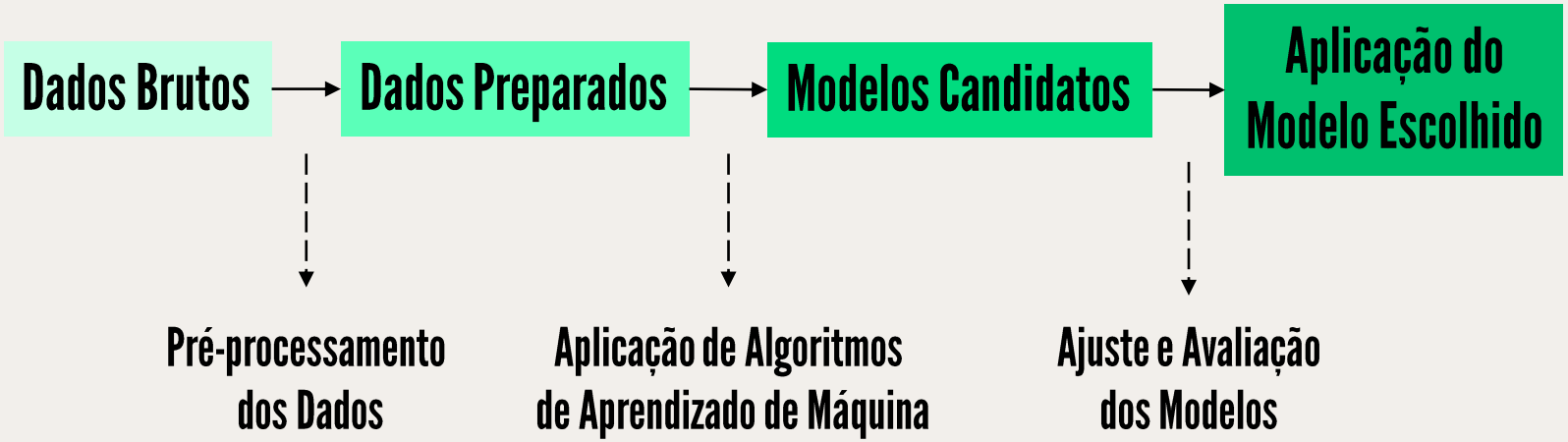
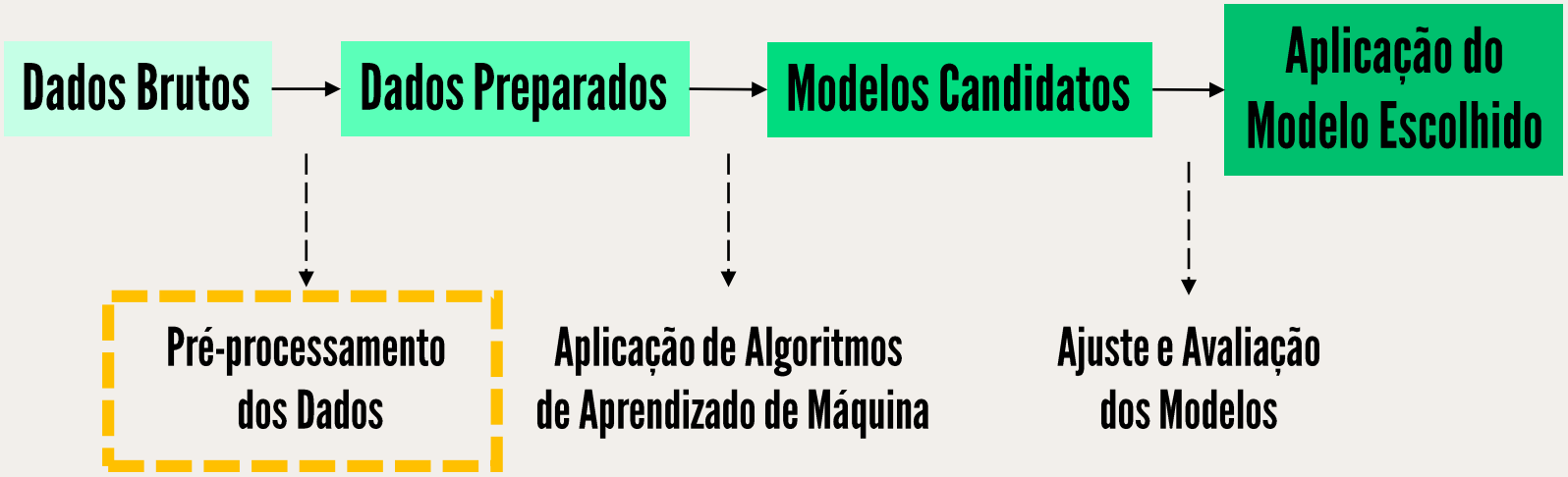
PRÉ-PROCESSAMENTO DE DADOS

