UNIVERSIDADE FEDERAL DE UBERLÂNDIA

Luiz Eugênio de Paula

# Adaptação de um algoritmo de agrupamento para aplicação em dados de acidentes de trabalho

**Uberlândia, Brasil**

**2023**

UNIVERSIDADE FEDERAL DE UBERLÂNDIA

Luiz Eugênio de Paula

# Adaptação de um algoritmo de agrupamento para aplicação em dados de acidentes de trabalho

Trabalho de conclusão de curso apresentado à Faculdade de Computação da Universidade Federal de Uberlândia, Minas Gerais, como requisito exigido parcial à obtenção do grau de Bacharel em Sistemas de Informação.

Orientador: Elaine Ribeiro de Faria Paiva

Universidade Federal de Uberlândia – UFU

Faculdade de Ciência da Computação

Bacharelado em Sistemas de Informação

Uberlândia, Brasil

2023

# Agradecimentos

Agradeço primeiramente a Deus por me dar saúde, força e ânimo para concluir mais esta etapa da minha vida.

À minha orientadora, Professora Dr[a]. Elaine Ribeiro de Faria, que com muita paciência e dedicação, me ajudou a concluir essa jornada com êxito.

À minha família, especialmente minha esposa Francielle, que sempre me apoiou e não deixou que eu desistisse.

Aos professores, mestres e doutores da FACOM, os quais fui aluno, que muito contribuíram para minha formação.

À Daniela Freitas Giacomelli, que confiou em mim e me convidou para este projeto e a todos que de alguma forma contribuíram para que eu finalizasse esse trabalho com sucesso.

# Resumo

A grande quantidade de acidentes de trabalho no Brasil leva o país a ter um alto custo com benefícios, além de impactar diretamente no SUS (Sistema único de Saúde) e na qualidade de vida dos trabalhadores. O Ministério Público do Trabalho (MPT) lançou o Observatório de Segurança e Saúde no Trabalho, onde é possível ter acesso aos dados de acidentes de trabalho do Brasil de forma organizada. O trabalho de mestrado **Técnicas de Agrupamento aplicadas aos Dados de Acidente de Trabalho** (GIACOMELLI, 2020) objetivou aplicar algoritmos de agrupamento de dados na base de acidentes de trabalho para tentar obter padrões úteis. Este trabalho apoiou o projeto de mestrado por meio do desenvolvimento de técnicas que permitissem executar experimentos com algoritmos de agrupamento na base de dados de acidente. Para isso, foi necessário tratar a base de dados, adaptar um algoritmo de agrupamento hierárquico para lidar com uma grande base de dados e implementar uma medida de distância para atributos categóricos. Este trabalho adaptou um software desenvolvido na linguagem java com implementações de novas funcionalidades para tratar a base, removendo atributos irrelevantes, objetos com informações nulas e realizando transformações de alguns atributos. No algoritmo de agrupamento, foram desenvolvidas novas implementações para o cálculo da medida de distância entre os objetos e para que a base de grande volume foi processada. Também foram desenvolvidas adaptações na medida de validação usada. Ao final do processo, as implementações permitiram a execução do algoritmo na base de acidentes de trabalho com um tempo de execução satisfatório.

**Palavras-chave**: Acidentes de trabalho, pré-processamento de dados, distância, atributos categóricos.

# Lista de ilustrações

# Lista de tabelas

# Lista de abreviaturas e siglas

| | |
|---|---|
| DBSCAN | *Density-Based Spatial Clustering of Applications with Noise*(Clustering Espacial Baseado em Densidade de Aplicativos com Ruído) |
| Eps | *Epsilon* (Raio do ponto central até a borda) |
| EM | *Expectation Maximization* |
| HDBSCAN | *Hierarchical Density-Based Spatial Clustering of Applications with Noise* |
| INSS | Instituto Nacional de Seguro Social |
| KDD | *Knowledge Discovery in Databases* |
| KNN | *k-nearest neighbor* (K-vizinhos mais próximos) |
| OSST | Observatório de Segurança e Saúde no Trabalho |
| MST | *Minimum Spanning Tree* |
| CNAE | Classificação Nacional de Atividades Econômicas |
| CAT | Comunicação de Acidente de Trabalho |

# Sumário

# 1 Introdução

Após a Segunda Guerra Mundial, o Brasil sofreu uma rápida transformação industrial. Fábricas foram construídas, as existentes foram expandidas e na agricultura começou o processo de diversificação (FREITAS, 2021). Ferramentas manuais foram perdendo espaço para as máquinas. Todas essas mudanças requeriam um número grande de trabalhadores, que iriam desempenhar uma atividade desconhecida (NOGUEIRA, 1987). Os treinamentos para os trabalhadores não eram bons e, diante disso, a taxa de acidentes de trabalho eram altas (NOGUEIRA, 1987).

Atualmente, o modo de trabalho da sociedade evoluiu acompanhando o desenvolvimento da tecnologia, porém os acidentes de trabalho também atingiram outros níveis (SOUZA, 2017). No Brasil há muitas empresas que trabalham de forma precária, colocando em risco a integridade física de seus funcionários. Esse fato reflete no alto número de acidentes de trabalho que consequentemente eleva os gastos previdenciários. Segundo o jornal O Globo, acidentes de trabalho custaram à previdência, entre 2012 e 2017, cerca de RS 26 bilhões (GLOBO, 2018). Estes acidentes de trabalho são rastreados mediante a CAT que é a Comunicação de Acidente de Trabalho. A CAT tem a função de notificar a Previdência Social sobre a ocorrência de um acidente, a fim de garantir a assistência ao trabalhador.

O problema gerado pelos acidentes de trabalho diz respeito à questão de integridade física dos trabalhadores, mas também gera grande impacto negativo na economia do país. Como consequência, recentes iniciativas começaram a utilizar os avanços da tecnologia da análise de dados a fim de gerar resultados que possam contribuir para mitigar tais problemas.

Uma das importantes iniciativas para divulgação dos dados de acidentes trabalhistas é o Observatório de Segurança e Saúde no Trabalho (SMARTLAB, 2019). Este observatório visa informar e subsidiar políticas públicas de prevenção de acidentes e doenças no trabalho, de modo que todas as ações, programas e iniciativas sejam orientadas por evidências (SMARTLAB, 2019).

Diante da importância de se entender melhor os acidentes de trabalho ocorridos no Brasil e da disponibilização dos dados pelo Observatório Digital, o uso de técnicas de mineração de dados e inteligência artificial é uma das possibilidades de análise desses dados. Tais técnicas permitem obter padrões que possam ser analisados, interpretados, classificados e rotulados, gerando assim conhecimento.

Dentre as técnicas de mineração de dados, destaca-se o agrupamento de dados (do inglês, *clustering*). O agrupamento visa identificar clusters distintos em uma base

de dados(TANTRUM; MURUA; STUETZLE, 2003). Considerando que os dados sobre acidentes de trabalho descrevem a característica do acidente, mas não possuem um rótulo (uma classificação) associada, a tarefa de agrupamento de dados torna-se uma alternativa interessante para sua análise.

O objetivo deste trabalho é adaptar um algoritmo de agrupamento de dados, denominado HDBSCAN* (CAMPELLO; MOULAVI; SANDER, 2013) para ser aplicado na base de acidentes de trabalho em busca de padrões que possam ajudar a entender os acidentes. Como exemplo, esses padrões podem sugerir:

- regiões que ocorrem acidentes similares;
- tipos mais frequentes de acidentes em uma determinada profissão;
- horários que ocorrem determinados tipos de acidente;
- partes do corpo que são mais afetadas em uma dada profissão.

## 1.1 Motivação

A motivação deste trabalho é apoiar o projeto de mestrado "Técnicas de Agrupamento aplicadas aos Dados de Acidente de Trabalho", desenvolvido na FACOM (GIACOMELLI, 2020), cujo objetivo é extrair conhecimento a partir de uma base de acidentes de trabalho do Brasil usando técnicas de agrupamento. Esta base de dados possui algumas características que tornam a aplicação de métodos de agrupamento uma tarefa desafiadora. Primeiramente, a base é composta por uma quantidade grande de atributos categórico nominais, o que dificulta o uso de vários algoritmos de agrupamento, que só lidam com valores numéricos. Em segundo lugar, cada atributo categórico da base possui uma grande quantidade de categorias possíveis, o que inviabiliza a conversão desses atributos para valores numéricos usando conversões clássicas da literatura. Além disso, a base possui uma quantidade excessiva de valores ausentes, o que exige a aplicação de técnicas de pré-processamento para remover e tratar tais valores.

O trabalho "Aplicação de técnicas de pré-processamento e agrupamento na base de dados de benefícios previdenciários do Ministério Público do Trabalho"(SILVA, 2018) investigou o uso dos algoritmos de agrupamento **k-means**, **canopy** e **EM** em uma base de dados semelhante. Na etapa de pré-processamento, esse trabalho converteu os atributos categóricos para numérico usando a conversão 1-de-n, que cria um atributo para cada possível categoria. Essa conversão gerou uma base de alta dimensionalidade. Os resultados desse trabalho não foram satisfatórios, pois segundo (SILVA, 2018) os valores do índice de silhueta foram baixos em todas as bases onde esses algoritmos foram aplicados.

O trabalho "Técnicas de Agrupamento aplicadas aos Dados de Acidente de Trabalho"(GIACOMELLI, 2020) investigou o uso de outros algoritmos de agrupamento, tais como o HDBSCAN* e CobWeb. A justificativa para a escolha do algoritmo HDSBSCN* se deu por ele ser hierárquico, baseado em densidade e pelos seus resultados interessantes na literatura recente.

No entanto, para a aplicação do algoritmo HDBSCAN* na base de dados de acidentes de trabalho, uma série de adaptações precisam ser feitas, as quais incluem:

- Identificar os pré-processamentos necessários, tais como o tratamento de valores ausentes e remoção de colunas indesejadas;
- Conversão de atributos categóricos para numéricos, caso eles possuam poucas categorias;
- Adaptação de medidas de distância para os atributos categóricos com muitos valores, de forma a não convertê-os em atributos numéricos, gerando uma base de alta dimensionalidade;
- Adaptar a etapa do algoritmo que calcula a matriz de distância entre todos os objetos da base, já que essa matriz não caberia em memória devido ao volume de dados da base de acidentes;
- Adaptar a medida de validação usada já que ela usa uma medida de distância.

## 1.2 Objetivos

### 1.2.1 Objetivo Geral

O objetivo geral deste trabalho é implementar alterações no código do algoritmo de agrupamento de dados HDBSCAN* a fim de que ele possa ser aplicado nos dados de acidentes de trabalho. As alterações incluem a implementação de uma nova medida de distância adaptada para atributos categóricos.

### 1.2.2 Objetivos específicos

Para alcançar o objetivo geral, faz-se necessário estabelecer alguns objetivos específicos, tais como:

- Pré-processar os dados para limpeza e normalização utilizando a aplicação ConverteDados, desenvolvida por (SILVA, 2018).
- Converter atributos categóricos para numéricos, quando os mesmos possuíssem poucas categorias.

- Adaptar o algoritmo HDBSCAN* com a inserção de novas medidas de distância entre atributos categóricos.
- Adaptar o algoritmo HDBSCAN* modificando a etapa que calcula a matriz de distância entre todos os objetos da base, já que a mesma não cabe em memória quando se utiliza a base de acidentes de trabalho.
- Adaptar a medida de validação usada para avaliar a qualidade do agrupamento gerado pelo HDBSCAN*, já que a mesma é baseada no cálculo de distância entre dois objetos.

## 1.3 Organização do Trabalho

O trabalho será disposto da seguinte forma:

- **Capítulo 2:** Apresenta uma visão sobre acidentes de trabalho, etapas da extração do conhecimento em bases de dados (KDD) e os trabalhos relacionados.
- **Capítulo 3:** Relata como o trabalho foi conduzido, qual o método empregado, quais etapas, bem como cada etapa foi conduzida.
- **Capítulo 4:** Tece as principais conclusões sobre o trabalho.

# 2 Referencial Teórico e Trabalhos Relacionados.

## 2.1 Introdução

Neste capítulo serão apresentados os principais conceitos relacionados ao trabalho desenvolvido. Também serão apresentados os principais trabalhos relacionados.

A Seção 2.2 apresenta os acidentes de trabalho, dados estatísticos e seus impactos socio-econômicos. A Seção 2.3 é uma abordagem sobre o KDD, suas etapas e a importância fundamental do agrupamento dos dados para a análise, obtenção de padrões e a extração de conhecimento. Na Seção 2.4 aborda-se a necessidade sobre analisar a base de dados para conhecer suas características e classificá-las. Na Seção 2.5 é tratada as questões sobre pré-processamento dos dados, como exemplo, a remoção de dados ausentes. A Seção 2.6 trata-se sobre a transformação dos dados, como de categóricos para numéricos, antes da aplicação do método de agrupamento. A Seção 2.7 é um breve resumo sobre os principais algoritmos de agrupamento. Na Seção 2.8 a abordagem é voltada para a apresentação do HDBSCAN*, foco principal deste trabalho. Na Seção 2.9 são apresentados os principais trabalhos relacionados.

