



Tecnologias e Sistemas de Informação

Sistemas Inteligentes

Aula 3 - Classificação e Árvores de Decisão

Prof. José Artur Quilici-Gonzalez
Email: jose.gonzalez@ufabc.edu.br



Roteiro

Introdução

Classificação

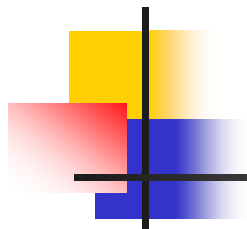
Árvores de Decisão

Indução de Árvores de Decisão

Treinamento, Aprendizado e Classificação

Matriz de Confusão

Referência Bibliográfica



INTRODUÇÃO

Exemplo I

- É de grande interesse conseguir classificar antecipadamente o tipo de problema apresentado por um paciente com base nos sintomas relatados e tomar medidas para combater a doença em seu estágio inicial
- Em muitos casos reais isso tem sido possível graças à análise minuciosa de Bases de Dados contendo anotações médicas de outros pacientes com soluções bem sucedidas previamente documentados

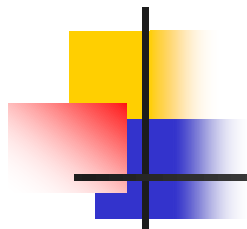




Exemplo II

Em instituições financeiras, para um gerente de banco nem sempre é algo simples fazer uma avaliação de risco sobre a concessão de empréstimos de alto valor

Com base em dados de transações anteriores e nas características específicas de cada cliente, é possível extrair automaticamente informações não óbvias que ajudam a classificar um correntista como bom ou mau pagador

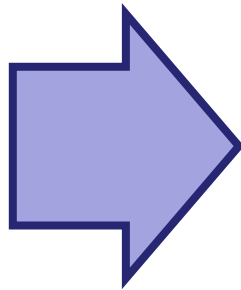


Classificação



Modelagem Preditiva

Classificação é uma forma de **Modelagem Preditiva**, isto é, com base nos **atributos de entrada** de um objeto é possível prever o **atributo de saída** desse objeto



Exemplos previamente rotulados em duas ou mais classes são utilizados num treinamento, cujo fim é criar uma estrutura de representação do conhecimento contido nessa Base de Dados



Aprendizado Supervisionado

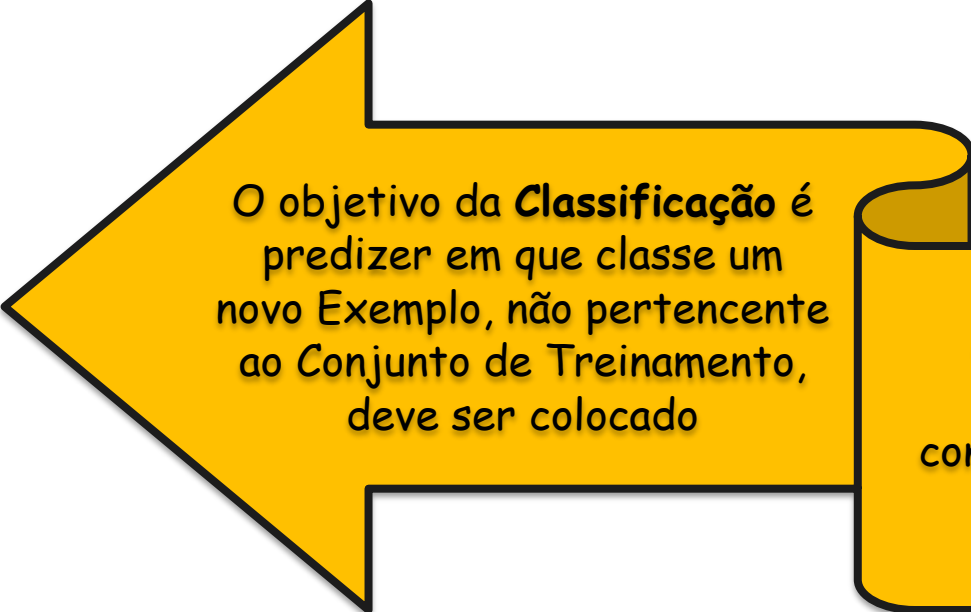
Quando se fala em Exemplos previamente rotulados geralmente se subentende que eles serão usados em **Aprendizado Supervisionado de Máquina**

Os rótulos dos Exemplos orientam o processo de Treinamento, ao final do qual se obtém um **Modelo** que sintetiza todo o conhecimento contido nas variáveis ou atributos

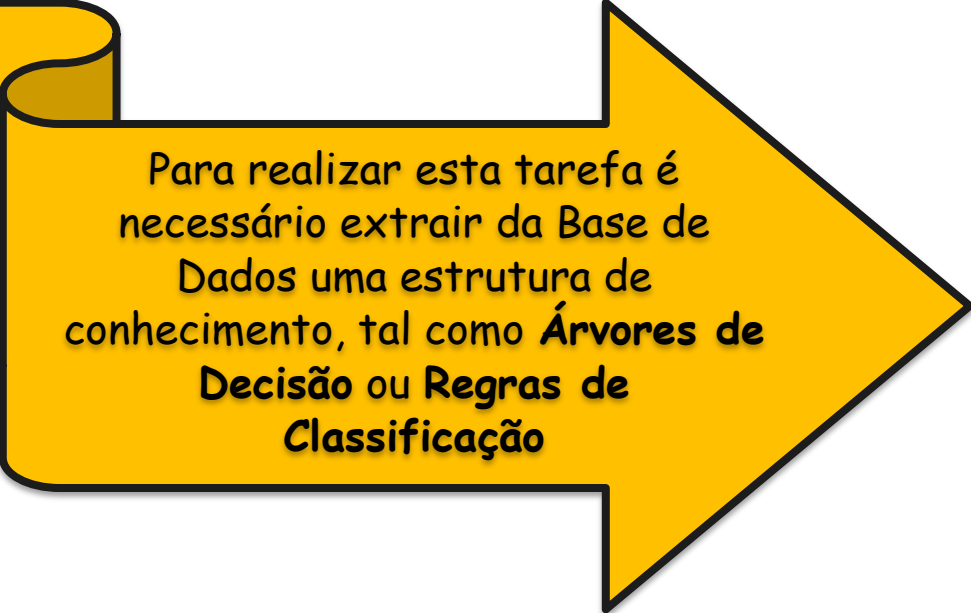
Este **Modelo** pode então ser usado para prever o valor da variável alvo de novos Exemplos desconhecidos



