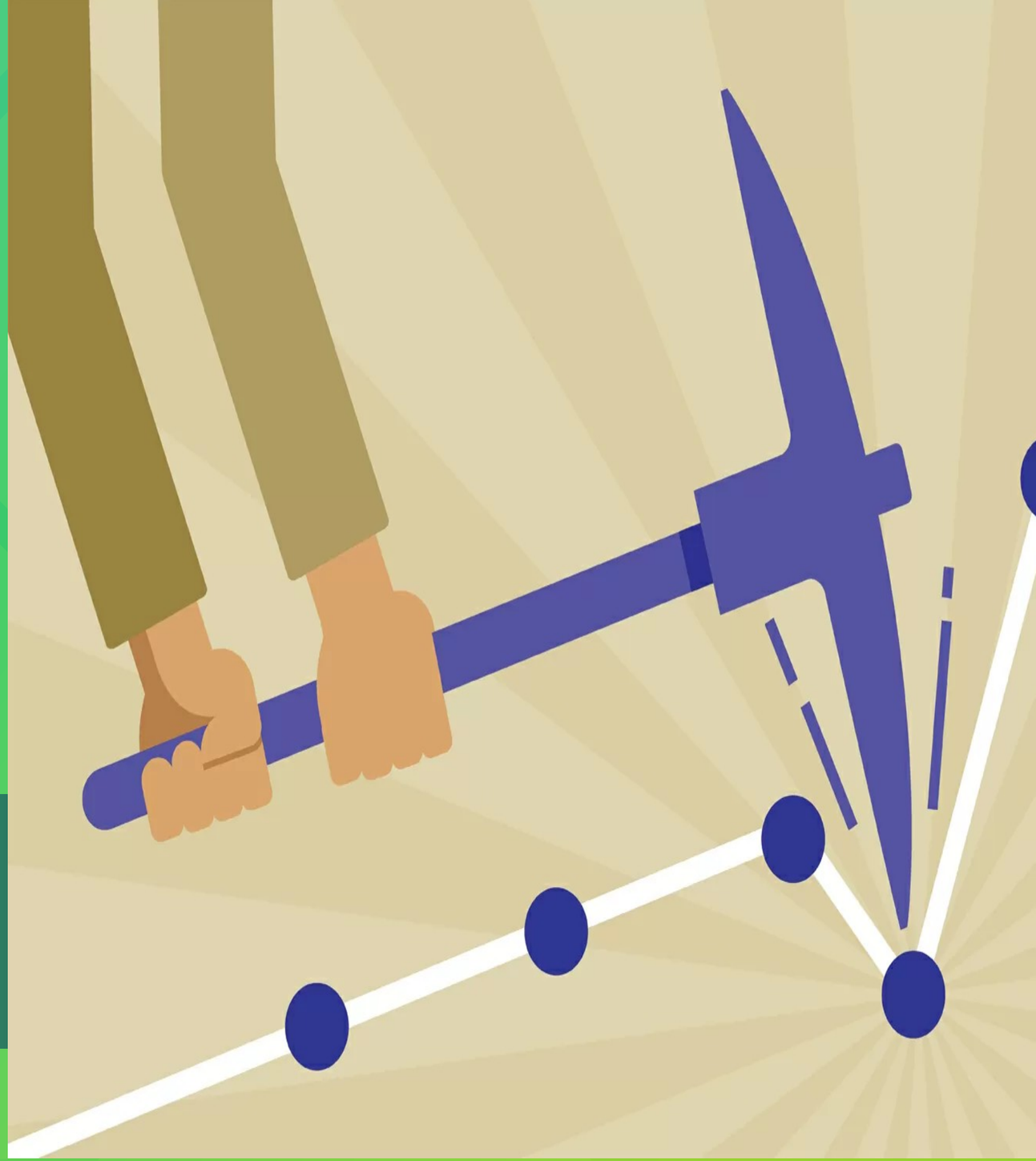
CÂMPUS

Bagé

TÓPICOS EM ADS II



INSTITUTO FEDERAL
Sul-rio-grandense

muito a descobrir

RODRIGO R SILVA

Pré-processamento: Limpeza e Transformação de Dados

Importância do Pré-processamento

- Antes de aplicar algoritmos de mineração de dados (classificação, agrupamento, regressão, etc.), é essencial preparar os dados. Bases de dados reais frequentemente apresentam problemas como:
- Valores ausentes (missing values).
- Dados redundantes ou inconsistentes.
- Escalas diferentes entre atributos (ex.: salário em milhares, idade em anos).
- Atributos irrelevantes ou ruidosos.