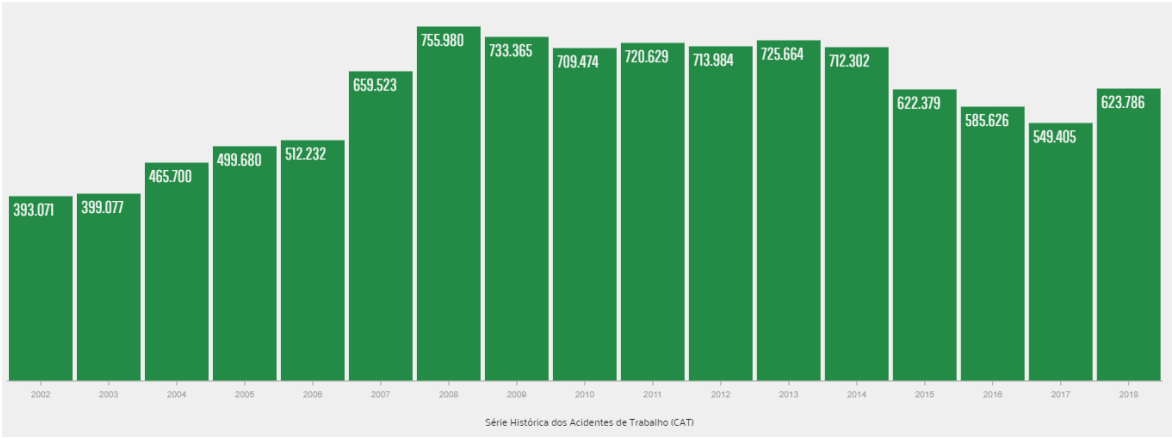
## 2.2 Acidentes de Trabalho

De acordo com Bittencourt (2019), os acidentes de trabalho no Brasil geram um grande impacto social e econômico, mas que não é lhe dado a devida importância. O fato é que as mortes, acidentes e doenças relacionadas ao trabalho são uma questão de saúde pública, muitas vezes ignoradas pela sociedade. No país, quarta posição no ranking mundial de acidentes de trabalho, a Previdência Social registra por ano cerca de 700 mil casos Bittencourt (2019), e, segundo dados do Observatório Digital de Segurança e Saúde do Trabalho, chega-se a contabilizar uma morte por acidente em serviço a cada três horas e 49 minutos (SMARTLAB, 2019).

De acordo com dados da Previdência oficial, entre 2014 e 2018 foram registrados no Brasil 1,8 milhões de afastamentos por acidente de trabalho e 6,2 mil óbitos. Na Bahia, esse número foi de 44.800 afastamentos e 272 mortes (BITTENCOURT, 2019).

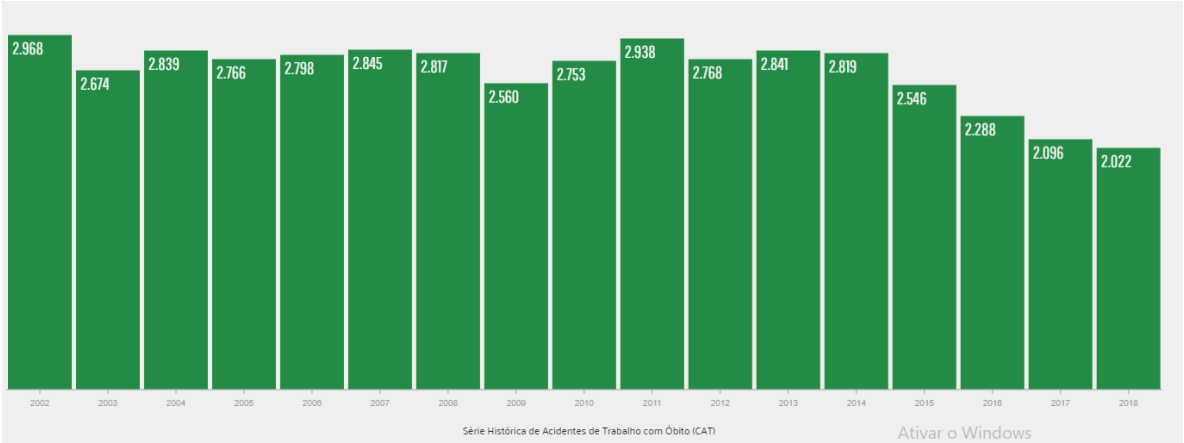
Na Figura 1, é apresentado um gráfico com a quantidade de acidentes de trabalhos, notificados anualmente de 2002 a 2018. Este gráfico só contempla os trabalhadores com vínculo empregatício regular e não contabiliza os acidentes com trabalhadores informais.

Figura 1 – Acidentes de trabalho notificados para a população com vínculo de emprego regular.




Fonte: (SMARTLAB, 2019)

Figura 2 – Série Histórica de Acidentes de Trabalho com Óbito.

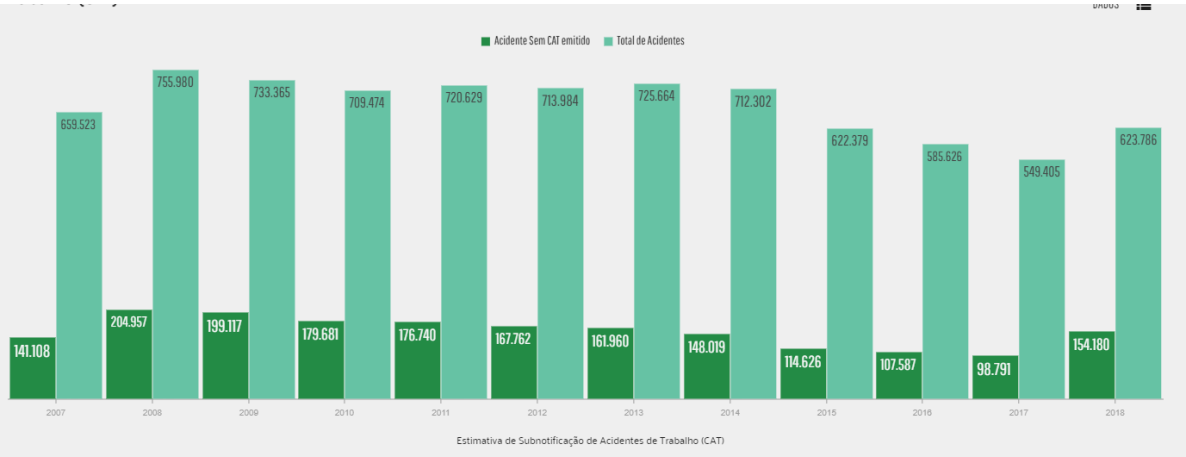



Fonte: (SMARTLAB, 2019)

Na Figura 2, é apresentada a taxa de óbitos anuais, na qual pode-se destacar que no ano 2018, mesmo apresentando o menor índice, tem-se ainda mais de 2000 acidentes de trabalho que resultaram em óbito. Vale ressaltar que essa estatística está relacionada com dados oficiais, ou seja, com trabalhadores com carteira assinada ou emprego formal. O que se pode imaginar é que essa realidade pode ser muito maior, visto a quantidade de trabalhadores que não tem carteira assinada, como os que trabalham por conta própria e entram na esteatítica dos informais, que somam 24,2 milhões de pessoas no Brasil, segundo o IBGE (IBGE, 2019). Uma estimativa da quantidade de acidentes de trabalho que são sub-notificados, pode ser visto na Figura 3, onde se tem uma projeção anual desta estimativa desde 2007 (SMARTLAB, 2019).

O alto número de acidentes de trabalho notificados e sub-notificados, mostrados na Figura 3 , gera um custo muito elevado para o país com aposentadorias e afastamento pelo INSS, como pode ser observado na Figura 4. Uma das possíveis formas de se reduzir o número de acidentes é por meio de políticas públicas que atuem na prevenção de tais aci-

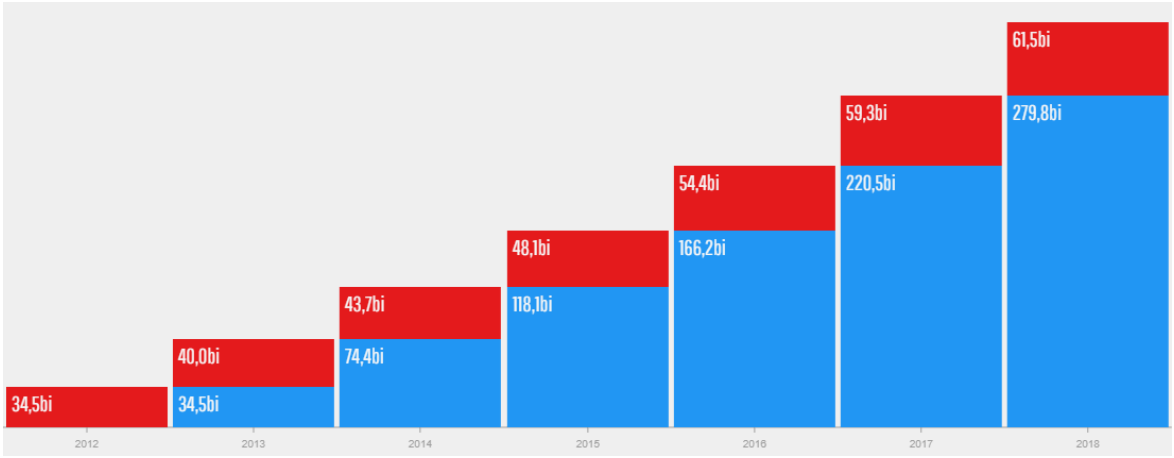
Figura 3 – Estimativa de Sub-notificação de Acidentes de Trabalho.




Fonte: (SMARTLAB, 2019)

dentes, como o Trabalho Seguro, criado pelo Tribunal Superior do Trabalho (TST)(TST, 2022). Esse tipo de estratégia tem um baixo custo e pode atuar nas principais fontes do problema (LABORE, 2021). A fim de que tais políticas sejam aplicadas, faz-se necessário ter um conhecimento amplo e assertivo sobre as principais causas desses acidentes.

Figura 4 – Valor pago a cada ano — Despesas acumuladas no período considerado.




Fonte: (SMARTLAB, 2019)

Atualmente técnicas computacionais para extração de conhecimento automático tem sido empregadas em grandes bases de dados visando descobrir padrões e regras significativas (BERRY; LINOFF, 1997).

Trabalhos recentes da área, como o trabalho de Danilo Silva (SILVA, 2018), aplicou técnicas de Inteligência Artificial e de Descoberta do Conhecimento em Bases de Dados (KDD) em dados de acidentes de trabalho visando auxiliar na descoberta de padrões. Já os trabalhos de (BRITO, 2019) e (RODRIGUES, 2019) utilizaram técnicas computacionais de agrupamento para criar estratégias visuais de análise de dados em cima da base de dados do MPT, com objetivos comuns de facilitar a compreensão dos dados pelos especialistas.

## 2.3 Descoberta do Conhecimento em Bases de Dados — (KDD)

O termo KDD, do inglês *Knowledge Discovery in Databases*, foi proposto pela primeira vez em 1989 no I *Workshop* KDD para enfatizar que o conhecimento é o produto final de uma descoberta feita a partir de uma base de dados (FAYYAD; PIATETSKY-SHAPIRO; SMYTH, 1996a). Popularizado em inteligência artificial e aprendizado de máquina, o KDD refere-se ao processo geral que envolve várias técnicas para o descobrimento de conhecimento útil a partir de uma base de dados (HAN; KAMBER; PEI, 2011). Dentro desse processo, pode-se dizer que mineração é apenas uma etapa específica, onde se aplicam algoritmos para extração de padrões de dados (LAROSE; LAROSE, 2014).

O processo KDD segue o caminho evolutivo, cruzando os campos da pesquisa com aprendizado de máquina, reconhecimento de padrões, base de dados, inteligência artificial, estatística, computação de alto desempenho e visualização de dados, objetivando extrair conhecimento de alto nível sobre dados de baixo nível, em contextos tradicionais de base de dados e Big data (FAYYAD; PIATETSKY-SHAPIRO; SMYTH, 1996b). O KDD enfatiza aspectos da busca de padrões compreensíveis que possam ser entendido como conhecimento útil ou interessante, enfatizando fortemente o trabalho como grandes conjuntos de dados do mundo real. Também existem muitas particularidades com as estatísticas, principalmente os métodos que utilizam análise de dados exploratórios(FAYYAD; PIATETSKY-SHAPIRO; SMYTH, 1996a).

### 2.3.1 Casos de Sucesso da Utilização do KDD

A aplicação do KDD tem sido cada vez mais utilizada com sucesso em muitas áreas distintas, o que tem contribuído para tomadas de decisões mais assertivas como, por exemplo:

- No campo da medicina tradicional chinesa (MTC), ajudando na pesquisa de ervas medicinais, fórmulas médicas e diagnósticos médicos (FENG et al., 2006).
- Na prevenção de incêndios, pode-se citar o estudo feito sobre a cidade de Manila, capital das Filipinas, onde se aplicou o (KDD) em um *dataset* sobre queimadas, que levou à identificação de padrões dos incêndios, gerando informações sobre como fazer uma prevenção (BALAHADIA et al., 2020).
- Na detecção de fraudes, sistemas como HNC Falcon e Nestor PRISM são usados para monitorar fraudes de cartão de crédito. O sistema FAIS, da *U.S. Treasury Financial Crimes Enforcement Network*, é usado para identificar transações financeiras que podem indicar atividade de lavagem de dinheiro (FAYYAD; PIATETSKY-SHAPIRO; SMYTH, 1996a).

- No setor de marketing e risco financeiro, o (KDD) é utilizado para identificar o comportamento financeiro dos usuários, que facilita a criação de campanhas de marketing diferentes para cada cluster (LEECE, 1999).

### 2.3.2 Critérios Práticos Para Projetos KDD

Segundo (FAYYAD; PIATETSKY-SHAPIRO; SMYTH, 1996a), os critérios práticos para projetos KDD são similares com outras aplicações de tecnologia de ponta e incluem o potencial impacto de uma aplicação como, ausência de soluções alternativas e o forte suporte organizacional para fazer o uso da tecnologia. Os aspectos técnicos incluem cuidados com a disponibilidade de dados suficientes, pois quanto mais complexo é o padrão procurado, mais dados (casos) são necessários. Todavia, um grande conhecimento acerca do padrão alvo e do comportamento dos atributos dos dados, podem diminuir consideravelmente o tamanho numérico desses casos.