Objetivo da Classificação



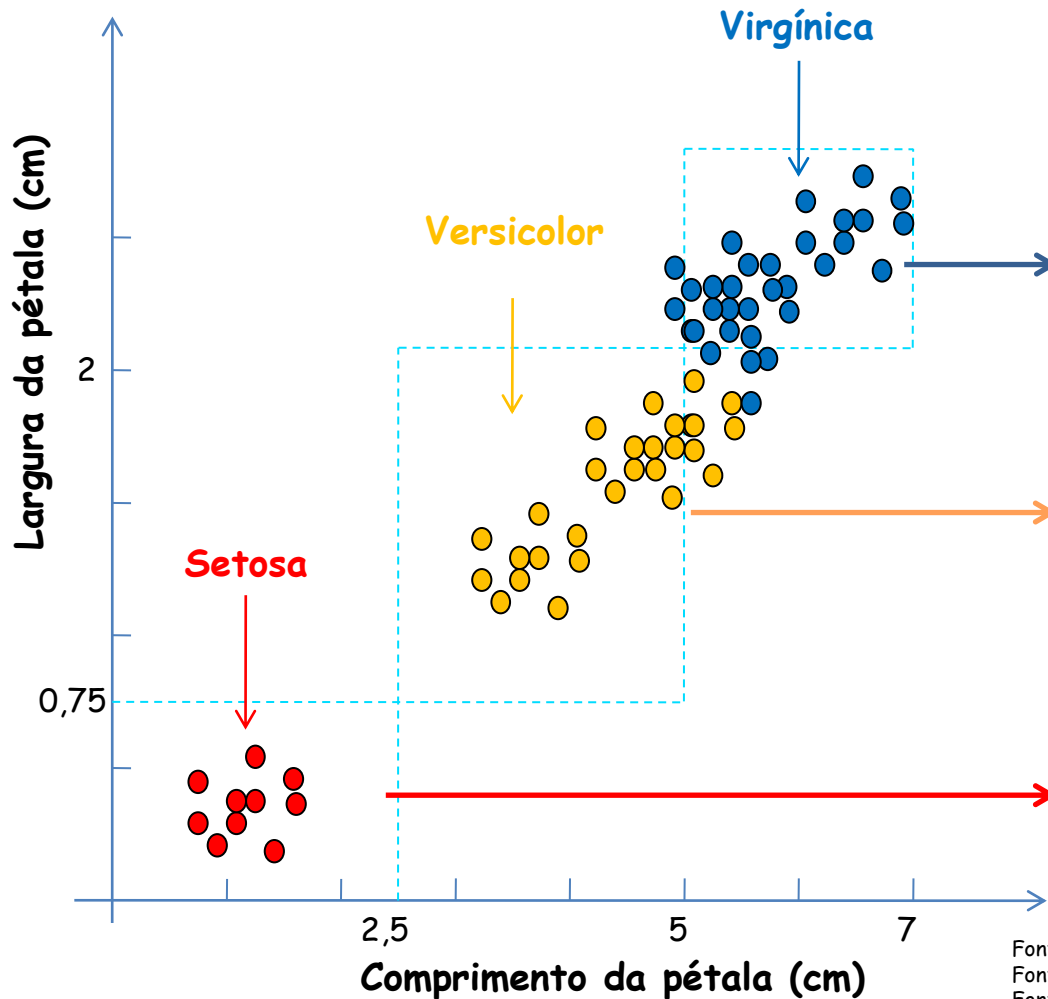
O objetivo da **Classificação** é prever em que classe um novo Exemplo, não pertencente ao Conjunto de Treinamento, deve ser colocado



Para realizar esta tarefa é necessário extrair da Base de Dados uma estrutura de conhecimento, tal como **Árvores de Decisão** ou **Regras de Classificação**

Comprimento X Largura (Íris)

Exemplo clássico introduzido por (Fisher, 1936)



De acordo com os atributos "Comprimento" e "Largura" da pétala, cada Exemplo dessa flor pode ser classificado em uma das três classes:

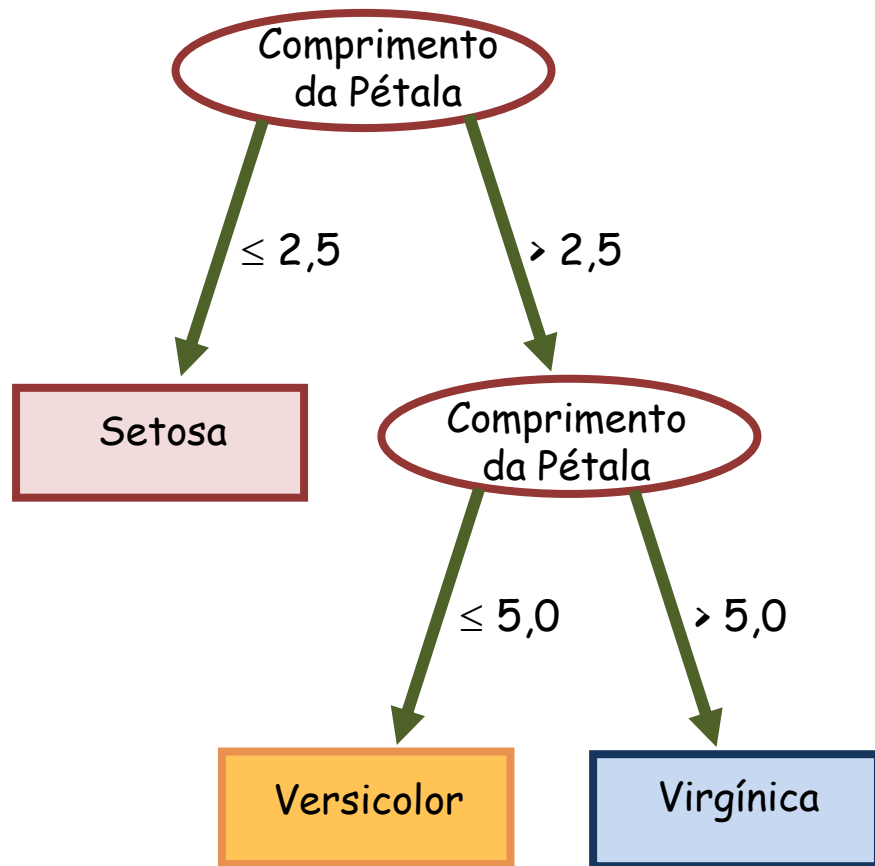
- Setosa
- Versicolor
- Virginica

Fonte: http://en.wikipedia.org/wiki/File:Iris_virginica.jpg (Acessado em 19.02.13).

Fonte: http://en.wikipedia.org/wiki/File:Iris_versicolor_3.jpg.

Fonte: http://en.wikipedia.org/wiki/File:Kosaciec_szczecinkowaty_Iris_setosa.jpg

Árvore de Decisão X Regras



If Comprimento da Pétala $\leq 2,5$ **then**
Classe = Setosa

If Comprimento da Pétala $> 2,5$ **and**
Comprimento da Pétala $\leq 5,0$ **then**
Classe = Versicolor

If Comprimento da Pétala $> 5,0$ **then**
Classe = Virgínica

Modelos simplificados de uma **Árvore de Decisão** e das correspondentes **Regras de Classificação** para o caso da flor Íris



Árvores de Decisão

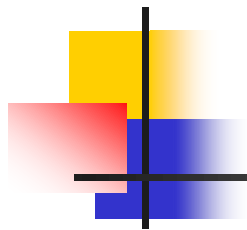
Árvores e Nós

Numa Árvore de Decisão cada **atributo** é representado por um **nó de decisão**, cuja função é testar o valor desse atributo

Uma **classe** é representada por um **nó folha**, que reúne todos os Exemplos que chegarem a ele depois de satisfazerem os testes dos nós de decisão intermediários

Numa Árvore, a classificação de um Exemplo implica percorrer toda a árvore a partir de um **nó raiz**, testando atributos nos **nós internos** até chegar a um **nó folha**, que lhe atribuirá uma **classe**