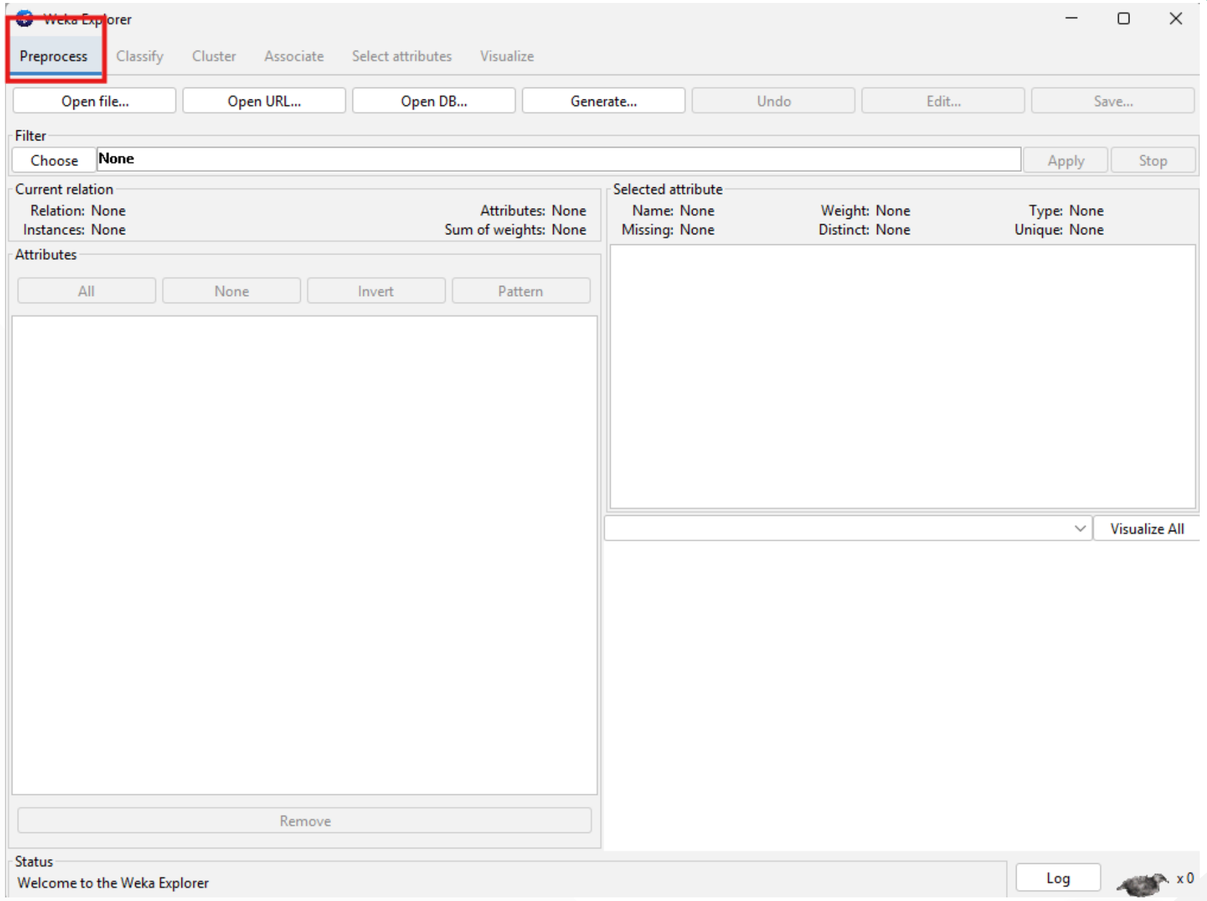
O **pré-processamento** busca melhorar a qualidade da base, aumentando a eficiência e a precisão dos modelos.

