com o maior número de elementos)

Erro (Sub-arvore) = $SOMA_i$ (P_i* Erro (Nó i))

Árvores de Decisão: Algoritmo C4.5 - Prunning



Árvores de Decisão: Algoritmo C4.5 - Prunning

Erro (B) = (6 - 4 + 2 -1) / (6 + 2) = **0.375**

Erro (Filho 1) = (5 - 3 + 2 -1) / (5 + 2) = 0.429

Erro (Filho 2) = (1 - 1 + 2 -1) / (1 + 2) = 0.333

Erro (SubArv) = 5/6 * 0.429 + 1/6 * 0.333 = **0.413**

Erro (B) < Erro (SubArv) então **Pruning**

Erro (D) = (3 - 2 + 2 -1) / (3 + 2) = **0.4**

Erro (Filho 1) = (2 - 1 + 2 -1) / (2 + 2) = 0.5

Erro (Filho 2) = (1 - 1 + 2 -1) / (1 + 2) = 0.33

Erro (SubArv) = 2/3 * 0.5 + 1/3 * 0.33 = **0.444**

Erro (D) < Erro (SubArv) então **Pruning**

Árvores de Decisão: Algoritmo C4.5 - Prunning

Erro (C) = (4 - 2 + 2 -1) / (4 + 2) = **0.5**

Erro (Filho 1) = Erro (D) = 0.4

Erro (Filho 2) = (1 - 1 + 2 -1) / (1 + 2) = 0.33

Erro (SubArv) = 3/4 * 0.4 + 1/4 * 0.33 = **0.3825**

Erro (C) > Erro(SubArv) Então **Não Pruning**

Índice Gini

O índice de Gini, também conhecido como impureza de Gini, **calcula a quantidade de probabilidade de um recurso específico que é classificado incorretamente quando selecionado aleatoriamente.**

Se todos os elementos estiverem vinculados a uma única classe, ela poderá ser chamada de pura.

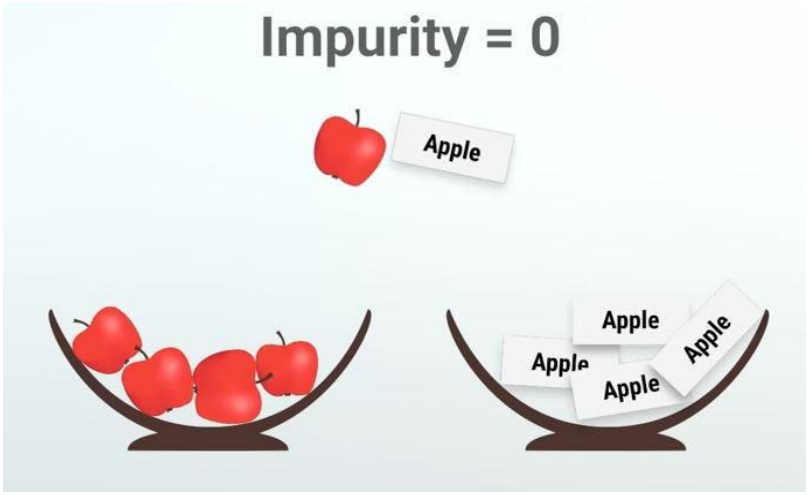
Vamos perceber o critério do Índice de Gini, como as propriedades da entropia, *o **índice de Gini varia entre os valores 0 e 1** , onde 0 expressa a pureza da classificação, ou seja todos os elementos pertencem a uma classe especificada ou apenas existe uma classe ali. E 1 indica a distribuição aleatória de elementos em várias classes. O valor de 0,5 do Índice de Gini mostra uma distribuição igualitária dos elementos em algumas classes.*

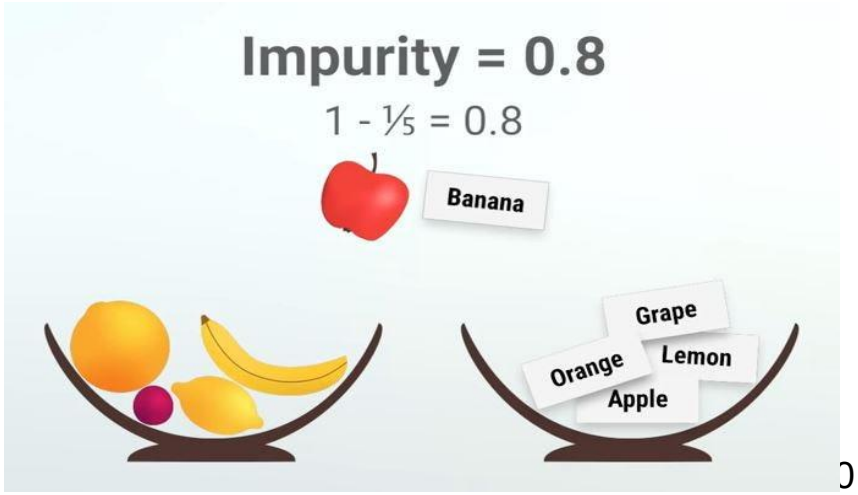
$$\text{Gini Index} = 1 - \sum_{i=1}^{n} (P_i)^2$$

Índice Gini

Imagine duas tigelas – uma cheia de maçãs e a outra cheia de rótulos. Se você escolher aleatoriamente um objeto da tigela esquerda e atribuir um rótulo à tigela direita, não há chances de estar incorreto, portanto, a impureza é 0.

Por outro lado, se você tem uma tigela com uma mistura de objetos e escolhe um objeto aleatoriamente e atribui um rótulo a ele, você tem 1 chance em 5 de estar correto e, portanto, 4 chances em 5 de estar incorreto, o que também é o valor do índice Gini.





Random Forest (floresta randômica)

Random Forest (floresta randômica)

- O Random Forest **irá criar muitas árvores de decisão, de maneira aleatória**, formando o que podemos enxergar como uma floresta, onde cada árvore será utilizada na escolha do resultado final, em uma espécie de votação

Random Forest (floresta randômica)

Métodos Ensemble

Para entender o algoritmo Random Forest, precisamos primeiramente conhecer os **métodos ensemble**, dos quais ele faz parte.

Estes métodos são construídos da mesma forma que algoritmos mais básicos, como regressão linear, árvore de decisão ou knn, por exemplo, mas possuem uma característica principal que os diferenciam, **a combinação de diferentes modelos para se obter um único resultado**.

Essa característica torna esses algoritmos mais robustos e complexos, levando a um maior custo computacional que costuma ser acompanhando de melhores resultados.

Random Forest (floresta randômica)

- **Seleção de amostras**
- Diferentemente do que acontece na criação de uma árvore de decisão simples, ao utilizar o RandomForest, o primeiro passo executado pelo algoritmo será **selecionar aleatoriamente algumas amostras dos dados de treino**, e não a sua totalidade.
- Nesta etapa é utilizado um *método de re-amostragem* (por exemplo o *bootstrap)* onde as amostras selecionadas podem ser repetidas na seleção. Com esta primeira seleção de amostras será construída a primeira árvore de decisão.

Random Forest (floresta randômica)

Seleção das variáveis para cada nó

Conforme vimos nos detalhes sobre a construção de uma árvore de decisão, para começar é preciso definir o primeiro nó da árvore (nó raiz), que será a primeira condição verificada, dando origem aos dois primeiros ramos.

Utilizando o algoritmo de entropia ou o índice Gini, será escolhida a melhor variável para compor o nó raiz, variando de acordo com o método utilizado.

No RandomForest a definição desta variável **não acontece com base em todas as variáveis disponíveis**. O algoritmo irá escolher de maneira **aleatória (random)** duas ou mais variáveis, e **então realizar os cálculos com base nas amostras selecionadas**, para definir qual dessas variáveis será utilizada no primeiro nó.

Random Forest (floresta randômica)

Seleção das variáveis para cada nó

Para escolha da variável do próximo nó, novamente serão escolhidas duas (ou mais) variáveis, **excluindo as já selecionadas anteriormente**, e o processo de escolha se repetirá. Desta forma a árvore será construída até o último nó. A quantidade de variáveis a serem escolhidas pode ser definida na criação do modelo.

É evidente que este não é o melhor método para construção de uma árvore de decisão. O algoritmo pode, sem querer, selecionar as duas piores variáveis na primeira seleção, escolhendo uma variável péssima para o primeiro nó. Mas como serão construídas muitas árvores, essa estratégia se torna poderosa.

Random Forest (floresta randômica)

Construção das próximas árvores

Na construção da próxima árvore, os dois processos anteriores se repetirão, levando a criação de uma nova árvore. Provavelmente **essa árvore será diferente da primeira**, pois tanto na seleção das amostras, quanto na seleção das variáveis, **o processo acontece de maneira aleatória**.

Podemos construir quantas árvores quisermos, sendo que quanto mais árvores criadas, melhor serão os resultados do modelo, até determinado ponto, onde uma nova árvore não conseguirá levar a uma melhora significativa no desempenho do modelo.

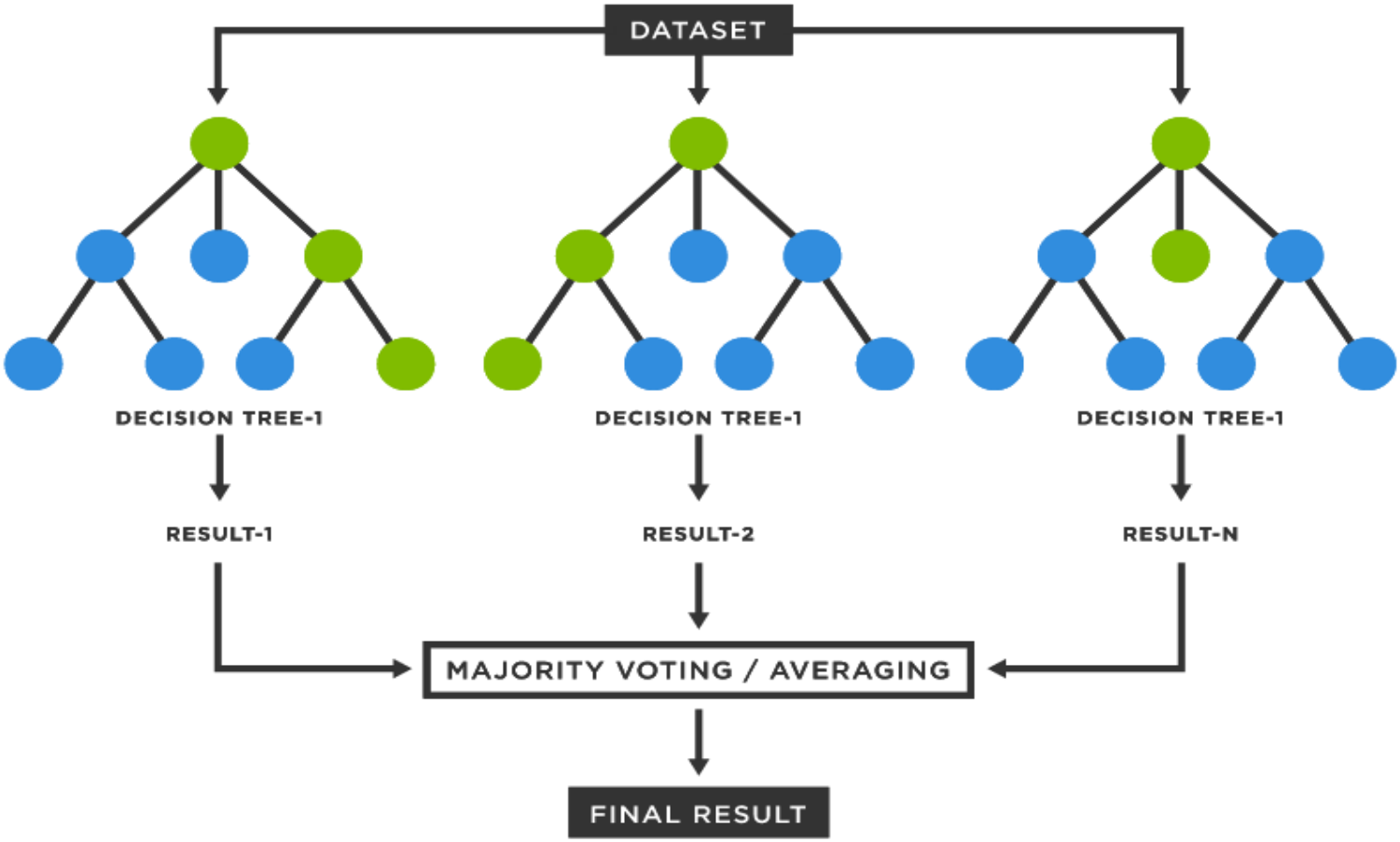
É importante lembrarmos que quanto mais árvores forem criadas, maior será o tempo de criação do modelo.

