Curso de Sistemas de Informação
Universidade Estadual de Mato Grosso do Sul

Reconhecimento de Padrões de Mortalidade Infantil no Mato Grosso do Sul

Kelvin Cauan da Silva Matos

Prof. Dr. Jorge Marques Prates (orientador)
Prof. Dr. Evandro Cesar Bracht (coorientador)

Dourados - MS
2025

Reconhecimento de Padrões de Mortalidade Infantil no Mato Grosso do Sul

Kelvin Cauan da Silva Matos

Este exemplar corresponde à redação final da monografia da disciplina Projeto Final de Curso devidamente corrigida e defendida por Kelvin Cauan da Silva Matos e aprovada pela Banca Examinadora, como parte dos requisitos para a obtenção do título de Bacharel em Sistemas de Informação.

Dourados, 11 de novembro de 2025

Prof. Dr. Jorge Marques Prates (orientador)

Prof. Dr. Evandro Cesar Bracht (coorientador)

Curso de Sistemas de Informação
Universidade Estadual de Mato Grosso do Sul

Reconhecimento de Padrões de Mortalidade Infantil no Mato Grosso do Sul

Kelvin Cauan da Silva Matos

Novembro de 2025

Banca Examinadora:

Prof. Dr. Evandro Cesar Bracht (Coorientador)
Área de Computação - UEMS

Prof. Dr. Cleber Valgas Gomes Mira
Área de Computação - UEMS

Profa. Dra. Jéssica Bassani de Oliveira
Área de Computação - UEMS

M381r Matos, Kelvin Cauan da Silva

Reconhecimento de padrões de mortalidade infantil no Mato Grosso do Sul / Kelvin Cauan da Silva Matos. – Dourados, MS: UEMS, 2025.
83 p.

Monografia (Graduação) – Sistemas de Informação – Universidade Estadual de Mato Grosso do Sul, 2025.
Orientador: Prof. Dr. Jorge Marques Prates.
Coorientador: Prof. Dr. Evandro Cesar Bracht.

1. Mortalidade infantil 2. Inteligência artificial 3. Aprendizado de máquina 4. DATASUS I. Prates, Jorge Marques II. Bracht, Evandro Cesar III. Título
CDD 23. ed. - 006.3

Ficha Catalográfica elaborada pela bibliotecária da Universidade Estadual de Mato Grosso do Sul (UEMS)
Bruna Peruffo Vieira – CRB 1/2959

Dedico este trabalho aos meus pais, pelo amor e apoio incondicional. Agradeço também à minha futura esposa e ao meu professor orientador, cuja orientação foi fundamental. A todos, meu sincero agradecimento.

“Não sou nada. Nunca serei nada. Não posso querer ser nada.
À parte isso, tenho em mim todos os sonhos do mundo.”
(Fernando Pessoa)

Agradecimentos

Primeiramente, gostaria de expressar minha profunda gratidão aos meus pais, Claudio e Lenice. Este caminho teria sido infinitamente mais árduo sem o apoio, o amor incondicional e a dedicação incansável deles. Sou eternamente grato por tudo que fizeram e continuam a fazer por mim. Minha gratidão se estende a todos os meus familiares e amigos que, de alguma forma, contribuíram para que eu chegasse até aqui. Cada palavra de apoio e cada gesto de carinho fez uma diferença imensurável.

Agradeço de coração à minha futura esposa, Mariana, que esteve ao meu lado em todos os momentos, especialmente nos mais difíceis. Sua presença foi e continua sendo uma fonte inestimável de força e motivação.

Ao meu professor orientador e coorientador, cuja sabedoria, paciência e dedicação foram fundamentais para a realização deste trabalho. Sou profundamente grato por seu apoio e por acreditar no meu potencial.

À professora Cláudia Cardoso, pela orientação, pelos conselhos e pelo incentivo ao longo deste percurso, que foram essenciais para o meu desenvolvimento. Agradeço a ambos pelas valiosas contribuições e por me guiarem em momentos cruciais durante a realização deste trabalho.

Resumo

A mortalidade infantil é um importante indicador de saúde pública e desenvolvimento socioeconômico. Este trabalho tem como objetivo desenvolver um modelo preditivo capaz de reconhecer padrões de mortalidade infantil no estado de Mato Grosso do Sul, utilizando dados públicos disponibilizados pelo DATASUS. A proposta é empregar técnicas de aprendizado de máquina para identificar variáveis com maior influência nos óbitos infantis e prever ocorrências com base em dados históricos. O estudo tem como finalidade identificar padrões relevantes que possam contribuir para a compreensão do cenário da mortalidade infantil no estado, além de oferecer informações que possam auxiliar em análises futuras e apoiar o monitoramento e o planejamento na área da saúde. Para isso, serão utilizados métodos estatísticos e computacionais aplicados à análise de dados de saúde, com foco na acurácia e interpretabilidade do modelo.

Palavras-chave: Ciência de Dados, Inteligência Artificial, Aprendizado de Máquina, DATASUS, Saúde Pública, Mortalidade Infantil.

Abstract

Infant mortality is a key indicator of public health and socioeconomic development. This work aims to develop a predictive model capable of recognizing patterns of infant mortality in the state of Mato Grosso do Sul, using public data made available by DATASUS. The proposal is to apply machine learning techniques to identify variables with the greatest influence on infant deaths and to predict occurrences based on historical data. The study aims to identify relevant patterns that can contribute to a better understanding of the infant mortality landscape in the state, as well as to provide insights that may support future analyses and assist in health monitoring and planning. To achieve this, statistical and computational methods will be applied to health data analysis, with a focus on model accuracy and interpretability.

Keywords: Data Science, Artificial Intelligence, Machine Learning, DATASUS, Public Health, Infant Mortality.

Sumário

Lista de Figuras

CAPÍTULO 1

Introdução

Neste capítulo, destacamos a relevância da mortalidade infantil como indicador de saúde pública e desenvolvimento socioeconômico. O estudo tem como objetivo desenvolver um modelo preditivo, utilizando dados do DATASUS, para identificar padrões e variáveis determinantes, apoiando o monitoramento e o planejamento em saúde no estado de Mato Grosso do Sul.

1.1 Contexto

A mortalidade infantil representa um dos maiores desafios para os sistemas de saúde pública, particularmente em regiões vulneráveis (Victora et al., 2011). A identificação precoce de fatores de risco e a implementação de políticas eficazes são essenciais para a redução dos índices de mortalidade. A análise de dados, juntamente com ferramentas avançadas como o Processo de Descoberta de Conhecimento em Bases de Dados (KDD) e a Inteligência Artificial (IA), é crucial para a formulação de estratégias mais eficazes na mitigação desse problema.

O KDD tem se consolidado como uma metodologia poderosa para extrair padrões significativos a partir de grandes volumes de dados, especialmente na área da saúde. Ao lado do KDD, a Inteligência Artificial desempenha um papel crucial, trazendo capacidades preditivas mais robustas e oferecendo novas maneiras de interpretar dados complexos. A combinação de técnicas de IA, ao exemplo do aprendizado de máquina, juntamente ao processo de descoberta de conhecimento permite uma análise mais profunda e precisa dos fatores que afetam a mortalidade infantil (Bernardini, 2017).

No processo de KDD, a etapa de seleção e pré-processamento de dados, que inclui a limpeza e organização dos dados provenientes de registros hospitalares e sistemas de saúde, é crucial para garantir a integridade dos resultados (CARVALHO and et al., 2011). Entretanto, é na fase de mineração de dados, com o uso de algoritmos de aprendizado supervisionado, que a IA se destaca. Algoritmos como redes neurais e árvores de decisão podem ser aplicados para

prever com maior acurácia diferentes desfechos de saúde em populações pediátricas (Tan et al., 2006). Essas técnicas permitem analisar padrões complexos, que podem ser imperceptíveis em análises tradicionais, ajudando a identificar com precisão os fatores que influenciam a mortalidade.

Além da extração de conhecimento, a etapa de interpretação dos resultados é fundamental para que os padrões identificados possam guiar decisões práticas na saúde pública. No contexto do Mato Grosso do Sul, a aplicação de técnicas de KDD e IA pode fornecer subsídios valiosos para a criação de políticas preventivas, melhor distribuição de recursos e aprimoramento dos serviços de atendimento (Victora et al., 2011).

Estudos mostram que a utilização de KDD na área da saúde tem um impacto significativo. O processo de descoberta de conhecimento pode revelar informações valiosas, muitas vezes não detectáveis por métodos estatísticos tradicionais (Fayyad et al., 1996; Soczek[1] and Orlovski, 2014).

No campo da bioinformática, Holzinger and Jurisica (2014) sugerem que a integração de técnicas interativas de aprendizado de máquina e KDD é o futuro da bioinformática, especialmente na análise de dados complexos, como os de saúde pública. O uso de abordagens integradas pode levar a uma melhor compreensão de fatores críticos que influenciam os resultados de saúde, como a mortalidade infantil. Além disso, Gomes et al. (2014) demonstraram que a mineração de dados pode ser usada eficazmente em serviços de atendimento de urgência, aumentando a capacidade de resposta a situações críticas.

Este estudo tem como objetivo identificar padrões de mortalidade infantil no estado de Mato Grosso do Sul, contribuindo para o avanço do conhecimento na área e apoiando o desenvolvimento de pesquisas voltadas à redução desses índices (Gomes et al., 2014; Victora et al., 2011).

1.2 Motivação e Justificativa

Segundo Victora et al. (2011), a previsão de mortalidade infantil é um componente essencial para a formulação de políticas públicas eficazes e para a melhoria da saúde pública. O crescimento exponencial de dados abertos disponíveis torna fundamental a aplicação de técnicas avançadas para análise e previsão de resultados relacionados à mortalidade, especialmente em populações vulneráveis.

Além disso, conforme Fayyad et al. (1996), o aumento significativo da quantidade de dados de saúde exige soluções inovadoras que garantam uma previsão precisa e eficaz das taxas de mortalidade. A utilização de inteligência artificial se destaca nesse contexto, pois permite a modelagem de grandes volumes de dados, possibilitando a identificação de padrões complexos que seriam difíceis de detectar por métodos tradicionais.

A capacidade de prever mortalidade infantil e outras causas de morte é crucial para a implementação de intervenções de saúde pública direcionadas e efetivas. Como destacam Holzinger and Jurisica (2014), informações precisas sobre grupos populacionais, causas predo-

minantes de morte e áreas geográficas mais afetadas permitem o desenvolvimento de estratégias preventivas e ações de intervenção para reduzir a mortalidade e melhorar a qualidade de vida. Além disso, a aplicação de técnicas automatizadas de previsão pode aumentar a eficiência dos serviços de saúde, permitindo que os gestores tomem decisões informadas sobre alocação de recursos e priorização de ações de saúde.

Os avanços tecnológicos em inteligência artificial têm aprimorado não apenas a precisão das previsões, mas também a capacidade de identificar e antecipar tendências relacionadas à mortalidade. Essa abordagem proativa possibilita a intervenção antecipada, evitando que situações críticas se agravem e garantindo processos mais eficazes e centrados no paciente. Nesse contexto, Gomes et al. (2014) destacam que a integração de técnicas de inteligência artificial na previsão de mortalidade permite a criação de modelos preditivos inovadores, capazes de revelar as complexas interações entre fatores de risco e mortalidade. Assim, a aplicação dessas técnicas torna-se essencial para uma resposta mais rápida e adequada a emergências sanitárias, contribuindo para uma gestão da saúde mais eficiente e humanizada.

Portanto, o reconhecimento de padrões de mortalidade infantil no Mato Grosso do Sul, como mencionado por Bernardini (2017), é essencial para ampliar o conhecimento sobre o tema e fornecer subsídios para estudos futuros que possam contribuir para a compreensão e redução desses índices.

1.3 Objetivo Geral

O objetivo geral deste projeto é identificar padrões de mortalidade infantil no estado de Mato Grosso do Sul, utilizando técnicas de KDD (Descoberta de Conhecimento em Bases de Dados) e inteligência artificial.

1.4 Objetivos Específicos

Para alcançar o objetivo geral, serão atendidos os seguintes objetivos específicos:

- Coletar e limpar os dados de mortalidade disponíveis no DATASUS, preparando-os para análise, a fim de garantir a qualidade e a integridade das informações;
- Analisar os dados para identificar características demográficas e socioeconômicas dos grupos mais afetados pela mortalidade infantil;
- Desenvolver um modelo preditivo utilizando técnicas de aprendizado de máquina para prever taxas de mortalidade com base em fatores de risco identificados;
- Identificar padrões e tendências sazonais nas taxas de mortalidade, correlacionando-as com variáveis ambientais e socioeconômicas.

1.5 Organização

Com o intuito de contextualizar e de apresentar a proposta de trabalho, o restante deste projeto está organizado da seguinte maneira:

- No Capítulo 2 é realizada a revisão da literatura, abordando estudos e pesquisas relacionados ao tema, de modo a fornecer uma base teórica sólida;
- No Capítulo 3 é descrita a metodologia empregada, incluindo as técnicas e ferramentas utilizadas para coleta, preparação e análise dos dados;
- No Capítulo 4 são apresentados os procedimentos de análise e a identificação dos padrões de mortalidade infantil no Estado do Mato Grosso do Sul;
- No Capítulo 5 são apresentados os principais resultados obtidos a partir dos experimentos e avaliações conduzidos;

CAPÍTULO 2

Revisão da Literatura

Neste capítulo, são apresentados estudos sobre a mortalidade infantil, seus desafios e o uso de mineração de dados e técnicas de aprendizado de máquina para sua análise.

2.1 Mortalidade Infantil

No início do século XX, a compreensão sobre a complexidade dos fatores envolvidos na mortalidade infantil começou a ganhar maior atenção. Gradualmente, tornou-se evidente que esse fenômeno não estava relacionado apenas às condições de vida da população, mas também a aspectos ligados à saúde materna e à qualidade da relação mãe-filho, fundamentais para o bem-estar e a sobrevivência da criança.

Como resultado dessas observações, o primeiro ano de vida passou a ser subdividido em faixas etárias específicas, com o objetivo de aprimorar a análise e a compreensão das causas dos óbitos infantis. Em 1938, o termo “mortalidade neonatal” foi introduzido, e, na década de 1950, as classificações que permanecem vigentes até os dias atuais foram consolidadas: natimortalidade ou mortalidade fetal (referente aos óbitos de fetos com 22 semanas ou mais de gestação), mortalidade neonatal precoce, perinatal, neonatal e pós-neonatal (Armstrong, 1986).

Dentro desse contexto, alguns conceitos fundamentais devem ser compreendidos. O nascimento consiste na expulsão ou extração do feto do organismo materno, independentemente do corte do cordão umbilical ou da separação da placenta, sendo que fetos com peso inferior a 500 gramas não são considerados viáveis para estatísticas perinatais. Considera-se que há vida ao nascimento quando o recém-nascido apresenta qualquer sinal vital, como respiração ou movimento espontâneo, caracterizando o chamado "nascido vivo". A morte infantil corresponde ao óbito ocorrido no primeiro ano de vida, sendo classificada em duas etapas: neonatal, nos primeiros 28 dias incompletos, e pós-neonatal, do 29º dia até completar um ano de vida.

2.1.1 Tendências e Desafios

Nas últimas décadas, o Brasil seguiu a tendência mundial de redução das taxas de mortalidade infantil. Contudo, esse avanço foi menos expressivo no componente neonatal, o que evidencia desafios persistentes relacionados à qualidade da assistência obstétrica e neonatal. A mortalidade neonatal reflete, de forma direta, a efetividade dos cuidados prestados às gestantes e aos recém-nascidos durante o pré-natal, o parto e o período neonatal imediato (Organização Mundial da Saúde, 2006).

Nos últimos anos, contudo, observou-se um preocupante retrocesso nas conquistas acumuladas nas últimas décadas em relação à redução da mortalidade infantil. Esse aumento nas taxas tem sido atribuído, principalmente, a cortes em programas sociais a partir de 2016, além das crises econômicas, políticas e sociais enfrentadas pelo país no período recente.

Para a promoção da saúde neonatal e redução da mortalidade infantil, a qualidade da assistência em todas as etapas — pré-concepção, gestação, parto e cuidados ao recém-nascido — é fundamental. No entanto, profundas desigualdades regionais, barreiras de acesso aos serviços de saúde, escassez de leitos de terapia intensiva neonatal e fragilidades estruturais e organizacionais dos serviços públicos contribuem para a manutenção de elevados índices de mortalidade neonatal (Organização Mundial da Saúde, 2006; Barbosa, 2004; Ribeiro and Silva, 2000; Schramm and Szwarcwald, 2000).

2.1.2 Indicadores de Mortalidade Infantil

A mortalidade infantil, por definição, abrange todos os óbitos de nascidos vivos entre o primeiro minuto de vida até o primeiro ano de idade. Sua mensuração é realizada por meio de indicadores como a **Taxa de Mortalidade Infantil (TMI)**, a **Taxa de Mortalidade Neonatal (TMN)** e a **Taxa de Mortalidade Pós-Neonatal (TMP)**, que avaliam os riscos de morte em diferentes períodos do primeiro ano de vida. Outro indicador fundamental é a **Taxa de Mortalidade Fetal (TMF)**, que se refere aos óbitos ocorridos antes do nascimento.

Os óbitos fetais têm grande relevância epidemiológica, pois, em muitos contextos, superam em número os óbitos neonatais precoces (aqueles ocorridos antes dos 7 dias de vida). Essas mortes estão fortemente associadas a falhas no acompanhamento pré-natal, ao manejo obstétrico inadequado e a deficiências na assistência nas horas que antecedem o parto. Estudos indicam que uma parcela significativa tanto das mortes fetais quanto das neonatais precoces poderia ser evitada com intervenções oportunas e adequadas, em mais da metade dos casos (Silva et al., 2016).

2.1.3 Principais Taxas Utilizadas

A mortalidade infantil pode ser avaliada por diversos índices. O mais utilizado é a **Taxa de Mortalidade Infantil (TMI)**, mas outros indicadores complementares são fundamentais para uma análise mais detalhada:

- **Taxa de Mortalidade Fetal (TMF):** Número de óbitos fetais (a partir da 22ª semana de

gestação, ou com peso igual ou superior a 500g, ou estatura de pelo menos 25 cm) por mil nascimentos totais.

- **Taxa de Mortalidade Neonatal (TMN):** Óbitos ocorridos até o 28º dia de vida.
- **Taxa de Mortalidade Pós-Neonatal (TMP):** Óbitos que ocorrem do 29º dia até completar o primeiro ano de vida.

