Metodologias e Técnicas de Mineração de Dados - Classificação

Prof. Leandro Almeida

lma3@cin.ufpe.br

Classificação

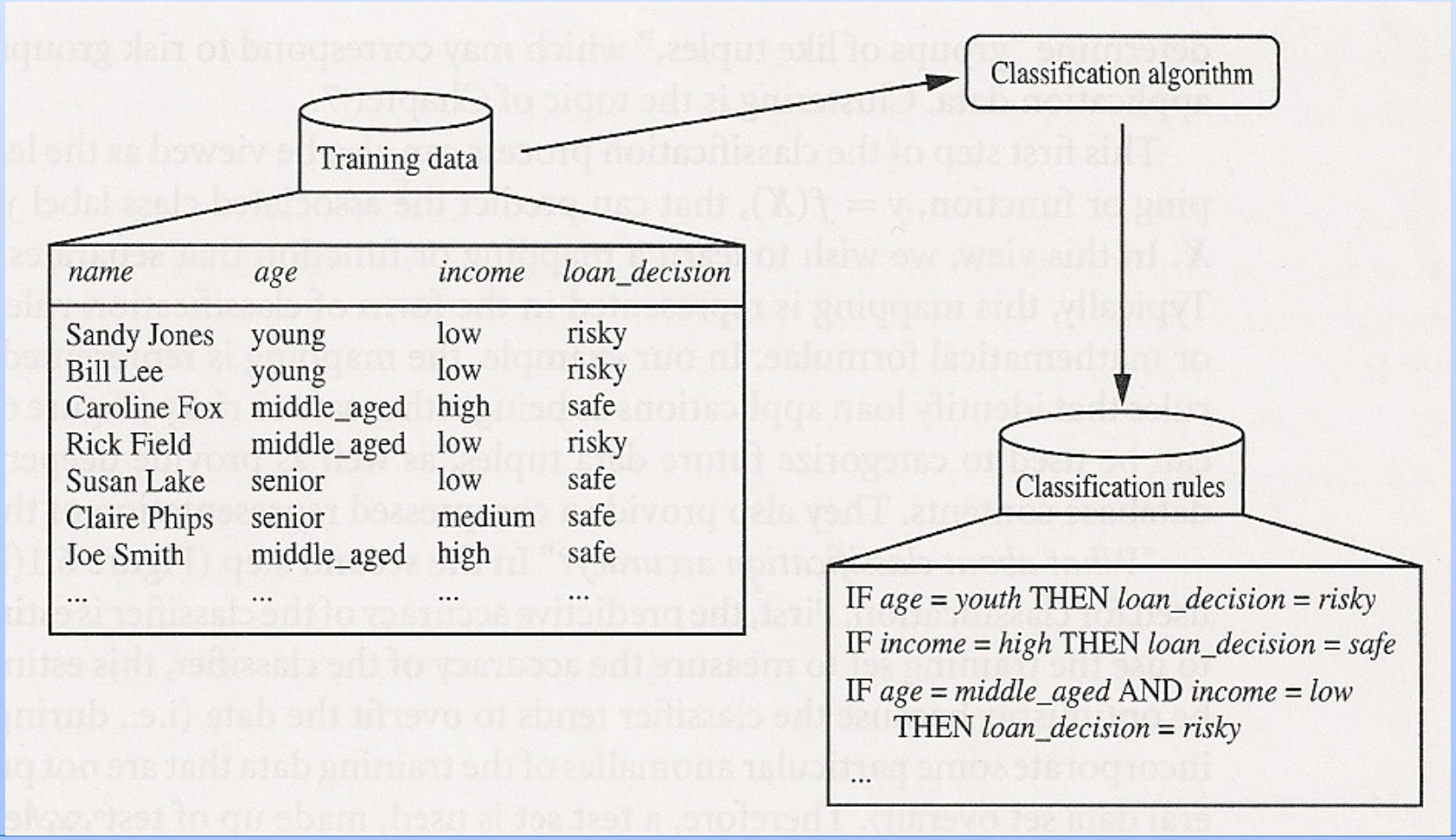
- As bases de dados são ricas fontes de informação "escondida" que podem ser usadas para a tomada de decisões inteligentes.

- A Classificação é uma das formas de análise de dados usadas para:
  - extrair modelos descrevendo classes.
  - Classifica dados (pode construir um modelo ou não) baseado em um conjunto de treinamento previamente rotulado e usa o modelo para classificar novas observações.

