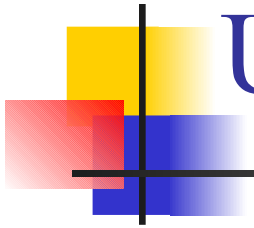
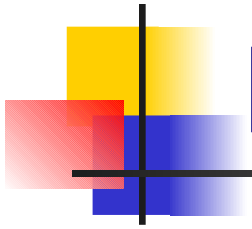


# Universidade Federal do Paraná

Mineração de Dados e
Aprendizado de Máquinas.

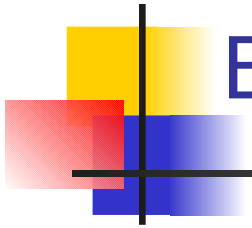
Aurora Trinidad Ramírez Pozo



# Roteiro

- Overview  a Descoberta de Conhecimento em Bases de Dados

# Descoberta de Conhecimento em Bancos de Dados



- um crescimento explosivo nos bancos de dados
- como interpretar e examinar estes dados ???
- necessidade de novas ferramentas e  técnicas para análise automática e inteligente de bancos de dados

# Descoberta de Conhecimento


agreguem valor aos seus negócios

# Posicionamento

# Transformar dados

- em informação e conhecimento
  - úteis para o suporte à decisão,
  - gerenciamento de negócios, controle de produção
  - análise de mercado ao projeto de engenharia e exploração científica

# KDD

- Descoberta de Conhecimento em Bases de Dados
- Knowledge Discovery in Databases
- ferramentas e técnicas empregadas para análise automática e inteligente destes imensos repositórios

# Etapas do Processo

- O processo de KDD é interativo, iterativo, cognitivo e exploratório, envolvendo vários passos
- muitas decisões sendo feitas pelo analista ( especialista do domínio dos dados)

# Conhecimento

1. Definição do tipo de conhecimento a descobrir
   - o que pressupõe uma compreensão do domínio da aplicação
   - bem como do tipo de decisão que tal conhecimento pode contribuir para melhorar.

# Seleção

2. Criação de um conjunto de dados alvo (Selection):
   - selecionar um conjunto de dados, ou focar num subconjunto, onde a descoberta deve ser realizada.

# Limpeza de Dados

3. Pré-processamento: operações básicas

- tais como remoção de ruídos quando necessário,
- coleta da informação necessária para modelar ou estimar ruído,
- escolha de estratégias para manipular campos de dados ausentes,
- formatação de dados de forma a adequá-los à ferramenta de mineração

# Redução de dados

4. Projeção (Transformation): localização de características úteis para representar os dados dependendo do objetivo da tarefa,

- visando a redução do número de variáveis e/ou instâncias a serem consideradas para o conjunto de dados,

# Mineração de dados

- 5. Datamining
  - selecionar os métodos a serem utilizados para localizar padrões nos dados,
  - seguida da efetiva busca por padrões de interesse numa forma particular de representação ou conjunto de representações;
  - busca pelo melhor ajuste dos parâmetros do algoritmo para a tarefa em questão.