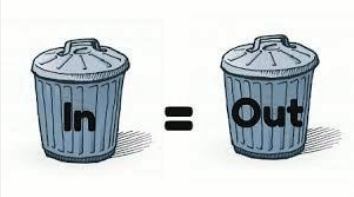
PROF. LETÍCIA RAPOSO
profleticiaraposo@gmaill.com

O PROCESSO DO APRENDIZADO DE MÁQUINA



O PROCESSO DO APRENDIZADO DE MÁQUINA





A qualidade do conhecimento extraído é amplamente determinada pela qualidade dos dados fornecidos como entrada.

TÉCNICAS DE PROCESSAMENTO DE DADOS

Utilizadas para *melhorar a qualidade dos dados* por meio da eliminação ou minimização de problemas, como:

- Valores incorretos, inconsistentes, duplicados ou ausentes;
- Atributos (variáveis explicativas) podem ser independentes ou relacionados;
- Os conjuntos de dados podem apresentar poucos ou muitos objetos, que, por sua vez, podem ter um número pequeno ou elevado de atributos.

Minimizar ou eliminar problemas existentes em um conjunto de dados/ tornar os dados mais adequados para uma posterior utilização.

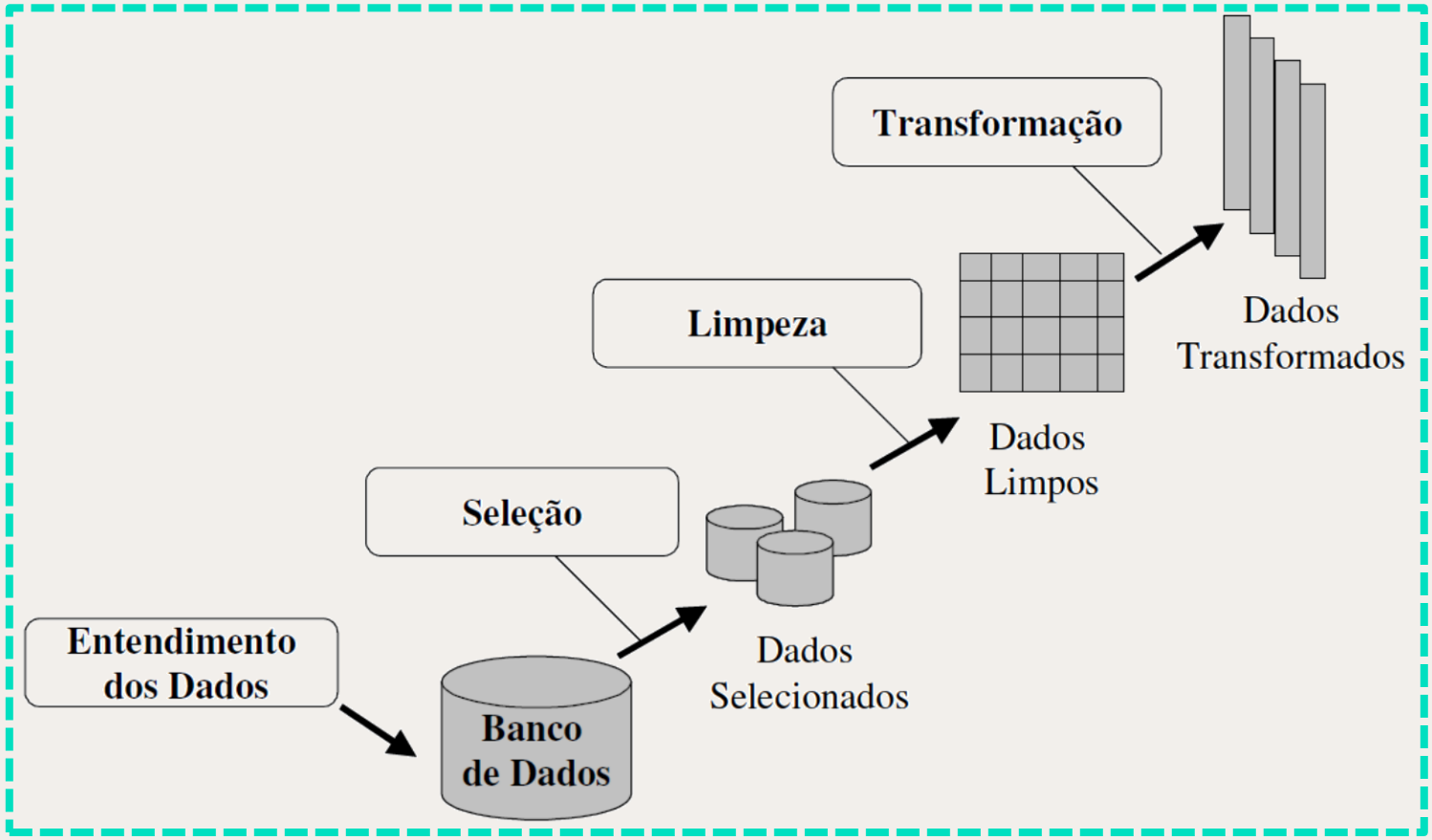
TAREFAS DE PRÉ-PROCESSAMENTO

- Eliminação manual de atributos;
- Integração de dados;
- Amostragem de dados;
- Dados desbalanceados;
- Limpeza de dados;
- Transformação de dados;
- Redução de dimensionalidade.

Processo semiautomático:
Depende da capacidade da pessoa que a conduz em identificar os problemas presentes nos dados e utilizar os métodos mais apropriados para solucionar cada um dos problemas.

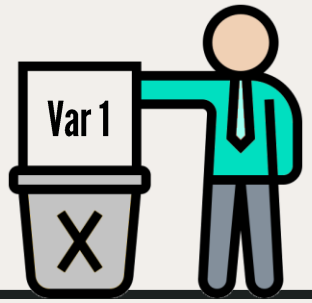
https://becominghuman.ai/data-preprocessing-a-basic-guideline-c0842b7883fa

TAREFAS DE PRÉ-PROCESSAMENTO



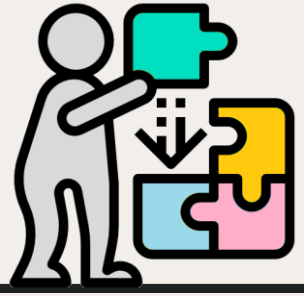
Adaptado de Rita de Cássia David das Neves, 2003

ELIMINAÇÃO MANUAL DE ATRIBUTOS



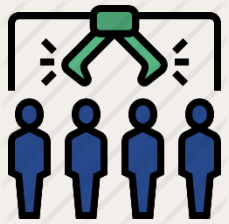
- Quando um atributo claramente *não contribui para a estimativa do valor do atributo alvo*.
  - Ex: nome, identificação de paciente.
- Quando um atributo possui o *mesmo valor para todos os objetos*.
  - Não contém informação que consiga distinguir os objetos.
- Um atributo não precisa ter exatamente o mesmo valor para todos os objetos para ser irrelevante → seleção de atributos.

INTEGRAÇÃO DOS DADOS



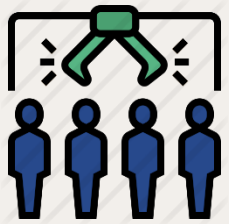
- Etapa a ser realizada *antes da aplicação de uma técnica de AM*.
- O(s) atributo(s) utilizado(s) para combinação deve(m) ter um *valor único* para cada objeto.
- Dificuldades:
  - Atributos correspondentes podem ter nomes diferentes em diferentes bases de dados;
  - Os dados a serem integrados podem ter sido atualizados em momentos diferentes.