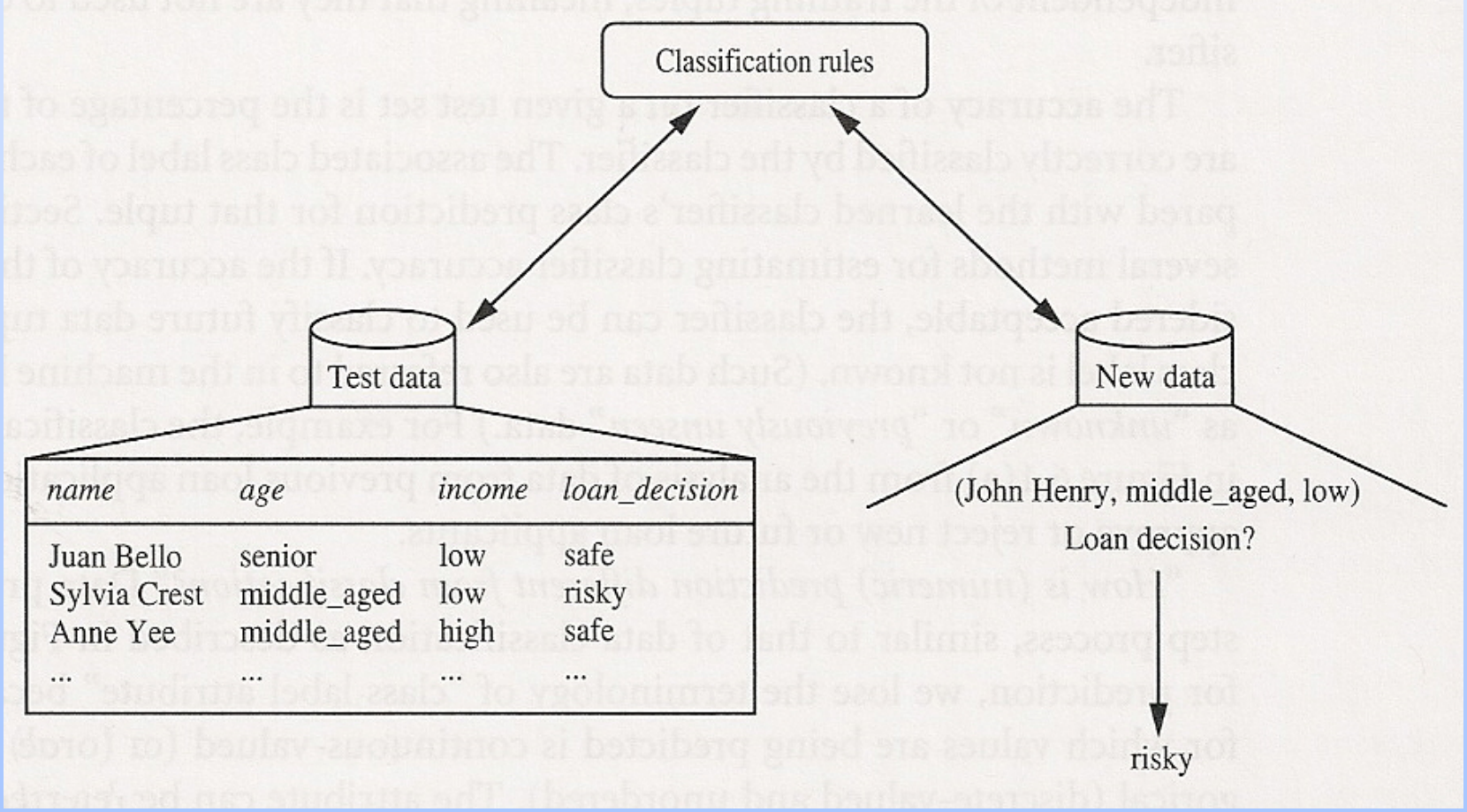
Classificação

- Um processo em duas etapas:
  - **Treinamento** (construção do modelo): descrição de um conjunto de classes *a priori*.
    - cada observação é oriunda de uma classe predefinida.
    - conjunto de observações usadas para construir o modelo.
      - Cada observação X contém um conjunto de atributos (características – x1,x2,x3,...,xn)
  - **Teste** (uso do modelo): classificação de observações desconhecidas.
    - observação é comparada com o resultado do modelo.
    - taxa de acerto é a porcentagem de observações do conjunto teste que são corretamente classificadas pelo modelo