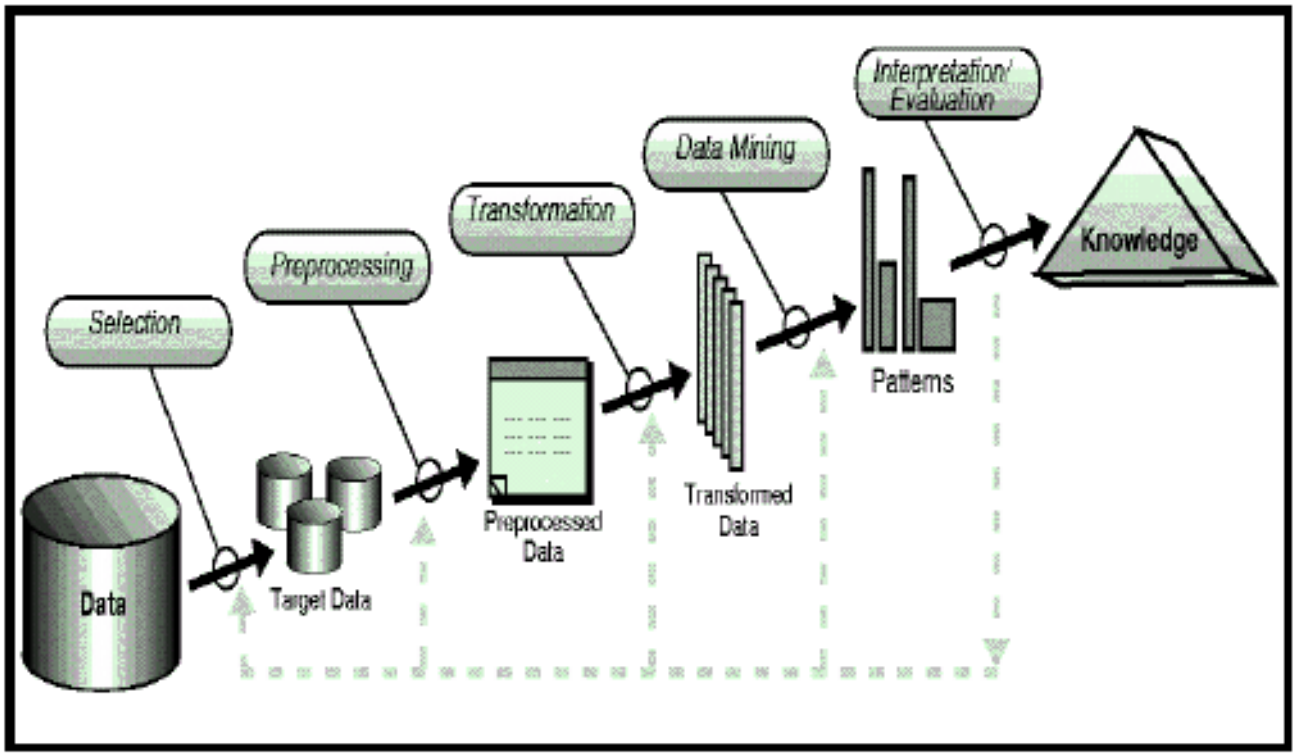
# Interpretação

- Interpretação dos padrões minerados (Interpretation/ Evaluation), com um possível retorno aos passos 1-6 para posterior iteração.

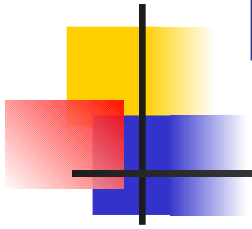
# Implantação

- Implantação do conhecimento descoberto (Knowledge):
- incorporar este conhecimento à performance do sistema,
- ou documentá-lo e reportá-lo às partes interessadas.

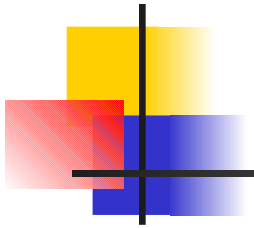
# Etapas de KDD [Fayyad et al. 1996]



# Técnicas e Algoritmos



- Bases de dados são altamente suscetíveis a dados ruidosos
- erros e valores estranhos
- incompletos (valores de atributos ausentes)
- e inconsistentes (discrepâncias semânticas)



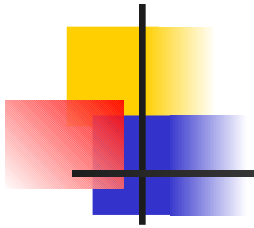
- Técnicas de pré-processamento e transformação de dados são aplicadas para aumentar a qualidade e o poder de expressão dos dados a serem minerados.
- Estas fases tendem a consumir a maior parte do tempo dedicado ao processo de KDD (aproximadamente 70%).

# Pré-processamento de Dados

- Rotinas de limpeza de dados tentam suprir valores ausentes,
- reduzir discrepâncias de valores ruidosos e corrigir inconsistências.

# Técnicas Valores Ausentes

1. Ignorar a tupla
2. Suprir valores ausentes
   a) manualmente;
   b) através de uma constante global;
   c) utilizando a média do atributo;
   d) utilizando a média do atributo para todas as instâncias da mesma classe;
   e) com o valor mais provável (regressão, inferência etc.).



- As técnicas 2b, 2c, 2d e 2e podem "viciar" os dados.
- A técnica 2e é uma estratégia interessante, pois em comparação com outros métodos utiliza um maior número de informações dos dados disponíveis.

# Ruídos nos dados

- Ruídos nos dados são erros aleatórios ou variâncias numa variável mensurada.
- A eliminação de ruídos pode ser realizada através de:

  1 - Interpolação;

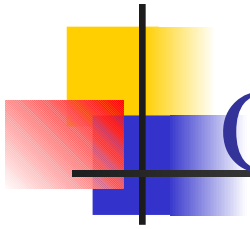
  2 - Agrupamento;

  3 - Inspeção humana e computacional combinadas;

  4 – Regressão.

# Inconsistências

- corrigidos manualmente através de referências externas.
- Rotinas de consistência evitam a inserção de dados incorretos
- Discrepâncias podem ser combatidas através de dependências funcionais.



# O que é mineração de dados

Mineração de Dados é um passo no processo de KDD que consiste na aplicação de análise de dados e algoritmos de descobrimento que produzem uma enumeração de padrões (ou modelos) particular sobre os dados.

Usama Fayyad, Ai Magazine, 1996.

# Mineração de dados

- Extrair informações úteis de bilhões de bits de dados.
- O processo não-trivial de identificar padrões válidos, novos, potencialmente úteis e compreensíveis em dados.
- Técnicas/ferramentas para apresentar e analisar dados.