O objetivo de uma Árvore de Decisão é retornar uma classe para um Exemplo desconhecido

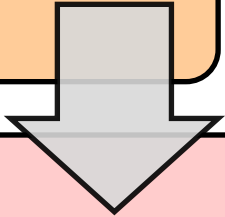


Indução de Árvores de Decisão

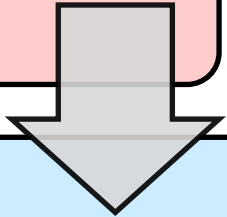


Como Gerar uma Árvore

Uma **Árvore de Decisão** pode ser construída de forma recursiva, dividindo sucessivamente o conjunto de atributos em subconjuntos



Primeiramente escolhemos um elemento do conjunto de atributos para ser o **nó raiz** e adicionamos uma **aresta** para cada um dos possíveis valores que este atributo pode assumir



A seguir, repetimos o processo em cada uma das arestas com os atributos restantes até que todos os Exemplos de um caminho pertençam à mesma classe

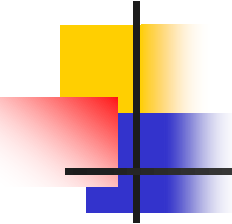


Tabela do Tempo

Será utilizado como exemplo a clássica Tabela do Tempo, introduzida por (Quinlan, 1986), para ilustrar como a ordem dos atributos escolhidos pode determinar o tamanho da Árvore de Decisão

Vamos considerar separadamente para **nó raiz** cada um dos quatro atributos possíveis e ver como o atributo de saída "Partida" se divide em "Sim" e "Não"

Dia	Temperatura	Umidade	Vento	Partida
Ensolarado	Elevada	Alta	Falso	Não
Ensolarado	Elevada	Alta	Verdadeiro	Não
Nublado	Elevada	Alta	Falso	Sim
Chuvoso	Amena	Alta	Falso	Sim
Chuvoso	Baixa	Normal	Falso	Sim
Chuvoso	Baixa	Normal	Verdadeiro	Não
Nublado	Baixa	Normal	Verdadeiro	Sim
Ensolarado	Amena	Alta	Falso	Não
Ensolarado	Baixa	Normal	Falso	Sim
Chuvoso	Amena	Normal	Falso	Sim
Ensolarado	Amena	Normal	Verdadeiro	Sim
Nublado	Amena	Alta	Verdadeiro	Sim
Nublado	Elevada	Normal	Falso	Sim
Chuvoso	Amena	Alta	Verdadeiro	Não

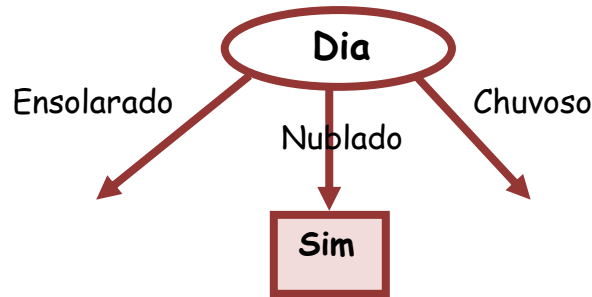
Pares de Atributos

Dentre os quatro atributos candidatos para nó raiz, o atributo "Dia" parece o mais promissor porque dentre as três arestas que teremos de colocar neste nó ("Ensolarado", "Nublado" e "Chuvoso"), a aresta para "Nublado" tem **todos** seus elementos pertencentes à mesma classe "Sim"

Dia	Partida	Temperatura	Partida	Umidade	Partida	Vento	Partida
Ensolarado	Sim	Elevada	Sim	Alta	Sim	Falso	Sim
Ensolarado	Sim	Elevada	Sim	Alta	Sim	Falso	Sim
Ensolarado	Não	Elevada	Não	Alta	Sim	Falso	Sim
Ensolarado	Não	Elevada	Não	Alta	Não	Falso	Sim
Ensolarado	Não	Amena	Sim	Alta	Não	Falso	Sim
Nublado	Sim	Amena	Sim	Alta	Não	Falso	Sim
Nublado	Sim	Amena	Sim	Alta	Não	Falso	Não
Nublado	Sim	Amena	Sim	Normal	Sim	Falso	Não
Nublado	Sim	Amena	Não	Normal	Sim	Verdadeiro	Sim
Chuvoso	Sim	Amena	Não	Normal	Sim	Verdadeiro	Sim
Chuvoso	Sim	Baixa	Sim	Normal	Sim	Verdadeiro	Sim
Chuvoso	Sim	Baixa	Sim	Normal	Sim	Verdadeiro	Não
Chuvoso	Não	Baixa	Sim	Normal	Sim	Verdadeiro	Não
Chuvoso	Não	Baixa	Não	Normal	Não	Verdadeiro	Não

Primeira Iteração da Árvore

Dia	Partida
Ensolarado	Sim
Ensolarado	Sim
Ensolarado	Não
Ensolarado	Não
Ensolarado	Não
Nublado	Sim
Nublado	Sim
Nublado	Sim
Nublado	Sim
Chuvoso	Sim
Chuvoso	Sim
Chuvoso	Sim
Chuvoso	Não
Chuvoso	Não



Como nas arestas "Ensolarado" e "Chuvoso" há elementos tanto da classe "Sim" como da "Não", outro atributo deve ser escolhido para cada aresta, e assim sucessivamente até que todos os elementos de um ramo pertençam a uma mesma classe

Como restam os atributos "Temperatura", "Umidade" e "Vento", vamos testar cada um deles em combinação com a aresta "Ensolarado"

Combinações com "Ensolarado"

Dia	Temp.	Partida
Ensolarado	Elevada	Não
Ensolarado	Elevada	Não
Ensolarado	Amena	Sim
Ensolarado	Amena	Não
Ensolarado	Baixa	Sim

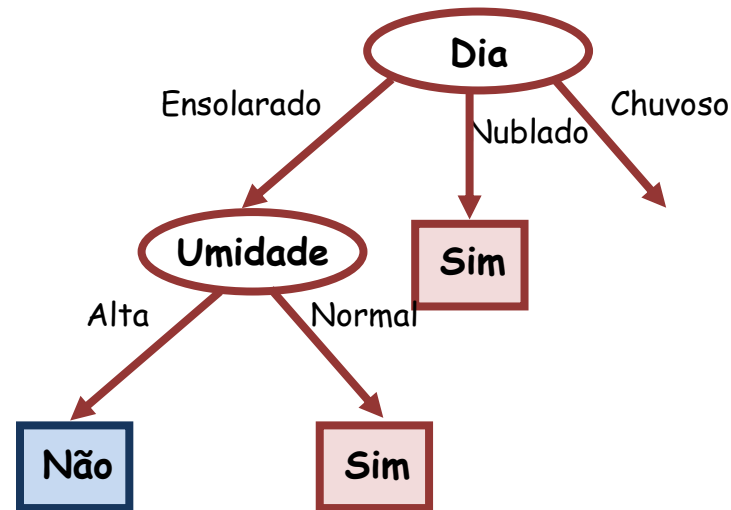
Dia	Umidade	Partida
Ensolarado	Alta	Não
Ensolarado	Alta	Não
Ensolarado	Alta	Não
Ensolarado	Normal	Sim
Ensolarado	Normal	Sim

