Resumos ADI - 2021/2022

Exemplos de sistemas de Aprendizagem

- Aprendizagem Simbólica;
- Redes Neuronais Artificiais;
- Raciocínio Baseado em Casos;
- Árvores de Decisão;
- Algoritmos Genéticos e Evolucionários;
- Máquinas de Vetores de Suporte;
- Inteligência de Grupo;
- Segmentação;
- Classificação

Sistemas de Aprendizagem/Machine Learning

Paradigma de computação em que a característica essencial do sistema se revela pela sua capacidade de aprender de modo autónomo e independente;

- A característica diferenciadora dos algoritmos de *Machine Learning* é a de que são algoritmos *data-driven*;
    - Um hipotético algoritmo aprenderia o que é uma mesa pela definição algorítmica da configuração de uma mesa;
    - Um algoritmo de *Machine Learning* aprende sem necessidade de que seja codificada a solução do problema;
    - Um algoritmo de *Machine Learning* aprende a partir de diversos exemplos de mesas, aprendendo a dizer se um determinado objeto é ou não é uma mesa.

Aprendizagem com Supervisão

Paradigma de aprendizagem em que os casos que se usam para aprender contêm informação acerca dos resultados pretendidos, sendo possível estabelecer uma relação entre os valores pretendidos e os valores produzidos pelo sistema;

- A grande maioria dos algoritmos de *Machine Learning* usa aprendizagem com supervisão;
- Aprendizagem supervisionada significa que os dados de entrada (x) e os resultados (y), tornam possível que o algoritmo aprenda uma função (f) de mapeamento dos dados nos resultados: $y = f(x)$;
- Diz-se supervisionada porque este mapeamento é acompanhado por um professor/ /treinador que supervisiona o processo de aprendizagem;
- Normalmente, são divididos em duas categorias:
    - Classificação: quando os resultados são discretos (preto, branco, cinza…);
    - Regressão: quando os resultados são contínuos (variação da temperatura ou da luz solar ao longo do dia)

Aprendizagem sem Supervisão

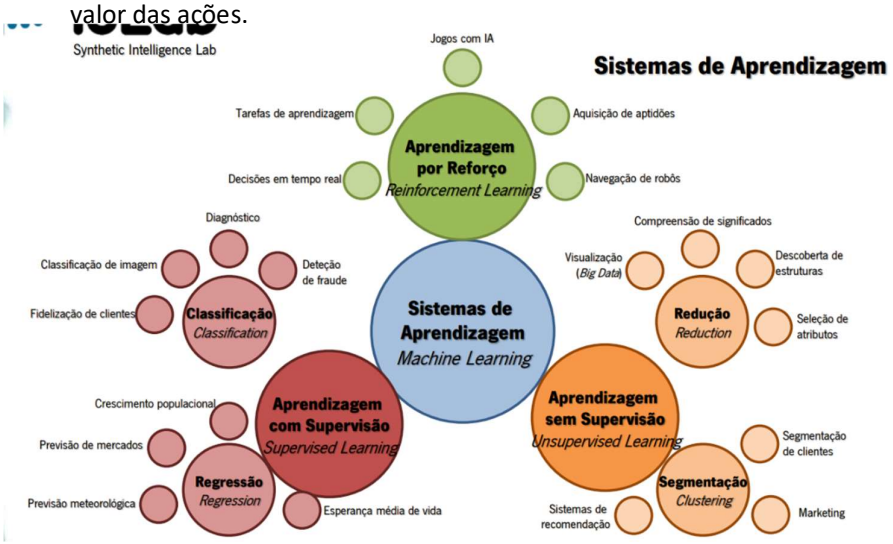
Paradigma de aprendizagem em que não são conhecidos resultados sobre os casos, apenas os enunciados dos problemas, tornando necessário a escolha de técnicas de aprendizagem que avaliem o funcionamento interno do sistema;

- A aprendizagem não supervisionada significa que existem dados de entrada (x) mas não existem os correspondentes resultados;
- O objetivo deste tipo de aprendizagem é o de modelar a estrutura ou a distribuição dos dados do problema;
- São, normalmente, divididos em duas categorias:
    - Segmentação: quando se pretende organizar os dados em grupos coerentes (agrupar clientes que compram bebidas açucaradas);
    - Associação: quando se pretende conhecer regras que associem o comportamento demonstrado pelos dados (pessoas que comprar bebidas açucaradas não compram bebidas alcoólicas);

Aprendizagem por Reforço

Paradigma de aprendizagem que, apesar de não ter informação sobre os resultados pretendidos, permite efetuar uma avaliação sobre se os resultados produzidos são bons ou maus;

- Algoritmos de *Reinforcement Learning* usam técnicas de auto-alimentação de sinais, com vista a melhorar os resultados, por influência da noção de recompensa/penalização;
- Não se pode comparar com Aprendizagem Supervisionada uma vez que a "opinião" sobre os resultados não é dada por um professor/treinador;
- Também não se pode considerar Aprendizagem não Supervisionada, uma vez que não existe ausência absoluta de informação sobre os resultados;
- A aprendizagem dá-se pela capacidade de crítica sobre os próprios resultados produzidos pelo algoritmo;
    - Q-Learning: assume que está a seguir uma política ótima e usa-a para atualização dos valores das ações;
    - SARSA: considera a política de controlo que está a ser seguida e atualiza o valor das ações.



Metodologias de Análise de Dados

Uma **Metodologia para Análise de Dados** descreve e cria um **conjunto de passos** pelos quais deverá passar o desenvolvimento de um **Projeto de Aprendizagem Automática (*Machine Learning*)** para a resolução de problemas.

Enquadrar um processo de Análise de Dados ao abrigo de uma metodologia:

- Garante maior robustez;
- Facilita a sua compreensão, implementação e desenvolvimento;
- Permite a replicação de processos;
- Auxilia no planeamento e na gestão do projeto;
- Confere "maturidade" ao processo;
- Encoraja a adoção de melhores práticas.

CRoss Industry Standard Process for Data Mining (Daimler Chrysler, SPSS, NCR)

- Objetivos:
  - Definir um processo de Análise de Dados para a indústria;
  - Construir e disponibilizar ferramentas de apoio;
  - Assegurar a qualidade dos projetos de Análise de Dados;
  - Reduzir os conhecimentos específicos necessários para conduzir um processo de Análise de Dados.

O CRISP-DM é um modelo de processos com vista a definir um "guião" para o desenvolvimento de projetos de AD, que se desenrola em 6 etapas:

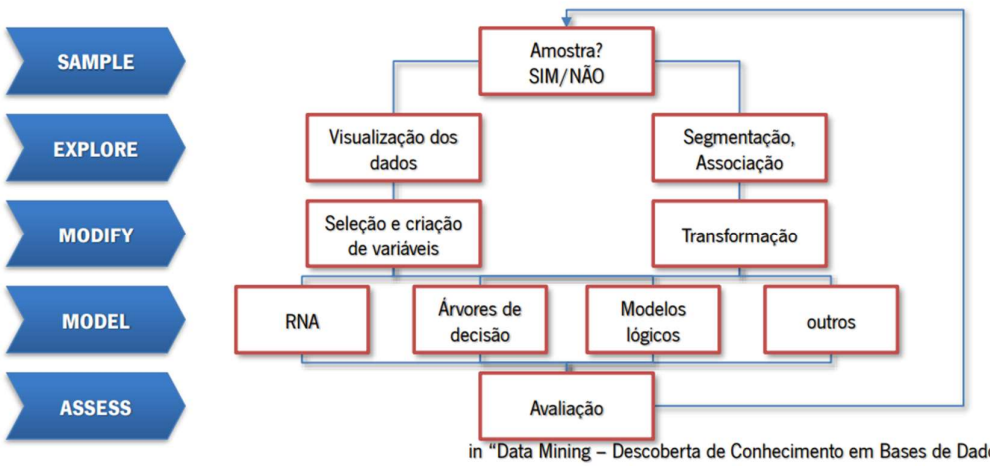
- **Business Understanding/Estudo do Negócio**:
  - Compreensão dos objetivos do projeto e definição do problema de AD;
- **Data Understanding/Estudos dos Dados**:
  - Obter os dados e identificar a qualidade dos dados;
- **Data Preparation/Preparação dos Dados**:
  - Seleção de atributos e limpeza dos dados;
- **Modeling/Modelação**:
  - Experimentação com as ferramentas de AD;
- **Evaluation/Avaliação**:
  - Comparação dos resultados com os objetivos do negócio;
- **Deployment/Desenvolvimento**:
  - Colocação do modelo em produção.