AMOSTRAGEM DOS DADOS



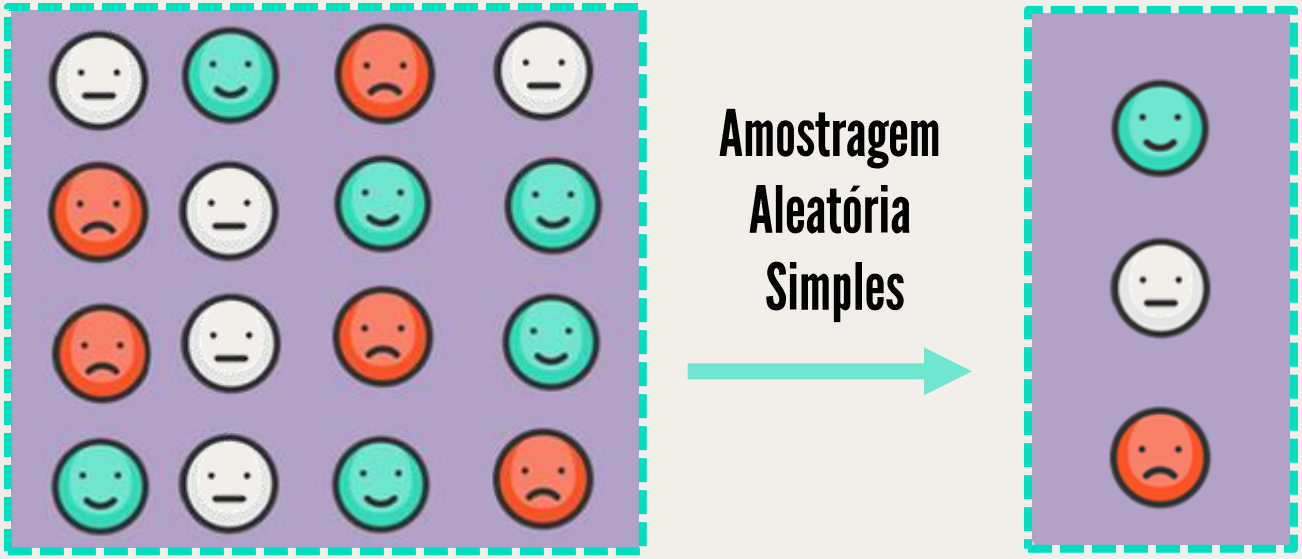
- Algoritmos de AM podem ter *dificuldades em lidar com um nº grande de objetos*.
  - Ex: k-NN (*k-Nearest Neighbor*): problemas de saturação de memória quando um conjunto de dados tem um grande número de exemplos.
- *Eficiência computacional versus acurácia*: + dados, + acurácia, - eficiência computacional:
  - Para se obter um bom compromisso entre acurácia e eficiência computacional, geralmente trabalha-se com uma amostra ou subconjunto dos dados;
  - Muitas vezes, o uso de uma amostra leva ao mesmo desempenho obtido com o uso do conjunto completo, porém com um custo computacional menor.

AMOSTRAGEM DOS DADOS



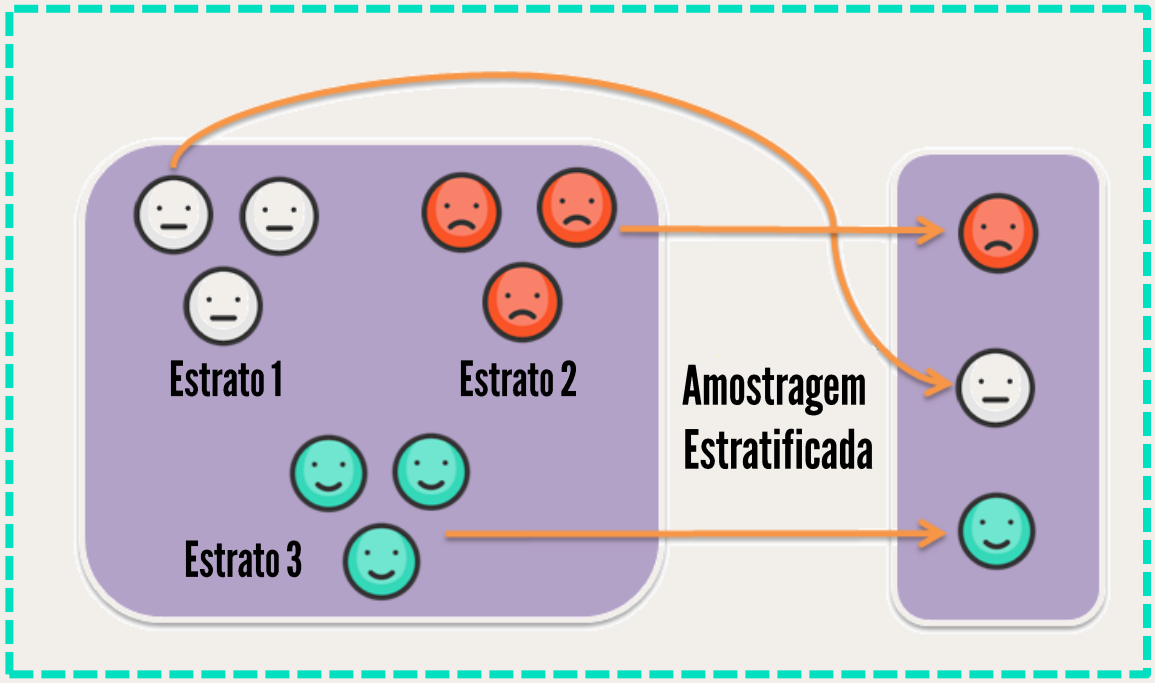
- Amostra pequena:
  - Pode *não ser representativa* → diferentes amostras da população gerando modelos diferentes;
  - Características importantes do problema ou distribuição que gerou os dados podem não estar presentes.
- O ideal é que a amostra não seja grande, mas que seus dados obedeçam à mesma distribuição estatística que gerou o conjunto de dados original.

