CIÊNCIA DE DADOS - 02

Prof. Júlio Cesar Nievola
PPGIa – PUCPR
27/abril/2019

Classificação: Definição

- Dada uma coleção de registros (*conjunto de treinamento*)
  - Cada registro contém um conjunto de *atributos* e um dos atributos é a *classe*.
- Encontrar um *modelo* para o atributo classe como uma função dos valores dos outros atributos.
- Objetivo: aos registros previamente desconhecidos deve ser assinalada uma classe tão precisamente quanto possível.
  - Um *conjunto de teste* é usado para determinar a precisão do modelo. Geralmente o conjunto de dados fornecido é dividido em conjuntos de treinamento e testes, com o conjunto de treinamento sendo usado para construir o modelo e o conjunto de testes sendo usado para validá-lo.

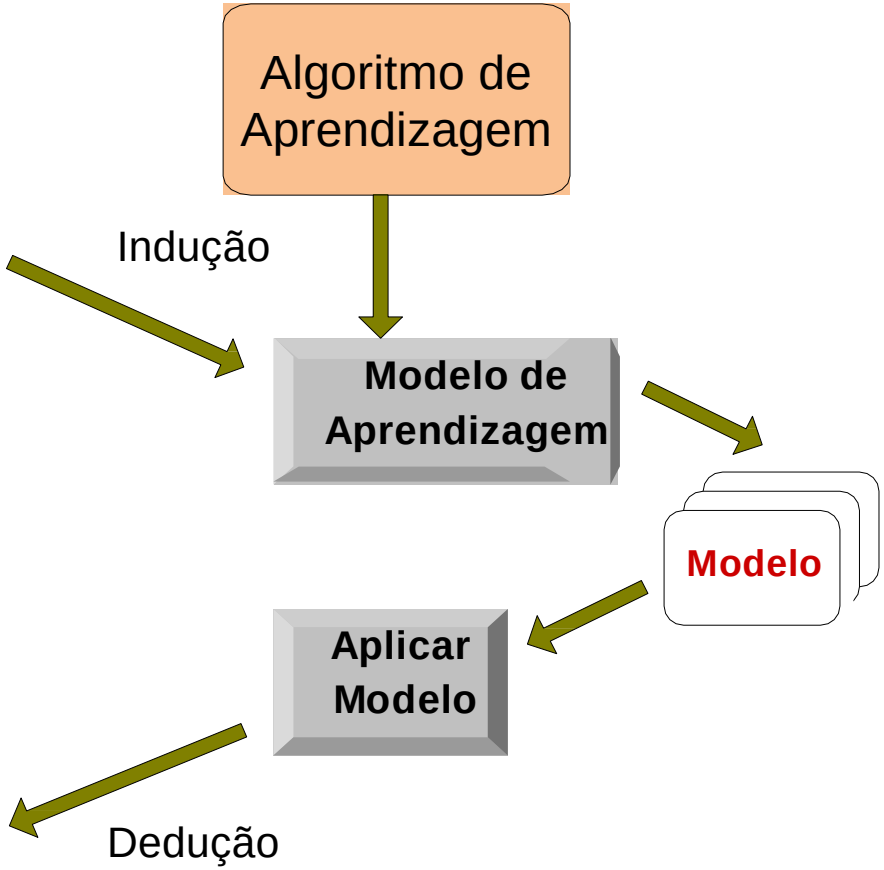
Ilustrando a Tarefa de Classificação

ID	Attrib1	Attrib2	Attrib3	Classe
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

Conjunto de Treinamento

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	?

Conjunto de Teste

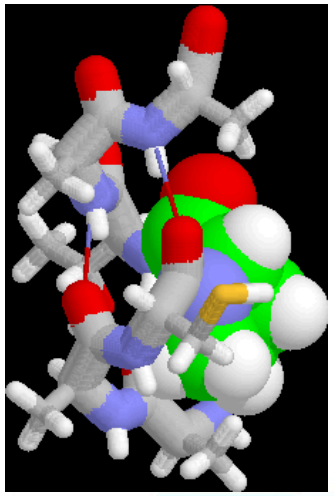


Exemplos da Tarefa de Classificação

- Prever se tumor em células é benigno ou maligno

- Classificar transações de cartão de crédito como legítimas ou fraude

- Classificar estruturas secundárias de proteínas como alpha-helix, beta-sheet, ou random coil



- Categorizar textos novos como finanças, tempo, lazer, esportes, etc

Técnicas de Classificação

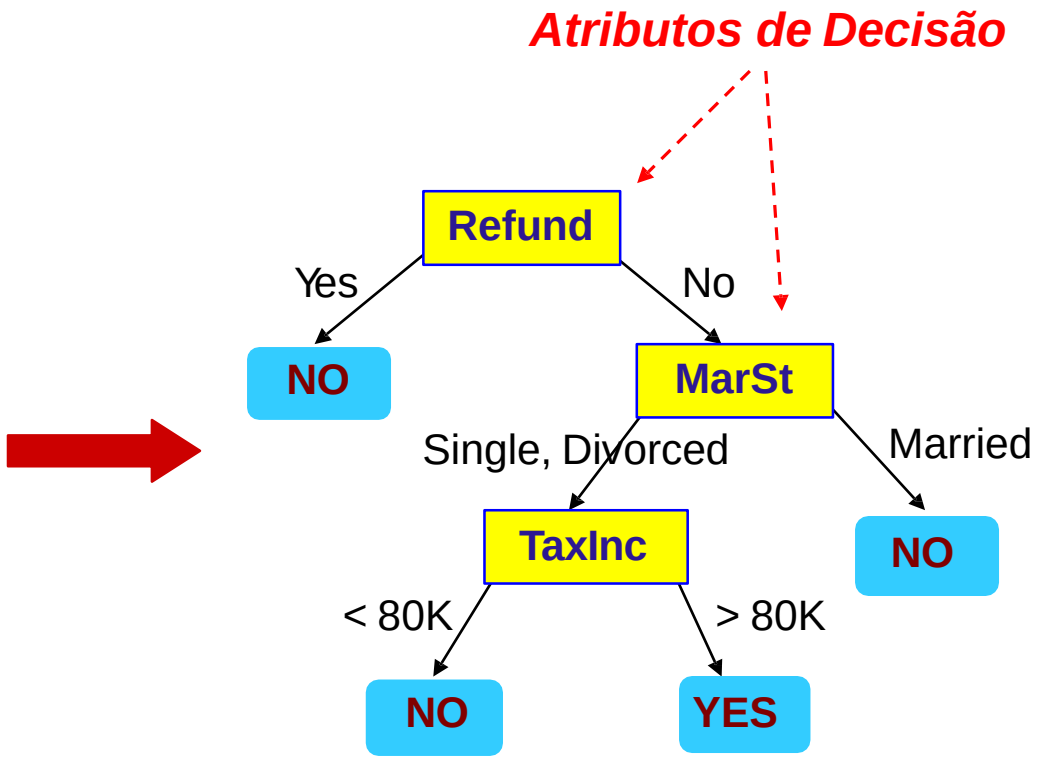
- Métodos baseados em Árvore de Decisão
- Métodos baseados em Regras
- Raciocínio baseado em Memória
- Redes Neurais Artificiais (e “Deep Learning”)
- Naïve Bayes e Redes Baiesianas de Crença
- Máquinas de Vetores de Suporte – SVM

MÉTODOS BASEADOS EM ÁRVORE DE DECISÃO

Exemplo de uma Árvore de Decisão

Tid	Refund (categórico)	Marital Status (categórico)	Taxable Income (contínuos)	Cheat (classe)
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

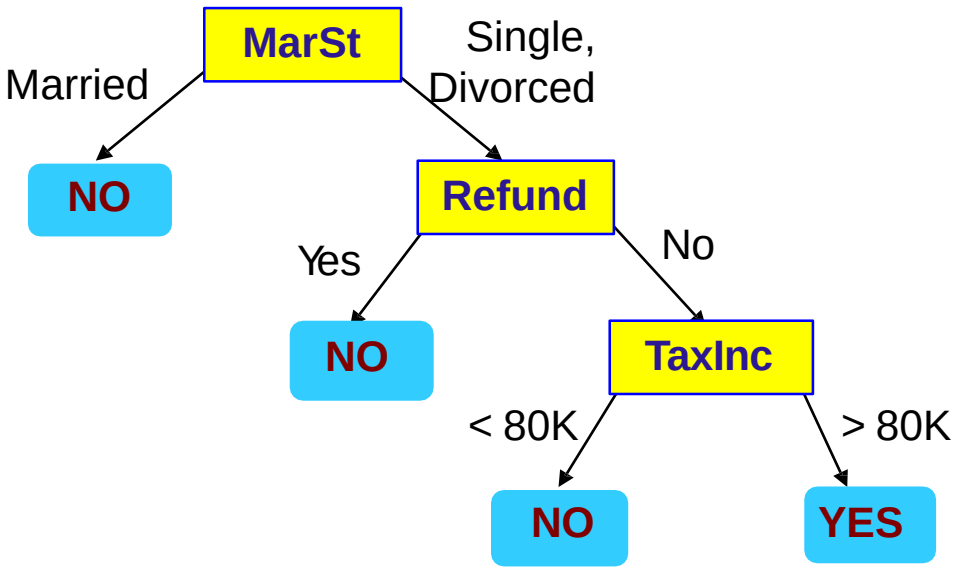
Dados de Treinamento



Modelo: Árvore de Decisão

Outro Exemplo de Árvore de Decisão

	categórico	categórico	contínuos	classe
Tid	**Refund**	**Marital Status**	**Taxable Income**	**Cheat**
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes



Pode haver mais de uma árvore que se ajusta aos mesmos dados!

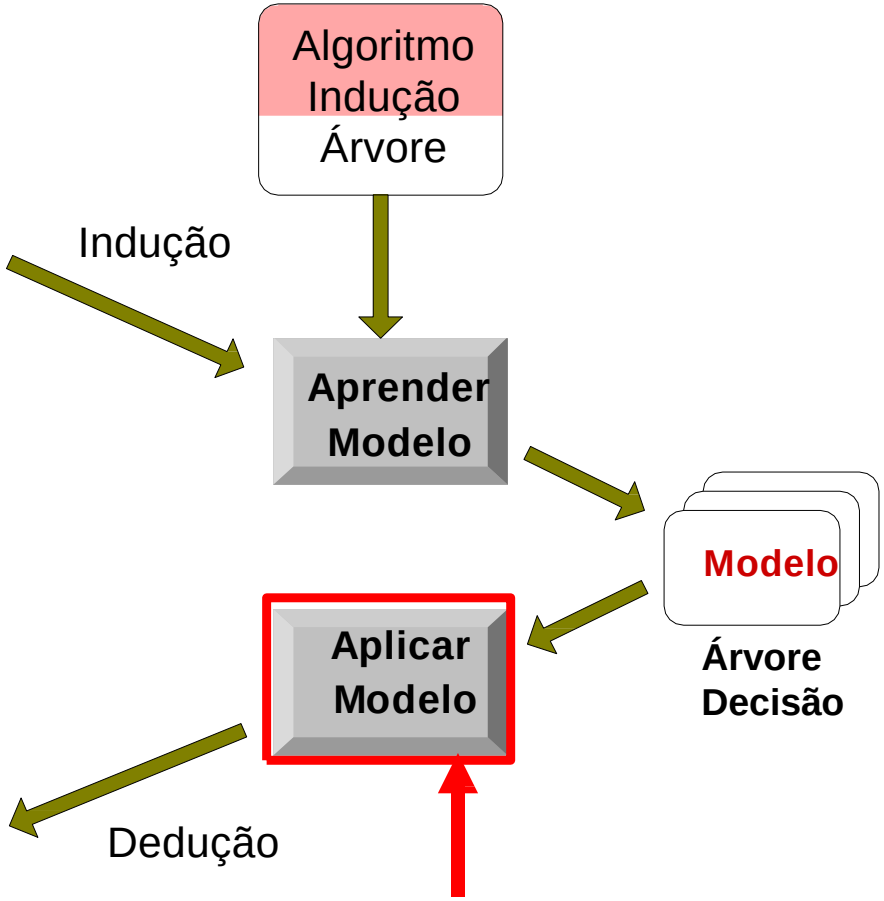
Tarefa de Classificação com Árvore de Decisão

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

Conjunto de Treinamento

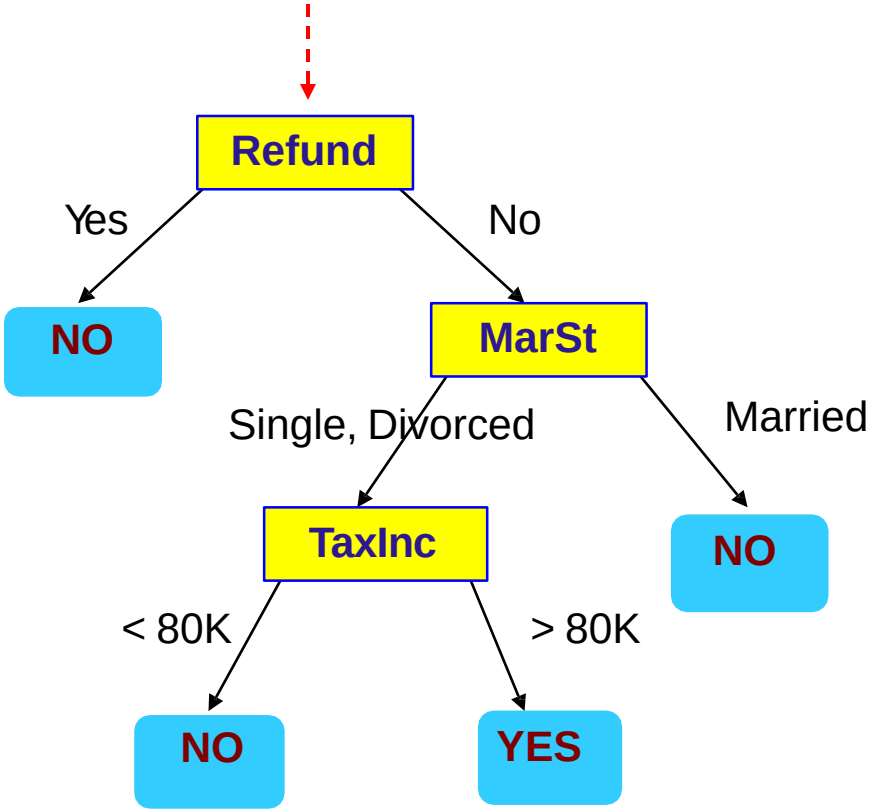
Tid	Attrib1	Attrib2	Attrib3	Class
11	No	Small	55K	?
12	Yes	Medium	80K	?
13	Yes	Large	110K	?
14	No	Small	95K	?
15	No	Large	67K	?

Conjunto de Teste



Aplicar Modelo aos Dados de Teste

Iniciar na raiz da árvore.