Classificação

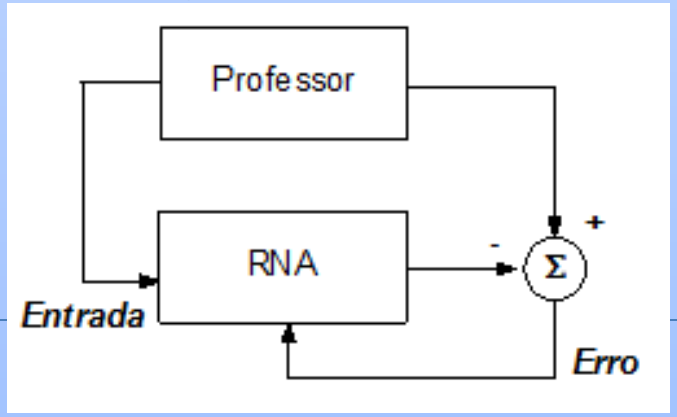


Classificação



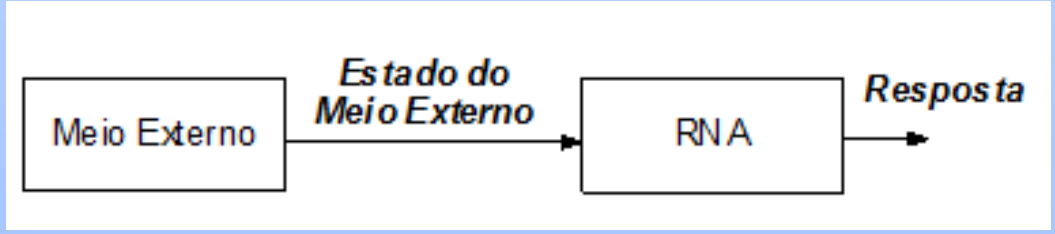
Aprendizagem

- Supervisionado (Classificação)
  - Supervisão: Os dados de treinamento são rotulados pelas classes as quais pertencem
  - Existência de um professor externo
    - Possui conhecimento sobre ambiente, representado por conjunto de pares (x, d)
  - As novas observações são classificadas com base no conjunto de aprendizagem



Aprendizagem

- Quem te ensina a enxergar? Escutar?
- Não-supervisionado (Clusterização)
  - Não tem crítico ou professor externo
  - As classes dos dados de treinamento são desconhecidas
  - O objetivo é formar / descobrir classes à partir dos dados automaticamente
  - Extração de características estatisticamente relevantes



Árvores de Decisão

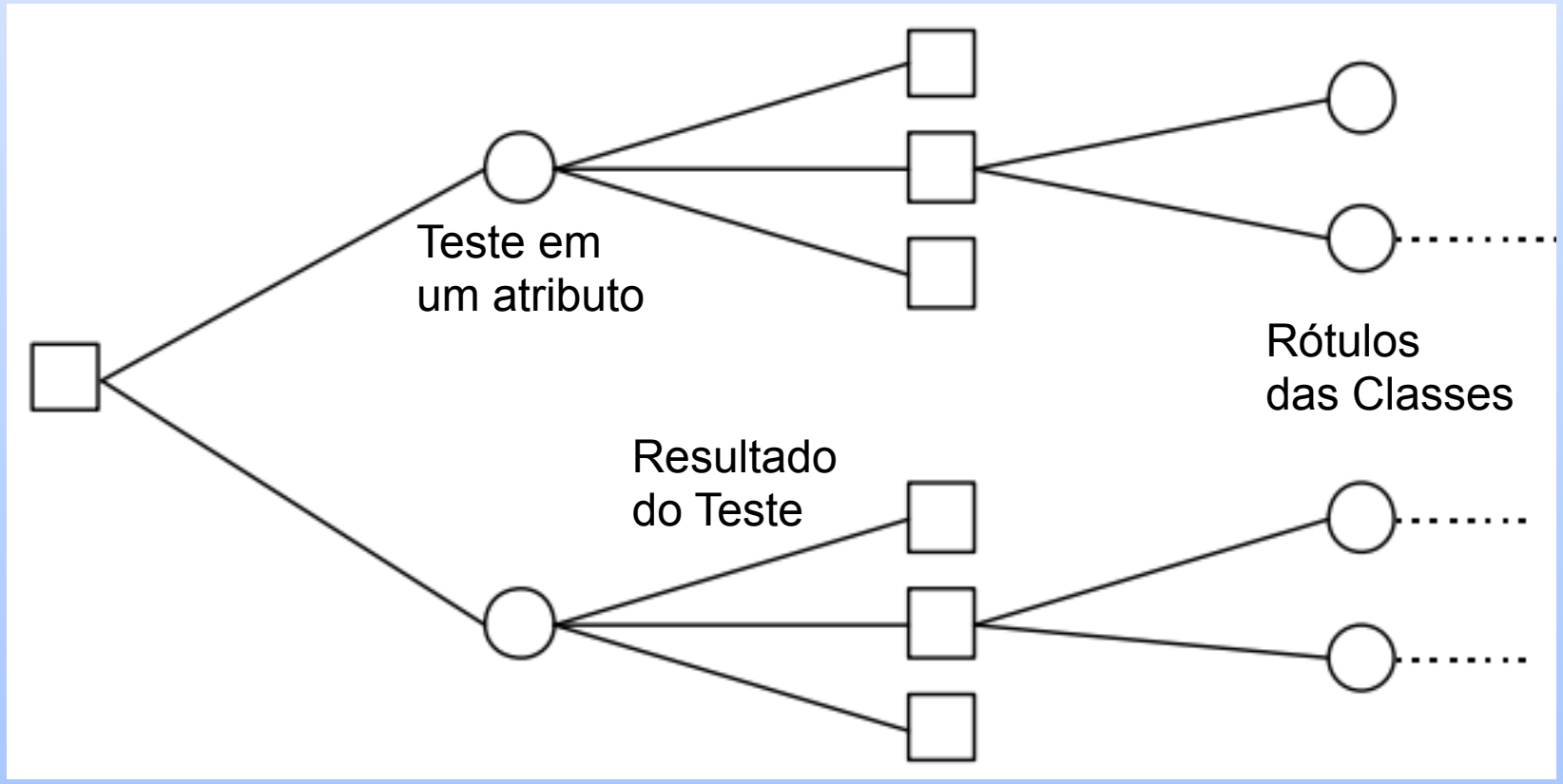
- Conceito:
  - Maneira **gráfica** de visualizar as consequências de decisões **atuais** e **futuras** bem como os eventos **aleatórios** relacionados.
  - Utiliza a estratégia de dividir para conquistar
    - Um problema complexo é decomposto em sub-problemas mais simples
    - Recursivamente, a mesma estratégia é aplicada a cada sub-problema

