

Micaella Barcellos Pereira | Higor Gonçalves Brandão
Philippe Leal Freire dos Santos | Fernando Luiz de Carvalho e Silva
Luiz Maurício de Oliveira Monteiro

Predição e Classificação das Variáveis
Determinantes de Baixo Peso em Recém-Nascidos

UMA ABORDAGEM COM MACHINE LEARNING



Atena
Editora
Ano 2026

Micaella Barcellos Pereira | Higor Gonçalves Brandão
Philippe Leal Freire dos Santos | Fernando Luiz de Carvalho e Silva
Luiz Maurício de Oliveira Monteiro

Predição e Classificação das Variáveis
Determinantes de Baixo Peso em Recém-Nascidos

UMA ABORDAGEM COM MACHINE LEARNING



Atena
Editora
Ano 2026

Editora chefe
Prof^a Dr^a Antonella Carvalho de Oliveira
Editora executiva
Natalia Oliveira Scheffer
Assistente editorial
Flávia Barão
Bibliotecária
Janaina Ramos

2026 by Atena Editora
Copyright © 2026 Atena Editora
Copyright do texto © 2026, o autor
Copyright da edição © 2026, Atena Editora
Os direitos desta edição foram cedidos à Atena Editora pelo autor.
Open access publication by Atena Editora



Todo o conteúdo deste livro está licenciado sob a Licença Creative Commons Atribuição 4.0 Internacional (CC BY 4.0).

O conteúdo desta obra, em sua forma, correção e confiabilidade, é de responsabilidade exclusiva dos autores. As opiniões e ideias aqui expressas não refletem, necessariamente, a posição da Atena Editora, que atua apenas como mediadora no processo de publicação. Dessa forma, a responsabilidade pelas informações apresentadas e pelas interpretações decorrentes de sua leitura cabe integralmente aos autores.

A Atena Editora atua com transparência, ética e responsabilidade em todas as etapas do processo editorial. Nosso objetivo é garantir a qualidade da produção e o respeito à autoria, assegurando que cada obra seja entregue ao público com cuidado e profissionalismo.

Para cumprir esse papel, adotamos práticas editoriais que visam assegurar a integridade das obras, prevenindo irregularidades e conduzindo o processo de forma justa e transparente. Nosso compromisso vai além da publicação, buscamos apoiar a difusão do conhecimento, da literatura e da cultura em suas diversas expressões, sempre preservando a autonomia intelectual dos autores e promovendo o acesso a diferentes formas de pensamento e criação.

Predição e classificação das variáveis determinantes de baixo peso em recém-nascidos: uma abordagem com machine learning

Revisão: Os autores
Indexação: Bruna Lorena da Costa Veiga

Dados Internacionais de Catalogação na Publicação (CIP)

P923 Predição e classificação das variáveis determinantes de baixo peso em recém-nascidos: uma abordagem com machine learning / Micaella Barcellos Pereira, Higor Gonçalves Brandão, Philippe Leal Freire dos Santos, Fernando Luiz de Carvalho e Silva e Luiz Maurício de Oliveira Monteiro. – Ponta Grossa - PR: Atena Editora, 2026.

Formato: PDF

Requisitos de sistema: Adobe Acrobat Reader

Modo de acesso: World Wide Web

Inclui bibliografia

ISBN 978-65-258-3982-0

DOI: <https://doi.org/10.22533/at.ed.820262804>

1. Baixo peso ao nascer. 2. Machine Learning. 3. Predição. 4. Saúde pública. 5. SINASC. I. Título.

CDD 362.1982

Atena Editora
Ponta Grossa – Paraná – Brasil
+55 (42) 3323-5493
+55 (42) 99955-2866
www.atenaeditora.com.br
contato@atenaeditora.com.br

Conselho Editorial

Prof. Dr. Alexandre Igor Azevedo Pereira – Instituto Federal Goiano

Prof^a Dr^a Amanda Vasconcelos Guimarães – Universidade Federal de Lavras

Prof. Dr. Antonio Pasqualetto – Pontifícia Universidade Católica de Goiás

Prof^a Dr^a Ariadna Faria Vieira – Universidade Estadual do Piauí

Prof. Dr. Arinaldo Pereira da Silva – Universidade Federal do Sul e Sudeste do Pará