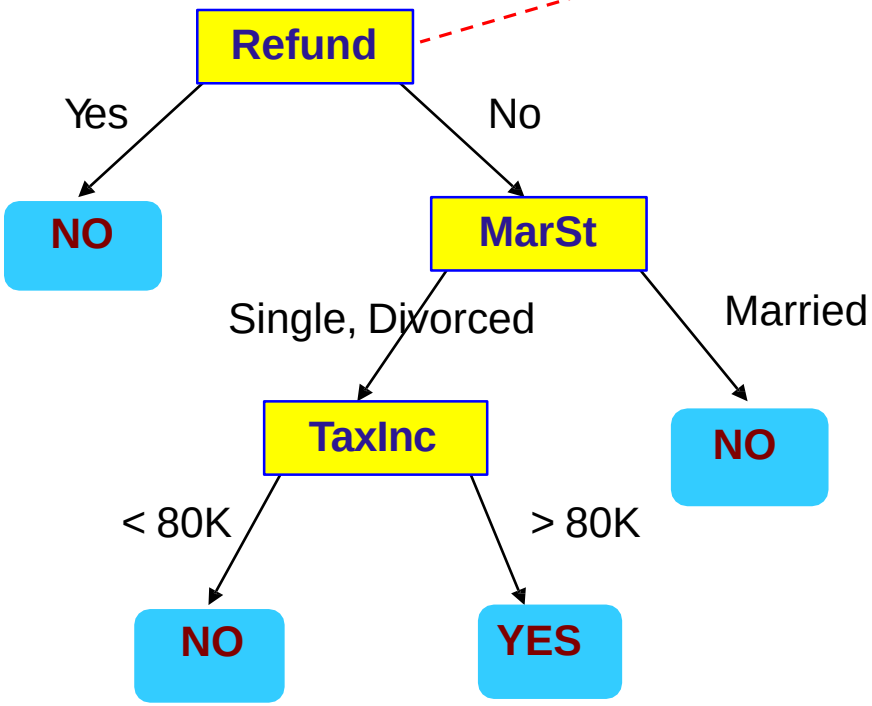
Dado de Teste

Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?

Aplicar Modelo aos Dados de Teste

Dado de Teste

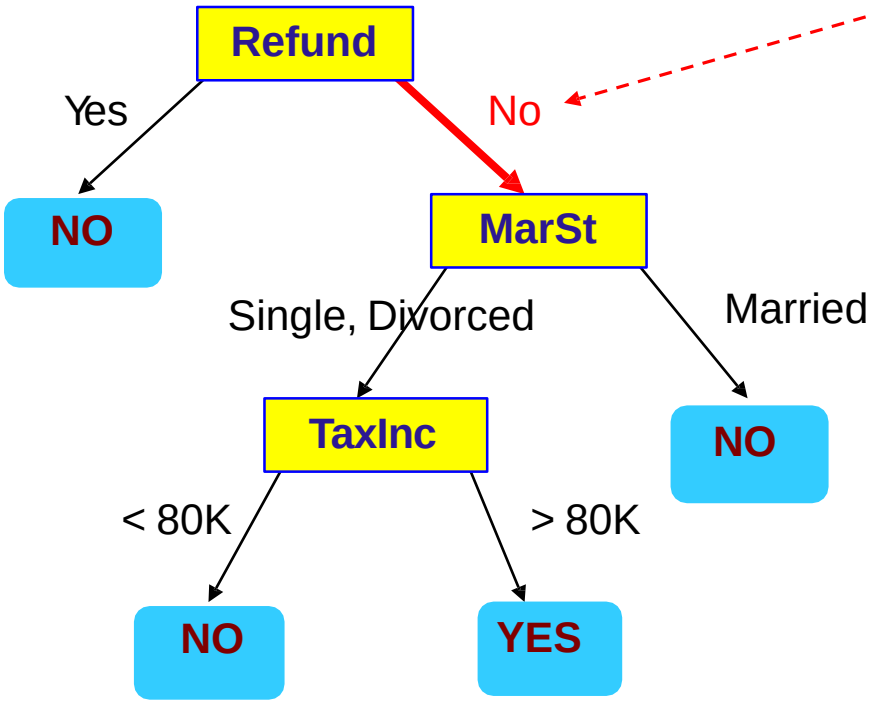
Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?



Aplicar Modelo aos Dados de Teste

Dado de Teste

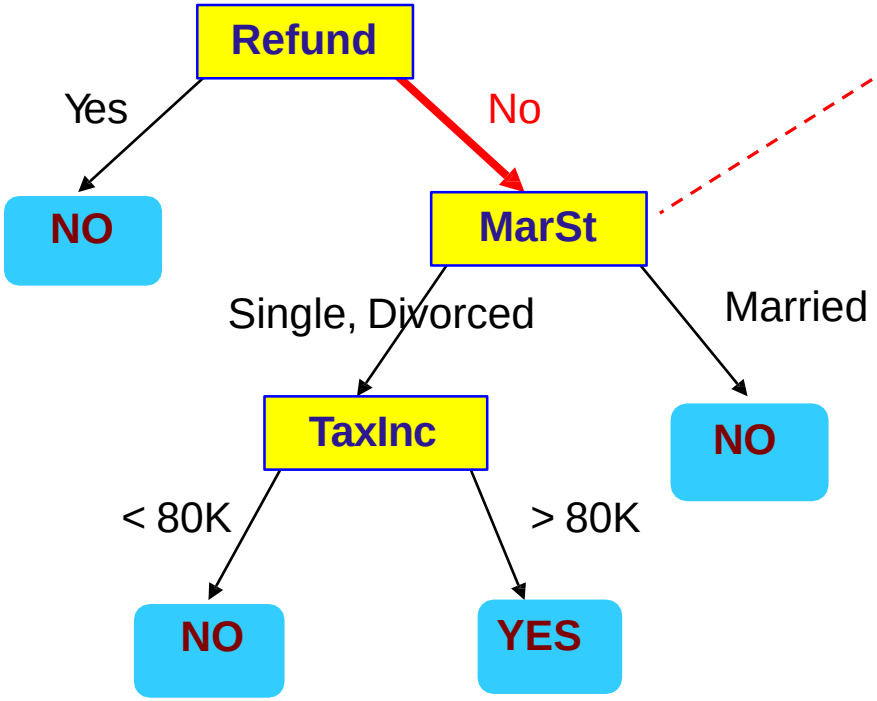
Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?



Aplicar Modelo aos Dados de Teste

Dado de Teste

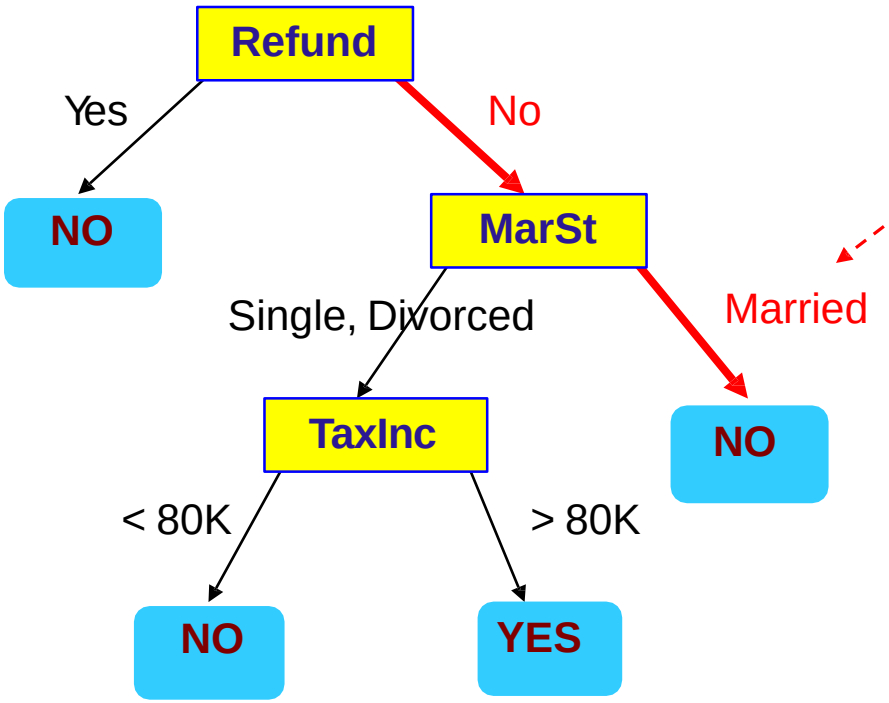
Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?



Aplicar Modelo aos Dados de Teste

Dado de Teste

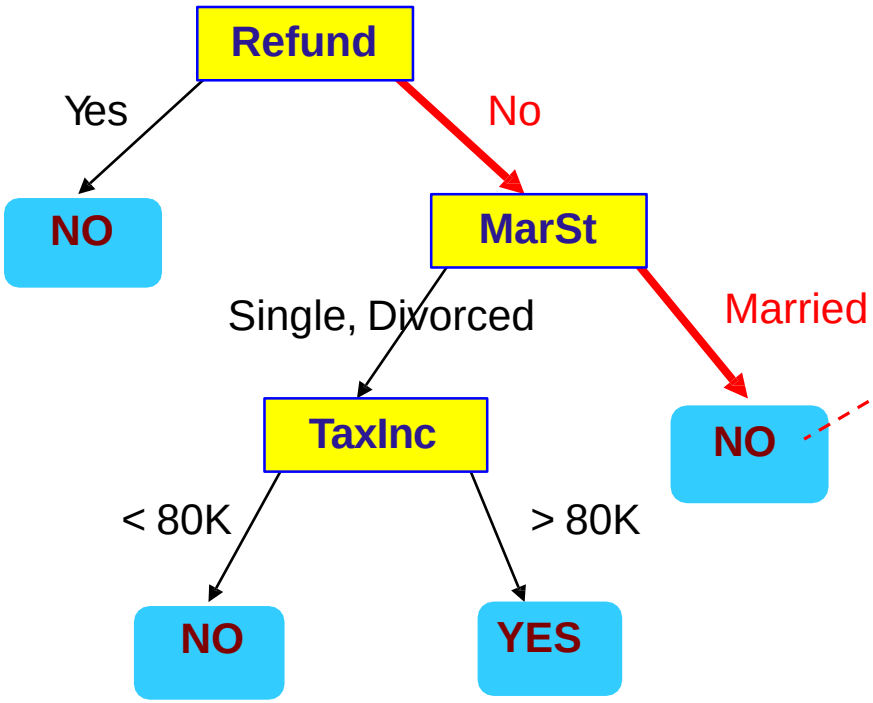
Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?



Aplicar Modelo aos Dados de Teste

Dado de Teste

Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?



Assinalar “No” ao Atributo Cheat

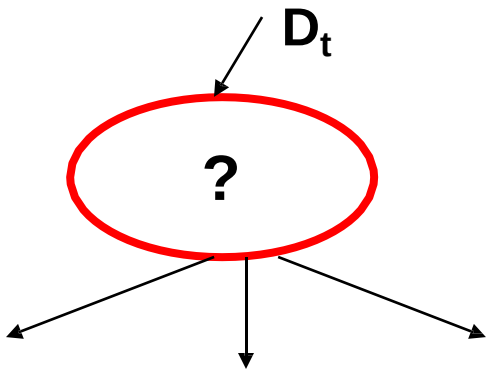
Indução de Árvore de Decisão

- Muitos Algoritmos:
  - Algoritmo de Hunt (um dos primeiros)
  - CART
  - ID3, C4.5
  - SLIQ, SPRINT

Estrutura Geral do Algoritmo de Hunt

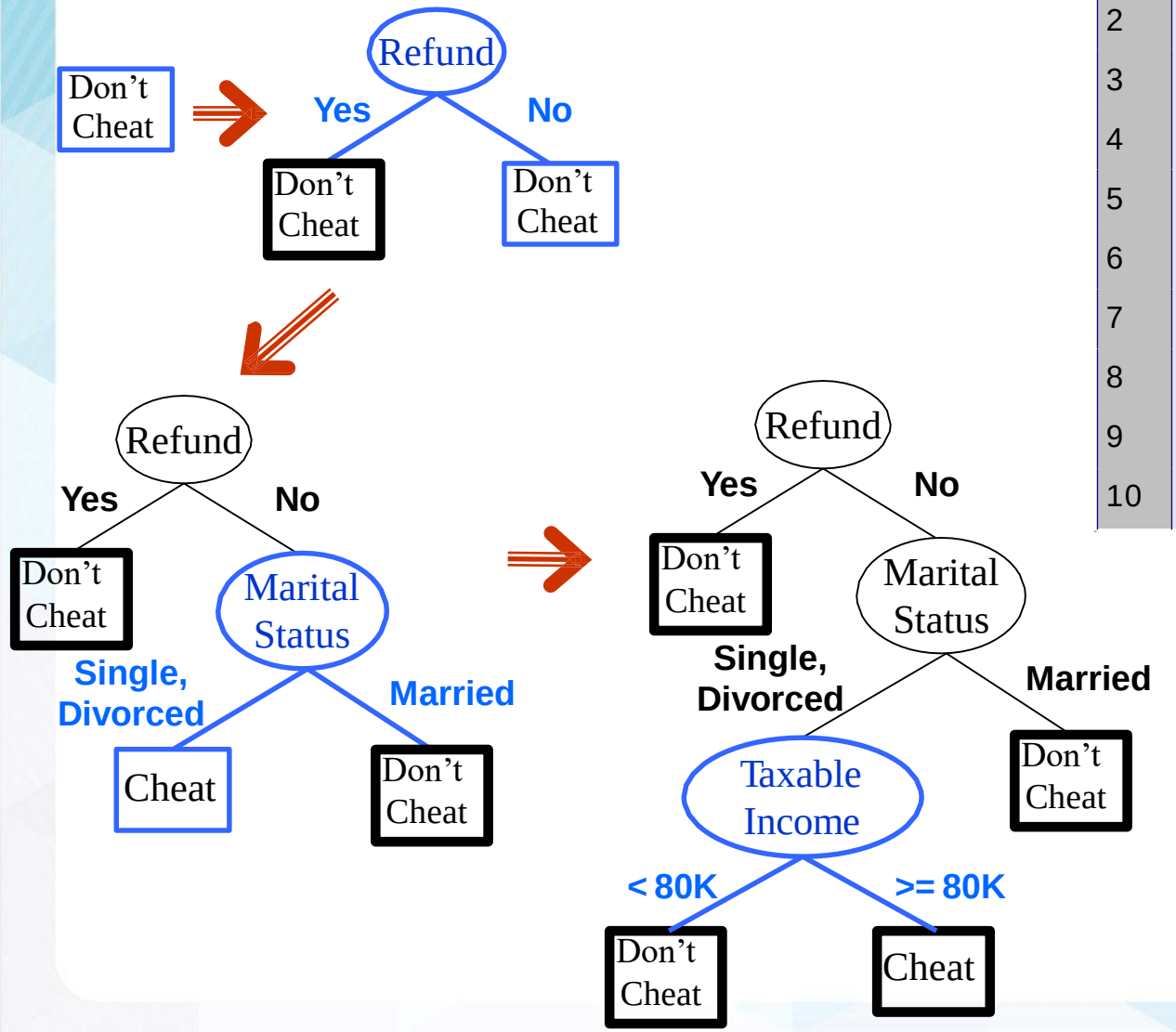
- Seja D_t o conjunto de registros de treinamento que chegam ao nó *t*
- Procedimento Geral:
  - Se D_t contém registros que pertencem à mesma classe y_t, então *t* é um nó folha com nome y_t
  - Se D_t é um conjunto vazio, então *t* é um nó folha com o nome da classe padrão y_d
  - Se D_t contém registros que pertencem a mais de uma classe, usar um teste em atributo para dividir os dados em sub-conjuntos menores. Aplicar recursivamente o procedimento para cada subconjunto.

Tid	Refund	Marital Status	Taxable Income	Cheat
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes



Algoritmo de Hunt

Tid	Refund	Marital Status	Taxable Income	Cheat
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes



Indução de Árvores

- Estratégia gulosa.
  - Dividir os registros baseado sobre o teste do atributo que otimiza determinado critério.

- Pontos a considerar
  - Determinar como dividir os registros
    - Como especificar a condição de teste do atributo?
    - Como determinar a melhor divisão?
  - Determinar quando parar de dividir

Como especificar a condição de teste?