- A capacidade de discriminação de uma árvore origina da:
  - Divisão do espaço definido pelos atributos em sub-espaços
  - A cada sub-espaço é associada uma classe

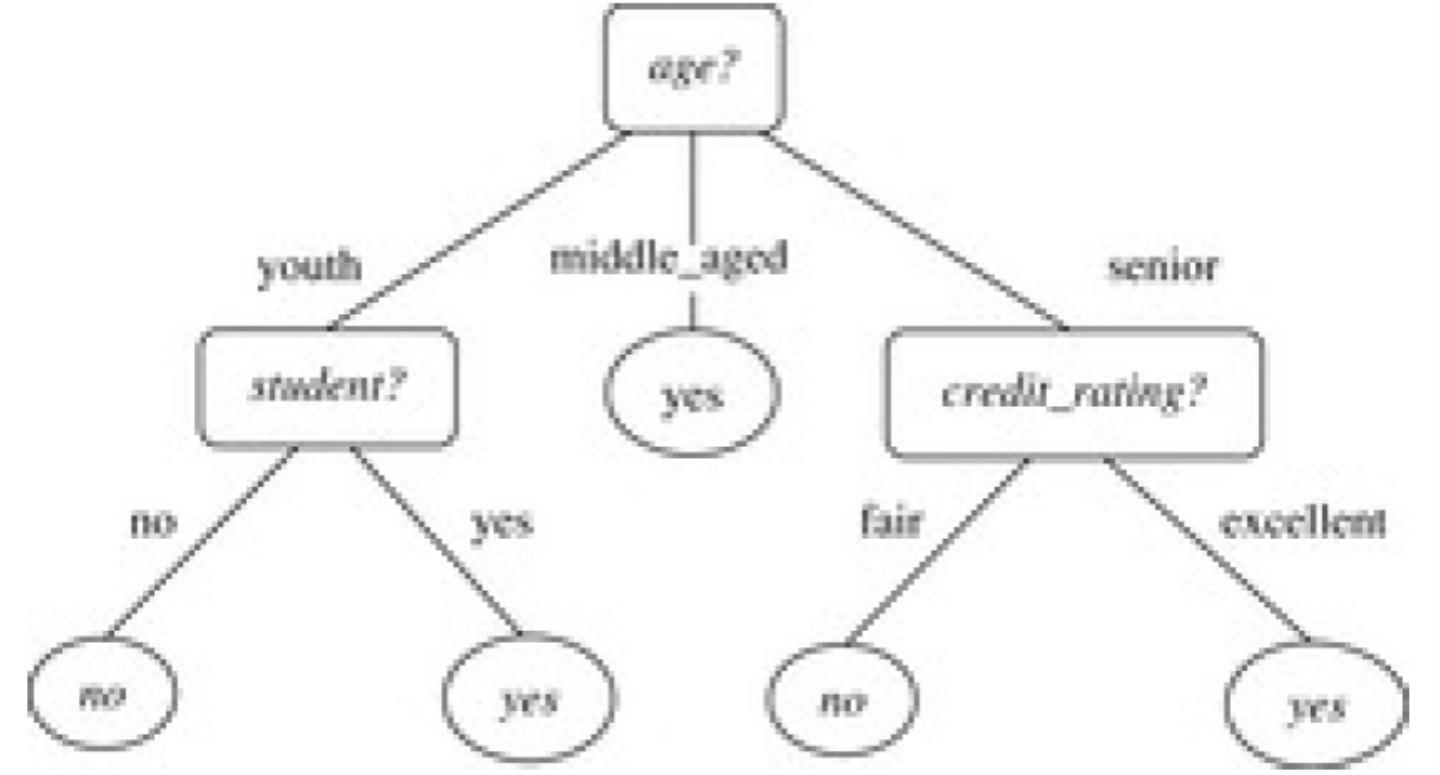
Árvores de Decisão

- Representação:
  - Estrutura semelhante a uma árvore
  - Os nós internos representam um teste em um atributo
  - Os ramos representam o resultado do teste
  - As folhas representam os rótulos das classes

Árvores de Decisão



Exemplo de Árvore de Decisão



Qual o perfil dos compradores de computador?

Árvores de Decisão

- A ideia é...
  - Dada uma tupla, X, que tem seu rótulo associado desconhecido, os valores dos atributos são testados na árvore de decisão.
  - É traçado um caminho a partir do nó raiz até um nó folha, que detém a classificação para X.
  - As árvores de decisão podem ser facilmente convertidas para classificação regras.

- As árvores de decisão são populares porque...
  - Não requerem qualquer conhecimento de domínio ou de ajuste de parâmetros, e, portanto, é apropriada para descoberta de conhecimento exploratório.