AMOSTRAGEM DOS DADOS



http://www.datasciencemadesimple.com/

AMOSTRAGEM DOS DADOS



http://www.datasciencemadesimple.com/

Teremos uma aula especial sobre isso!

DADOS DESBALANCEADOS

- Tópico da área de classificação dos dados.
- Quando o *número de exemplos varia para as diferentes classes*:
  - Natural em alguns domínios;
  - Problema com geração / coleta de dados.
- Várias técnicas de AM não conseguem lidar com esse problema.
  - Tendência a classificar na(s) classe(s) majoritária(s).
  - Alternativa: balanceamento artificial.

DADOS DESBALANCEADOS

- Redefinir o tamanho do conjunto de dados:
  - *Oversampling*: risco de os objetos acrescentados representarem situações que nunca ocorrerão; risco de *overfitting;*
  - *Undersampling*: possível perda de dados de grande importância, risco de *underfitting*.
- Utilizar diferentes custos de classificação para as diferentes classes - Dificuldade de determinar a definição dos custos.
- Induzir um modelo para uma classe:
  - A classe minoritária, majoritária ou ambas são aprendidas separadamente;
  - Pode ser utilizado algoritmo de classificação para uma classe apenas.

Teremos uma aula especial sobre isso!

QUALIDADE DE DADOS

- Conjuntos de dados podem também apresentar dificuldades relacionadas à qualidade dos dados.
- Exemplos mais frequentes são:

Dados ruidosos: possuem erros ou valores diferentes do esperado.

Dados inconsistentes: não combinam ou contradizem valores de outros atributos do mesmo objeto.

Dados redundantes: dois ou mais objetos têm os mesmos valores para todos os atributos ou dois ou mais atributos têm os mesmos valores para dois ou mais objetos.

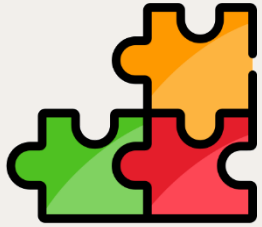
Dados incompletos: ausência de valores para alguns dos atributos em parte dos dados (fácil detecção).

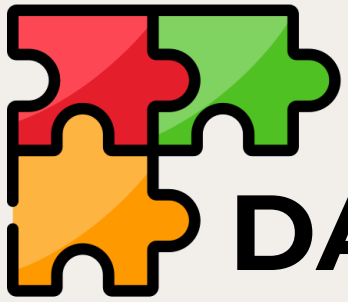
QUALIDADE DE DADOS

- Problemas podem ocorrer nos *processos de medições e na coleta de dados*.
- Exemplos de causas:
  - Falha humana;
  - Falha no processo de coleta de dados;
  - Limitações do dispositivo de medição;
  - Má fé;
  - Valor do atributo alvo muda com o tempo.

DADOS INCOMPLETOS

- Não é raro um objeto não ter o valor de um ou mais atributos.
- Possíveis causas:
  - Atributo não foi considerado quando os primeiros dados foram coletados (E-mail);
  - Desconhecimento do valor do atributo por ocasião do preenchimento (Tipo sanguíneo);
  - Distração, mal entendido ou declinamento na hora do preenchimento;
  - Não necessidade ou obrigação de apresentar um valor para atributo(s) de algumas instâncias (Renda);
  - Inexistência de valor para o atributo em algumas instâncias (Partos para sexo masculino);
  - Problema com dispositivo / processo de coleta.





DADOS INCOMPLETOS

- *Eliminar os objetos com valores ausentes*: geralmente empregada quando um dos atributos ausentes é o atributo classe.

Não é indicada quando:

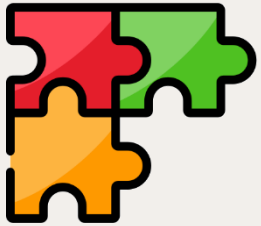
- Ocorre com poucos atributos do exemplo;
- Número de atributos com valores ausentes varia muito entre os exemplos com esse problema;
- Há risco de descartar dados importantes.

- *Definir e preencher manualmente valores para os atributos com valores ausentes* (não é factível quando número de atributos ou objetos com valores ausentes for muito grande);

- *Empregar algoritmos de AM que lidam internamente com valores ausentes*. Ex: árvores de decisão;

- *Estimativa de valores ausentes*.