- Depende dos tipos de atributos
  - Nominal
  - Ordinal
  - Contínuo
- Depende do número de formas de dividir
  - 2-way split
  - Multi-way split

Critério de Parada para Árvores de Decisão

- Parar de expandir um nó quando todos os registros pertencem à mesma classe
- Parar de expandir um nó quando todos os registros tem valores de atributo similar
- Terminação precoce

Classificação baseada em Árvore de Decisão

- Vantagens:
  - Sem dificuldades de construção
  - Extremamente rápida para classificar registros desconhecidos
  - Fácil de interpretar para árvores pequenas
  - Precisão comparável a outras técnicas de classificação para muitos conjuntos com dados simples

Exemplo: C4.5

- Construção simples de busca em profundidade.
- Usa Ganho de Informação
- Ordena Atributos Contínuos em cada nó.
- Todos os dados devem caber na memória.
- Não adequado para grandes bases de dados.
  - Necessita de ordenação *out-of-core*.

Outras Questões

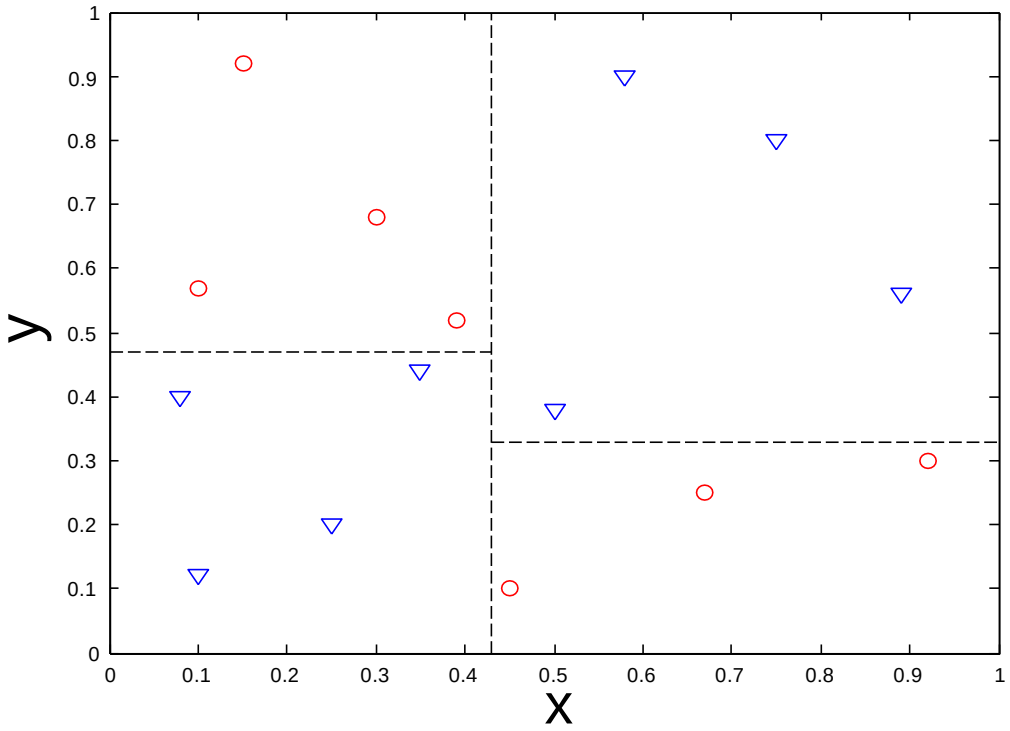
- Fragmentação de Dados
- Superfície de Separação
- Replicação de Árvore

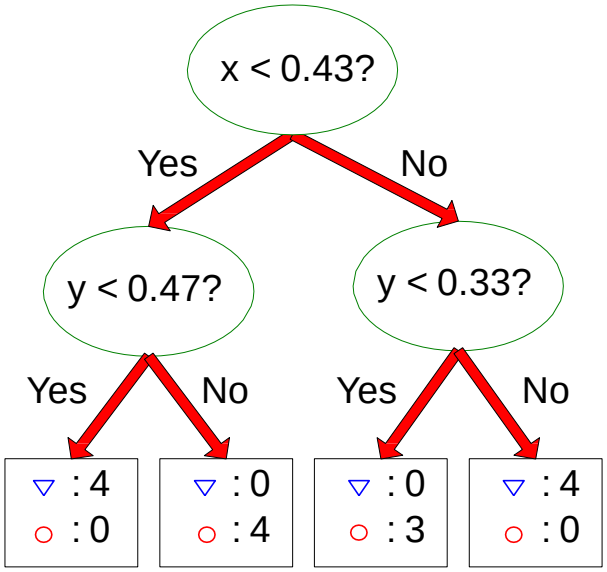
Fragmentação de Dados

- Número de instâncias torna-se menor à medida que se caminha na árvore

- Número de instâncias nos nós folha pode ser muito pequeno para tomar qualquer decisão estatisticamente significante

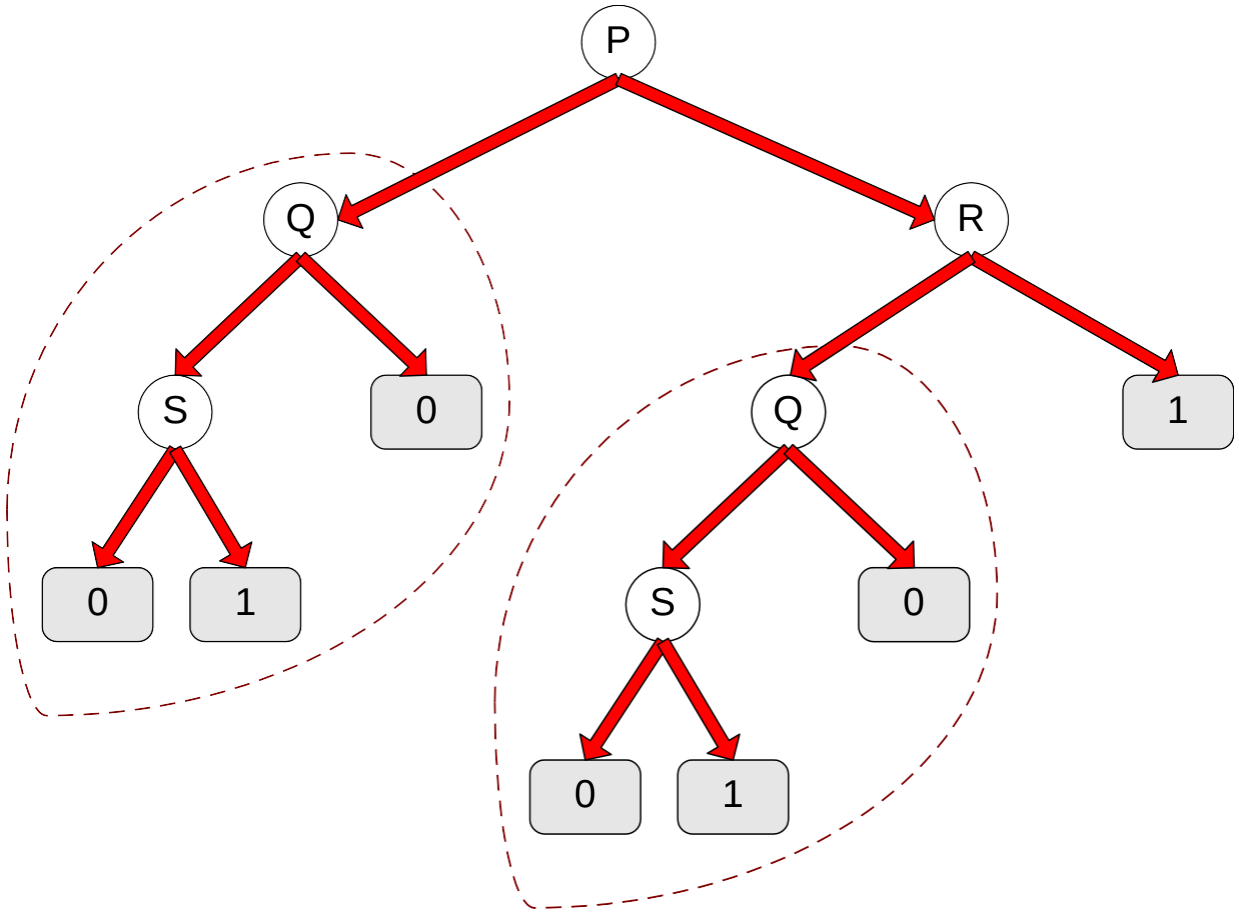
Superfície de separação





- A linha limite entre duas regiões vizinhas com diferentes classes é conhecida como superfície de separação
- A superfície de separação é paralela aos eixos pois a condição de teste envolve um único atributo por vez

Replicação de Árvore



- **Mesma sub-árvore aparece em vários ramos**

MÉTODOS BASEADOS EM REGRAS DE CLASSIFICAÇÃO

Classificadores Baseados em Regras

- Classificar registros usando uma coleção de regras “se…então…”
- Regra: (*Condição*) $\rightarrow y$
  - em que
    - *Condição* é uma conjunção de atributos
    - *y* é o rótulo da classe
  - *LHS*: antecedente da regra ou condição
  - *RHS*: conseqüente da regra
  - Exemplos de regras de classificação:
    - (Blood Type=Warm) $\wedge$ (Lay Eggs=Yes) $\rightarrow$ Birds
    - (Taxable Income < 50K) $\wedge$ (Refund=Yes) $\rightarrow$ Evade=No

Classificador Baseado em Regras (Exemplo)

Name	Blood Type	Give Birth	Can Fly	Live in Water	Class
human	warm	yes	no	no	mammals
python	cold	no	no	no	reptiles
salmon	cold	no	no	yes	fishes
whale	warm	yes	no	yes	mammals
frog	cold	no	no	sometimes	amphibians
komodo	cold	no	no	no	reptiles
bat	warm	yes	yes	no	mammals
pigeon	warm	no	yes	no	birds
cat	warm	yes	no	no	mammals
leopard shark	cold	yes	no	yes	fishes
turtle	cold	no	no	sometimes	reptiles
penguin	warm	no	no	sometimes	birds
porcupine	warm	yes	no	no	mammals
eel	cold	no	no	yes	fishes
salamander	cold	no	no	sometimes	amphibians
gila monster	cold	no	no	no	reptiles
platypus	warm	no	no	no	mammals
owl	warm	no	yes	no	birds
dolphin	warm	yes	no	yes	mammals
eagle	warm	no	yes	no	birds

R1: (Give Birth = no) $\wedge$ (Can Fly = yes) $\rightarrow$ Birds

R2: (Give Birth = no) $\wedge$ (Live in Water = yes) $\rightarrow$ Fishes

R3: (Give Birth = yes) $\wedge$ (Blood Type = warm) $\rightarrow$ Mammals

R4: (Give Birth = no) $\wedge$ (Can Fly = no) $\rightarrow$ Reptiles

R5: (Live in Water = sometimes) $\rightarrow$ Amphibians

Regras

- Uma regra *r* cobre uma instância **x** se os atributos da instância satisfazem a condição da regra

R1: (Give Birth = no) $\wedge$ (Can Fly = yes) $\rightarrow$ Birds

R2: (Give Birth = no) $\wedge$ (Live in Water = yes) $\rightarrow$ Fishes

R3: (Give Birth = yes) $\wedge$ (Blood Type = warm) $\rightarrow$ Mammals

R4: (Give Birth = no) $\wedge$ (Can Fly = no) $\rightarrow$ Reptiles

R5: (Live in Water = sometimes) $\rightarrow$ Amphibians

Name	Blood Type	Give Birth	Can Fly	Live in Water	Class
hawk	warm	no	yes	no	?
grizzly bear	warm	yes	no	no	?

A regra R1 cobre hawk => Bird

A regra R3 cobre grizzly bear => Mammal

Cobertura e Precisão de uma Regra

- Cobertura de uma regra:
  - Fração de registros que satisfazem o antecedente da regra
- Precisão de uma regra:
  - Fração dos registros que satisfazem tanto o antecedente quanto o conseqüente da regra

Tid	Refund	Marital Status	Taxable Income	Class
1	Yes	**Single**	125K	**No**
2	No	Married	100K	**No**
3	No	**Single**	70K	**No**
4	Yes	Married	120K	**No**
5	No	Divorced	95K	**Yes**
6	No	Married	60K	**No**
7	Yes	Divorced	220K	**No**
8	No	**Single**	85K	**Yes**
9	No	Married	75K	**No**
10	No	**Single**	90K	**Yes**

(Status=Single) → No

Cobertura = 40%, Precisão = 50%

Como um Classificador baseado em Regras Funciona?

R1: (Give Birth = no) $\wedge$ (Can Fly = yes) $\rightarrow$ Birds

R2: (Give Birth = no) $\wedge$ (Live in Water = yes) $\rightarrow$ Fishes

R3: (Give Birth = yes) $\wedge$ (Blood Type = warm) $\rightarrow$ Mammals

R4: (Give Birth = no) $\wedge$ (Can Fly = no) $\rightarrow$ Reptiles

R5: (Live in Water = sometimes) $\rightarrow$ Amphibians

Name	Blood Type	Give Birth	Can Fly	Live in Water	Class
lemur	warm	yes	no	no	?
turtle	cold	no	no	sometimes	?
dogfish shark	cold	yes	no	yes	?

Um lemur dispara R3, então é classificado como mammal

Uma turtle dispara tanto R4 quanto R5

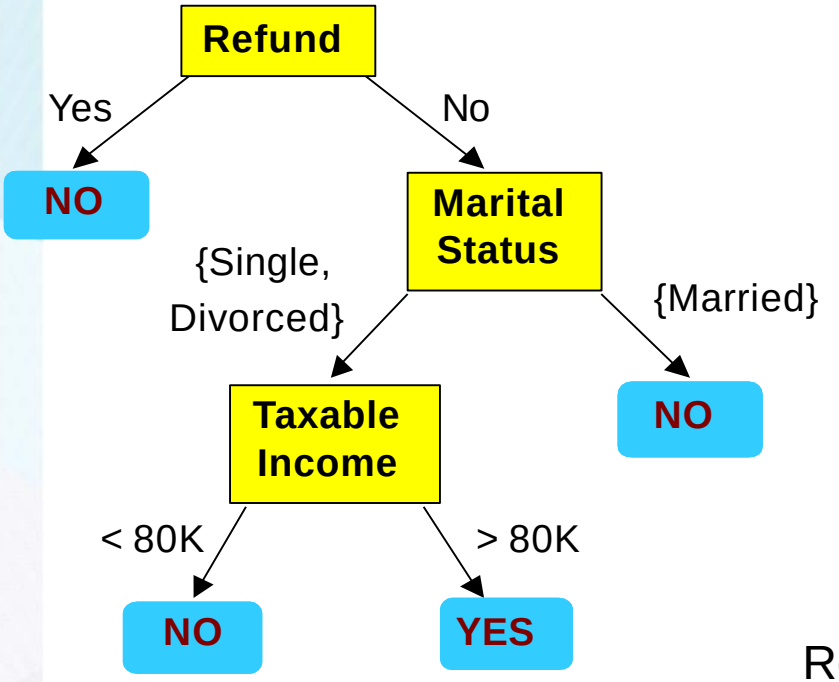
Um dogfish shark não dispara nenhuma das regras

Características dos Classificadores Baseados em Regras

- Regras mutuamente exclusivas
  - Classificador contém regras mutuamente exclusivas se as regras são independentes entre si
  - Cada registro é coberto por no máximo uma regra

- Regras exaustivas
  - Classificador tem cobertura exaustiva se ele leva em conta toda possível combinação dos valores dos atributos
  - Cada registro é coberto por pelo menos uma regra