# Mineração de dados

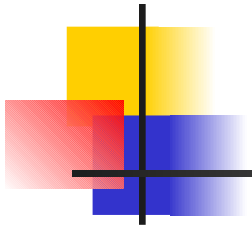
- descobre padrões, tendências, infere regras
- suporta, revisa e examina decisões

# Exemplo de conhecimento extraído

- Banco de dados de lojas de produtos eletrônicos
- OLAP
  - Quantos videogames do tipo XYZ foram vendidos para o cliente ABC na data dd/mm/aa?
- Mineração
  - Se (idade < 18) E (profissão = "estudante") Então (compra= "videogame") (90%)
  - Utilidade: estratégias de marketing.

# Áreas de pesquisa relacionadas

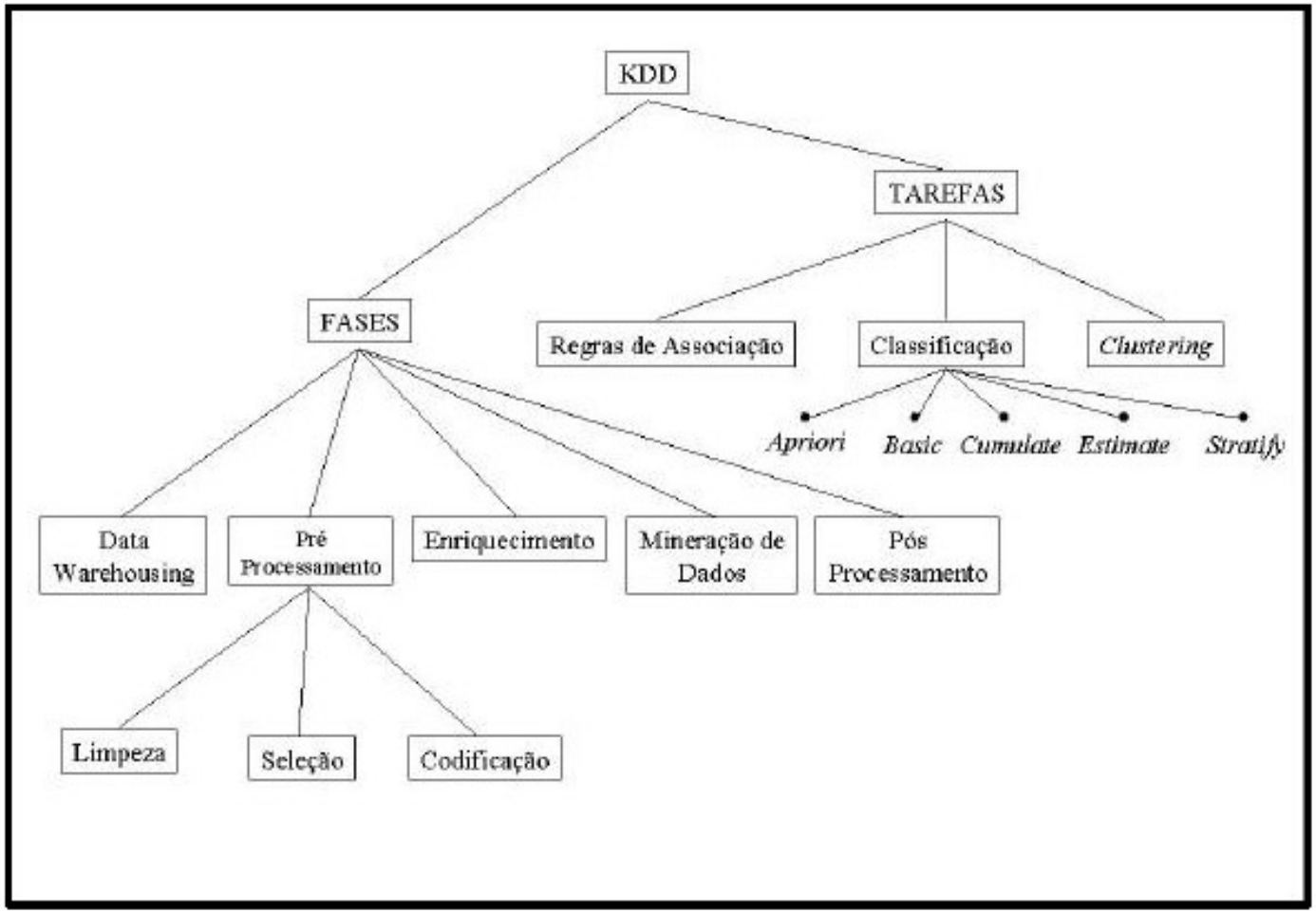
- Aprendizagem de máquina, reconhecimento de padrões, bancos de dados, estatística e Visualização de dados.

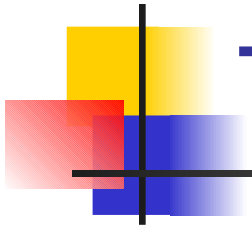


# Machine Learning

- Abordagens
  - Baseado em lógica
  - Algoritmos genéticos
  - Programação genética
  - Redes neurais
- Tarefas
  - Associação
  - Agrupamento (Clustering)
  - Classificação

# Taxonomia do processo de KDD





# Tarefa de Classificação

- Cada exemplo pertence a uma classe pré-definida
- Cada exemplo consiste de:
  - Um atributo classe
  - Um conjunto de atributos preditores
- O objetivo é predizer a classe do exemplo dado seus valores de atributos preditores.