Etapas de Pré-processamento no WEKA

- Carregamento dos Dados

→ Arquivos suportados: .arff, .csv, .xrff, entre outros.

→ A aba Preprocess é a principal interface no WEKA para estas tarefas.



Limpeza dos Dados

A etapa de limpeza é uma das mais críticas no pré-processamento, pois problemas de qualidade impactam diretamente a performance dos algoritmos. No WEKA, várias ferramentas estão disponíveis na aba Preprocess para esse fim.

Tratamento de Valores Ausentes

- Bases reais frequentemente possuem campos não preenchidos.
- Estratégias no WEKA:
  - → Remover instâncias com valores ausentes:
  - → Filtro Choose → Filter → Unsupervised → Instance → RemoveWithValues.
  - → Substituir valores ausentes automaticamente:

    Filtro Choose → Filter → Unsupervised → Attribute → ReplaceMissingValues.

    - → Numéricos: substitui pela média.
    - → Nominais: substitui pela moda.
- Importante avaliar: se há muitos valores ausentes, pode ser melhor eliminar o atributo.

Limpeza dos Dados

Remoção de Atributos Irrelevantes ou Redundantes

- Dados podem conter atributos que não agregam informação útil.
- Exemplo: um campo ID ou número de protocolo não ajuda na classificação.
- Estratégias no WEKA:
  - Remove (Choose → Filter → Unsupervised → Attribute → Remove): exclui manualmente colunas escolhidas.
  - AttributeSelection: usa algoritmos para selecionar apenas atributos relevantes, reduzindo dimensionalidade.

Limpeza dos Dados

Detecção e Tratamento de Outliers

- Outliers = valores muito distantes da distribuição normal dos dados.
- Impactam algoritmos baseados em distância (KNN, clustering).
- WEKA não possui uma função “detectar outlier” explícita, mas:
  - → Pode-se usar filtros de discretização para agrupar valores extremos em categorias específicas.
  - → Também é possível analisar estatísticas descritivas em Visualize All e remover manualmente instâncias discrepantes com RemoveWithValues.

Padronização de Consistência

- Algumas vezes dados chegam em formatos diferentes (ex.: “Masculino/Feminino” vs. “M/F”).
- No WEKA, o ideal é uniformizar usando filtros de conversão de atributos (ex.: NominalToBinary).

Filtro AddExpression

- O filtro AddExpression cria novos atributos derivados a partir de expressões matemáticas aplicadas sobre os atributos já existentes em uma base de dados.
- Ele funciona como uma forma de engenharia de atributos, permitindo gerar variáveis que representem combinações ou transformações de outras, enriquecendo a base antes da aplicação de algoritmos de mineração.
- **Onde encontrar no WEKA**
  - → Caminho: Preprocess → Choose → Filter → Unsupervised → Attribute → AddExpression
  - → Tipo: Unsupervised Attribute Filter (não supervisionado, ou seja, não depende da classe).

Filtro AddExpression

Configurações principais

Ao selecionar o filtro, algumas opções podem ser configuradas:

- expression:

  Campo principal onde se escreve a fórmula desejada.

  Exemplos de operadores aceitos:

  → +, -, *, / → operações aritméticas.

  → pow(a,b) → potência.

  → sqrt(a) → raiz quadrada.

  → log(a) → logaritmo natural.

  → exp(a) → exponencial.

- range:

  Intervalo de atributos sobre os quais a expressão será aplicada.

  name: Nome que será dado ao novo atributo gerado.

Exemplo: Arquivo iris.arff

Filtro NominalToBinary

- O NominalToBinary é um filtro do WEKA utilizado para converter atributos nominais (categóricos) em atributos binários (0/1).
- Essa transformação é chamada de codificação binária ou one-hot encoding.
- Ela é necessária porque muitos algoritmos de mineração de dados e aprendizado de máquina trabalham melhor com dados numéricos do que com categorias textuais.

- **Onde encontrar**
  - Preprocess → Choose → Filter → Unsupervised → Attribute → NominalToBinary

Ele é um filtro não supervisionado de atributos (não depende da classe).

- **Como funciona**
  - Cada valor possível de um atributo nominal é transformado em uma nova coluna binária.
  - Na nova representação:
    - → O valor presente recebe 1.
    - → Os demais recebem 0.