Além disso, a **Mortalidade Perinatal** inclui tanto os óbitos fetais tardios (a partir da 22ª semana de gestação) quanto os óbitos neonatais precoces (até o 7º dia de vida), representando um importante indicador da qualidade da assistência obstétrica e neonatal.

2.2 KDD - Descoberta de Conhecimento em Bancos de Dados

O processo de *Descoberta de Conhecimento em Bancos de Dados* (KDD) foi introduzido por Fayyad et al. (1996) como uma metodologia sistemática para a descoberta de conhecimento útil em grandes bases de dados. Ele envolve uma série de etapas interligadas que abrangem desde a preparação dos dados até a interpretação dos resultados. O objetivo central do KDD é transformar dados brutos em conhecimento acionável, utilizando uma combinação de técnicas de pré-processamento, análise estatística, aprendizado de máquina e visualização.

Segundo Fayyad et al. (1996), o processo de KDD pode ser dividido nas seguintes etapas principais:

- **Seleção:** identificação dos dados relevantes para a tarefa em questão;
- **Pré-processamento:** limpeza, transformação e organização dos dados para garantir a qualidade e a consistência;
- **Formatação:** conversão dos dados para formatos adequados à aplicação de algoritmos de mineração de dados;
- **Mineração de Dados:** aplicação de técnicas de aprendizado de máquina e estatística para identificar padrões e tendências;
- **Interpretação e Avaliação:** validação dos resultados obtidos e conversão em conhecimento utilizável.

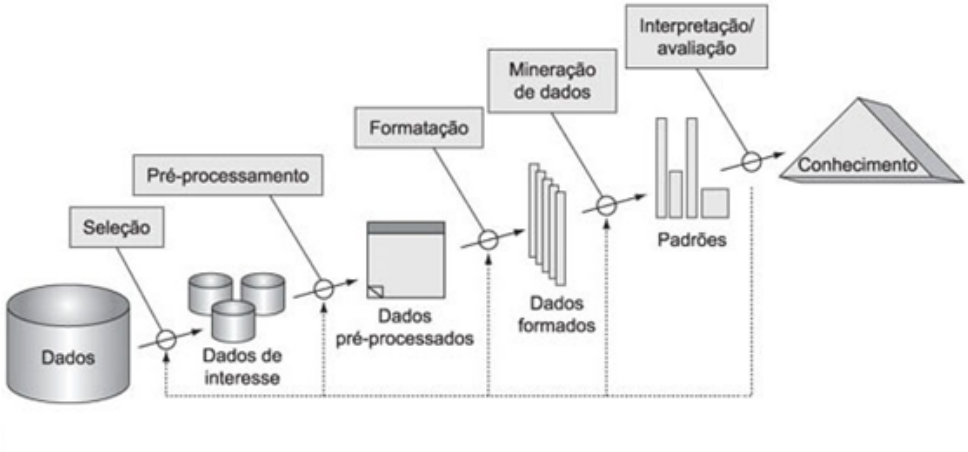

Figura 2.1: Diagrama das etapas do processo de KDD, adaptado de (Fayyad et al., 1996).

As aplicações modernas do KDD frequentemente incorporam técnicas de aprendizado de máquina interativo e ferramentas de visualização avançadas, promovendo uma análise mais aprofundada e precisa dos dados (Holzinger and Jurisica, 2014). Esses avanços possibilitam a adaptação de abordagens às necessidades específicas de cada contexto, otimizando a qualidade dos insights obtidos e fortalecendo a tomada de decisão baseada em dados.

2.3 DATASUS

O **DATASUS** (Departamento de Informática do Sistema Único de Saúde) é uma iniciativa do Ministério da Saúde do Brasil, responsável pela coleta, armazenamento e disponibilização de dados relacionados à saúde pública. Seu objetivo principal é fornecer suporte à formulação de políticas públicas por meio do acesso facilitado a informações de saúde de todo o território nacional. As bases de dados disponibilizadas pelo DATASUS abrangem uma variedade de temas, como morbidade, mortalidade, nascimentos, atendimentos hospitalares e recursos assistenciais.

Os dados utilizados neste projeto foram obtidos por meio da plataforma **OpenDataSUS**, mais especificamente do conjunto de dados do **Sistema de Informações sobre Mortalidade (SIM)**[1]. Esses dados são fornecidos pelo DATASUS e contêm registros de óbitos ocorridos no Brasil, classificados segundo causas básicas, faixas etárias, localização geográfica, entre outros critérios.

Os arquivos de mortalidade estão organizados anualmente e são disponibilizados no formato *CSV (Comma-Separated Values)*, o que facilita seu uso em análises computacionais e em processos de mineração de dados. Cada arquivo anual representa um recorte temporal específico, permitindo a realização de estudos longitudinais e a identificação de padrões ou tendências ao longo do tempo.

Para este projeto, o único filtro aplicado foi o de **ano**, de modo a obter os dados de

[1] https://opendatasus.saude.gov.br/dataset/sim

mortalidade separados por período. Esses arquivos foram baixados diretamente da plataforma OpenDataSUS, sem o uso da ferramenta TABNET.

Esses arquivos são amplamente compatíveis com ferramentas de análise estatística e linguagens de programação. Neste trabalho, utilizaremos a linguagem *Python* para realizar todas as etapas de análise dos dados, incluindo o pré-processamento, mineração e visualização de informações relevantes.

Portanto, a integração dos dados do DATASUS ao processo de descoberta de conhecimento permite não apenas compreender melhor os fatores associados à mortalidade no país, mas também oferece subsídios para a elaboração de políticas públicas mais eficazes, baseadas em evidências extraídas de grandes volumes de dados.

2.4 Banco de Dados Relacional

Para organizar e manipular de forma eficiente os dados extraídos do DATASUS, foi utilizado um **banco de dados relacional**, mais especificamente o **PostgreSQL**. Esse tipo de banco de dados oferece uma estrutura robusta para armazenamento, consulta e gerenciamento de grandes volumes de dados, sendo amplamente utilizado em aplicações de ciência de dados e aprendizado de máquina.

Foi decidido neste projeto utilizar o PostgreSQL não apenas pelo seu desempenho, mas também com o objetivo de armazenar os dados já tratados, garantindo a padronização e a consistência entre eles. Além disso, essa estratégia facilita o processo de treinamento do modelo preditivo, uma vez que os dados se encontram organizados de forma estruturada. Outro fator importante é a sua portabilidade e fácil leitura, o que permite que em futuras melhorias os dados possam ser disponibilizados de maneira online, ampliando as possibilidades de integração com sistemas externos.

O PostgreSQL foi escolhido por sua confiabilidade, suporte a tipos de dados complexos e compatibilidade com ferramentas de análise estatística e linguagens de programação como o **Python**. Sua capacidade de realizar consultas SQL avançadas facilita o acesso rápido a subconjuntos específicos de dados, o que é essencial para o desenvolvimento de modelos preditivos.

Essa abordagem permite que os dados fiquem organizados de forma acessível e reutilizável, possibilitando a realização de análises complexas e a aplicação de algoritmos de aprendizado de máquina com maior eficiência. Além disso, a integração com o PostgreSQL contribui para a automatização do fluxo de trabalho e para a escalabilidade do projeto.

2.5 Algoritmos de Aprendizado de Máquina

Os algoritmos de aprendizado de máquina podem ser classificados em três tipos principais: supervisionado, não supervisionado e por reforço. No aprendizado supervisionado, o modelo aprende a partir de dados rotulados, mapeando entradas para saídas conhecidas, como identificar se uma imagem é de um “ônibus” ou um “pedestre”. No aprendizado não supervisionado, o modelo busca padrões ocultos sem rótulos, como agrupar imagens semelhantes. Já

no aprendizado por reforço, o agente ajusta suas ações com base em recompensas e punições, como aprender a jogar um jogo (Russell and Norvig, 2016).

2.5.1 Algoritmo de Aprendizado Supervisionado: Árvore de Decisão

A **árvore de decisão** é um método popular para classificação e regressão, que aprende a mapear entradas a saídas rotuladas por meio de divisões hierárquicas nos dados. O modelo constrói uma estrutura em forma de árvore, onde cada nó representa uma condição baseada em uma característica, e os ramos levam a resultados possíveis. Por exemplo, uma árvore de decisão pode ser usada para prever se um aluno será aprovado em uma disciplina com base em variáveis como frequência, notas e tempo de estudo.

2.5.2 Algoritmo de Aprendizado Supervisionado: Random Forest

O **Random Forest** é um algoritmo de aprendizado supervisionado baseado no princípio das árvores de decisão, mas com uma abordagem mais robusta. Em vez de construir apenas uma árvore, ele gera um conjunto de árvores (floresta), cada uma treinada a partir de amostras diferentes dos dados e subconjuntos aleatórios de atributos. Esse processo é chamado de *bagging* (bootstrap aggregating).

A Figura 2.2 ilustra o funcionamento do algoritmo. Primeiramente, são selecionadas **amostras** dos dados originais. Em seguida, para cada subconjunto, é construída uma **árvore de decisão** que gera previsões parciais. Por fim, o **Random Forest** combina todas essas previsões em um único resultado: no caso de classificação, a classe mais votada; e no caso de regressão, a média das estimativas.

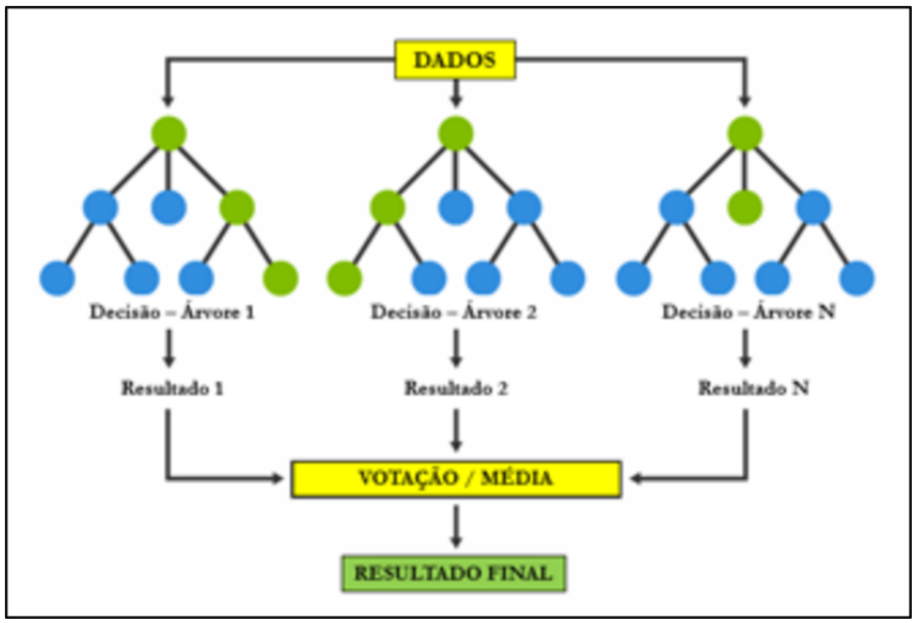

Figura 2.2: Esquema ilustrativo do funcionamento do Random Forest. Fonte: elaborado pelo autor.

Essa estratégia reduz o risco de sobreajuste (*overfitting*), comum em árvores de decisão individuais, ao diminuir a variância do modelo por meio da combinação de múltiplas árvores treinadas sobre subconjuntos de dados e atributos. Como destacado por James et al. (2013), o *Random Forest* tende a apresentar melhor capacidade de generalização ao explorar diversidade entre os modelos base, resultando em previsões mais estáveis e menos sensíveis a ruídos nos dados.

Por sua robustez, o Random Forest é amplamente utilizado em problemas complexos de classificação e regressão, sendo capaz de lidar com variáveis de diferentes tipos, lidar com valores ausentes e capturar relações não lineares entre os dados. Em aplicações práticas, como a análise de mortalidade infantil, ele permite identificar padrões ocultos e estimar a relevância de cada variável para o desfecho, fornecendo não apenas previsões confiáveis, mas também insights interpretáveis para apoio à tomada de decisão.

2.5.3 Algoritmo de Aprendizado Não Supervisionado: K-Means

O **K-Means** é um algoritmo de agrupamento que organiza dados não rotulados em k clusters. Ele funciona iterativamente, otimizando a distância entre os dados e os centros dos clusters (MacQueen, 1967; Lloyd, 1982). Uma aplicação comum é a segmentação de clientes com base em padrões de compras, identificando grupos de consumidores com comportamentos semelhantes.

2.5.4 Algoritmo de Aprendizado por Reforço: Deep Q-Network (DQN)

O **Deep Q-Network (DQN)** combina aprendizado por reforço com redes neurais profundas. Ele utiliza uma rede neural para aproximar a função Q, permitindo que o agente tome decisões mesmo em ambientes com grandes espaços de estados (Mnih et al., 2015). Um exemplo clássico de aplicação é ensinar um agente a jogar jogos de Atari, onde o DQN aprende a maximizar a pontuação com base nas interações com o ambiente do jogo.

2.6 Redes Neurais Artificiais

As Redes Neurais Artificiais (RNA) têm como inspiração as redes neurais biológicas presentes no cérebro humano (CARVALHO and et al., 2011). O sistema nervoso é formado pelos neurônios, responsáveis pelo aprendizado e raciocínio humano. Os neurônios são compostos por dendritos, que são terminais de entrada, e axônios, que são longos terminais de saída. Esse funcionamento é a base das RNAs, onde, a partir de uma entrada específica, a saída será calculada com base no conhecimento aprendido durante o treinamento com um conjunto de dados, semelhante aos outros algoritmos supervisionados de Aprendizado de Máquina (AM) (CARVALHO, 2019).

O algoritmo mais simples das RNAs consiste em uma regra de correção de erros, otimizando uma função quadrática do erro entre as respostas preditas pela RNA e os rótulos dos exemplos (CARVALHO and et al., 2011).

As unidades de processamento, conhecidas como neurônios artificiais, são conectadas por canais de comunicação associados a um peso, que determina a intensidade de uma decisão. Essas unidades computam funções matemáticas e estão distribuídas nas camadas da RNA (CARVALHO and et al., 2011). O comportamento inteligente da RNA emerge das interações entre as unidades de processamento da rede, podendo envolver milhares de unidades para a classificação dos dados (CARVALHO, 2019).

Nas unidades de processamento, são realizadas as seguintes tarefas: valores são apresentados na entrada, cada sinal é multiplicado por um peso, que indica sua influência na saída da unidade. A soma ponderada dos sinais gera um nível de atividade, e, caso esse nível exceda um certo limite, a unidade gera uma resposta de saída. De acordo com CARVALHO and et al. (2011), cada terminal de entrada do neurônio recebe um valor, e esse valor é ponderado e combinado por uma função matemática, chamada função de ativação. A resposta do neurônio é a saída dessa função para a entrada. Existem diversas funções de ativação propostas na literatura, como as funções linear, limiar e sigmoidal. A função de ativação linear (Figura 2.3 (a)) retorna como saída o valor de u. A função limiar (Figura 2.3 (b)) define quando o resultado da função será 0 ou 1. A função sigmoidal (Figura 2.3 (c)) pode ter diferentes inclinações, representando uma aproximação contínua e distinta da função limiar.

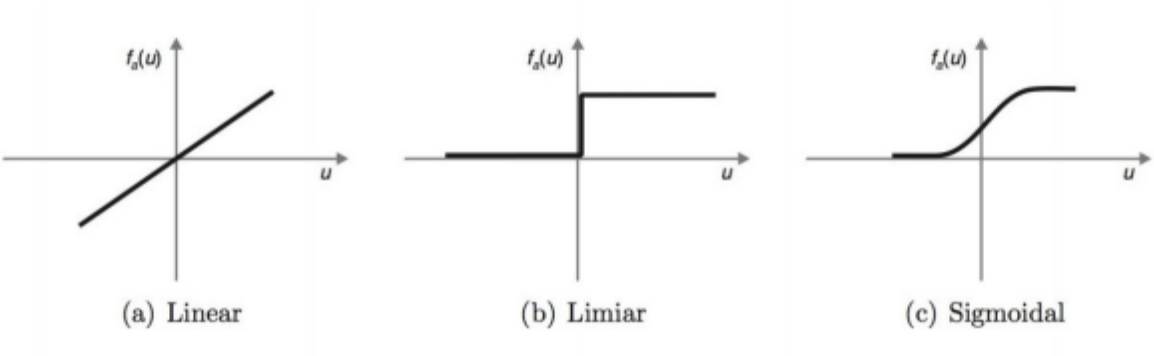

Figura 2.3: Exemplos de funções de ativação. Fonte: CARVALHO and et al. (2011)

A RNA possui dois aspectos principais: a arquitetura e o aprendizado. A arquitetura diz respeito ao tipo e número de unidades de processamento e à forma como elas estão conectadas. O aprendizado está relacionado às regras que ajustam os pesos e as informações nas conexões da rede (CARVALHO and et al., 2011).

Em uma rede neural, os neurônios podem estar distribuídos em uma ou mais camadas. Quando há mais de uma camada, a rede é chamada de rede multicamada. Em redes multicamadas, os neurônios podem receber entradas da camada anterior e enviar suas saídas para camadas subsequentes, contendo camadas intermediárias e a camada de saída. Em redes multicamadas, os neurônios podem se conectar de diferentes formas, tais como: ’

- **Completamente conectada ou densa:** onde todos os neurônios de uma camada estão conectados à camada anterior e/ou à próxima;

- **Parcialmente conectada:** onde os neurônios estão conectados a alguns da camada anterior e/ou da próxima;
- **Localmente conectada:** onde os neurônios estão conectados a um neurônio em uma região específica bem definida (CARVALHO and et al., 2011).

Para o aprendizado e ajuste dos parâmetros de uma RNA, a literatura descreve diversos algoritmos de treinamento, sendo os principais voltados para o ajuste dos pesos associados às conexões da rede. Cada algoritmo possui regras que determinam quando e como alterar os valores dos pesos. Entre as arquiteturas de RNAs e algoritmos para treinamento mais conhecidos, destacam-se: a Rede Perceptron e Adaline, Perceptron multicamadas e o algoritmo Back-propagation (Aggarwal, 2015).

