

Mineração de Dados

04 - Introdução à Classificação e Árvores de Decisão

Marcos Roberto Ribeiro



Instituto Federal Minas Gerais - Campus Bambuí

2018

Introdução

- A classificação consiste em organizar objetos em uma dentre diversas categorias (ou classes) pré-definidas
- Exemplos de aplicações:
 - Detecção de mensagens de spam
 - Classificação de galáxias baseadas em imagens
- Os dados usadas para classificação são conjuntos de registros
- Cada registro tem um atributo discreto que identifica sua classe
- O objeto da classificação é construir um modelo com base em objetos classificados previamente
- O modelo construído pode então ser usado para descrever o conjunto de objetos ou para prever a classe de novos objetos
- No caso da previsão, é importante que o modelo seja genérico o suficiente para ter uma boa taxa de acerto

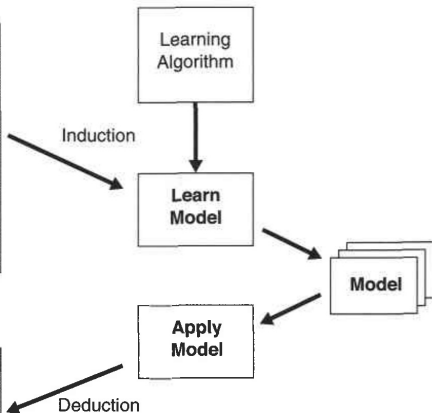
Abordagem Geral dos Classificadores

Training Set

Tid	Attrib1	Attrib2	Attrib3	Class
1	Yes	Large	125K	No
2	No	Medium	100K	No
3	No	Small	70K	No
4	Yes	Medium	120K	No
5	No	Large	95K	Yes
6	No	Medium	60K	No
7	Yes	Large	220K	No
8	No	Small	85K	Yes
9	No	Medium	75K	No
10	No	Small	90K	Yes

Test Set

Tid	Attrib1	Attrib2	Attrib3	Class
11	No	Small	55K	?
12	Yes	Medium	80K	?
13	Yes	Large	110K	?
14	No	Small	95K	?
15	No	Large	67K	?



Avaliação de Desempenho do Classificador

- A avaliação de desempenho do classificador é feita considerando a quantidade de registros classificados corretamente e incorretamente

Precisão

$$\text{Precisão} = \frac{\text{Número de previsões corretas}}{\text{Número total de previsões}}$$

Taxa de erros

$$\text{Taxa de erros} = \frac{\text{Número de previsões erradas}}{\text{Número total de previsões}}$$

Árvores de Decisão

- A ideia básica de uma árvore de decisão é pegar um problema complexo e decompô-lo em sub-problemas menores, de modo que os novos problemas tenham uma complexidade menor em relação ao anterior
- Essa estratégia é então aplicada recursivamente a cada sub-problema
- O exemplo nos mostra como funciona uma árvore de decisão

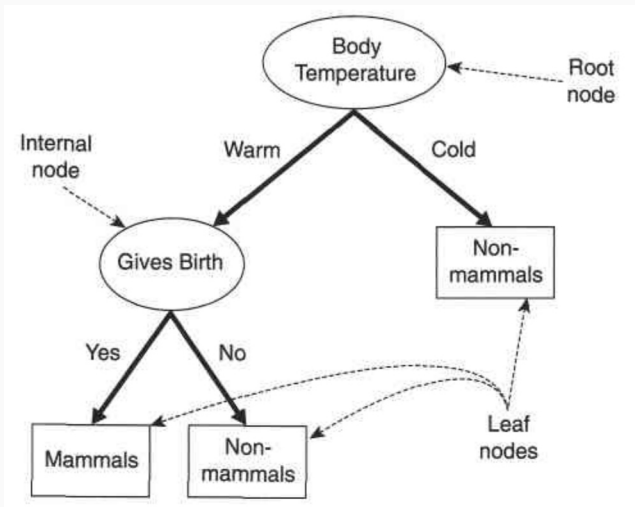
Funcionamento de Uma Árvore de Decisão I