Random Forest (floresta randômica)

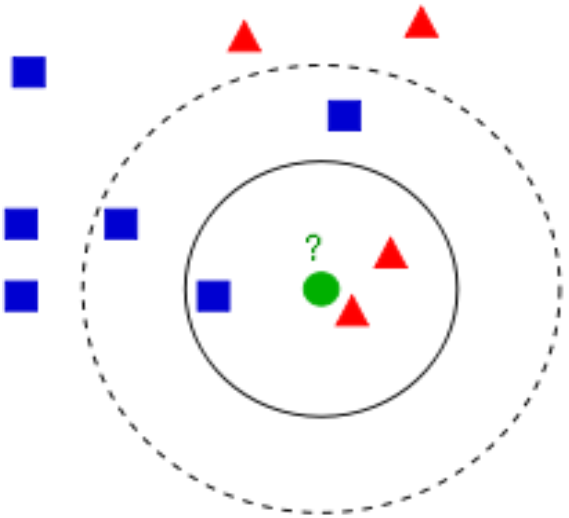
Prevendo novos valores

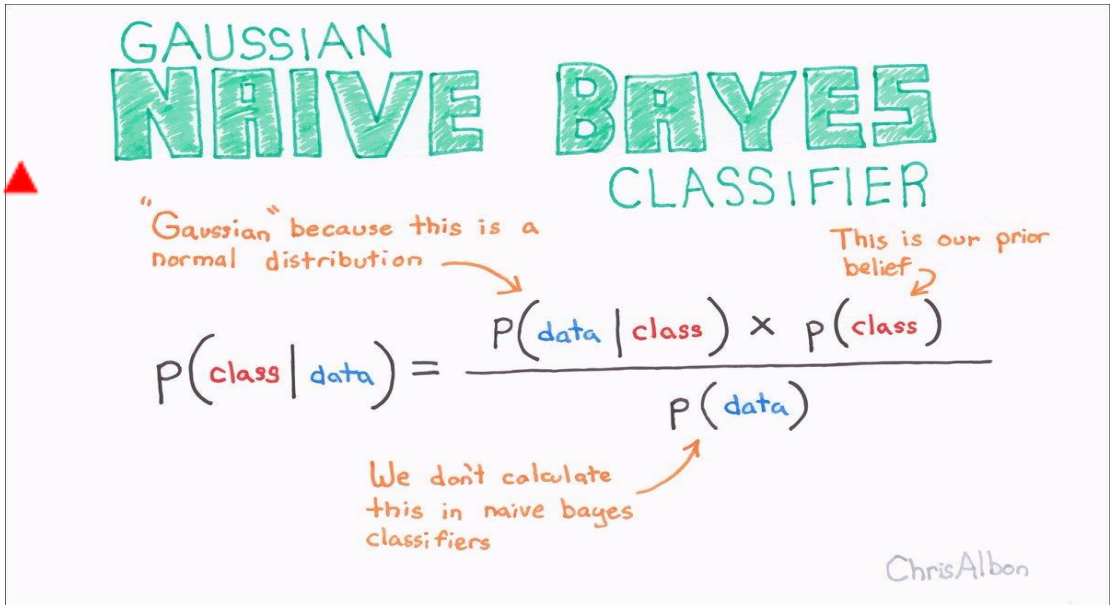
Com o modelo de machine learning devidamente criado, podemos apresentar novos dados e obter o resultado da previsão. **Cada árvore criada irá apresentar o seu resultado**, sendo que em problemas de regressão será realizada a média dos valores previstos, **e esta média informada como resultado final**, e em problemas de classificação **o resultado que mais vezes foi apresentado será o escolhido**.

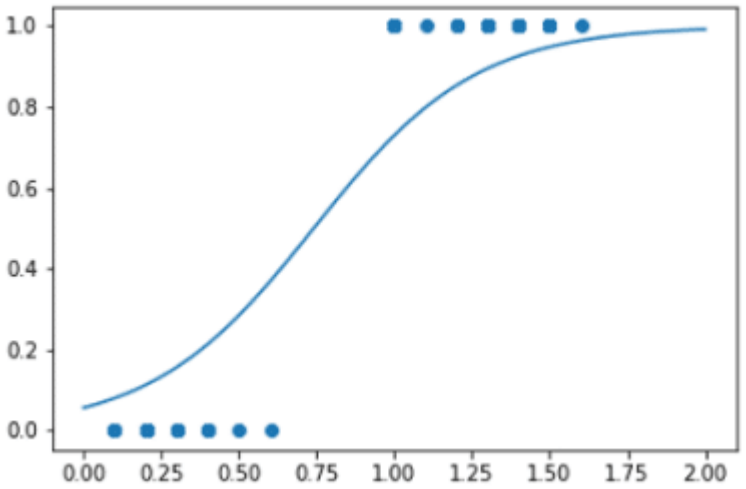
Random Forest (floresta randômica)

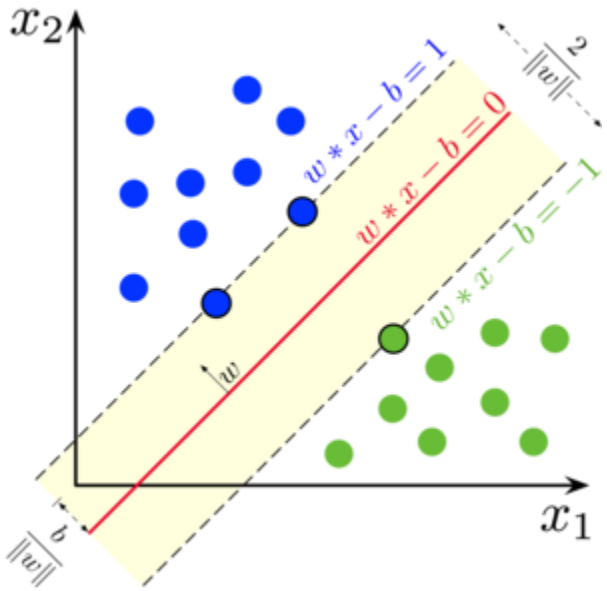


3. Abordagem Estatística









Classificadores Bayesianos: Naive Bayes

É uma técnica de classificação baseada no teorema de Bayes com uma suposição de independência entre preditores.

Um classificador Naive Bayes assume que a presença de uma característica particular em uma classe não está relacionada com a presença de qualquer outra característica.

Classificadores Bayesianos: Naive Bayes

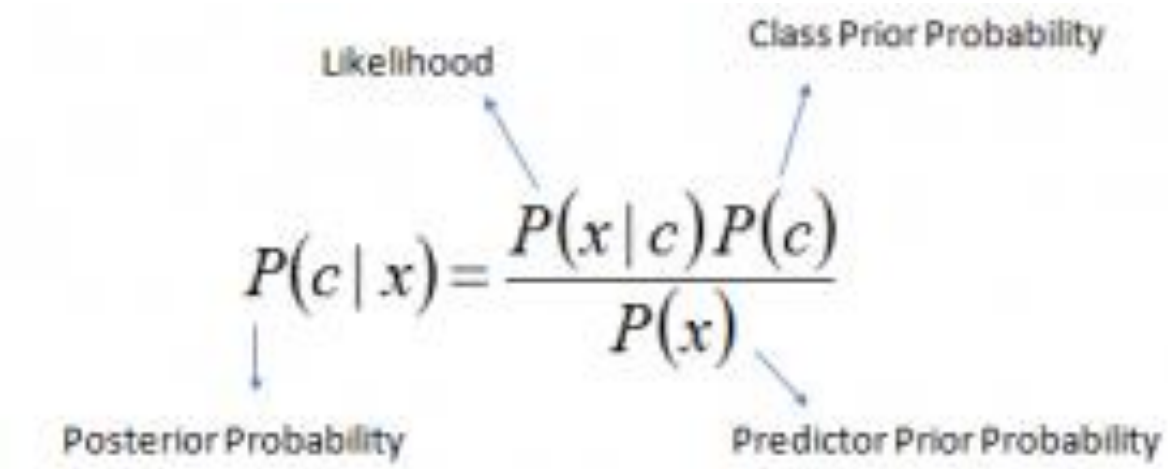
Por exemplo, uma fruta pode ser considerada como uma maçã se é vermelho, redondo, e cerca de 3 centímetros de diâmetro.

Mesmo se essas características dependem umas das outras ou da existência das outras características, um classificador Naive Bayes consideraria todas essas propriedades para contribuir de forma independente para a probabilidade de que este fruto é uma maçã.

Classificadores Bayesianos: Naive Bayes

O modelo bayesiano é fácil de construir e útil para conjuntos de dados muito grandes. .

O teorema de Bayes fornece uma maneira de calcular $P(c \mid x)$, onde **c** é a classe (alvo) e **x** é o atributo preditor.



$$P(c \mid X) = P(x_1 \mid c) \times P(x_2 \mid c) \times \cdots \times P(x_n \mid c) \times P(c)$$

Classificadores Bayesianos: Naive Bayes

Probabilidade condicional

$P(B|A) = P(A \cap B) / P(A)$ (1)

$P(A|B) = P(A \cap B) / P(B)$ (2)

De (2) podemos ter: $P(A \cap B) = P(A|B) \, . \, P(B)$ (3)

Substituindo (3) em (1) chegamos a regra de Bayes:

$P(B|A) = P(A|B) \, . \, P(B) / P(A)$ (regra de Bayes)

Classificadores Bayesianos: Naive Bayes

Generalização da regra de Bayes:

$= P(A_1 = a_1 \cap \ldots \cap A_k = a_k \mid B = b) * P(B = b) / P(A_1 = a_1 \cap \ldots \cap A_k = a_k)$

Considerando independência entre os atributos:

$= P(A_1 = a_1 \mid B = b) *\ldots P(A_k = a_k \mid B = b) * P(B = b) /$

$P(A_1 = a_1) *\ldots P(A_k = a_k)$

Classificadores Bayesianos: Naive Bayes

Dia	Aspecto	Temperatura	Umidade	Vento	Decisão
1	Sol	Quente	Alta	Fraco	N
2	Sol	Quente	Alta	Forte	N
3	Nublado	Quente	Alta	Fraco	S
4	Chuva	Agradável	Alta	Fraco	S
5	Chuva	Fria	Normal	Fraco	S
6	Chuva	Fria	Normal	Forte	N
7	Nublado	Fria	Normal	Forte	S
8	Sol	Agradável	Alta	Fraco	N
9	Sol	Fria	Normal	Fraco	S
10	Chuva	Agradável	Normal	Fraco	S
11	Sol	Agradável	Normal	Forte	S
12	Nublado	Agradável	Alta	Forte	S
13	Nublado	Quente	Normal	Fraco	S
14	Chuva	Agradável	Alta	Forte	N

Classificadores Bayesianos: Naive Bayes

Qual será a decisão, se o dia estiver com sol, temperatura fria, umidade alta e vento forte ?

P(Jogar = S / Aspecto = Sol e Temperatura = Fria e Umidade = Alta e Vento = Forte) = ?

$$\frac{P(Sol/S) * P(Fria/S) * P(Alta/S) * P(Forte/S) * P(S)}{P(Sol) * P(Fria) * P(Alta) * P(Forte)}$$

(2/9 * 3/9 * 3/9 * 3/9 * 9/14) **/** (5/14 * 4/14 * 7/14 * 6/14) = 0,0053/0,028 = **0,189**

Classificadores Bayesianos: Naive Bayes

P(Jogar = N / Aspecto = Sol e Temperatura = Fria e

Umidade = Alta e Vento = Forte) = ?

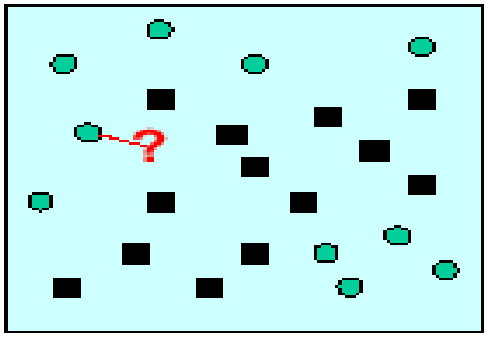
$$\frac{P(Sol/N) * P(Fria/N) * P(Alta/N) * P(Forte/N) * P(N)}{P(Sol) * P(Fria) * P(Alta) * P(Forte)}$$

(3/5 * 1/5 * 4/5 * 3/5 * 5/14) / (5/14 * 4/14 * 7/14 * 8/14) = 0,0206/0,028 = **0,734**

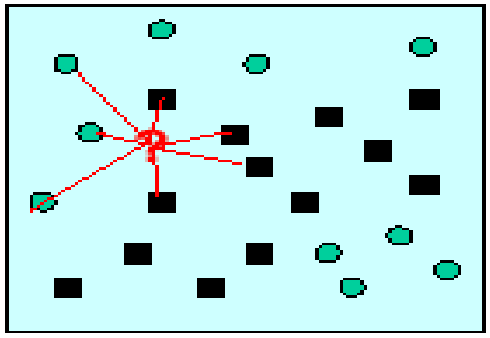
K- Nearest Neighbor

Exemplo:

A classificação de **?** (**F(?)**), será a classificação de X_i ($F(X_i)$), onde X_i é a instancia mais próxima de **?**.