Sample, Explore, Modify, Model and Assess;

- Produto de *Data Mining* desenvolvido pelo SAS Institute Inc.;
- Definição SAS:
  - "*Data Mining* é o processo de extrair conhecimento e relações complexas de grandes volumes de dados."
- Motivação:
  - necessidade de definir, padronizar e integrar sistemas ou processos de *Data Mining* nos ciclos de produção.
- Desenvolvimento focado na ferramenta SAS Enterprise Miner.

Divide o processo de *Data Mining* em 5 etapas:

- **Sample/Amostragem**: • Extração de dados do universo do problema;
  - Baseia o processo de Data Mining no conceito de "amostra" do problema;
  - Amostra pequena e significativa;
  - Proporciona flexibilidade e rapidez no tratamento dos dados.
- **Explore/Exploração**:
  - Exploração visual e/ou numérica das tendências;
  - Refinamento do processo de descoberta (mining);
  - Técnicas estatísticas: regressão linear, mínimos quadrados, distribuição de Poisson, etc.;
  - Procura de tendências imprevistas nos dados;
- **Modify/Modificação**:
  - Concentração de todas as modificações necessárias;
  - Inclusão de informação;
  - Seleção ou introdução de novas variáveis;
  - Objetivo: criar, selecionar e adaptar variáveis para a próxima etapa;
- **Model/Modelação**:
  - Definição das técnicas de construção de modelos de Data Mining: redes neuronais artificiais, árvores de decisão, regressão linear, etc.;
  - Dependente do tipo de dados presentes em cada modelo (p.ex., RNA são mais adequadas quando os dados do problema apresentam relacionamentos complexos);
- **Assess/Avaliação**:
  - Aferição do desempenho do modelo construído para Data Mining;
  - Aplicação do modelo a uma amostra de dados de teste;
  - Procedimento de ajuste do modelo.



in "Data Mining – Descoberta de Conhecimento em Bases de Dados"
Manuel Filipe Santos, Carla Azevedo

Predictive Model Markup Language;

- Desenvolvido por investigadores de Data Mining e várias empresas (NCR, SPSS, etc.);
- A especificação PMML encontra-se em fase de desenvolvimento e consolidação (versão 4.2.1);
- Utilizada por diversas aplicações (IBM DB2 Data Warehouse Edition v.10.5, SAS Enterprise Miner v.5.1, v.5.3, v.7.1, v.13.1, SPSS Statistics v.21); (http://www.dmg.org/products.html)
- Expandir para transformá-la num padrão para o WWW;
- O PMML é uma linguagem para descrever modelos de DM;
- Utiliza XML para representar modelos de DM.

Objetivos:

- Permitir que aplicações utilizem diversas fontes de dados sem se preocuparem com as diferenças entre elas;
- Permitir a utilização combinada e/ou cooperativa de modelos de Data Mining;
- Permitir a administração de modelos de Data Mining baseados em áreas de negócio.

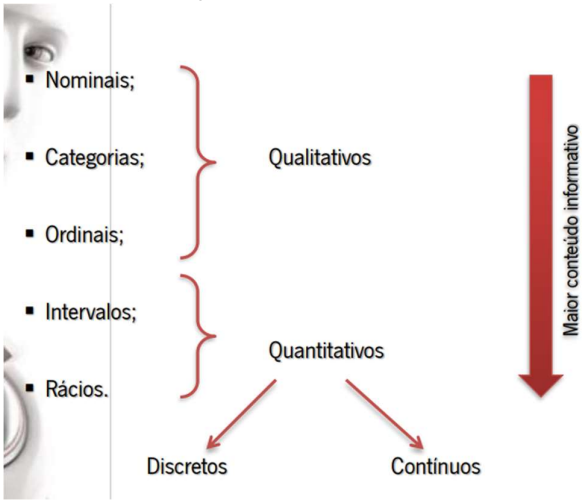
Preparação de Dados

- O principal objetivo da preparação dos dados consiste em transformar os datasets por forma a que a informação neles contida esteja adequadamente exposta à ferramenta de extração de conhecimento;
- A preparação dos dados "também prepara o preparador" por forma selecionar os modelos de EC mais adequados;
- Os dados têm de ser formatados para se adequarem a uma determinada ferramenta de EC;
- Os dados recolhidos do "mundo real":
    - são incompletos;
        - falta de valores em alguns atributos;
        - falta de alguns atributos;
        - dados agregados ou generalizados;
        - Código postal: 4710-... Braga;
        - Nº de filhos: "";
    - contêm lixo;
        - identificam valores impossíveis;
        - Salário: -1.000EUR;
        - Idade: 321;
        - Data: 31/novembro/2017;
        - País: Catalunha;
    - podem conter inconsistências.
        - encontram-se discrepâncias entre valores ou nomes;
        - Idade = 35; Data de nascimento = 31/maio/1969;
        - Sexo: "M/F"; "0/1"; "Masculino/Feminino/Desconhecido";
        - diferenças entre valores de registos duplicados.

Tipos de Dados

Os tipos dos dados diferem na sua natureza e na quantidade de informação que proporcionam: **Qualitativos ou Quantitativos**.

- **Nominais**:
  - Atribui nomes únicos a objetos:
    - Não existe outra informação que se possa deduzir;
    - Nomes de pessoas;
    - Códigos de identificação;
- **Categorias**:
  - Atribui categorias a objetos:
    - Podem ser valores numéricos, mas são não ordenados;
    - Código postal;
    - Sexo;
    - Cor dos olhos;
- **Ordinais**:
  - Os valores podem ser ordenados naturalmente;
    - Classificação: excelente, bom, suficiente, etc.;
    - Temperatura: frio, morno, quente;
- **Intervalos**:
  - É possível calcular a distância entre dois valores;
    - Temperatura;
    - Humidade;
- **Rácios**:
  - Os valores podem ser utilizados para determinar um rácio significativo entre eles:
    - Salário;
    - Balanço bancário.

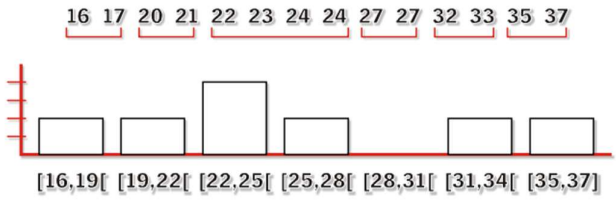


Tarefas de Preparação de Dados

- **Discretização/Enumeração**:
  - Redução de dados com importante aplicação a dados numéricos;
- **Limpeza**:
  - Preenchimento de valores de atributos;
  - Remoção de lixo dos dados;
  - Remoção de valores impossíveis;
  - Resolução de inconsistências;
- **Integração**:
  - Múltiplas fontes de dados (BD's, ficheiros, papel, web, etc.);
- **Transformação**:
  - Normalização e agregação de dados;
- **Redução**:
  - Obtenção de representações de dados menos volumosas, mas com capacidade para produzir idênticos resultados analíticos;
  - Redução de dimensões;
  - Compressão de dados.