De Árvores de Decisão para Regras



Classification Rules

(Refund=Yes) ==> No

(Refund=No, Marital Status={Single,Divorced}, Taxable Income<80K) ==> No

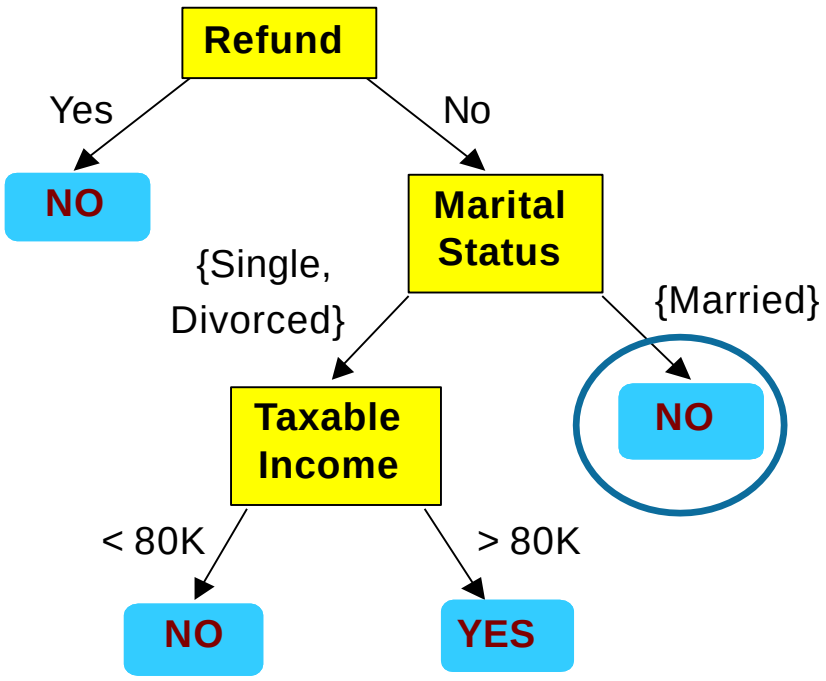
(Refund=No, Marital Status={Single,Divorced}, Taxable Income>80K) ==> Yes

(Refund=No, Marital Status={Married}) ==> No

Regras são mutuamente exclusivas e exaustivas

Conjunto de regras contém tanta informação quanto a árvore

Regras Podem ser Simplificadas



Tid	Refund	Marital Status	Taxable Income	Cheat
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

Regra Inicial: $(\text{Refund=No}) \wedge (\text{Status=Married}) \rightarrow \text{No}$

Regra Simplificada: $(\text{Status=Married}) \rightarrow \text{No}$

Efeito da Simplificação de Regras

- Regras deixam de ser mutuamente exclusivas
  - Um registro pode disparar mais de uma regra
  - Solução?
    - Conjunto de regras ordenadas
    - Conjunto não ordenado de regras – usar esquema de votação
- Regras deixam de ser exaustivas
  - Um registro pode não disparar nenhuma regra
  - Solução?
    - Usar uma classe default

Conjunto de Regras Ordenadas

- Regras são ordenadas de acordo com sua prioridade, chamada lista de decisão
- Quando um registro de teste é apresentado
  - Ele é assinalado ao rótulo de classe da regra de mais alta ordem que tenha sido disparada
  - Se nenhuma das regras dispara, é assinalado à classe default

R1: (Give Birth = no) $\wedge$ (Can Fly = yes) $\rightarrow$ Birds

R2: (Give Birth = no) $\wedge$ (Live in Water = yes) $\rightarrow$ Fishes

R3: (Give Birth = yes) $\wedge$ (Blood Type = warm) $\rightarrow$ Mammals

R4: (Give Birth = no) $\wedge$ (Can Fly = no) $\rightarrow$ Reptiles

R5: (Live in Water = sometimes) $\rightarrow$ Amphibians

Name	Blood Type	Give Birth	Can Fly	Live in Water	Class
turtle	cold	no	no	sometimes	?

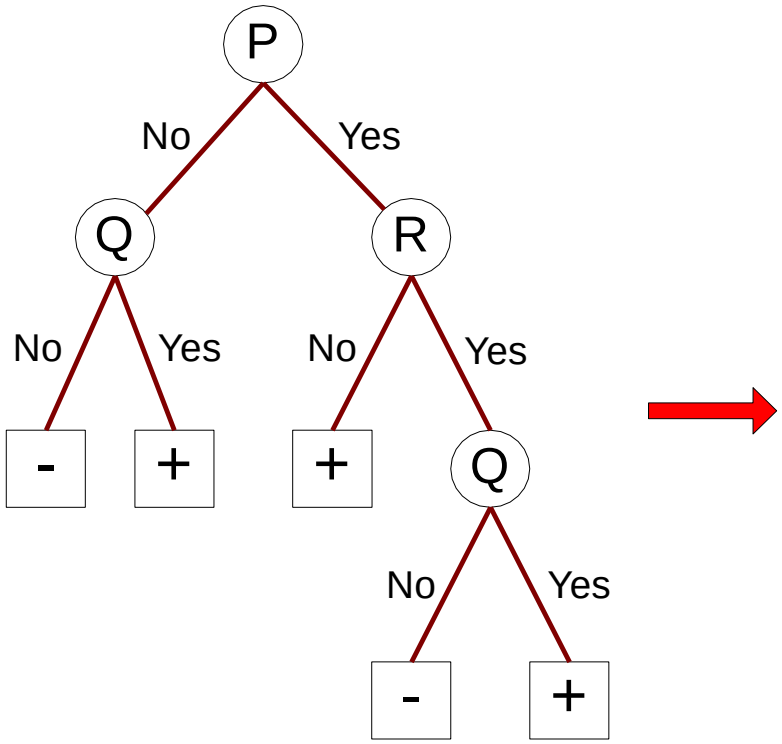
Construindo Regras de Classificação

- Método Direto:
  - Extrair regras diretamente dos dados
  - e.g.: RIPPER, CN2, 1R de Holte
- Método Indireto:
  - Extrair regras de outros modelos de classificação (e.g. árvores de decisão, redes neurais etc.)
  - e.g: C4.5rules

Sumário do Método Direto

- Crescer uma única regra
- Remover Instâncias da regra
- Podar a regra (se necessário)
- Adicionar regra ao conjunto atual de regras
- Repetir

Métodos Indiretos



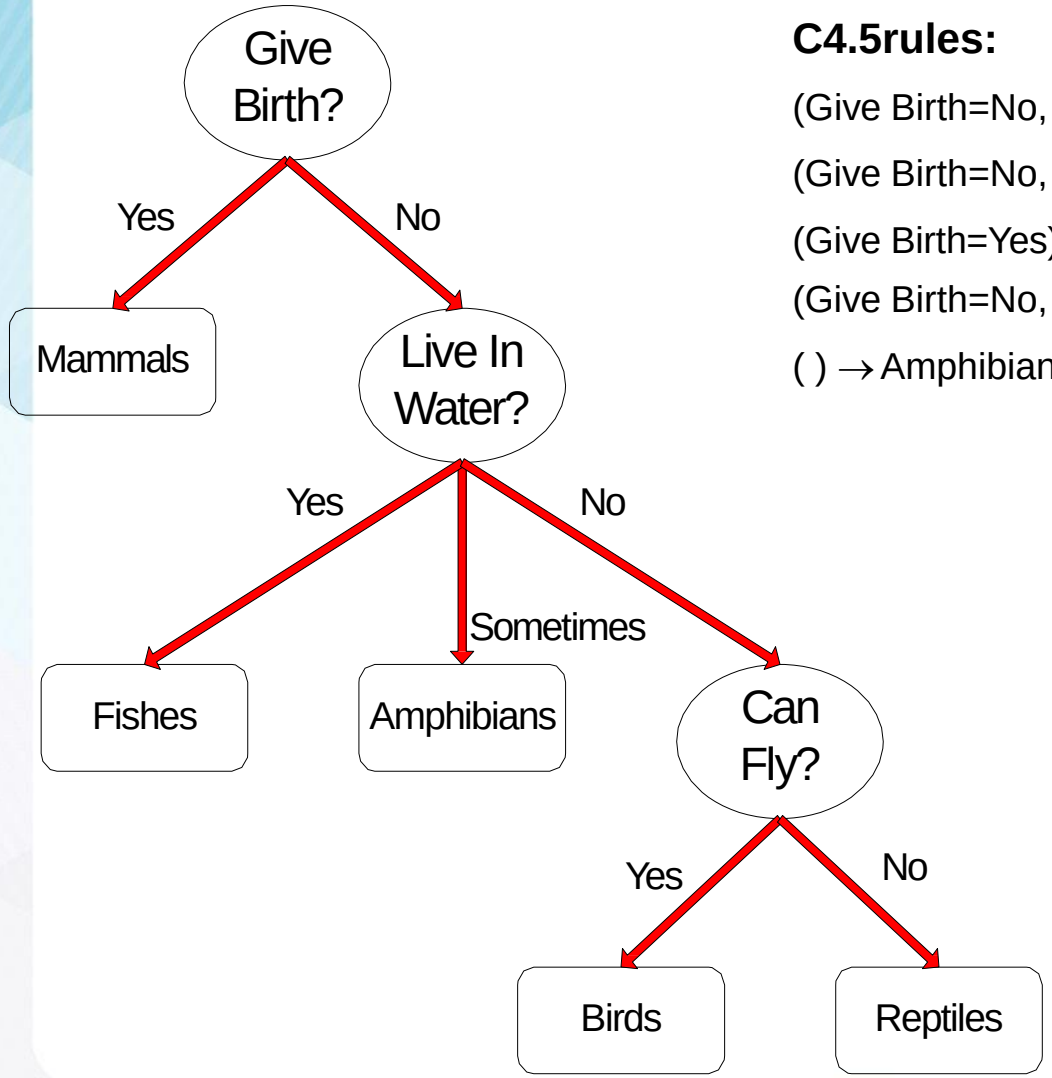
Rule Set

r1: (P=No,Q=No) ==> -
r2: (P=No,Q=Yes) ==> +
r3: (P=Yes,R=No) ==> +
r4: (P=Yes,R=Yes,Q=No) ==> -
r5: (P=Yes,R=Yes,Q=Yes) ==> +

Exemplo

Name	Give Birth	Lay Eggs	Can Fly	Live in Water	Have Legs	Class
human	yes	no	no	no	yes	mammals
python	no	yes	no	no	no	reptiles
salmon	no	yes	no	yes	no	fishes
whale	yes	no	no	yes	no	mammals
frog	no	yes	no	sometimes	yes	amphibians
komodo	no	yes	no	no	yes	reptiles
bat	yes	no	yes	no	yes	mammals
pigeon	no	yes	yes	no	yes	birds
cat	yes	no	no	no	yes	mammals
leopard shark	yes	no	no	yes	no	fishes
turtle	no	yes	no	sometimes	yes	reptiles
penguin	no	yes	no	sometimes	yes	birds
porcupine	yes	no	no	no	yes	mammals
eel	no	yes	no	yes	no	fishes
salamander	no	yes	no	sometimes	yes	amphibians
gila monster	no	yes	no	no	yes	reptiles
platypus	no	yes	no	no	yes	mammals
owl	no	yes	yes	no	yes	birds
dolphin	yes	no	no	yes	no	mammals
eagle	no	yes	yes	no	yes	birds

C4.5 versus C4.5rules versus RIPPER



C4.5rules:

(Give Birth=No, Can Fly=Yes) → Birds

(Give Birth=No, Live in Water=Yes) → Fishes

(Give Birth=Yes) → Mammals

(Give Birth=No, Can Fly=No, Live in Water=No) → Reptiles

( ) → Amphibians

RIPPER:

(Live in Water=Yes) → Fishes

(Have Legs=No) → Reptiles

(Give Birth=No, Can Fly=No, Live In Water=No) → Reptiles

(Can Fly=Yes,Give Birth=No) → Birds

() → Mammals

C4.5 versus C4.5rules versus RIPPER

C4.5 e C4.5rules:

		CLASSE PREVISTA				
		Amphibians	Fishes	Reptiles	Birds	Mammals
CLASSE REAL	Amphibians	2	0	0	0	0
	Fishes	0	2	0	0	1
	Reptiles	1	0	3	0	0
	Birds	1	0	0	3	0
	Mammals	0	0	1	0	6

RIPPER:

		CLASSE PREVISTA				
		Amphibians	Fishes	Reptiles	Birds	Mammals
CLASSE REAL	Amphibians	0	0	0	0	2
	Fishes	0	3	0	0	0
	Reptiles	0	0	3	0	1
	Birds	0	0	1	2	1
	Mammals	0	2	1	0	4

Vantagens de Classificadores baseados em Regras

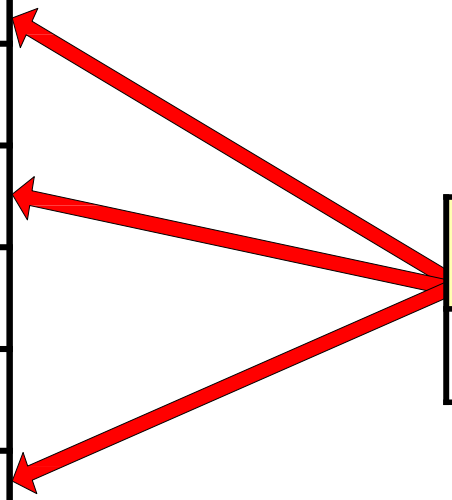
- Tão expressivos quanto as árvores de decisão
- Fáceis de interpretar
- Fáceis de gerar
- Podem classificar rapidamente novas instâncias
- Desempenho comparável às árvores de decisão

MÉTODOS BASEADOS EM INSTÂNCIA ("LAZY LEARNING")

Classificadores baseados em Instância

Casos armazenados

Atr1	……	AtrN	Classe
			A
			B
			B
			C
			A
			C
			B



- **Armazena os registros de treinamento**
- **Usa registros de treinamento para prever o rótulo da classe para casos não-vistos**