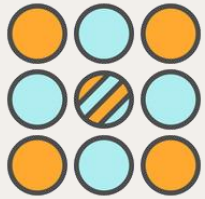
DADOS INCOMPLETOS



Heurísticas para estimativa:

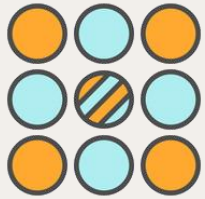
- *Criação de um novo valor*:
  - Problema: algoritmo assumir que valor desconhecido representa um conceito importante.
- *Média (mediana, moda) de todos os valores do atributo (ou apenas os da mesma classe)*:
  - Para série de valores, entre valores anterior e posterior;
- *Média (mediana, moda) dos vizinhos mais próximos*.
- *Valor induzido por algum estimador*:
  - Atributo ausente seria o atributo alvo e os demais atributos seriam os de entrada.
  - K-NN muito utilizadp

DADOS INCONSISTENTES



- Dados inconsistentes são aqueles que possuem *valores conflitantes em seus atributos*.
- Comuns no processo de integração dos dados (metros em um banco, centímetros em outro; siglas com letras maiúsculas e/ou minúsculas).
- Atributos de entrada:
  - Ex. Dados com código postal inválido para o nome de rua especificado, pessoa com 2 m pesando 10 Kg.
    - Erro / engano;
    - Proposital (fraude).
- Atributo de saída:
  - Podem levar a exemplos conflitantes:
    - Ex.: valores iguais para atributos de entrada e diferentes para atributo de saída.

DADOS INCONSISTENTES



Algumas inconsistências são de fácil detecção.

- Violação de relações conhecidas entre atributos:
  - Ex.: Valor de atributo A é sempre menor que valor de atributo B.
- Valor inválido para o atributo:
  - Ex.: altura com valor negativo.
- Em outros casos, informações adicionais precisam ser verificadas.

DADOS REDUNDANTES

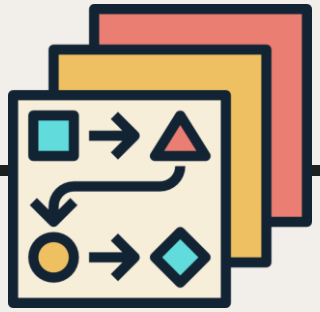
- Valores que *não trazem informação nova* (atributos e objetos).
- Objetos redundantes participam mais de uma vez do processo de ajuste de parâmetros de um modelo, contribuindo mais que os outros objetos → falsa impressão que esse perfil é mais importante que os demais.
- Atributos redundantes pode supervalorizar um dado aspecto dos dados ou tornar mais lento o processo de indução.

Dados (quase) duplicados - Ex.: Pessoas em diferentes bancos de dados com mesmo endereço e pequenas diferenças nos nomes.
Deduplicação: Detectar e eliminar (ou combinar) duplicações.

DADOS COM RUÍDOS

- Dados que contêm *objetos que, aparentemente, não pertencem à distribuição que gerou os dados analisados*.
- Ruído: variância ou erro aleatório no valor gerado ou medido para o atributo.
- *Dados com ruídos podem levar a um superajuste do modelo*.
- Não é possível ter certeza de que um valor apresenta ruído:
  - Tem-se apenas um indício, a menos que seja inconsistente.
  - *Outliers* podem sugerir a presença de ruído.

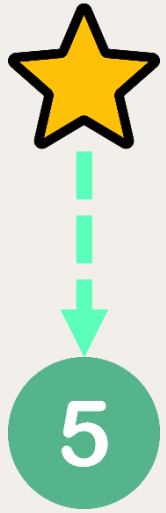
TRANSFORMAÇÃO DE DADOS



- Conversão de valores simbólicos para numéricos;
- Conversão de valores numéricos para simbólicos;
- Binarização;
- Normalização de valores numéricos;
- Tradução de atributos.

CONVERSÃO SIMBÓLICO-NUMÉRICO

- Algumas técnicas trabalham apenas com valores numéricos:
  - Valores simbólicos precisam ser convertidos para numéricos;
  - *Redes neurais, SVM*.
- Conversão depende de:
  - Ordenação dos valores: presente ou ausente
  - Número de valores: = 2 (binários) ou > 2.



CONVERSÃO NOMINAL PARA BINÁRIO

- *A inexistência de relação de ordem deve continuar para os valores numéricos gerados*.
- 1-de-c (codificação canônica):
  - Sequencia de c bits, em que c é igual a número de categorias ou possíveis valores;
  - Diferença entre valores: distância de Hamming: número de posições em que as sequências apresentam valores diferentes;
  - Cada sequência possui apenas um bit com o valor 1 e os demais com valor zero;
  - Moda = posição com maior número de valores 1;
  - Valores escalares podem virar vetores longos.