1-nearest neighbor



7-nearest neighbor

Se **k=1**, na figura **?** seria classificado como **O**

Se **k=7**, na figura **?** seria classificado como **#**

Outra alternativa, do algoritmo, é dar peso a contribuição de cada um dos k-vizinhos de acordo com sua distancia.

K- Nearest Neighbor

x = < idade(x), altura(x), peso(x), classe(x)>, onde **classe** pode ser "sim", "não")

Exemplo: **joão** = (<36, 1.80, 76>, ???) **a ser classificado**

josé = (<30, 1.78, 72>, sim)

maria = (<25, 1.65, 60>, sim)

anastácia = (<28, 1.60, 68>, não)

Calculo da distância euclidiana: $d(x,y) = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + ... + (x_p - y_p)^2}$

d(joão,josé) = $[(36-30)^2 + (1.80-1.78)^2 + (76-72)^2]^{1/2}$

= $(36+0.0004+16)^{1/2}$= **7,21**

d(joão,maria) = $(121+0.0225+256)^{1/2}$ = 19,41

d(joão, anastácia) = $(64+0.04+64)^{1/2}$ = 11,32

Regressão Logística

A **regressão logística** é um modelo de regressão amplamente utilizado para tarefas de classificação, é usada para encontrar a probabilidade de sucesso/falha de um evento, usada quando a variável dependente for binária (0/1, Verdadeiro / Falso, Sim / Não).

Por exemplo, um banco quer construir um modelo que preveja quais de seus clientes irão *inadimplir* em seus empréstimos, em função de sua "pontuação de crédito".

O atributo meta, pode assumir os valores 0 ou 1. Podemos formular o problema como uma probabilidade:

- se a probabilidade de inadimplir for **> = 0,5**, o cliente será considerado inadimplente (valor 1)
- se a probabilidade for **< 0,5**, o cliente será considerando adimplente (valor 0)

Regressão Logística

No modelo de regressão linear, definimos uma variável dependente **y** que é uma função de variáveis independentes.

Em nosso caso: **Y = b + a X**,

onde "**X**" é a variável independente (pontuação de crédito), "**a**" é o coeficiente de X, "**b**" o intercepto, e "**Y**" a variável dependente.

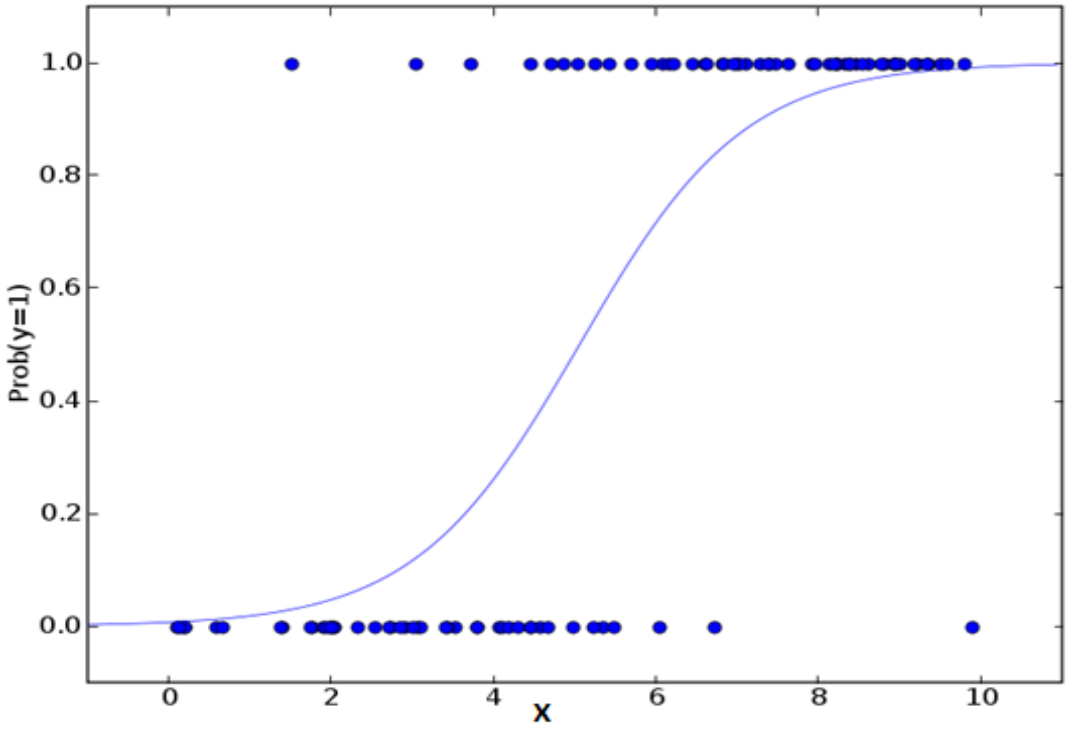
No exemplo em questão, a variável dependente (Y) é a estimativa da probabilidade de o cliente inadimplir, ou seja,

Probabilidade (cliente ser inadimplir) = Y = b + a X.

Regressão Logística

Para que o valor da probabilidade seja sempre um valor entre 0 e 1, utilizamos uma função logística (ou sigmoide) = $e^y / (1 + e^y)$, então Probabilidade (cliente inadimplir) = $e^y / (1 + e^y)$,

onde Y = b + a X, mais pode ser expandido para **n** variáveis dependentes



Avaliação do algoritmo de aprendizado

Na matriz de confusão, temos 2 tipos de erros:

- Erro Tipo I - Falso Positivo: classificar um colaborador que deixa a empresa o modelo identifica como “não vai deixar a empresa”.

De uma perspectiva de negócios, este é o erro menos desejável, já que um colaborador com caraterísticas (muito provável) para abandonar o serviço **não é detectado**, afetando a empresa, já que estamos perdendo o cliente.

- Tipo II Erro - Falso Negativo: classificar um colaborador com caraterísticas de ficar na empresa, como um provável ”deixar a empresa ” (ser rotativo) .

Na perceptiva dos negócios, o Erro Tipo II é aceitável, pois não afeta a empresa;

4. Abordagem Biológica: Redes Neurais Artificiais

Redes Neurais Artificiais: Conceitos

As **Redes Neurais Artificias** são modelos computacionais de processamento de dados inspirados nos **neurônios biológicos** e na **estrutura paralela do cérebro** , isto é, na forma como o cérebro humano funciona.

As **RN**, apresentam algumas características muito interessantes:

Redes Neurais Artificiais: Conceitos

Características:

- As Redes aprendem por experiência , não necessitando explicitar os algoritmos para executar uma determinada tarefa **(aprendizado)**;

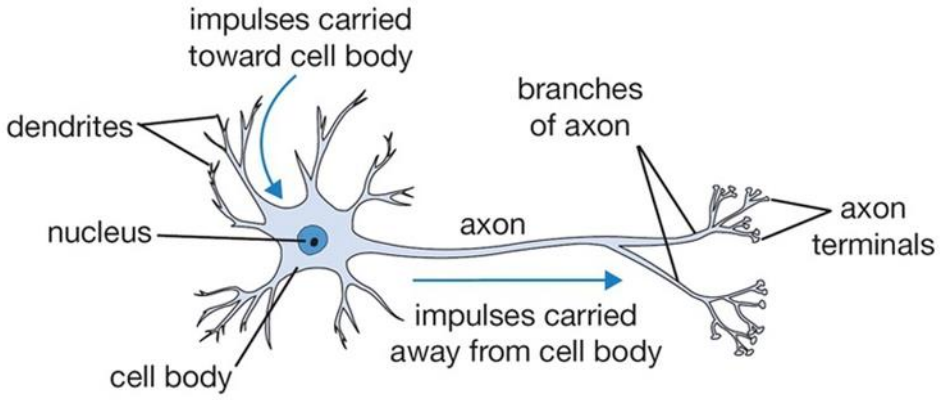
- As redes são capazes de generalizar o seu conhecimento a partir de exemplos anteriores **(generalização);**

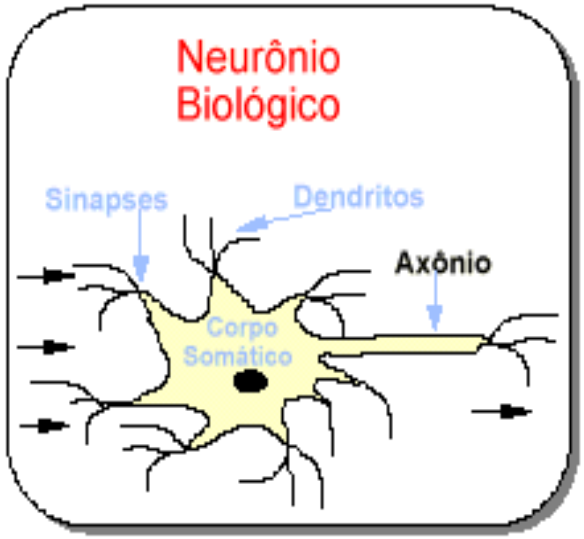
- As redes são capazes de lidar com **ruídos** e **distorções** respondendo corretamente aos novos padrões. Isto é, a "alteração de caraterísticas por ruídos ou distorções" não causa o mal funcionamento da rede neural (**robustez**).

Redes Neurais Artificiais: Conceitos

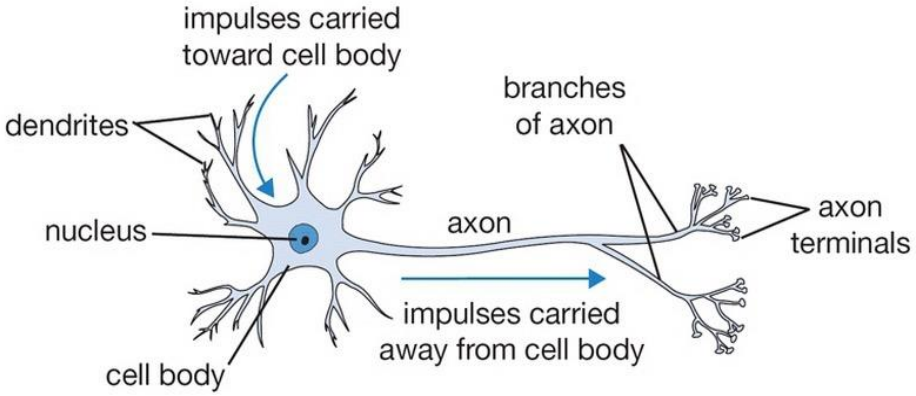
O neurônio biológico é entendido como sendo basicamente o dispositivo elementar do sistema nervoso, que possue **muitas entradas** e **uma saída**.

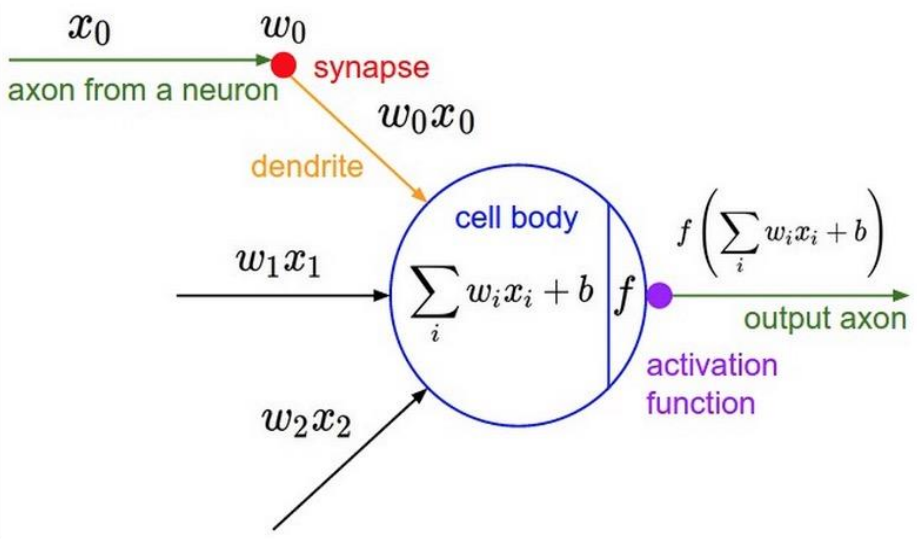
A partir do corpo celular, ou **soma**, (o centro dos processos metabólicos da célula nervosa) projetam-se extensões filamentares, os **dendritos**, e o **axônio**.