Casos Novos

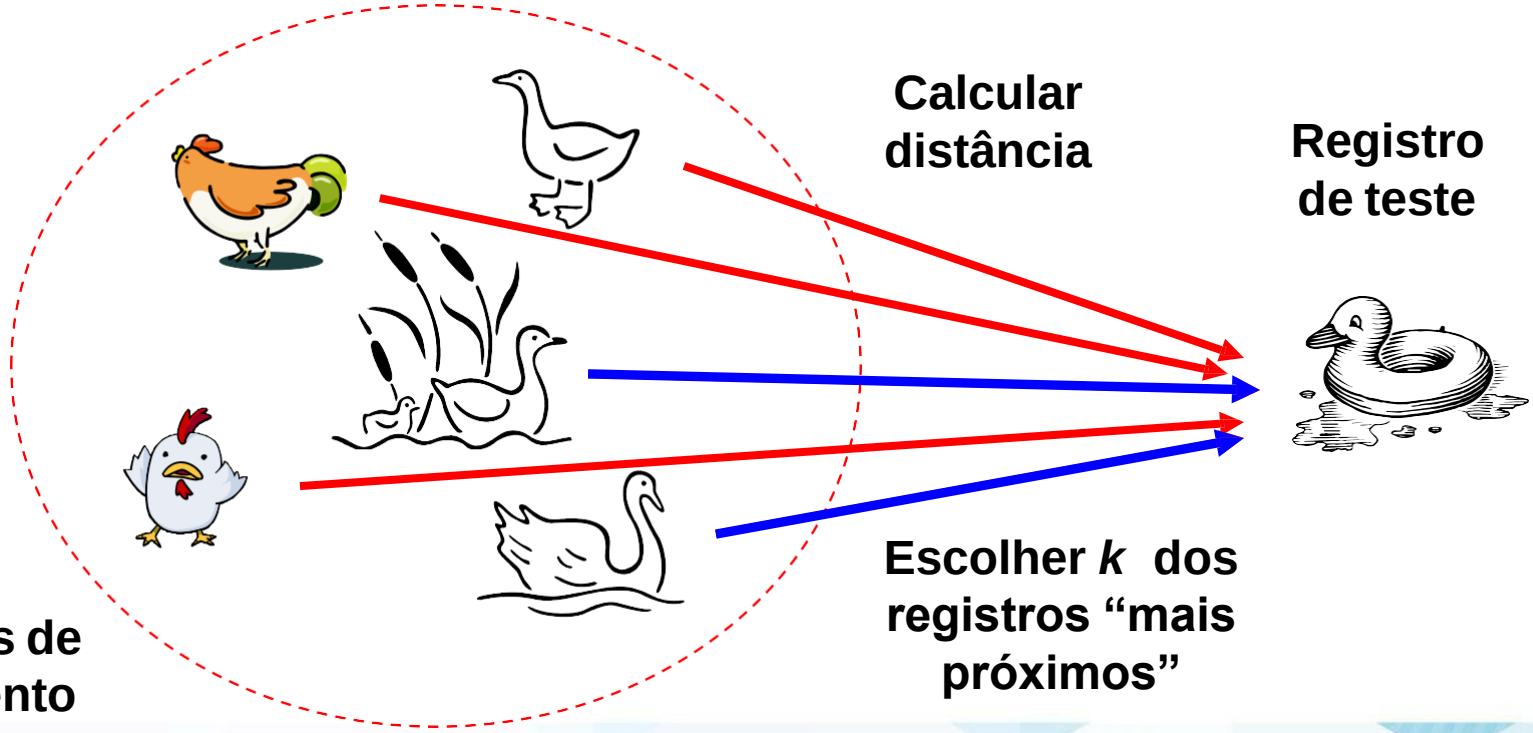
Atr1	……	AtrN

Classificadores baseados em Instância

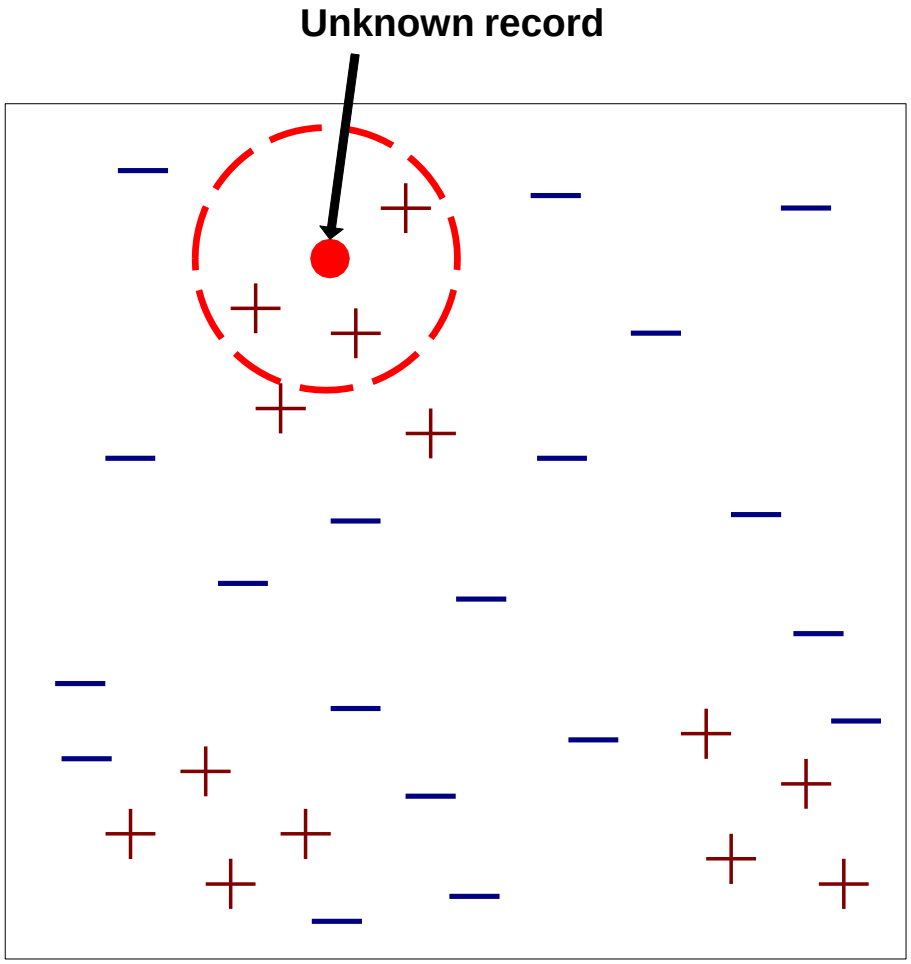
- Exemplos:
  - Rote-learner
    - Memoriza todos os dados de treinamento e realiza classificação somente se atributos do registro correspondem exatamente a um dos exemplos de treinamento
  - Vizinho mais próximo
    - Usa k pontos “mais próximos” (vizinhos mais próximos) para realizar classificação

Classificadores Vizinho Mais Próximo

- Idéia Básica:
  - Se ele anda como um pato, grasna como um pato, então é provavelmente um pato

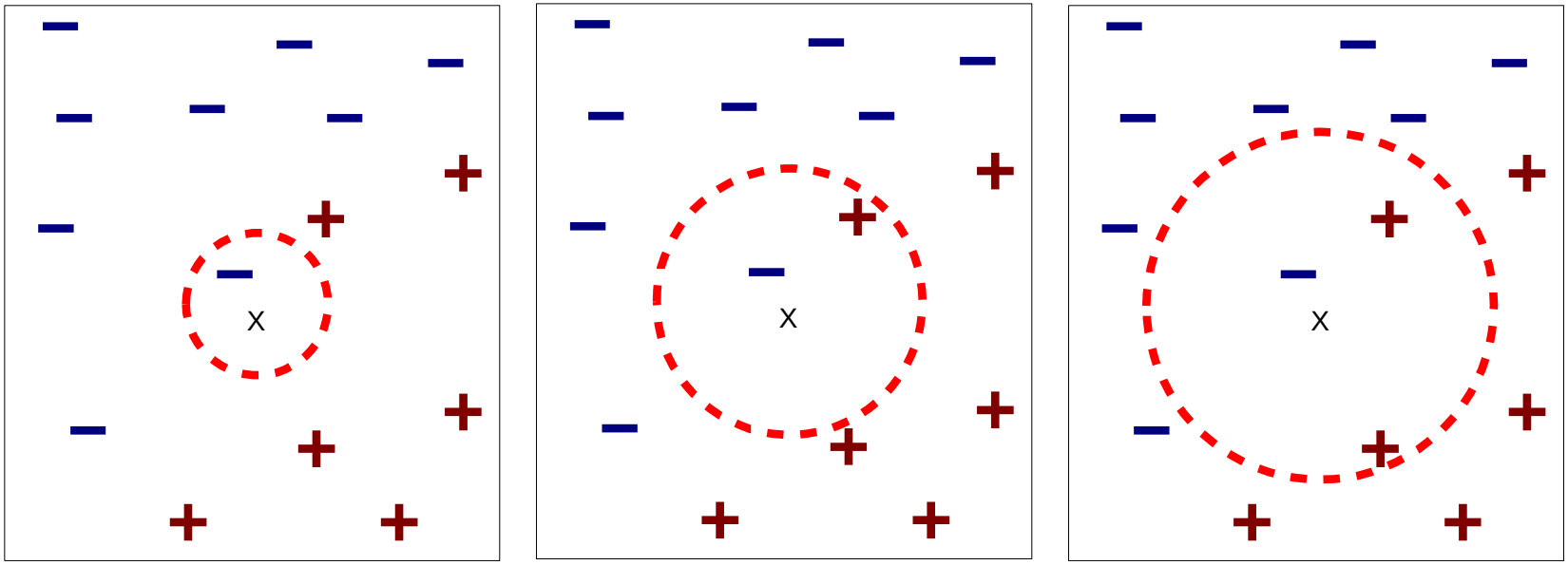


Classificadores Vizinho Mais Próximo



- Requer três coisas
  - Conjunto de registros armazenados
  - Métrica de distância para calcular distância entre registros
  - Valor de *k*, o número de vizinhos mais próximos a recuperar
- Para classificar registro desconhecido:
  - Calcular distância para outros registros de treinamento
  - Identificar *k* vizinhos mais próximos
  - Usar rótulos da classe dos vizinhos mais próximos para determinar o rótulo da classe do registro desconhecido (e.g., pelo voto da maioria)

Definição do Vizinho Mais Próximo

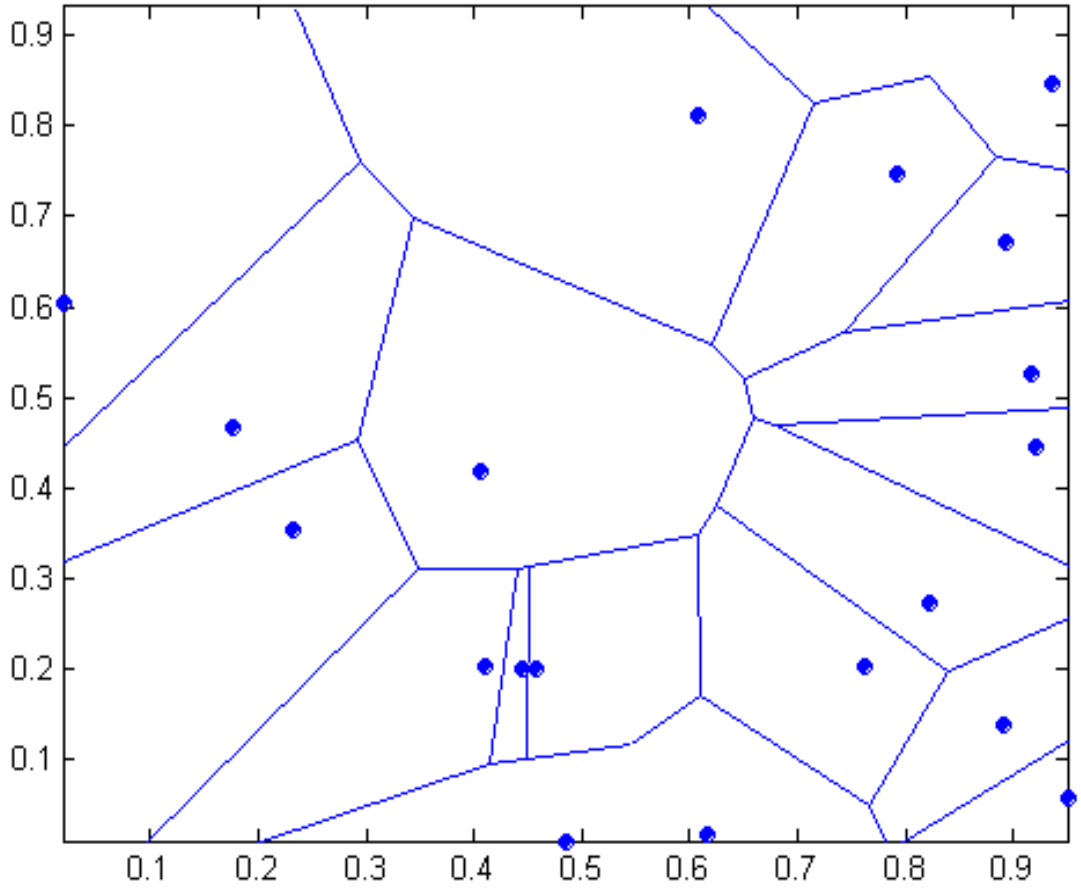


(a) 1-nearest neighbor (b) 2-nearest neighbor (c) 3-nearest neighbor

K-vizinhos mais próximos de um registro *x* são pontos de dados que tem a *k*-ésima menor distância para *x*

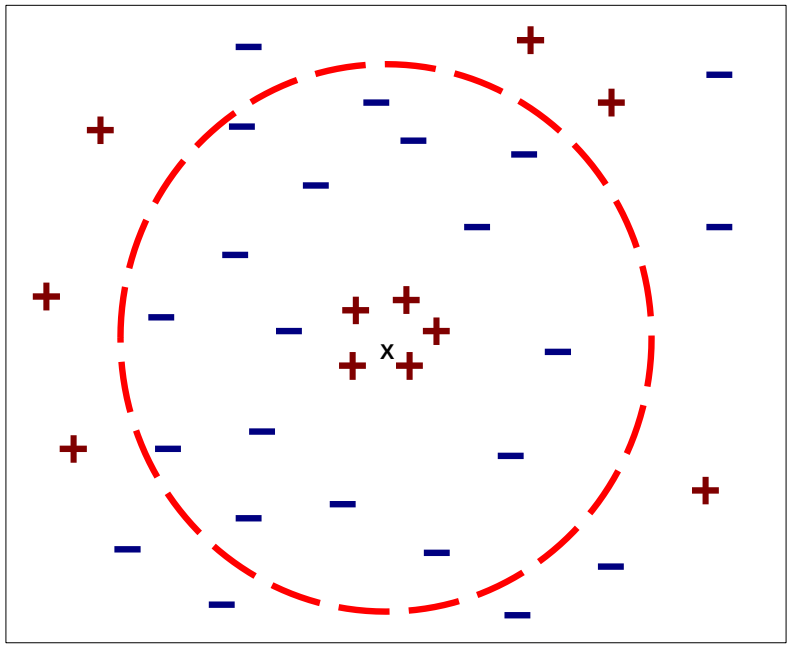
Vizinho Mais Próximo (1-NN)

Diagrama de Voronoi



Vizinho Mais Próximo – 1

- Escolhendo o valor de *k*:
  - Se *k* é muito pequeno, há sensibilidade ao ruído
  - Se *k* é muito grande, a vizinhança pode incluir pontos de outras classes



Vizinho Mais Próximo – 2

- Questões de escalamento
  - Atributos podem ter de ser escalados para evitar medidas de distâncias serem dominadas por um dos atributos
  - Exemplo:
    - altura de uma pessoa pode variar de 1.5m a 1.8m
    - peso de uma pessoa pode variar de 45kg a 150kg
    - renda de uma pessoa pode variar de $1k a $100k

Vizinho Mais Próximo – 3

- Classificador *k*-NN é lazy learner (“aprendizagem preguiçosa”)
  - Não constrói explicitamente modelos
  - Diferente da “aprendizagem ávida” (“eager learner”) como árvore de decisão e sistemas baseados em regras
  - Classificação de registros desconhecidos é relativamente cara computacionalmente

MÉTODOS BASEADOS EM TÉCNICAS ESTATÍSTICAS

Classificador Bayesiano

- Quadro probabilístico para resolução de problemas de classificação
- Probabilidade Condicional:

$$P(C \mid A) = \frac{P(A,C)}{P(A)}$$

$$P(A \mid C) = \frac{P(A,C)}{P(C)}$$

- Teorema de Bayes:

$$P(C \mid A) = \frac{P(A \mid C)P(C)}{P(A)}$$

Exemplo de Teorema de Bayes

- Dados:
  - Um médico sabe que meningite causa enrijecimento do pescoço em 50% dos casos
  - Probabilidade a priori de paciente ter meningite é 1/50.000
  - Probabilidade a priori de paciente ter pescoço rijo é 1/20

- Se um paciente tem pescoço rijo, qual é a probabilidade de ter meningite?

$$P(M \mid S) = \frac{P(S \mid M)P(M)}{P(S)} = \frac{0.5 \times 1/50000}{1/20} = 0.0002$$