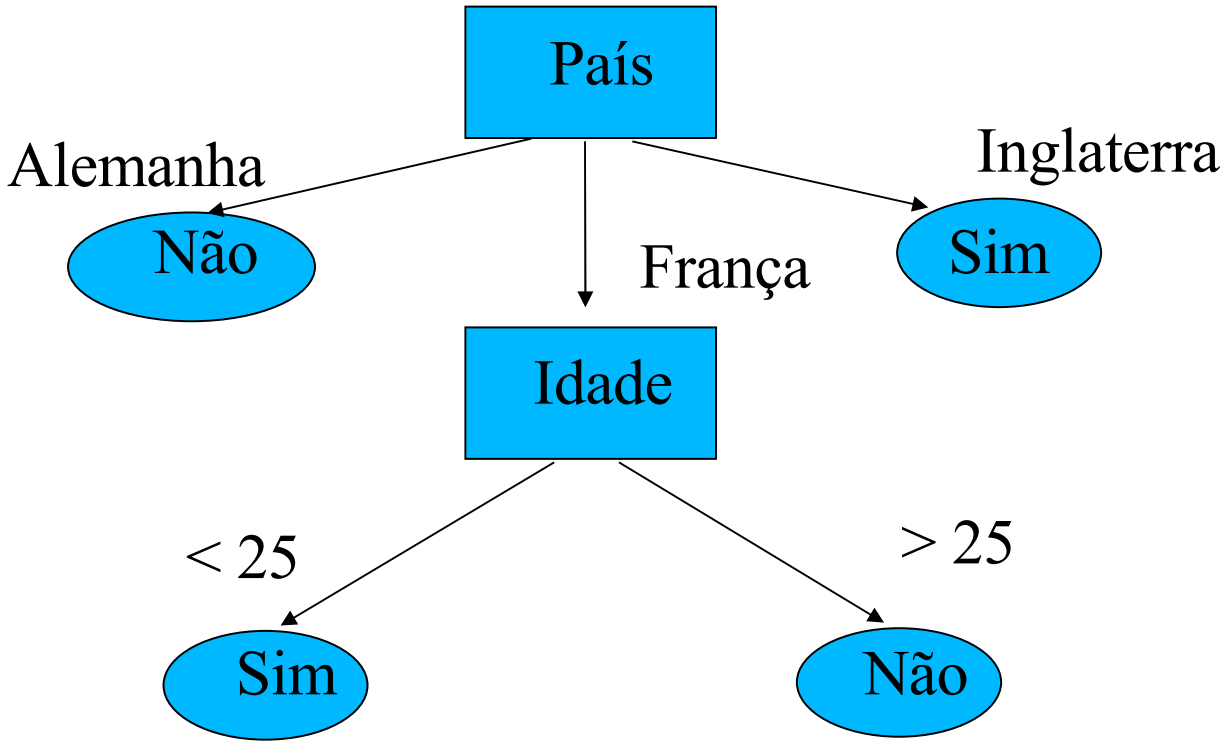
# Exemplo: Extraído de Freitas & Lavington 98

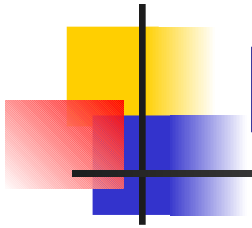
- Uma editora internacional publica o livro “Guia de Restaurantes Franceses na Inglaterra” em 3 países: Inglaterra, França e Alemanha.
- A editora tem um banco de dados sobre clientes nesses 3 países, e deseja saber quais clientes são mais prováveis compradores do livro (para fins de mala direta direcionada).
  - Atributo meta: comprar (sim/não)
- Para coletar mais dados: enviar material de propaganda para uma amostra de clientes, registrando se cada cliente que recebeu a propaganda comprou ou não o livro.

# Exemplo de Classificação

| Sexo | País | Idade | Compra |
|---|---|---|---|
| M | França | 25 | Sim |
| M | Inglaterra | 21 | Sim |
| F | França | 23 | Sim |
| F | Inglaterra | 34 | Sim |
| F | França | 30 | Não |
| M | Alemanha | 21 | Não |
| M | Alemanha | 20 | Não |
| F | Alemanha | 18 | Não |
| F | França | 34 | Não |
| M | França | 55 | Não |

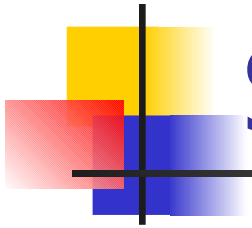
# Árvores de Decisão





# Regras de associação

- 90% das mulheres com carros esporte vermelhos e cães pequenos usam
  - Chanel 5;
- O número de regras de associação que podem ser encontrados em um banco de dados é quase infinito.