Discretização/Enumeração

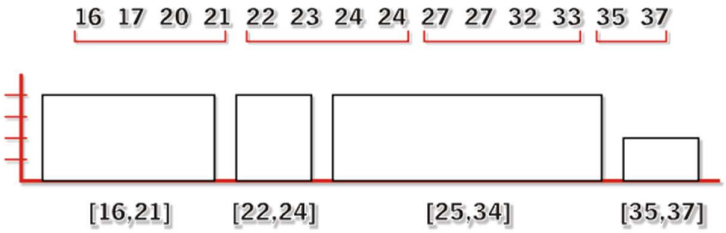
- Utiliza-se a discretização (ou enumeração) para reduzir o número de valores de um atributo contínuo, dividindo-o em intervalos;
  - Os métodos mais utilizados (Naïve Bayes, CHAID, etc.), requerem valores discretos;
  - Redução do tamanho dos dados;
  - Método utilizado para produzir sumariação dos dados;
  - (Sinónimo de binning.)
- Discretização de igual largura/ Equal-width binning:
  - Divide a gama de valores em N intervalos de igual largura, resultando numa grelha uniforme;
  - Sendo A e B os limites da gama de valores, a largura dos intervalos será L = (B - A) / N:



  - Vantagens:
    - Simples e fácil de implementar;
    - Produz abstrações de dados razoáveis;

- Desvantagens:
  - Não supervisionado;
  - Quem determina N?;
  - Sensível a valores fronteira.

- Discretização de igual altura/ Equal-height binning:
  - Divide a gama de valores em N intervalos, contendo, cada um, aproximadamente a mesma quantidade de valores:



  - Normalmente preferida à discretização de igual largura, uma vez que permite evitar o "amontoar" de valores;
  - Na prática, utiliza-se uma discretização de "quase-igual" altura, garantindo intervalos mais intuitivos;
  - Deverá impedir a dispersão de valores frequentes por diferentes intervalos;
  - Deverá criar intervalos separados para valores especiais ("0").

(outros métodos de descritização)

- Método 1R:
  - Método supervisionado, baseado na divisão por binning;
- Discretização baseada em Entropia;
- Discretização baseada em Impurezas;
- Deteção de limites;
- etc.

Limpeza

- Ausência de valores em determinados atributos devido a:
  - inconsistência;
  - dados não registados;
  - análise incorreta;
  - dados registados de forma errada;
  - etc.
- A ausência de dados pode revelar algo sobre que campos não foram preenchidos!

Como tratar:

- Ignorar os registos onde faltam os dados e lidar, apenas com os dados conhecidos;
  - não aconselhável se a quantidade de dados em falta em cada atributo for elevada;

- Ignorar os atributos onde faltam os dados;
    - não aconselhável se os atributos onde acontece revelarem informação importante;
- Preencher (manualmente) os dados em falta:
    - é mais trabalhoso preencher ou é mais difícil adivinhar?
- Preencher os dados em falta com um mesmo valor ("talvez"):
    - pode criar tendências nos dados ou novas classes;
- Preencher com o valor médio do atributo:
    - pouco impacto negativo, desde que o desvio padrão não seja grande;
- Preencher com o valor mais frequente do atributo;
- Quantos mais valores "inventados", maior o desvio dos dados que caracterizam o problema face à realidade que o problema ilustra!

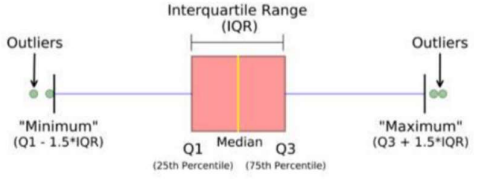
EVITAR ADICIONAR DISTORÇÃO AOS DADOS

Integração

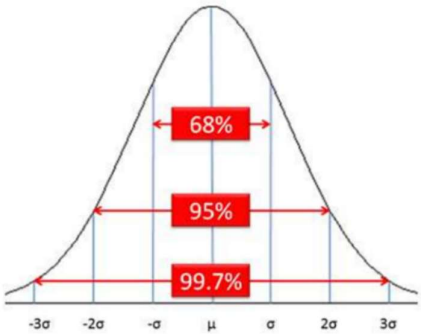
- Os dados que caracterizam o problema podem ter proveniências diversas;
- O objetivo da integração é o de compor um conjunto de peças de informação numa coleção coerente e integrada de dados.
- Detetar e resolver conflitos entre os dados:
    - qual a fonte de dados mas fiável, quando os valores que transportam são inconsistentes?
- Integração exige "conhecimento do negócio".

Transformação

- Alisamento (smoothing):
    - Remover lixo/ruído dos dados (binning, regressão, clustering);
- Agregação:
    - Pressupõe que o resultado sumaria os dados iniciais; (resumo de vendas trimestrais, durante 5 anos, em valores anuais)
- Generalização:
    - Hierarquização de conceitos:
        - distrito → cidade → rua;
        - Valores diferentes: 18 → centenas → (largos) milhares
- Construção de atributos:
    - Construção de novos atributos a partir de outros (cálculo do preço líquido baseado no preço ilíquido e no IVA);
- Uniformização:
    - Pretende evitar que atributos com uma gama alargada de valores sobressaiam em relação a outros atributos com menor quantidade de valores:
        - Normalização (normalization: [0;1]);
        - Padronização (standardization/Z-score normalization: xɲ=0; ∂=1);



- Deteção de valores atípicos:
    - Por visualização:
        - Box plots
        - Z-Score (desvio padrão)



Redução

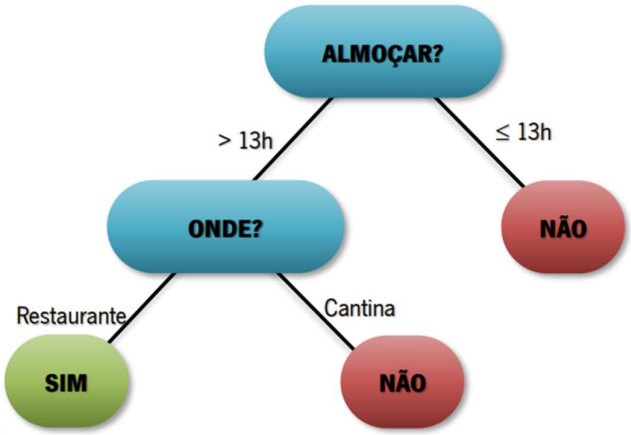
- Um *Data Warehouse* pode armazenar largos terabytes de dados;
- Realizar tarefas de EC em tais quantidades de dados pode tornar-se impraticável!
- A **Redução de dados** pretende obter uma representação reduzida do volume de dados, mas produzindo os mesmos (ou quase os mesmos) resultados analíticos.

Estratégias:

- Construção de cubos de dados:
    - as operações de agregação são aplicadas de modo a construir cubos de dados;
- Redução de dimensões:
    - remoção de atributos que se mostrem irrelevantes, redundantes ou pouco interessantes para a análise;
    - Principle Component Analysis (PCA);
- Compressão de dados:
    - aplicação de técnicas de compressão ou de transformação para comprimir a representação dos dados originais;
- Redução de quantidade:
    - redução do volume de dados (técnicas paramétricas ou não paramétricas);
- Discretização e generalização de conceitos:
    - redução da quantidade de valores por atributo.

Árvores de Decisão

- Uma Árvore de Decisão é um grafo hierarquizado (árvore!) em que:
    - Cada ramo representa a seleção entre um conjunto de alternativas;
    - Cada folha representa uma decisão;
    - Cada nodo interno testa um atributo do dataset;
    - Cada ramo identifica um valor (ou conjunto de valores) do nodo testado;
    - Cada folha representa uma decisão;



ALMOÇAR	ONDE	DECISÃO
12h30	Cantina	NÃO
13h15	Cantina	NÃO
13h10	Restaurante	SIM
11h00	Restaurante	NÃO
13:30	Cantina	NÃO

Modelos de Decisão

- Paradigmas de criação de modelos de decisão:
    - Top-down:
        - O modelo é construído a partir do conhecimento de especialistas;
        - O "todo" é dividido em "partes";
    - Bottom-up:
        - O modelo é construído pela identificação de relações entre os atributos do dataset;
        - O modelo é induzido por "generalização" dos dados;
- Árvores de Decisão seguem o Paradigma Bottom-up:
    - Toda a informação sobre cada item de dados (ou objeto) deve estar definido numa coleção fixa e finita de atributos;
    - Deste modo, objetos distintos não podem requerer coleções distintas de atributos;
    - Quando o conjunto dos níveis de decisão é conhecido a priori, a construção do modelo segue um paradigma de aprendizagem supervisionado;
    - Quando o conjunto dos níveis de decisão é calculado pelo modelo, a sua construção segue um paradigma de aprendizagem não supervisionado;
    - Os níveis de decisão podem ser de 2 tipos:
        - Contínuos: problemas de regressão;
            - O atributo de decisão representa uma sequência, conjunto ou intervalos de possíveis valores;
            - As folhas da árvore de decisão identificam intervalos ou conjuntos de valores;
        - Discretos: problemas de classificação;
            - O atributo de decisão representa uma categoria ou uma classe;
            - Os valores representados nas folhas da árvore de decisão são as categorias ou classes;
    - Quantidade de objetos >> níveis de decisão;

Dada uma árvore de decisão (treinada), o processo de decisão desenvolve-se do seguinte modo:

1. Começar no nodo correspondente ao atributo "raiz";
2. Identificar o valor do atributo;
3. Seguir pelo ramo correspondente ao valor identificado,
4. Alcançar o nodo relativo ao ramo percorrido;
5. Voltar a 2. até que o nodo seja uma "folha";
6. O nodo alcançado indica a decisão para o problema.