Outra consideração é a relevância dos atributos. É importante ter atributos relevantes para resultados satisfatórios. A qualidade dos dados implica diretamente no resultado desejado, ou seja, os dados devem conter os atributos necessários para obter as informações. Além disso, um baixo nível de ruído é outra consideração a ser observada. A alta quantidade de ruído dificulta a identificação de padrões, a menos que um grande número de casos possa mitigar ruídos aleatórios e ajudar a esclarecer os padrões agregados.

Também deve-se dar atenção aos valores ausentes. Muitos valores podem estar ausentes devido a diversos fatores como erro humano, falha de hardware, falha no processo de registro, entre outros. Esses valores ausentes afetam negativamente a qualidade do resultado. Existem várias estratégias (e variações dessas estratégias) para lidar com dados ausentes, cada uma contendo vantagens e desvantagens (TAN; STEINBACH; KUMAR, 2005).

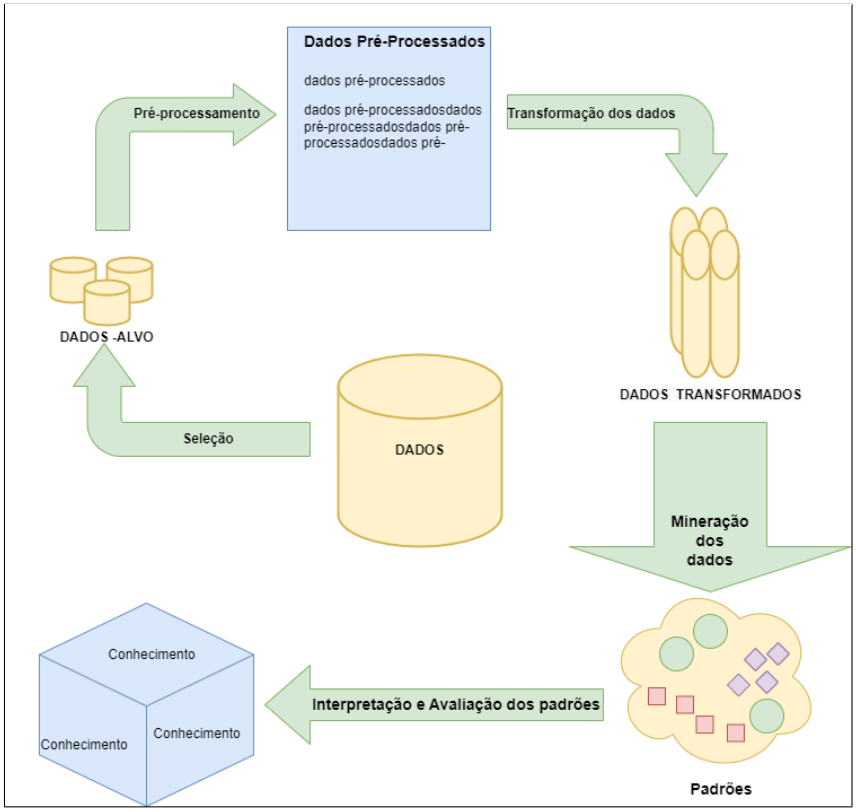
A escolha correta do algoritmo de mineração é outro fator primordial. Escolher o algoritmo segundo o conhecimento sobre a base de dados tem forte impacto nos dados. Todos esses critérios devem ser avaliados cuidadosamente para se escolher o algoritmo ideal para cada problema.

### 2.3.3 Etapas do Processo de KDD

O processo do KDD é interativo e iterativo, possuindo várias etapas conforme pode-se observar na Figura 5. As principais etapas, segundo (FAYYAD; PIATETSKY-SHAPIRO; SMYTH, 1996a), estão listadas a seguir.

- **Entendimento**: Consiste em desenvolver um entendimento do domínio da aplicação, utilizando conhecimentos anteriores e relevantes para desenvolver o objetivo do

Figura 5 – Etapas do Processo de KDD.




Fonte: Adaptado de (FAYYAD; PIATETSKY-SHAPIRO; SMYTH, 1996a).

processo KDD, conforme o objetivo definido.

- **Seleção de Dados**: Constitui-se em selecionar um conjunto de dados ou focar em um subconjunto de variáveis, ou amostras de dados, no qual deverá ser trabalhada a descoberta do conhecimento.

- **Limpeza e Pré-Processamento dos Dados**: Consiste nas operações básicas que incluem remover os ruídos, coletando as informações necessárias para contabilizar padrões ou modelos. Também inclui a decisão de estratégias para lidar com campos de dados ausentes.

- **Redução e Projeção de Dados**: Consiste em encontrar recursos úteis para representar os dados dependendo do objetivo da tarefa. Usando métodos de redução ou transformação de dimensionalidade, o número efetivo de variáveis pode ser reduzido, ou representações contantes para os dados podem ser encontrados.

- **Análise, Modelo Exploratório e Seleção de Hipóteses**: Constitui-se em escolher o(s) algoritmo(s) de mineração de dados e selecionar o(s) método(s) a serem usados para procurar padrões de dados. Esse processo inclui decidir quais modelos e parâmetros podem ser apropriados.

- **Escolha da Tarefa de Mineração de Dados**: Consiste na realização de pequisas com interesse em uma forma de representação específica ou em um conjunto dessas representações, incluindo regras ou árvores de classificação, regressão e agrupamento. O usuário pode ajudar significativamente o método de mineração de dados executando as etapas anteriores corretamente.

- **Mineração de Dados**: Consiste na escolha do(s) algoritmo(s) de mineração de dados, baseado-se no objetivo geral e na consequente estrutura imposta aos dados. A decisão do(s) algoritmo(s) envolve a escolha de modelos, parâmetros, formas de execução, etc.;

- **Interpretação Adequada dos Resultados da Mineração**: Consiste na interpretação dos padrões de mineração, possivelmente retornando a qualquer uma das etapas anteriores para iteração adicional. Esta etapa também pode envolver visualização dos padrões extraídos e modelos ou visualização dos dados. Os padrões devem representar um conhecimento novo.

- **Conhecimento Prévio obtido**: Consiste na utilização do conhecimento descoberto usando o diretamente, ou incorporando-o em outro sistema para ação adicional ou simplesmente documentando-o e relatando-o às partes interessadas.

### 2.3.4 Desafios de Pesquisa e Aplicação

Segundo (FAYYAD; PIATETSKY-SHAPIRO; SMYTH, 1996a), alguns dos principais desafios de pesquisa e aplicação encontrados por quem trabalha com KDD são:

- **Grande volume de dados**: Registros mensurando vários gigabytes são comuns em banco de dados hoje em dia, e não é difícil encontrar armazenamentos na casa dos terabytes (1012 bytes)(FAYYAD; PIATETSKY-SHAPIRO; SMYTH, 1996a).

- **Alta dimensionalidade**: Além do alto número de registros, também pode haver um grande número de campos (atributos, variáveis), aumentando assim a dimensionalidade do problema. Isso aumenta a chance de falsos padrões pelos algoritmos, sendo assim necessário o uso de conhecimento prévio para analisar a base, identificar, retirar os atributos e variáveis irrelevantes (FAYYAD; PIATETSKY-SHAPIRO; SMYTH, 1996a)

- **Avaliação e significância estatística**: Um problema (relacionado ao sobre-ajuste) ocorre quando o sistema está procurando muitos modelos possíveis, é um problema estatístico quando se compara várias populações. Este ponto é frequentemente esquecido por muitas tentativas iniciais de KDD. Uma maneira de lidar com o problema é usar métodos que ajustam a estatística de teste em função da pesquisa, como a *Correção de Bonferroni* (ARMSTRONG, 2014).

- **Modificação de dados e conhecimento**: Modificar dados pode invalidar padrões previamente descobertos. Além disso, as variáveis medidas em um determinado banco de dados podem ser modificadas, excluídas ou aumentada com novas medições ao longo do tempo. Como soluções podemos incluir métodos incrementais para atualizar os padrões e tratar a mudança como uma oportunidade de descoberta.

- **Ruídos e dados ausentes**: Este é um problema agudo, especialmente, em base de dados. Importantes atributos podem estar ausentes se o banco de dados não foi projetado com essa visão em mente. Possíveis soluções incluem estratégias estatísticas mais sofisticadas para identificar variáveis ocultas e dependências (HECKERMAN, 1997).

- **Relações complexas entre campos**: Atributos e variáveis com relações complexas exigem algoritmos capazes de lidar com essa complexidade para obter conhecimento.

- **Compreensão dos padrões**: O objetivo final do processo de KDD é tornar o conhecimento "descoberto" compreensível para os seres humanos. As possíveis soluções incluem representações gráficas (HECKERMAN, 1997; Buntine, 1996), regras, estruturação, geração de linguagem natural e técnicas para visualização de dados e conhecimento. Estratégias de refinamento de regras (por exemplo, (MAJOR; MANGANO, 1995)) podem ser usadas para resolver problemas relacionados(FAYYAD; PIATETSKY-SHAPIRO; SMYTH, 1996a).

- **Interação do usuário e conhecimento prévio**: Muitos métodos do KDD não são verdadeiramente interativos e, incorporar conhecimento prévio pode não ser uma tarefa trivial. Outro fator a ser considerado é que podem existir vários formatos de estruturas complexas na base de dados como estruturas espaciais, temporais, multimídia entre outras, o que dificulta ainda mais a tarefa.

## 2.4 Tipos de Atributos

Antes de tudo, no processo de mineração, deve-se ter total conhecimento dos tipos de dados/atributos contidos na base de dados para uma análise detalhada. Isso requer um conhecimento detalhado das características que os atributos podem ter. Nesta seção será apresentada as principais características dos diferentes tipos de atributos.

### 2.4.1 Dados numéricos ou Quantitativos

Segundo (HAN; KAMBER; PEI, 2011), dados numéricos são dados que têm significado como medida, altura, peso, QI ou pressão sanguínea de uma pessoa; ou são uma contagem, como quantidade de páginas.

Os dados numéricos podem ser divididos em dois tipos: discretos e contínuos.

- Dados discretos representam itens que podem ser contados; eles assumem valores possíveis que podem ser listados.
- Dados contínuos representam medições; seus valores possíveis não podem ser contados e só podem ser descritos usando intervalos na reta numérica real.

### 2.4.2 Dados Categóricos ou Qualitativos

Dados categóricos são dados decorrentes da observação de variáveis categóricas, ou seja, aqueles que identificam para cada caso uma categoria. Os dados categóricos representam características como, por exemplo: sexo, estado civil, cidade natal ou os tipos de filmes. Dados categóricos podem assumir valores numéricos (como "1" indicando masculino e "2" indicando feminino), mas esses números não têm significado matemático. As categorias podem ser derivadas de variáveis nominais ou ordinais.

### 2.4.3 Dados Ordinais

Dados ordinais misturam dados numéricos e categóricos. Os dados se enquadram em categorias, mas os números colocados nas categorias têm significado. Por exemplo, a classificação de uma pesquisa de satisfação de 1 a 5 fornece dados ordinais. Os dados ordinais são frequentemente tratados como categóricos. No entanto, ao contrário dos dados categóricos, os números têm significado matemático.

## 2.5 Pré-processamento de dados

Os dados no mundo real, são geralmente dados incompletos, cheios de ruídos, inconsistentes e essas são características presentes na maioria das bases de dados (HAN; KAMBER; PEI, 2011). Diante desse cenário se torna necessário uma etapa onde os dados sejam tratados e normalizados antes da aplicação dos algoritmos de mineração de dados.

Dentro de todo processo de mineração de dados, a etapa de pré-processamento talvez seja a mais importante para se ter um bom resultado ao final do processo de KDD (TAN; STEINBACH; KUMAR, 2005). A qualidade do resultado de um modelo final depende da qualidade dos dados que irão ser submetidos a ele. Além disso, os dados a serem trabalhados devem obedecer às características dos algoritmos ao qual irão ser submetidos. Caso isso não ocorra, deve-se trabalhar a etapa de pré-processamento. Por exemplo, um algoritmo que só trabalha com dados numéricos precisa converter dados categóricos e nominais para numéricos antes de sua execução.

As principais etapas envolvidas no pré-processamento de dados podem ser classificadas como, limpeza de dados, integração de dados, redução de dados e transformação de dados. As rotinas de limpeza de dados funcionam para "limpar" os dados, preenchendo os valores ausentes, suavizando dados ruidosos, identificando ou removendo *outliers* e resolvendo inconsistências (HAN; KAMBER; PEI, 2011). Esse procedimento é de extrema importância, pois dados ruidosos podem causar confusão para o procedimento de mineração, resultando em uma saída não confiável.

## 2.6 Transformação de Dados

Transformação de dados é o processo de converter a estrutura ou formato de dados para outro formato, que consiste em facilitar sua análise. De acordo com (HAN; KAMBER; PEI, 2011) a transformação dos dados envole:

- **Agregação**: consiste na aplicação de técnicas de agregação (resumo). Esta etapa é normalmente usada na construção de um banco de dados para análise dos dados em várias granularidades.
- **Generalização**: onde os dados de baixo nível ou "primitivos" (brutos) são substituídos por conceitos de nível superior por meio do uso de hierarquias. Por exemplo, atributos categóricos como bairros podem ser generalizados para um nível superior como cidade. Atributos numéricos como idade poder ser mapeados para um nível superior como criança, jovem, adulto e idoso.
- **Normalização**: Nesta etapa, dados de atributo são dimensionados para pertencer a um pequeno intervalo especificado.
- **Construção de Atributos**: Nesta etapa novos atributos são construídos e adicionados ao conjunto de atributos fornecido para ajudar no processo de mineração, contribuindo para melhorar a precisão e compreensão da estrutura dos dados de alta dimensão.