Classificador Naïve Bayes

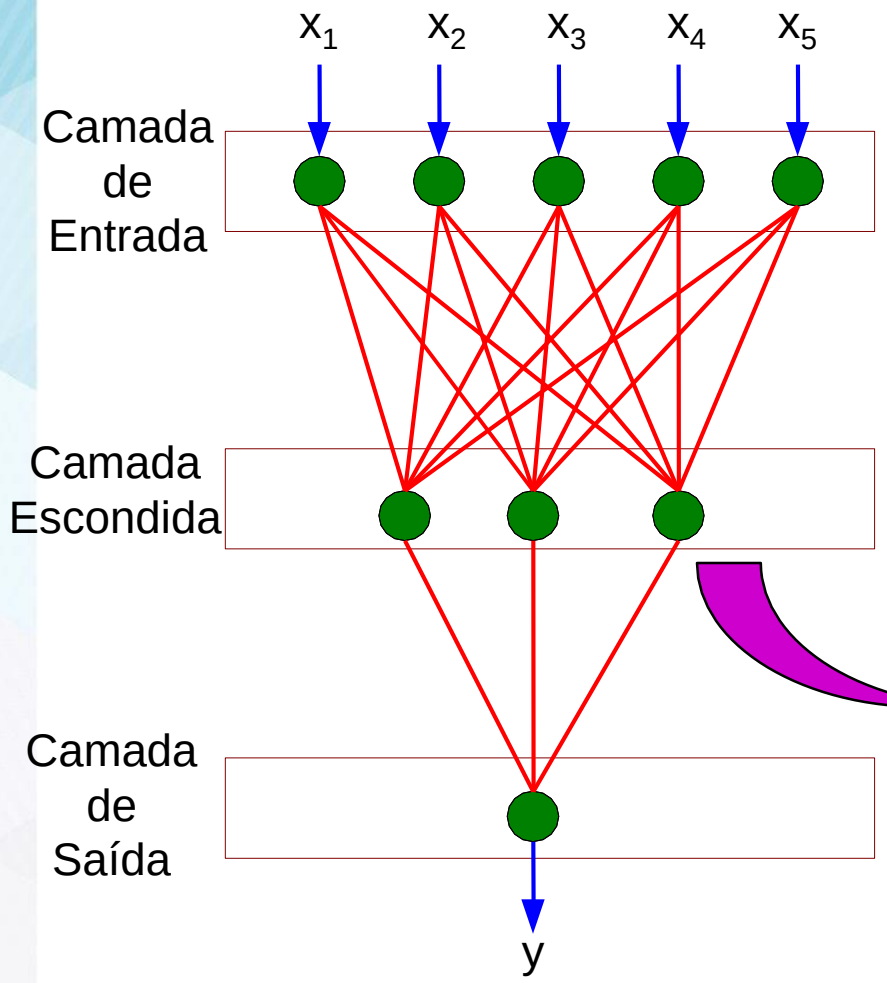
- Assume independência entre atributos A_i quando a classe é dada:
  - $P(A_1, A_2, \ldots, A_n \mid C) = P(A_1 \mid C_j)\, P(A_2 \mid C_j) \ldots P(A_n \mid C_j)$
  - Pode estimar $P(A_i \mid C_j)$ para todo A_i e C_j.
  - Novo ponto é classificado como C_j se $P(C_j) \prod P(A_i \mid C_j)$ é máximo.

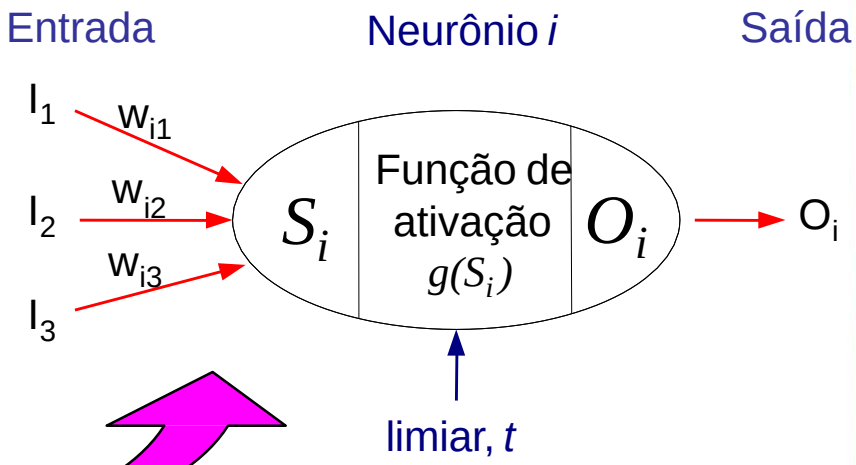
Naïve Bayes (Sumário)

- Robusto a pontos isolados de ruído
- Suporta valores faltantes ignorando a instância durante cálculos da estimativa de probabilidade
- Robusto a atributos irrelevantes
- Pressuposição de Independência pode não se sustentar para alguns atributos
  - Usar outros técnicas tais como Bayesian Belief Networks (BBN)

MÉTODOS BASEADOS EM REDES NEURAIS ARTIFICIAIS (E “DEEP LEARNING”)

Estrutura Geral de uma RNA

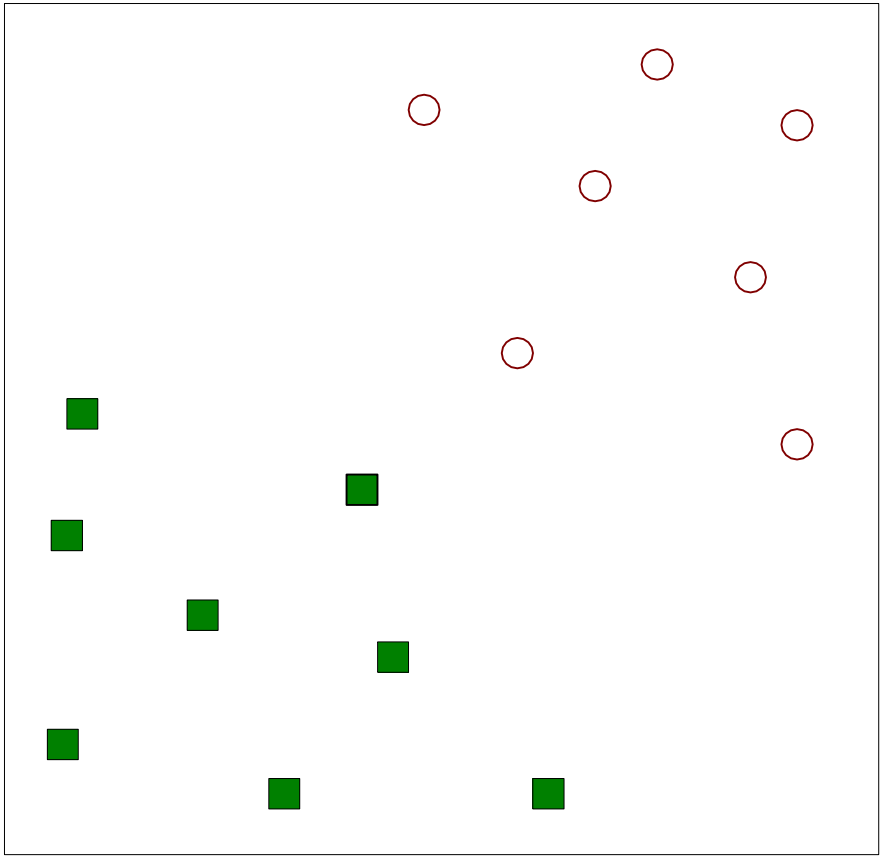




Treinar uma RNA significa aprender os pesos dos neurônios

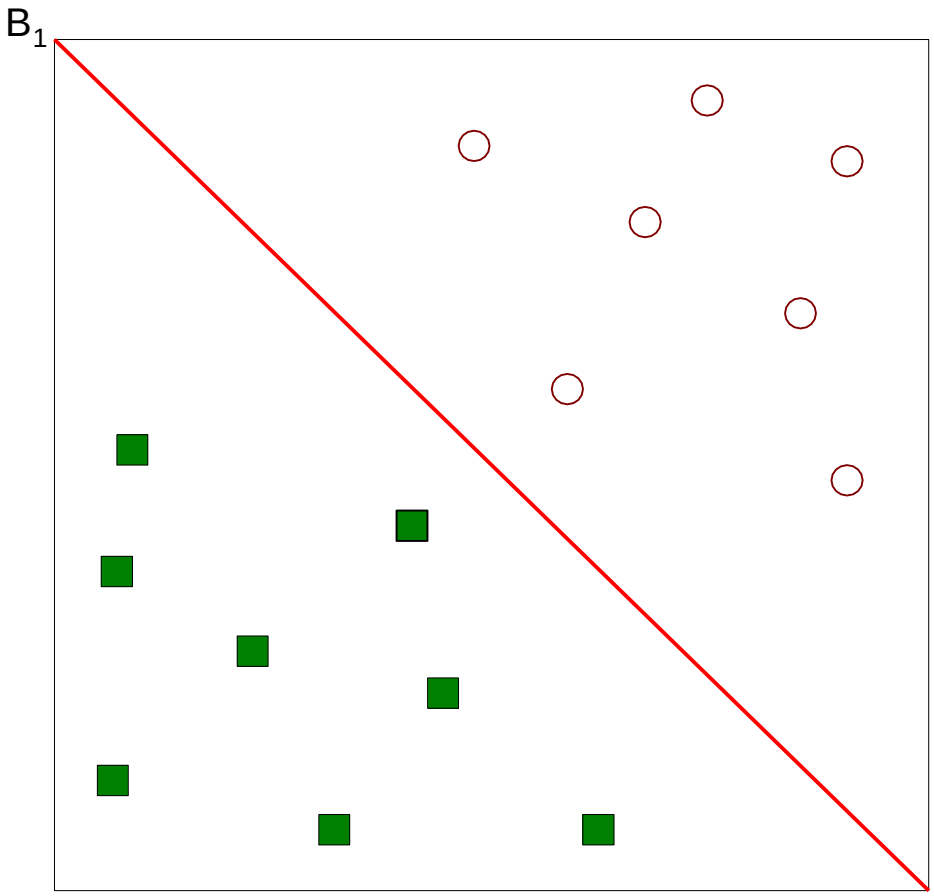
MÉTODO BASEADOS EM SVM (“SUPPORT VECTOR MACHINES”)

Support Vector Machines – 1



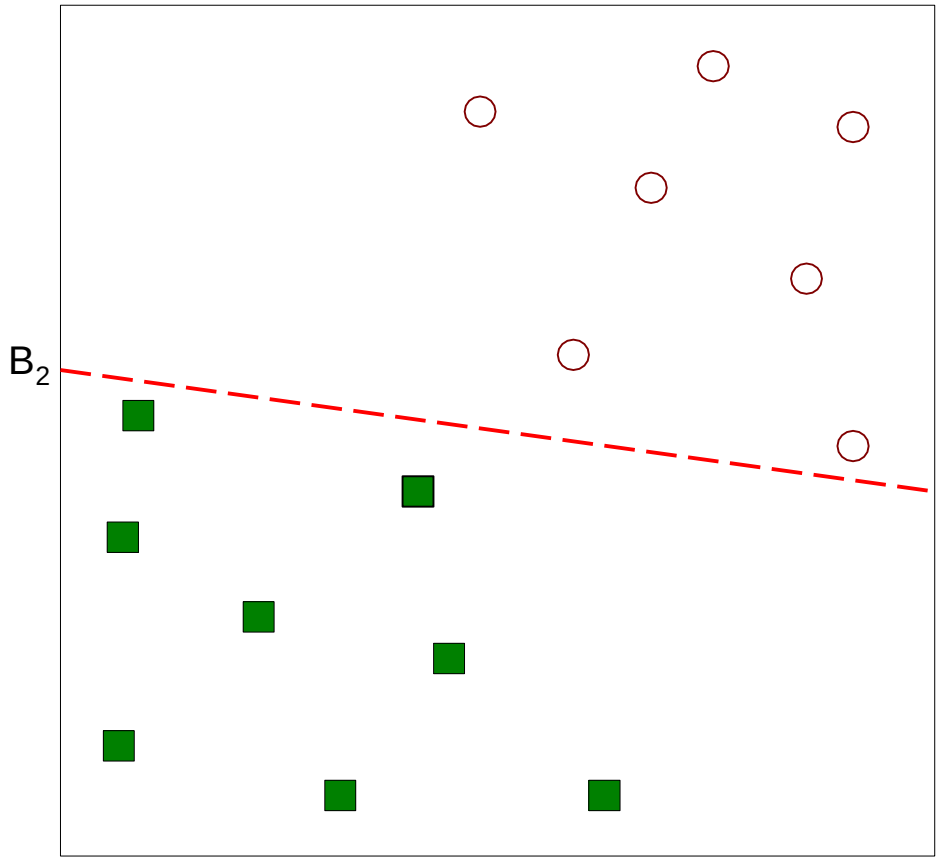
- Encontrar um hiperplano linear (superfície de decisão) que irá separar os dados