Dia	Vento	Partida
Ensolarado	Falso	Sim
Ensolarado	Falso	Não
Ensolarado	Falso	Não
Ensolarado	Verdade	Sim
Ensolarado	Verdade	Não

"Umidade" parece ser a escolha mais promissora porque todos os elementos de "Umidade=Alta" correspondem à classe "Não" e todos os elementos com "Umidade=Normal" pertencem à classe "Sim"

Segunda Iteração da Árvore

Dia	Umidade	Partida
Ensolarado	Alta	Não
Ensolarado	Alta	Não
Ensolarado	Alta	Não
Ensolarado	Normal	Sim
Ensolarado	Normal	Sim



Portanto, temos mais dois nós folhas aqui, favorecendo a construção de uma árvore mais compacta

Para a terceira aresta, ou seja, "Dia=Chuvoso", restam duas alternativas: "Temperatura" e "Vento"

Atributo "Vento"

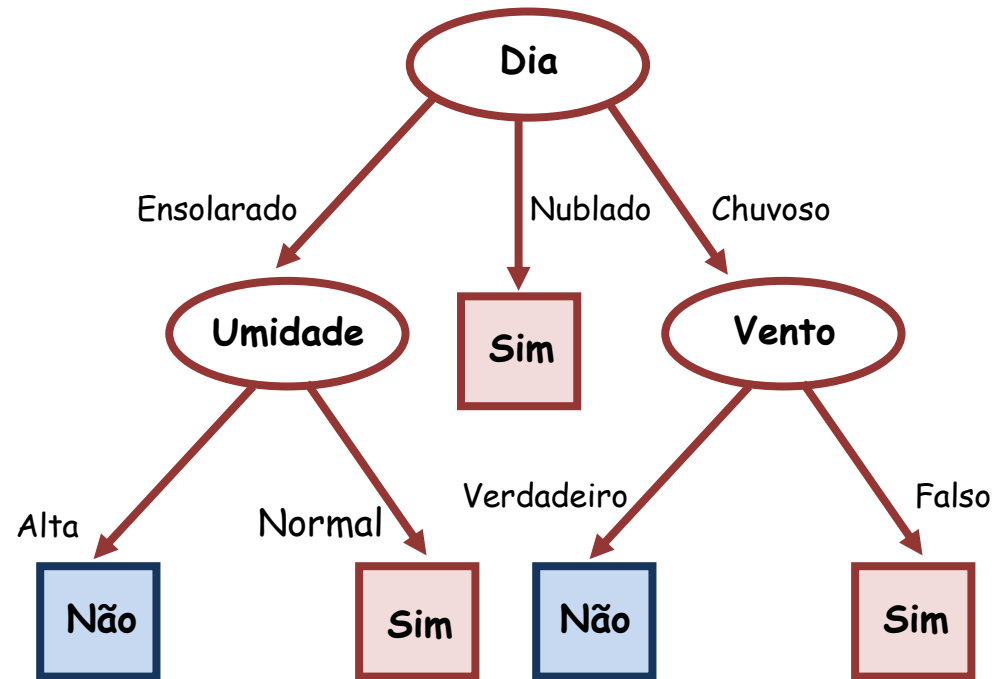
Dia	Temperatura	Partida
Chuvoso	Baixa	Não
Chuvoso	Baixa	Sim
Chuvoso	Amena	Sim
Chuvoso	Amena	Sim
Chuvoso	Amena	Não

Dia	Vento	Partida
Chuvoso	Falso	Sim
Chuvoso	Falso	Sim
Chuvoso	Falso	Sim
Chuvoso	Verdadeiro	Não
Chuvoso	Verdadeiro	Não

Comparando as duas tabelas, nota-se que o atributo "Vento" é o mais indicado para esta iteração porque todos os elementos de "Vento=Falso" estão classificados como "Sim" e todos os elementos de "Vento=Verdadeiro" estão classificados como "Não"

Terceira Iteração da Árvore

Dia	Vento	Partida
Chuvoso	Falso	Sim
Chuvoso	Falso	Sim
Chuvoso	Falso	Sim
Chuvoso	Verdadeiro	Não
Chuvoso	Verdadeiro	Não



Nesta iteração o algoritmo termina, pois todos os exemplos da tabela foram avaliados e classificados em suas respectivas classes



Informação num Atributo

Por trás do critério de seleção de atributos aqui apresentado de forma intuitiva, há uma sólida justificativa matemática introduzida por (Quinlan, 1986), baseada na **Teoria da Informação** de **Claude Shannon**, capaz de avaliar a quantidade de informação do melhor atributo dentre os candidatos para teste em um determinado nó

$$Info(Tabela) = - \sum_{i=1}^N p_i \log_2 p_i$$

sendo p_i a proporção de "Sim"s e "Não"s associados a um atributo (a quantidade de informação ou entropia é medida em bits, ou frações de bits!)



Cálculo da Informação

Por exemplo, na Tab. 3.1 temos apenas duas classes ("Sim" e "Não"), sendo que dos 14 Exemplos, 9 pertencem à classe "Sim" e 5 à classe "Não"

$$Info(Tab. 3.1) = (-9/14 \log_2 9/14) + (-5/14 \log_2 5/14) = 0,94bits$$

Uma forma alternativa de interpretar estes números, é pensar que estamos interessados em medir o grau de "impureza" de um conjunto de respostas

Se todas as respostas forem apenas "Sim" ou apenas "Não", então o grau de impureza do conjunto é 0

Há mais detalhes de como calcular a informação de um atributo no texto de Teoria desta unidade

Para efeito comparativo, vamos supor que algum critério arbitrário de escolha da ordem dos atributos tenha sido utilizado e que o nó raiz contenha o atributo "Umidade"

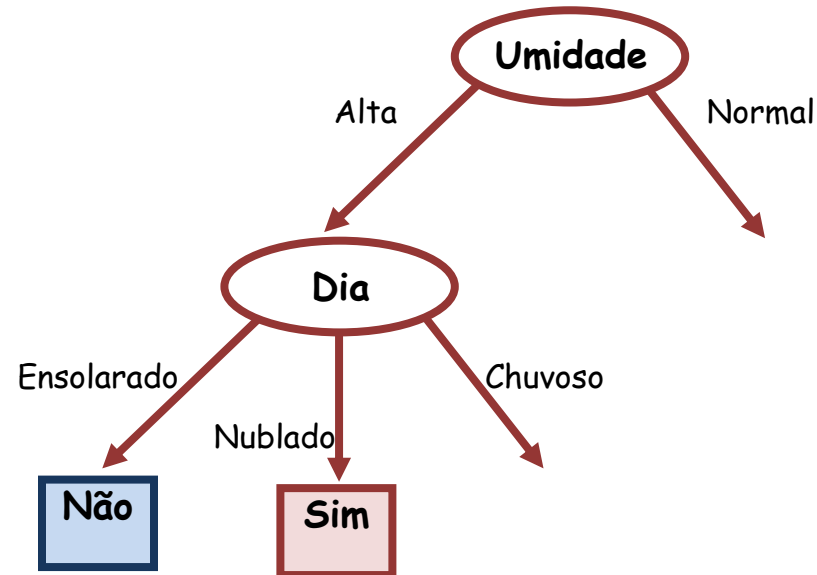


Este atributo possui exemplos misturados pertencentes a classes distintas, portanto é necessário um novo teste, i.e., escolher um novo atributo para teste