Filtro NominalToBinary

- **Aplicações**
  - Classificação e Regressão: algoritmos como regressão logística, redes neurais e KNN exigem atributos numéricos.
  - Árvores de Decisão: não exigem transformação, mas podem se beneficiar dependendo do pré-processamento.
  - Análise de Similaridade: melhora a performance em algoritmos de clusterização (K-Means, EM).
- **Vantagens e Limitações**
- ✅ **Vantagens:**
  - Permite utilizar atributos nominais em algoritmos que só aceitam dados numéricos.
  - Garante uma representação clara e sem ambiguidade.
- ⚠ **Limitações:**
  - Pode aumentar muito a dimensionalidade da base, caso existam atributos com muitos valores distintos (ex.: um atributo "cidade" com 500 cidades diferentes → gera 500 novas colunas).
  - Isso pode causar sparsity (matriz esparsa) e aumentar o tempo de processamento.

Exemplo: Arquivo weather.nominal.arff

Filtro RenameNominalValues

- O RenameNominalValues é um filtro de pré-processamento do WEKA utilizado para renomear os rótulos de valores nominais (categóricos) de um atributo.
- Ele não altera a estrutura dos dados nem cria novos atributos — apenas substitui os nomes das categorias, mantendo os mesmos registros e quantidade de instâncias.

- **Onde encontrar**
  - Preprocess → Choose → Filter → Unsupervised → Attribute → RenameNominalValues
  - Categoria: Unsupervised Attribute Filter (não supervisionado).

- **Para que serve**
  - Padronizar rótulos que aparecem de formas diferentes (ex.: “Masculino” vs. “M”).
  - Tornar os nomes das categorias mais claros ou compatíveis com exigências de algoritmos.
  - Uniformizar bases de dados vindas de diferentes fontes.
  - Corrigir erros de digitação nos rótulos nominais.

Filtro RenameNominalValues

- **Configurações principais**

O filtro possui alguns parâmetros configuráveis:

- attributeIndices:

  Define quais atributos nominais terão os valores renomeados.

  →Ex.: first (primeiro atributo), last (último), 1-3 (atributos de 1 a 3).

- replace:

  Valor (rótulo nominal) a ser substituído.

  →replacement:

  Novo valor (novo rótulo) que entrará no lugar.

  →invertSelection:

  Se marcado como True, inverte a seleção dos atributos (aplica em todos os não escolhidos).

Exemplo: Arquivo weather.nominal.mod.arff

Filtro RenameAttribute

- O filtro RenameAttribute é usado para alterar o nome de um atributo em uma base de dados.
- Ele não modifica os valores do atributo, apenas muda o rótulo (nome da coluna) para algo mais significativo, padronizado ou legível.

- **Onde encontrar**
  - Preprocess → Choose → Filter → Unsupervised → Attribute → RenameAttribute
  - Categoria: Unsupervised Attribute Filter (não supervisionado, pois não depende da classe).

- **Para que serve**
  - Padronização de nomes de atributos: útil quando a base vem de diferentes fontes e possui nomes inconsistentes.
  - Melhora da legibilidade: nomes curtos ou confusos podem ser substituídos por algo mais claro.
  - Compatibilidade com algoritmos: alguns algoritmos ou softwares externos exigem atributos sem espaços ou caracteres especiais.
  - Didática: em sala de aula, facilita o entendimento dos atributos ao deixar os nomes mais intuitivos.

Filtro RenameAttribute

- **Configurações principais**

O filtro tem opções simples:

- attributeIndices:
- Indica qual atributo será renomeado.
  - Ex.: first (primeiro atributo), last (último), 1 (atributo 1), 2-4 (atributos de 2 a 4).
- Replace:
- Nome atual do atributo que será substituído.
- replacement:

Novo nome do atributo.

Exemplo: Arquivo weather.nominal.mod.arff

Filtro Remove

- O filtro Remove é utilizado para excluir atributos ou instâncias de uma base de dados.
- Ele é um dos filtros mais básicos e frequentemente usados no WEKA, sendo essencial para limpeza e redução de dados.