O algoritmo *Back-propagation* é utilizado em redes multicamadas, onde a função de ativação é contínua, diferenciável e, preferencialmente, não decrescente. A função sigmoidal é um exemplo de função que segue esse comportamento. O algoritmo é composto por duas fases: a fase para frente *(forward)* e a fase para trás *(backward)*. Na fase forward, a entrada do objeto é processada na primeira camada intermediária da rede, onde é ponderada pelo peso associado a essa entrada. Cada neurônio na camada aplica a função de ativação, resultando em um valor de saída, que será a entrada da próxima camada. Esse processo é repetido até os neurônios da camada de saída, onde o valor produzido é comparado ao valor esperado. A diferença entre a saída produzida e a saída desejada para cada neurônio da camada de saída indica o erro cometido pela rede com base na entrada fornecida. Esse valor de erro é utilizado na fase *backward* para ajustar os pesos das conexões. Os ajustes são feitos camada por camada (Aggarwal, 2015).

2.7 Trabalhos Relacionados

Aguiar (2019) apresenta uma pesquisa multidisciplinar com o objetivo de desenvolver modelos preditivos para a mortalidade infantil no Estado do Ceará. O estudo utilizou dados de óbitos infantis e nascimentos vivos, aplicando algoritmos de inteligência artificial e técnicas de mineração de dados para gerar insights que apoiem a formulação de políticas públicas. A pesquisa se baseou em uma abordagem quantitativa, complementada por revisões bibliográficas, para sustentar a análise dos dados coletados.

Para alcançar seus objetivos, o autor utilizou algoritmos de Redes Neurais aplicados a 20 variáveis de 8.159 óbitos infantis e 10.235 nascimentos vivos. Com isso, foi possível identificar os principais fatores associados à mortalidade infantil, como baixo peso ao nascer, prematuridade e a falta de acompanhamento pré-natal. A metodologia demonstrou a capacidade da inteligência artificial em analisar grandes volumes de dados e gerar informações úteis para a tomada de decisões na gestão da saúde pública.

Os resultados foram promissores, com uma acurácia média de 97,41% (+/- 0,40%) na predição de casos de mortalidade infantil. Esses achados ressaltam o potencial da inteligência

artificial como ferramenta acessível e de baixo custo para apoiar a formulação de estratégias voltadas à redução da mortalidade infantil no Brasil, destacando sua aplicabilidade em cenários reais de saúde pública.

Sartorelli et al. (2017) investigaram, em um município do Paraná, entre os anos de 2010 e 2014, as relações entre fatores diversos e a mortalidade infantil, aplicando técnicas de mineração de dados. Os algoritmos utilizados incluíram J48 e Redes Neurais de Percepção Multicamadas (NPP), analisando dados de nascimentos e óbitos para identificar padrões. Os resultados apontaram associações significativas entre baixo peso ao nascer, idade gestacional e mortalidade infantil, reforçando a vulnerabilidade de bebês com essas características.

Entre os achados mais relevantes, constatou-se que 56,8% dos óbitos infantis analisados eram evitáveis, destacando a necessidade de atenção adequada à saúde da gestante durante o pré-natal. Um ponto interessante foi a ausência de associação estatisticamente significativa entre a escolaridade materna e a mortalidade infantil na amostra estudada, indicando que outros fatores foram mais influentes naquele contexto. O estudo também revelou uma conexão entre o histórico materno de nascidos mortos e a prematuridade, fator que aumenta consideravelmente o risco de óbito infantil.

Os autores enfatizam a importância de políticas públicas voltadas para a saúde materna e neonatal, com foco na qualificação do pré-natal e na assistência ao parto. Tais ações são essenciais para reduzir os índices de mortalidade infantil, considerando os fatores determinantes identificados e a necessidade de intervenções preventivas eficazes.

Santos et al. (2023) exploraram o uso de técnicas de inteligência artificial para prever a mortalidade em pacientes com insuficiência cardíaca, uma condição com alta taxa de mortalidade. O estudo teve como principal motivação o desenvolvimento de ferramentas que auxiliem na gestão e prognóstico dos pacientes, revisando diversas pesquisas que aplicaram IA nesse contexto e analisando diferentes abordagens e conjuntos de dados disponíveis.

A metodologia adotada foi uma revisão bibliográfica abrangente de estudos que utilizaram IA para essa finalidade. Os autores investigaram o uso de algoritmos de aprendizado de máquina, como Redes Neurais Artificiais, Árvores de Decisão e Máquinas de Vetores de Suporte, aplicados a dados clínicos (idade, sexo, histórico médico), exames laboratoriais e dados de prontuários eletrônicos. Além disso, foram analisados estudos que utilizaram dados de imagem, como ecocardiogramas, processados por meio de aprendizado profundo.

Os resultados revisados demonstraram o potencial da IA em aprimorar a previsão de mortalidade, com modelos capazes de identificar padrões complexos nos dados e apresentar precisão superior aos métodos tradicionais. A aplicação da IA mostrou-se eficaz em diversos aspectos, como previsão de readmissão hospitalar, estratificação de risco e análise de imagens cardíacas. A revisão concluiu que a IA é uma ferramenta promissora para apoiar médicos na tomada de decisões e melhorar o manejo da insuficiência cardíaca, embora enfatize a necessidade de mais estudos e validações em populações diversas.

Santos (2018) investigaram o uso de algoritmos de aprendizado de máquina (Machine

Learning – ML) para prever a mortalidade hospitalar em pacientes com Insuficiência Cardíaca (IC). A pesquisa destacou o potencial do ML em apoiar a tomada de decisões clínicas, permitindo a identificação precoce de pacientes com maior risco de óbito. O objetivo principal foi comparar a eficácia de diferentes algoritmos de ML no contexto da predição de mortalidade, utilizando dados reais de pacientes.

A metodologia utilizou o banco de dados MIMIC-III, contendo registros de pacientes internados em unidades de terapia intensiva, com foco nos diagnosticados com IC. Foram avaliados quatro algoritmos: Regressão Logística, Support Vector Machines (SVM), Random Forest e Redes Neurais Artificiais (RNA). A performance dos modelos foi medida por métricas como acurácia, precisão, recall, F1-score e área sob a curva ROC (AUC), garantindo uma análise abrangente do desempenho preditivo de cada abordagem.

Os resultados indicaram que o algoritmo Random Forest foi o mais eficaz na predição de mortalidade hospitalar, superando os demais algoritmos avaliados. As Redes Neurais Artificiais também apresentaram resultados promissores, ocupando a segunda posição em termos de performance. Esses achados reforçam o valor das técnicas de ML na saúde pública, especialmente para a identificação de pacientes de alto risco, permitindo a otimização de intervenções clínicas. A dissertação concluiu que o ML é uma ferramenta poderosa para melhorar as decisões médicas e as estratégias de cuidado.

Hansen et al. (2022) investigaram os efeitos da pandemia de COVID-19 na saúde mental, com foco em ansiedade e depressão. O estudo utilizou aprendizado de máquina para analisar dados coletados por meio de uma pesquisa online realizada nos Estados Unidos nos primeiros meses da pandemia (maio-junho de 2020). Os dados incluíram informações demográficas, exposição à COVID-19, estressores relacionados à pandemia e fatores de resiliência, além de escalas padronizadas para medir sintomas de ansiedade (GAD-7) e depressão (PHQ-9).

A metodologia envolveu o pré-processamento dos dados da pesquisa, seguido da aplicação de algoritmos de aprendizado de máquina supervisionado. Foram testados modelos como Regressão Logística, Árvores de Decisão, Random Forest, Support Vector Machines (SVM) e Redes Neurais Artificiais, avaliados por métricas como acurácia, precisão, recall, F1-score e AUC. Esses modelos foram treinados para predizer níveis de ansiedade e depressão com base nas variáveis coletadas, identificando padrões e relações relevantes.

Os resultados revelaram os principais preditores de saúde mental durante a pandemia, destacando fatores como preocupações financeiras e mudanças no estilo de vida como os mais associados ao aumento de ansiedade e depressão. O estudo também comparou a performance dos algoritmos, contribuindo para o aprimoramento de estratégias de identificação precoce de indivíduos em risco.

Bertozzo (2022) abordou o uso de técnicas de Aprendizado de Máquina (ML) para prever internações hospitalares no Sistema Único de Saúde (SUS), com foco em Causas Sensíveis à Atenção Primária (CSAP). O objetivo principal foi melhorar a gestão de recursos e o planejamento no sistema de saúde, uma vez que a previsibilidade dessas internações permite a alocação

eficiente de leitos, profissionais e materiais. A escolha das CSAP foi estratégica, considerando seu potencial para intervenções preventivas e redução de custos.

A metodologia envolveu a análise de dados do Sistema de Informações Hospitalares do SUS (SIH/SUS) referentes ao estado do Paraná entre 2014 e 2018. O pré-processamento incluiu etapas de limpeza, tratamento de valores ausentes e seleção de variáveis relevantes, como dados demográficos, diagnósticos e procedimentos realizados. Diversos algoritmos foram avaliados, como Regressão Logística, Árvores de Decisão, Random Forest e Redes Neurais Artificiais. O desempenho dos modelos foi analisado utilizando métricas como AUC, acurácia, precisão, recall e F1-score, com validação cruzada k-fold para maior robustez.

Os resultados evidenciaram o potencial das técnicas de ML, com o Random Forest destacando-se pelo melhor desempenho, apresentando AUC superior a 0.80. Entre as variáveis mais preditivas, estavam idade, sexo, presença de comorbidades e histórico de internações anteriores. O estudo concluiu que o ML pode auxiliar gestores e profissionais de saúde na tomada de decisões, viabilizando ações preventivas mais eficazes e otimização da alocação de recursos no SUS.

O estudo de An et al. (2020) explorou a aplicação de aprendizado de máquina para prever a mortalidade em pacientes diagnosticados com COVID-19 na Coreia do Sul. A pesquisa utilizou uma coorte nacional com 10.237 pacientes, fornecida pelo Korean Centers for Disease Control and Prevention (KCDC). Foram analisadas variáveis demográficas (idade, sexo) e clínicas (sintomas, comorbidades como diabetes, doenças pulmonares e asma, além de residência em asilos) com o objetivo de criar modelos preditivos capazes de apoiar a triagem e a gestão de recursos em saúde.

Os algoritmos de aprendizado de máquina avaliados incluíram LASSO (Least Absolute Shrinkage and Selection Operator), SVM Linear (Máquina de Vetores de Suporte Linear), SVM com função de base radial (RBF), Random Forest e k-NN (k-Vizinhos Mais Próximos). O desempenho dos modelos foi mensurado com métricas como sensibilidade (identificação correta de casos de óbito), especificidade (identificação correta de casos de sobrevivência) e AUC (área sob a curva ROC), oferecendo uma visão abrangente sobre a eficácia dos métodos.

Os resultados indicaram que os modelos LASSO e SVM Linear apresentaram o melhor desempenho preditivo, com sensibilidade em torno de 90%, especificidade acima de 91% e valores de AUC próximos de 0,96. Fatores como idade avançada, sexo masculino, presença de sintomas, residência em asilos e comorbidades (diabetes, doença pulmonar crônica e asma) foram identificados como preditores significativos de mortalidade. O estudo destacou o potencial das técnicas de aprendizado de máquina para apoiar decisões clínicas e otimizar a alocação de recursos durante a pandemia de COVID-19.

O estudo de Silva et al. (2010) objetivou construir um modelo preditivo para o abandono do tratamento da tuberculose (TB), utilizando características sociodemográficas de indivíduos notificados no Sistema de Informação de Agravos de Notificação (SINAN) em João Pessoa-PB, entre 2001 e 2008. A pesquisa, de natureza observacional descritiva com abordagem quan-

titativa, buscou identificar variáveis que pudessem auxiliar na decisão sobre a modalidade de tratamento mais adequada: supervisionado ou autoadministrado. A amostra inicial de 5.164 registros foi refinada para 1.925 após a exclusão de dados duplicados e ausentes, focando na variável "situação de encerramento"(cura ou abandono).

A metodologia empregou análise estatística inferencial por meio de regressão logística binária, com nível de significância de 5%. O método Backward foi utilizado para ajustar o modelo, selecionando as variáveis preditoras mais relevantes: sexo, idade, cor, tipo de entrada (novo caso, recidiva ou reingresso após abandono) e escolaridade. A qualidade do ajuste foi avaliada pela função desvio (deviance) e pela curva ROC (Receiver Operating Characteristic). O software R versão 2.11.1 foi utilizado para as análises estatísticas. Pontos de corte para o modelo foram determinados pelos critérios de Youden e da mínima distância para otimizar a identificação de indivíduos com maior propensão ao abandono.

Os resultados indicaram que as variáveis significativas para o abandono do tratamento foram escolaridade (ensino médio com OR=1,56 e ensino superior com OR=42,99), cor (OR=2,29 para outras cores em comparação a brancos e pardos) e reingresso após abandono (OR=2,71). A qualidade do ajuste da regressão logística foi considerada boa, com baixo valor de deviance e um p-valor alto. A área sob a curva ROC (AUC) foi de 0,722, indicando um ajuste razoável. O estudo concluiu que a consideração dessas variáveis sociodemográficas, além das relacionadas ao tratamento em si, pode contribuir para a melhoria da adesão ao tratamento da TB, direcionando a modalidade supervisionada para os indivíduos com maior risco de abandono e otimizando os recursos disponíveis.

O estudo de Miranda et al. (2023) teve como objetivo principal construir um modelo preditivo para identificar a probabilidade de retenção de pessoas vivendo com HIV/AIDS (PVHA) no cuidado em saúde. Reconhecendo a importância da retenção para o sucesso do tratamento, supressão viral e controle da epidemia, os pesquisadores buscaram desenvolver uma ferramenta para ajudar profissionais de saúde a identificar indivíduos com maior risco de abandono do acompanhamento clínico. A criação desse modelo visava subsidiar a implementação de estratégias de intervenção mais eficazes e contribuir para o alcance das metas de controle da epidemia.

A metodologia utilizada foi a construção de um modelo estatístico por meio de uma árvore de decisão. Os dados foram extraídos de uma pesquisa anterior sobre abandono do acompanhamento clínico ambulatorial especializado de PVHA, utilizando prontuários como fonte secundária. A amostra foi composta por 260 casos de AIDS, selecionados de uma população de 1.941 adultos notificados no SINAN/AIDS na Paraíba entre 2007 e 2013, maiores de 18 anos, e vinculados a um ambulatório especializado de referência no estado. Para a definição de retenção, foi considerado o comparecimento ao serviço para consulta ou exame em um intervalo máximo de sete meses. As variáveis incluídas no modelo foram: idade, estado civil, residência (mesmo município do serviço), orientação sexual, uso de bebida alcoólica, uso de TARV, resultado da última carga viral, internações no serviço e número de comprimidos utilizados por dia. O programa Weka versão 3.7.8 foi utilizado para a construção da árvore de decisão.

Os resultados indicaram que o modelo de árvore de decisão apresentou um percentual de acerto de 80,4% na classificação dos indivíduos quanto à provável retenção ou não no cuidado. Nove variáveis apresentaram ganho de informação significativo, resultando em 24 regras de decisão. A variável "uso da TARV"teve o maior ganho de informação. O estudo também revelou que indivíduos com carga viral indetectável tendem a manter-se em tratamento. O perfil da amostra mostrou predominância de homens, solteiros, heterossexuais e com idade entre 20 e 39 anos. As medidas de validação do modelo indicaram boa sensibilidade (aproximadamente 80%) para identificar os indivíduos com maior risco de não retenção. O estudo concluiu que fatores sociodemográficos, hábitos de vida e variáveis clínicas influenciam a retenção, e a árvore de decisão pode ser uma ferramenta eficaz para ajudar na identificação dos indivíduos com maior risco de abandono e no planejamento de intervenções.

CAPÍTULO 3

Metodologia

Para a elaboração deste capítulo, utilizou-se como referência o processo do KDD. Esse processo, já detalhado em seção anterior deste trabalho, orienta de forma sistemática as etapas necessárias para a descoberta de conhecimento a partir de dados. Assim, a metodologia aqui descrita segue, de maneira adaptada, esse fluxo, servindo como guia para a execução do trabalho proposto.

3.1 Seleção

A etapa de seleção do KDD consiste em escolher quais dados serão utilizados no processo de mineração, garantindo que as informações relevantes estejam disponíveis para análise. Neste projeto, essa fase envolveu tanto a coleta de dados quanto a identificação das variáveis mais adequadas para o treinamento do modelo preditivo, assegurando que os dados escolhidos refletissem com precisão o problema a ser modelado. Detalhes sobre a coleta e preparação dos dados são apresentados nas seções seguintes.

3.1.1 Tipo e Natureza do Estudo

A presente pesquisa caracteriza-se como uma investigação de abordagem quantitativa, cuja finalidade é testar hipóteses e verificar relações entre variáveis, por meio de métodos estatísticos. De acordo com CRESWELL (2010), a pesquisa quantitativa permite a análise objetiva de dados numéricos, visando à identificação de padrões, explicação de fenômenos e formulação de generalizações a partir dos resultados obtidos. Essa abordagem utiliza instrumentos específicos de mensuração, que asseguram a coleta sistemática de dados capazes de subsidiar inferências teóricas.

Do ponto de vista de sua natureza, a pesquisa classifica-se como aplicada, pois busca gerar conhecimentos práticos, direcionados à solução de um problema concreto: a previsão da

Mortalidade Infantil no estado de Mato Grosso do Sul, no período de 1980 a 2024. Segundo GIL (2007), a pesquisa aplicada é orientada à aplicação imediata dos conhecimentos adquiridos, com vistas a contribuir para a resolução de problemas específicos da sociedade.

Quanto aos fins, trata-se de uma pesquisa exploratória, cujo objetivo é proporcionar maior familiaridade com o fenômeno estudado, permitindo torná-lo mais explícito e compreensível. De acordo com CAUCHICK et al. (2017), pesquisas exploratórias são especialmente indicadas quando o tema é pouco conhecido ou quando se busca desenvolver abordagens inovadoras para o tratamento de dados. Nesta investigação, a inovação está centrada no desenvolvimento de uma metodologia preditiva baseada em Inteligência Artificial, aplicada ao contexto da saúde pública.

O presente estudo foi desenvolvido com foco territorial no estado de Mato Grosso do Sul, localizado na Região Centro-Oeste da República Federativa do Brasil. Os dados utilizados nesta pesquisa correspondem a informações secundárias, obtidas de fontes oficiais, abrangendo o período de 1980 a 2024. A escolha desse intervalo temporal buscou garantir uma análise abrangente e longitudinal, permitindo o reconhecimento de padrões, tendências e variações ao longo de mais de quatro décadas de registros.