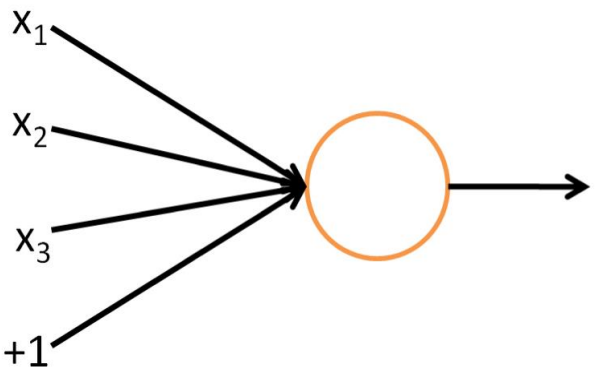




Redes Neurais Artificiais: Conceitos







Neurónio = processador de informação: recebe e envia impulso a milhares de neurónios

- **dendritos**: canais de entrada
- **corpo celular**: órgão de cômputo (soma ponderada, função não linear)
- **axónio**: canal de saída e distribuição a outros neurónios

Redes Neurais Artificiais: Conceitos

Breve Histórico

- McCulloch & Pitts (1943): modelo computacional para o neurônio artificial. Não possuía capacidade de aprendizado;
- Hebb (1949): modelo de aprendizado (Hebbian Learning Rule);
- Rosenblatt (1957): Perceptron, com grande sucesso em certas aplicações e problemas em outras aplicações aparentemente similares;
- Minsky & Papert (Perceptrons 1969): prova matemática de que as redes Perceptron são incapazes de solucionar problemas simples tipo OU-EXCLUSIVO;
- Rumelhart (início da década de 80): novos modelos que superaram os problemas dos Perceptrons.

Redes Neurais Artificiais: Perceptron

Vamos agora explicar a matemática neste modelo de neurônio.

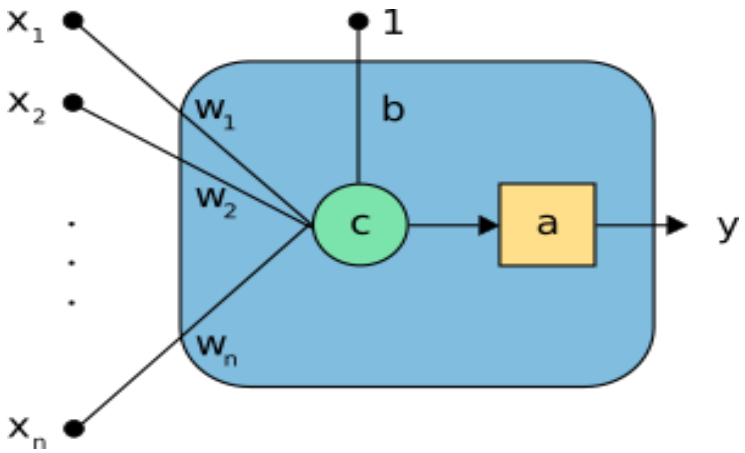
Elementos do Modelo Perceptron

Um neurônio é o principal componente de uma rede neural, e o perceptron é o modelo mais usado.

A figura é uma representação gráfica de um perceptron.

No neurônio podemos ver os seguintes elementos:

- As **entradas** (x1 , ..., xn).
- O viés **b** e os **pesos sinápticos** (w1 , ..., wn).
- A função de **soma** ou **combinação**, c (·).
- A função de **ativação** a (·).
- A **saída y**.



Redes Neurais Artificiais: Perceptron

Função de Soma

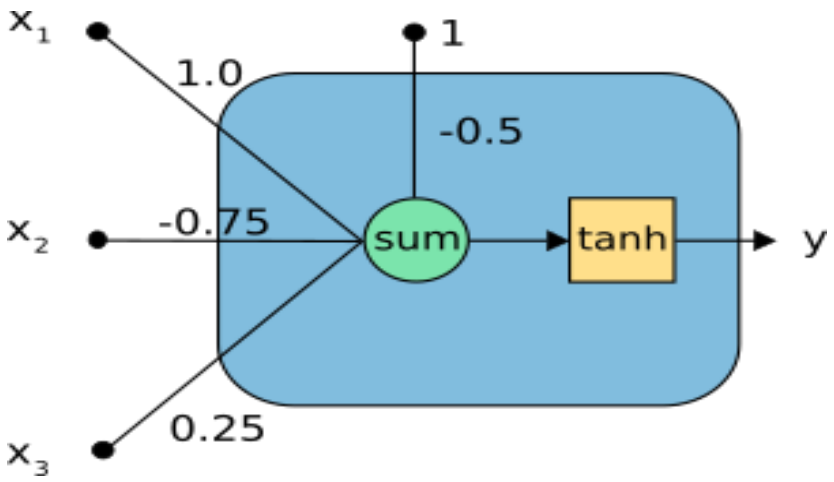
A função soma pega o vetor de entrada x para produzir um valor.

No perceptron, a "soma" é calculada como o viés mais a combinação linear dos pesos sinápticos e das entradas:

$$c = b + \sum w_i \cdot x_i \qquad i = 1, \ldots, n.$$

Considere o neurônio do nosso exemplo, o valor de combinação deste perceptron para um vetor de entrada x = (-0.8, 0.2, -0.4) é:

$$c = (1{,}0 * -0{,}8) + (-0{,}75 * 0{,}2) + (0{,}25 * -0{,}4) + (-0{,}5 * 1) = -1{,}55$$

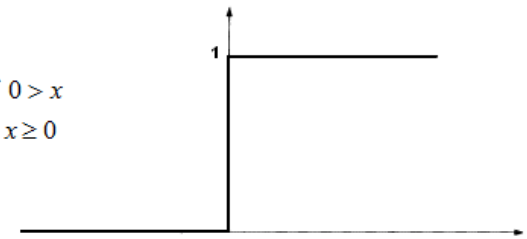


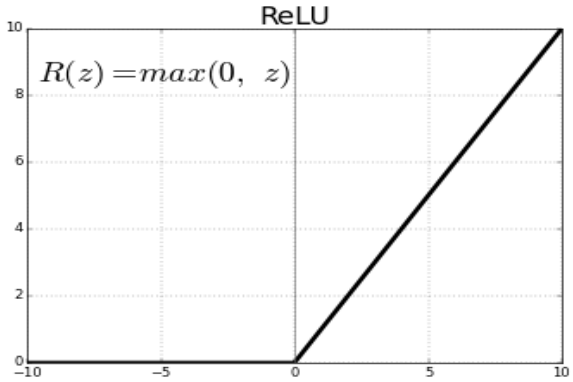
Redes Neurais Artificiais: Perceptron

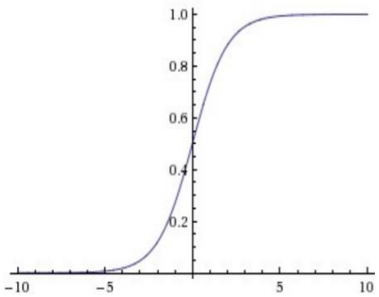
Função de Ativação

A função de ativação irá definir a saída do neurônio em termos de sua combinação. Na prática, podemos considerar muitas funções úteis de ativação. As mais utilizadas são a logística (sigmoide), a tangente hiperbólica e as funções lineares.

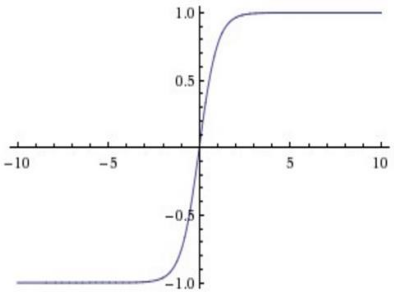
$$f(x) = \begin{cases} 0 \text{ if } 0 > x \\ 1 \text{ if } x \geq 0 \end{cases}$$







Sigmoid



tanh

Redes Neurais Artificiais: Perceptron

Função de Ativação

Portanto, a expressão final da saída de um neurônio como uma função da entrada é: $y = a\,(b + w \cdot x)$.

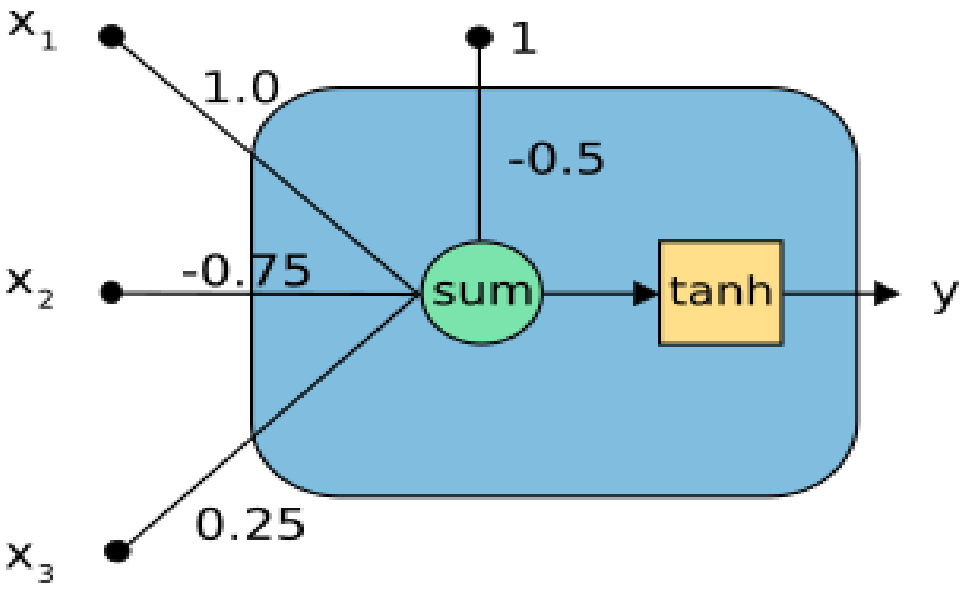
Considere o perceptron do nosso exemplo.

Se aplicarmos uma entrada x = (-0.8, 0.2, -0.4), a saída y será a seguinte:

y = tanh ((1,0 * -0,8) + (-0,75 * 0,2) + (0,25 * -0,4) + (-0,5 * 1))=

tanh (-1,55) = -0,91

.



Redes Neurais Artificiais

Exemplo da aula anterior:

210 m^2 - 4 dormitórios - R$ 399 mil

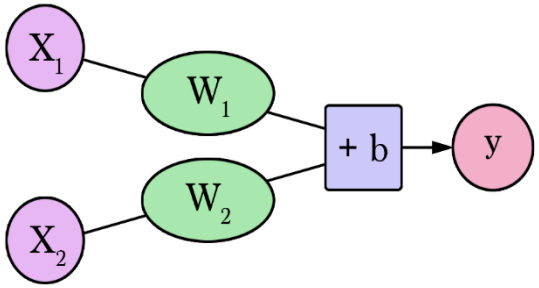
160 m^2 - 2 dormitórios - R$ 329 mil

240 m^2 - 3 dormitórios - R$ 369,00 mil

Então:

Saída = P_1 * Tamanho (m^2) + P_2 * Dormitórios (número) + intercepto;

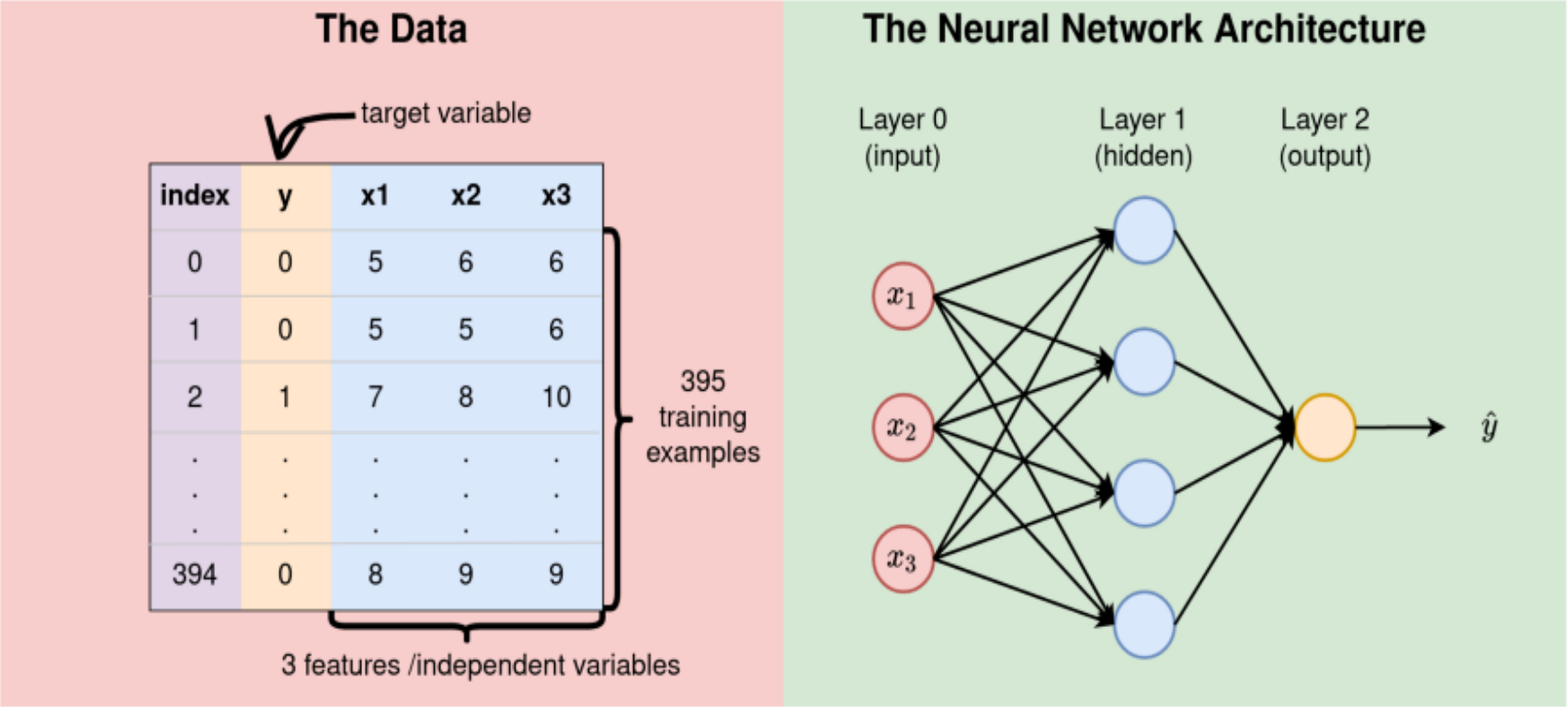
Y = W1 * X1 + W2 * X2 + viés



$$y = W_1 x_1 + W_2 x_2 + b$$

Redes Neurais Artificiais: Arquiteturas

Exemplo de cálculo:



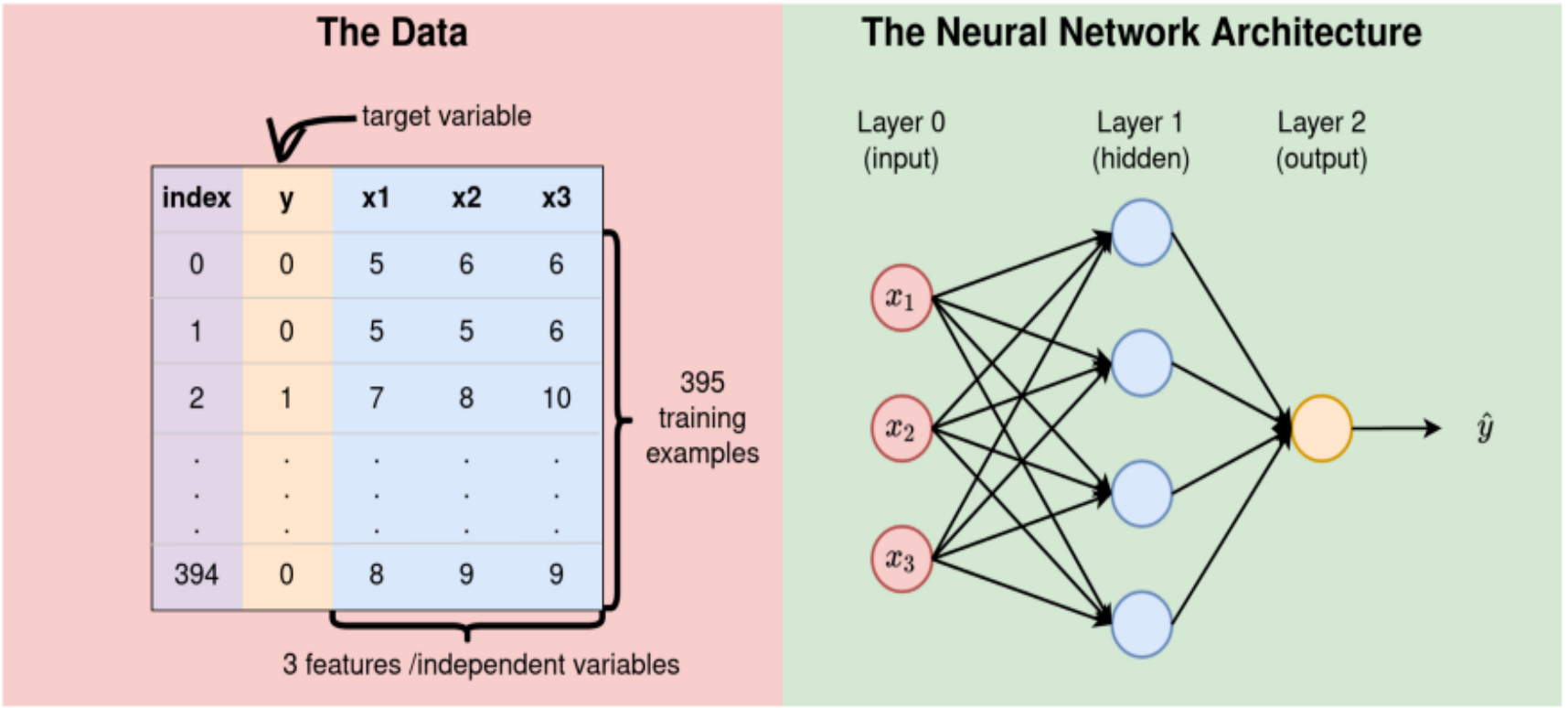
Rede neural com 3 entradas, duas camadas de neurônios (camada oculta com 4 neurônios) e a saída (1 neurônio).

Redes Neurais Artificiais: Arquiteturas

2. Os dados:

Os dados que usaremos neste exemplo contém **3 variáveis X1, X2 e X3** (por isso escolhemos 3 neurônios na camada de entrada da arquitetura) **com um target,** (vamos chamá-lo de **y**) que é 0 ou 1.

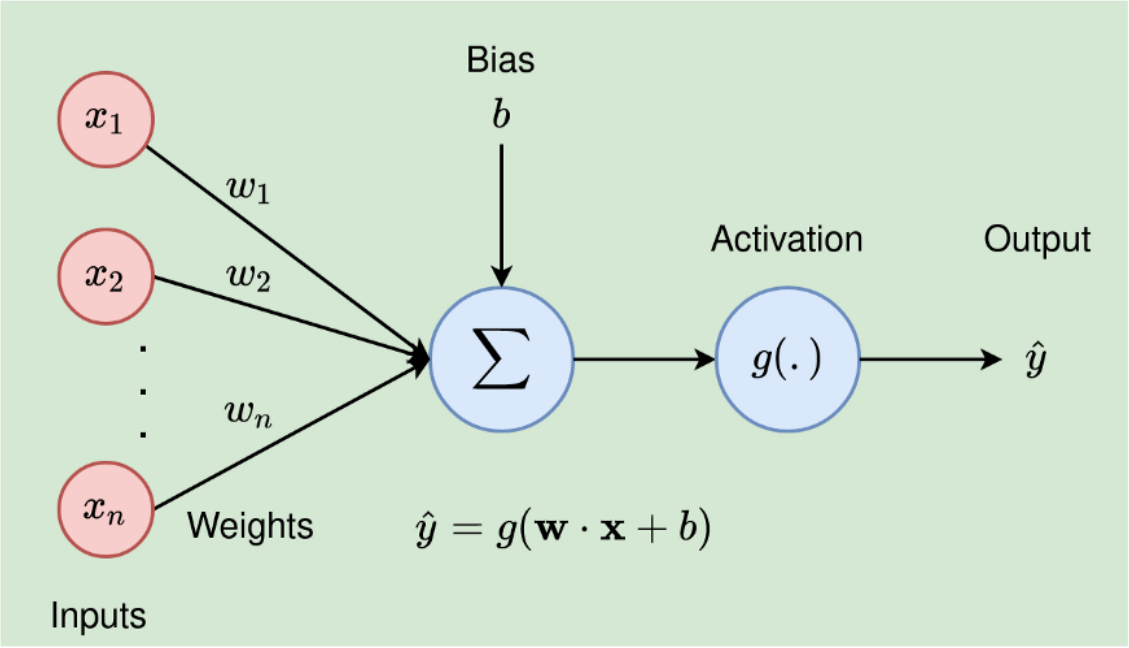
Isso significa que estamos lidando com um problema de classificação binária.



Redes Neurais Artificiais: Arquiteturas

3. Inicialização do parâmetros

Inicializaremos os pesos aleatoriamente com valores entre 0 e 1, enquanto o bias sempre será inicializado com 0.

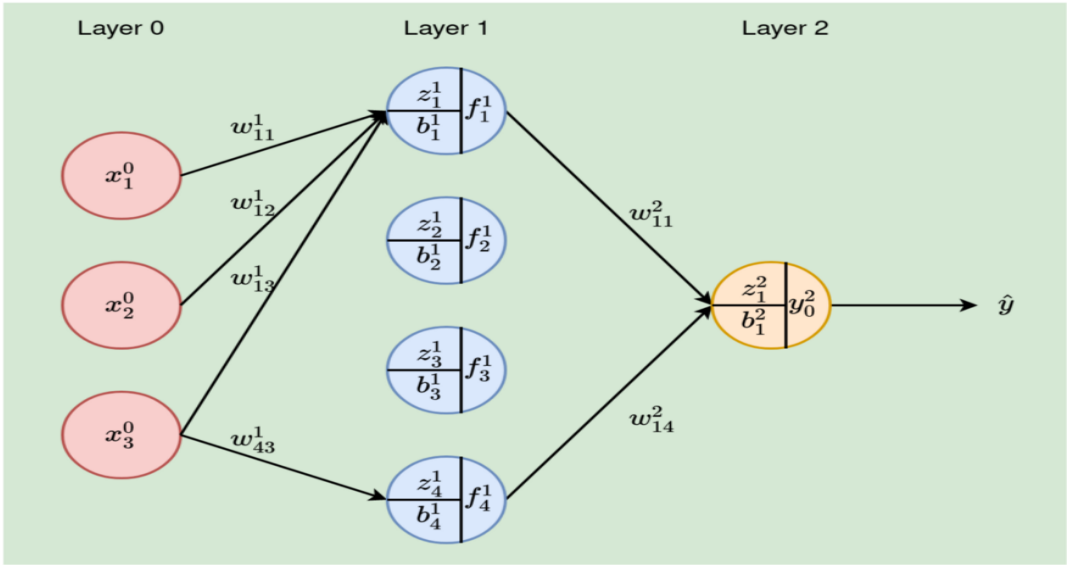


Redes Neurais Artificiais: Arquiteturas

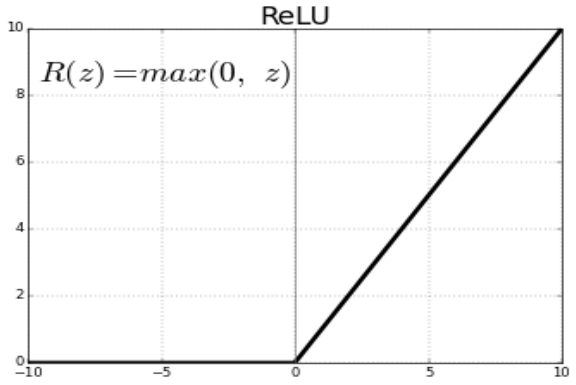
4. Cálculos nos Neurônios da Camada Oculta

Para o primeiro neurônio na camada oculta, precisamos calcular f^1_1 (resultado da função de ativação g^1), o que significa que precisamos de valores iniciais para os três pesos w^1_{11}, w^1_{12} e w^1_{13}.

Vamos inicializá-los da seguinte forma w^1_{11}=0,3, w^1_{12}=0,8 e w^1_{13}=0,62. E como dito anteriormente, vamos definir o viés, b^1_1=0. Vamos introduzir uma função de ativação chamada Rectified Linear Unit (ReLU) que usaremos como função g^1.



$$f(x) = \max(x, 0) = \begin{cases} 0, & \text{for } x < 0 \\ x, & \text{for } x > 0 \end{cases}$$



Redes Neurais Artificiais: Arquiteturas

4. Cálculos nos Neurônios da Camada Oculta

Cálculos no primeiro neurônio da camada=1.

$w^1_{11} * x_1 + w^1_{12} * x_2 * + w^1_{13} * x_3 + b^1_1 * 1 =$

$0.3 * 7 + 0{,}8 * 8 + 0{,}62 * 10 + 0 * 1 = 14{,}7$

ReLu (14,7) = max(14,7;0)=14,7

então: saída do primeiro neurônio da camada 1 = $f^1_1 = 14{,}7$

Podemos calcular f^1_2, f^1_3 e f^1_4 da mesma maneira, dado que:

$w^1_{21}=0{,}9$, $w^1_{22}=0{,}1$, $w^1_{23}=0{,}1$ e $b^1_2=0$, para o cálculo de f^1_2,

$w^1_{31}=0{,}7$, $w^1_{32}=0{,}2$, $w^1_{33}=0{,}4$, e $b^1_3=0$, para o cálculo de f^1_3, e,

$w^1_{41}=0{,}01$, $w^1_{42}=0{,}5$, $w^1_{43}=0{,}2$ e $b^1_4=0$, para o cálculo de f^1_4.

Teremos $f^1_2=8{,}1$, $f^1_3=10{,}5$ e $f^1_4=6{,}07$.