Conjunto de dados de vertebrados

Name	Blode	Skin	Gives Birth	Aquatic	Aerral	Has Legs	Hibernates	Class
human	warm	hair	yes	no	no	yes	no	mammal
python	cold	scales	no	no	no	no	yes	reptile
salmon	cold	scales	no	yes	no	no	no	fish
whale	warm	hair	yes	yes	no	no	no	mammal
frog	cold	none	no	semi	no	yes	yes	amphibian
komodo dragon	cold	scales	no	no	no	yes	no	reptile
bat	warm	hair	yes	no	yes	yes	yes	mammal
pigeon	warm	feathers	no	no	yes	yes	no	bird
cat	warm	fur	yes	no	no	yes	no	mammal
leopard shark	cold	scales	yes	yes	no	no	no	fish
turtle	cold	scales	no	semt	no	yes	no	reptile
penguin	warm	feathers	no	semi	no	yes	no	bird
porcopine	warm	quills	yes	no	no	yes	yes	mammal
eel	cold	scales	no	yes	no	no	no	fish
salamander	cold	none	no	semi	no	yes	yes	amphibian

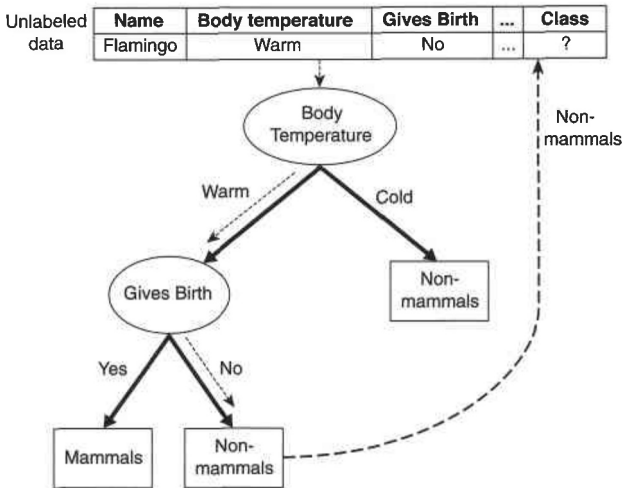
- Como identificar mamíferos e não mamíferos usando este conjunto de dados?

Funcionamento de Uma Árvore de Decisão II



Funcionamento de Uma Árvore de Decisão III

Como classificar um novo registro?



Construindo Árvores de Decisão

- Se tivermos um conjunto de dados com algumas dezenas de registros, a construção de uma árvore de decisão não é tão trivial
- Na verdade, pode existir um número muito grande de possíveis árvores de decisão
- Apesar disto, muitos algoritmos eficientes foram desenvolvidos para obter árvores de decisão precisas sobre grandes conjuntos de dados
- A grande maioria destes algoritmos segue a ideia geral do algoritmo de Hunt

Algoritmo de Hunt

- A árvore de decisão cresce recursivamente particionando os o conjunto inicial de registros em subconjuntos mais puros
- Se D_t o conjunto de registros associados ao nó t e $C = \{c_1, \dots, c_n\}$ as classes associadas destes registros
 1. Se todos os registros em D_t são da mesma classe c_i , então t é um nó folha rotulado com c_i
 2. Caso contrário:
 - Use uma **condição de teste** para particionar os registros em subconjuntos menores
 - Crie um nó filho para cada resultado da condição de teste
 - Distribua os registros nos filhos usando a condição de teste
 - Aplique o algoritmo recursivamente no nós filhos

Exemplo

ID	Casa Própria	Estado Civil	Renda	Inadimplente
1	sim	solteiro	125.000	não
2	não	casado	100.000	não
3	não	solteiro	70.000	não
4	sim	casado	120.000	não
5	não	divorciado	95.000	sim

ID	Casa Própria	Estado Civil	Renda	Inadimplente
6	não	casado	60.000	não
7	sim	divorciado	220.000	não
8	não	solteiro	85.000	sim
9	não	casado	75.000	não
10	não	solteiro	90.000	sim