3.1.2 População e Amostra do Estudo

Segundo MASCARENHAS (2012), denomina-se população ou universo o conjunto formado por pessoas, famílias, empresas, animais ou qualquer outro tipo de elemento que compartilhe características relevantes para um determinado estudo. Já a amostra representa uma fração dessa população, sendo o segmento efetivamente selecionado para a análise.

Na perspectiva estatística, a qualidade dos resultados obtidos está diretamente relacionada à qualidade dos dados utilizados. Não há técnica analítica suficientemente sofisticada que consiga compensar a utilização de dados incorretos ou inconsistentes. Assim, inferências realizadas a partir de amostras suficientemente grandes e adequadamente selecionadas podem apresentar níveis de precisão comparáveis aos obtidos por meio do estudo de toda a população, desde que respeitados os princípios estatísticos básicos (WHEELAN, 2016).

Neste estudo, foram analisados os óbitos infantis registrados no Brasil, conforme dados disponibilizados pelo Sistema de Informações sobre Mortalidade (SIM), vinculado ao Departamento de Informática do Sistema Único de Saúde (Datasus). Para fins desta pesquisa, consideram-se óbitos infantis aqueles ocorridos em crianças com menos de um ano de idade.

A amostra utilizada compreende os óbitos infantis ocorridos no estado de Mato Grosso do Sul, no período de 1980 a 2024. Esses dados foram obtidos a partir da base pública do Ministério da Saúde[1].

[1] https://opendatasus.saude.gov.br/dataset/sim

3.1.3 Dados Coletados

Os dados analisados neste estudo foram extraídos da base pública do SIM, disponibilizada pelo Departamento de Informática do Sistema Único de Saúde (Datasus), abrangendo informações sobre mortalidade infantil e nascidos vivos no estado de Mato Grosso do Sul, entre 1980 e 2024. Esses arquivos, conforme destacado em capítulos anteriores, encontram-se organizados em formato *CSV* (Comma-Separated Values), o que favorece sua utilização em análises computacionais e facilita o processo de pré-processamento e mineração de dados.

Após o processo de coleta e pré-processamento, foi possível verificar a dimensão e a consistência das informações. No total, segundo os dados coletados, o estado registrou **228.025 óbitos** ao longo do período analisado. Desses, **14.995 correspondem a óbitos infantis**, sendo que apenas **9.334 registros** apresentaram dados completos e consistentes, possibilitando sua inclusão nas análises. Além disso, entre os demais casos de mortalidade — aqueles ocorridos após o primeiro ano de vida — apenas **5.232 registros** atenderam aos critérios de qualidade estabelecidos, tornando-se também elegíveis para utilização no estudo.

Portanto, **213.459 registros** foram desconsiderados devido a inconsistências, como campos incompletos ou caracteres inválidos, assegurando que apenas dados de qualidade fossem aproveitados nas análises subsequentes.

3.1.4 Variáveis de Estudo

A definição das variáveis de estudo é essencial para garantir a validade e consistência metodológica da pesquisa. Neste trabalho, todas as variáveis independentes foram consideradas de forma direta, sem qualquer nível hierárquico, para treinar o modelo preditivo.

A variável dependente corresponde ao óbito infantil, definido como o falecimento de crianças com até 12 meses de vida, sendo o desfecho central da investigação. O modelo preditivo foi desenvolvido para estimar o risco de mortalidade infantil no estado de Mato Grosso do Sul, no período de 1980 a 2024.

As variáveis independentes englobam informações socioeconômicas, demográficas, maternas e neonatais, abrangendo fatores como raça ou cor da mãe, idade materna, sexo da criança, peso ao nascer. Todas essas variáveis foram consideradas simultaneamente pelo modelo, sem categorização hierárquica, permitindo uma análise direta do seu impacto sobre o risco de óbito infantil.

3.2 Pré-processamento

O pré-processamento, etapa essencial do KDD, consiste em preparar e transformar os dados brutos coletados para que possam ser analisados de forma confiável e consistente. Esta fase inclui limpeza, padronização, tratamento de valores faltantes e transformação de variáveis, garantindo que os dados estejam prontos para a modelagem preditiva.

Nesta pesquisa, os dados brutos obtidos do DATASUS foram tratados para corrigir inconsistências e padronizar campos, como a variável de idade, originalmente fornecida em di-

ferentes formatos, e convertida para valores em dias. Ajustes adicionais foram realizados para assegurar a correta inserção das informações no banco de dados.

Para o pré-processamento, foi utilizada a linguagem Python, que permitiu automatizar a limpeza, leitura, transformação e inserção dos registros no banco de dados PostgreSQL. Foram empregadas bibliotecas específicas para cada etapa do processo:

- **psycopg2**: conexão e inserção dos dados no PostgreSQL;
- **csv**: leitura e manipulação dos arquivos do DATASUS;
- **collections**: organização e estruturação de dados em formatos adequados para processamento;
- **datetime**: manipulação de datas, incluindo o cálculo da idade em dias.

3.2.1 Pré-processamento dos Dados

O pré-processamento dos dados foi uma etapa essencial para assegurar a qualidade e consistência das informações utilizadas neste trabalho. Nessa fase, os registros brutos provenientes do DATASUS, em especial do Sistema de Informações sobre Mortalidade (SIM), foram submetidos a um tratamento rigoroso antes de serem empregados nas análises. Esse tratamento incluiu a eliminação de inconsistências, a remoção de valores nulos ou inválidos e a padronização de formatos de datas, códigos e categorias.

Uma das principais estratégias adotadas foi a criação de **estruturas de mapeamento (hashmaps)**, utilizadas para converter códigos em valores descritivos. Esse procedimento garantiu que os dados ficassem padronizados e legíveis já no momento de sua utilização, evitando a dependência de códigos "puros". Os principais mapeamentos implementados foram:

- **Ocupações**: processadas a partir da Classificação Brasileira de Ocupações (CBO), disponibilizada pelo Ministério do Trabalho e Emprego[2]. O arquivo, fornecido em formato CSV, foi convertido em uma estrutura de dicionário (*hashmap*) utilizando *Python*, o que permitiu o acesso rápido às descrições durante o pré-processamento. O trecho de código a seguir ilustra a implementação realizada:

Código 3.1: Construção do dicionário de ocupações a partir do arquivo CSV

```
# Ocupações
mapa_ocupacoes = ler_csv_para_dict('cbo2002.csv')
def ler_csv_para_dict(caminho_csv):
    mapa = {}
    with open(caminho_csv, newline='', encoding='latin-1') as csvfile:
        leitor = csv.DictReader(csvfile, delimiter=';')
        for linha in leitor:
            codigo = linha['CODIGO'].strip()
```

[2] http://www.mtecbo.gov.br/cbosite/pages/home.jsf

```
            titulo = linha['TITULO'].strip()
            mapa[codigo] = titulo
    return mapa
```

- **Cidade**: os códigos e nomes dos municípios foram obtidos a partir da base oficial do IBGE[3], onde foi aplicado um filtro para selecionar exclusivamente as cidades do estado de Mato Grosso do Sul. O mapeamento foi construído manualmente e inserido diretamente no código, resultando em uma estrutura de dicionário (*hashmap*) em *Python*. O trecho de código a seguir ilustra a implementação realizada:

Código 3.2: Construção do dicionário de municípios de Mato Grosso do Sul

```
# Cidades - Mato Grosso do Sul
city_map = {
    '500270': 'Campo Grande',
    '500320': 'Corumbá',
    '500370': 'Dourados',
    '500630': 'Ponta Porã',
    '500830': 'Três Lagoas',
    # ... demais municípios de Mato Grosso do Sul
}
```

- **Raça/Cor**: definida conforme os códigos especificados pelo DATASUS, representados no sistema da seguinte forma:

```
1 - Branca
2 - Preta
3 - Amarela
4 - Parda
5 - Indígena
```

A conversão desses códigos foi implementada em *Python* através de um dicionário, como ilustrado a seguir:

Código 3.3: Mapeamento dos códigos de raça/cor

```
race_map = {
    '1' : 'Branca',
    '2' : 'Preta',
    '3' : 'Amarela',
    '4' : 'Parda',
    '5' : 'Indígena'
}
```

[3] https://www.ibge.gov.br/explica/codigos-dos-municipios.php#MS

- **Escolaridade da mãe**: definida com base nos códigos especificados no *Manual de Colunas* do DATASUS, representados da seguinte forma:

```
0 - Sem escolaridade
1 - Fundamental I (1ª a 4ª série)
2 - Fundamental II (5ª a 8ª série)
3 - Médio (antigo 2° Grau)
4 - Superior incompleto
5 - Superior completo
9 - Ignorado
```

O mapeamento foi implementado em *Python* através do seguinte dicionário:

Código 3.4: Mapeamento dos códigos de escolaridade da mãe

```python
school_map = {
    '0': 'Sem escolaridade',
    '1': 'Fundamental I (1 a 4 série)',
    '2': 'Fundamental II (5 a 8 série)',
    '3': 'Médio (antigo 2 Grau)',
    '4': 'Superior incompleto',
    '5': 'Superior completo',
    '9': 'Ignorado'
}
```

- **Sexo**: padronizado a partir dos códigos registrados nos arquivos do DATASUS, definidos como:

```
M ou 1 - Masculino
F ou 2 - Feminino
I, 0 ou 9 - Ignorado
```

O mapeamento foi implementado em *Python* através do seguinte dicionário:

Código 3.5: Mapeamento dos códigos de sexo

```python
sex_map = {
    'M': 'Masculino',
    '1': 'Masculino',
    'F': 'Feminino',
    '2': 'Feminino',
    'I': 'Ignorado',
    '0': 'Ignorado',
    '9': 'Ignorado'
}
```

Além desses mapeamentos, foi necessário implementar funções específicas para tratar campos codificados de forma particular nos arquivos do DATASUS. Dois exemplos importantes foram o tratamento das datas de nascimento e óbito, e a conversão do campo de idade em dias, já que os registros traziam valores codificados em diferentes unidades (minutos, horas, meses e anos).

Código 3.6: Função para converter string no formato ddmmyyyy para objeto date

```
def converter_data(data_str: str):
    try:
        return datetime.strptime(data_str, '%d%m%Y').date()
    except:
        return None
```

A função acima foi utilizada para padronizar os campos de datas, que eram originalmente fornecidos como strings no formato `ddmmyyyy`. Com isso, foi possível garantir consistência no manuseio temporal, permitindo a realização de operações de comparação, ordenação e cálculos de intervalos de tempo.

Código 3.7: Função para converter código de idade para dias

```
def idade_para_dias(idade: str) -> int:
    if not idade or len(idade) != 3:
        return 0
    unidade = idade[0]
    valor = int(idade[1:3])

    if unidade == '1': # minutos para dias
        return max(1, valor // (60*24))
    elif unidade == '2': # horas para dias
        return max(1, valor // 24)
    elif unidade == '3': # meses para dias (aprox. 30 dias por mês)
        return valor * 30
    elif unidade in ['4', '0']: # anos para dias
        return valor * 365
    elif unidade == '5': # mais de 100 anos (aprox.)
        return 365 * 100
    return 0
```

Já a segunda função foi responsável por converter o código de idade — fornecido pelo DATASUS em um formato compacto que mistura unidade e valor — em um número inteiro correspondente ao total de dias vividos. Essa padronização foi crucial para análises consistentes relacionadas à faixa etária, especialmente na identificação e estudo de casos de mortalidade infantil.

3.2.2 Modelo de Entidades e Relacionamentos (MER)

Para o desenvolvimento deste projeto, foi necessário criar um banco de dados relacional que permitisse o armazenamento organizado de dados limpos e estruturados, facilitando a aplicação de modelos preditivos para análise de sobrevivência infantil. O banco foi planejado para fornecer informações consistentes tanto para análises estatísticas quanto para o treinamento de algoritmos preditivos. A modelagem seguiu três níveis: conceitual, lógico e físico. O Modelo Conceitual descreve, de forma abstrata e independente de SGBD, as entidades, atributos e relacionamentos. O Modelo Lógico traduz essas entidades em tabelas com campos, chaves e restrições. Por fim, o Modelo Físico corresponde à implementação concreta no PostgreSQL 16 (HEUSER, 2009).

No projeto, foram criadas três tabelas: `OBITO_INFANTIL`, `SOBREVIVENTES` e `NOVOS_PACIENTES`, todas com as mesmas colunas, que armazenam informações detalhadas sobre os pacientes e suas mães, conforme descrito a seguir:

- `CONTADOR`: identificador único do registro na tabela; valor extraído diretamente do DATASUS.
- `NATURAL`: município de nascimento da criança.
- `SEXO`: sexo da criança.
- `IDADE`: idade da criança no momento do registro.
- `RACACOR`: classificação de raça/cor da criança.
- `CODMUNOCOR`: código do município de residência da criança.
- `IDADEMAE`: idade da mãe no momento do nascimento da criança.
- `OCUPMAE`: ocupação da mãe.
- `ESCMAE`: nível de escolaridade da mãe.
- `PESO`: peso ao nascer da criança, registrado em gramas.

Na tabela `NOVOS_PACIENTES`, além das colunas já descritas, foram incluídos dois campos adicionais: `RESULTADO`, utilizado para armazenar a saída gerada pelo modelo preditivo após a avaliação do caso, e `NOME`, que permite identificar rapidamente o paciente correspondente ao registro analisado.

A tabela `OBITO_INFANTIL` armazena os registros de crianças que foram a óbito antes de completar um ano de vida (Figura 3.1).

Figura 3.1: Colunas da tabela `OBITO_INFANTIL`. Fonte: elaborado pelo autor.

A tabela `SOBREVIVENTES` registra indivíduos nascidos vivos que sobreviveram além do primeiro ano de vida e possui informações similares à tabela `OBITO_INFANTIL`, permitindo comparações diretas entre crianças sobreviventes e não sobreviventes (Figura 3.2). Nesse contexto, a denominação `SOBREVIVENTES` refere-se especificamente aos casos que não evoluíram para mortalidade infantil, ou seja, crianças que não foram a óbito antes de completar um ano de vida, justificando o uso do nome da coluna.

Figura 3.2: Colunas da tabela `SOBREVIVENTES`. Fonte: elaborado pelo autor.

A tabela `NOVOS_PACIENTES` é destinada ao armazenamento de novos registros inseridos para avaliação pelo modelo preditivo, com destaque para os campos `RESULTADO` e `NOME` (Figura 3.3).

Figura 3.3: Colunas da tabela `NOVOS_PACIENTES`. Fonte: elaborado pelo autor.

3.3 Formatação

A etapa de formatação, dentro do processo de KDD, tem como objetivo organizar os dados já limpos e pré-processados em uma estrutura compatível com o banco de dados utilizado no projeto. Diferente do pré-processamento, que se concentra na limpeza e padronização dos dados brutos, a formatação trata da preparação final para armazenamento, garantindo que os registros estejam organizados de acordo com o modelo lógico e físico definido no banco relacional.

Como explicado no capítulo de seleção, os dados foram coletados em arquivos CSV e armazenados em um diretório dedicado contendo todos os anos estudados. Para processá-los, foi desenvolvido um código em *Python* capaz de ler cada linha de cada arquivo, validar os dados e determinar se poderiam ser utilizados. Durante essa leitura, a rotina contabilizava registros válidos e inválidos. Sempre que uma linha era considerada válida, ela era automaticamente inserida no banco de dados, tanto na tabela de sobreviventes quanto na tabela de óbitos infantis.

Nesta pesquisa, o *Python* foi a tecnologia escolhida para esta etapa, assim como foi utilizado no pré-processamento. Funções específicas foram desenvolvidas para realizar a leitura dos arquivos CSV, aplicar validações, transformar os dados no formato adequado e inserir automaticamente os registros no banco PostgreSQL. Isso garantiu consistência entre as variáveis e viabilizou a integração com os modelos preditivos.

3.3.1 Conexão com o Banco de Dados

O script inicia estabelecendo a conexão com o banco PostgreSQL utilizando a biblioteca `psycopg2`:

Código 3.8: Estabelecimento da conexão com o banco de dados

```
1 import psycopg2
```

```
2
3  conexao = psycopg2.connect(
4      host='192.168.100.18',
5      database='db_tcc',
6      user='postgres',
7      password='postgres',
8      port=5432
9  )
10 cursor = conexao.cursor()
```

Os parâmetros da função `connect` definem o servidor, o banco, as credenciais de acesso e a porta utilizada pelo PostgreSQL. A variável `cursor` permite executar comandos SQL dentro da conexão estabelecida.

3.3.2 Funções Auxiliares

Algumas funções foram criadas para modularizar o código e facilitar o processamento:

- **Identificação de óbito infantil:** a função `eh_obito_infantil` determina se um registro corresponde a óbito infantil com base no código de idade fornecido:

Código 3.9: Função para identificar óbito infantil

```
1  def eh_obito_infantil(idade: str) -> bool:
2      if not idade or len(idade) != 3:
3          return False
4      unidade = idade[0]
5      valor = idade[1:]
6      return (unidade in ['1', '2', '3']) or (unidade == '4' and valor == '00')
```

- **Inserção de dados:** a função `inserir_dados` realiza a inserção de registros na tabela correspondente, utilizando SQL parametrizado para garantir segurança e integridade:

Código 3.10: Função para inserir registros no banco de dados

```
1  def inserir_dados(objeto, tabela, cursor):
2      sql = f'''
3          INSERT INTO {tabela}
4          ("natural", "dtnasc", "sexo", "idade", "racacor",
5           "codmunocor", "idademae", "escmae", "ocupmae", "peso")
6          VALUES (%s, %s, %s, %s, %s, %s, %s, %s, %s, %s)
7      '''
8      cursor.execute(sql, (
9          objeto["natural"],
10         objeto["data_nascimento"],
11         objeto["sexo"],
12         objeto["idade"],
```

```
        objeto["raca_cor"],
        objeto["codigo_municipio"],
        objeto["idade_mae"],
        objeto["escolaridade_mae"],
        objeto["ocupacao_mae"],
        objeto["peso"]
    ))
```

3.3.3 Processamento dos Arquivos CSV

Esta etapa realiza o **processamento completo dos dados coletados**, percorrendo todos os arquivos CSV referentes aos anos de estudo, que foram previamente coletados e armazenados em um diretório específico. Cada linha dos arquivos é avaliada individualmente, aplicando as funções desenvolvidas anteriormente para conversão de datas, cálculo da idade em dias e identificação de óbito infantil.

Além disso, são utilizados todos os **hashmaps** descritos no capítulo anterior, assim como as funções previamente implementadas para conversão de datas, cálculo da idade em dias e identificação de óbito infantil. Esses recursos permitem que os valores categóricos e as informações derivadas sejam corretamente transformados e inseridos no banco de dados, assegurando consistência e integridade dos registros.