Support Vector Machines – 2



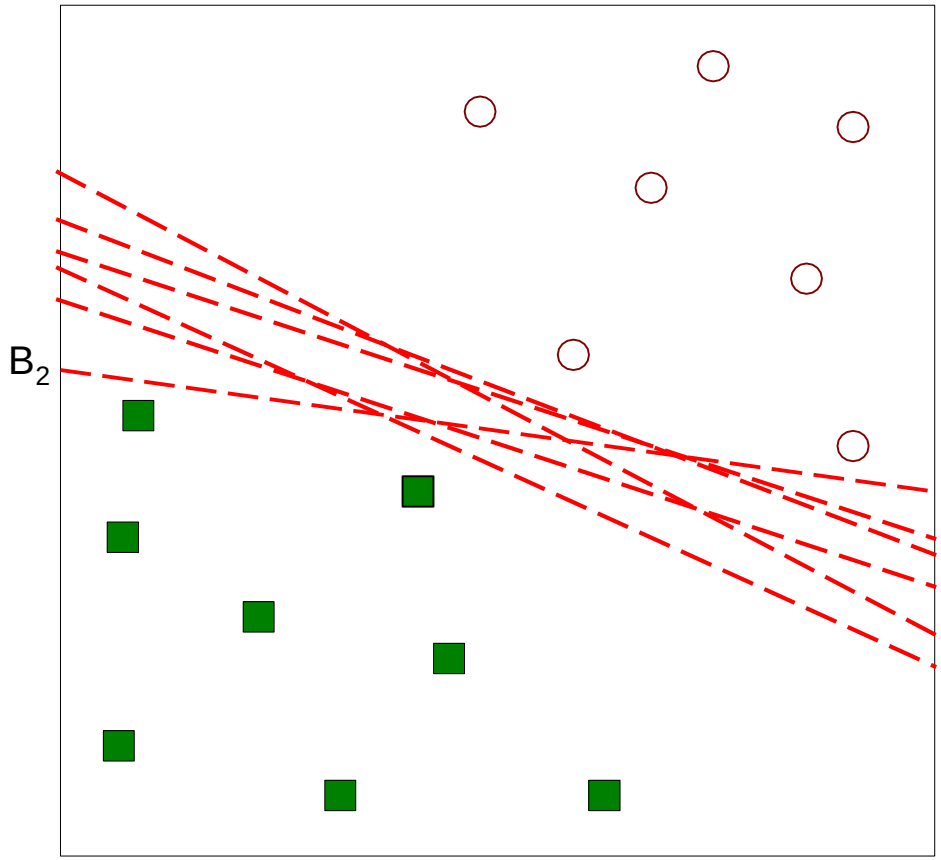
- Uma possível solução

Support Vector Machines – 3



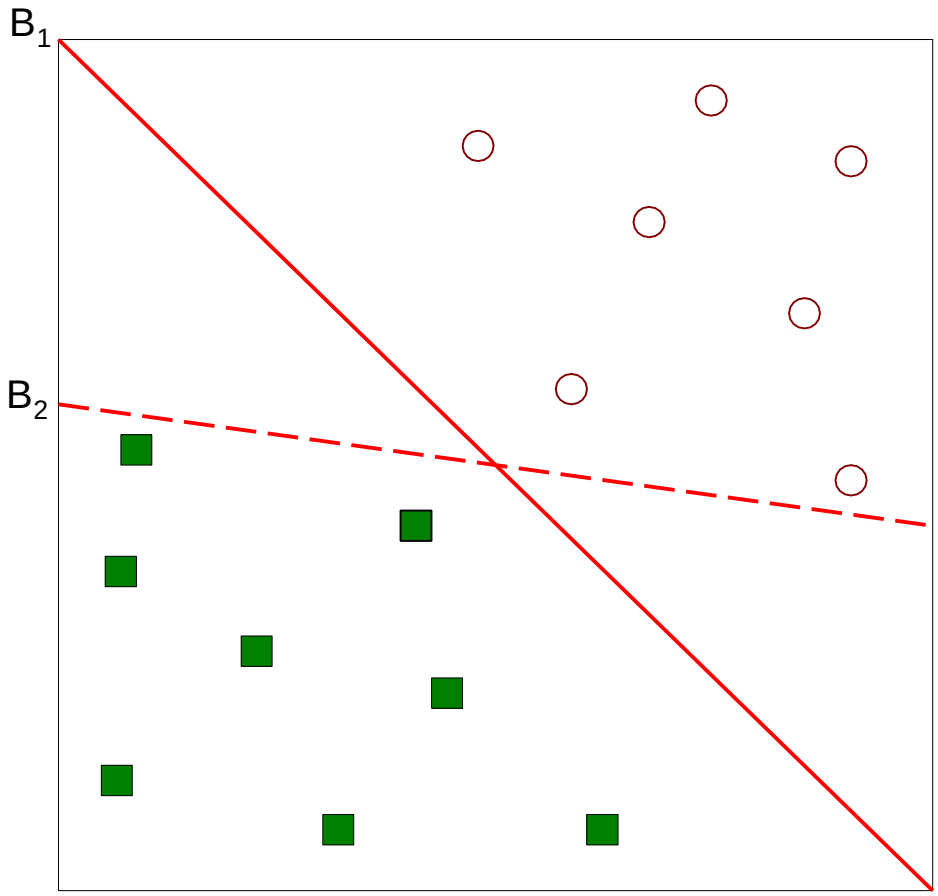
- Outra solução possível

Support Vector Machines – 4



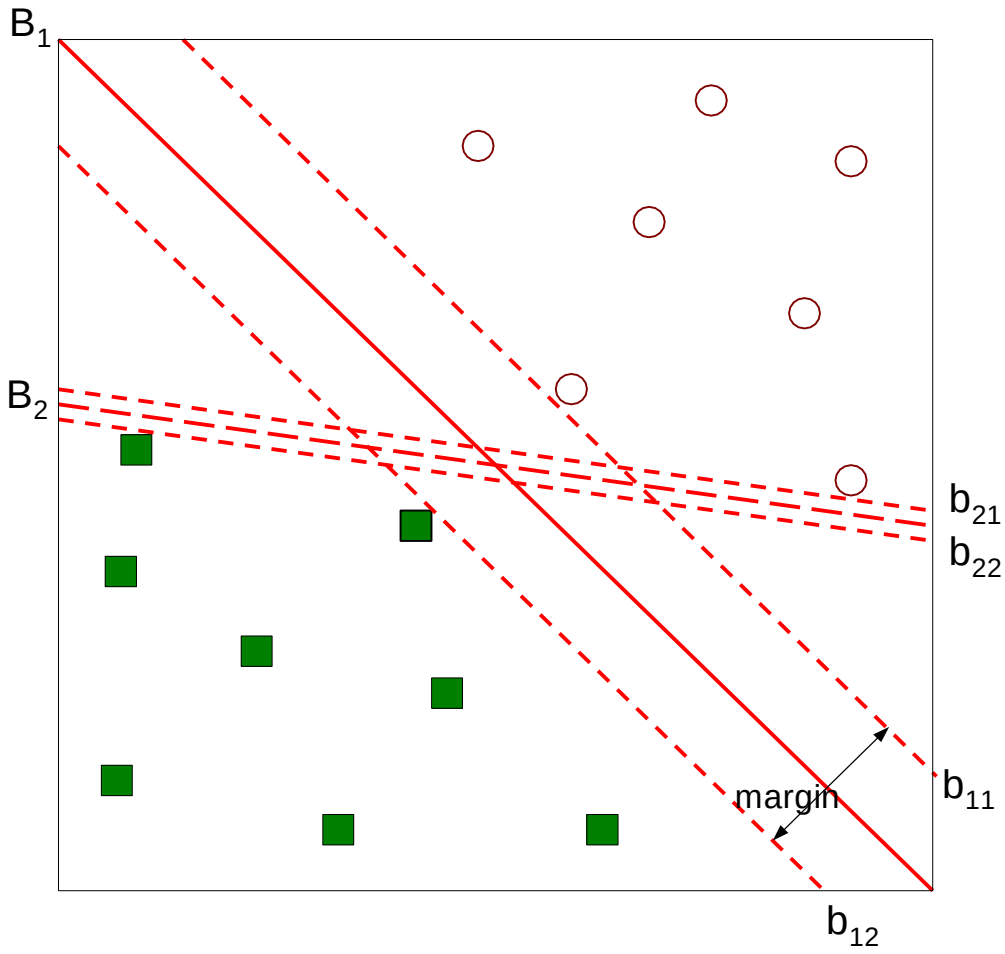
- Outras soluções possíveis

Support Vector Machines – 5



- Qual é melhor? B1 ou B2?
- Como se define melhor?

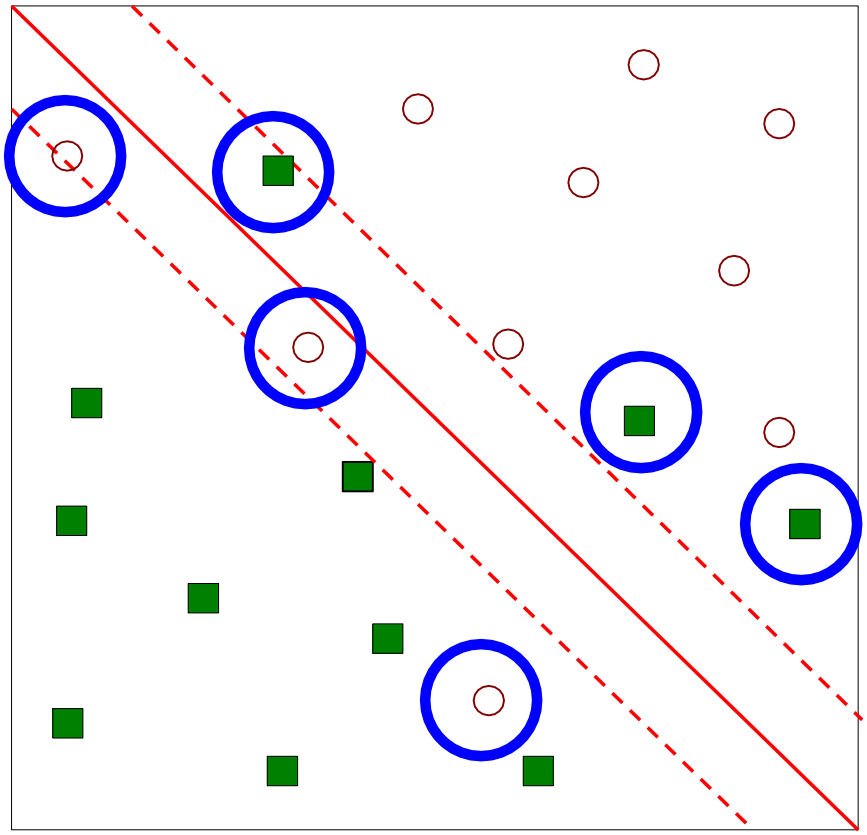
Support Vector Machines – 6



- Achar hiperplano que maximiza a margem => B_1 é melhor que B_2

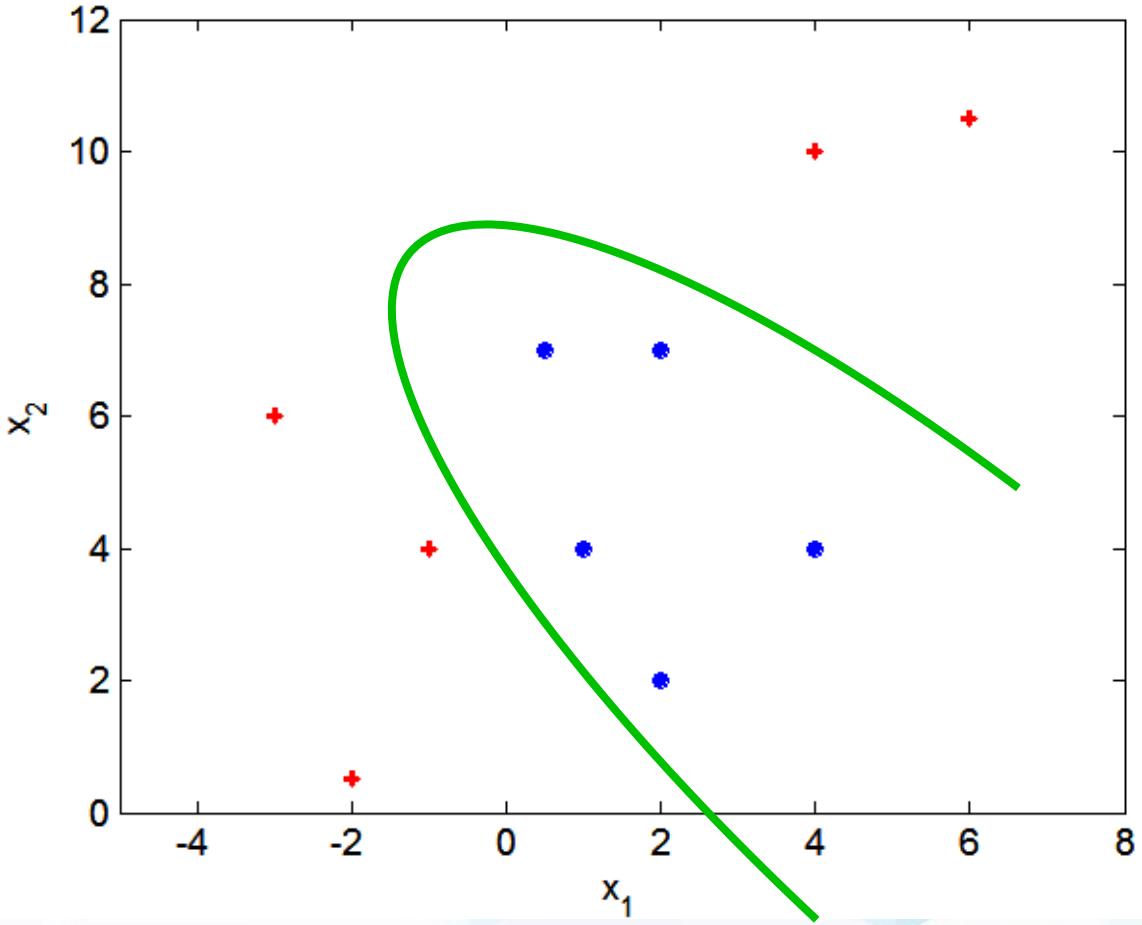
Support Vector Machines – 7

- E se o problema não for linearmente separável?



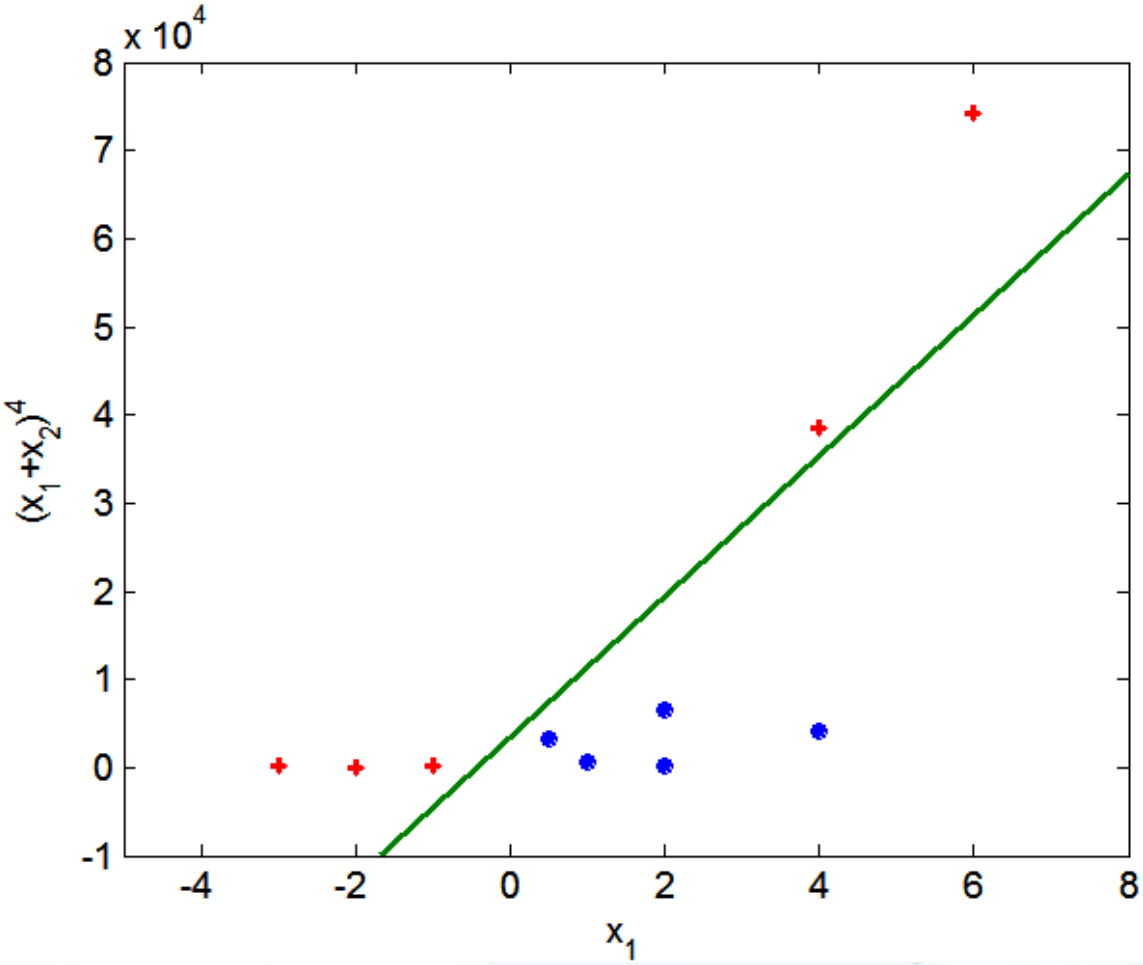
Não-linear Support Vector Machines – 8

- E se a superfície de decisão não for linear?