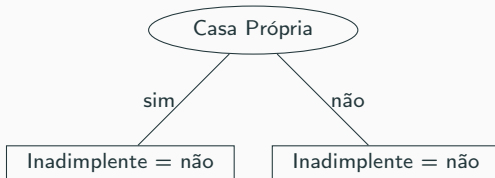
Inadimplente = não

- A maioria não é bom pagador
- Porém, existem alguns inadimplentes
- Então dividimos usando o atributo **Casa Própria**

Exemplo

ID	Casa Própria	Estado Civil	Renda	Inadimplente
1	sim	solteiro	125.000	não
2	não	casado	100.000	não
3	não	solteiro	70.000	não
4	sim	casado	120.000	não
5	não	divorciado	95.000	sim

ID	Casa Própria	Estado Civil	Renda	Inadimplente
6	não	casado	60.000	não
7	sim	divorciado	220.000	não
8	não	solteiro	85.000	sim
9	não	casado	75.000	não
10	não	solteiro	90.000	sim

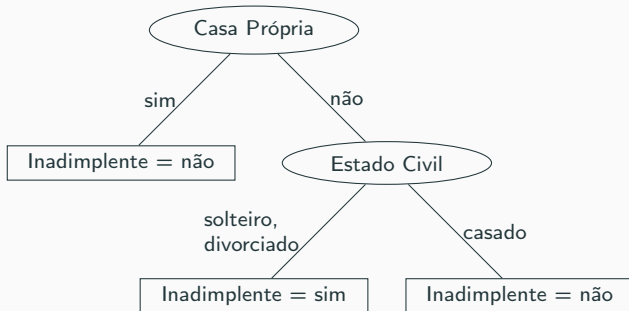


Continuamos dividindo porque os dados ainda não estão puros

Exemplo

ID	Casa Própria	Estado Civil	Renda	Inadimplente
1	sim	solteiro	125.000	não
2	não	casado	100.000	não
3	não	solteiro	70.000	não
4	sim	casado	120.000	não
5	não	divorciado	95.000	sim

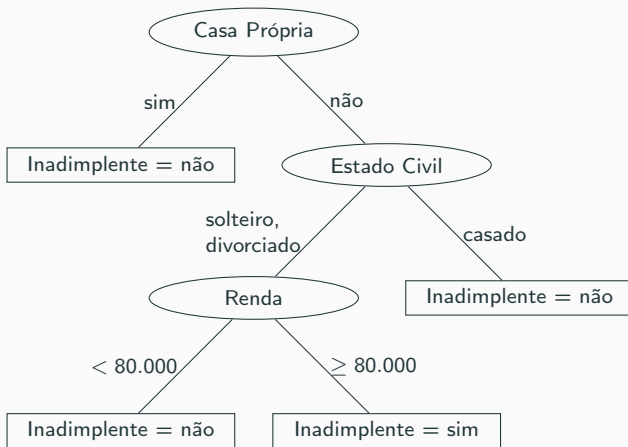
ID	Casa Própria	Estado Civil	Renda	Inadimplente
6	não	casado	60.000	não
7	sim	divorciado	220.000	não
8	não	solteiro	85.000	sim
9	não	casado	75.000	não
10	não	solteiro	90.000	sim



Exemplo

ID	Casa Própria	Estado Civil	Renda	Inadimplente
1	sim	solteiro	125.000	não
2	não	casado	100.000	não
3	não	solteiro	70.000	não
4	sim	casado	120.000	não
5	não	divorciado	95.000	sim

ID	Casa Própria	Estado Civil	Renda	Inadimplente
6	não	casado	60.000	não
7	sim	divorciado	220.000	não
8	não	solteiro	85.000	sim
9	não	casado	75.000	não
10	não	solteiro	90.000	sim