Prof. Dr. Benedito Rodrigues da Silva Neto – Universidade Federal de Goiás

Prof. Dr. Cirênio de Almeida Barbosa – Universidade Federal de Ouro Preto

Prof. Dr. Cláudio José de Souza – Universidade Federal Fluminense

Prof^a Dr^a Daniela Reis Joaquim de Freitas – Universidade Federal do Piauí

Prof^a Dr^a. Dayane de Melo Barros – Universidade Federal de Pernambuco

Prof. Dr. Eloi Rufato Junior – Universidade Tecnológica Federal do Paraná

Prof^a Dr^a Érica de Melo Azevedo – Instituto Federal do Rio de Janeiro

Prof. Dr. Fabrício Menezes Ramos – Instituto Federal do Pará

Prof. Dr. Fabrício Moraes de Almeida – Universidade Federal de Rondônia

Prof^a Dr^a Glécilla Colombelli de Souza Nunes – Universidade Estadual de Maringá

Prof. Dr. Humberto Costa – Universidade Federal do Paraná

Prof. Dr. Joachin de Melo Azevedo Sobrinho Neto – Universidade de Pernambuco

Prof. Dr. João Paulo Roberti Junior – Universidade Federal de Santa Catarina

Prof^a Dr^a Juliana Abonizio – Universidade Federal de Mato Grosso

Prof. Dr. Julio Candido de Meirelles Junior – Universidade Federal Fluminense

Prof^a Dr^a Keyla Christina Almeida Portela – Instituto Federal de Educação, Ciência e Tecnologia do Paraná

"Puuuuuxa vida."

Philippe Leal

RESUMO

Este trabalho aborda a predição de baixo peso ao nascer (BPN) no município de Campos dos Goytacazes, no estado do Rio de Janeiro a partir da aplicação de técnicas de *Machine Learning* (ML), utilizando dados do Sistema de Informações sobre Nascidos Vivos (SINASC) referentes ao período de 2012 a 2023. O BPN, definido pela Organização Mundial da Saúde como peso inferior a 2.500g, é um importante indicador de saúde pública por estar fortemente associado à mortalidade neonatal e a complicações futuras, como doenças cardíacas, diabetes e atraso no desenvolvimento. Apesar da relevância do tema, ainda há uma carência de estudos regionais que explorem o uso de modelos preditivos aplicados a dados brasileiros, especialmente em municípios de médio porte como Campos dos Goytacazes, o que evidencia uma lacuna entre a disponibilidade de dados públicos e a sua efetiva utilização para suporte à gestão da saúde materno-infantil. O estudo teve como objetivos principais: (i) implementar diferentes modelos de ML para prever o risco de BPN, (ii) comparar e avaliar a performance dos modelos implementados, identificando aquele com melhor desempenho, e (iii) determinar os principais fatores que influenciam no BPN por meio de técnicas de análise estatística e da interpretação dos modelos utilizando o algoritmo SHAP. Foram selecionados e tratados 399.888 registros, distribuídos em 17 variáveis socioeconômicas, clínicas e obstétricas. Após as etapas de pré-processamento e balanceamento, foram implementados os algoritmos Árvore de Decisão, Regressão Logística, AdaBoost e XGBoost. A avaliação foi realizada por meio de métricas padrão (acurácia, precisão, recall, F1-score, especificidade e AUC) e pela interpretação de variáveis via algoritmo SHAP. Os resultados apontaram a duração da gestação como o fator mais determinante em todos os modelos, seguida do Grupo de Robson, além de variáveis como Apgar, número de consultas pré-natais e tipo de gravidez. Entre os algoritmos, o XGBoost apresentou o melhor desempenho geral, seguido pelo AdaBoost, evidenciando a eficácia de métodos baseados em *boosting* para este tipo de problema. Conclui-se que os modelos de ML, em especial o XGBoost, oferecem contribuições relevantes para a identificação precoce de gestações com risco de BPN, podendo servir de apoio a políticas públicas e estratégias de saúde gestacional.