Durante o processamento, cada registro passa por uma série de **validações e transformações**, assegurando que apenas dados consistentes e completos sejam inseridos nas tabelas correspondentes (`sobreviventes` e `obitos_infantis`). Qualquer registro que não atenda aos critérios definidos é contabilizado como inválido, permitindo a geração de estatísticas detalhadas sobre os erros encontrados, sem comprometer a integridade do banco de dados.

Esta abordagem garante não apenas a qualidade e consistência dos dados armazenados, mas também mantém a rastreabilidade de problemas nos registros, facilitando análises futuras e ajustes no processo de extração e formatação.

Código 3.11: Leitura, validação e inserção de dados

```
registros_lidos = 0
registros_validos = 0
registros_invalidos = 0
erros_por_campo = {
    "DTNASC": 0,
    "IDADE": 0,
    "NATURAL": 0,
    "SEXO": 0,
    "RACACOR": 0,
    "CODMUNOCOR": 0,
    "IDADEMAE": 0,
    "OCUPMAE": 0,
    "ESCMAE": 0,
    "PESO": 0
```

```
}

for nome_arquivo in os.listdir(caminho_base):
   if nome_arquivo.endswith('.csv'):
      caminho_arquivo = os.path.join(caminho_base, nome_arquivo)
      with open(caminho_arquivo, 'r', encoding='utf-8') as arquivo_csv:
         leitor_csv = csv.DictReader(arquivo_csv, delimiter=';')

         for linha in leitor_csv:
            registros_lidos += 1
            erros_linha = []
            try:

               dt_nasc = linha.get('DTNASC')
               idade_str = linha.get('IDADE')
               natural = linha.get('NATURAL')
               sexo = linha.get('SEXO')
               raca = linha.get('RACACOR')
               cod_municipio = linha.get('CODMUNOCOR')
               idade_mae = idade_para_dias(linha.get('IDADEMAE'))
               ocup_mae = linha.get('OCUPMAE')
               esc_mae = linha.get('ESCMAE')
               peso = linha.get('PESO')
               data_nasc = converter_data(dt_nasc)
               dias_idade = idade_para_dias(idade_str)

               if data_nasc is None:
                  erros_linha.append("DTNASC")
               if dias_idade is None:
                  erros_linha.append("IDADE")
               if not natural:
                  erros_linha.append("NATURAL")
               if not sexo or sexo not in sex_map:
                  erros_linha.append("SEXO")
               if not raca or raca not in race_map:
                  erros_linha.append("RACACOR")
               if not cod_municipio or cod_municipio not in city_map:
                  erros_linha.append("CODMUNOCOR")
               if not idade_mae:
                  erros_linha.append("IDADEMAE")
               if not ocup_mae or ocup_mae not in mapa_ocupacoes:
                  erros_linha.append("OCUPMAE")
               if not esc_mae or esc_mae not in school_map:
                  erros_linha.append("ESCMAE")
               if not peso or not peso.isdigit():
                  erros_linha.append("PESO")

```

```
                for campo in erros_linha:
                    erros_por_campo[campo] += 1

                if erros_linha:
                    registros_invalidos += 1
                    continue #


                if(eh_obito_infantil(idade_str)):
                    tabela_destino = 'obitos_infantil'
                else:
                    tabela_destino = 'sobreviventes'

                dados = {
                    "natural": natural,
                    "data_nascimento": data_nasc,
                    "sexo": sex_map.get(sexo),
                    "idade": dias_idade,
                    "raca_cor": race_map.get(raca),
                    "codigo_municipio": city_map.get(cod_municipio),
                    "idade_mae": idade_mae,
                    "escolaridade_mae": school_map.get(esc_mae),
                    "ocupacao_mae": mapa_ocupacoes.get(ocup_mae),
                    "peso": peso,
                }

                inserir_dados(
                    dados=dados,
                    tabela=tabela_destino,
                    cursor=cursor
                )

                registros_validos += 1
            except Exception:
                registros_invalidos += 1
                continue

conexao.commit()
cursor.close()
conexao.close()

print(f'Total de registros lidos: {registros_lidos}')
print(f'Registros válidos inseridos: {registros_validos}')
print(f'Registros inválidos ignorados: {registros_invalidos}')
print("Erros por campo:", erros_por_campo)
```

Ao final da execução do script, foram contabilizados todos os registros lidos, válidos e

inválidos, e também gerado o detalhamento dos erros por campo. Ao todo, **213.459 registros** foram desconsiderados por apresentarem dados incompletos, isto é, linhas com campos ausentes ou com valores inválidos, impedindo sua inclusão nas análises. Após o processo de filtragem e validação, restaram **9.334 registros válidos**, o que representa aproximadamente **4,19%** do total de dados originalmente processados, os quais foram utilizados para o treinamento e avaliação do modelo preditivo.

Esta abordagem modularizada e controlada garante que apenas registros consistentes e corretamente formatados sejam armazenados no banco de dados, proporcionando uma base confiável tanto para análises estatísticas quanto para a aplicação dos modelos preditivos subsequentes.

3.4 Mineração de Dados

A etapa de mineração de dados, no contexto do processo de KDD, consiste na aplicação de técnicas analíticas e algoritmos específicos para identificar padrões, relações e tendências relevantes nos dados previamente tratados. Diferentemente das etapas de seleção, pré-processamento e formatação, que focam na preparação dos dados, a mineração representa o momento em que o conhecimento é efetivamente extraído do conjunto de informações.

Nesta pesquisa, a mineração de dados envolveu a construção de um **modelo preditivo** utilizando *Python*. O modelo foi treinado com base nas duas bases de dados principais geradas durante a formatação: a tabela de *sobreviventes* e a tabela de *óbitos infantis*. Durante o treinamento, o algoritmo aprendeu a identificar quais características estavam associadas a casos de óbito e quais estavam relacionadas à sobrevivência, reconhecendo padrões relevantes para a predição.

Uma vez treinado, o modelo foi aplicado à tabela *novos_pacientes*, composta por registros cujo desfecho ainda não havia sido definido. Dessa forma, é capaz de analisar cada caso individualmente e fornecer uma estimativa da probabilidade de óbito, contribuindo para a avaliação de risco com base nas características observadas nos dados históricos.

O uso de *Python* nesta etapa possibilitou não apenas a construção e o treinamento do modelo, mas também a integração direta com as bases de dados PostgreSQL, garantindo consistência e compatibilidade com a estrutura previamente definida.

Como resultado do treinamento, o modelo alcançou uma **acurácia de 90%** e uma **ROC AUC de 0,98**, demonstrando sua eficácia em diferenciar corretamente casos de óbito e sobrevivência. A *acurácia* representa a proporção de previsões corretas realizadas pelo modelo em relação ao total de amostras avaliadas, enquanto a métrica *ROC AUC* (Área sob a Curva Característica de Operação do Receptor) indica a capacidade do modelo em distinguir entre as classes positivas e negativas — quanto mais próximo de 1, melhor o desempenho discriminativo. Mais detalhes sobre a criação, parametrização e validação do modelo preditivo serão apresentados no próximo capítulo, onde será detalhada a metodologia completa de modelagem aplicada nesta pesquisa.

3.5 Interpretação e Avaliação

A etapa de interpretação e avaliação constitui a fase final do processo de KDD, momento em que os resultados obtidos na mineração de dados são examinados de forma crítica, a fim de verificar sua validade, relevância e aplicabilidade prática. Diferentemente das etapas anteriores — voltadas à preparação, limpeza e transformação dos dados —, esta fase concentra-se em compreender o significado dos padrões identificados, avaliar o desempenho dos modelos preditivos e estabelecer a utilidade das descobertas para apoiar a tomada de decisão.

No contexto desta pesquisa, a interpretação foi realizada a partir da análise detalhada dos resultados produzidos pelo modelo preditivo treinado na etapa anterior. Foram utilizadas métricas como acurácia e área sob a curva ROC (ROC AUC), que permitiram confirmar a capacidade do modelo em diferenciar corretamente entre casos de óbito infantil e de sobrevivência. Paralelamente, investigou-se a importância relativa das variáveis, com o objetivo de identificar quais características dos pacientes exerceram maior influência no desfecho, fornecendo assim subsídios para uma compreensão mais aprofundada dos fatores associados ao risco de mortalidade infantil.

Com o objetivo de proporcionar uma interpretação clara e objetiva, foram desenvolvidos diversos **gráficos interativos**, concebidos para traçar perfis de óbitos infantis e oferecer suporte visual à análise. Esses gráficos permitem a identificação de padrões relevantes de forma acessível, facilitando a exploração e compreensão dos dados por especialistas da área.

É importante destacar que, embora os gráficos apresentem informações estruturadas e consistentes, **os dados exibidos não são extraídos diretamente do banco de dados**. Em vez disso, eles foram obtidos por meio da execução de consultas SQL específicas, e posteriormente inseridos manualmente no código de cada gráfico. Essa abordagem possibilitou a visualização imediata dos resultados sem depender de integrações dinâmicas com o banco, garantindo um controle preciso sobre os valores apresentados e a consistência dos dados utilizados para análise.

Para a construção desses gráficos, optou-se pelo framework **React**, que oferece renderização reativa e gerenciamento eficiente de componentes, permitindo a criação de interfaces dinâmicas e responsivas. Cada gráfico foi encapsulado como um **componente reutilizável**, assegurando modularidade, consistência visual e facilidade de manutenção.

A biblioteca **ApexCharts**, integrada via *react-apexcharts*, foi utilizada para viabilizar gráficos interativos de múltiplos tipos, com suporte à exibição de valores percentuais, legendas personalizadas, paletas de cores diferenciadas e exportação em formato PNG. Essa combinação de tecnologias garantiu que os gráficos fossem não apenas visualmente claros, mas também funcionais e adaptáveis, oferecendo uma experiência de exploração de dados eficiente e intuitiva.

Como exemplo das visualizações implementadas, a Figura 3.4 apresenta um gráfico de barras que ilustra a distribuição percentual de óbitos e sobrevivências por sexo. Este gráfico permite comparar de forma clara a proporção de meninos e meninas afetados, evidenciando possíveis diferenças de risco associadas ao sexo. Outro exemplo relevante é o gráfico de linhas

exibido na Figura 3.5, que mostra a incidência de óbitos e sobrevivências nas principais cidades do estado de Mato Grosso do Sul, permitindo identificar regiões com maior vulnerabilidade e realizar uma análise espacial detalhada dos casos.

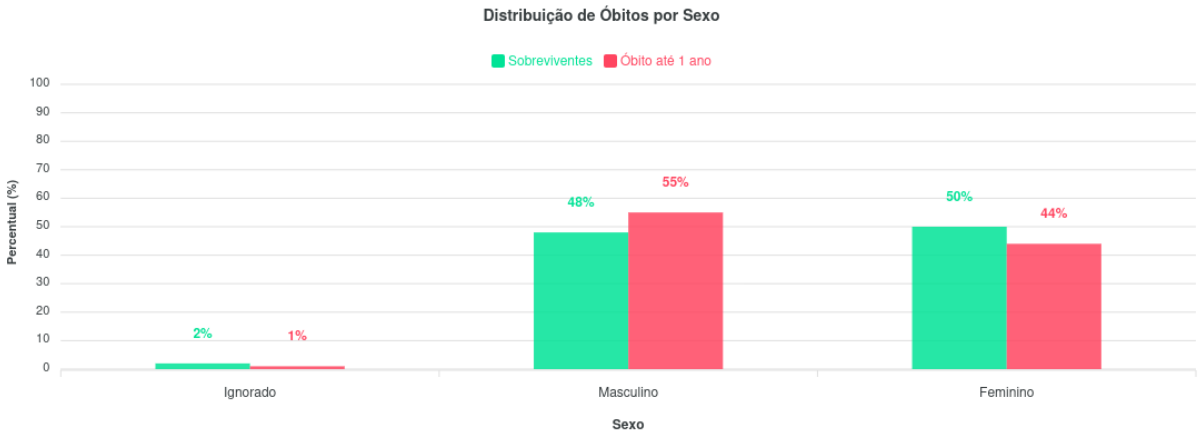

Figura 3.4: Distribuição percentual de óbitos e sobrevivências por sexo, gerada com ApexCharts e integrada ao React.

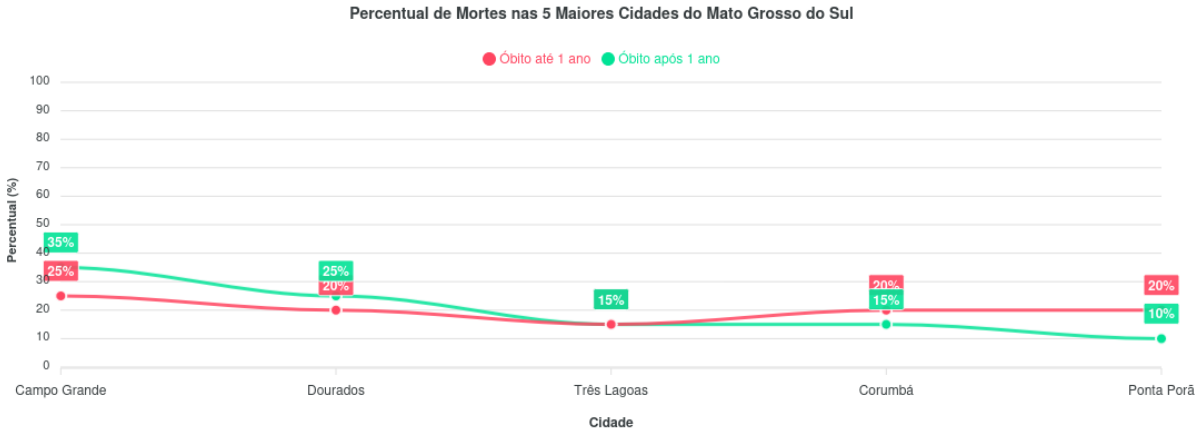

Figura 3.5: Distribuição de óbitos e sobrevivências nas principais cidades do Mato Grosso do Sul, visualizada com gráficos interativos de ApexCharts.

Mais detalhes sobre os demais gráficos gerados neste estudo, incluindo distribuições por idade materna, escolaridade e outros fatores relevantes, serão apresentados no Capítulo de Resultados. estudo, incluindo distribuições por idade da mãe, escolaridade e outros fatores relevantes, serão apresentados no Capítulo de Resultados.

CAPÍTULO

4

Análise e Identificação de Padrões de Mortalidade Infantil no Estado do Mato Grosso do Sul

A análise, o reconhecimento de padrões relacionados à mortalidade infantil no Estado do Mato Grosso do Sul foram realizados por meio da criação e aplicação de um modelo preditivo. Essa abordagem permitiu explorar dados históricos, compreender relações entre variáveis demográficas, sociais e clínicas e, a partir disso, estimar a probabilidade de ocorrência de novos casos de óbito infantil.

Um modelo preditivo pode ser compreendido como uma ferramenta estatística ou computacional que, a partir de dados previamente observados, identifica padrões e utiliza esse conhecimento para antecipar desfechos futuros. Em outras palavras, ele aprende com exemplos históricos para realizar inferências sobre situações ainda não ocorridas (James et al., 2013).

No contexto desta pesquisa, o modelo foi desenvolvido com o propósito de estimar a probabilidade de mortalidade infantil, fornecendo subsídios para a compreensão dos fatores que mais influenciam esse desfecho e contribuindo para a formulação de políticas públicas em saúde infantil.

Neste capítulo, serão detalhados não apenas as etapas de construção do modelo, mas também a análise visual desenvolvida para acompanhar o treinamento e a avaliação de desempenho do preditor. Por meio de gráficos e representações visuais, será possível observar como o modelo aprendeu com os dados históricos e de que forma os resultados se manifestam em termos de acurácia, relevância das variáveis e capacidade de generalização.

4.1 Processo de geração do modelo preditivo

Para a construção do modelo preditivo, utilizou-se a base de dados previamente estruturada, conforme descrito no capítulo anterior, onde foram detalhados os procedimentos de extração, tratamento e padronização dos registros provenientes do DATASUS. Esses dados históricos, referentes à mortalidade infantil e aos indivíduos que sobreviveram além do primeiro ano de vida no estado de Mato Grosso do Sul, foram organizados em um banco de dados relacional. Essa etapa inicial assegurou consistência e confiabilidade às informações, além de fornecer uma estrutura otimizada para consultas, manipulações e análises estatísticas.

Com base nessas análises, elaborou-se um conjunto consolidado a partir de duas bases distintas: a de *óbitos infantis* e a de *sobreviventes*. Esse conjunto *(dataset)* serviu como insumo para o treinamento e validação do modelo, garantindo uma amostra representativa e equilibrada.

A modelagem foi conduzida por meio de técnicas de aprendizado de máquina, com destaque para o algoritmo *Random Forest*, escolhido pela sua robustez em lidar com dados complexos e heterogêneos. O objetivo central foi estimar a probabilidade de óbito infantil com base nos atributos registrados nas diversas colunas do banco de dados, abrangendo informações demográficas, sociais e maternas dos pacientes. Entre as variáveis consideradas, destacam-se a idade materna, escolaridade da mãe, ocupação, peso ao nascer e raça/cor, fatores estes reconhecidos como relevantes na literatura especializada.

O processamento e a implementação do modelo foram realizados de forma automatizada por scripts desenvolvidos em Python, assegurando reprodutibilidade e eficiência em todas as etapas. O modelo resultante apresenta capacidade de identificar padrões complexos e não lineares, oferecendo previsões fundamentadas em raciocínio indutivo e teoria das probabilidades Feijoo (2010).

Por fim, o processo prevê a atualização contínua do modelo à medida que novos dados forem incorporados ao sistema, garantindo que o aprendizado permaneça alinhado às condições mais recentes observadas na população. Dessa forma, a ferramenta proposta configura-se como um recurso de apoio à análise preditiva e à tomada de decisão em políticas públicas de saúde infantil.

4.2 Construção e Validação do Modelo Preditivo

O algoritmo *Random Forest* foi empregado para o treinamento do modelo, utilizando o método *RandomizedSearchCV* para a otimização automática de hiperparâmetros. Essa técnica executa uma busca aleatória sobre um espaço pré-definido de combinações de parâmetros, selecionando amostras de forma estocástica e avaliando o desempenho de cada configuração. Diferentemente do *Grid Search*, que realiza uma busca exaustiva em todas as combinações possíveis, o *Randomized Search* oferece uma alternativa mais eficiente, reduzindo substancialmente o custo computacional e, ainda assim, alcançando resultados comparáveis ou superiores em muitos cenários.