Uma Árvore de Decisão pode ser utilizada para fazer **classificação**:

- Decidir sobre se ou onde almoçar: classificação binária (SIM/NÃO)
- Prever quem sobreviveu ao acidente do Titanic: classificação binária (SIM/NÃO)
- Classificar um conjunto de imagens: classificação múltipla (laranja, kiwi, romã,…)

Uma Árvore de Decisão pode ser utilizada para fazer **regressão**:

- Regressão linear, polinomial, múltipla, entre outras;
- Prever o preço do petróleo/gás/combustíveis: escala contínua ou real, em € ou $
- Estimar a temperatura para o dia de amanhã: escala continua, em ºC ou ºF

Construção de um Modelo de Árvores de Decisão

- A construção de um modelo baseado em Árvores de Decisão desenvolve-se através de:
    - Observação de exemplos (objetos);
    - Generalização por indução;
    - Criação do modelo;
    - Apresentação de um problema;
    - Obtenção da resolução do problema (previsão);
- Algoritmo ID3: Iterative Dichotomiser 3
    - Desenvolvido por Ross Quinlan;
    - Constrói uma árvore de decisão, a partir da raiz até às folhas;
    - Principal problema:
        - "Qual o melhor atributo para ser a raiz da árvore de decisão?"
- Entropia:
    - Noção:
        - Entropia é uma medida da incerteza associada a um conjunto de objetos;
        - A entropia identifica o grau de desorganização dos dados;
    - Dada uma coleção S de dados contendo exemplos positivos e negativos de um determinado conceito, a entropia de S é definida por:
        - $Entropia\ S \equiv -(p+)\ (\log_2 p+)\ -(p-)\ (\log_2 p-)$
    - onde $p+$ e $p-$ representam, respetivamente, a proporção de exemplos positivos e de exemplos negativos em S;
    - Considere a função de entropia definida no gráfico relativa a um atributo binário, sendo que:
        - $p+ \in [0,1]$
        - $p- \equiv (1 - p+) \in [0,1]$

- A entropia é 0 (zero) quando todos os objetos de S são do mesmo valor;
  - Se todos os objetos forem positivos:
    - $p+ = 1$
    - $p- = 1 - 1 = 0$
  - então: •
    - $Entropia\ (S) = -1 \log2\ 1 - 0 \log2\ 0 = -1 \times 0 - 0 \times \log2\ 0 = 0$
- Quando a entropia é 1 (um), S tem igual proporção de objetos positivos e negativos;
  - Sendo:
    - $p+ = 0{,}5$
    - $p- = 1 - 0{,}5 = 0{,}5$
  - então:
    - $Entropia\ S = -0{,}5 \log2\ 0{,}5 - 0{,}5 \log2\ 0{,}5 = -0{,}5 \times -1 - 0{,}5 \times -1 = 0{,}5 + 0{,}5 = 1$

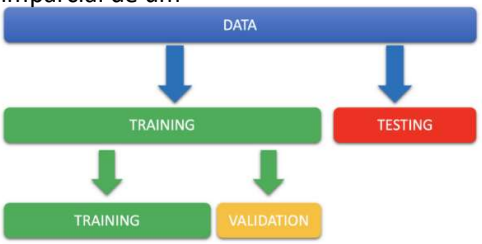
- Ganho de Informação:
  - Esta métrica mede a redução esperada na entropia;
  - Decisão sobre qual o atributo que será selecionado para ser nodo;
  - O atributo com a maior redução de entropia é a melhor escolha para ser nodo; (para reduzir a profundidade da árvore)
  - $Ganho(S, A)$ de um atributo A relativamente a uma coleção S define-se como:
    - $Ganho\ S, A\ = \ Entropiaoriginal\ (S) - \ Entropiarelativa\ (S) = Entropia\ (S) - \sum_{v \in valores\ A} \frac{|Sv|}{|S|} \times Entropia\ Sv$ (ver exemplo página 21 ppt semana 6)
  - Sendo:
    - S cada valor v de todos os valores possíveis do atributo A;
    - Sv subconjunto de S para o qual o atributo A tem o valor v;
    - Sv quantidade de objetos em Sv;
    - S quantidade de objetos em S;

- Algoritmo C4.5
  - Extensão do algoritmo ID3;
  - Desenvolvido, igualmente, por Ross Quinlan;
  - Melhorias face ao ID3:
  - Manipula atributos contínuos e discretos:
    - Para lidar com atributos contínuos, é definido um limite (threshold) usado para dividir os valores **acima** e **abaixo** do limite;
  - Lida com missing values:
    - Assinala os missing values que não serão usados nos cálculos de **ganho** e **entropia**;
  - Permite a atribuição de pesos aos atributos;
  - Permite fazer a poda da árvore:
    - Retrocede 1 iteração na árvore e **remove ramos** que contribuem menos ou não contribuem para a definição da solução, **substituindo-os por folhas**;

        - Tree Pruning (poda da árvore)
            - Porque uma Árvore de Decisão pode resultar num modelo de decisão “demasiado” adaptado aos dados de treino;
            - Cada folha pode representar um caso ou conjunto de casos muito específicos;
- Algoritmo J48
    - Implementação do algoritmo C4.5 open source em JAVA na plataforma WEKA;
    - WEKA: Waikato Environment for Knowledge Analysis;
- Algoritmo CART: Classification and Regression Tree
    - Introduzido por Breiman, praticamente em paralelo com o ID3 de Ross Quinlan;
    - Um mesmo algoritmo que partilha as semelhanças de modelos de classificação e de regressão;
- ▪ Algoritmo CHAID: Chi-square Automatic Interaction Detection
    - Opera a separação dos dados em modo multi-nível, enquanto que o CART usa modos binários para essa divisão;
    - Adequado para grandes datasets;
    - Frequentemente utilizado em estudos de marketing para segmentação de mercados;
- **Pontos fortes:**
    - Facilmente compreensíveis;
    - Podem ser convertidas para regras;
    - Manipulam missing values;
    - Configuração simples (não tem demasiados parâmetros de configuração);
- **Pontos fracos:**
    - Inadequadas para problemas caracterizados por muitas interações entre os atributos;
    - Falta de poder expressivo;
    - Não isenta a réplicas de subárvores;

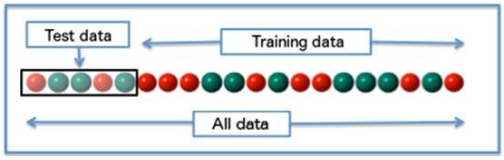
Avaliação de Modelos

- Após a criação (treino) de um modelo usando uma técnica de aprendizagem (*machine learning*), é necessário avaliar o seu desempenho;
- A medição do desempenho de um modelo é feita com dados não apresentados durante o treino;
- Dados de treino:
    - Conjunto de dados usado para ajustar o modelo;
- Dados de validação:
    - Conjunto de dados usado para fornecer uma avaliação imparcial de um ajuste do modelo, no conjunto de dados de treino;
- Dados de teste:
    - Conjunto de dados usado para fornecer uma avaliação imparcial de um modelo final ajustado ao conjunto de dados de treino.