A transformação de dados é uma parte importante para o processo de mineração, pois todo seu processo influencia de forma direta na formação dos agrupamentos. Cada escolha do processo de transformação, normalização e criação de novos atributos em alguns cenários implica na adaptação dos algoritmos que realizam os cálculos de distâncias. Se o algoritmo está adaptado para calcular atributos numéricos, é preciso garantir que na base de dados não exista dados categóricos. Caso exista, é necessário fazer a transformação do dado sem alterar a sua característica de informação (semântica). Esta parte do processo inspira atenção, pois o algoritmo HDBSCAN*, objeto de estudo e modificação deste trabalho, utiliza algoritmos para cálculo da distância que são baseados em atributos

numéricos. No entanto, a base alvo deste trabalho, a base de dados CATWEB, possui a maioria dos atributos categóricos.

## 2.7 Técnicas de Agrupamento

Segundo (BINDRA; MISHRA, 2017), a principal tarefa no processo de mineração de dados é o agrupamento. Ele desempenha um papel extremamente importante em todo o processo de KDD, pois a categorização de dados é uma das etapas mais rudimentares na descoberta de conhecimento. É uma tarefa de aprendizagem não supervisionada usada para análise exploratória de dados, para encontrar alguns padrões não revelados que estão presentes nos dados, mas não podem ser categorizados claramente.

Técnicas de agrupamento têm sido amplamente utilizada em inúmeras aplicações, incluindo pesquisa de mercado, reconhecimento de padrões, análise de dados e processamento de imagens. Nos negócios, o agrupamento pode ajudar os profissionais de marketing a descobrir clusters distintos em suas bases de clientes e caracterizar clusters de clientes com base em padrões de compra (HAN; KAMBER; PEI, 2011). Algoritmos diferentes podem ser usados para separar dados de natureza semelhante. Ao contrário dos algoritmos de classificação, algoritmos de agrupamento pertencem ao grupo de algoritmos não-supervisionados. Nesta seção descrevem-se alguns algoritmos de agrupamento e suas principais características.

### 2.7.1 Diferentes Tipos de Agrupamentos

As técnicas de agrupamento visam encontrar clusters úteis de objetos, cujo propósito difere pelos objetivos. Existem várias abordagens de um cluster que se mostram úteis.

#### 2.7.1.1 Bem Separados

Um cluster é um grupo de objetos onde cada objeto é muito similar (está perto) a outros objetos do mesmo cluster e não similar a objetos de outros clusters. Esta definição de cluster é satisfeita somente quando os dados contêm agrupamentos naturais que estão muito distantes um dos outros (TAN; STEINBACH; KUMAR, 2005).

#### 2.7.1.2 Baseado em Protótipos

Um cluster é um grupo de instâncias onde elas são muito similares entre si. Isto implica que as instâncias estão mais perto do protótipo (modelo) que define o cluster do que dos protótipos que definem outros clusters. Em resumo, são similares ao protótipo que define o cluster e dissimilares em relações aos protótipos que definem outros clusters.

Para dados com atributos contínuos, o protótipo de um cluster é muitas vezes um centroide (a média de todos os pontos do cluster). Quando um centroide não é o ponto mais representativo, tal quando os dados são atributos categóricos, neste caso, muitas vezes o ponto mais representativo passa a ser um medóide (STRUYF; HUBERT; ROUSSEEUW, 1997).

#### 2.7.1.3 Baseado em Grafos

Um cluster baseado em grafos é definido quando os dados de um cluster são representados em um grafo, onde os pontos são objetos e as linhas representam conexões entre esses objetos. Os objetos desse cluster não podem possuir conexões com objetos de outros clusters. Os *contiguity-based clusters* são exemplos de clusters baseados em grafos, onde dois objetos são conectados somente se estiverem dentro de uma determinada distância entre si(TAN; STEINBACH; KUMAR, 2005).

#### 2.7.1.4 Baseado em Densidade

O cluster é a região mais densa, cercado por regiões menos densa. Os algoritmos baseado em densidade são mais utilizados quando os dados são irregulares ou entrelaçados e quando ruídos e *outliers* estão presentes na base de dados.

### 2.7.2 Técnica de Agrupamento Particional

A versão mais simples e fundamental da análise de agrupamento é o particionamento, que organiza os objetos de um conjunto em vários clusters ou agrupamentos exclusivos. Para manter a especificação do problema concisa, pode-se assumir que o número de agrupamentos é considerado conhecimento de base. Este parâmetro é o ponto inicial para o método de particionamento. Formalmente, dado um conjunto de dados $D$, com $N$ objetos, o objetivo é agrupar os objetos em $k$ clusters.

Algoritmos de agrupamento particionais organizam os objetos em $k$ partições ( $k \leq n$), onde cada partição representa um cluster (HAN; KAMBER; PEI, 2011). Esses clusters são formados para otimizar um critério de partição objetiva, como uma função de dissimilaridade com base na distância, de modo que os objetos num cluster sejam "semelhantes" entre si e "diferentes" para objetos de outros clusters em termos de atributos do conjunto de dados.

Como representante dessa técnica pode-se citar o algoritmo K-means(HAN; KAMBER; PEI, 2011).

### 2.7.3 Técnicas de Agrupamento Hierárquico

Agrupamento hierárquico (também chamado de análise de cluster hierárquico ou HCA) é um método de análise de cluster que tem em vista construir uma hierarquia de clusters. As estratégias para agrupamento hierárquico, geralmente, se enquadram em dois tipos:

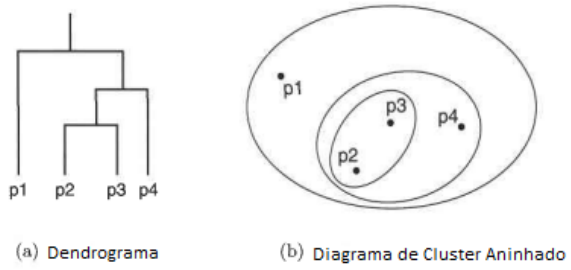
- **Aglomerativa**: Esta é uma abordagem *bottom up*, onde cada instância inicialmente representa um cluster e pares de clusters são unidos à medida que se sobe na hierarquia(HAN; KAMBER; PEI, 2011). Um único cluster torna-se a raiz da hierarquia. Para a etapa de união, o algoritmo de agrupamento encontra os dois clusters que estão mais próximos um do outro de acordo com alguma medida de similaridade e combina os dois para formar um cluster. Como dois clusters são unidos a cada iteração, onde cada cluster contém pelo menos um objeto, esse passo requer no máximo $n$ iterações.
- **Divisivo**: Esta é uma abordagem *top down*, onde todos os objetos começam formando um cluster e as divisões são realizadas recursivamente à medida que se desce na hierarquia (HAN; KAMBER; PEI, 2011). Cada divisão resulta em vários subclusters menores e particiona recursivamente esses clusters em outros menores. O processo de particionamento continua até que cada cluster no nível mais baixo seja coerente o suficiente — contendo apenas um objeto ou os objetos dentro de um cluster que são semelhantes entre si.

De acordo com (TAN; STEINBACH; KUMAR, 2005), as técnicas de agrupamento hierárquico aglomerativo são as mais comuns. Seus resultados são geralmente apresentados em um dendrograma (diagrama que representa uma árvore), que exibe os relacionamentos cluster-subcluster e a ordem onde os clusters foram unidos (visualização aglomerativa) ou divididos (visualização divisiva). Para conjuntos de pontos bidimensionais, um agrupamento hierárquico também pode ser representado graficamente usando um diagrama de agrupamento aninhado. A representação gráfica do dendrograma e do agrupamento aninhado podem ser observadas na Figura 6.

#### 2.7.3.1 Definição da Proximidade entre Clusters - Agrupamento Hierárquico Aglomerativo

A proximidade entre clusters é a medida de qual é a distância entre eles e como eles se relacionam. Esta informação pode ser usada para identificar padrões, reduzir variâncias e tornar modelos estatísticos mais precisos. A proximidade entre clusters também pode ser usada para descobrir relações mais complexas entre dados que não podem ser detectadas com ferramentas de análise de dados mais simples e também ajudam na identificação de áreas problemáticas, na identificação de padrões espaciais e na identificação de *outliers*.

Figura 6 – Agrupamento hierárquico de quatro pontos mostrado como um dendrograma e como um agrupamento aninhado.




(a) Dendrograma (b) Diagrama de Cluster Aninhado

Fonte: (TAN; STEINBACH; KUMAR, 2005)

É importante utilizar a proximidade entre os clusters para melhorar a análise de dados e para maximizar a acurácia das previsões. Como exemplo de técnicas de proximidade de agrupamento hierárquico aglomerativo, pode-se citar MIN, MAX e *Group Average*.

- **MIN** — A proximidade entre dois clusters é definida como o mínimo da distância (máximo de similaridade) entre quaisquer dois pontos nos dois clusters diferentes (TAN; STEINBACH; KUMAR, 2005). Usando a terminologia de grafos, começa com os pontos mais próximos entre os clusters adicionando links entre eles, deixando os pontos mais distantes em segundo plano. A técnica de link único é boa para lidar com formas não elípticas, mas é sensível a ruídos e valores discrepantes.

- **MAX** — A proximidade entre dois clusters é definida considerando os dois pontos mais distantes (mínimo de similaridade) entre os eles. Usando a terminologia de grafos, começando com todos os pontos do cluster, adicionando links entre os pontos um de cada vez, os links mais curtos primeiro, um cluster de pontos não será um cluster até que todos os pontos nele estejam completamente vinculados. Também chamada de link completo, esta técnica é menos suscetível a ruídos e outliers (TAN; STEINBACH; KUMAR, 2005).

- ***Group Average*** — Para esta técnica, a proximidade entre dois clusters é definida como a média da distância entre todos os pares de pontos nos dois clusters. Esta é uma abordagem intermediária entre as abordagens de Max e Min (TAN; STEINBACH; KUMAR, 2005). Assim, para a média, o cluster calcula a distância utilizando a Equação 2.1

$$proximidade\ (C_i, C_j\ ) = \frac{\left(\sum_{y\epsilon\ C_j}^{x\epsilon\ C_i}\ distancia\,(x,y)\ \right)}{M_i\ \cdot M_j} \tag{2.1}$$

onde $C_i$ e $C_j$ representam clusters, $x$ e $y$ representam objetos e $M_i$ e $M_j$ representam os tamanhos dos clusters $C_i$ e $C_j$ respectivamente.

### 2.7.4 Agrupamentos Baseados em Densidade

Para encontrar clusters de forma arbitrária, alternativamente, pode-se modelar os clusters como regiões densas no espaço de dados, separadas por regiões de baixa densidade. Esta é a principal estratégia por trás dos métodos de agrupamento baseados em densidade, que podem descobrir agrupamentos de formato não esférico. A densidade de um objeto X qualquer, pode ser medida pelo número de objetos próximos de X. O DBSCAN (ESTER et al., 1996) é um algoritmo desenvolvido para encontrar objetos centrais, ou seja, objetos que possuem regiões densas. Ele conecta objetos centrais e suas vizinhanças para formar regiões densas como clusters (HAN; KAMBER; PEI, 2011).

Na Figura 7, têm-se exemplos de objetos em formato oval e formato de "S", onde outros métodos têm dificuldades de execução e o DBSCAN retorna ótimos resultados.

Figura 7 – Aglomerados arbitrários e em forma de "S"

Fonte: (HAN; KAMBER; PEI, 2011)

O algoritmo DBSCAN possui uma abordagem baseada no centro para classificar um ponto da base de dados em um dentre três tipos:

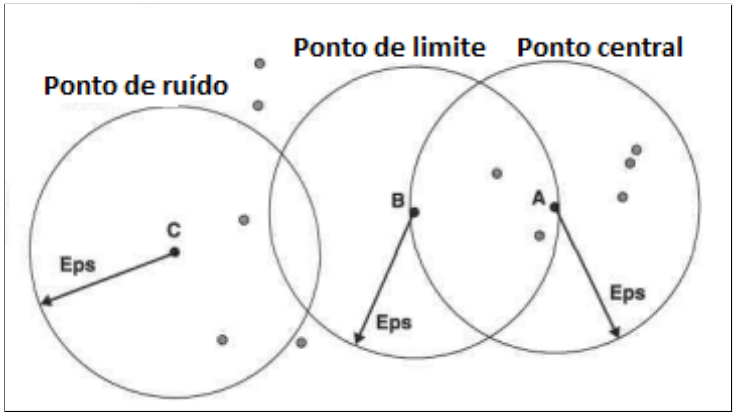
- (1) — no interior de uma região densa (um ponto central),
- (2) — na borda de uma região densa (um ponto limite), ou
- (3) — em uma região pouco povoada (um ruído ou ponto de fundo).

Na Figura 8, é exemplificado os três tipos de pontos usados pelo algoritmo DBSCAN.

## 2.8 HDBSCAN*

Esta seção apresenta o algoritmo HDBSCAN* (*Hierarchical Density-Based Spatial Clustering of Applications with Noise*), que foi o algoritmo de agrupamento usado neste trabalho. Segundo (CAMPELLO; MOULAVI; SANDER, 2013), o HDBSCAN* é uma variante hierárquica do DBSCAN e atual estado da arte no que se refere a agrupamento

Figura 8 – Pontos de ruído, limite e central.




Fonte: (HAN; KAMBER; PEI, 2011)

hierárquico e tem ganhado muita atenção em muitos campos diferentes de pesquisas nos últimos anos.