Exemplo:	
Azul	001
Amarelo	010
Vermelho	100

PSEUDOATRIBUTOS

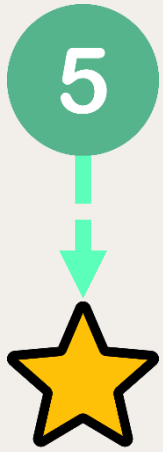
- Binário (b), inteiro (i) ou real (r).
- Imagine que um atributo seja nome de país:
  - Existem 193 países (192 representados na ONU + Vaticano).
  - Transformar 4 pseudoatributos:
    - Continente: 7 valores (b);
    - PIB: 1 valor (i, r);
    - População: 1 valor (i, r);
    - Área: 1 valor (i, r).

CONVERSÃO ORDINAL PARA BINÁRIO

- *Quando existe relação de ordem, a codificação deve preservar essa relação*.
- Codificar para valor inteiro positivo:
  - Ex. Pequeno (1), médio (2) e grande (3).
- Algumas técnicas trabalham apenas com valores binários:
  - Codificar cada valor por um vetor binário.
    - Código de Grey:
      - 000, 001, 011, 010, 110, 100.
    - Código termômetro:
      - 00000, 00001, 00011, 00111, 01111, 11111.

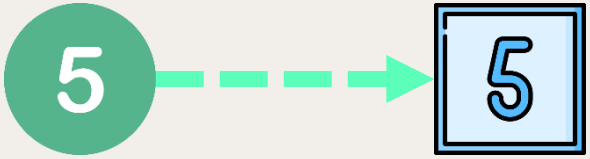
CONVERSÃO NUMÉRICO-SIMBÓLICA

- Discretização de valores:
  - Transformar valores numéricos em intervalos ou categorias.
- Sub-tarefas:
  - Definição do número de categorias geralmente feita pelo usuário.
- Definição de como mapear valores dos atributos contínuos para essas categorias:
  - Definição da frequência (pode gerar intervalos de tamanhos muito diferentes) / largura (afetado por *outliers*) dos intervalos;
  - Uso de um algoritmo de agrupamento de dados;
  - Inspeção visual.



TRANSFORMAÇÃO DE ATRIBUTOS NUMÉRICOS

- *Valor numérico de um atributo pode precisar ser transformado em outro*.
- Limites de valores para atributos distintos podem ser muito diferentes ou atributos em escalas diferentes:
  - Evitar que um atributo predomine sobre outro (a menos que isso seja importante).
- Aplicada aos valores de um dado atributo de todos os objetos:
  - Ex.: supor que apenas a magnitude do valor de um atributo é importante.
    - Converter valor de todos os atributos para o valor absoluto;
    - -4, 5 e –2 se tornam 4, 5 e 2.
- Variações:
  - Normalização;
  - Tradução.



NORMALIZAÇÃO

- *Faz com que o conjunto de valores de um atributo tenha uma dada propriedade*.
- Pela amplitude:
  - Reescala;
  - Padronização.
- Pela distribuição: muda escala de valores.
  - Ex.: função log.

Valiosa para os métodos que calculam distâncias entre atributos.

- **Por exemplo, um método como o k-vizinhos mais próximos tende a dar mais importância para os atributos que possuem um intervalo maior de valores. Outros métodos como redes neurais são reconhecidamente melhor treinadas quando os valores dos atributos são pequenos.**

Não é de grande utilidade para a maioria dos métodos que induzem representações simbólicas, tais como árvores de decisão, uma vez que tende a diminuir a compreensibilidade do modelo gerado.

REESCALA (MIN-MAX)

- Para reescalar os valores de um atributo:

$$v_{Novo} = \min + \frac{v_{Atual} - \text{menor}}{\text{maior} - \text{menor}} (\max - \min)$$

Valores limites da nova escala

- Permite converter todos os valores de um atributo para o intervalo [0, 1] (usar max = 1 e min = 0).

PADRONIZAÇÃO

- Para padronizar os valores de um atributo:

1. Adicionar ou subtrair uma medida de localização;
2. Multiplicar ou dividir por uma medida de escala.

- Se os valores têm uma distribuição Gaussiana:
  - Subtrair a média;
  - Dividir pelo desvio padrão;
  - Produz conjunto de valores com distribuição normal (0,1).

$$v_{Novo} = \frac{v_{Atual} - \mu}{\sigma}$$

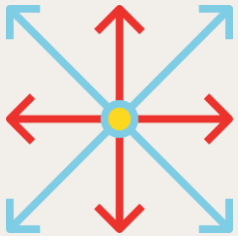
TRADUÇÃO

Ocorre devido a limitações no formato utilizado para armazenar o atributo.