Árvores de Decisão

- Algumas árvores de decisão (ID3, C4.5 e CART) adotam uma abordagem gulosa, ou seja em top-down de forma recursiva dividindo para conquistar.
- A maioria dos algoritmos de indução de árvores de decisão também seguem a abordagem *top-down*, que começa com um conjunto de treinamento de tuplas e seus rótulos de classe associados.
  - O conjunto de treinamento é recursivamente dividido em partes menores com a construção da árvore.

Árvores de Decisão

- Fases:
  - Construção da árvore
    - No início, todos os exemplos de treinamento estão na raiz.
    - Os exemplos são particionados recursivamente com base nos atributos selecionados.
  - Poda da árvore
    - Identificar e remover ramos que refletem ruídos e aberrações.

Árvores de Decisão

- Construção da árvore:

1. Escolher um atributo.
2. Estender a árvore adicionando um ramo para cada valor do atributo.
3. Passar os exemplos para as folhas (tendo em conta o valor do atributo escolhido).
4. Para cara folha
    1. Se todos os exemplos são da mesma classe, associar essa classe a folha.
    2. Se não, repetir os passos 1 a 4.

Árvores de Decisão

- Exemplo:
  - Atributos binários: Porta *And*

$A \wedge B$		
0	0	0
0	1	0
1	0	0
1	1	1

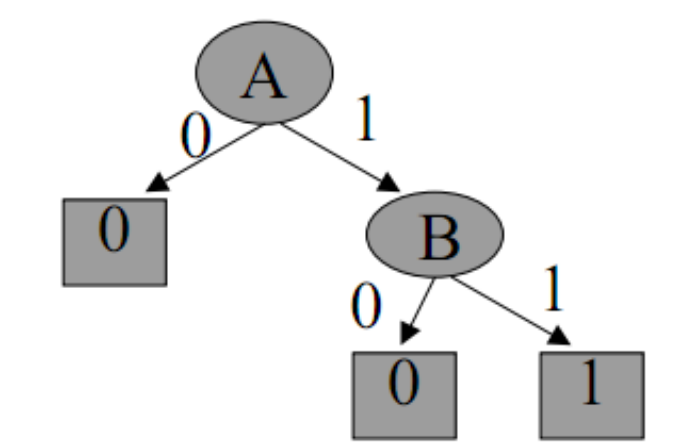


Tabela verdade da função OR

Entradas		Saída
A	B	S
0	0	0
0	1	1
1	0	1
1	1	1

Tabela verdade da função XOR

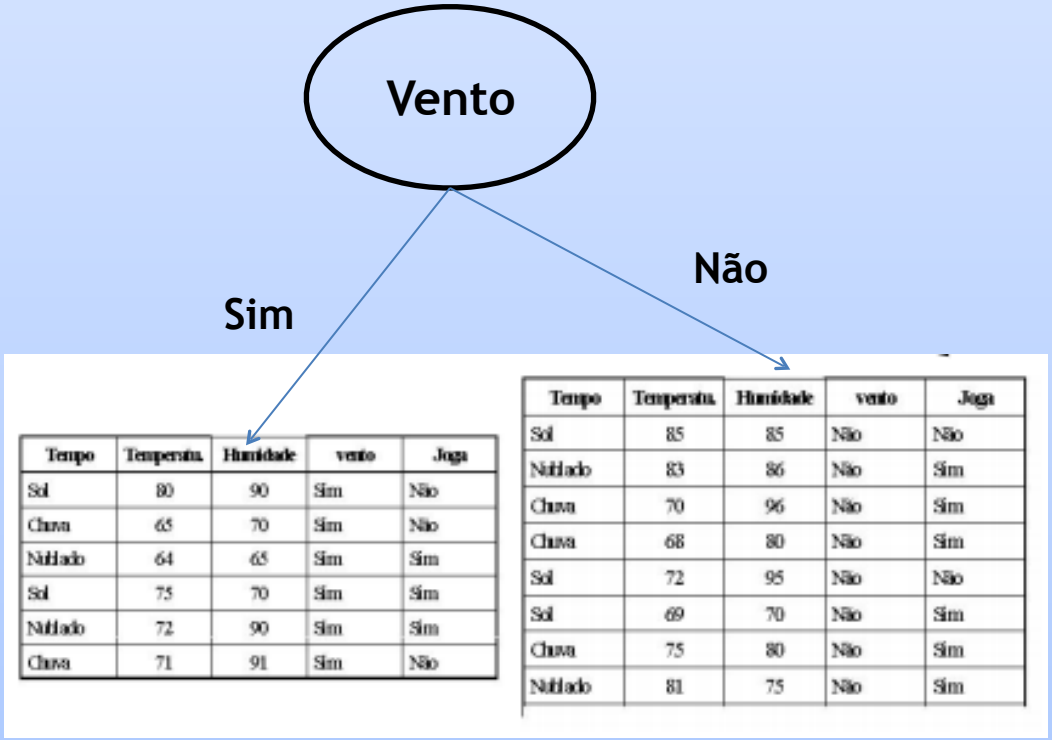
Entradas		Saída
A	B	S
0	0	0
0	1	1
1	0	1
1	1	0

  - Como seria para representar:
    - Or e Xor?