É um algoritmo baseado em densidade, o que significa que para cada concentração de elementos em um espaço amostral é criado um novo cluster. Em seu trabalho, (CAMPELLO et al., 2015) cita que o algoritmo generaliza e melhora as técnicas existentes de clusterização baseadas em densidade em relação a diferentes aspectos. Que o HDBSCAN* fornece como resultado uma hierarquia de clustering completa composta de todos os possíveis clusters baseados em densidade seguindo o modelo não paramétrico adotado, para uma faixa infinita de limiares de densidade.

É um algoritmo hierárquico, ou seja, para determinar se cada ponto do espaço pertence a um ou outro cluster, é aplicada a técnica do agrupamento hierárquico, que se baseia na distância dos dados desse espaço amostral. Para calcular essas distâncias, diferentes medidas podem ser usadas, como Euclidiana, Pearson, Manhattan, etc., escolhidas conforme os dados da base.

Esse algoritmo também é ideal para ser aplicado em bases com ruídos, pois como o algoritmo é baseado em densidade, no processo de formação dos clusters, pontos que não pertençam a nenhum cluster, que estão em regiões de baixa densidade, são interpretados como pontos de ruído (CAMPELLO et al., 2015). As subseções 2.8.1, 2.8.2 e 2.8.3 a seguir detalham o processo do HDBSCAN*.

### 2.8.1 Cálculo da Distância

De acordo com a implementação do algoritmo HDBSCAN*, utilizada neste trabalho, inicialmente a base de dados deve ser carregada na memória do computador para o cálculo das distâncias entre os objetos e assim gerar a matriz de distâncias.

O cálculo da matriz de distância é realizado considerando dois parâmetros, que são o *dataset* (matriz de dados) e o parâmetro que representa qual a medida de distância

a ser utilizado.

Ao calcular a matriz de distância com a métrica definida, o algoritmo assumirá que, em vez de receber um vetor de objetos em um espaço vetorial, ele está recebendo uma matriz de distância entre todos os pares de objetos. Esta matriz de distância é o cálculo da distância de um ponto para cada ponto no conjunto de dados e a distância Euclidiana é geralmente utilizada como padrão nas implementações, mas pode ser alterada, conforme as características dos atributos ou opção de quem está manipulando.

Ao final deste processo, a saída é uma matriz com as distâncias que será utilizada para a construção da *Minimum Spanning Tree*.

### 2.8.2 Minimum Spanning Tree

Nesta etapa será gerada a árvore geradora de custo mínimo ou de peso mínimo, do inglês *Minimum Spanning Tree* (MST). A MST é um subconjunto das arestas ligadas de um grafo. A ponta que liga todos os vértices juntos é ponderado, sem ciclos e com o peso total mínimo de borda possível. É uma árvore geradora cuja soma dos pesos das arestas é a menor possível.

Com a matriz de distâncias geradas na seção 2.7.2, o próximo passo e calcular a MST para descobrir as áreas densas no espaço. Essas áreas são relativas, e diferentes agrupamentos podem ter diferentes densidades. Conceitualmente, o que será feito é considerar os dados como um grafo ponderado com os pontos sendo considerados vértices e uma aresta entre quaisquer dois pontos com peso igual à distância de alcançabilidade mútua desses pontos(TAN; STEINBACH; KUMAR, 2005).

A técnica utilizada é encontrar um conjunto mínimo de arestas, onde o algoritmo implementa a teoria dos grafos para resultar a árvore geradora mínima (MST). A técnica implementa a MST de maneira muito eficiente por meio do algoritmo de Prim. A complexidade de tempo do Algoritmo de Prim é $O((V + E)logV$ porque cada aresta é inserida na fila de prioridade apenas uma vez e a inserção na fila de prioridade leva tempo logarítmico (SöRENSEN; JANSSENS, 2005).

### 2.8.3 Computando a Hierarquia

Nesta etapa do algoritmo é computada a hierarquia e a árvore de clusters. A função responsável, recebe como entrada os parâmetros da MST, computada na etapa anterior. Essa função tem como saída a hierarquia de cluster. É nesse passo que cada aresta do grafo criado, serão analisadas por distância, em ordem decrescente, construindo cada nível da hierarquia, composto por objetos conectados que são os clusters e objetos isolados que são os ruídos.

Em suma, o HDBSCAN* calcula a matriz de distância entre todos os objetos da base e a seguir constrói a árvore geradora mínima a partir dessas distâncias. O próximo passo é calcular a hierarquia a partir da árvore geradora mínima, gravando ambas no arquivo e retornando a hierarquia do agrupamento. Por fim, identifica os *outliers* e removê-os. O algoritmo 1 dá uma visão macro das etapas do HDBSCAN*.

**Algorithm 1** - Algoritmo HDBSCAN*. Fonte: adaptado de (CAMPELLO; MOULAVI; SANDER, 2013)

```
Inicio
enquanto o < tamanho do conjunto de dados faça
    Lê o-ésimo objeto da base;
    Adiciona o objeto o na matriz M de objetos;
fim enquanto
enquanto o < nro de objetos da matriz M faça
    Calcula a distância do o-ésimo objeto para todos os objetos do conjunto de dados.
fim enquanto
Calcula a árvore geradora mínima estendida a partir de um grafo ponderado, onde as
distâncias de alcance mútuo são as arestas;
Constrói a hierarquia HDBSCAN* a partir da árvore estendida geradora mínima;
Encontra os clusters proeminentes da hierarquia;
Identifica os outliers;
Fim
```

## 2.9 Trabalhos Relacionados

Esta seção é para apresentar os trabalhos relacionados com esta pesquisa. Serão apresentados os dois trabalhos antecessores que também usaram base de dados semelhantes como foco em agrupamento

O trabalho (SILVA, 2018) aplicou técnicas de pré-processamento e agrupamento de dados na base de benefícios previdenciários do Ministério Público do Trabalho. No pré-processamento, os atributos categóricos foram convertidos em numéricos usando a codificação 1-de-n. Os algoritmos de agrupamento usados foram *k-means*, *canopy* e EM, os quais não apresentaram bons resultados devido à alta dimensionalidade dos dados gerados a partir do pré-processamento. A medida de validação foi a silhueta simplificada.

O trabalho de mestrado **Técnicas de Agrupamento aplicadas aos Dados de Acidente de Trabalho** (GIACOMELLI, 2020), objetivou encontrar padrões na base de dados de acidentes de trabalho. Para isso, dois algoritmos de agrupamento foram usados, o *CobWeb* e o *HDBSCAN**. Para a execução do *HDBSCAN**, uma série de adaptações foram necessárias, as quais foram implementadas neste projeto de TCC.

Também deve-se citar a plataforma SmartLab que é uma iniciativa conjunta do (MPT) e da OIT Brasil (SMARTLAB, 2019), o qual centraliza os dados em uma única

plataforma de forma segura e transparente, beneficiando as pesquisas científicas que visam obter informações para contribuição social. A base de dados usada para validar as técnicas desenvolvidas neste projeto foi extraída desta plataforma.

## 2.10 Considerações Finais

Neste capítulo foi apresentado uma visão geral sobre acidentes de trabalho, descoberta do conhecimento (KDD), *clustering* (Agrupamentos), tipos de dados, pré-processamento, transformação de dados e o algoritmo HDBSCAN*.

No próximo capítulo será apresentado todo o desenvolvimento do projeto, os detalhes da base de dados CATWEB, as características dos seus atributos, o pré-processamento dos dados, as adaptações no software de pré-processamento, as funções criadas e as que foram adaptadas. Também serão apresentadas as novas funções de cálculo de distâncias implementadas no algoritmo HDBSCAN*, conforme as características dos atributos, bem como todas as adaptações realizadas no algoritmo, para que o mesmo tivesse condições de processar a base CATWEB.

# 3 Desenvolvimento

## 3.1 Introdução

Nos capítulos anteriores, foram apresentados a fundamentação teórica e os trabalhos que estão diretamente relacionados ao tema proposto neste projeto. Este capítulo irá apresentar a proposta deste trabalho, descrever as adaptações realizadas no software que foi utilizado para processar os dados e também descrever as adaptações realizadas no algoritmo HDBSCAN*. A utilização do HDBSCAN* como algoritmo de agrupamento para este projeto, deve-se porque dentre os algoritmos analisados, este é o que possui as características mais próximas para se trabalhar com a base de dados CATWEB do Ministério Público do Trabalho.

A proposta deste trabalho foi apoiar o projeto de mestrado "Técnicas de Agrupamento aplicadas aos Dados de Acidente de Trabalho"(GIACOMELLI, 2020), realizando adaptações no Algoritmo HDBSCAN*. As adaptações no HDBSCAN*, em uma visão macro, visam permitir o processamento dos atributos categóricos nominais, implementando os cálculos das distâncias considerando tais atributos. Todos os detalhes dessas adaptações estão descritas na seção 3.4. Para o pré-processamento dos dados, utilizou-se um software desenvolvido na linguagem Java (SILVA, 2018), onde foram implementadas novas funções de pré-processamento e adaptadas algumas já existentes. Todas as informações sobre este software e as mudanças nele realizadas estão na seção 3.3.

O primeiro passo foi analisar a base de dados e desenvolver soluções para que o HDBSCAN* pudesse calcular as distâncias dos atributos categóricos nominais, pois o mesmo estava preparado para atributos numéricos. Para cada atributo categórico, a proposta foi analisar e desenvolver o melhor cálculo de distância correspondente e implementá-lo separadamente no algoritmo, a fim de que quando o algoritmo execute a coluna daquele atributo, aplique o cálculo apropriado de distância.

As funções de cálculo de distância implementadas são as seguintes: função para cálculo da distância entre atributos do tipo real, função para cálculo da distância entre atributos binários (critério da dissimilaridade), função para cálculo de distância entre atributos nominais usando uma lista circular. Ao final é calculada a média aritmética entre todos os atributos. Todos os detalhes sobre as implementações dessas funções de distâncias estão na subseção 3.4.2. Todas essas funções implementadas substituíram a função de cálculo de distância Euclidiana que veio implementada no algoritmo utilizado como base.

O segundo passo foi rodar os experimentos após as implementações e adaptações

no código. Os resultados dos agrupamentos obtidos foram analisados no trabalho (GIACOMELLI, 2020) e estas análises ajudaram a direcionar novas mudanças para refinar o algoritmo. As seções a seguir descrevem todo o desenvolvimento deste projeto.

## 3.2 Base de Dados CATWEB

A base CATWEB é a base de dados do Ministério Público do Trabalho contendo informações dos acidentes de trabalho de todo Brasil notificadas via CAT (Comunicado de Acidente de Trabalho). É constituída em sua maioria de atributos categóricos nominais e possuí quase 4 milhões de instâncias/objetos contendo acidentes no período de 2012 a 2017, que foi período definido para o projeto (GIACOMELLI, 2020).

A base possui 18 atributos e utilizar todos os atributos na mineração não seria uma boa ideia, pois existem muitos atributos que são irrelevantes para o trabalho de acordo com (GIACOMELLI, 2020). Além disso, como um dos objetivos é interpretar os resultados, quanto mais atributos, mais difícil torna a interpretação e sendo assim optou-se por iniciar com o mínimo de atributos possível, sendo escolhidos os mais significativos para o problema.

Outra motivação para se pensar na diminuição do número de atributos é a complexidade computacional, pois a presença de atributos irrelevantes impacta negativamente no tempo de processamento, pois a base possui 3.879.755 instâncias e como se sabe, a maioria dos algoritmos de *clustering* são impactados pelo número instâncias, mas também pelo número de atributos da base. Como o algoritmo HDBSCAN* precisa encontrar a distância entre todos os pares de objetos da base, e como a medida de distância é calcula para cada atributo, quanto mais atributos, mais tempo será gasto na sua execução.

Um ponto importante a ser considerado sobre os atributos é a conversão dos atributos nominais para numéricos. Como o HDBSCAN* é um algoritmo que trabalha com atributos numéricos, seria necessário converter todos os atributos categóricos nominais em numéricos. Como consequência, essa conversão aumentaria o número de atributos. Considerando um dos algoritmos mais clássicos de conversão, o 1-de-n, cada categoria do atributo seria representada por um atributo numérico (ver seção 2.4.2). Por exemplo, o atributo feriado, que possui 12 valores encontrados na base, seria convertido para 12 atributos, cada um representando um feriado diferente.

**Ex:**

**Feriados da base** = **1** atributo categórico nominal e **12** atributos numéricos

Considerando a conversão de todos os atributos nominais da base CATWEB para

numéricos, aumentaria muito a dimensionalidade da base, o que inviabilizaria a utilização de vários algoritmos de agrupamento.

Também levou-se em conta a quantidade de memória, pois como todos os dados computados são armazenados na memória, por mais que as máquinas atuais disponham de um tamanho grande de memória, deve-se levar esse fator em consideração para não haver um estouro de memória durante a execução do processo.

Considerando todos esses fatores, o trabalho (GIACOMELLI, 2020) optou por escolher sete atributos considerados fundamentais para o propósito, os quais foram usados neste TCC. Na Tabela 3.2, a coluna **Atributo** é uma transcrição do cabeçalho referente a cada atributo da base de dados. A coluna **Tipo** é o tipo do atributo. A coluna **Útil** informa se o atributo foi ou não selecionado para a mineração. E por fim, na coluna **Definição**, tem-se a definição do que significa o atributo, lembrando-se que em várias instâncias esses atributos têm informações nulas ou vazias.