Os principais hiperparâmetros ajustados foram:

- **Número de árvores** (`n_estimators`): influencia diretamente a estabilidade e a capacidade de generalização do modelo;
- **Profundidade máxima das árvores** (`max_depth`): controla o nível de detalhamento das divisões, prevenindo o *overfitting*;
- **Número mínimo de amostras para divisão de um nó** (`min_samples_split`): evita divisões excessivamente específicas, promovendo maior robustez;
- **Número mínimo de amostras em folhas** (`min_samples_leaf`): assegura representatividade estatística em cada folha terminal;
- **Estratégia de seleção de atributos** (`max_features`): define o número de variáveis consideradas em cada divisão, impactando a diversidade entre as árvores.

Conforme destacado por (Bergstra and Bengio, 2012), a busca aleatória tende a ser mais eficaz do que abordagens sistemáticas como o *Grid Search*, especialmente em problemas onde apenas uma fração dos hiperparâmetros exerce influência significativa sobre o desempenho final.

Neste estudo, foram testadas 20 combinações distintas de parâmetros, avaliadas por meio de validação cruzada (*cross-validation*) com três dobras (*folds*). O modelo final foi selecionado com base na melhor média de desempenho em termos de *Area Under the ROC Curve* (AUC). Essa métrica avalia a capacidade do classificador em distinguir corretamente as classes positivas e negativas, independentemente do limiar de decisão adotado.

A utilização da AUC é especialmente relevante em contextos de **desequilíbrio amostral**, isto é, quando uma das classes está representada em quantidade muito menor do que a outra — como ocorre, por exemplo, em bases onde há muitos casos de sobrevivência e poucos de óbito. Nesses cenários, métricas tradicionais como a acurácia podem gerar interpretações enganosas, pois um modelo que sempre prevê a classe majoritária ainda assim apresentaria alta taxa de acertos. Em contraste, a AUC considera simultaneamente as taxas de verdadeiros positivos e falsos positivos, fornecendo uma medida mais robusta e justa do desempenho do modelo frente a distribuições desbalanceadas de classes.

Para avaliar o modelo treinado, utilizou-se o conjunto de teste e diversas métricas complementares:

- **Relatório de classificação** (*precision, recall, f1-score*): detalha a performance por classe. Por exemplo, uma *recall* alta para a classe `óbito` indica que o modelo identifica corretamente a maioria dos casos de risco;

- **Acurácia**: proporção de previsões corretas no total de exemplos, fornecendo uma visão geral do desempenho. Por exemplo, se o modelo classificar corretamente 950 de 1000 pacientes, a acurácia será 95%;
- **ROC AUC**: avalia a capacidade do modelo de separar positivos e negativos ao longo de diferentes limiares, sendo fundamental quando as classes não são balanceadas.

Os resultados indicaram desempenho satisfatório, evidenciando boa capacidade de generalização. A combinação dessas métricas permite avaliar tanto a precisão global quanto a sensibilidade e robustez das previsões, oferecendo uma análise completa da eficácia do modelo.

4.3 Treinamento do Modelo

Nesta seção, detalha-se o fluxo completo de preparação de dados, treinamento do modelo, ajuste de hiperparâmetros e predição em registros inéditos, com o objetivo de construir um modelo robusto capaz de estimar a probabilidade de óbito infantil. Este modelo serve como ferramenta de apoio à tomada de decisão em saúde pública, permitindo identificar grupos de maior risco e priorizar ações preventivas.

O primeiro passo consiste na importação das bibliotecas necessárias:

Código 4.1: Importação das bibliotecas utilizadas para o pré-processamento, modelagem e avaliação do modelo Random Forest.

```
from sqlalchemy import create_engine
from sklearn.model_selection import train_test_split, RandomizedSearchCV
from sklearn.preprocessing import StandardScaler
from sklearn.ensemble import RandomForestClassifier
from scipy.stats import randint
from sklearn.metrics import accuracy_score, roc_auc_score,
classification_report
import joblib
```

Essas bibliotecas são fundamentais para cada etapa do processo: `pandas` é utilizada para manipulação eficiente de dados; `SQLAlchemy` permite extrair dados diretamente do banco PostgreSQL de forma segura; `scikit-learn` oferece todas as ferramentas de modelagem, pré-processamento, validação e avaliação; `scipy.stats` define distribuições aleatórias para otimização de hiperparâmetros; e `joblib` possibilita salvar e carregar o modelo treinado, garantindo reprodutibilidade.

Em seguida, aplica-se o *StandardScaler*, responsável por padronizar as variáveis numéricas do conjunto de dados. Esse processo transforma cada atributo para que possua média igual a zero e desvio-padrão igual a um, conforme o exemplo a seguir:

Código 4.2: Configuração da conexão com o banco de dados PostgreSQL e definição dos arquivos de persistência do modelo Random Forest.

```
engine = create_engine(
    "postgresql+psycopg2://postgres:postgres@192.168.100.18:5432/db_tcc"
)

model_file = "random_forest_model.pkl"
scaler_file = "scaler.pkl"
columns_file = "model_columns.pkl"
```

Essa configuração garante que tanto os dados quanto os artefatos do modelo possam ser acessados de forma consistente, permitindo que novas previsões utilizem o mesmo pré-processamento aplicado durante o treinamento.

Os dados são então extraídos das tabelas `obitos_infantil` e `sobreviventes`. Cada registro recebe um rótulo binário (`label`) indicando sua classe, sendo 1 para óbito e 0 para sobrevivência. A concatenação das tabelas resulta em um único dataset consolidado:

Código 4.3: Leitura dos dados de óbitos e sobreviventes do banco de dados e unificação em um único *DataFrame* para análise.

```
df_mortos = pd.read_sql("SELECT * FROM obitos_infantil", engine)
df_mortos["label"] = 1

df_sobreviventes = pd.read_sql("SELECT * FROM sobreviventes", engine)
df_sobreviventes["label"] = 0

df = pd.concat([df_mortos, df_sobreviventes], ignore_index=True)
```

Este passo é crucial para garantir que o conjunto de dados seja *balanceado*, ou seja, que possua uma proporção semelhante de exemplos pertencentes a cada classe — no caso, pacientes que sobreviveram e pacientes que foram a óbito.

Um conjunto desbalanceado, onde uma classe é muito mais frequente que a outra, pode levar o modelo a aprender de forma enviesada, priorizando a classe majoritária e negligenciando a minoritária. Ao balancear os dados, assegura-se que o algoritmo tenha oportunidade igual de aprender os padrões característicos de ambas as situações clínicas, resultando em previsões mais estáveis, justas e representativas da realidade.

Antes do treinamento, realiza-se a limpeza e transformação dos dados. A exclusão de atributos irrelevantes, como contadores ou identificadores pessoais, evita que informações sem valor preditivo influenciem o modelo. Variáveis categóricas são convertidas em formato numérico (*one-hot encoding*), permitindo que o algoritmo de *Random Forest* interprete corretamente cada característica:

Código 4.4: Pré-processamento dos dados: remoção de colunas irrelevantes, codificação de variáveis categóricas e definição das variáveis preditoras (`X`) e da variável-alvo (`y`).

```
colunas_excluir = ["contador", "nome"]
X = pd.get_dummies(df.drop(
    columns=[c for c in ["contador","nome"] if c in df.columns] + ["label"]
```

```
), drop_first=True)
y = df["label"]
X = pd.get_dummies(X, drop_first=True)
```

A transformação das variáveis garante que todas as características sejam representadas numericamente, evitando vieses e permitindo que o modelo capture relações complexas entre os atributos e o resultado de interesse.

Em seguida, o conjunto de dados é dividido em treino e teste. É adotada uma proporção de 80% para treinamento e 20% para teste, mantendo a proporção de classes (*stratify*) para preservar o balanceamento:

Código 4.5: Divisão do conjunto de dados em amostras de treinamento e teste, mantendo a proporção entre classes por meio do parâmetro `stratify`.

```
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.2, random_state=42, stratify=y
)
```

A normalização dos dados é um passo fundamental antes de treinar o modelo. Foi utilizada a classe `StandardScaler`, que transforma cada variável para que tenha média zero e desvio padrão igual a um. Esse procedimento é importante porque diferentes colunas podem ter escalas muito distintas; por exemplo, na base de dados temos:

- `idade` (1–365 dias) e `peso` (2–5 kg), que possuem valores numéricos contínuos em escalas muito pequenas;
- `idademae` (15–45 anos), que possui valores muito maiores;
- variáveis categóricas transformadas em *one-hot encoding*, como `sexo` (masculino/feminino) ou `racacor` (branca, parda, preta, indígena), que se tornam 0 ou 1.

Sem padronização, atributos com valores mais altos, como `idademae`, poderiam dominar o processo de aprendizado, fazendo com que o modelo dê menos importância a variáveis menores, como `peso`. A normalização iguala a influência de todas as variáveis numéricas e binárias no treinamento:

Código 4.6: Padronização dos dados com o *StandardScaler*, garantindo que todas as variáveis possuam média zero e desvio padrão unitário.

```
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)
```

O ajuste do modelo foi realizado utilizando `RandomizedSearchCV`, que realiza uma busca aleatória sobre diferentes combinações de hiperparâmetros do *Random Forest*. Diferente da busca exaustiva (*Grid Search*), que testa todas as combinações possíveis, o

`RandomizedSearchCV` seleciona aleatoriamente um número definido de combinações (`n_iter`), permitindo explorar um espaço maior de parâmetros com menor custo computacional.

Os intervalos e opções definidos para os hiperparâmetros do modelo foram escolhidos com base em boas práticas reportadas na literatura e em testes empíricos realizados durante a calibração inicial do modelo. Cada parâmetro exerce impacto direto na capacidade preditiva e na complexidade da floresta aleatória (*Random Forest*):

- **`n_estimators (100 a 500):`** define o número de árvores na floresta. Valores menores podem gerar modelos instáveis, enquanto valores muito altos aumentam o custo computacional sem ganhos significativos após certo ponto. O intervalo escolhido permite equilibrar desempenho e tempo de treinamento.
- **`max_depth (3 a 30):`** controla a profundidade máxima das árvores. Limitar a profundidade evita o sobreajuste (*overfitting*), garantindo que o modelo aprenda padrões gerais em vez de ruídos específicos do conjunto de treinamento.
- **`min_samples_split (2 a 20):`** indica o número mínimo de amostras necessárias para dividir um nó. Valores baixos permitem divisões mais detalhadas, enquanto valores mais altos promovem árvores mais generalistas e menos propensas ao sobreajuste.
- **`min_samples_leaf (1 a 20):`** define o número mínimo de amostras exigidas em cada folha da árvore. Aumentar esse valor tende a suavizar as previsões e reduzir a variância, melhorando a capacidade de generalização.
- **`max_features ("sqrt", "log2", None):`** determina quantas variáveis são consideradas ao dividir um nó. As opções `"sqrt"` e `"log2"` são amplamente recomendadas, pois introduzem diversidade entre as árvores, reduzindo a correlação entre elas e aprimorando o desempenho global do modelo.

Esses intervalos foram, portanto, definidos para garantir um equilíbrio entre capacidade de aprendizado, robustez e custo computacional, permitindo que o processo de busca aleatória (`RandomizedSearchCV`) encontrasse uma combinação eficiente e estável de parâmetros para o modelo proposto.

Código 4.7: Definição do espaço de busca e otimização dos hiperparâmetros do modelo *Random Forest* utilizando o método *RandomizedSearchCV*.

```
param_dist = {
    "n_estimators": randint(100, 500),
    "max_depth": randint(3, 30),
    "min_samples_split": randint(2, 20),
    "min_samples_leaf": randint(1, 20),
    "max_features": ["sqrt", "log2", None],
```

```
}

rf = RandomForestClassifier(random_state=42)

search = RandomizedSearchCV(
    rf,
    param_distributions=param_dist,
    n_iter=20,
    cv=3,
    scoring=["roc_auc", "accuracy"],
    refit="roc_auc",
    n_jobs=-1,
    random_state=42
)

search.fit(X_train_scaled, y_train)
model = search.best_estimator_

print("Melhores parâmetros encontrados:", search.best_params_)
```

Os hiperparâmetros ajustados no *Random Forest* desempenham papéis específicos na construção e no desempenho do modelo:

- `n_estimators`: número de árvores na floresta. Mais árvores aumentam a estabilidade, mas também o custo computacional;
- `max_depth`: profundidade máxima de cada árvore, controlando o nível de detalhamento e evitando *overfitting*;
- `min_samples_split` e `min_samples_leaf`: número mínimo de amostras para dividir um nó e para formar uma folha, evitando divisões baseadas em poucos exemplos;
- `max_features`: número de atributos considerados em cada divisão, influenciando a diversidade das árvores.

O *Random Forest* combina múltiplas árvores de decisão treinadas em subconjuntos diferentes dos dados e das variáveis. Esse *ensemble* aumenta a robustez do modelo, reduzindo o *overfitting*, ou seja, a tendência de um modelo de se ajustar excessivamente aos ruídos ou padrões específicos do conjunto de treino, comprometendo sua capacidade de generalização para novos dados.

A escolha de *ROC AUC* como métrica principal para *refit* garante que o modelo final seja ajustado (*reajustado*) utilizando toda a base de treinamento com os hiperparâmetros que apresentaram melhor desempenho, maximizando sua habilidade de discriminar entre classes mesmo em cenários com desequilíbrio amostral, como quando há muito mais sobreviventes que óbitos.

Código 4.8: Avaliação do modelo treinado com base nas métricas de desempenho: acurácia, AUC e relatório de classificação.

```
y_pred = model.predict(X_test_scaled)
y_proba = model.predict_proba(X_test_scaled)[:, 1]

print("\nRelatório de classificação:")
print(classification_report(y_test, y_pred))
print("Acurácia:", accuracy_score(y_test, y_pred))
print("ROC AUC:", roc_auc_score(y_test, y_proba))
```

Concluído o processo de treinamento, o modelo ajustado, juntamente com o escalonador (`scaler`) e a lista de colunas utilizadas durante o treinamento, foi armazenado em disco. O escalonador é responsável por normalizar as variáveis de entrada, garantindo que cada característica seja representada em uma mesma escala e evitando que atributos com magnitudes maiores tenham influência desproporcional no modelo. Já a lista de colunas preserva a estrutura exata dos dados utilizados durante o treinamento, assegurando que novas observações sejam avaliadas de forma consistente, mesmo que apresentem diferentes conjuntos de variáveis ou ordens distintas de colunas.

Esse procedimento garante a preservação da configuração obtida e assegura sua reutilização em fases subsequentes do estudo, nas quais serão realizadas predições com novos conjuntos de dados. A adoção dessa prática contribui significativamente para a reprodutibilidade, a robustez metodológica e a integridade das análises preditivas realizadas.

Código 4.9: Salvamento do modelo treinado, do escalador e das colunas utilizadas, garantindo a reprodutibilidade do processo de predição.

```
joblib.dump(model, model_file)
joblib.dump(scaler, scaler_file)
joblib.dump(X.columns.tolist(), columns_file)
print(f"\nModelo salvo em: {model_file}")
print(f"Scaler salvo em: {scaler_file}")
print(f"Colunas salvas em: {columns_file}")
```

Além do armazenamento do modelo treinado, os resultados do processo de ajuste de hiperparâmetros foram registrados utilizando o *TensorBoard*. Essa ferramenta gera arquivos de log no formato `.tfevents`, que documentam de forma estruturada informações sobre cada iteração da busca aleatória, incluindo os valores testados para os hiperparâmetros e as métricas de desempenho correspondentes, como *ROC AUC* e acurácia.

A partir desses arquivos de log, é possível criar visualizações gráficas interativas do desempenho do modelo ao longo das diferentes combinações de parâmetros. Essas visualizações oferecem uma análise complementar às métricas numéricas tradicionais, facilitando a interpretação do impacto de cada configuração de hiperparâmetro na capacidade preditiva do modelo e auxiliando na identificação das combinações mais eficazes. A análise detalhada desses re-

gistros será apresentada no capítulo de resultados, evidenciando a integração entre avaliação quantitativa e interpretação visual.

Código 4.10: Registro dos resultados da busca de hiperparâmetros no TensorBoard, incluindo métricas de desempenho e configurações testadas.

```
log_dir = os.path.join(os.getcwd(), "logs", "random_forest")
os.makedirs(log_dir, exist_ok=True)
writer = SummaryWriter(log_dir)

results = search.cv_results_
for i in range(len(results['params'])):
   mean_roc = results['mean_test_roc_auc'][i]
      params_str = ", ".join(
      [f"{k}:{v}" for k, v in results['params'][i].items()]
   )
   writer.add_scalar("RandomizedSearch/ROC_AUC", mean_roc, i)
   writer.add_text("RandomizedSearch/Parameters", params_str, i)

writer.close()
print(f"Logs do TensorBoard em: {log_dir}")
```

4.4 Aplicação do Modelo Preditivo

Neste capítulo, o modelo preditivo previamente treinado é aplicado a um novo conjunto de dados, representando pacientes ainda não avaliados no contexto do estudo. O objetivo desta etapa é gerar previsões probabilísticas de mortalidade infantil, utilizando as transformações e configurações definidas durante o treinamento.

A base de dados `novos_pacientes` foi construída especificamente para esta análise. Inicialmente, foram geradas 100 linhas derivadas de registros existentes em outras tabelas do estudo, mas com modificações deliberadas para simular novas observações. Entre essas alterações, destacam-se mudanças na cidade de residência, variações no peso do paciente, ajustes na escolaridade da mãe e a criação de nomes fictícios para a coluna `nome`. Dessa forma, preserva-se a plausibilidade dos dados, evitando replicação direta de registros previamente analisados. Adicionalmente, registros suplementares foram inseridos manualmente, preenchendo atributos estratégicos para contemplar cenários ainda não representados, aumentando a diversidade do conjunto e fortalecendo a robustez da avaliação preditiva.

Conforme abordado anteriormente, as bibliotecas **Pandas**, **SQLAlchemy** e **Joblib** constituem ferramentas centrais para o desenvolvimento do estudo. A **Pandas** possibilita manipulação eficiente de dados tabulares, enquanto a **SQLAlchemy** fornece integração segura com o banco de dados PostgreSQL, facilitando a extração de registros. Por fim, a **Joblib** permite serializar e desserializar objetos do Python, assegurando que modelos e pré-processamentos possam ser salvos e reutilizados de forma consistente.