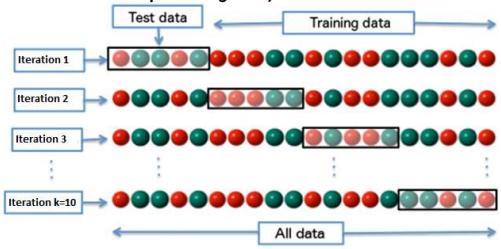
Hold-Out Validation

- Método de particionamento de dados;
- Divide o conjunto de dados em dados de treino e dados de teste;
- Separa-se uma parte *(hold-out)* do conjunto de dados para treino/teste (80/20; 75/25;…)



Cross Validation

- Método de validação por cruzamento de dados;
- Consiste em dividir o conjunto de dados em k partes (k *folds*);
  - A cada iteração, o método utiliza k-1 partes (*folds*) para treino e 1 parte (*fold*) para teste;
  - O processo repete-se durante k vezes;
- O erro final é dado pela média dos valores parciais dos erros.

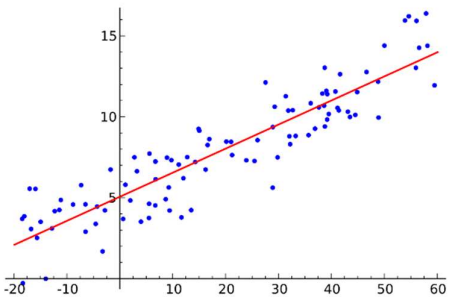


- **Leave-one-out (K=N)**:
  - Caso particular em que o número de casos N é igual ao número de *folds* k;
- Se o *dataset* for grande, um valor pequeno para k pode ser suficiente, uma vez que teremos uma quantidade grande de dados para treino;
- Se o *dataset* for pequeno, um valor grande de k ≈ N pode revelar-se mais adequado para maximizar a quantidade de dados para treino;
- Quanto maior a quantidade de *folds*, melhor a estimativa do erro, mais baixo será o viés(*bias*) e menor será o sobreajuste (*overfitting*);
- De facto, o valor de k depende do valor de N!

Técnicas de Regressão

Regressão

- Quão bem uma determinada variável independente prevê outra variável dependente?
- A regressão é um procedimento estatístico que determina a equação para a linha reta que melhor se ajusta a um conjunto específico de dados.

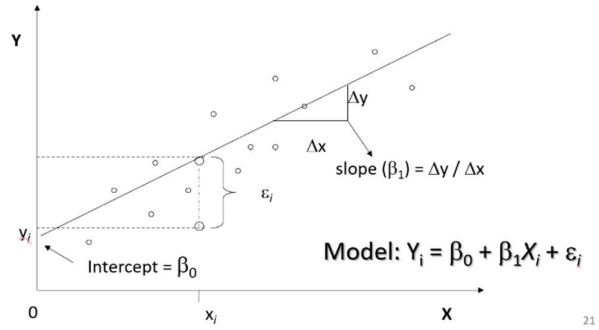


Regressão Linear

- Tem como objetivo prever o valor de um resultado, Y, com base no valor de uma variável de previsão, X;
    - Como "encaixar" uma linha reta num conjunto de dados;
    - Usar esta linha para estimar a resolução de problemas.

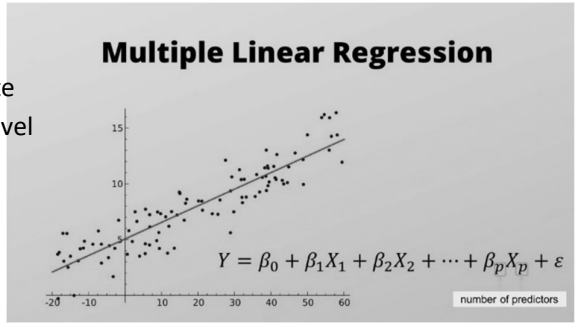
Como funciona:

- O método dos mínimos quadrados minimiza a soma dos erros ao quadrado:
    - y_i: valor verdadeiro
    - $f(x_i, \beta)$: valor previsto / linha ajustada
- O resíduo para uma observação é a diferença entre a observação (valor y) e a linha ajustada:
    - $r_i = y_i - f(x_i, \beta)$
- O método dos mínimos quadrados procura os parâmetros ótimos, minimizando a soma S:
    - $S = \sum_{i=1}^{n} r_i^2$



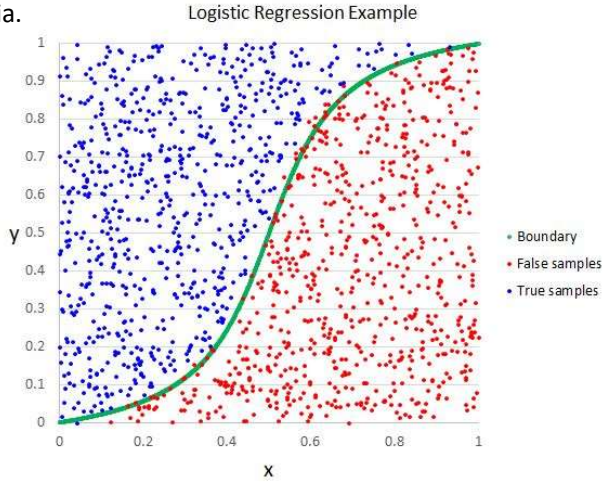
Regressão Linear Múltipla

- A regressão múltipla é usada para determinar o efeito de diversas variáveis independentes, $x_1, x_2, x_3, \ldots$ numa variável dependente, y;
- As diferentes variáveis x_i são combinadas de forma linear e cada uma tem seu próprio coeficiente de regressão:
    - $y = a_1 \cdot x_1 + a_2 \cdot x_2 + \cdots + a_n \cdot x_n + b + \varepsilon$
- Os parâmetros a_i refletem a contribuição independente de cada variável independente x_i, para o valor da variável dependente, y.



Regressão Logística

- A diferença essencial entre regressão linear e **regressão logística** é que esta é usada quando a variável dependente é de natureza binária.
- Em contraste, a **regressão linear** é usada quando a variável dependente é contínua e a natureza da linha de regressão é linear.
- A Regressão Logística é uma técnica de **classificação**:
    - Empréstimo (SIM/NÃO)
    - Diagnóstico (São/Doente)
    - Vinho (Branco/Rosé/Tinto)

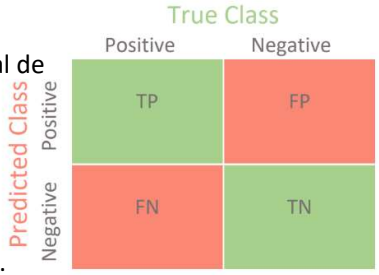


Métricas de Qualidade

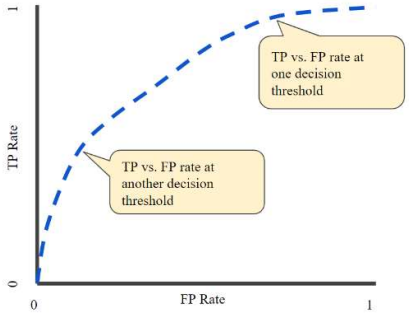
- Porquê métricas de qualidade?
  - Para avaliar o desempenho do modelo.
- As métricas são usadas para monitorizar e medir o desempenho de um modelo:
  - Erro Médio Absoluto (*Mean Absolute Error* - MAE)
  - Erro Médio Quadrado (*Mean Squared Error* - MSE)
  - Precisão (*Precision*)
  - F1-Score,
  - entre outras…
- No entanto, depende do problema em mãos:
  - É um problema de classificação?
  - De regressão?
  - Séries temporais?