Árvores de Decisão

Conjunto de Dados Original

Tempo	Temperatu.	Humidade	vento	Joga
Sol	85	85	Não	Não
Sol	80	90	Sim	Não
Nublado	83	86	Não	Sim
Chuva	70	96	Não	Sim
Chuva	68	80	Não	Sim
Chuva	65	70	Sim	Não
Nublado	64	65	Sim	Sim
Sol	72	95	Não	Não
Sol	69	70	Não	Sim
Chuva	75	80	Não	Sim
Sol	75	70	Sim	Sim
Nublado	72	90	Sim	Sim
Nublado	81	75	Não	Sim
Chuva	71	91	Sim	Não



Qual o melhor atributo?

Árvores de Decisão

- Como medir a “habilidade” de um dado atributo discriminar as classes?

- Medidas de seleção do atributo:
  - É uma heurística para seleção do melhor critério de separação que melhor separa uma dada partição de dados.
  - Prover um *ranking* para cada atributo. O melhor score é escolhido como critério de separação.
  - Os três métodos mais populares são: ***Information Gain*** (ID3), ***Gain Ratio*** (C4.5) e ***Gini Index*** (CART).

Algoritmo ID3

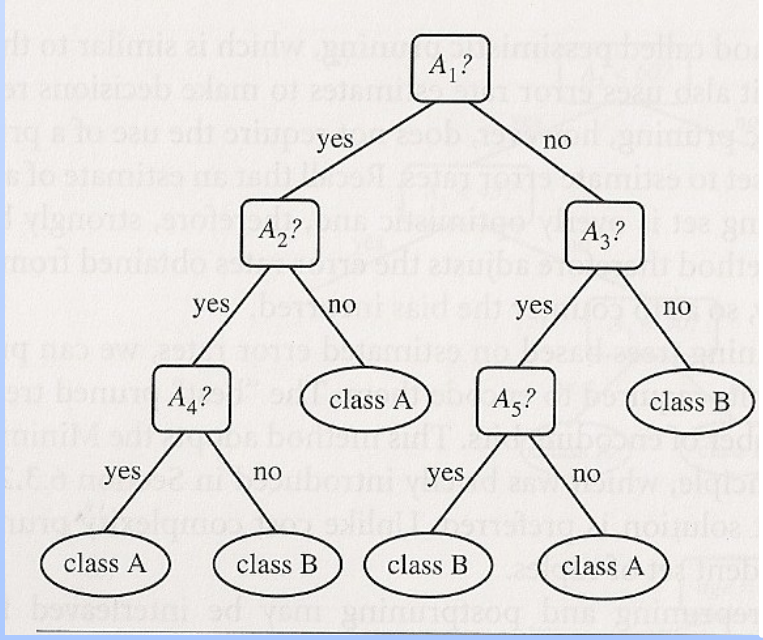
- Algoritmos mais conhecidos ID3 (Quinlan, 1986) e C4.5 (Quinlan, 1993)
- Algoritmo ID3
  - Considere um conjunto de dados para treinamento
  - Ele constrói a árvore em uma abordagem top-down considerando a questão: “Qual atributo é o mais importante e, portanto, deve ser colocado na raiz da árvore?”
  - Para isso cada atributo é testado e sua capacidade para se tornar nó raiz avaliada
  - Cria-se tantos nós filhos da raiz quantos valores possíveis esse atributo puder assumir (caso discreto)
  - Repete-se o processo para cada nó filho da raíz e assim sucessivamente

Árvores de Decisão

- Poda da Árvore
  - Ruídos ou dados aberrantes podem refletir em anomalias na construção de uma árvore de decisão.
  - Métodos de poda da árvore evitam o *overfitting* retirando os ramos menos confiáveis.
  - Medidas de significância podem ser usadas para avaliar a relevância de uma divisão.
  - Se uma partição de ramos num nó resultar numa falha perante um *threshold* especificado esta divisão é interrompida.

Árvores de Decisão

- Poda da Árvore



Árvores de Decisão no Python

- Script disponível em “decision-tree-cross-valid.py”
  - Possível criar uma árvore para uma divisão estratificada dos dados
  - Possível validar árvores sob diferentes divisões estratificadas dos dados
  - Possível converter a árvore em regras de decisão a serem utilizadas em um processo decisório
    - A regras apresentam uma saída no estilo [n-exemplos, classe]
    - random-estate controla a semente para números aleatórios
    - DecisionTreeClassifier implementa uma versão do C4.5 para classificação e regressão de dados

Aprendizado baseado em instâncias

- Ao contrário das outras abordagens, não ocorre a construção de um modelo de classificação explícito.
  - Novos exemplos são classificados com base na comparação direta e similaridade aos exemplos de treinamento.
- Treinamento pode ser fácil, apenas memorizar exemplos.
- Teste pode ser caro pois requer comparação com todos os exemplos de treinamento.
- Também conhecido como:
  - Aprendizado baseado em casos
  - Aprendizado baseado em exemplos
  - Vizinho mais próximo (“Nearest Neighbor” = kNN)
  - Aprendizado baseado em memorização
  - Aprendizado “lazy”

Medidas de Similaridade/Distância

- Métodos baseados em instância precisam de uma função para determinar a similaridade ou distância entre duas instâncias.
- Para atributos contínuos, distância euclidiana é a mais utilizada:

$$d(x_i, x_j) = \sqrt{\sum_{p=1}^{n} (a_p(x_i) - a_p(x_j))^2}$$

onde $a_p(x)$ é o valor do p-ésimo atributo da instância x.