Código 4.11: Importação das bibliotecas necessárias para o carregamento do modelo e integração com o banco de dados PostgreSQL.

```
import pandas as pd
from sqlalchemy import create_engine
import joblib
```

Após a configuração das bibliotecas, o modelo preditivo gerado no capítulo anterior é carregado para aplicação em novos dados. Este modelo, treinado com o algoritmo Random Forest, juntamente com o escalonador (`scaler`) utilizado para normalização das variáveis e a lista de colunas do conjunto de treinamento, é carregado a partir de arquivos previamente salvos. Esse procedimento assegura que os novos dados sejam avaliados utilizando a mesma estrutura e pré-processamentos empregados durante o treinamento, garantindo consistência, integridade e reprodutibilidade das predições.

Código 4.12: Configuração da conexão com o banco de dados e carregamento dos arquivos do modelo e pré-processamento treinados.

```
# ====== Configurações ======
engine = create_engine(
    "postgresql+psycopg2://postgres:postgres@192.168.100.18:5432/db_tcc"
)

model_file = "random_forest_model.pkl"
scaler_file = "scaler.pkl"
columns_file = "model_columns.pkl"

# ====== Carregar modelo e pré-processamento ======
model = joblib.load(model_file)
scaler = joblib.load(scaler_file)
cols = joblib.load(columns_file)

print("Modelo carregado com sucesso.")
```

Em seguida, os registros de novos pacientes são carregados diretamente do banco de dados, garantindo que cada observação seja identificada corretamente, mesmo na presença de valores ausentes. O pré-processamento dos dados reproduz rigorosamente as transformações aplicadas durante o treinamento, assegurando que o modelo receba dados compatíveis com a estrutura esperada.

Código 4.13: Leitura dos registros de novos pacientes a partir do banco de dados e tratamento de valores ausentes na coluna `nome`.

```
# ====== Carregar novos pacientes ======
query = "SELECT * FROM novos_pacientes"
novos = pd.read_sql(query, engine)

# Preencher valores ausentes na coluna 'nome'
```

```
novos["nome"] = novos["nome"].fillna("Sem Nome")
```

Em seguida, realiza-se o pré-processamento, incluindo remoção de colunas irrelevantes, codificação de variáveis categóricas e reindexação para coincidir com o conjunto de colunas do modelo:

Código 4.14: Pré-processamento dos novos registros: remoção de colunas irrelevantes, aplicação de *one-hot encoding* e alinhamento das colunas com o modelo treinado.

```
# ====== Pré-processamento ======
colunas_excluir = ["contador", "nome"]

# Remover colunas de identificação e irrelevantes
novos_proc = novos.drop(
    columns=[c for c in colunas_excluir if c in novos.columns]
)

# Transformar variáveis categóricas em variáveis binárias (one-hot encoding)
novos_proc = pd.get_dummies(novos_proc, drop_first=True)

# Garantir que a estrutura de colunas seja igual  utilizada no treinamento
novos_proc = novos_proc.reindex(columns=cols, fill_value=0)
```

Neste estágio, cada passo possui finalidade específica:

- **Remoção de colunas irrelevantes ou de identificação**, como `contador` e `nome`, para evitar que atributos não informativos influenciem o modelo;
- **Codificação de variáveis categóricas** em formato binário (*one-hot encoding*) com `drop_first=True`, prevenindo multicolinearidade;
- **Reindexação das colunas** para garantir compatibilidade com o modelo treinado, preenchendo colunas ausentes com zero.

Após o pré-processamento, as predições são realizadas utilizando o modelo e o escalonador previamente carregados, e os resultados são armazenados diretamente no banco de dados, assegurando rastreabilidade e integração com os registros originais:

Código 4.15: Aplicação do modelo Random Forest sobre os novos registros e gravação das probabilidades de óbito no banco de dados.

```
# ====== Aplicar modelo para predição ======
novos_scaled = scaler.transform(novos_proc)
probs = model.predict_proba(novos_scaled)[:, 1]

# Registrar resultados no banco de dados
novos["resultado"] = probs
novos.to_sql("novos_pacientes", engine, if_exists="replace", index=False)
```

```
8
9 print("\nProbabilidade de óbito registrada com sucesso no banco.")
```

Dessa forma, cada predição é associada corretamente ao paciente correspondente, garantindo reprodutibilidade, consistência e integridade científica. A persistência dos resultados no banco de dados também permite análises futuras detalhadas e a integração do modelo em fluxos operacionais de monitoramento de mortalidade infantil.

CAPÍTULO 5

Resultados

Este capítulo detalha os resultados obtidos a partir da aplicação do modelo preditivo desenvolvido nos capítulos anteriores, destacando tanto o desempenho quantitativo do modelo quanto a interpretação das predições geradas para os novos pacientes. Além disso, são analisadas as características demográficas e socioeconômicas da população estudada, fornecendo contexto para os padrões observados e os resultados do modelo.

É importante ressaltar que todo o código produzido durante a pesquisa, incluindo scripts para pré-processamento, treinamento do modelo e geração dos gráficos apresentados, está disponibilizado no repositório GitHub[1], permitindo consulta detalhada e reprodutibilidade das análises.

5.1 Análise Exploratória dos Dados

Nesta etapa, foram examinadas variáveis demográficas e socioeconômicas de pacientes e de suas mães, de forma a caracterizar o perfil da população do estado de Mato Grosso do Sul e identificar fatores potencialmente associados à mortalidade infantil. Para isso, foram avaliadas informações como raça/cor, ocupação, idade e escolaridade materna. Esses atributos fornecem contexto essencial para a compreensão das predições do modelo e permitem correlacionar padrões observados com desfechos clínicos.

Os gráficos a seguir ilustram a distribuição de variáveis demográficas e socioeconômicas relevantes, comparando dois grupos da população estudada: **óbitos infantis com menos de 1 ano de idade** e **sobreviventes com mais de 1 ano de vida**. Essa análise permite identificar padrões e concentrações significativas, fornecendo subsídios tanto para a interpretação dos resultados do modelo preditivo quanto para futuras decisões em políticas de saúde pública.

O gráfico apresentado na Figura 5.1 descreve a distribuição racial/cor dos indivíduos.

[1] https://github.com/CauanDev/Modelo-Preditivo-TCC

Observa-se que os grupos indígenas e amarelos estão representados em proporções menores em ambos os grupos. No grupo de sobreviventes, a maior proporção é observada na categoria *branco*, com 35%, enquanto entre os óbitos de até 1 ano a maior concentração se encontra na categoria *parda*, representando 45%. Esses dados indicam diferenças significativas na composição racial entre os grupos analisados, sugerindo possíveis associações com risco de mortalidade infantil.

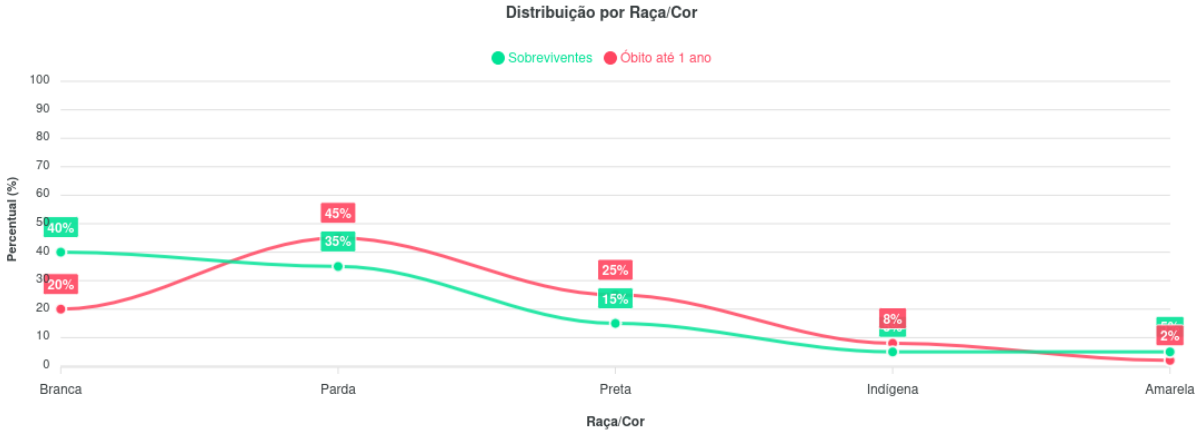

Figura 5.1: Distribuição racial/cor

A Figura 5.2 apresenta a distribuição percentual das ocupações das mães. Para este gráfico, foram incluídas apenas cinco profissões selecionadas por sua recorrência no conjunto de dados: *donas de casa*, *faxineiras*, *professoras*, *enfermeiras* e *engenheiras*. Observa-se que, no grupo de sobreviventes, predominaram ocupações com maior remuneração, enquanto no grupo de óbitos até 1 ano as categorias mais frequentes foram *dona de casa* e *faxineira*. Essa diferenciação sugere a influência de fatores socioeconômicos na mortalidade infantil.

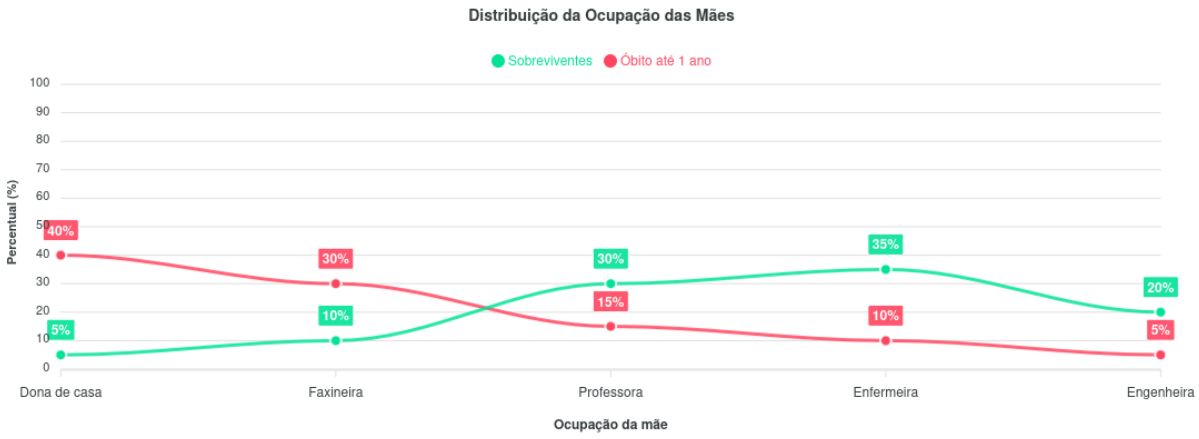

Figura 5.2: Distribuição da ocupação das mães.

O gráfico da Figura 5.3 compara a distribuição etária das mães no momento do parto. Entre os óbitos até 1 ano, observa-se maior concentração nas idades de 20 a 29 anos (45%),

seguida de mães com menos de 20 anos (30%). Já entre os sobreviventes, a faixa etária predominante é de 30 a 39 anos, correspondendo a 50% dos casos. Esses padrões sugerem que a idade materna pode exercer influência significativa no desfecho infantil.

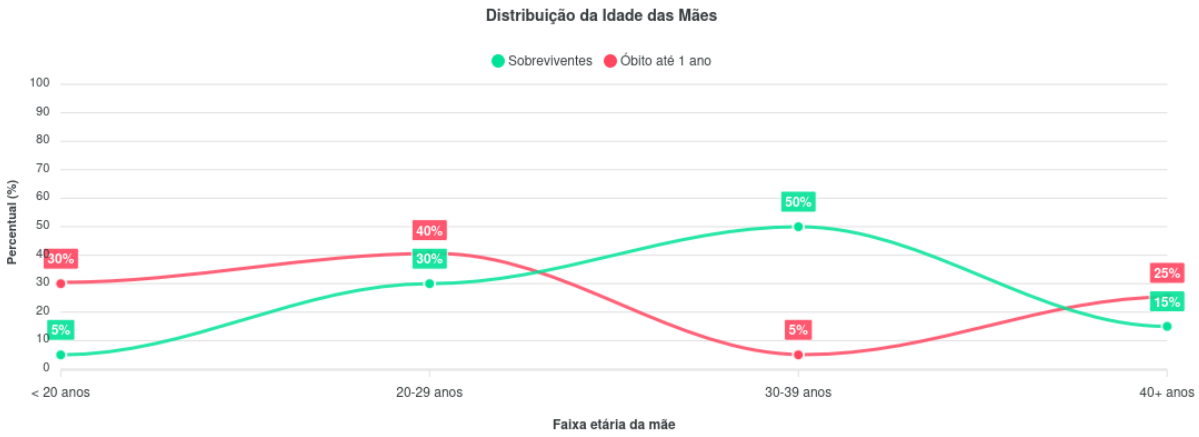

Figura 5.3: Distribuição da idade das mães no momento do parto.

Por fim, a Figura 5.4 apresenta a distribuição percentual da escolaridade das mães. No grupo de sobreviventes, a maioria das mães alcançou o ensino médio (47,26%), enquanto entre os óbitos a maior proporção se encontra no ensino fundamental (48%). Adicionalmente, 24,39% das mães de sobreviventes concluíram o ensino superior, comparado a apenas 1,61% no grupo de óbitos. Esses dados reforçam a relação entre escolaridade materna e desfecho infantil, indicando que maior nível educacional pode estar associado a menor risco de mortalidade.

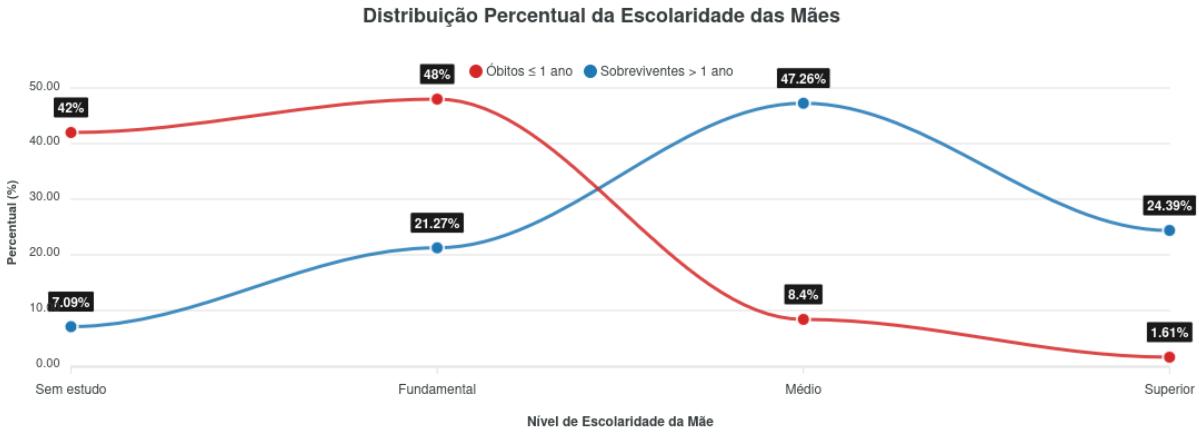

Figura 5.4: Percentual de escolaridade das mães.

A análise desses gráficos revela padrões importantes na população estudada: observa-se predominância de indivíduos do grupo pardo, diversidade ocupacional, variação significativa da idade materna e diferentes níveis de escolaridade, fatores que podem impactar direta ou indiretamente os desfechos de mortalidade infantil. Esses resultados fornecem contexto essencial para

a interpretação das predições do modelo e fundamentam as decisões metodológicas adotadas ao longo do estudo.

5.2 Resultados do Modelo

A avaliação do modelo Random Forest revelou resultados expressivos não apenas em termos de métricas globais, mas também na aplicação prática sobre o conjunto de dados da base `novos_pacientes`. Todos os registros dessa tabela foram preenchidos com os resultados previstos pelo modelo, fornecendo estimativas individuais de risco de mortalidade para cada paciente. É importante destacar que todos os dados utilizados e apresentados foram gerados de forma simulada e aleatória, não correspondendo a nenhum caso real. A Tabela 5.1 apresenta alguns exemplos desses resultados calculados pelo modelo, ilustrando a precisão e a consistência das previsões em diferentes casos.

NOME	RESULTADO	DTNASC	RACACOR	CODMUNOCOR	IDADEMAE	ESCMAE	PESO
Mariana Silva	78.65%	2019-05-01	Parda	Campo Grande	14.235	Fundamental II	1.395
Thiago Almeida	36.19%	2018-04-09	Branca	Corumbá	9.125	Superior Completo	2.080
Camila Costa	27.12%	2020-07-11	Branca	Três Lagoas	9.125	Médio	2.164
Joana Souza	92.30%	2022-02-15	Parda	Dourados	5.840	Fundamental I	1.200
Lucas Pereira	15.50%	2021-08-30	Branca	Campo Grande	28.400	Superior Completo	2.500
Ana Clara	88.75%	2023-01-20	Negra	Corumbá	5.840	Sem escolaridade	1.100
Felipe Santos	40.10%	2019-11-05	Parda	Três Lagoas	21.900	Médio	2.050

Tabela 5.1: Alguns exemplos de resultados previstos pelo modelo preditivo Random Forest para a base `novos_pacientes`. Fonte: elaborado pelo autor.

Para aprofundar a análise do desempenho do modelo, foi gerado no TensorBoard um gráfico ilustrativo da evolução da métrica ROC AUC ao longo das 20 iterações do RandomizedSearchCV. Cada ponto no eixo X representa uma combinação distinta de hiperparâmetros, enquanto o eixo Y apresenta o ROC AUC médio obtido em cada configuração. A linha azul com marcadores circulares evidencia os valores individuais de ROC AUC, enquanto a linha adicional representa o desempenho médio global, permitindo visualizar a variabilidade do modelo frente às diferentes combinações testadas. Observa-se que alguns pontos atingem picos próximos de 0.975, enquanto outros registram valores mínimos em torno de 0.73, indicando que determinadas configurações são significativamente superiores a outras.