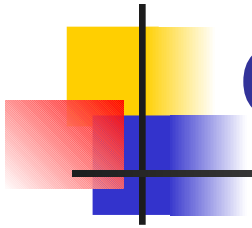


# Supermercado

- Itens de compras de clientes
  - Leite, pão, manteiga
  - Arroz, feijão
  - Leite, café, pão
  - Pão, manteiga
- Leite => Pão
- Arroz => Feijão
- Pão => Manteiga

# Associação vs. Classificação [Freitas 2000]

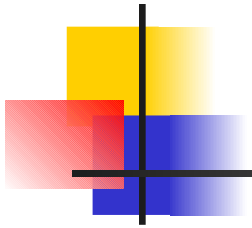
- Associação:  problema é "simétrico": todos os items podem aparecer ou no antecedente ou no consequente de uma regra;
- qualidade de uma regra é avaliada por fatores de Conf e Sup definidos pelo usuário;
- definição do problema é determinística: o sistema deve encontrar todas regras com Sup e Conf maior ou igual a limiares pré-definidos;
- Na maioria da literatura, o desafio é projetar algoritmos eficientes.



# Classificação:

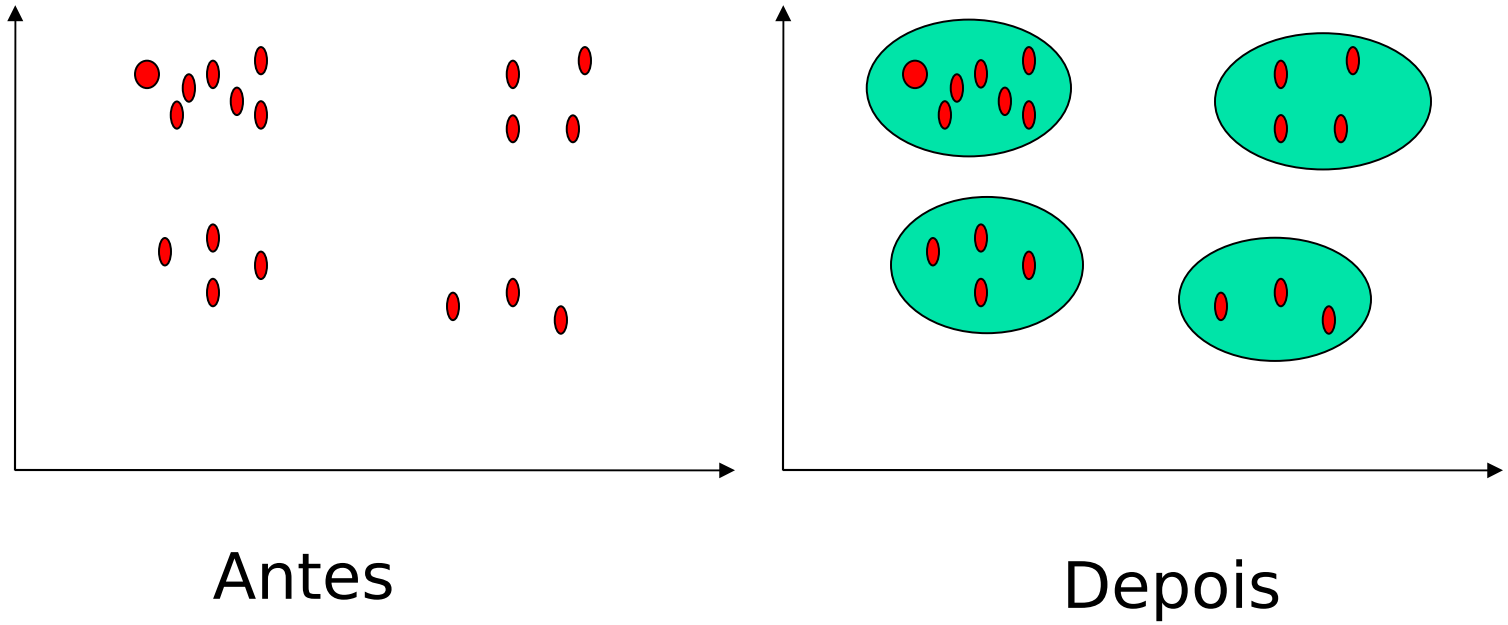
- problema é assimétrico: um único atributo meta a ser previsto, dados demais atributos;
- regras são avaliadas em dados de teste não vistos durante treinamento (prever o futuro);
- qualidade de uma regra é muito mais difícil de avaliar, logo não é muito claro quais regras deveriam ser descobertas pelo sistema;
- eficiência ainda é importante, mas o desafio principal é projetar algoritmos eficazes.
- problema é não-determinístico (indução)

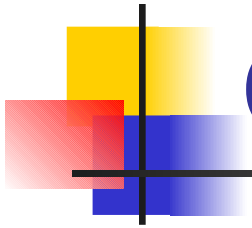
# “Clustering” (Agrupamento)



- O sistema “inventa” classes, agrupando registros semelhantes (isto é, com valores de atributos semelhantes) em uma mesma classe.