[illegible]

Segunda Iteração da Árvore Arbitrária

Dia	Umidade	Partida
Ensolarado	Alta	Não
Ensolarado	Alta	Não
Ensolarado	Alta	Não
Nublado	Alta	Sim
Nublado	Alta	Sim
Chuvoso	Alta	Sim
Chuvoso	Alta	Não

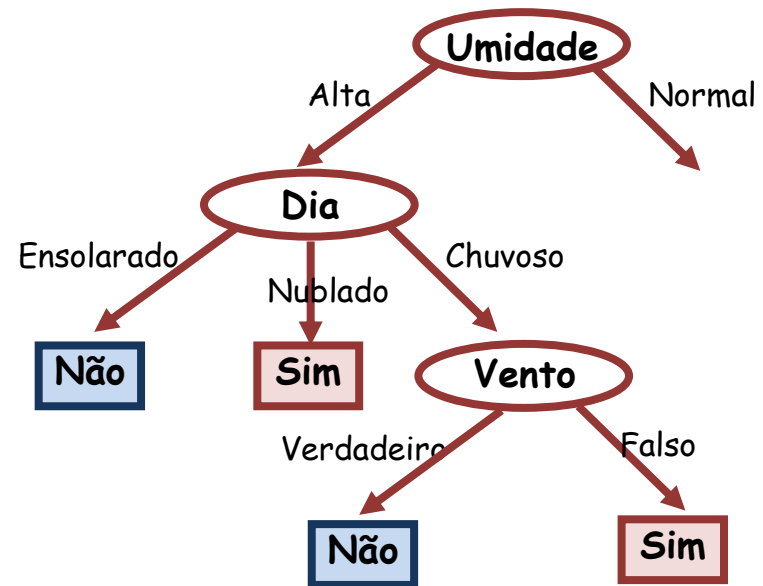


O atributo escolhido arbitrariamente agora foi "Dia" e as combinações possíveis com "Umidade" são mostradas

A aresta para "Dia=Chuvoso" exige um novo teste já que há duas respostas distintas possíveis

Terceira Iteração da Árvore Arbitrária

Dia	Umidade	Vento	Partida
Chuvoso	Alta	Falso	Sim
Chuvoso	Alta	Verdadeiro	Não



Esta região da Árvore de Decisão está encerrada, com dois novos nós folhas

Vamos agora considerar a aresta correspondente a "Umidade=Normal" e supor que o novo atributo será "Dia"

Quarta Iteração da Árvore Arbitrária

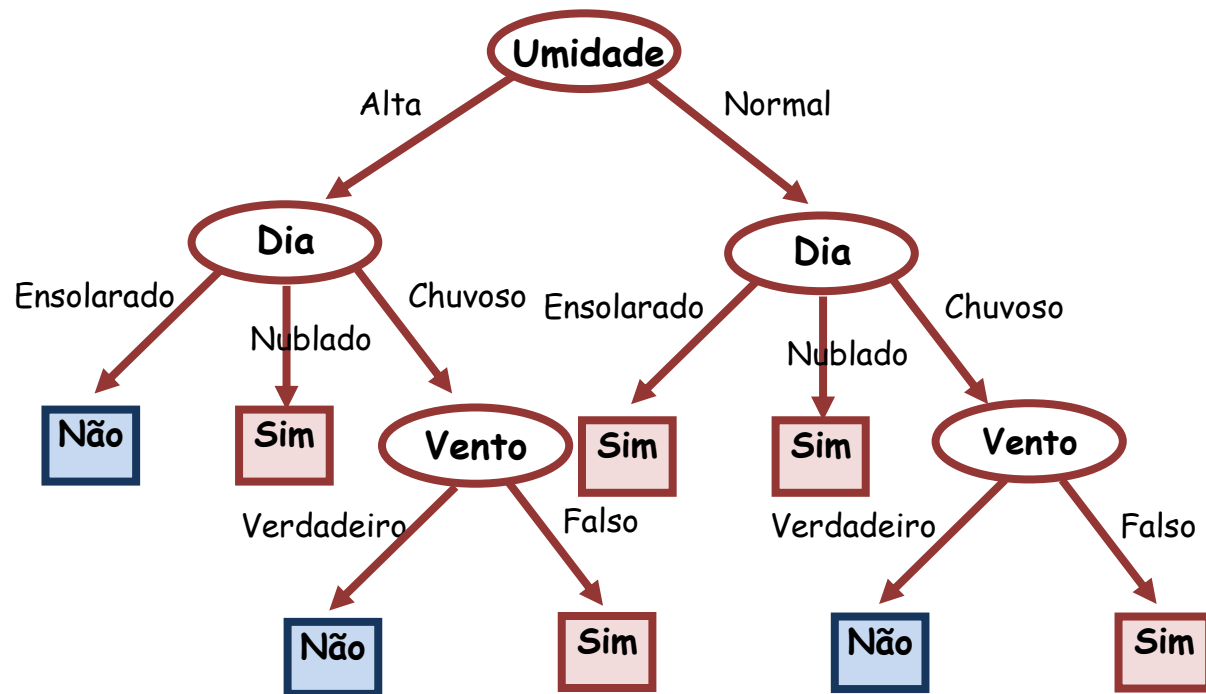
Dia	Umidade	Partida
Ensolarado	Normal	Sim
Ensolarado	Normal	Sim
Nublado	Normal	Sim
Nublado	Normal	Sim
Chuvoso	Normal	Sim
Chuvoso	Normal	Sim
Chuvoso	Normal	Não



Como a opção de "Dia=Chuvoso" exige um novo teste, vamos supor que o atributo escolhido tenha sido "Vento"

Árvore Não-Compacta

Dia	Umidade	Vento	Partida
Chuvoso	Normal	Falso	Sim
Chuvoso	Normal	Falso	Sim
Chuvoso	Normal	Verdadeiro	Não



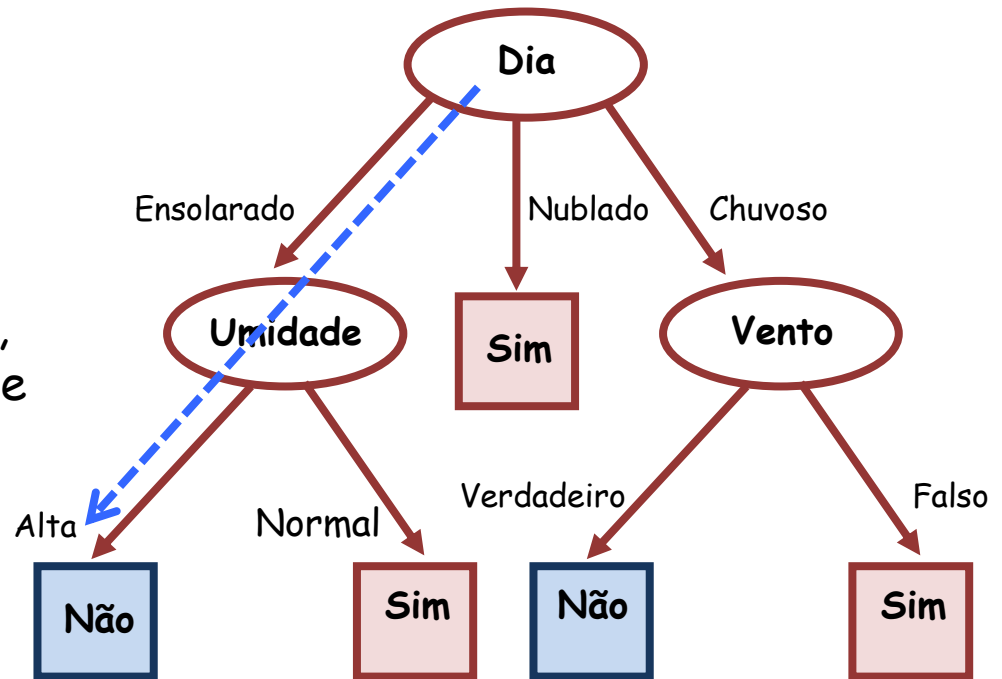
Com este teste, o algoritmo se encerra já que todos os exemplos foram devidamente considerados e se encaixaram num dos caminhos possíveis da Árvore de Decisão