- Para atributos discretos, a distância é 0 se eles tem o mesmo valor e 1 se eles são diferentes.
- Para compensar as diferenças de unidade entre os atributos contínuos, temos que normalizá-los para ficar no intervalo [0,1].

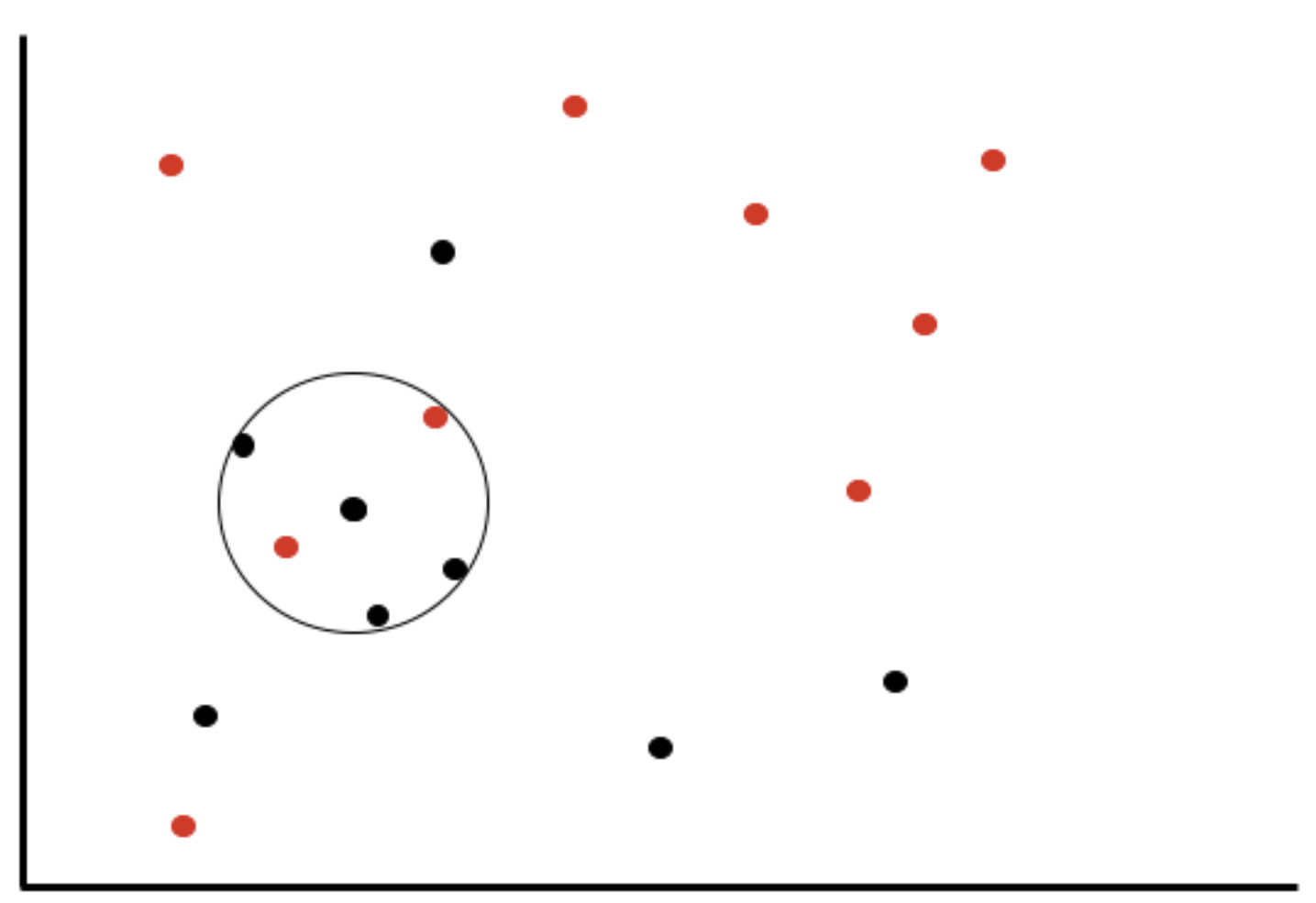
Outras Medidas de Distância

- Distância de Mahalanobis
  - Métrica que faz uma normalização em relação à variância.
- Similaridade de Cosseno
  - Cosseno do ângulo entre os vetores.
  - Usado para classificação de textos e outros dados de alta dimensão.
- Correlação de Pearson
  - Coeficiente de correlação usado em estatística.
  - Muito usado em bioinformática.
- Distância de edição
  - Usado para medir distância entre strings.
  - Usado em classificação de textos e bioinformática.