# Clusters





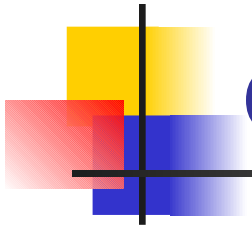
# Cluster

- Após clustering, pode-se aplicar métodos de classificação e sumarização para descobrir regras de classificação (que discriminem registros de diferentes classes) e regras de sumarização (que produzem descrições características de cada classe)

# Classificação versus clustering.

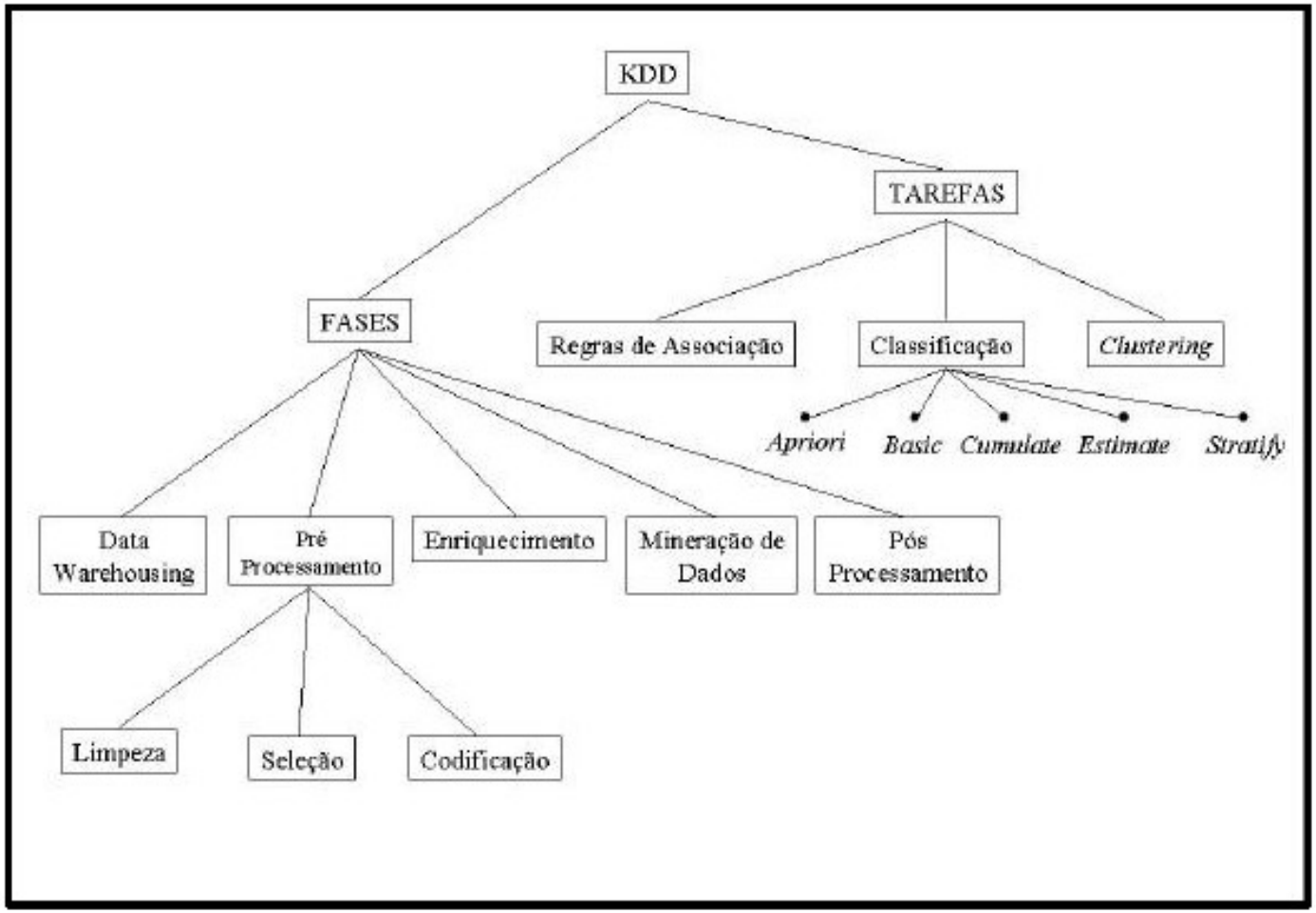
- Classificação:
- há um único atributo meta, e os demais atributos são previsores;
- parte do problema consiste em determinar automaticamente a importância dos atributos previsores;
- há medidas objetivas para medir a qualidade da classificação (ex. taxa de acerto);
- classificação é usada principalmente para previsão.

# Classificação versus clustering



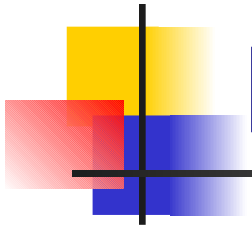
- Clustering:
- não há um atributo especial;
- a importância de cada atributo é geralmente considerada equivalente à dos demais;
- é difícil medir a qualidade de clustering;
- Clustering é usado principalmente para exploração e sumarização de dados.