Algoritmo ID3

1. Crie o nó raiz R associado a D
2. Chama a rotina $GeraArvore(R, D, A)$, onde D é o conjunto de dados e A os atributos

Rotina $GeraArvore(N, D, A)$

- 1) Se todas as tuplas de D são da mesma classe c , então transforma N em folha com rótulo c
- 2) Senão, se $A = \{\}$, então transforma N numa folha com rótulo igual a classe mais frequente em D
- 3) Senão, calcule $X = \text{Ganho}(A)$ ¹
 - a) Etiqueta N com o atributo X
 - b) Para cada valor $x \in X$
 - Cria filho F ligado a N por um ramo com rótulo x
 - Associe a F o conjunto de registros $D' = \{d \in D \mid d.X = x\}$
 - Chama a rotina $GeraArvore(D', A - \{X\})$

¹Atributo com maior ganho de informação

Cálculo do Ganho de Informação

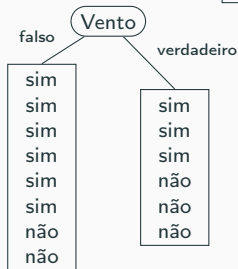
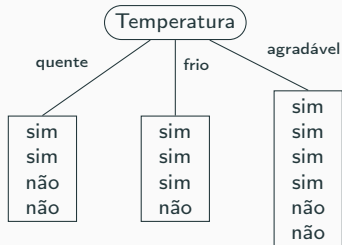
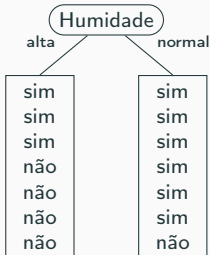
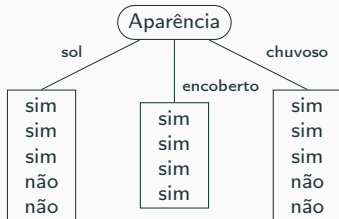
- Vamos considerar um exemplo para tentar decidir se um determinado jogo pode ser jogado

Base de Dados sobre Jogo

Aparência	Temperatura	Humidade	Vento	Jogo
sol	quente	alta	falso	não
sol	quente	alta	verdade	não
encoberto	quente	alta	falso	sim
chuvoso	agradável	alta	falso	sim
chuvoso	frio	normal	falso	sim
chuvoso	frio	normal	verdade	não
encoberto	frio	normal	verdade	sim
sol	agradável	alta	falso	não
sol	frio	normal	falso	sim
chuvoso	agradável	normal	falso	sim
sol	agradável	normal	verdade	sim
encoberto	agradável	alta	verdade	sim
encoberto	quente	normal	falso	sim
chuvoso	agradável	alta	verdade	não

Escolha dos Atributos

- Qual atributo devemos escolher para dividir a árvore na raiz?
- O ideal é aquele que produz filhos mais puros



Ganho de Informação

- A escolha dos atributos se dá com base no ganho de informação
- O ganho de informação considera o grau de impureza do nó pai (antes da divisão) e o grau de impureza dos filhos após a divisão
- O grau de impureza é calculado usando a entropia

Grau de Impureza de um Nó N

$$I(N) = \sum_{i=1}^n \frac{n_i}{T} \text{Entropia}(F_i)$$

T : Quantidade total de registros

n_i : Quantidade de registros na folha F_i

- A entropia mede o quão desorganizada está a informação

Entropia de uma Folha F_i

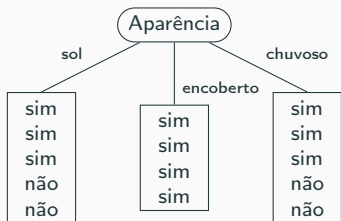
$$\text{Entropia} = -\left(\frac{N_{\text{sim}}}{n_i} \log \frac{N_{\text{sim}}}{n_i} + \frac{N_{\text{não}}}{n_i} \log \frac{N_{\text{não}}}{n_i}\right)$$

n_i : Quantidade de registros na folha F_i

$N_{\text{não}}$: Quantidade “não” na folha F_i

N_{sim} : Quantidade “sim” na folha F_i

Grau de Impureza Dividindo pelo Atribuo Aparência



$$I(\text{Aparência}) = \frac{5}{14} \text{Entropia}(\text{Folha 1}) \\ + \frac{4}{14} \text{Entropia}(\text{Folha 2}) \\ + \frac{5}{14} \text{Entropia}(\text{Folha 3})$$