As duas árvores classificam corretamente todos os exemplos da Tabela do Tempo, mas a mais compacta deve ser a preferida

Como Gerar Regras a partir de uma Árvore de Decisão

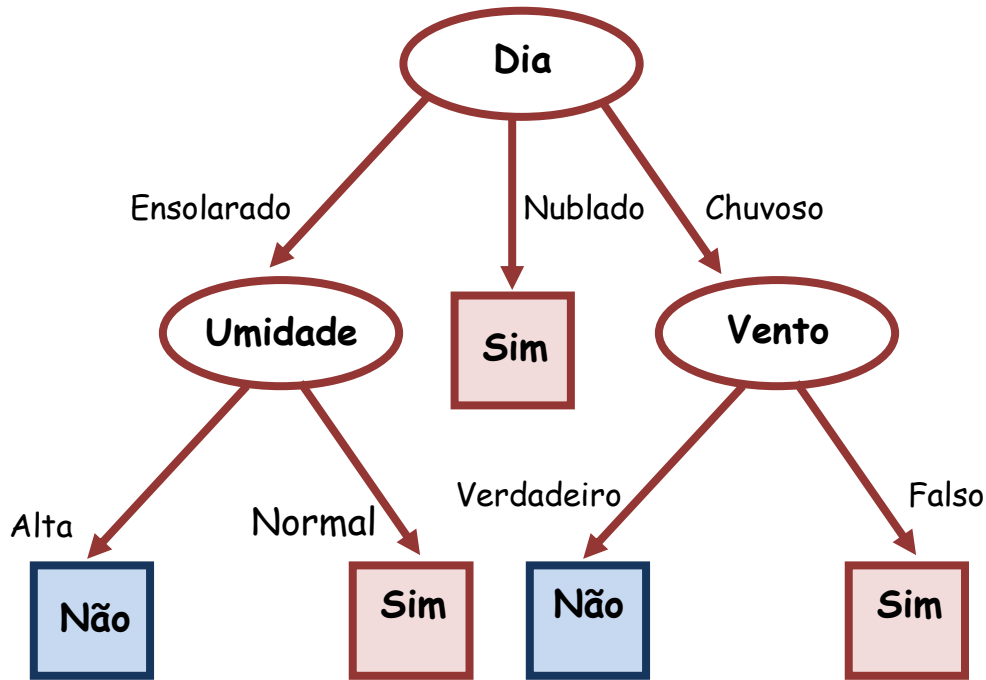
Partindo-se do nó raiz "Dia", seguindo pela aresta correspondente à condição "Ensolarado", passando pelo nó interno "Umidade" e, finalmente, tomando a aresta "Alta", chega-se ao nó folha "Não"

Dessa forma, podemos gerar a primeira regra relativa à classe "Não"



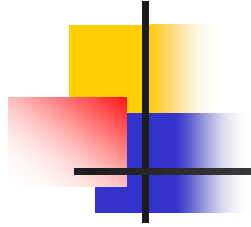
IF (Dia = Ensolarado) **AND** (Umidade = Alta) **THEN** (Partida = Não)

Exemplos de Regras



IF [(Dia = Ensolarado) **AND** (Umidade = Alta)] **OR** [(Dia = Chuvoso) **AND** (Vento = Verdadeiro)] **THEN** (Partida = **Não**)

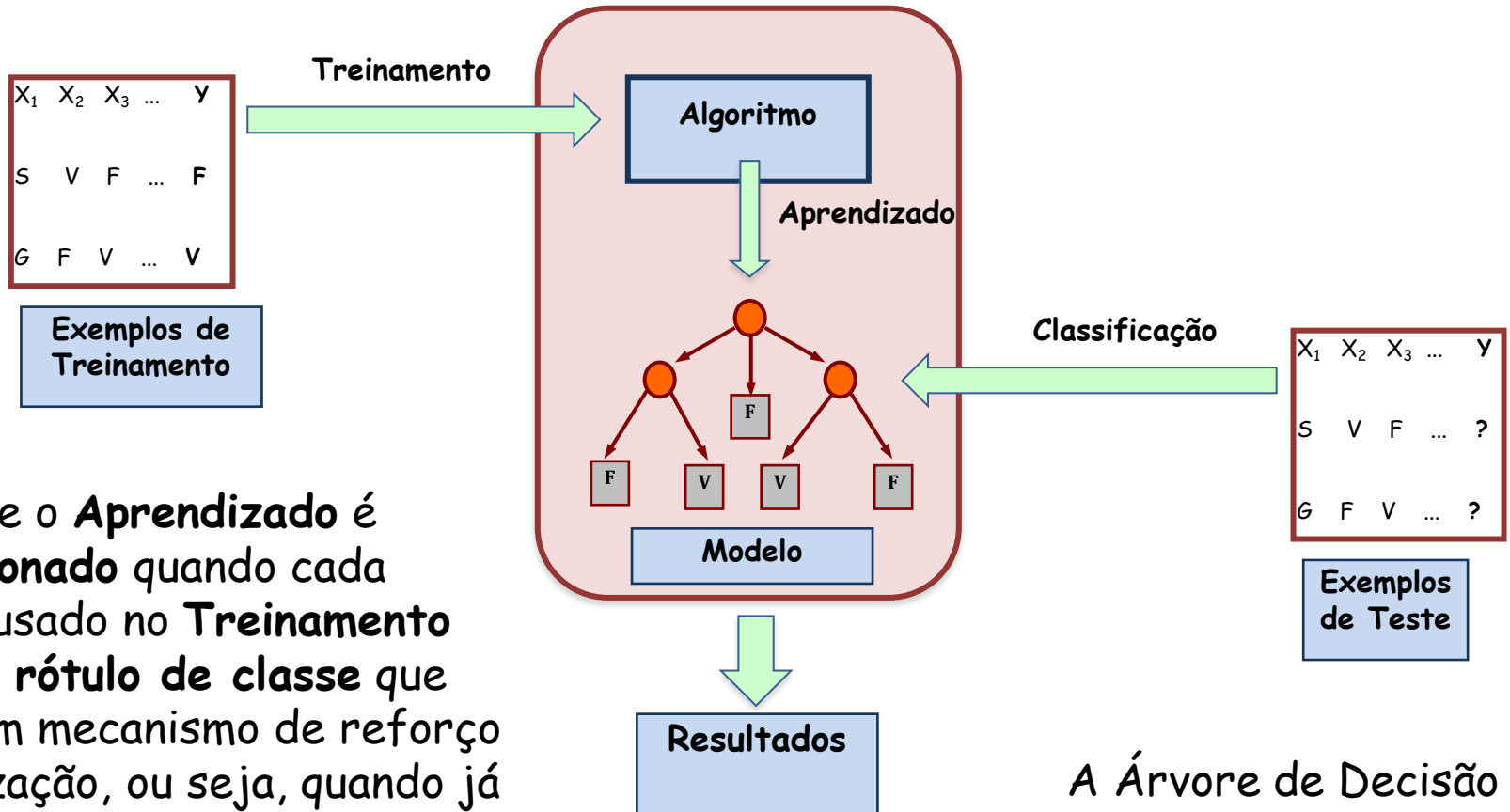
IF [(Dia = Ensolarado) **AND** (Umidade = Normal)] **OR** [(Dia = Nublado)] **OR** [(Dia = Chuvoso) **AND** (Vento = Falso)] **THEN** (Partida = **Sim**)