Redes Neurais Artificiais: Arquiteturas

5. Cálculos na última camada

Na camada final, vamos inicializar os parâmetros (pesos) e bias: w^2_{11}=0,58, w^2_{12}=0,1, w^2_{13}=0,1, w^2_{14}=0,42 e b^2_1=0.

Sobre a função de ativação nesta camada — como estamos lidando com classificação binária, podemos usar a função sigmoide / logística porque esta função gera valores entre 0 e 1 e, portanto, pode ser interpretada como uma probabilidade de previsão de uma classe. A função sigmoide é definida como: $f(x) = \frac{1}{1+e^{-x}} \in (0,1)$

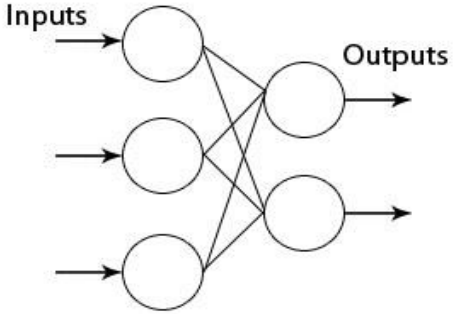
Por tanto, os cálculos na última camada da rede:

0,58 * 14,7 + 0,1 * 8,1 + 0,1 * 10,5 + 0,42 * 6,07 + 1 * 0 = 12,9

Sigmoide (12,9) = 1 / (1 + exp (-12,9)) = 0, 99

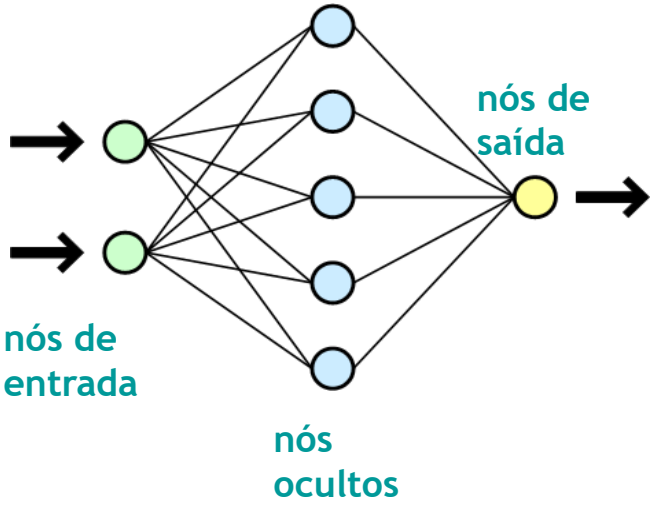
Redes Neurais Artificiais: Arquiteturas

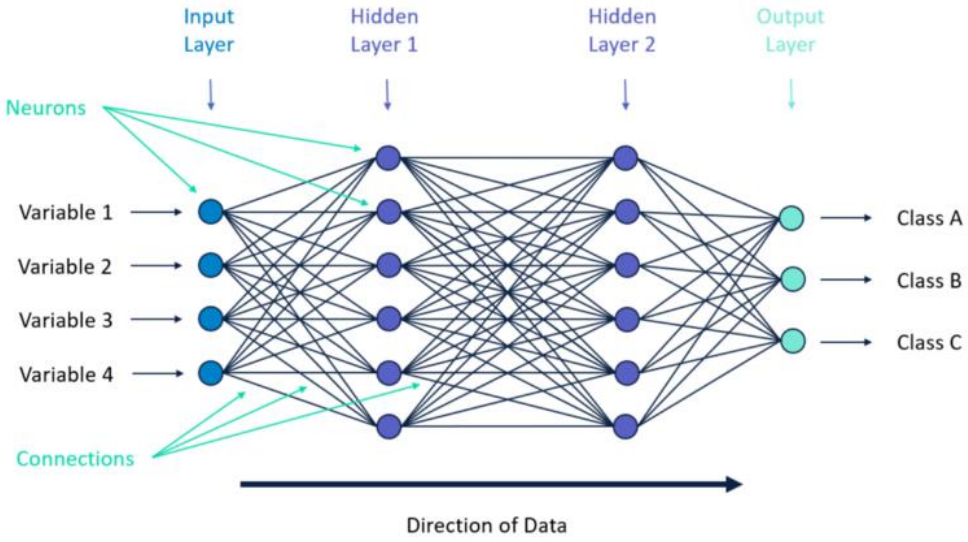
Modelos mais populares:
Perceptron



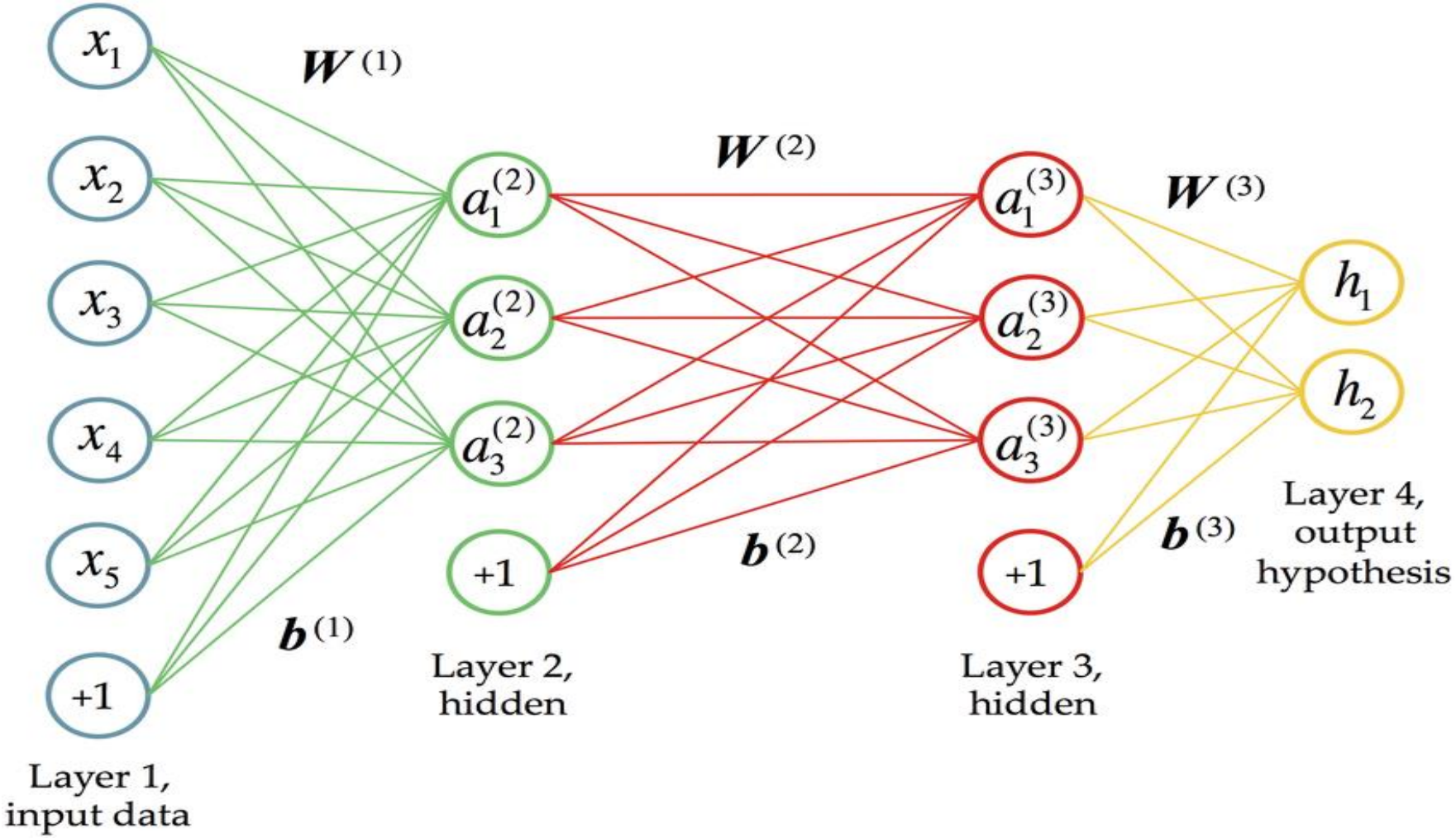
Modelos mais populares:

Perceptron multicamadas



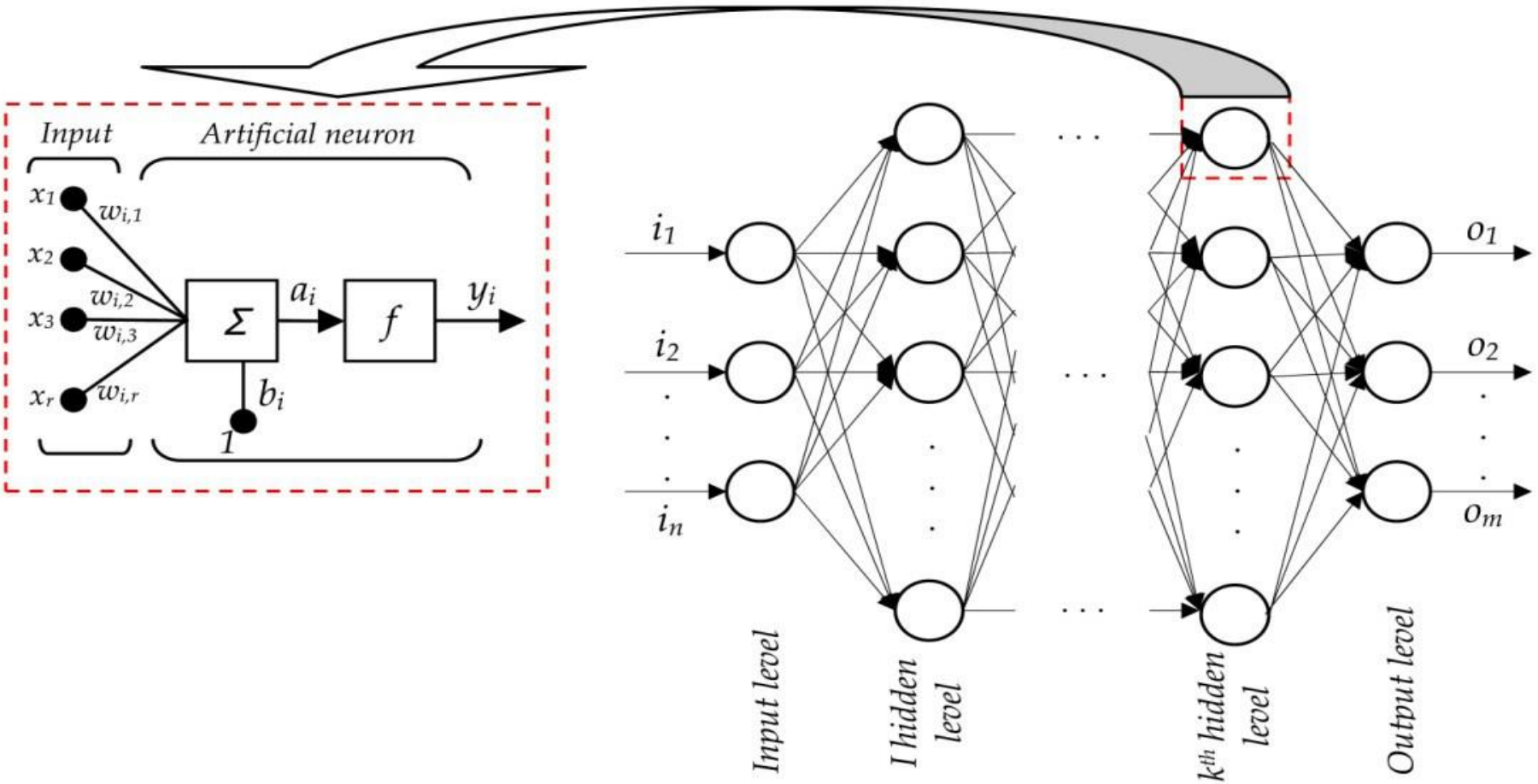


Redes Neurais Artificiais: Arquiteturas



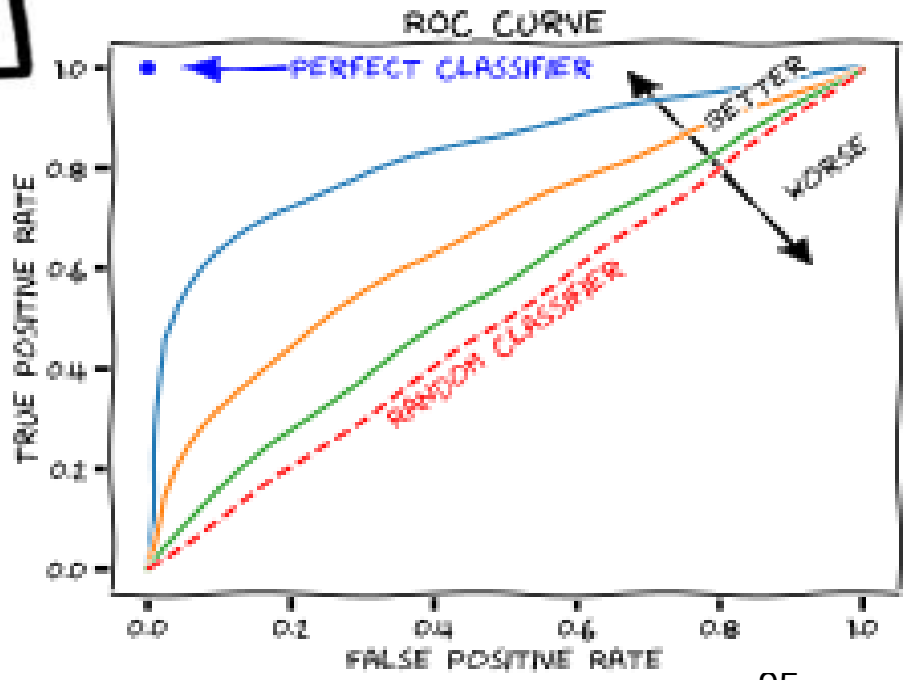
Redes Neurais Artificiais: Arquiteturas

Modelos mais populares: **Perceptron multicamadas**



5. Modelos Analíticos: Avaliação e Validação

		Predicted: True	Predicted: False
Actual	True	True Positives	False Negatives
	False	False Positives	True Negatives



Modelos Analíticos: Avaliação e Validação

Um algoritmo de aprendizagem deve ser avaliado tendo em conta o seu desempenho (capacidade de generalização) naqueles exemplos que não foram usados para construir o classificador, isto é,

o desempenho de um classificador é medido em termos da sua capacidade preditiva nos futuros exemplos .