$$\text{Entropia}(\text{Folha 1}) = \frac{2}{5} \log_2 \frac{2}{5} + \frac{3}{5} \log_2 \frac{3}{5} = 0.971$$

$$\text{Entropia}(\text{Folha 2}) = \frac{4}{4} \log_2 \frac{5}{5} + \frac{0}{4} \log_2 \frac{0}{4} = 0$$

$$\text{Entropia}(\text{Folha 3}) = \frac{3}{5} \log_2 \frac{3}{5} + \frac{2}{5} \log_2 \frac{2}{5} = 0.971$$

$$I(\text{Aparência}) = \frac{5}{14} 0.971 + \frac{4}{14} 0 + \frac{5}{14} 0.971 \\ = 0.693$$

Cálculo do Ganho de Informação

Grau de Impureza das Demais Divisões

$$\begin{aligned}I(\text{Temperatura}) &= \frac{4}{14}\text{entropia}(\text{Folha 1}) + \frac{6}{14}\text{entropia}(\text{Folha 2}) + \frac{4}{14}\text{entropia}(\text{Folha 3}) \\ &= 0.911\end{aligned}$$

$$I(\text{Humidade}) = \frac{7}{14}\text{entropia}(\text{Folha 1}) + \frac{7}{14}\text{entropia}(\text{Folha 2}) = 0.788$$

$$I(\text{Vento}) = \frac{8}{14}\text{entropia}(\text{Folha 1}) + \frac{6}{14}\text{entropia}(\text{Folha 2}) = 0.892$$

Impureza do Pai (registros em um única folha)

$$\frac{9}{14} \log_2 \frac{9}{14} + \frac{5}{14} \log_2 \frac{5}{14} = 0.940$$

Ganho de Informação (diferença das impurezas)

$$\text{Aparência} = 0.940 - 0.693 = 0.247$$

$$\text{Temperatura} = 0.940 - 0.911 = 0.029$$

$$\text{Humidade} = 0.940 - 0.788 = 0.152$$

$$\text{Vento} = 0.940 - 0.892 = 0.020$$

Cálculo do Ganho de Informação

Grau de Impureza das Demais Divisões

$$\begin{aligned}I(\text{Temperatura}) &= \frac{4}{14}\text{entropia}(\text{Folha 1}) + \frac{6}{14}\text{entropia}(\text{Folha 2}) + \frac{4}{14}\text{entropia}(\text{Folha 3}) \\ &= 0.911\end{aligned}$$

$$I(\text{Humidade}) = \frac{7}{14}\text{entropia}(\text{Folha 1}) + \frac{7}{14}\text{entropia}(\text{Folha 2}) = 0.788$$

$$I(\text{Vento}) = \frac{8}{14}\text{entropia}(\text{Folha 1}) + \frac{6}{14}\text{entropia}(\text{Folha 2}) = 0.892$$

Impureza do Pai (registros em um única folha)

$$\frac{9}{14} \log_2 \frac{9}{14} + \frac{5}{14} \log_2 \frac{5}{14} = 0.940$$

Ganho de Informação (diferença das impurezas)

$$\text{Aparência} = 0.940 - 0.693 = 0.247$$

$$\text{Temperatura} = 0.940 - 0.911 = 0.029$$

$$\text{Humidade} = 0.940 - 0.788 = 0.152$$

$$\text{Vento} = 0.940 - 0.892 = 0.020$$

Cálculo do Ganho de Informação

Grau de Impureza das Demais Divisões

$$\begin{aligned}I(\text{Temperatura}) &= \frac{4}{14}\text{entropia}(\text{Folha 1}) + \frac{6}{14}\text{entropia}(\text{Folha 2}) + \frac{4}{14}\text{entropia}(\text{Folha 3}) \\ &= 0.911\end{aligned}$$

$$I(\text{Humidade}) = \frac{7}{14}\text{entropia}(\text{Folha 1}) + \frac{7}{14}\text{entropia}(\text{Folha 2}) = 0.788$$

$$I(\text{Vento}) = \frac{8}{14}\text{entropia}(\text{Folha 1}) + \frac{6}{14}\text{entropia}(\text{Folha 2}) = 0.892$$