Treinamento, Aprendizado e Classificação

Treinamento, Aprendizado e Classificação

Sistema Inteligente Simples



Diz-se que o **Aprendizado** é **Supervisionado** quando cada Exemplo usado no **Treinamento** possui um **rótulo de classe** que orienta um mecanismo de reforço ou penalização, ou seja, quando já se sabe antecipadamente a qual classe determinado elemento pertence

A Árvore de Decisão resultante representa o **Modelo gerado** ou o **Conceito aprendido**

Classificação

Tendo descrito o processo de criação ou indução de uma Árvore de Decisão, vamos ver agora que resultados de classificação podemos obter com esta árvore

Esta Tabela apresenta alguns novos Exemplos que usaremos para teste

Usando qualquer das duas Árvore de Decisão geradas, o resultado para o primeiro Exemplo de Teste é "Sim",
para o segundo Exemplo, a resposta é "Não",
para o terceiro Exemplo, a resposta é "Sim"
e para o quarto Exemplo, a resposta é "Não"

Dia	Temperatura	Umidade	Vento	Partida
Ensolarado	Elevada	Alta	Falso	Não
Ensolarado	Elevada	Alta	Verdadeiro	Não
Nublado	Elevada	Alta	Falso	Sim
Chuvoso	Amena	Alta	Falso	Sim
Chuvoso	Baixa	Normal	Falso	Sim
Chuvoso	Baixa	Normal	Verdadeiro	Não
Nublado	Baixa	Normal	Verdadeiro	Sim
Ensolarado	Amena	Alta	Falso	Não
Ensolarado	Baixa	Normal	Falso	Sim
Chuvoso	Amena	Normal	Falso	Sim
Ensolarado	Amena	Normal	Verdadeiro	Sim
Nublado	Amena	Alta	Verdadeiro	Sim
Nublado	Elevada	Normal	Falso	Sim
Chuvoso	Amena	Alta	Verdadeiro	Não
Ensolarado	Amena	Normal	Falso	????
Ensolarado	Baixa	Alta	Verdadeiro	????
Nublado	Baixa	Alta	Verdadeiro	????
Chuvoso	Elevada	Normal	Falso	????



Overfitting e Poda

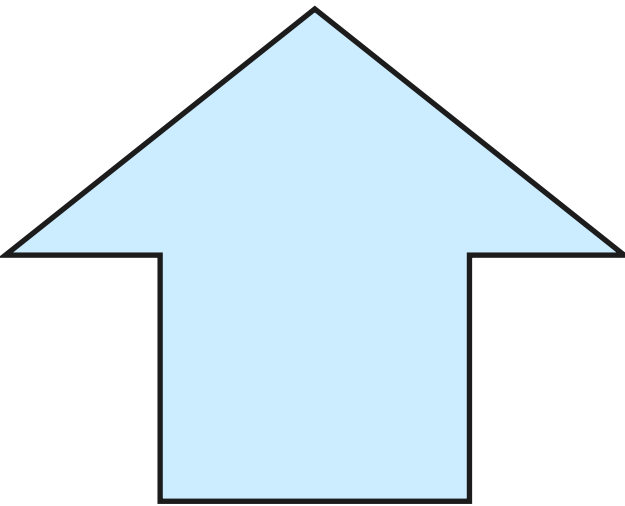
Para alguns **Exemplos de Teste** os resultados produzidos por Árvores compactas podem não ser os mesmos de Árvores não-compactas, muito embora os resultados para todos os **Exemplos de Treinamento** tenham sido os mesmos

O que pode ocorrer com Árvores não-compactas é que algumas das arestas refletem um **Superajuste (Overfitting)** aos **Exemplos de Treinamento**

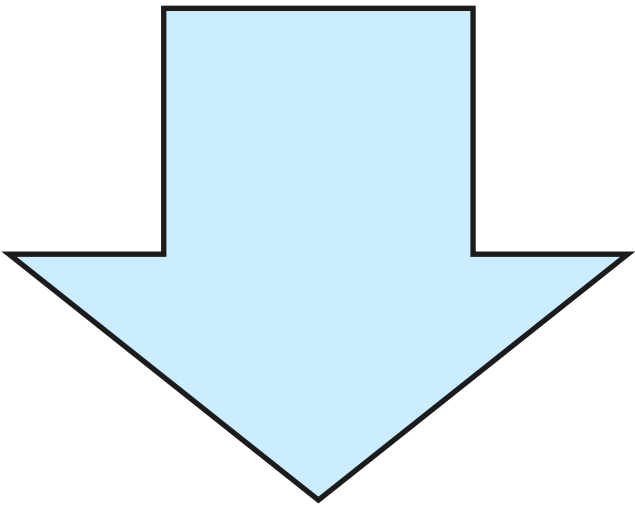
Se neste conjunto de **Exemplos de Treinamento** houver ruído ou *outliers*, a estrutura resultante da Árvore de Decisão pode não refletir as relações essenciais entre os atributos da Base de Dados