Não-linear Support Vector Machines – 9

- Transformar dados para espaço de maior dimensão

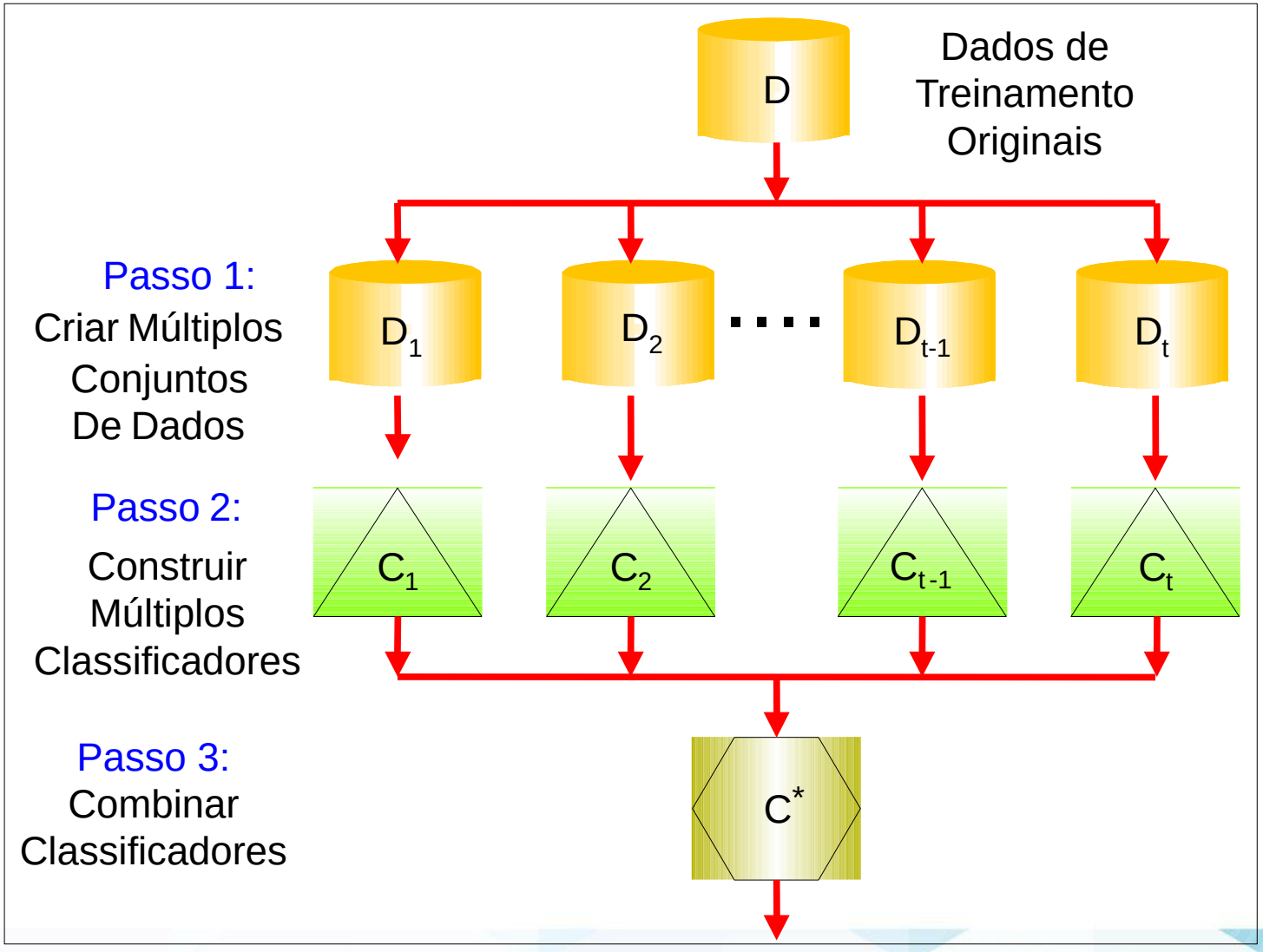


MÉTODOS BASEADOS NA COMBINAÇÃO DE VÁRIOS CLASSIFICADORES

Combinação de Classificadores

- Construir um conjunto de classificadores a partir dos dados de treinamento
- Prever o rótulo da classe de registros previamente desconhecidos através da agregação de previsões feitas por múltiplos classificadores

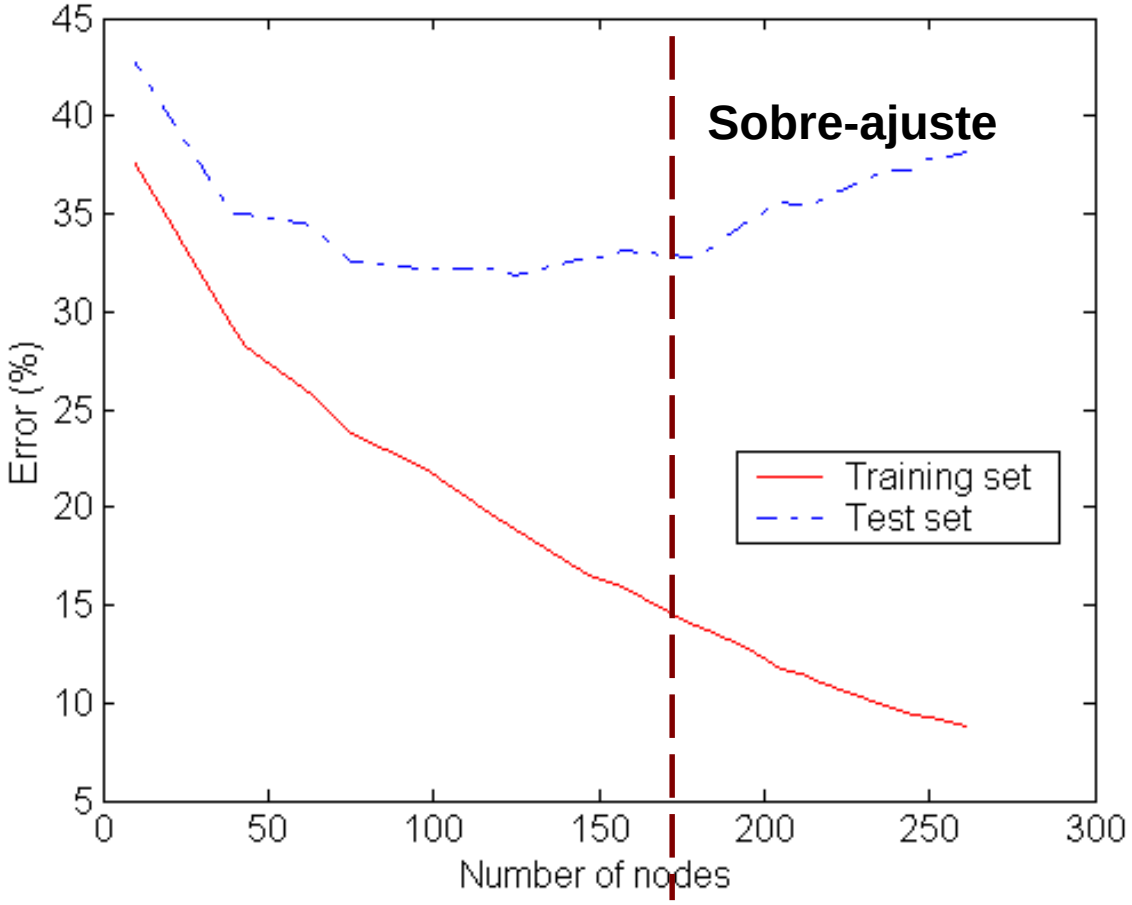
Idéia Geral



Questões Práticas de Classificação

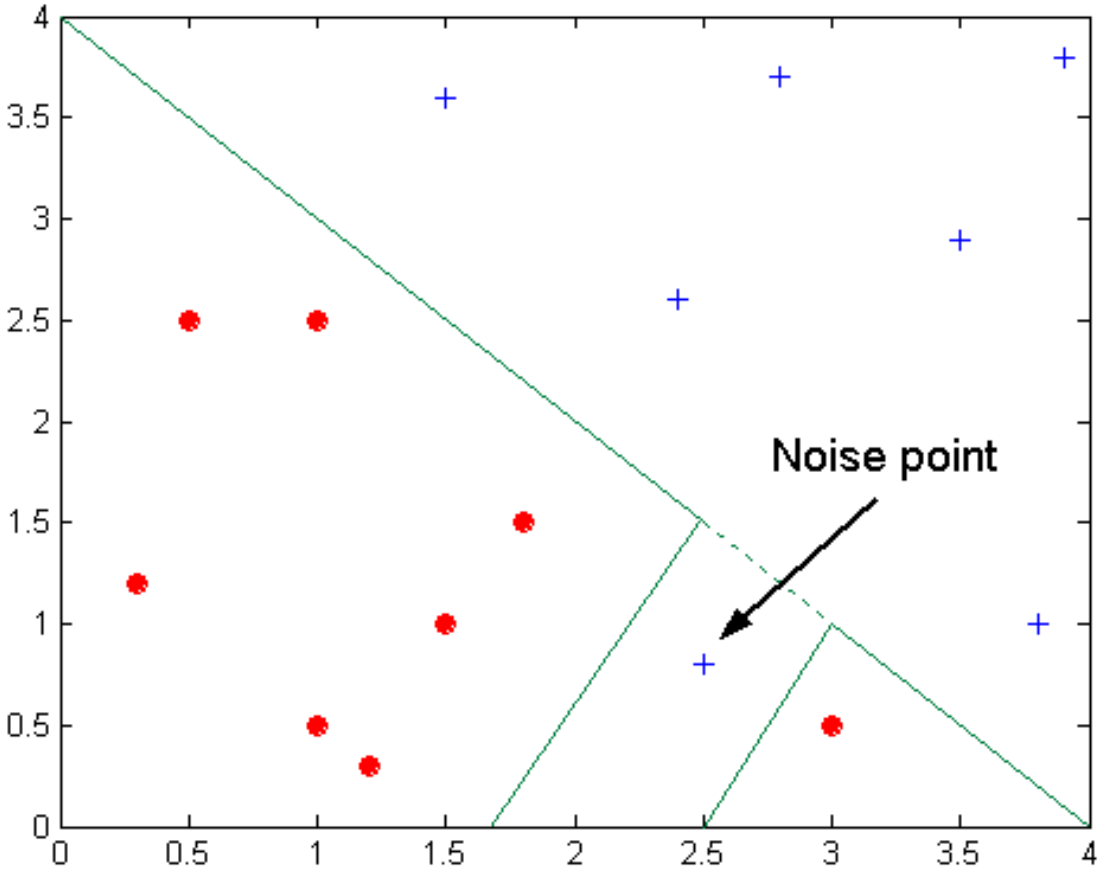
- Sub- e sobre-ajuste (Underfitting e Overfitting)
- Valores Faltantes
- Custos de Classificação

Sub- e Sobre-aiuste



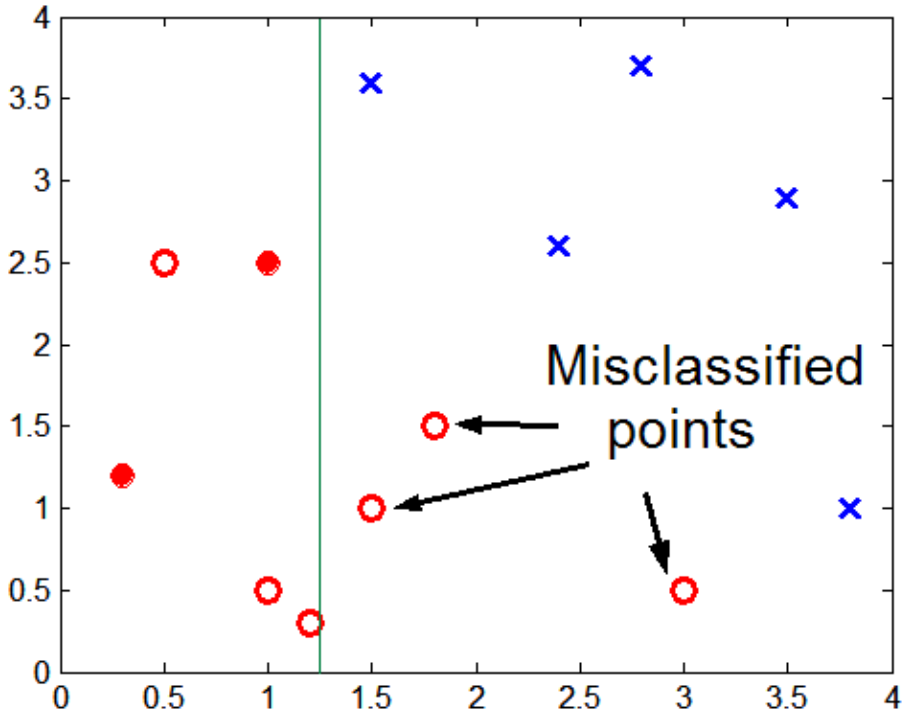
Sub-ajuste: Quando o modelo é simples demais, tanto o erro de treinamento quando o erro de teste são elevados

Sobre-ajuste devido ao Ruído



Fronteira decisória é distorcida pelo ponto de ruído

Sobre-ajuste devido a Exemplos Insuficientes



- Falta de dados na metade inferior do diagrama torna difícil prever corretamente os rótulos da classe naquela região
- Número insuficiente de registros na região faz árvore de decisão prever exemplos de teste usando outros registros de treinamento que são irrelevantes para a tarefa de classificação

Observações – Sobreajuste

- Sobreajuste resulta em modelos que são mais complexos que o necessário
- Erro de treinamento não é mais uma boa estimativa de quão bem a árvore desempenhará em registros previamente desconhecidos
- Necessita de novas formas para estimar erros

Estimando Erro de Generalização

- ~~Erro de Re-substituição~~: erro sobre treinamento (Σ e(t))

- Erro de Generalização: erro sobre teste (Σ e'(t))

Navalha de Occam (Occam's Razor)

- Dados dois modelos com erro de generalização similares, deve-se preferir o modelo mais simples em relação ao modelo mais complexo

- Para modelos complexos, há uma maior chance que tenha sido ajustado acidentalmente pelos erros nos dados

- Portanto, deve-se incluir a complexidade do modelo durante a avaliação do modelo

Avaliação do Modelo

- Métricas para Avaliação de Desempenho
  - Como avaliar o desempenho de um modelo?
- Métodos para Avaliação de Desempenho
  - Como obter estimativas confiáveis?
- Métodos para Comparação de Modelos
  - Como comparar o desempenho relativo entre vários modelos?

Métricas para Avaliação de Desempenho – 1

- Foco na capacidade preditiva de um modelo
  - Em lugar de quão rápido ele classifica ou constrói modelos, escalabilidade, etc.
- Matriz de Confusão:

		CLASSE PREVISTA	
CLASSE REAL		Classe=Sim	Classe=Não
	Classe=Sim	a	b
	Classe=Não	c	d

a: TP (verdadeiro positivo)

b: FN (falso negativo)

c: FP (falso positivo)

d: TN (verdadeiro negativo)

Métricas para Avaliação de Desempenho - 2

	CLASSE PREVISTA		
CLASSE REAL		Classe=Sim	Classe=Não
	Classe=Sim	a (TP)	b (FN)
	Classe=Não	c (FP)	d (TN)

- Métrica mais comum de ser utilizada:

$$Precisão = \frac{a + d}{a + b + c + d} = \frac{TP + TN}{TP + TN + FP + FN}$$

Limitação da Precisão

- Considere um problema com 2 classes
  - Número de exemplos da Classe 0 = 9990
  - Número de exemplos da Classe 1 = 10

- Se o modelo prevê tudo como sendo classe 0, precisão é 9990/10000 = 99.9 %
  - Precisão pode enganar porque o modelo não detecta qualquer exemplo da classe 1

Medidas Sensíveis ao Custo

$$\text{Precision (p)} = \frac{a}{a+c}$$

$$\text{Recall (r)} = \frac{a}{a+b}$$

$$\text{F - measure (F)} = \frac{2rp}{r+p} = \frac{2a}{2a+b+c}$$

- Precision tem tendência para C(Sim|Sim) & C(Sim|Não)
- Recall tem tendência para C(Sim|Sim) & C(Não|Sim)
- F-measure tem tendência para todos exceto C(Não|Não)

$$\text{Precisão Ponderada} = \frac{w_1 a + w_4 d}{w_1 a + w_2 b + w_3 c + w_4 d}$$

Métodos de Estimação

- Holdout
  - Reserva 2/3 para treinamento e 1/3 para teste
- Sub-amostragem aleatória
  - Holdout repetido
- Validação Cruzada
  - Particionar dados em k subconjuntos disjuntos
  - k-fold: treinar em *k-1* partições, testar no restante
  - Leave-one-out: $k = n$
- Amostragem Estratificada
  - Sobre-amostragem versus sub-amostragem
- Bootstrap
  - Amostragem com reposição

ESCOLA
POLITÉCNICA