# Taxonomia do processo de KDD



# Software Weka

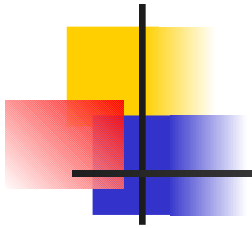
Waikato 2004, Witten & Frank 2000



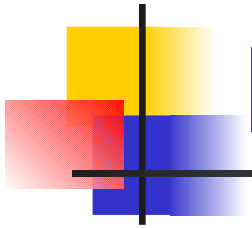
# Ferramenta

- algoritmos de
  - preparação de dados
  - aprendizagem de máquina (mineração)
  - validação de resultados
- /public/soft/linux/weka...
- Java –jar weka.jar

# Interface e Funcionalidades



- (A) Open File, Open URL, Open DB
- (B) No botão filter é possível efetuar sucessivas filtragens de atributos e instâncias na base de dados previamente carregada
  - Seleção
  - Discretização
  - Normalização
  - Amostragem



# Formato arff (header)

```
% 1. Title: Iris Plants Database
%
% 2. Sources:
% (a) Creator: R.A. Fisher
% (b) Donor: Michael Marshall (MARSHALL%PLU@io.arc.nasa.gov)
% (c) Date: July, 1988
%
 @RELATION iris

 @ATTRIBUTE sepallength NUMERIC
 @ATTRIBUTE sepalwidth NUMERIC
 @ATTRIBUTE petallength NUMERIC
 @ATTRIBUTE petalwidth NUMERIC
 @ATTRIBUTE class {Iris-setosa,Iris-versicolor,Iris-virginica}
```

# Formato arff (corpo)

@DATA

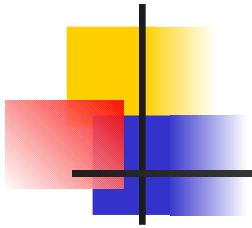
5.1,3.5,1.4,0.2,Iris-setosa

4.9,3.0,1.4,0.2,Iris-setosa

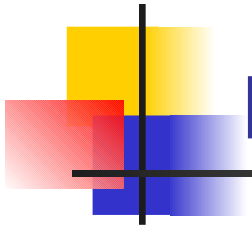
4.7,3.2,1.3,0.2,Iris-setosa

4.6,3.1,1.5,0.2,Iris-setosa

5.0,3.6,1.4,0.2,Iris-setosa



# Regras de Associação

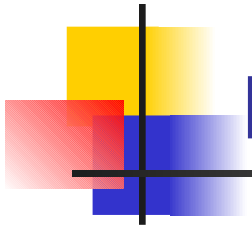


# Descoberta de Regras de Associação

- Definição original: tipo especial de dados, chamado “basket data” (dados de cesta)[Agrawal et al 96]
- Cada registro corresponde a uma transação de um cliente, com itens assumindo valores binários (sim/não), indicando se o cliente comprou ou não o respectivo item.

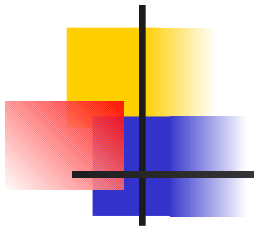
# Exemplo: [Freitas & Lavington 98]

| | leite | café | cerveja | pão | manteiga | arroz | feijão |
|---|---|---|---|---|---|---|---|
| 1 | não | sim | não | sim | sim | não | não |
| 2 | sim | não | sim | sim | sim | não | não |
| 3 | não | sim | não | sim | sim | não | não |
| 4 | sim | sim | não | sim | sim | não | não |
| 5 | não | não | sim | não | não | não | não |
| 6 | não | não | não | não | sim | não | não |
| 7 | não | não | não | sim | não | não | não |
| 8 | não | não | não | não | não | não | sim |
| 9 | não | não | não | não | não | sim | sim |
| 10 | não | não | não | não | não | sim | não |



# Descoberta de Regras de Associação

- Uma regra de associação é um relacionamento
  - SE (X) ENTÃO (Y), onde X e Y são conjuntos de itens, com interseção vazia.
- A cada regra são atribuídos 2 fatores:
  - Suporte (Sup.) = No. de registros com X e Y /No. Total de registros
  - Confiança (Conf.) = No. de registros com X e Y/ No. de registros com X
- Tarefa: descobrir todas as regras de associação com um mínimo Sup e um mínimo Conf.