- Alguns algoritmos podem ter dificuldades com formato original.
- Exemplos:
  - Conversão de hora para valor inteiro;
  - Conversão de data para valor inteiro;
  - Conversão de rua para código postal.

REDUÇÃO DE DIMENSIONALIDADE

- Alguns conjuntos podem ter um *número muito grande de atributos*.
  - Ex.: objeto é um vetor com frequência de cada palavra que aparece em um texto.
- Reduzir dimensão:
  - Agregação de atributos: criar novos atributos que são uma combinação dos atributos originais;
  - Seleção de atributos.



SELEÇÃO DE ATRIBUTOS

- Embutida: seleção é feita pelo algoritmo de AM (ex: árvores de decisão);
- Filtro;
- *Wrapper*.

- Identificar atributos importantes;
- Melhorar desempenho de algoritmo de para indução de modelos;
- Reduzir a necessidade de memória e tempo de processamento;
- Eliminar atributos irrelevantes e reduzir ruído;
- Simplificar o modelo gerado e tornar mais fácil sua interpretação;
- Facilitar a visualização dos dados;
- Reduzir custo de coleta de dados.

Teremos uma aula especial sobre isso!

FILTROS

Seleção de atributos independe do algoritmo de AM utilizado.

Ex.: verifica correlação entre atributos.

VANTAGENS

- Não depende do algoritmo de AM: os atributos selecionados podem ser utilizados por diferentes algoritmos de AM.
- Baixo custo computacional: podem ser muito rápidos.
- Conseguem lidar de forma eficiente com uma grande quantidade de dados.

DESVANTAGENS

- Ignora interação com o algoritmo: pode levar a modelos pouco eficientes.
- Pode ignorar dependências entre atributos.

WRAPPERS

Utilizam o algoritmo de AM para selecionar atributos.

Ex.: atributos que levaram a menos erros de classificação para uma rede MLP.

VANTAGENS

- Melhor conjunto para um dado algoritmo: pode selecionar também melhor número de atributos.
- Geralmente melhora desempenho obtido pelo algoritmo.

DESVANTAGENS

- Risco de overfitting;
- Desempenho depende do algoritmo de indução;
- Custo computacional elevado;
- Precisa ser repetido quando um novo algoritmo de AM for utilizado.

SELEÇÃO DE SUBCONJUNTO

Quatro aspectos precisam ser tratados:

- Ponto de início da busca e da geração de subconjuntos;
- Estratégia de busca;
- Estratégia de avaliação;
- Critério de parada (nº máximo de alternativas testadas, nº de atributos, tempo de processamento).

GERAÇÃO DE SUBCONJUNTOS

Existem quatro alternativas:

- Geração para trás (*backward generation*):
  - Começa com todos os atributos e remove um por vez.
- Geração para frente (*forward generation*):
  - Começa sem nenhum atributo e inclui um atributo por vez.
- Geração bidirecional (*bidirectional generation*):
  - Busca pode começar em qualquer ponto e atributos podem ser adicionados e removidos.
- Geração estocástica (*random generation*):
  - Ponto de partida da busca e atributos a serem removidos ou adicionados são decididos de forma estocástica.

ESTRATÉGIA DE BUSCA

- Define o algoritmo usado para realizar a busca.
- Busca completa (exponencial ou exaustiva):
  - Avalia todos os possíveis subconjuntos.
- Busca heurística (sequencial):
  - Utiliza regras e métodos para conduzir a busca;
  - Não garante que uma solução ótima seja encontrada.
- Busca não-determinística:
  - Fazem algum tipo de escolha aleatória;
  - Boa solução pode ser encontrada antes do final da busca;
  - Não garante ótimo.

REFERÊNCIAS

- Inteligência Artificial: Uma Abordagem de Aprendizado de Máquina, por Katti Faceli, Ana Carolina Lorena, João Gama, André C. P. L. F. de Carvalho;
- Notas de aula do curso Mineração de Dados em Biologia Molecular, ministrado por André C. P. L. F. de Carvalho;
- FRIEDMAN, Jerome; HASTIE, Trevor; TIBSHIRANI, Robert. The elements of statistical learning. New York: Springer series in statistics, 2001.
- GARCÍA, Salvador; LUENGO, Julián; HERRERA, Francisco. Data preprocessing in data mining. Cham, Switzerland: Springer International Publishing, 2015;
- KOTSIANTIS, S. B.; KANELLOPOULOS, Dimitris; PINTELAS, P. E. Data preprocessing for supervised leaning. International Journal of Computer Science, v. 1, n. 2, p. 111-117, 2006.