## 3.3 Pré-Processamento dos Dados do MPT

Ao analisar os dados da base CATWEB do MPT, verificaram-se várias inconsistências que podem afetar o resultado da tarefa de mineração. Estas inconsistências vão desde atributos com dados nulos ou vazios até a falta de padronização dos dados. Os atributos categóricos nominais foram os mais problemáticos, pois a escrita não padronizada dificultou o processo. Como exemplo, pode-se citar o atributo `nm_cidade`. Os nomes das cidades muitas vezes são digitados erroneamente, sendo assim, uma cidade tem seu nome escrito de várias formas diferentes. Sem uma correção desses dados, essas inconsistências mudariam o resultado da mineração. Na Tabela 3.3, pode ser visto exemplos de dados corrigidos da base do MPT.

Para correção e transformação de dados, este trabalho utilizou um software desenvolvido por Danilo Silva (SILVA, 2018). Este software foi construído utilizando a linguagem de programação JAVA, é uma aplicação *desktop* e foi desenvolvido para uma base similar a esta, a de benefícios previdenciários. Para corrigir os atributos da base de dados CATWEB, foi necessário desenvolver novos módulos no software e para a transformação de atributos foi necessário adaptar as funções já existentes.

As transformações foram feitas para converter o tipo de alguns atributos, por exemplo, o atributo categórico `cd_tipo_sexo_empregado_cat`, que foi transformado para numérico/binário (masculino = 0 e feminino = 1), de forma que fossem processados pelo algoritmo HDBSCAN*.

Todas as alterações seguiram o mesmo padrão desenvolvido por (SILVA, 2018) e não foi alterado a usabilidade do software original, bastando o usuário fornecer o caminho

Tabela 1 – Atributos da base de dados CATWEB.

| **Atributo** | **Tipo** | **Útil** | **Definição** |
|---|---|---|---|
| st_acidente_feriado | Categórico nominal | Não | Indica se o acidente aconteceu no feriado e qual foi o feriado. |
| ds_agente_causador | Categórico nominal | Sim | O que causou o acidente como: queda, máquina agrícola, etc. |
| ano_cat | Numérico intervalar | não | Ano em que foi registrado o CAT. |
| ds_cnae_classe_cat | Categórico nominal | Sim | Classificação Nacional da Atividade Econômica do trabalhador. |
| dt_acidente | Numérico Intervalar | Não | Data em que aconteceu o acidente. |
| st_dia_semana_acidente | Categórico ordinal | Não | Dia da semana em que ocorreu o acidente. |
| ds_emitente_cat | Categórico nominal | Não | Quem registrou o CAT. Ex: empregado, empregador, etc. |
| hora_acidente | Numérico intervalar | Não | Hora em que ocorreu o acidente. |
| idade_cat | Numérico Racional | Sim | Idade do trabalhador que sofreu o acidente. |
| cd_indica_obito | Binário | Não | Indica se o acidentado morreu ou não. |
| nm_municipio | Categórico nominal | Não | Cidade onde ocorreu o acidente de trabalho. |
| nome_uf | categórico nominal | Não | Estado onde ocorreu o acidente de trabalho. |
| ds_natureza_lesao | Categórico nominal | Não | Tipo de lesão, exemplo: escoriação, fratura, etc. |
| ds_cbo | Categórico nominal | Não | Classificação brasileira da ocupação (cbo) do trabalhador. |
| ds_parte_corpo_atingida | Categórico nominal | Sim | Parte do corpo do trabalhador atingida ou impactada pelo acidente. |
| cd_tipo_sexo_empregado_cat | Categórico nominal | Sim | Sexo do acidentado. |
| ds_tipo_acidente | Categórico nominal | Sim | Tipo do acidente. |
| ds_tipo_local_acidente | Categórico nominal | Sim | Local onde ocorreu o acidente de trabalho. |

Fonte: Elaborado pelo autor (2022)

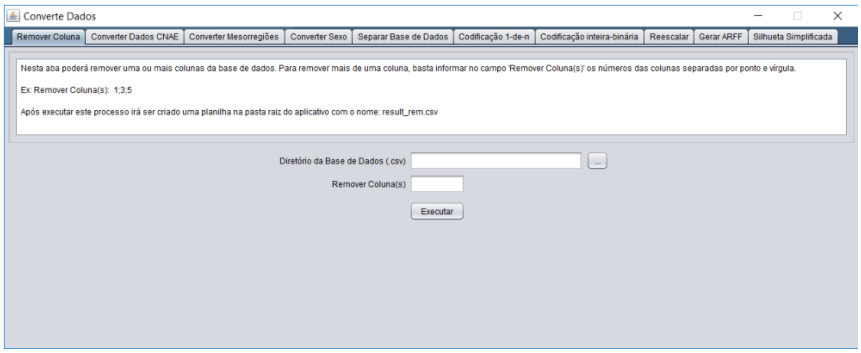
Tabela 2 – Exemplo de correção dos dados para o atributo Nome da Cidade.

| Atributo 1 | Atributo 2 | Atributo Corrigido |
|---|---|---|
| Eldorado dos Carajas | Eldorado do Carajas | Eldorado do Carajas |
| Brasopolis | Brazopolis | Brazopolis |
| Itapage | Itapaje | Itapaje |
| Serido | Sao Vicente do Serido | Sao Vicente do Serido |
| Poxoreo | Poxoreu | Poxoreu |

Fonte: Elaborado pelo autor (2022)

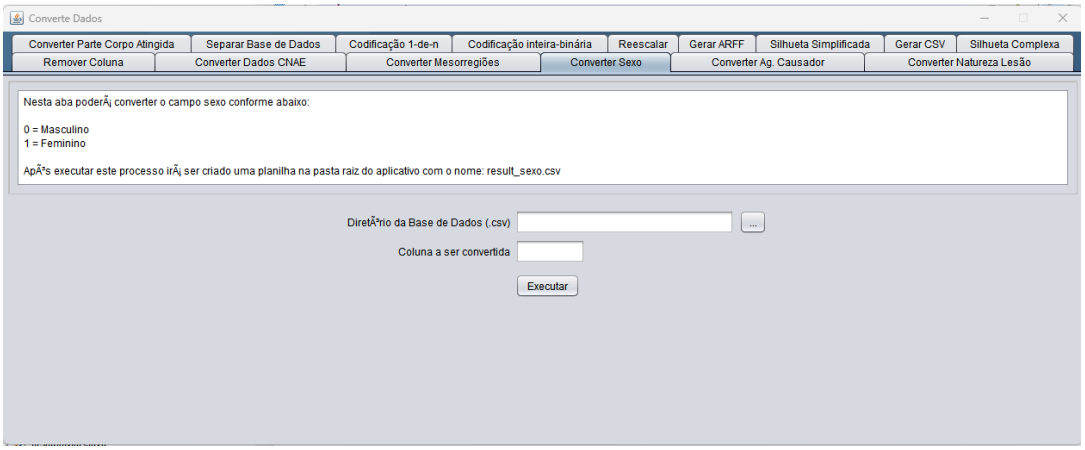
da planilha dos dados e indicar o número da coluna a ser processada. Na Figura 9 tem-se a tela principal do software original e na Figura 10 tem-se a tela principal do software com as modificações desenvolvidas neste trabalho.

Figura 9 – Software desenvolvido por Danilo.




Fonte: (SILVA, 2018)

Figura 10 – Software modificado para tratar a base de dados do (MPT).




Fonte: Elaborado pelo autor (2022)

A seguir, segue as implementações realizadas para a transformação dos dados.

- **Converter Mesorregiões**: Esta implementação já existia, sendo necessário realizar uma modificação para atender a base CATWEB. Esta modificação foi implementar

uma função para realizar correções nos nomes de cidades que estavam incorretos, devido a erro de escrita para a forma correta. Esta função cria uma tabela com um novo atributo chamado de `nm_mesorregiao`. Mesorregião, segundo o Instituto Brasileiro de Geografia e Estatística (IBGE), são subdivisões dos estados brasileiros, que consideram semelhanças nos aspectos econômicos e sociais (ESTATÍSTICA, 1990). Nela, considerou os critérios estabelecidos pelo IBGE. Para utilizar esta função é necessário baixar a planilha de mesorregiões contidas no site do Datasus, que contém as colunas cidade, estado e mesorregião. Após a conversão, os 5.285 municípios existentes na base, foram convertidos em 165 mesorregiões distintas.

- **Converter Natureza da Lesão**: Esta implementação consistiu em mapear os dados da coluna `ds_natureza_lesao` e realizar a remoção de acentuação.

- **Converter Parte do Corpo Atingida**: Esta implementação consistiu em mapear todos os dados sobre o atributo `ds_parte_corpo_atingida` da base de dados CATWEB e depois agrupá-los em 5 grupos: cabeça, pescoço, tronco, membros inferiores e membros superiores conforme a divisão da anatomia humana. Com essa informação criou-se uma tabela chamada parte do corpo atingida.

- **Agrupar o Ag. Causador**: Esta implementação consistiu inicialmente em mapear todos os agentes causadores da base, que resultou em um total de 302 itens. A partir disso eles foram agrupados por semelhanças, o que resultou em 22 grupos. Com essas informações, criou-se uma tabela com os dados do grupo e do agente causador. Em seguida implementou-se a função de conversão que utiliza tal tabela como parâmetro de entrada para conversão da coluna `ds_agente_causador`.

- **Silhueta Complexa**: Este foi o nome escolhido neste trabalho para a implementação que adaptou o método de validação da silhueta simplificada. De acordo com (WANG et al., 2017), Silhueta é uma das medidas internas mais populares e eficazes para avaliar a qualidade de um agrupamento. Silhueta simplificada é uma versão computacionalmente simplificada da Silhueta. O algoritmo 2 é uma representação básica desta medida de validação.

  Para atendar as características da base dos Acidentes de Trabalho do Ministério Público (CATWEB), foi necessário alterar a silhueta simplificada. A mudança basicamente foi trocar a função que calcula a distância. Na silhueta simplificada está implementada a distância Euclidiana. Na silhueta complexa implementou-se a mesma medida de distância que foi implementada neste trabalho e usada no HDBSCAN* (descrita na seção 3.4).

**Algorithm 2** - Algoritmo Silhueta Simplificada.

`Inicio`

**enquanto** $i < N$ onde $N$ é o tamanho da amostra **faça**

1º. $a_i$ = Para o i-ésimo objeto, calcule a sua distância média para todos os outros objetos em seu cluster.

2º. $b_i$ = Para o i-ésimo objeto e qualquer cluster que não contenha o objeto, calcule a distância média do objeto para todos os objetos no cluster fornecido. Encontre o mínimo desse valor em relação a todos os clusters.

3º. $S_i = \frac{(b_i - a_i)}{max(a_i, b_i)}$. Cálculo da silhueta para o i-ésimo objeto.

**fim enquanto**

4º. $silhueta = \frac{\sum_{i=1}^{N} S_i}{N}$

return $silhueta$

O valor da silhueta simplificada pode variar entre -1 e 1.

$Fim$

Fonte: adaptado de (TAN; STEINBACH; KUMAR, 2005)

## 3.4 Adaptações no HDBSCAN*

Os algoritmos de mineração são desenvolvidos para trabalhar usando um tipo de dado específico, o que geralmente leva os usuários e pesquisadores a realizarem transformações nos tipos de dados. Esse é o caso deste trabalho, o qual teve que adequar as funções que calculam as distâncias usadas no algoritmo HDBSCAN*, para que os atributos categóricos pudessem ser processados.

### 3.4.1 Leitura da Base

A primeira mudança implementada no algoritmo, foi na função que lê o arquivo da base de dados. A implementação original carrega a base de dados na memória e em seguida começa a construir a matriz de distância entre cada par de objetos e vai alocando-a na memória. Esta técnica pode ser do ponto de vista computacional mais eficiente, porém para o tamanho da base de dados de Acidentes de Trabalho do Ministério Público (CATWEB), se tornou inviável para este projeto, pois estourava os 8GB de memória RAM das máquinas utilizadas. Além de alocar a matriz de distâncias, ainda estava também sendo alocada a base de dados em simultâneo.

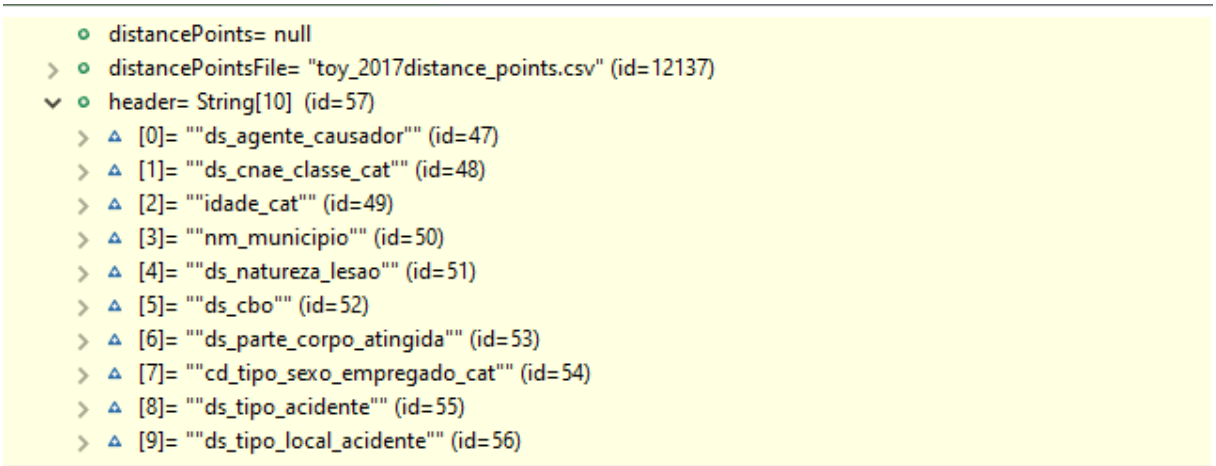
A solução implementada foi criar um arquivo com a matriz de distância. Cada

operação realizada é gravada no arquivo e sendo assim não consome memória para alocar a matriz de distâncias simultaneamente com a base de dados carregada. Ao final do processo temos um arquivo texto com as informações da matriz de distância.