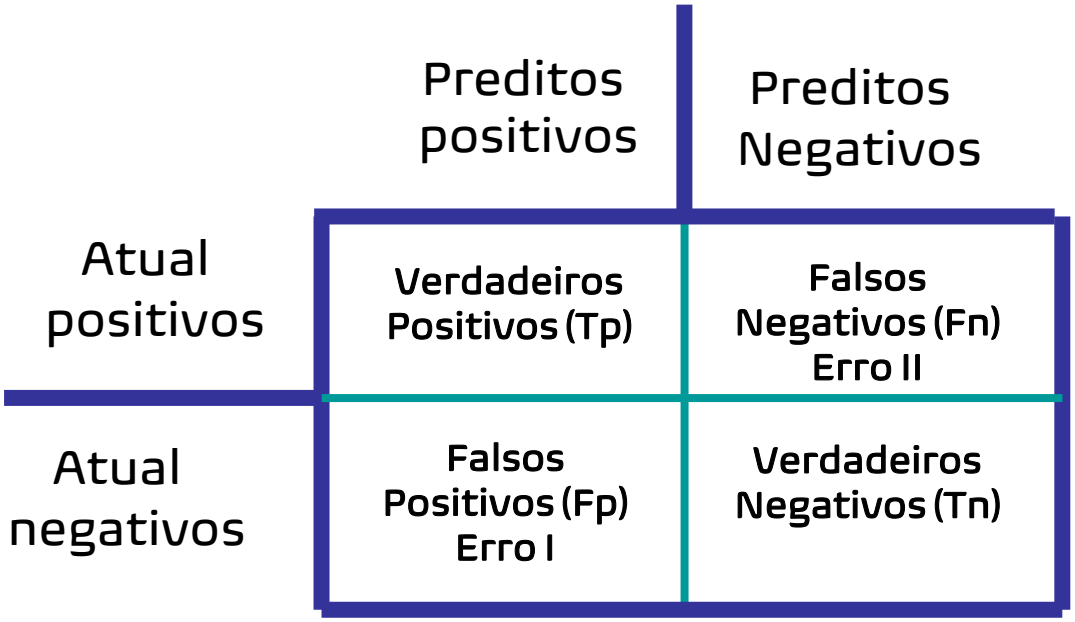
Modelos Analíticos: Avaliação e Validação

Taxa de erro de um classificador: proporção de exemplos incorrectamente classificados: **Taxa de Erro = Erros / Total**

Ideia básica: Particionar o conjunto de dados disponível em dois conjuntos:

- **conjunto de treino**: exemplos que são usados pelo algoritmo de aprendizagem para construir o classificador;
- **conjunto de teste**: exemplos que são usados para estimar a taxa de erro.

Modelos Analíticos: Avaliação e Validação



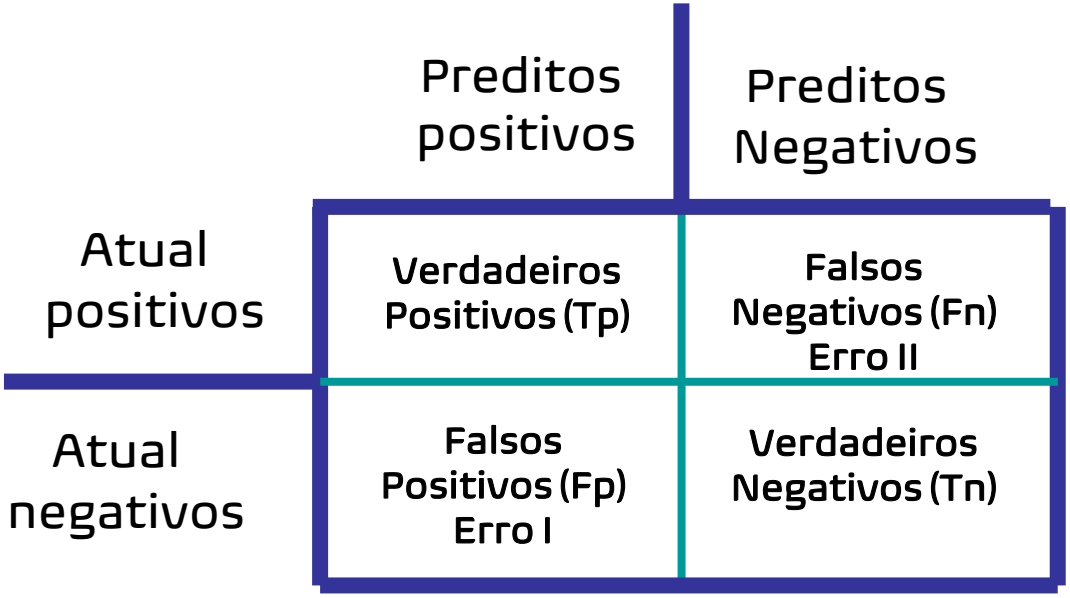
Verdadeiro positivo (Tp): número de previsões corretas dos itens positivos (item positivo predito como positivo).

Falsos negativos (Fn): número de previsões incorretas dos itens positivos (item positivo predito como negativo).

Falso positivo (Fp): número de previsões incorretas dos itens negativos (item negativo predito como positivo).

Verdadeiro negativo (TN): número de previsões corretas dos itens negativos (item negativo predito como negativo)

Modelos Analíticos: Avaliação e Validação



Recall: de todas as situações (positivas/negativas) esperadas, quantas estão corretas.

Recall (sensibilidade) = Tp / (Tp + Fn)

Recall Negativos = Tn / (Tn + Fp)

Precisão = Tp / (Tp + Fp)

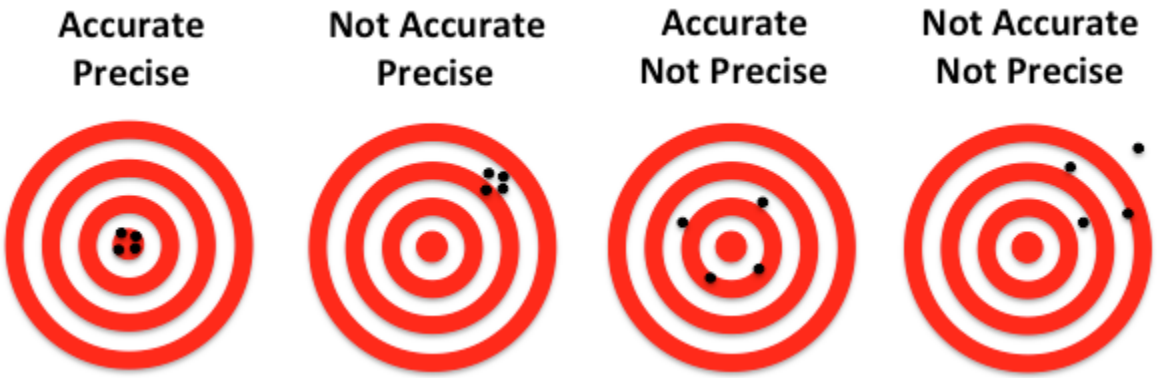
Precisão Negativos = Tn / (Fn + Tn)

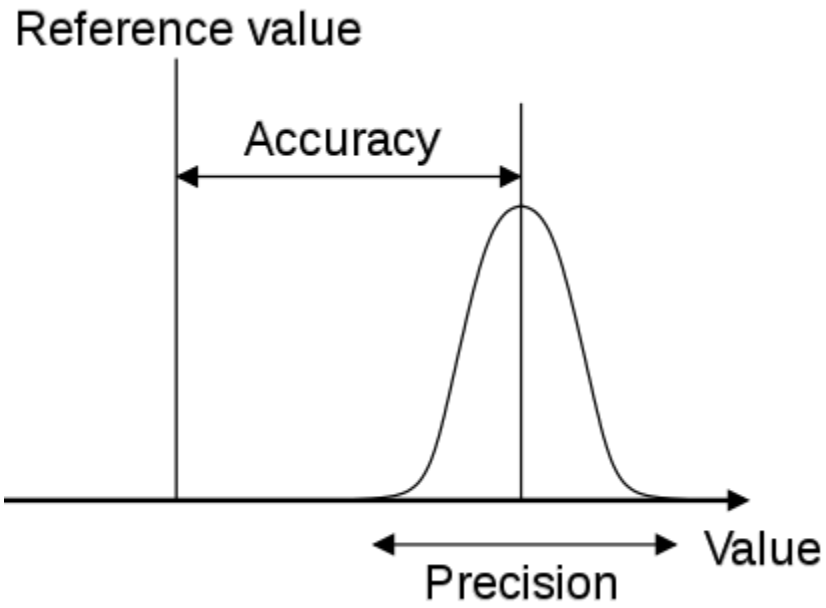
Precisão: de todas as classificações (positivas/negativas) do modelo, quantas estão corretas.

Acurácia (taxa de acerto): é a proporção de predições corretas:

Acurácia = (Tp + Tn) / (Tp + Fp + Tn + Fn);

Modelos Analíticos: Avaliação e Validação

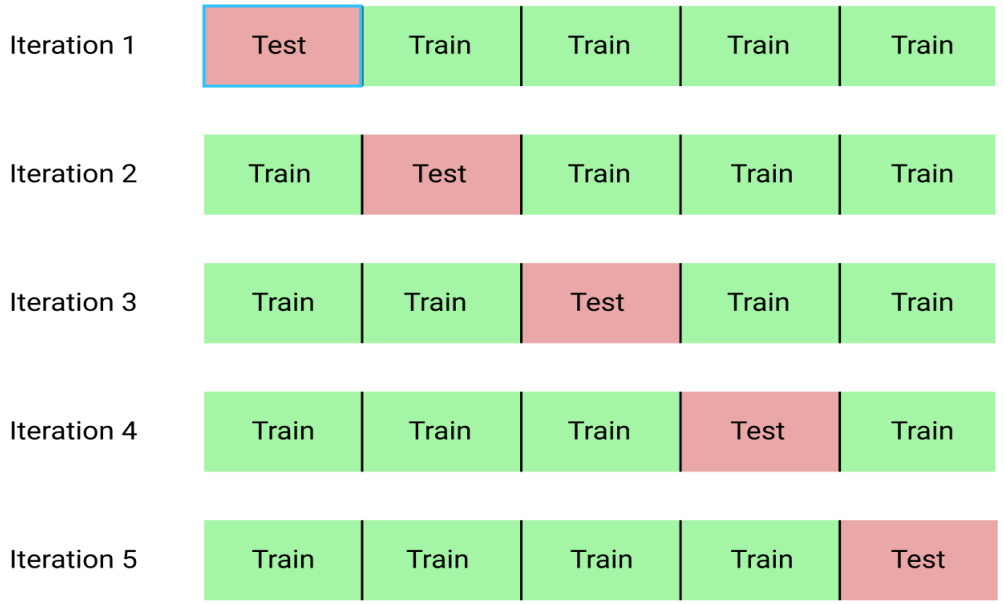




Modelos Analíticos: Avaliação e Validação

- **Cross-validation:** o conjunto de dados é dividido em N subconjuntos,com k dados cada conjunto. Sendo que as fases de teste são realizadas N vezes, com 1 conjunto como sendo de teste e os N-1restante de treinamento

Em cada interação, um desses subconjuntos é selecionado para ser o conjunto de teste, enquanto os demais compõem o conjunto de treinamento.