Modelos de Qualificação

- Matrizes de Confusão
  - Tabela utilizada para descrever o desempenho de um modelo de classificação.
- *Accuracy*
  - Quantidade de previsões corretas dividido pela quantidade total de observações:
    - $Accuracy = \frac{TP+TN}{TP+TN+FP+F}$



- Precisão (*Precision aka Sensitivity*)
  - É uma medida da exatidão;
  - Determina a proporção de itens relevantes entre todos os itens:
    - $Precision = \frac{TP}{TP+FP}$
- *Recall* (*aka Specificity*)
  - É uma medida de completude;
  - Determina a proporção de itens relevantes obtidos:
    - $Precision = \frac{TP}{TP+FN}$
- ROC curve:
  - A curva *Receiver Operating Characteristics* (ROC) encontra o desempenho de um modelo de classificação em diferentes limites de classificação;
  - Reduzindo o patamar (*threshold*) de classificação, são classificados mais itens como positivos, aumentando os falsos positivos e os verdadeiros positivos.
- AUC curve:
  - A Area Under the Curve (AUC) mede a área abaixo da curva ROC;



- Mede quão bem as previsões são classificadas, em vez de avaliar os seus valores absolutos (varia de 0 a 1);
- Um modelo cujas previsões estão 100% erradas tem uma AUC de 0; aquele cujas previsões estão 100% corretas tem uma AUC de 1.

Modelos de Regressão

- Erro Médio Absoluto (*Mean Absolute Error* - MAE)
  - Mede a magnitude média dos erros num conjunto de previsões (não considera a direção):
    - $MAE = \frac{1}{n}\sum_{j=1}^{n}\left|y_j - \hat{y}_j\right|$
      em que n é a quantidade de observações, y_j e $\hat{y}_j$ são, respetivamente, a observação atual e o valor previsto
- Erro Médio Quadrado (*Mean Squared Error* - MSE)
  - Consiste no cálculo da média das diferenças, ao quadrado, entre os erros num conjunto de previsões (não considera a direção):
    - $MSE = \frac{1}{n}\sum_{j=1}^{n}\left(y_j - \hat{y}_j\right)^2$
      em que n é a quantidade de observações, y_j e $\hat{y}_j$ são, respetivamente, a observação atual e o valor previsto.
- Raiz Quadrada do Erro Médio Quadrado (Root *Mean Squared Error* - RMSE)
  - Consiste no cálculo raiz quadrada da média das diferenças, ao quadrado, entre os erros num conjunto de previsões (não considera a direção):
    - $RMSE = \frac{1}{n}\sqrt{\sum_{j=1}^{n}\left(y_j - \hat{y}_j\right)^2}$
      em que n é a quantidade de observações, y_j e $\hat{y}_j$ são, respetivamente, a observação atual e o valor previsto.

- Três das métricas mais comuns usadas para medir a precisão de variáveis contínuas;
- Todas expressam o erro médio de previsão do modelo (valores mais baixos são melhores);
- Todos variam de 0 a ∞ e são indiferentes à direção dos erros;
- MAE e RMSE expressam o erro de previsão na mesma unidade da variável de interesse;
- MSE e RMSE, ao elevar o erro ao quadrado, dão um peso relativamente alto para erros grandes;

- MSE e RMSE são mais úteis quando grandes erros são especialmente indesejáveis.

#	Error	|Error|	$Error^2$
1	1	1	1
2	-1	1	1
3	3	3	9
4	3	3	9

MAE	MSE	RMSE
2	5	2.24

#	Error	|Error|	$Error^2$
1	0	0	0
2	0	0	0
3	0	0	0
4	10	10	100

MAE	MSE	RMSE
2.5	25	5

Segmentação (Clustering)

A segmentação/clustering de dados é um processo através do qual se **particiona** um conjunto de **dados** em **segmentos/ clusters** de menor dimensão, que agrupam conjuntos de dados similares.

Um segmento/cluster é uma coleção de valores/objetos que:

- São **similares** entre si, dentro de um **mesmo segmento**;
- São **diferentes** dos valores/ objetos de **outros segmentos**

Medidas de similaridade:

- distância **Euclidiana** ou de **Manhattan**, para **atributos contínuos**;
- coeficiente de **Jacqard**, para atributos **discretos**/**binários**;
- etc.

Aplicações da Segmentação

- Como uma ferramenta *per si*, para pesquisar "dentro" dos dados, sobre a distribuição dos seus valores;
- Como uma das fases do **pré-processamento**, por forma a **organizar** os dados a submeter a outros algoritmos;
- Em problemas de **reconhecimento de padrões** (*pattern matching*);
- No **processamento de imagem**;
- Na pesquisa em mercados económicos;
- etc.

Utilização da Segmentação

A deteção de segmentos é útil:

- quando se suspeita da existência de agrupamentos "naturais", que podem representar grupos de clientes, de produtos ou de bens que partilhem (muita) informação;
- quando existam muitos padrões diferentes nos dados, dificultando a tarefa de identificar um determinado padrão;
- a criação de segmentos semelhantes **reduz** a complexidade do problema.

Exemplos de aplicação

Marketing

- ajuda na descoberta de grupos de clientes para desenvolver estratégias de comercialização;

Previsão de sismos:

- a observação de epicentros sismológicos permite identificar segmentos comuns de falhas continentais;

Seguradoras:

- identificação de grupos de utentes que representam maior risco de contratação;

Banca:

- identificação de categorias de clientes (económicas, sociais, etc.)

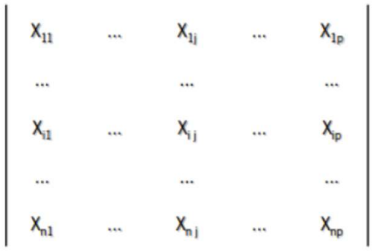
Cluster

Dado um conjunto de objetos, coloque-os em grupos de forma que os objetos em um grupo sejam semelhantes (ou relacionados) entre si e diferentes (ou não relacionados) dos objetos em outros grupos.

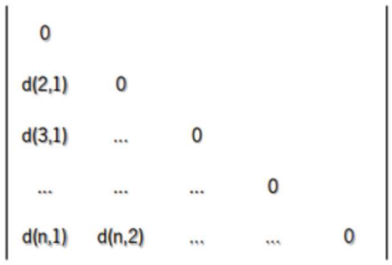
- **Intra-Cluster** → distances are **minimized**
- **Inter-Cluster** → distances are **maximized**

Tipos de dados para análise

- **Matriz de dados**
  - Representa 'n' objetos com 'p' atributos;



- **Matriz de distâncias**
  - Mede a proximidade entre pares de objetos;
  - Tanto mais similar quanto mais próximo de 0 (zero)



- **Atributos contínuos**
  - **normalizar os dados**: evita que os resultados dependam das unidades de medida;
  - normalmente, utilizam-se **medidas de distancia** para calcular a proximidade (similaridade) entre objetos;
    - Distância Euclidiana: é a medida de distancia geométrica no espaço (a mais usada):

$$d(x,y) = \sqrt{(x1 - x2)^2 + (y1 - y2)^2} \quad \text{(para 2 dimensões)}$$

    - distância Manhattan: mede a distância pela diferença entre os pontos (função não quadrática):

$$d(x,y) = |x1 - x2| + |y1 - y2| \qquad \text{(para 2 dimensões)}$$

    - distância Minkowski: mede o peso progressivo em função da distância dos pontos.

- **<u>Atributos binários</u>**:
  - são classificados em:
    - **Simétricos**: significado de ser 0 é o <u>mesmo</u> de ser 1;
    - **Assimétricos**: significado de ser 0 é <u>diferente</u> de ser 1;
  - a similaridade calculada com base em atributos simétricos é designada **similaridade invariante**; no caso oposto diz-se **similaridade não-invariante**;
  - tabela de contingência para os dados binários:
    - coeficiente simples (simétricos):

$$d(i,j) = \frac{b+c}{a+b+c+d}$$

	Sexo	Febre	Tosse	Dor
João	M	Sim	Não	Não
Maria	F	Sim	Não	Sim
José	M	Sim	Sim	Não

João \ Maria	Sexo	M	F	Soma
	M	a = 0	b = 1	a+b
	F	c = 0	d = 0	c+d
	Soma	a+c	b+d	

    - coeficiente Jaccard (assimétricos):

$$d(i,j) = \frac{b+c}{a+b+c}$$

	Sexo	Febre	Tosse	Dor
João	M	Sim	Não	Não
Maria	F	Sim	Não	Sim
José	M	Sim	Sim	Não

João \ Maria	F/T/D	S	N	Soma
	S	a = 1	b = 0	a+b
	N	c = 1	d = 1	c+d
	Soma	a+c	b+d	

- **Atributos nominais:**
  - trata-se de uma generalização dos atributos binários, em que os dados podem assumir mais do que 2 valores;
  - Método 1:
    - matching simples;

$$d(i,j) = \frac{nº\ variáveis - nº\ matches}{nº\ variáveis}$$

  - Método 2:
    - Utilizar variáveis binárias;
    - Criar uma variável binária para cada valor nominal.