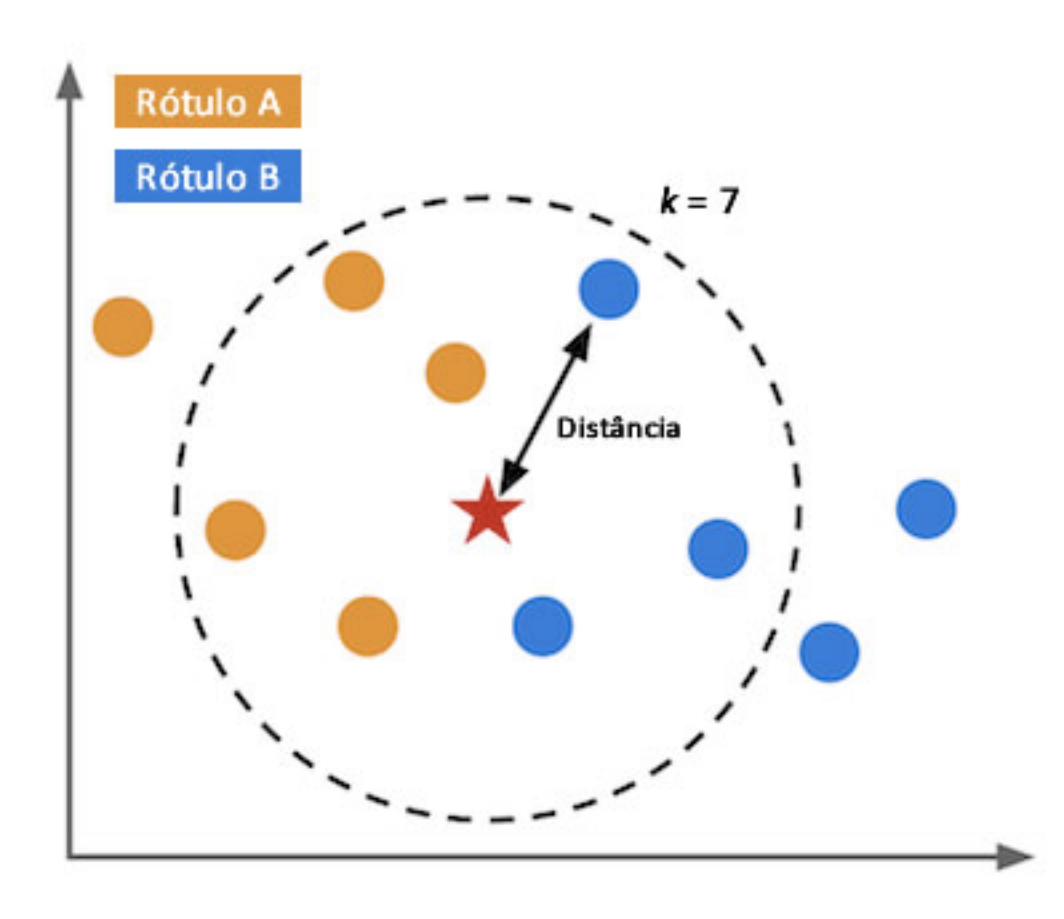
K-Vizinhos Mais Próximos

- Calcular a distância entre o exemplo de teste e cada exemplo de treinamento.
- Escolher os k vizinhos mais próximos e classificar a instância de teste com a classe mais frequente entre os vizinhos mais próximos.
- Usar mais de um vizinho reduz a vulnerabilidade a ruídos.
  - Porém um número muito grande de vizinhos pode incluir “vizinhos” distantes.
  - Melhor valor de k depende dos dados.
  - Normalmente usamos k ímpar pra evitar empates.

K-Vizinhos Mais Próximos (k=5)

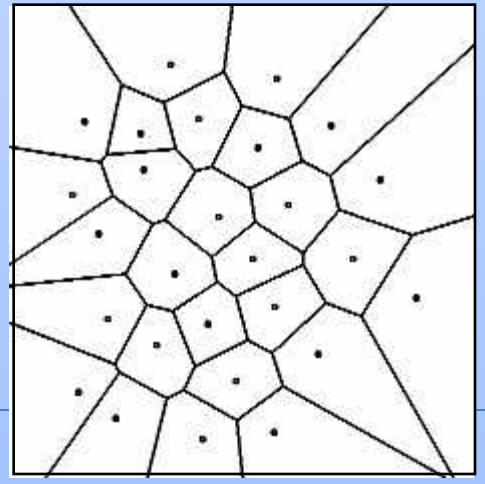


Tarefas de Mineração - Classificação



K-Vizinhos Mais Próximos

- Embora não seja necessário calculá-la, a regra de classificação implicitamente usada é baseada em regiões do espaço de atributos, ao redor de cada exemplo de treinamento.
- Para 1-NN com distância euclidiana, o diagrama de Voronoi mostra como o espaço é dividido:



K-Vizinhos Mais Próximos

- Implementação e inutilização via Python:
  - knn-algorithm.py
  - knn-algorithm2.py