- **Onde encontrar**
  - Preprocess → Choose → Filter → Unsupervised → Attribute → Remove (para atributos).
  - Preprocess → Choose → Filter → Unsupervised → Instance → RemoveWithValues (para instâncias).

O filtro Remove geralmente se refere à remoção de atributos (colunas).

- **Para que serve**
  - Eliminar atributos irrelevantes ou redundantes (ex.: IDs, códigos, nomes que não contribuem para o modelo).
  - Reduzir dimensionalidade manualmente, antes de aplicar técnicas automáticas como AttributeSelection.
  - Preparar a base para um experimento específico, quando nem todos os atributos são necessários.

Filtro Remove

- **Configurações principais**
  - attributeIndices:

    Indica quais atributos serão removidos.

    Exemplos:

    - 1 → remove o primeiro atributo.
    - last → remove o último atributo.
    - 1-3 → remove os atributos de 1 a 3.
    - 2,4,6 → remove os atributos 2, 4 e 6.
  - invertSelection:
    - False (padrão): remove apenas os atributos indicados.
    - True: remove todos os atributos exceto os indicados.

Exemplo: Arquivo weather.nominal.mod.arff

Filtro ChangeDateFormat

O ChangeDateFormat é um filtro usado para alterar o formato de atributos do tipo data em uma base de dados.

Ele permite transformar a forma como os valores de data/hora são representados (ex.: incluir horas, reduzir apenas para ano/mês, mudar padrão de escrita etc.).

👉 Isso é útil porque diferentes fontes de dados podem trazer datas em formatos distintos, e muitos algoritmos de mineração precisam de padronização para processá-las corretamente.

- **Onde encontrar**
- Preprocess → Choose → Filter → Unsupervised → Attribute → ChangeDateFormat
- Categoria: Unsupervised Attribute Filter (não supervisionado).

- **Para que serve**
- Padronizar datas que vêm em formatos diferentes.
- Reduzir a granularidade temporal (ex.: de "2025-09-03 10:25:00" para apenas "2025-09").
- Compatibilizar bases de dados antes de uni-las.
- Preparar atributos temporais para uso em algoritmos que trabalham com tempo.

Filtro ChangeDateFormat

- **Configurações principais**

O filtro possui alguns parâmetros configuráveis:

- attributeIndices

  Indica quais atributos do tipo date terão o formato alterado.

  Exemplos:

  - first → aplica no primeiro atributo.
  - last → aplica no último atributo.
  - 2 → aplica no atributo 2.
  - 1-3 → aplica nos atributos de 1 a 3.

- dateFormat

  Define o novo formato da data usando padrões do Java SimpleDateFormat. Exemplos:

  - yyyy-MM-dd → 2025-09-03
  - dd/MM/yyyy → 03/09/2025
  - yyyy-MM → 2025-09
  - MM-yyyy HH:mm → 09-2025 14:30

Filtro MergeManyValues

O MergeManyValues é um filtro do WEKA que permite agrupar (fundir) vários valores nominais de um atributo em uma única categoria.

Ele é muito útil quando um atributo possui muitos valores diferentes, mas para o processo de mineração de dados faz sentido reduzir a granularidade das categorias, simplificando a análise.

- **Onde encontrar**
- Preprocess → Choose → Filter → Unsupervised → Attribute → MergeManyValues
- Categoria: Unsupervised Attribute Filter.

- **Para que serve**
- Reduzir número de categorias em atributos nominais muito detalhados.
- Simplificar análise quando diferenças entre valores não são relevantes para o modelo.
- Evitar sparsity (esparsidade) após uso de NominalToBinary, que gera muitas colunas quando há valores demais.
- Agrupar classes raras em uma categoria única ("outros").

Filtro MergeManyValues

- **Configurações principais**

O filtro possui os seguintes parâmetros:

- attributeIndex

Define qual atributo nominal será modificado.

Exemplos:

first/last → primeiro atributo/último atributo.

2 → segundo atributo.

- mergeValueRange

Define o intervalo de valores nominais que serão agrupados.

Exemplos:

2-4 → funde os valores da posição 2 até 4 da lista de categorias.

1,3 → funde os valores 1 e 3.

- label

Nome que será dado ao novo valor resultante da fusão.

Exemplo: "outros" ou "agrupado".