- **Atributos ordinais:**
  - a ordem é relevante:
    - primeiro, segundo, terceiro, ..., penúltimo, último;
  - podem ser tratados como atributos contínuos, sendo que a ordenação dos valores define uma classificação:
    - 1, 2, 3, ..., Máx;
  - as similaridades devem ser calculadas utilizando os mesmos métodos que para os atributos contínuos.

- **Atributos mistos:**
  - o conjunto de dados pode conter diversos tipos de atributos;
  - tipicamente, utiliza-se uma função pesada para ponderar e medir os efeitos de cada atributo

Principais métodos de segmentação

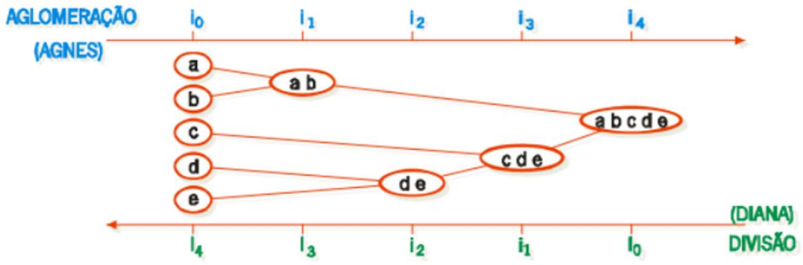
- **Particionamento**
  - Criar várias partições e adotar um critério de avaliação
  - Uma divisão de objetos de dados em subconjuntos não sobrepostos (clusters.)
- **Hierarquização**
  - decompor hierarquicamente o conjunto de dados;
  - Produz um conjunto de clusters aninhados organizados como uma árvore hierárquica.
- **Outros**
  - Baseados na Densidade:
    - aumentar o segmento enquanto a densidade de pontos estiver num determinado limite (utilizam-se funções de conectividade e densidade);
  - Baseados no Modelo:
    - criar modelos hipotéticos para cada segmento e testar a capacidade de adequação de cada ponto ao segmento.

Algoritmos de Particionamento

- Particionar um conjunto de dados 'D' contendo 'n' objetos num conjunto de 'k' segmentos/clusters;
- Sendo dado 'k', particionar 'D' em 'k' segmentos de forma a otimizar o critério de particionamento:
    - Ótimo Global: enumeração exaustiva de todas as partições
    - Métodos heurísticos:
        - **k-means**: cada segmento é representado pelo **centro** do segmento (centroid);
        - **k-medoids**: cada segmento é representado por **um dos elementos** do segmento (medoid)
- **Método k-means** Sendo dado 'k' (número de segmentos), seguir os 4 passos:
    - 1. Dividir os objetos em 'k' subconjuntos não vazios;
    - 2. Calcular o centro de cada segmento (centroid);
    - 3. Atribuir cada objeto ao centroid mais próximo;
    - 4. Voltar ao ponto 2.; parar quando não houver mais possibilidades de atribuição.
    - <u>Vantagens</u>
        - Relativamente eficiente: sendo 'n' o número de objetos, 'k' o número de segmentos e 'i' o número de iterações, normalmente acontece k,i << n;
        - Termina com ótimos locais.
    - <u>Desvantagens</u>
        - Aplicável, apenas, quando é possível calcular a média;
        - É necessário identificar o número de segmentos a priori;
        - Incapacidade de lidar com ruído nos dados;
        - Inadequado para determinar segmentos côncavos.
- **Método k-medoids**
    - Medoids são objetos **representativos** do conjunto de dados;
    - Inicia-se com um conjunto de medoids que, iterativamente, vão sendo substituidos por outros não-medoids desde que a **distância** do segmento resultante seja **melhorada**.
    - <u>Vantagens e Desvantagens</u>
        - É mais **robusto** do que o método k-means na presença de **dados ruidosos**, uma vez que os objetos selecionados são **menos influenciáveis** por valores extremos do que a **média** (mean);
        - Produz **bons resultados** para conjuntos de dados de **pequenas dimensões**;
        - Não se comporta tão bem quando se pretende a sua aplicação em conjuntos de dados de grandes dimensões

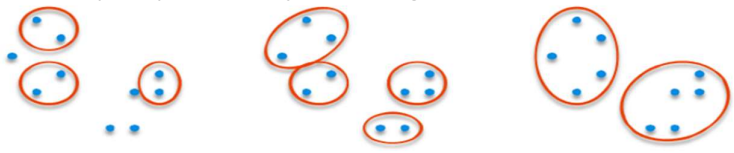
Algoritmos de Hierarquização

- Utilizam a matriz de distâncias como critério de segmentação;
- Os dados são agrupados em árvores de segmentos;
- Não requerem a definição do número de segmentos a procurar;
- Exigem a definição de uma condição de paragem:
    - quantidade de segmentos;
    - distância mínima entre objetos;
    - etc.
- Existem dois tipos de algoritmos de hierarquização:
    - Aglomeração: estratégia bottom-up;
    - Divisão: estratégia top-down
- Aglomeração:
    - Inicia-se formando segmentos com um objeto, para todos os objetos;
    - Prossegue juntando segmentos atómicos em segmentos cada vez mais amplos.
- Divisão:
    - Inicia-se com todos os objetos em um só segmento que se vai subdividindo em segmentos de menor dimensão;
    - Aplicação prática muito rara



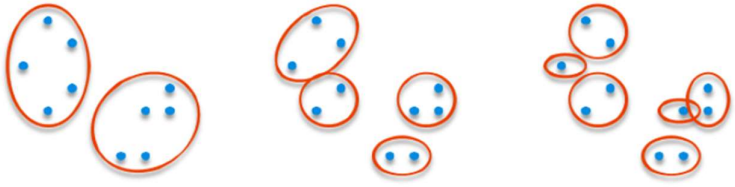
AGNES: Agglomerative Nesting

Iterativamente, **vai juntando objetos** que apresentam menores valores de dissemelhança: os conjuntos C1 e C2 são juntos se os objetos de C1 e de C2 produzem o menor valor de distância Euclidiana entre quaisquer dois objetos de segmentos distintos.



DIANA: Divisive Analysis

Iterativamente e **partindo de um segmento composto por todos os objetos**, **dividir** em **segmentos menores** que **maximizam** a **distância** euclidiana entre **objetos vizinhos de segmentos diferentes.**



Vantagens e Desvantagens

- Dificuldades com o aumento de atributos ou de objetos: à medida que aumentam os objetos a agrupar, aumenta o tempo necessário para procurar tais grupos;
- Não é necessário especificar o número de segmentos 'k'; basta "cortar" a árvore no nível 'k-1';
- Produz melhores resultados do que os algoritmos k-means;
- Uma hierarquia traduz alguma organização dos segmentos, ao contrário de um simples conjunto de segmentos.

Outros Algoritmos

- **BIRCH**: Balanced Iterative Reducing and Clustering using Hierarchies;
    - Usa árvores com características sobre os segmentos e ajusta, iterativamente, a qualidade dos segmentos;
    - É construída uma árvore que captura informação necessária para realizar as operações de segmentação:
        - Clustering Feature: contém informação sobre o **segmento**;
        - Clustering Feature Tree: contém informação sobre a **organização arbórea da hierarquia**.
- **CURe**: Clustering Using Representatives
    - Seleciona **pontos dispersos do segmento** e vai **reduzindo o tamanho** do segmento em direção ao seu **centro**;
    - Usa **múltiplos pontos** representativos;
    - Em cada iteração, **dois segmentos com o par de pontos representativos mais próximos são juntos.**
- **DBSCAN**: Density Based Spatial Clustering of Applications with Noise;
    - Algoritmo baseado no cálculo de valores de densidade e de conectividade locais;
    - Características assinaláveis:
        - capaz de descobrir segmentos de **formas não regulares**;
        - capaz de lidar com **ruído nos dados**;
        - algoritmo de **um só passo** (scan);
        - obriga à **definição de parâmetros de densidade como condição de paragem.**

Redes Neuronais Artificiais

- Uma **Rede Neuronal Artificial** (RNA) é um sistema computacional de base conexionista para a resolução de problemas.
- Uma RNA é concebida com base num **modelo** simplificado do **sistema nervoso central** dos seres humanos.
- Uma RNA é definida por uma estrutura interligada de unidades computacionais, designadas **neurónios**, com capacidade de **aprendizagem**.