Impureza do Pai (registros em um única folha)

$$\frac{9}{14} \log_2 \frac{9}{14} + \frac{5}{14} \log_2 \frac{5}{14} = 0.940$$

Ganho de Informação (diferença das impurezas)

$$\text{Aparência} = 0.940 - 0.693 = 0.247$$

$$\text{Temperatura} = 0.940 - 0.911 = 0.029$$

$$\text{Humidade} = 0.940 - 0.788 = 0.152$$

$$\text{Vento} = 0.940 - 0.892 = 0.020$$

- Uma árvore de decisão pode ser transformada em um conjunto de regras de classificação
- Para cada caminho, da raiz até uma folha, tem-se uma regra de classificação
- Os rótulos e os valores nos atributos são usados para definir as condições das regras

1. Finalizar a árvore de decisão da base de dados de jogos
2. Calcular a precisão desta árvore usando a base de dados disponível
3. Preparar a base de dados para o Weka, executar os algoritmos de árvore de decisão do Weka e comparar com a árvore desenvolvida

Características das Árvores de Decisão I

- A construção das árvores de decisão não requer conhecimento sobre o tipo de distribuição de dados
- Encontrar uma árvore de decisão ótima é um problema NP-completo. Muitos algoritmos de árvore de decisão empregam uma abordagem baseada em heurística para guiar sua busca
- As técnicas desenvolvidas para a construção de árvores de decisão são computacionalmente baratas, tornando possível a construção rápida de modelos, mesmo quando o tamanho do conjunto de treinamento é muito grande. Além disso, uma vez que uma árvore de decisão foi construída, a classificação de um registro de teste é extremamente rápida, com uma complexidade de pior caso de $O(w)$, onde, w é a altura da árvore.

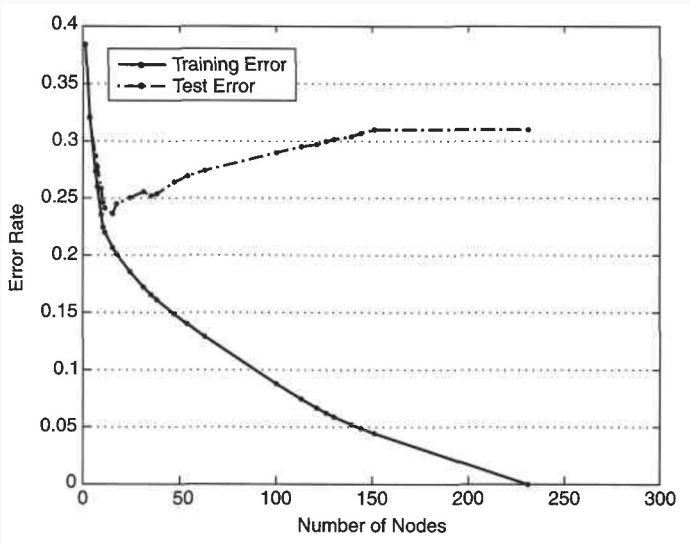
Características das Árvores de Decisão II

- Árvores de decisão, especialmente árvores de menor porte, são relativamente fáceis de interpretar. As precisões das árvores também são comparáveis a outras técnicas de classificação para muitos conjuntos de dados simples
- A presença de atributos redundantes não afeta negativamente a precisão das árvores de decisão. Um dos dois atributos redundantes não será usado para dividir depois que o outro atributo tiver sido escolhido
- Como a maioria dos algoritmos de árvore de decisão emprega uma abordagem de particionamento recursiva, o número de registros se torna menor nas folhas. Assim, o número de registros pode ser muito pequeno para tomar uma decisão estatisticamente significativa sobre a classe. Isso é conhecido como problema de **fragmentação de dados**. Uma solução possível é proibir a divisão adicional quando o número de registros ficar abaixo de um determinado limite