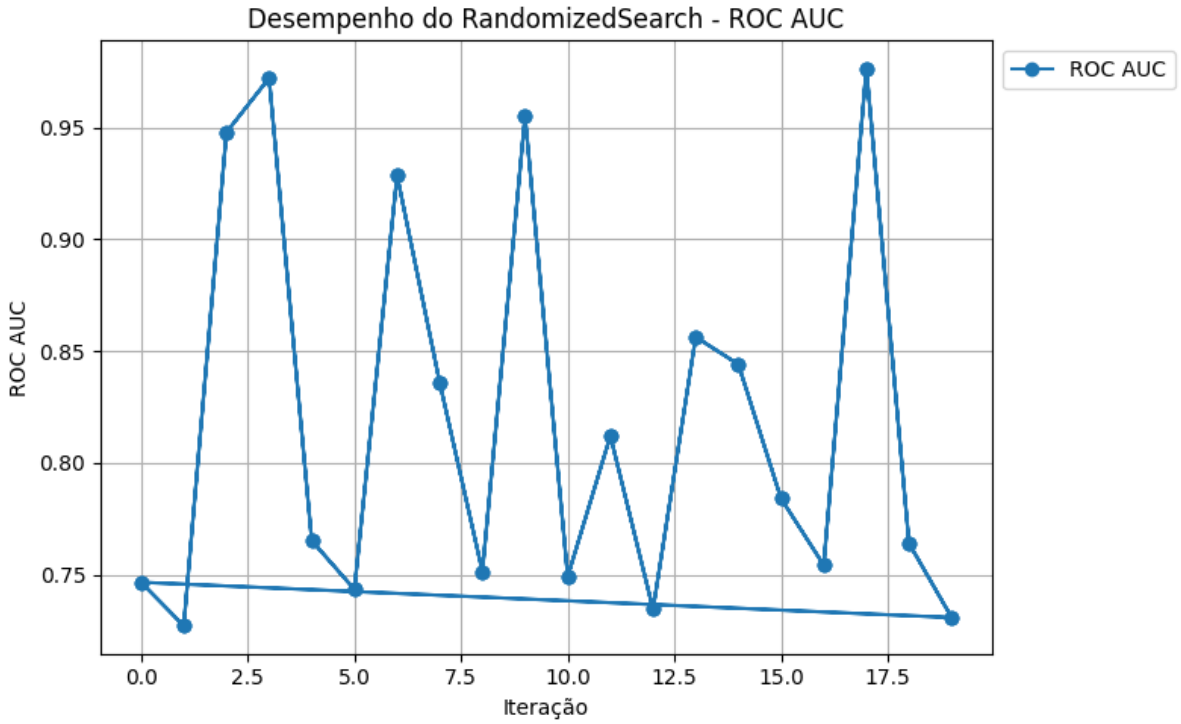

Figura 5.5: Evolução do ROC AUC durante o RandomizedSearchCV. Fonte: elaborado pelo autor.

A análise do gráfico apresentado na Figura 5.5 permite extrair conclusões importantes sobre o desempenho do modelo:

- **Variabilidade do Desempenho**: O modelo apresenta diferenças expressivas entre combinações de hiperparâmetros, com picos de ROC AUC próximos a 0,975 e mínimos em torno de 0,73, revelando a sensibilidade do Random Forest às escolhas de parâmetros.
- **Identificação de Hiperparâmetros Promissores**: As iterações que atingem os valores mais elevados de ROC AUC correspondem às configurações mais adequadas, devendo ser priorizadas para a seleção do modelo final.
- **Robustez do Modelo**: Apesar das flutuações observadas, a maioria das iterações mantém ROC AUC acima de 0,7, demonstrando a consistência e a confiabilidade do modelo mesmo frente a combinações subótimas de parâmetros.

Além das inferências derivadas do gráfico, o desempenho global do modelo Random Forest sobre o conjunto de teste da base `novos_pacientes` revelou resultados expressivos nas métricas tradicionais:

- **Acurácia**: 0,9084, indicando que aproximadamente 91% das previsões estão corretas;
- **ROC AUC**: 0,9826, evidenciando a elevada capacidade do modelo em discriminar entre pacientes com diferentes níveis de risco de mortalidade.

Esses achados corroboram a eficácia das etapas de pré-processamento e ajuste de hiperparâmetros, demonstrando que o modelo Random Forest é capaz de fornecer previsões robustas, consistentes e altamente discriminativas mesmo em dados inéditos.

5.3 Desafios Encontrados

Durante o desenvolvimento deste estudo, foram identificados diversos desafios, particularmente relacionados à disponibilidade e à integridade dos dados. Como mencionado anteriormente, do total de 228.025 registros disponíveis, 213.459 ($\approx 93,6\%$) foram desconsiderados por apresentarem dados incompletos ou inconsistentes, tornando-os inadequados para análise e modelagem preditiva. Essa restrição evidenciou a necessidade de procedimentos rigorosos de filtragem e validação, bem como o impacto significativo da qualidade dos dados na robustez e confiabilidade dos resultados obtidos.

A elevada proporção de registros descartados evidenciou a importância de realizar uma análise criteriosa da qualidade dos dados, bem como a necessidade de aplicar filtros rigorosos para assegurar a consistência das variáveis analisadas. Além disso, a presença de múltiplos campos inválidos em um mesmo registro dificultou a consolidação das informações, exigindo métodos sistemáticos para contabilizar e classificar os erros sem duplicidade na interpretação dos dados.

A Tabela 5.2 apresenta um resumo detalhado da quantidade de registros descartados em função de cada campo, permitindo identificar quais informações apresentaram maior inconsistência no conjunto de dados. Vale ressaltar que os valores por campo não são somatórios exclusivos, pois algumas linhas podem apresentar múltiplos campos inválidos e, portanto, serem contabilizadas em mais de uma categoria de erro:

Motivo	**Quantidade de registros**
Campo de raça ou cor ausente ou inválido	21.178
Campo de idade da mãe ausente ou inválido	214.337
Campo de escolaridade da mãe ausente ou inválido	216.122
Campo de ocupação da mãe ausente ou inválido	214.643
Campo de peso ao nascer ausente ou inválido	215.328
Campo de data de nascimento ausente ou inválido	423
Falta de informação sobre idade	58
Campo de sexo ausente ou inválido	12
Total	**213.459 registros**

Tabela 5.2: Motivos para exclusão de registros de óbitos. Fonte: elaborada pelo autor, com base na filtragem dos dados brutos do DATASUS.

Além dos desafios relacionados à qualidade dos dados, o processo de integração e trans-

formação dos registros exigiu considerável esforço computacional e planejamento cuidadoso, de modo a garantir que as funções implementadas fossem eficientes, escaláveis e capazes de lidar com grandes volumes de informações sem perda de precisão. Trabalhar com dados de saúde pública demanda práticas rigorosas de engenharia de dados, validação de informações e rastreabilidade de erros.

O treinamento do modelo Random Forest também apresentou desafios relevantes. A otimização dos hiperparâmetros via RandomizedSearchCV exigiu múltiplas iterações de teste e validação, demandando tempo significativo de processamento e consumo elevado de memória. Para a execução do treinamento, foi utilizado um computador com as seguintes especificações: processador AMD Ryzen 5 5600G, 16 GB de memória RAM, 1 TB de armazenamento SSD e placa de vídeo de 8 GB, o que proporcionou suporte adequado, mas ainda assim limitado frente ao volume de dados processados e à complexidade do modelo.

CAPÍTULO 6

Conclusão

O presente trabalho teve como propósito o desenvolvimento de um modelo preditivo voltado ao reconhecimento de padrões associados à mortalidade infantil no estado de Mato Grosso do Sul. Para essa tarefa, foi adotado o algoritmo *Random Forest*, direcionado à previsão do risco de mortalidade e ao consequente preenchimento integral da base `novos_pacientes` com resultados individualizados gerados pelo modelo. O processo de ajuste de hiperparâmetros foi conduzido por meio do *RandomizedSearchCV*, o que possibilitou a identificação de configurações mais adequadas e, portanto, maior eficácia preditiva. Adicionalmente, o emprego do *TensorBoard* proporcionou uma análise visual abrangente da evolução da métrica *ROC AUC*, permitindo não apenas confirmar a robustez do modelo, mas também evidenciar a variabilidade de desempenho diante das diferentes combinações de parâmetros avaliadas.

Os resultados obtidos indicam que o modelo apresenta elevado desempenho preditivo, com métricas globais de acurácia de 0,9084 e *ROC AUC* de 0,9826. Além disso, a previsão individualizada de cada registro da base demonstra a capacidade do modelo de gerar estimativas consistentes e discriminativas, fornecendo informações valiosas para a análise de risco em novos pacientes.

Apesar dos resultados promissores, algumas limitações foram identificadas. O desempenho do modelo depende fortemente da qualidade dos dados e das transformações aplicadas no pré-processamento, e o tratamento de variáveis complexas pode impactar a profundidade da análise. Além disso, a variabilidade observada nas iterações do *RandomizedSearchCV* destaca a necessidade de refinamento contínuo na escolha de hiperparâmetros para garantir consistência máxima.

Para trabalhos futuros, recomenda-se aprofundar a compreensão e o tratamento de variáveis complexas, levando em consideração as características individuais dos pacientes, bem como aprimorar a eficiência do modelo, especialmente em relação ao tempo de processamento

em bases de dados de grande volume. Além disso, sugere-se expandir a análise preditiva, incorporando novas métricas clínicas e índices de risco, e implementar mecanismos de monitoramento contínuo, de modo a assegurar que o sistema mantenha desempenho elevado à medida que novos dados sejam incorporados.

Adicionalmente, recomenda-se desenvolver um mecanismo automatizado, como uma **API**, que permita que os gráficos interativos leiam os dados diretamente do banco de dados, eliminando a necessidade de inserção manual dos valores no código. Essa abordagem possibilitaria atualizações em tempo real, maior confiabilidade dos dados apresentados e facilitaria a manutenção e escalabilidade do sistema, tornando a análise preditiva mais robusta e eficiente.

Em síntese, este estudo demonstra que a integração de técnicas avançadas de aprendizado de máquina com boas práticas de pré-processamento e análise de dados possibilita previsões robustas, confiáveis e individualizadas, representando uma contribuição significativa para sistemas automatizados de apoio à decisão clínica.

Referências Bibliográficas

Charu C. Aggarwal. *Data Mining: The Textbook*. Springer, Cham, 2015. ISBN 978-3-319-14141-1. doi: 10.1007/978-3-319-14142-8.

Wellington Sousa Aguiar. *Desenvolvimento de modelos preditivos de mortalidade infantil com base em inteligência artificial no estado do Ceará*. Tese de doutorado, Universidade Federal do Ceará, Fortaleza, CE, 2019. URL http://www.repositorio.ufc.br/handle/riufc/72867.

Chansik An, Hyunsun Lim, Dong-Wook Kim, Jung Hyun Chang, Yoon Jung Choi, and Seong Woo Kim. Machine learning prediction for mortality of patients diagnosed with covid-19: a nationwide korean cohort study. *Scientific Reports*, 10(1):1–10, 2020. doi: 10.1038/s41598-020-75205-w.

D. Armstrong. The invention of infant mortality. *Sociology of Health & Illness*, 8(3): 226–243, 1986.

G. M. Barbosa. Mortalidade infantil: aspectos epidemiológicos e sociais. *Cadernos de Saúde Pública*, 20(1):23–32, 2004.

James Bergstra and Yoshua Bengio. Random search for hyper-parameter optimization. *Journal of Machine Learning Research*, 13(Feb):281–305, 2012. URL http://www.jmlr.org/papers/v13/bergstra12a.html.

Flavia Cristina Bernardini. Introdução ao processo de kdd e md. Disponível em: https://www.professores.uff.br/fcbernardini/wp-content/uploads/sites/68/2017/08/01-IntroduÃğÃčo-a-KDD-e-DM.pdf Acesso em 28/03/2023, 2017.

Richard Junior Bertozzo. Aplicação de técnicas de aprendizado de máquina na predição de internações hospitalares no sistema Único de saúde (sus). Master's thesis, Universidade Tecnológica Federal do Paraná (UTFPR), 2022. URL http://repositorio.utfpr.edu.br/jspui/handle/1/27864.

André CARVALHO and et al. Inteligência artificial: Uma abordagem de aprendizado de máquina. 2011.

André Ponce de Leon F. de CARVALHO. Redes neurais artificiais. http://conteudo.icmc.usp.br/pessoas/andre/research/neural/, 2019. Acesso em: 02 jun. 2019.

Paulo A. CAUCHICK, Lucila N. CAMPOS, Charbel J. JABBOUR, and Ana Beatriz L. JABBOUR. *Elaboração de Artigos Acadêmicos: Estruturas, métodos e técnicas*. Elsevier, Rio de Janeiro, 1ª ed. edition, 2017.

John W. CRESWELL. *Projeto de pesquisa: Métodos qualitativo, quantitativo e misto*. Artmed, Porto Alegre, 3ª ed. edition, 2010.

Usama Fayyad, Gregory Piatetsky-Shapiro, and Padhraic Smyth. From data mining to knowledge discovery in databases. *AI magazine*, 17(3):37–37, 1996.

A. M. L. C. Feijoo. Objetivos da inferência estatística. In *A pesquisa e a estatística na psicologia e na educação*, pages 31–38. Centro Edelstein de Pesquisas Sociais, Rio de Janeiro, 2010. ISBN 978-85-7982-048-9. Disponível em: http://books.scielo.org.

Antônio Carlos GIL. *Como elaborar projetos de pesquisa*. Editora Atlas, São Paulo, 4ª ed. edition, 2007.

D. C. Gomes, D. R. Carvalho, M. R. Cubas, and M. A. H. Shmeil. Mineração de dados no serviço de atendimento de urgências. *J. Health Inform*, 6(4):141–146, oct 2014. Out./dez.

Lucas Hansen, Pushpa Raj Bhattarai, Sujan Ghimire, Hari Prasad Neupane, Kiran Kafle, Christopher Callahan, Sucharitha Myneni, and Krishna Khanal. A machine learning analysis of covid-19 mental health data. *Scientific Reports*, 12(1):15052, 2022. doi: 10.1038/s41598-022-19314-1. URL https://doi.org/10.1038/s41598-022-19314-1.

Carlos Alberto HEUSER. *Projeto de Banco de Dados*. Ed. Bookman, Porto Alegre, 6ª edição edition, 2009.

Andreas Holzinger and Igor Jurisica. Knowledge discovery and data mining in biomedical informatics: The future is in integrative, interactive machine learning solutions. *Interactive knowledge discovery and data mining in biomedical informatics: state-of-the-art and future challenges*, pages 1–18, 2014.

Gareth James, Daniela Witten, Trevor Hastie, and Robert Tibshirani. *An Introduction to Statistical Learning: with Applications in R*. Springer, New York, 2013. ISBN 978-1461471370.

Stuart P. Lloyd. Least squares quantization in pcm. *IEEE Transactions on Information Theory*, 28(2):129–137, 1982.

J. MacQueen. Some methods for classification and analysis of multivariate observations. In *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, pages 281–297, 1967.

Sidnei A. MASCARENHAS. *Metodologia Científica*. Pearson Education do Brasil, São Paulo, 2012.

Wenislayne de Araújo Miranda, Leidyanny Barbosa de Medeiros, João Agnaldo do Nascimento, Kátia Suely Queiroz Silva Ribeiro, Jordana de Almeida Nogueira, and Oriana Deyze Correia Paiva Leadebal. Modelo preditivo de retenção no cuidado especializado em hiv/aids. Master's thesis, Universidade Federal da Paraíba, João Pessoa, PB, Brasil, 2023. Dissertação de Mestrado.

Volodymyr Mnih et al. Human-level control through deep reinforcement learning. *Nature*, 518:529–533, 2015.

Organização Mundial da Saúde. 47º conselho diretor da organização pan-americana da saúde: Saúde neonatal no contexto da saúde materna, neonatal e da criança para o cumprimento das metas de desenvolvimento do milênio da declaração do milênio das nações unidas, 2006. 28p.

V. S. Ribeiro and A. A. M. Silva. Tendências da mortalidade neonatal em são luís, maranhão, brasil, de 1979 a 1996. *Cadernos de Saúde Pública*, 16(2):429–438, 2000.

Stuart Russell and Peter Norvig. *Inteligência Artificial: Uma Abordagem Moderna*. Prentice Hall, São Paulo, 3 edition, 2016.

Hellen Geremias dos Santos. Comparação da performance de algoritmos de machine learning para a análise preditiva em saúde pública e medicina. Dissertação de mestrado, Faculdade de Saúde Pública da Universidade de São Paulo (USP), 2018. URL https://pesquisa.bvsalud.org/portal/resource/pt/biblio-970061.

Whendel Muniz dos Santos, Gabriel Vitor Santos Cavalcanti, José Matheus Vilela Fonseca, and Nayara da Silva Melo. PrevisÃo de mortalidade por insuficiÊncia cardÍaca a partir de inteligÊncia artificial. *Revista Foco*, 16(9), 2023. doi: 10.54751/revistafoco.v16n9-175. URL https://ojs.focopublicacoes.com.br/foco/article/view/3182.

Ana Paula Sartorelli, Denilsen Carvalho Gomes, Marcia Regina Cubas, and Deborah Ribeiro Carvalho. Fatores que contribuem para a mortalidade infantil utilizando a mineração de dados. *Saúde e Pesquisa*, 10(1):33–41, jul. 2017. doi: 10.17765/

1983-1870.2017v10n1p33-41. URL https://periodicos.unicesumar.edu.br/index.php/saudpesq/article/view/5879.

J. M. A. Schramm and C. L. Szwarcwald. Mortalidade infantil no brasil: tendências recentes e desigualdades regionais. *Revista de Saúde Pública*, 34(3):303–312, 2000.

Ana Valeska S. Silva, Rejane M. C. Oliveira, Zélia G. Mota, and Edna Maria Chaves. *Mortalidade infantil na atenção à saúde da criança e da mulher*. Sanar, Salvador, 2016.

Eveline de Almeida Silva, Ulisses Umbelino dos Anjos, and Jordana de Almeida Nogueira. Modelo preditivo ao abandono do tratamento da tuberculose. Dissertação de mestrado, Universidade Federal da Paraíba (UFPB), João Pessoa, PB, Brasil, 2010. URL https://chooser.crossref.org/?doi=10.5935%2F0103-1104.20140018.

Felipe Cebulski Soczek[1] and Regiane Orlovski. Mineração de dados: Conceitos e aplicação de algoritmos em uma base de dados na área da saúde. *Revista Científica Semana Acadêmica*, 1(50):25, 2014.

Pang-Ning Tan, Michael Steinbach, and Vipin Kumar. *Introduction to Data Mining*. Addison-Wesley, Boston, 2006.

C. G. Victora, E. M. L. Aquino, M. C. Leal, C. A. Monteiro, F. C. Barros, and C. L. Szwarcwald. Maternal and child health in brazil: progress and challenges. *Lancet*, 377: 1863–1876, 2011.

Charles WHEELAN. *Estatística: o que é, para que serve e como funciona*. Editora Zahar, Rio de Janeiro, 1ª edição edition, 2016.