Para o cálculo da matriz de distância, o arquivo é aberto e toda operação é realizada mediante a leitura do arquivo. O processo é mais lento, mas resolveu-se o problema de estouro de memória nas máquinas que foram utilizadas. Para os demais processos, a opção de salvar os dados na memória foi usada.

Outra implementação realizada, foi criar um identificador de atributos para identificar qual o atributo está sendo analisado, e sendo assim aplicar a função de cálculo de distância específica para o atributo. Para esta operação, a base de dados deve conter o nome do atributo no cabeçalho de cada coluna. Na Figura 11 pode-se ver o parâmetro *header* contendo a lista de atributos. É possível identificar o nome de cada atributo e a sua posição na lista.

Figura 11 – Lista com os nomes dos atributos de cada coluna da base de dados.




Fonte: Elaborado pelo autor (2022)

Nesta etapa, pode-se citar como modificações importantes a implementação da leitura dos rótulos das colunas e a substituição da alocação da matriz na memória pela criação de um arquivo com os dados desta matriz. Na implementação original não é considerada a entrada com o rótulo dos atributos no cabeçalho/header da base de dados, pois como ele só trata atributos numéricos, só é necessário um único algoritmo de distância. É importante destacar que na adaptação realizada, tomou-se o cuidado para não deixar o algoritmo lento, com mais iterações, o que poderia aumentaria o seu custo de processamento.

Considerando as transformações em atributos, para o atributo `idade_cat` em um primeiro momento considerou-se tratar este atributo como numérico. Sendo assim, foi necessário criar as variáveis *"maxValue"* para capturar o seu maior valor encontrado na base e *"minValue"* para capturar o seu menor valor contido na base de dados CATWEB. Estas

informações eram necessárias para determinar o limite superior e inferior do conjunto de dados do atributo `idade_cat` para aplicação na primeira função de cálculo empregada. Essa abordagem foi utilizada inicialmente, mas descartada nos testes seguintes, onde o atributo foi convertido em nominal e passou a utilizar a abordagem do Critério da Dissimilaridade Binária, conforme Tabela 3.4.2.1.

Outras implementações importantes realizadas foram as diferentes funções para os cálculos da distância entre os objetos, as quais serão vistas na subseção 3.4.2 a seguir.

### 3.4.2 Cálculo da Distância

Um dos desafios deste projeto foi implementar as medidas para calcular a dissimilaridade entre dois objetos considerando os diferentes tipos de atributos da base. A implementação original calcula a dissimilaridade utilizando a distância Euclidiana. Para isso, a implementação considera todos os dados como atributos numéricos. Como os atributos da base CATWEB são atributos categóricos nominais, em sua maioria, é necessário aplicar medidas que consideram tais características, sendo assim, foi empregado funções de cálculo de distância para cada tipo de atributo.

Para atributos categóricos nominais, o critério da dissimilaridade adotado foi da dissimilaridade binária, conforme apresentado no Algoritmo 3. Para os atributos numéricos, a distância é calculada conforme a Equação 3.2. Para o atributo categórico ordinal, adotou-se o critério da lista circular, que pode ser observado na Equação 3.1. Cada uma dessas estratégias são apresentadas a seguir.

Todas essas alterações tiveram que ser cuidadosamente planejada para se enquadrar dentro das características do algoritmo, levando-se em consideração a tentativa de otimizar todo o processo.

#### 3.4.2.1 Critério da Dissimilaridade Binária

Para o uso do critério da dissimilaridade binária, calcula-se a dissimilaridade para atributos categóricos nominais comparando se o atributo a ser considerado na instância $x$ é igual ou não ao atributo da instância $y$. O valor retornado é 0 se forem iguais e 1 se forem diferentes. No Algoritmo 3 tem-se esta representação detalhada, onde $x_i$ e $y_j$ são atributos de entrada, onde $x_i$ representa o i-ésimo atributo da instância $x$ e $y_j$ representa o j-ésimo atributo da instância $y$ vizinha a ser comparada.

No atributo `idade_cat`, que originalmente é numérico, inicialmente trabalhou-se com ele neste formato, mas depois optou-se por transformá-lo em categórico nominal, utilizando uma faixa composta por 4 grupos (menor idade, jovem adulto, adulto e idoso) conforme Tabela 3.4.2.1.

Tabela 3 – Classificação faixa etária

| Menor Idade | Jovem Adulto | Adulto | Idoso |
|---|---|---|---|
| idade < 18 | idade >= 18 < 35 | idade >= 35 < idade 55 | idade >= 55 |

Fonte: Elaborado pelo autor (2022)

Após a classificação da idade em faixa etária, conforme Tabela 3.4.2.1, utiliza-se o critério da dissimilaridade binária. Essa abordagem de distância utilizando o critério da dissimilaridade binária foi empregada em todos os atributos categóricos nominais.

**Algorithm 3** Algoritmo critério da dissimilaridade.

**Entrada:** $x_i$
**Entrada:** $y_j$
  **se** $x_i == y_j$ **então**
    $distancia \leftarrow 0$
  **senão**
    $distancia \leftarrow 1$
  **fim se**
**Saída:** $distancia$

#### 3.4.2.2 Lista Circular

Definiu-se como Lista Circular todo atributo categórico ordinal com segmentos cíclicos, que podem ser representados como atributos numéricos. Como exemplo, os dias da semana e os meses do ano. Neste trabalho foi implementado este método para o cálculo de distância no algoritmo para o atributo `st_dia_semana_acidente` referente aos dias da semana. Na tabela 3.4.2.2 tem-se a representação numérica considerada na lista e o dia da semana o qual esse número representa.

Tabela 4 – Representação dos dias da semana (Lista Circular)

| Dia Semana | Representação Numérica |
|---|---|
| Domingo | 0 |
| Segunda-feira | 1 |
| Terça-feira | 2 |
| Quarta-feira | 3 |
| Quinta-feira | 4 |
| Sexta-feira | 5 |
| Sábado | 6 |

Fonte: Elaborado pelo autor (2022)

A Equação 3.1 é a função utilizada para os cálculos de distância da lista circular, adaptada para o atributo `st_dia_semana_acidente`. Esta equação correspondente à

menor distância entre dois dias da semana, não importando a ordem dos fatores, onde, $x_{diasemana}$ corresponde ao valor do atributo Dia da semana da primeira instância, $y_{diasemana}$ corresponde ao valor do atributo Dia da semana da segunda instância e $Tam$ representa o tamanho da lista.

$$d = min(|x_{diasemana} - y_{diasemana}|, (Tam - |y_{diasemana} - x_{diasemana}|)) \tag{3.1}$$

Na Tabela 3.4.2.2 têm-se alguns exemplos de distâncias calculadas:

Tabela 5 – Exemplos de distâncias calculadas com a função 3.1

| Comparações | Distância |
|---|---|
| segunda x quarta | 2.0 [1,3] |
| quarta x segunda | 2.0 [3,1] |
| sábado x domingo | 6.0 [6,0] |
| domingo x sábado | 6.0 [0,6] |

Fonte: Elaborado pelo autor (2022)

#### 3.4.2.3 Critério da Distância Numérica

Outra medida que foi utilizada, aqui chamada de distância numérica, foi criada para os atributos do tipo numérico (inteiro e real) como, por exemplo, o atributo `idade_cat`. A fim de produzir um valor numérico entre 0 e 1, assim como foi feito nas demais medidas de distância, exceto na lista circular, cada dado deve ser re-escalado para o intervalo entre 0 e 1 antes de calcular a distância.

Esta operação está representada na Equação 3.2 onde $x_i$ é o i-ésimo atributo da instância $x$ e $y_i$ é o i-ésimo atributo da instância $y$, $max$ é o maior valor encontrado na base para o atributo $i$ e $min$ é o menor valor do atributo $i$ contido na base de dados.

$$d = |(\frac{x_i - min}{max - min}) - (\frac{y_i - min}{max - min})| \tag{3.2}$$

#### 3.4.2.4 Composição da Distância

Após o cálculo da distância para cada atributo, é necessário realizar a composição das distâncias em relação a todos os atributos da base de dados. Para isso adotou-se a média aritmética conforme Equação 3.3.

$$d(x, y) = \frac{\sum_{i=1}^{n} d_i(x, y)}{n} \tag{3.3}$$

Na Equação 3.3, $n$ é o número de atributos, $d_i(x, y)$ a distância entre as instância $x$ e $y$ em relação ao atributo $i$. Vale destacar que o emprego da média aritmética é uma recomendação da literatura clássica de agrupamento para abordagens não ponderadas (TAN; STEINBACH; KUMAR, 2005), o que implica que todos os atributos têm o mesmo peso.

É importante frisar que todas as adaptações realizadas só contemplam os segmentos do algoritmo que computam as distâncias. As etapas que envolvem a criação dos clusters, desde o cálculo da hierarquia, construção e propagação da árvore e cálculo de *outliers* não sofreram adaptações.

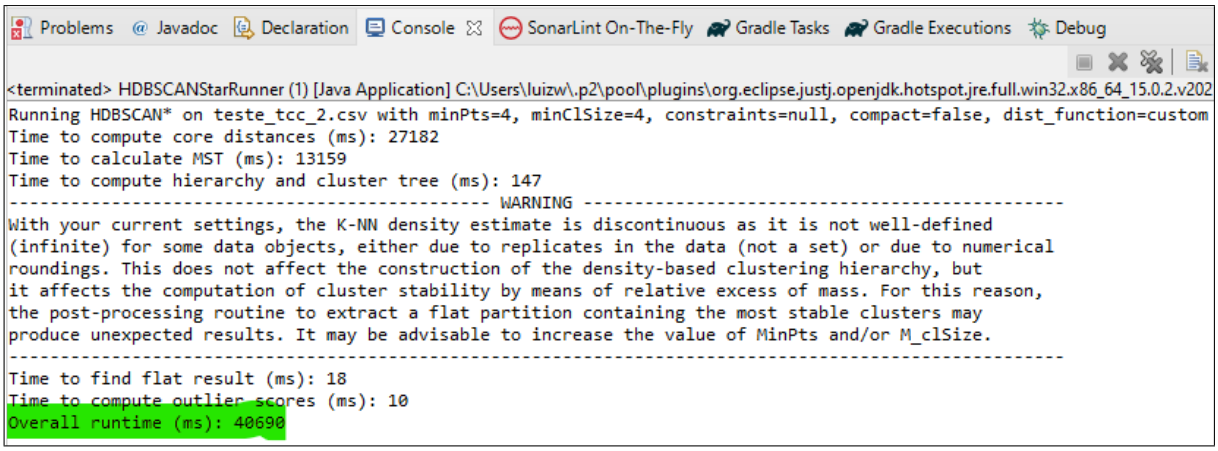
#### 3.4.2.5 Tempo de Processamento

Como foram realizadas várias implementações e adaptações no algoritmo HDBSCAN*, uma das preocupações era não aumentar o tempo de processamento do algoritmo. Como foram feitas modificações que envolvem a escrita e leitura em arquivos, e como essas são operações que envolvem um tempo extra, foi necessário verificar como o algoritmo seria impactado pelas mesmas.

A Figura 12 mostra o tempo de processamento gasto pelo algoritmo usar uma base de dados contendo 4608 instâncias. O tempo de execução para essa base foi de 40690 milissegundos. A Figura 13 é referente a execução do algoritmo em uma base contendo 9216 instâncias, cujo tempo total de execução foi de 179671 milissegundos.

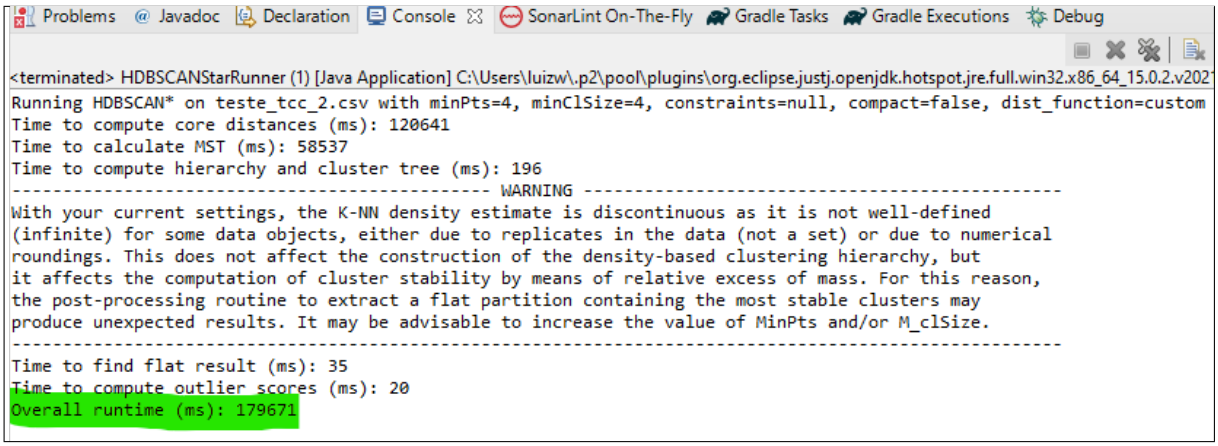
É importante notar que ao dobrar o número de instâncias, o tempo de execução aumentou aproximadamente 4 vezes. Mas mesmo com esse aumento, ainda foi factível executar os experimentos necessários na base CATWEB.

Figura 12 – Processamento com 4608 instancias.




Fonte: Elaborado pelo autor (2023)

Figura 13 – Processamento com 9216 instancias.