Overfitting de Modelo

- Os erros cometidos por um modelo de classificação são geralmente divididos em dois tipos:
 - Erro de treinamento:** também conhecido como erro de re-substituição ou erro aparente, é o número de erros de classificação incorreta cometidos nos registros de treinamento
 - Erro de generalização:** é o erro esperado do modelo em registros não vistos anteriormente
- Um bom modelo de classificação não deve apenas se ajustar bem os dados de treinamento, mas também classificar com precisão os registros que nunca viu antes
- Ou seja, um bom modelo deve ter baixo erro de treinamento e baixo erro de generalização
- Quando o modelo possui baixo erro de treinamento e alto erro de generalização, dizemos que há **overfitting**

Exemplo



Overfitting Devido a Presença de Ruídos

Conjunto de Treinamento com Ruído

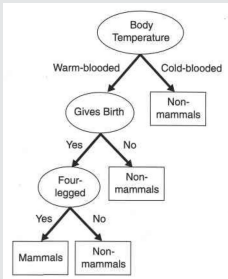
Name	Blood	Birth	Four-Legged	Hibernates	Class
porcupine	warm-blooded	yes	yes	yes	yes
cat	warm-blooded	yes	yes	no	yes
bat	warm-blooded	yes	no	yes	no
whale	warm-blooded	yes	no	no	no
salamander	cold-blooded	no	yes	yes	no
komodo dragon	cold-blooded	no	yes	no	no
python	cold-blooded	no	no	yes	no
salmon	cold-blooded	no	no	no	no
eagle	warm-blooded	no	no	no	no
guppy	cold-blooded	yes	no	no	no

Registros a serem classificados

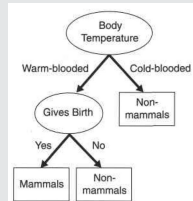
Name	Blood	Birth	Four-Legged	Hibernates
human	warm-blooded	yes	no	no
dolphin	warm-blooded	yes	no	no

Overfitting Devido a Presença de Ruídos

Árvore com Overfitting



Árvore sem Overfitting

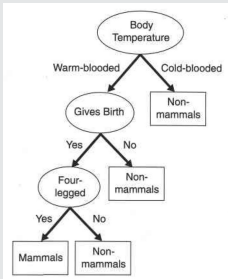


Registros a serem classificados

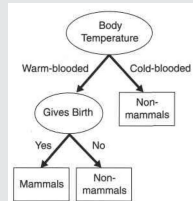
Name	Blood	Birth	Four-Legged	Hibernates
human	warm-blooded	yes	no	no
dolphin	warm-blooded	yes	no	no

Overfitting Devido a Presença de Ruídos

Árvore com Overfitting



Árvore sem Overfitting



Registros a serem classificados

Name	Blood	Birth	Four-Legged	Hibernates
human	warm-blooded	yes	no	no
dolphin	warm-blooded	yes	no	no

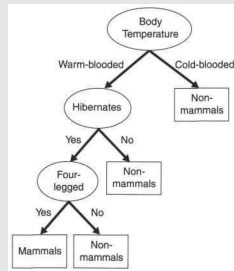
Overfitting Devido a Falta de Amostras Representativas

- Modelos que tomam suas decisões de classificação com base em um pequeno número de registros de treinamento também estão suscetíveis ao overfitting.
- Tais modelos podem ser gerados devido à falta de amostras representativas nos dados de treinamento

Exemplo

Name	Blood	Birth	Four-Legged	Hibernates	Class
salamander	cold-blooded	no	yes	yes	no
guppy	cold-blooded	yes	no	no	no
eagle	warm-blooded	no	no	no	no
poorwill	warm-blooded	no	no	yes	no
platypus	warm-blooded	no	yes	yes	yes

- A árvore possui 0% de erro no treinamento
- Mas classifica errado registros como humanos, elefantes e golfinhos