Poda

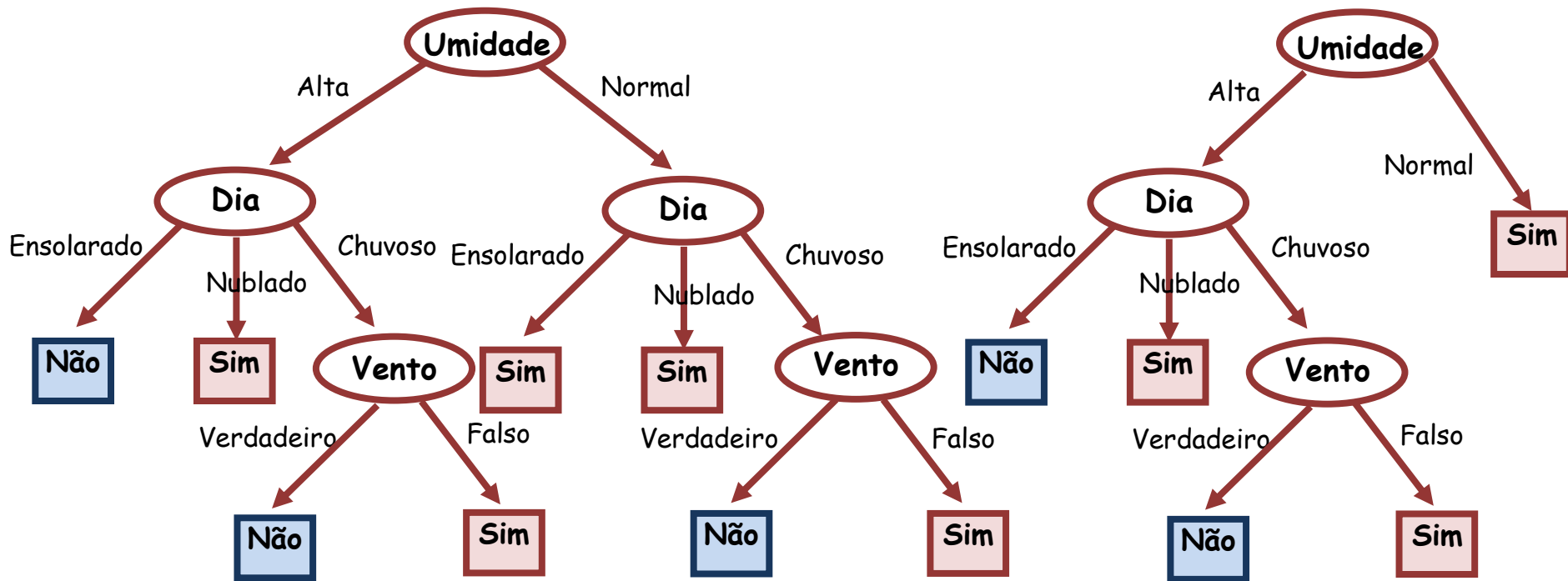


Para evitar o *overfitting*, muitos algoritmos se valem de uma técnica conhecida como "Poda", que consiste em eliminar algumas arestas da Árvore de Decisão com base em medidas estatísticas dos Exemplos



A Poda pode ocorrer sobre uma Árvore de Decisão concluída, com a eliminação de algumas arestas consideradas não necessárias, ou durante a construção da Árvore de Decisão, com a introdução precoce de um nó folha em arestas com baixa importância estatística, por exemplo

Exemplo de Poda



A Árvore da esquerda poderia ter algumas de suas arestas removidas sem comprometer seriamente a taxa de erros na classificação dos Exemplos apresentados

Dos sete exemplos da Tabela do Tempo que apresentam "Umidade=Normal", seis deles têm rótulo de classe "Sim"



MATRIZ DE CONFUSÃO

Predição e Diagnóstico

Ao se gerar uma Árvore de Decisão o que se espera é que ela classifique corretamente Exemplos desconhecidos, mas na prática às vezes verifica-se a ocorrência de classificações equivocadas

Isso também ocorre com o diagnóstico de profissionais

Quando um especialista deseja detectar a presença ou não de uma doença, ele solicita exames laboratoriais para auxiliá-lo a formular um diagnóstico positivo ou negativo sobre a provável doença





Combinações de Respostas

Se as respostas possíveis para um diagnóstico forem "Positivo" e "Negativo", quatro combinações de **resultados previstos** e **resultados reais** podem ocorrer

Se o paciente for portador da doença e o médico acertar no diagnóstico, dizemos que este caso é um **Verdadeiro Positivo** ou VP

Se o paciente não for portador da doença e o médico acertar no diagnóstico, dizemos que este caso é um **Verdadeiro Negativo** ou VN

Se o paciente for portador da doença, e o médico errar no diagnóstico afirmando que ele está são, dizemos que este caso é um **Falso Negativo** ou FN

Se o paciente não for portador da doença, e o médico errar no diagnóstico dizendo que ele está doente, dizemos que este caso é um **Falso Positivo** ou FP



Matriz de Confusão

As quatro combinações possíveis de resultados costumam ser representadas por uma matriz que recebe o nome de "**Matriz de Confusão**"

	Positivo Previsto	Negativo Previsto
Positivo Real	Verdadeiro Positivo (VP)	Falso Negativo (FN)
Negativo Real	Falso Positivo (FP)	Verdadeiro Negativo (VN)

Os valores contidos numa Matriz de Confusão podem ser utilizados para avaliar o desempenho de uma Árvore de Decisão

O que se espera nos resultados é que os casos positivos sejam classificados como positivos e os negativos como negativos, ou seja, o desejável é que as taxas de sucesso para Verdadeiro Positivo e Verdadeiro Negativo sejam altas, e que as taxas de Falso Positivo e Falso Negativo sejam baixas



Precisão ou Acurácia

Fazendo uma relação entre os Exemplos corretamente classificados, i.e., Verdadeiro Positivo (VP) + Verdadeiro Negativo (VN), com o número total de classificações (VP+VN+FP+FN), podemos definir uma métrica de desempenho para a taxa de acertos ou sucesso, conhecida como **Precisão** ou **Acurácia** de uma Árvore de Decisão

$$Acurácia = \frac{VP + VN}{VP + VN + FP + FN} \times 100\%$$



Considerações Finais

A geração de **Árvores de Decisão** normalmente é comparativamente mais rápida que outros métodos de classificação

Árvores de Decisão pequenas são fáceis de entender e Árvores grandes podem ser convertidas em Regras de Classificação

Geralmente a taxa de acerto de classificação de Exemplos de teste, ou seja, a Acurácia das Árvores de Decisão, é compatível com outros métodos equivalentes, ou um pouco abaixo de métodos mais complexos

Porém, em Aprendizado de Máquina raramente se encontra um método com desempenho superior que seus pares para qualquer conjunto de dados



Referência Bibliográfica



Referência Bibliográfica I

- FISHER, R. A. **The Use of Multiple Measurements in Taxonomic Problems.** Annals of Eugenics, Vol. 7, Issue 2, pages 179-188, 1936. In <http://onlinelibrary.wiley.com/doi/10.1111/j.1469-1809.1936.tb02137.x/abstract>. Acessado em 20.02.2013.
- HAN, J. & KAMBER, M. **Data Mining: Concepts and Techniques.** San Francisco: Morgan Kaufmann Publishers, 2008.
- PINHEIRO, C. A. R. **Inteligência Analítica: Mineração de Dados e Descoberta de Conhecimento.** Rio de Janeiro: Editora Ciência Moderna Ltda., 2008.
- QUINLAN, J. R. **Induction of Decision Trees.** Machine Learning, Vol. 1, No. 1, pp. 81-106. Boston: Kluwer Academic Publishers, 1986.
- REZENDE, S. O. (Organizadora). **Sistemas Inteligentes: Fundamentos e Aplicações.** Barueri: Editora Manole Ltda., 2005.



Referência Bibliográfica II

- ROCHA, M.; CORTEZ, P. & NEVES, J. M. **Análise Inteligente de Dados: Algoritmos e Implementação em Java**. Lisboa: Editora de Informática, 2008.
- TAN, P.N.; STEINBACH, M. & KUMAR, V. **Introdução ao Data Mining Mineração de Dados**. Rio de Janeiro: Editora Ciência Moderna Ltda., 2009.
- WITTEN, I. H. & FRANK, E. **Data Mining: Practical Machine Learning Tools and Techniques**. Second Edition. Amsterdam: Morgan Kaufmann Publishers, 2005.