Modelos Analíticos: Avaliação e Validação

F-measure: essa medida , também conhecida por medida F, combina Precision e Recall, inserindo um parâmetro B para indicar a importância relativa.

Esse parâmetro pode assumir valores entre 0 e infinito, sendo que o valor é determinado pelo usuário.
Caso **B** assuma o valor **0**, então a medida **F** apresentará somente o valor da medida **Precision**.

Caso **B** assuma o valor **infinito**, então **F** apresentará o valor de **Recall**.

O valor de **B** geralmente é usado da seguinte forma: **0.5**, representando que o valor de Precision é mais relevante que o valor de Recall; **1.0**, representando a mesma relevância para ambos; ou **2.0**, representando que o valor de Recall é mais relevante que o valor de Precision.

Modelos Analíticos: Avaliação e Validação

A medida F pode ser calculada usando :

$$\text{F-measure}(M, ci) = ((1+B^2) \cdot \text{Precision}(M, ci) \cdot \text{Recall}(M, ci)) / (B^2.\text{Precision}(M, ci)+\text{Recall}(M, ci))$$

Onde:

M: conjunto classificado; ci: classe pré-definida

B: parâmetro indicador da relevância

Então: $\text{F-measure}(M, ci) = ((1 + B^2).tp)/(1 + B^2.tp + fp + B^2.fn)$

Modelos Analíticos: Avaliação e Validação

Quando temos uma classificação com k classes, cada elemento da matriz de confusão pode ser representado como M(Ci,Cj) que representa o número de elementos classificados como Cj e que pertencem a Ci.

Classes	Predita C1	Predita C2		Predita Ck
Verdade C1	M (C1,C1)	M (C1,C2)		M (C1,Ck)
Verdade C2				
Verdade Ck	M (Ck,C1)			M (Ck,Ck)

Modelos Analíticos: Avaliação e Validação

Neste caso, o erro da classificação pode ser
calculado como:
Erro = 1/n (Somai=1..n (Somaj=1..n M (Ci,Cj) * Cost(Ci,Cj)))

Onde Cost(Ci, Cj) = 0 quando i=j e Cost(Ci,Cj)> 0 quando i<>j
representando a penalidade quando o classificador comete
um erro.

Curva ROC e AUC

Curva ROC é uma representação gráfica da taxa dos verdadeiros positivos, versus taxa dos falsos positivos.

O melhor classificador é aquele que tem maior AUC (area under the curve), que varia entre 0 e 1 (1 = área de um quadrado unitário).

O valor 0.5 é a área de um classificador aleatório, isto é, nenhum classificador real deve ter um valor AUC < 0.5.

Curva ROC e AUC

O desempenho de cada classificador é representado como um único ponto (**FPR= taxa falsos positivos, TPR= taxa verdadeiros positivos**) no gráfico ROC.

		Pred	
		P	N
Real	P	400	100
	N	12000	87500

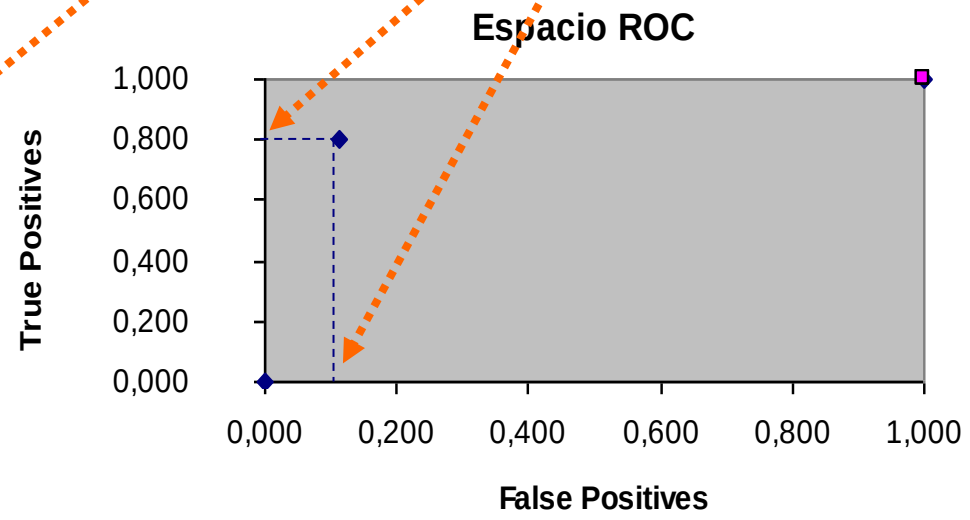
		Pred	
		P	N
Real	P	0,8	0,2
	N	0,121	0,879

TPR= 400 / 500 = 0.8

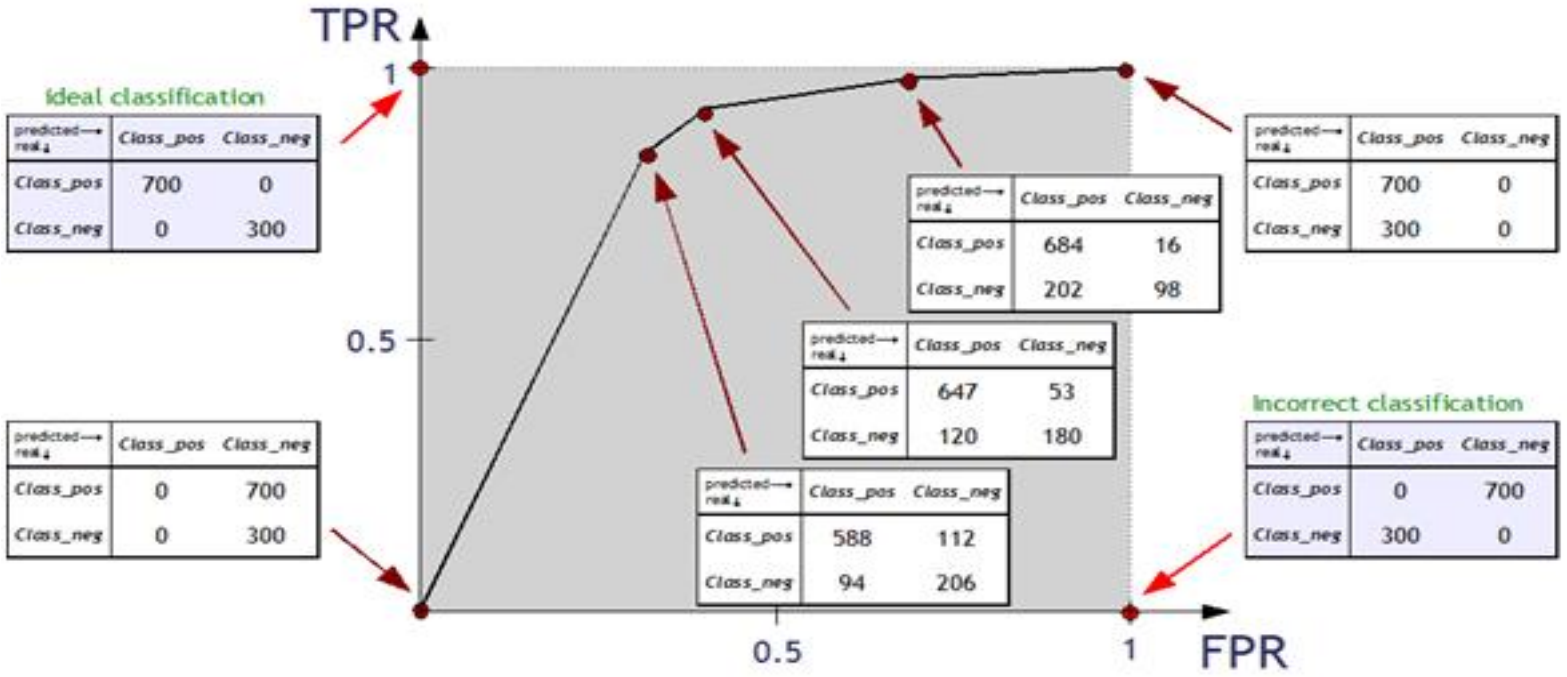
FPR= 12000 / 99500 = 0.121

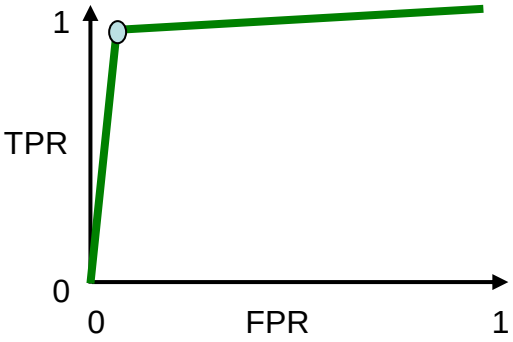
FNR= 100 / 500 = 0.2

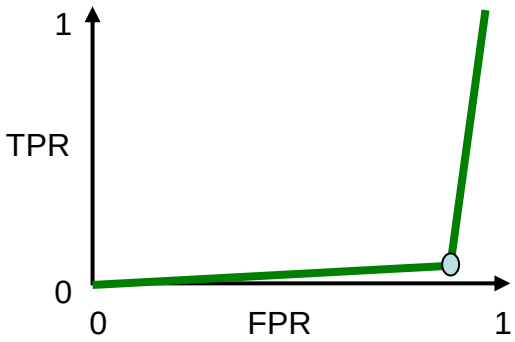
TNR= 87500 / 99500 = 0.879%



Curva ROC e AUC







Bom classificador

- Alto TPR.
- Baixo FPR

Mau classificador

- Baixo TPR
- Alto FPR

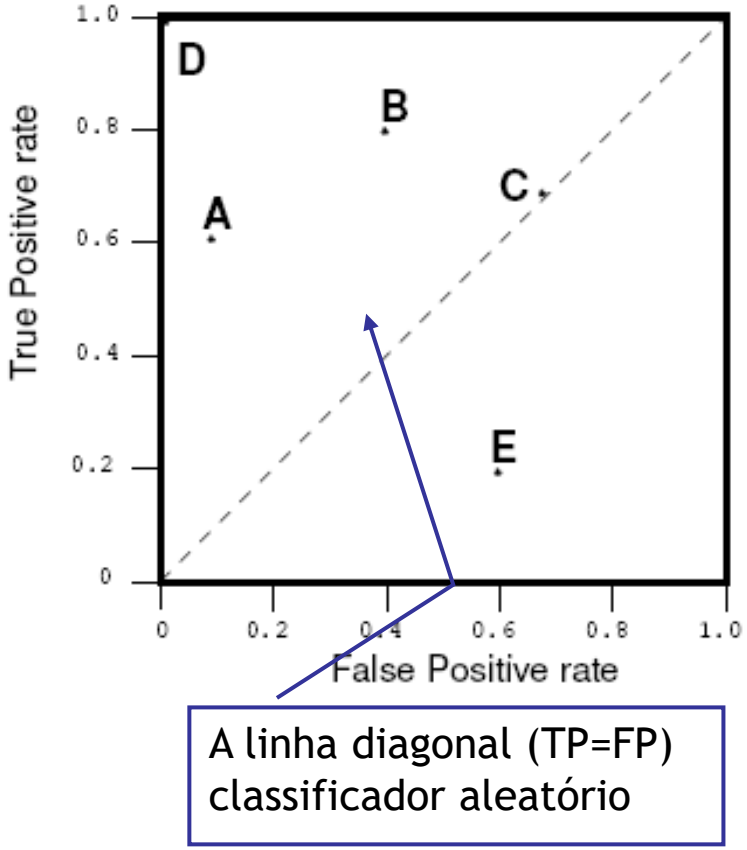
Curva ROC e AUC

D – o melhor (mais ao noroeste: taxa TP maior, taxa FP menor)

E - o pior (abaixo da diagonal – pior do que aleatório)

A é **mais conservador** que **B** (alto TP, baixo FP $\Rightarrow$ classifica um exemplo como positivo com mais precaução do que B, só quando existir uma forte evidência)

Classificadores **conservadores** (lado esquerdo) vs. **liberais** (lado direito)



Avaliação do algoritmo de aprendizado

Na matriz de confusão, temos 2 tipos de erros:

- Erro Tipo I - Falso Positivo: classificar um colaborador que deixa a empresa o modelo identifica como “não vai deixar a empresa”.

De uma perspectiva de negócios, este é o erro menos desejável, já que um colaborador com caraterísticas (muito provável) para abandonar o serviço **não é detectado**, afetando a empresa, já que estamos perdendo o cliente.

- Tipo II Erro - Falso Negativo: classificar um colaborador com caraterísticas de ficar na empresa, como um provável ”deixar a empresa ” (ser rotativo) .

Na perceptiva dos negócios, o Erro Tipo II é aceitável, pois não afeta a empresa;

Aprendizado de Máquina

Prof. Aran Morales, UNISUL – Anima Educação

Apostila – Analises Preditiva - Classificação