Neurónio

- **Unidade computacional** de composição da RNA.
- **Identificado** pela sua **posição** na rede.
- Caracterizado pelo **valor do estado**

Axónio

- **Via de comunicação** entre os neurónios.
- Pode **ligar qualquer neurónio**, incluindo o próprio.
- As ligações podem **variar** ao longo do **tempo**.
- A informação circula em **um só sentido**.

Sinapses

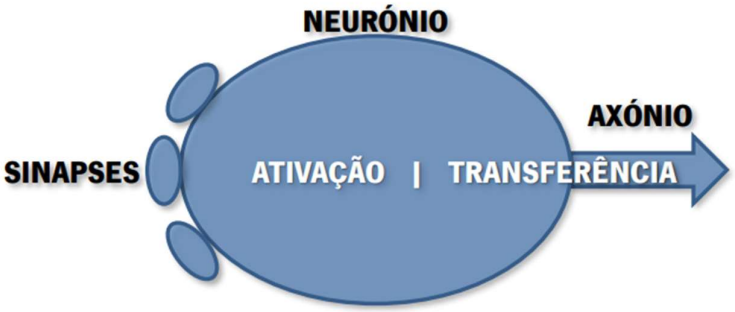
- **Ponto de ligação** entre axónios e neurónios.
- O **valor da sinapse** determina o **peso** (importância) do sinal a entrar no neurónio: excitativo, inibidor ou nulo.
- A **variação no tempo determina a aprendizagem** da RNA.

Ativação

- O valor de ativação é representado por um **único valor**.
- O valor de ativação **varia com o tempo**.
- A gama de valores varia com o modelo adotado (normalmente está dependente das entradas e de algum efeito de memória).

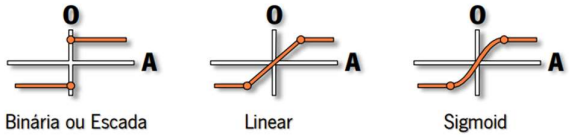
Transferência

- O valor de transferência de um neurónio determina **o valor** que é **colocado na saída** (transferido através do axónio).
- É calculado como uma função do valor de ativação (eventualmente com algum efeito de memória).



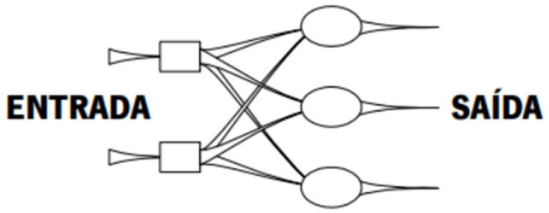
Tarefas dos neurónios

- Cálculo do valor de saída (output = Oi), função do valor de ativação, por uma função de transferência (f T): Oi = fT (Ai)

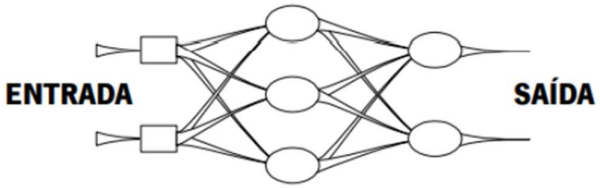


- Cálculo do valor de ativação (Aj). ▪ Varia no tempo com o seu próprio valor e o de outras entradas (wi ; I): Aj = F(Aj−1; Ij ;∑wi,j ∗ Oi)
- Aprendizagem: regras de modificação dos pesos (wi).

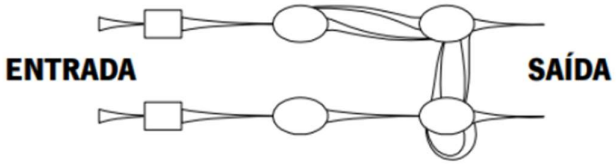
▼ **Feedforward, Single layer**



▼ **Feedforward, multi-layer**

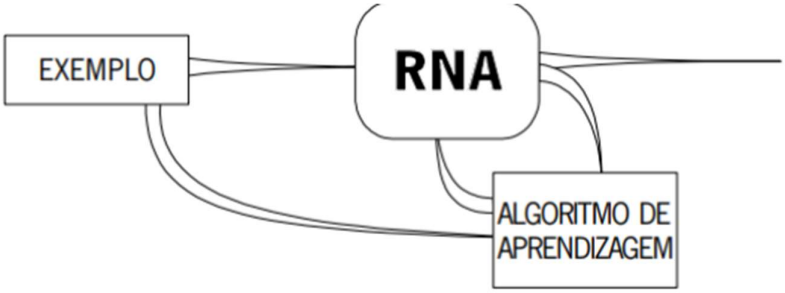


▼ **Recurrent**

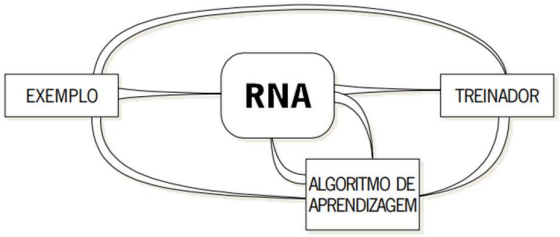


Paradigmas de Aprendizagem

- **Sem supervisão**: (p.ex., quando dois neurónios adjacentes têm variações da ativação no mesmo sentido, então o peso da ligação deve ser progressivamente aumentado.)



- **Com supervisão**: (p.ex., os ajustes nos pesos das ligações são efetuados por forma a minimizar o erro produzido pelos resultados da RNA.)
- **De reforço**: o exemplo contém, apenas, uma indicação sobre a correção do resultado.



Regras de Aprendizagem

- O treino de uma RNA corresponde à **aplicação de regras de aprendizagem**, por forma a fazer **variar os pesos das ligações** (sinapses);
- Regras de aprendizagem mais comuns:
    - **Hebbian Learning Rule**
        - Desenvolvida por Donald Hebb em 1949 para o treino não supervisionado de RNAs;
        - Se dois neurónios adjacentes sofrem variações no mesmo sentido, o peso da ligação deve aumentar;
        - Se as variações acontecem em sentido oposto, o peso da ligação deve diminuir;
        - Não havendo variação, o peso deve manter-se inalterado;
        - Os pesos são inicializados a zero;
    - **Perceptron Learning Rule**
        - Desenvolvida para aprendizagem supervisionada;
        - Os pesos iniciais são atribuídos aleatoriamente;
        - Os inputs são processados pela rede e comparados com o output desejado;
        - Calcula-se o erro produzido pela rede na forma:
        - A função de alteração dos pesos usa este erro para calcular a atualização dos seus valores;
    - **Widrow-Hoff Learning Rule**
        - Desenvolvida por Bernard Widrow e Marcian Hoff;
        - A principal diferença para Perceptron Learning é a de que é usado um sinal linear e não binário para calculo do erro e consequente atualização dos pesos;
    - **Competitive Learning Rule**
        - Desenvolvida para aprendizagem não supervisionada;
        - Os neurónios de output competem entre si para representarem o padrão do input;

        - O neurónio com maior output para um dado input é declarado vencedor, sendo o único a alterar os pesos;
    - **Correlation Learning Rule**
    - **Outstar Learning Rule (Grossberg Rule)**

Especificações

- Quantidade de neurónios:
    - na camada de entrada;
    - na camada de saída;
    - nas camadas intermédias;
- Níveis (ou camadas) da RNA;
- Ligações entre neurónios;
- Topologia das ligações;
- Esquema de atribuição e atualização dos pesos;
- Funções:
    - de transferência;
    - de ativação;
    - de aprendizagem;
- Métodos de treino.

pdf daa páginas 55-64