Avaliando o Desempenho de um Classificador

- Geralmente, é útil medir o desempenho do modelo no conjunto de testes, pois essa medida fornece uma estimativa imparcial de seu erro de generalização
- A precisão ou a taxa de erro calculada a partir do conjunto de testes também pode ser usada para comparar o desempenho relativo de diferentes classificadores
- Vamos considerar os seguintes métodos:
 - *Holdout*
 - Sub-amostragem aleatória
 - Validação cruzada
 - *Bootstrap*

Holdout (ou Split-Sample)

- Uma das técnica mais simples. Faz uma única partição da amostra
- A maior parte dos dados é usada para treinamento, e o restante é usado como dados de teste (ou validação)
- Divisão dos dados
 - Treinamento: 2/3 (por exemplo)
 - Teste: dados restantes
- Indicado para uma grande quantidade de dados
- Problemas com pequenas quantidade de dados:
 - Problemas em pequena quantidade de dados
 - Poucos dados de treinamento
 - Quanto menor o conjunto de treinamento, maior a variância do classificador (instabilidade)
 - Quanto Menor o conjunto de teste, menos confiável
 - Classes sub-representadas ou super-representadas

Sub-Amostragem Aleatória (Random Subsampling)

- Múltiplas execuções de Holdout, com diferentes partições treinamento-teste escolhidas de forma aleatória
- Não pode haver interseção entre os conjuntos de teste e treinamento
- Permite uma estimativa de erro mais precisa
- O erro de classificação é a média dos erros de cada execução

Validação Cruzada (ou *Cross Validation*) I

- Classe de métodos de particionamento de dados
- É comumente usado quando a quantidade de dados disponível é pequena
- Consiste em particionar o conjunto de dados em subconjuntos mutualmente exclusivos
- Utiliza-se alguns subconjuntos para treinamento e o restante para teste

Validação Cruzada (ou *Cross Validation*) II

Método k-fold


- consiste em dividir o conjunto de dados em k subconjuntos mutualmente exclusivos de mesmo tamanho
- A cada iteração, uma das k partições é usada para testar o modelo
- As outras $k-1$ são usadas para treinar o modelo
- Cada objeto participa o mesmo número de vezes do treinamento ($k-1$ vezes)
- Cada objeto participa o mesmo número de vezes do teste (1 vez)
- A taxa de erro é a média dos erros das k partições

Validação Cruzada (ou *Cross Validation*) III

Método *Leave-one-out*

- Trata-se de um caso específico do k-fold
- Nesse caso, o valor de k é igual ao número total de dados (N)
- Cada objeto participa o mesmo número de vezes do treinamento N-1 vezes
- Cada objeto participa o mesmo número de vezes do teste 1 vez
- Vantagens:
 - Investigação completa sobre a variação do modelo em relação aos dados utilizados
 - Estimativa de erro é não tendenciosa, ou seja, tende à taxa verdadeira
- Desvantagens:
 - Alto custo computacional
 - Indicado para uma quantidade pequena de dados

- É uma técnica de amostragem
- Visa a obtenção de um “novo” conjunto de dados, por amostragem do conjunto de dados original
- Ao invés de usar sub-conjuntos dos dados, usa-se sub-amostras
- Funciona melhor que cross-validation para conjuntos muito pequenos
- A amostragem é feita com reposição (sem substituição)
 - Dado um conjunto com N objetos
 - Sorteia-se um objeto para compor a sub-amostra
 - Devolve-se o objeto sorteado ao conjunto de dados
 - Repete-se esse processo até compor uma sub-amostra de tamanho N
- A sub-amostra gerada será o conjunto de treinamento
- Os objetos restantes (que não fazem parte do treinamento) são o conjunto de teste
- De modo geral, a sub-amostra tem 63,2% de objetos não repetidos
- Processo é repetido b vezes
- O resultado é a média dos experimentos

 Quinlan, J. R. (1986).
Induction of decision trees.
Machine learning, 1(1):81–106.

 TAN, P.-N., STEINBACH, M., and KUMAR, V. (2009).
Introdução ao data mining: mineração de dados.
Ciência Moderna, Rio de Janeiro.