Sup. = No. de registros com X e Y /No. Total de registros,

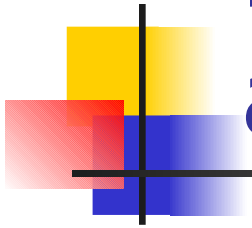
Conf = No. de registros com X e Y/ No. de registros com X

- Conjunto de Items Frequente: café, pão. Sup. = 0,3
- Regra: SE (café) ENTÃO (pão). Conf. = 1
- Conjunto de Items Frequente: café, manteiga. Sup. = 0,3
- Regra: SE (café) ENTÃO (manteiga). Conf. = 1
- Conjunto de Items Frequente: pão, manteiga. Sup = 0,4
- Regra: SE (pão) ENTÃO (manteiga). Conf. = 0,8

## Sup. = No. de registros com X e Y /No. Total de registros, Conf = No. de registros com X e Y/ No. de registros com X

- Regra: SE (manteiga) ENTÃO (pão). Conf. = 0,8
- Conjunto de Items Frequente: café,pão,manteiga Sup.=0,3
- Regra: SE (café E pão) ENTÃO (manteiga). Conf.=1
- Regra: SE (café E manteiga) ENTÃO (pão). Conf.=1
- Regra: SE (café) ENTÃO (manteiga E pão). Conf.=1

# Descobrindo regras de associação



- Algoritmo tem 2 fases.
- Fase I: Descobrir conjuntos de itens frequentes. Descobrir todos os conjuntos de itens com suporte maior ou igual ao mínimo suporte especificado pelo usuário.
- Fase II: Descobrir regras com alto fator de confiança. A partir dos conjuntos de itens frequentes, descobrir regras de associação com fator de confiança maior ou igual ao especificado pelo usuário.

# Calculando o suporte de conjuntos de itens

- Suporte = No. de transações contendo o conjunto de itens, dividido pelo No. total de transações.
- Fase I: Passo 1: Calcular suporte de conjuntos com 1 item.
  - leite: Sup = 0,2;  café: Sup = 0,3; cerveja: Sup = 0,2; pão: Sup = 0,5; manteiga: Sup = 0,5; arroz: Sup = 0,2; feijão: Sup = 0,2;
  - Itens frequentes (Sup >= 0,3): café, pão, manteiga

# Calcular suporte de conjuntos com 2 itens

- Passo 2: Calcular suporte de conjuntos com 2 itens
- Otimização: Se um item I não é frequente, um conjunto com 2 itens, um dois quais é o item I, não pode ser frequente. Logo, conjuntos contendo item I podem ser ignorados.
  - Conjunto de itens: café, pão. Sup = 0,3.
  - Conjunto de itens: café, manteiga. Sup = 0,3.
  - Conjunto de itens: manteiga, pão. Sup = 0,4.
  - Conjuntos de itens frequentes (Sup > = 0,3):

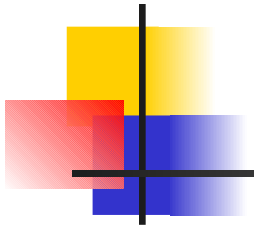
⑦ {café, pão}, {café, manteiga}, {manteiga, pão}

# Calcular suporte de conjuntos com 3 itens.

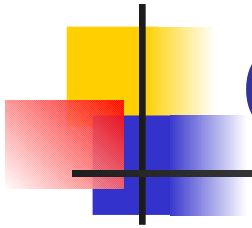
- Passo 3: Calcular suporte de conjuntos com 3 itens.
  - Otimização:Se o conjunto de itens {I, J} não é frequente, um conjunto com 3 itens incluindo os itens {I, J} não pode ser frequente. Logo, conjuntos contendo itens {I, J} podem ser ignorados.
  - Conjunto de itens: café, pão, manteiga. Sup = 0,3.
  - Conjuntos de itens frequentes (Sup >= 0,3): {café, pão, manteiga}.

# Fator de confiança de regras

- Calculando fator de confiança de regras candidatas, geradas a partir de conjuntos de itens frequentes.
  - Conf. da regra ``SE X ENTÃO Y'' é No. de transações contendo X e Y dividido pelo No. de transações com X.
- Conjunto de itens: {café, pão}.
  - SE café ENTÃO pão. Conf = 1,0.
  - SE pão ENTÃO café. Conf = 0,6.

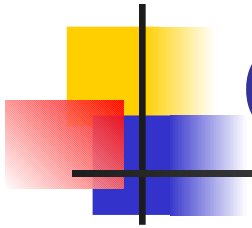


- Conjunto de itens: {café, manteiga}.
  - SE café ENTÃO manteiga. Conf = 1,0.
  - SE manteiga ENTÃO café. Conf = 0,6.
- Conjunto de itens: {manteiga, pão}.
  - SE manteiga ENTÃO pão. Conf = 0,8.
  - SE pão ENTÃO manteiga. Conf = 0,8.



# Confiança de regras

- Conjunto de itens: {café, manteiga, pão}.
  - SE café, pão ENTÃO manteiga. Conf = 1,0.
  - SE café, manteiga ENTÃO pão. Conf = 1,0.
  - SE manteiga, pão ENTÃO café. Conf = 0,75.
  - SE café ENTÃO pão, manteiga. Conf = 1,0.
  - SE pão ENTÃO café, manteiga. Conf = 0,6.
  - SE manteiga ENTÃO café, pão. Conf = 0,6.



# Confiança de regras

- Finalmente, seleciona-se regras com Conf. maior ou igual ao valor mínimo especificado pelo usuário (ex. 0,8).