Fonte: Elaborado pelo autor (2023)

## 3.5 Considerações Finais

Neste capítulo foram apresentados os detalhes da base de dados CATWEB e as características dos seus atributos. Também foram apresentadas as dificuldades para lidar com tais dados, sendo uma delas a falta de padronização que os mesmos possuem. Além disso, foi apresentado o software de pré-processamento de dados que trata tais problemas. Foram também apresentadas as mudanças que foram realizadas no algoritmo HDBSCAN* para atender a base de dados.

Este trabalho não apresentará os resultados do experimento com a base CATWEB e o algoritmo HDBSCAN*, pois o objetivo deste trabalho era apoiar o projeto de mestrado (GIACOMELLI, 2020), cooperando na tomada das decisões das estratégias a serem adotadas e implementando as mudanças no algoritmo HDBSCAN*. Todos os resultados dos experimentos estão presentes na dissertação (GIACOMELLI, 2020).

O próximo capítulo apresentará as principais contribuições deste trabalho e ideias para possíveis trabalhos futuros.

# 4 Conclusão

Este trabalho apresenta as modificações realizadas na implementação do algoritmo HDBSCAN* para que o mesmo pudesse ser aplicado na base de dados CATWEB de acidentes de trabalho. Como a maioria dos atributos são categóricos, várias alterações foram realizadas para que o algoritmo pudesse ser executado em uma base de dados grande, como a CATWEB. Também foram desenvolvidas técnicas para cálculo da distância entre dois objetos, conforme o tipo dos atributos. Ao final criou-se um cálculo de distância entre objetos que produz resultados entre 0 e 1 para cada atributo, independente do seu tipo, e calcula a média desses valores.

Várias atualizações foram realizadas no software de pré-processamento (SILVA, 2018) para que o mesmo pudesse atender a base de dados CATWEB. Foram utilizadas muitas variações da base de dados CATWEB, com trocas de atributos, análises de quais atributos seriam relevantes, a fim de diminuir a dimensionalidade sem impactar na qualidade do resultado final.

Como resultado, (GIACOMELLI, 2020) observa que as técnicas utilizadas para o cálculo das distâncias não foram eficientes, pois ao que tudo indica, a utilização de valores de distâncias 0 e 1 para os dados categóricos levou o algoritmo ao resultado insatisfatório. Analisando as estratégias adotadas, nota-se que para futuros projetos, deve-se considerar trabalhar as distâncias dos atributos categóricos de forma independente utilizando novas estratégias de cálculos para cada atributo e também avaliar a equação que compõe a distância final.

## 4.1 Contribuições

As seguintes contribuições foram desenvolvidas por este trabalho para os dados do CATWEB:

1. Atualização do software de pré-processamento de dados para processar os dados do CATWEB;
2. Adaptação da implementação do algoritmo HDBSCAN* para o processamento dos atributos categóricos da base CATWEB;
3. Implementação de uma nova medida de cálculo de distância entre objetos da base CATWEB.

## 4.2 Considerações finais e trabalhos futuros.

A partir do resultado obtido, pode-se definir os próximos passos a serem seguidos como possíveis trabalhos futuros:

1. Utilizar novas técnicas no cálculo das distâncias dos atributos categóricos, bem como avaliar os atributos quanto a sua semântica.
2. Continuar a investigação pela busca de padrões utilizando o algoritmo HDBSCAN* usando a base de dados CATWEB de acidentes de trabalho.
3. Avaliar se a média aritmética é uma medida ideal para calcular a composição da distância em relação aos atributos da mesma instância.

# Referências

ARMSTRONG, R. A. When to use the bonferroni correction. *Ophthalmic and Physiological Optics*, v. 34, n. 5, p. 502–508, 2014. Disponível em: <https://onlinelibrary.wiley.com/doi/abs/10.1111/opo.12131>. Citado na página 19.

BALAHADIA, F. F. et al. Application of spatiotemporal analysis and knowledge discovery for databases in the bureau of fire protection as incident report system: Tool for improving fire services. *International Journal of Computing Sciences Research*, v. 5, p. 519–533, 2020. Citado na página 16.

BERRY, M.; LINOFF, G. *Data Mining Techniques: For Marketing, Sales, and Customer Support*. [S.l.]: John Wiley, 1997. ISBN 9780471179801. Citado na página 15.

BINDRA, K.; MISHRA, A. A detailed study of clustering algorithms. In: *2017 6th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO)*. [S.l.: s.n.], 2017. p. 371–376. Citado na página 23.

BITTENCOURT, F. *Brasil ocupa quarta posição no ranking de acidentes de trabalho*. 2019. Disponível em: <https://atarde.com.br/empregos/brasil-ocupa-quarta-posicao-no-ranking-de-acidentes-de-trabalho-1054181>. Acesso em: 21 jan. 2022. Citado na página 13.

BRITO, L. L. *A strategy for temporal visual analysis of labor accident data - Dissertação de Mestrado*. Universidade Federal de Uberlândia, 2019. Disponível em: <https://repositorio.ufu.br/handle/123456789/28278>. Citado na página 15.

Buntine, W. A guide to the literature on learning probabilistic networks from data. *IEEE Transactions on Knowledge and Data Engineering*, v. 8, n. 2, p. 195–210, April 1996. ISSN 1041-4347. Citado na página 20.

CAMPELLO, R. J. G. B.; MOULAVI, D.; SANDER, J. Density-based clustering based on hierarchical density estimates. In: PEI, J. et al. (Ed.). *Advances in Knowledge Discovery and Data Mining*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013. p. 160–172. ISBN 978-3-642-37456-2. Citado 3 vezes nas páginas 10, 27 e 30.

CAMPELLO, R. J. G. B. et al. Hierarchical density estimates for data clustering, visualization, and outlier detection. *ACM Trans. Knowl. Discov. Data*, Association for Computing Machinery, New York, NY, USA, v. 10, n. 1, jul 2015. ISSN 1556-4681. Disponível em: <https://doi.org/10.1145/2733381>. Citado na página 28.

ESTATÍSTICA, I. B. de Geografia e. *Divisão do Brasil em mesorregiões e microrregiões geográficas*. Instituto Brasileiro de Geografie e Estatística - Rio de Janeiro, 1990. ISBN 2408456300. Disponível em: <https://biblioteca.ibge.gov.br/index.php/biblioteca-catalogo?view=detalhes&id=22269>. Citado na página 37.

ESTER, M. et al. A density-based algorithm for discovering clusters in large spatial databases with noise. In: *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*. [S.l.]: AAAI Press, 1996. (KDD'96), p. 226–231. Citado na página 27.

FAYYAD, U.; PIATETSKY-SHAPIRO, G.; SMYTH, P. From data mining to knowledge discovery in databases. *AI Magazine*, v. 17, n. 3, p. 37, Mar. 1996. Disponível em: <https://ojs.aaai.org/aimagazine/index.php/aimagazine/article/view/1230>. Citado 5 vezes nas páginas 16, 17, 18, 19 e 20.

FAYYAD, U.; PIATETSKY-SHAPIRO, G.; SMYTH, P. Knowledge discovery and data mining: Towards a unifying framework. In: *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*. IEEE Computer Society, 1996. (KDD'96), p. 82–88. ISSN 1941-1294. Disponível em: <https://www.aaai.org/Papers/KDD>. Citado na página 16.

FENG, Y. et al. Knowledge discovery in traditional chinese medicine: State of the art and perspectives. *Artificial Intelligence in Medicine*, v. 38, n. 3, p. 219–236, 2006. ISSN 0933-3657. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0933365706001047>. Citado na página 16.

FREITAS, E. d. *Industrialização do Brasil*. 2021. Disponível em: <https://brasilescola.uol.com.br/brasil/industrializacao-do-brasil.htm>. Acesso em: 10 dez. 2022. Citado na página 9.

GIACOMELLI, D. F. Técnicas de agrupamento de dados aplicadas aos dados de acidente de trabalho - dissertação de mestrado. Universidade Federal de Uberlândia, 2020. Disponível em: <https://repositorio.ufu.br/handle/123456789/29531>. Citado 9 vezes nas páginas 3, 10, 11, 30, 32, 33, 34, 44 e 45.

GLOBO, J. O. *Acidentes de trabalho custaram R$ 26 bilhões à Previdência entre 2012 e 2017, diz MPT*. 2018. Disponível em: <https://g1.globo.com/economia/noticia/acidentes-de-trabalho-custaram-r-26-bi-a-previdencia-entre-2012-e-2017.ghtml>. Acesso em: 24 out. 2021. Citado na página 9.

HAN, J.; KAMBER, M.; PEI, J. *Data Mining: Concepts and Techniques*. 3rd. ed. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2011. ISBN 0123814790. Citado 9 vezes nas páginas 16, 20, 21, 22, 23, 24, 25, 27 e 28.

HECKERMAN, D. Bayesian networks for data mining. *Data Min. Knowl. Discov.*, v. 1, p. 79–119, 1997. Citado na página 20.

IBGE. *Desemprego cai para 11,8%, mas 12,6 milhões ainda buscam trabalho*. 2019. Disponível em: <https://agenciadenoticias.ibge.gov.br/agencia-noticias/2012-agencia-de-noticias/noticias/25314-desemprego-cai-para-11-8-mas-12-6-milhoes-ainda-buscam-trabalho>. Acesso em: 10 dez. 2022. Citado na página 14.

LABORE. *Como investir para reduzir custos em Saúde e Segurança do Trabalho?* 2021. Disponível em: <http://laboreweb.com.br/como-investir-para-reduzir-custos-em-saude-e-seguranca-do-trabalho/>. Acesso em: 05 jan. 2023. Citado na página 15.

LAROSE, D. T.; LAROSE, C. D. *Discovering knowledge in data: an introduction to data mining*. [S.l.]: John Wiley & Sons, 2014. v. 4. Citado na página 16.

LEECE, D. Applying data visualization and knowledge discovery in databases to segment the market for risky financial assets. *Managerial and Decision Economics*, Wiley Online Library, v. 20, n. 5, p. 267–280, 1999. Citado na página 17.

MAJOR, J. A.; MANGANO, J. J. Selecting among rules induced from a hurricane database. *Journal of Intelligent Information Systems*, Springer, v. 4, n. 1, p. 39–52, Jan 1995. ISSN 1573-7675. Citado na página 20.

NOGUEIRA, D. P. Prevention of accidents and injuries in brazil. *Ergonomics*, Taylor e Francis, v. 30, n. 2, p. 387–393, 1987. PMID: 3582351. Disponível em: <https://doi.org/10.1080/00140138708969723>. Citado na página 9.

RODRIGUES, M. P. *A strategy for temporal visual analysis of labor accident data - Dissertação de Mestrado*. Universidade Federal de Uberlândia, 2019. Disponível em: <https://repositorio.ufu.br/handle/123456789/28282>. Citado na página 15.

SILVA, D. A. d. *Aplicação de técnicas de pré-processamento e agrupamento na base de dados de benefícios previdenciários do ministério público do trabalho - Trabalho de Conclusão de curso*. Universidade Federal de Uberlândia, 2018. 66 p. Disponível em: <https://repositorio.ufu.br/handle/123456789/22118>. Citado 8 vezes nas páginas 10, 11, 15, 30, 32, 34, 36 e 45.

SMARTLAB. *Observatorio Digital De Segurança E Saúde No Trabalho*. 2019. Disponível em: <https://smartlabbr.org/sst>. Acesso em: 14 setembro. 2019. Citado 5 vezes nas páginas 9, 13, 14, 15 e 30.

SöRENSEN, K.; JANSSENS, G. K. An algorithm to generate all spanning trees of a graph in order of increasing cost. *Pesquisa Operacional*, SciELO Brasil, v. 25, n. 2, p. 219–229, ago. 2005. Disponível em: <https://doi.org/10.1590/s0101-74382005000200004>. Citado na página 29.

SOUZA, R. *Brasil tem 700 mil acidentes de trabalho por ano*. 2017. Disponível em: <https://www.em.com.br/app/noticia/economia/2017/06/05/internas_economia,874113/brasil-tem-700-mil-acidentes-de-trabalho-por-ano.shtml>. Acesso em: 21 jan. 2023. Citado na página 9.

STRUYF, A.; HUBERT, M.; ROUSSEEUW, P. Clustering in an object-oriented environment. *Journal of Statistical Software*, v. 1, n. 4, p. 1–30, 1997. ISSN 1548-7660. Disponível em: <https://www.jstatsoft.org/v001/i04>. Citado na página 24.

TAN, P.-N.; STEINBACH, M.; KUMAR, V. *Introduction to Data Mining, (First Edition)*. USA: Addison-Wesley Longman Publishing Co., Inc., 2005. ISBN 0321321367. Citado 9 vezes nas páginas 17, 21, 23, 24, 25, 26, 29, 38 e 43.

TANTRUM, J.; MURUA, A.; STUETZLE, W. Assessment and pruning of hierarchical model based clustering. In: *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*. [S.l.: s.n.], 2003. p. 197–205. Citado na página 10.

TST, T. S. d. T. *O Programa Trabalho Seguro*. 2022. Disponível em: <https://www.tst.jus.br/web/trabalhoseguro/apresentacao>. Acesso em: 10 out. 2022. Citado na página 15.

WANG, F. et al. An analysis of the application of simplified silhouette to the evaluation of k-means clustering validity. In: *Machine Learning and Data Mining in Pattern Recognition: 13th International Conference, MLDM 2017, New York, NY, USA, July 15-20, 2017, Proceedings 13*. [s.n.], 2017. p. 291–305. ISBN 978-3-319-62415-0. Disponível em: <https://arrow.tudublin.ie/scschcomcon/207/>. Citado na página 37.