Palavras-chave: Baixo peso ao nascer. *Machine Learning*. Predição. Saúde pública. SINASC.

ABSTRACT

This study addresses the prediction of low birth weight (LBW) in the city of Campos dos Goytacazes, in the state of Rio de Janeiro, through the application of Machine Learning (ML) techniques, using data from the Live Birth Information System (SINASC) covering the period from 2012 to 2023. LBW, defined by the World Health Organization as a weight below 2,500g, is an important public health indicator, as it is strongly associated with neonatal mortality and future complications such as heart disease, diabetes, and developmental delays. Despite its relevance, there is still a shortage of regional studies exploring the use of predictive models applied to Brazilian health data, especially in medium-sized municipalities such as Campos dos Goytacazes. This highlights a research gap between the availability of public data and its effective use to support maternal and child health management. The main objectives were: (i) to implement different ML models to predict the risk of LBW, (ii) to compare and evaluate the performance of the implemented models, identifying the one with the best performance, and (iii) to determine the main factors influencing LBW through statistical analysis techniques and model interpretation using the SHAP algorithm. A total of 399,888 records were selected and processed, distributed across 17 socioeconomic, clinical, and obstetric variables. After the preprocessing and balancing steps, Decision Tree, Logistic Regression, AdaBoost, and XGBoost algorithms were implemented. Evaluation was performed using standard metrics (accuracy, precision, recall, F1-score, specificity, and AUC), as well as variable interpretation through the SHAP algorithm. The results identified gestational duration as the most decisive factor across all models, followed by the Robson Group, in addition to variables such as Apgar score, number of prenatal consultations, and type of pregnancy. Among the algorithms, XGBoost achieved the best overall performance, followed by AdaBoost, highlighting the effectiveness of boosting-based methods for this type of problem. It is concluded that ML models, especially XGBoost, provide relevant contributions to the early identification of pregnancies at risk of LBW, potentially supporting public policies and gestational health strategies.

Keywords: Low birth weight. Machine Learning. Prediction. Public health. SINASC.

LISTA DE ILUSTRAÇÕES

Figura 1 – Matriz de Correlação	23
Figura 2 – Representação de uma árvore de decisão	27
Figura 3 – Função sigmoide	29
Figura 4 – Fluxograma	40
Figura 5 – Distribuição dos Registros por Sexo	47
Figura 6 – Distribuição de Recém-nascidos por Peso (Normal vs Baixo Peso)	48
Figura 7 – Relação de Valores Ausentes por Atributo	50
Figura 8 – Gráfico de Setores da Distribuição de Peso	51
Figura 9 – Matriz de Confusão do XGBoost (Normal vs Baixo Peso)	70
Figura 10 – Valores médios de importância dos <i>Shapley Values</i> para o XGBoost	72
Figura 11 – Valores locais de importância dos <i>Shapley Values</i> para o XGBoost	73

LISTA DE TABELAS

Tabela 1 – Matriz de Confusão	32
Tabela 2 – IDH e população das cidades de acordo com o último Censo disponível	43
Tabela 3 – Quantidade de Nascidos Vivos por Cidade Seleccionada	43
Tabela 4 – Quantidade de Valores Ignorados nas Variáveis Seleccionadas	48
Tabela 5 – Dicionário das Variáveis Seleccionadas (Parte 1)	49
Tabela 6 – Dicionário das Variáveis Seleccionadas (Parte 2)	50
Tabela 7 – Métricas principais dos modelos de classificação	69
Tabela 8 – Três principais variáveis por modelo segundo os valores médios de SHAP	73

LISTA DE CÓDIGOS

5.1 Seleção das Bibliotecas	54
5.2 Seleção dos Dados	55
5.3 Seleção dos Dados	56
5.4 Criação da base com as colunas selecionadas	57
5.5 Codificação das Variáveis Categóricas	58
5.6 Binarização da variável de interesse	58
5.7 Exclusão dos valores ignorados	59
5.8 Remoção das linhas com valores ausentes	59
5.9 Preenchimento com a mediana	60
5.10 Subamostragem Aleatória	60
5.11 Separação da base entre conjuntos de treino e teste	61
5.12 Criação das Variáveis de Entrada	62
5.13 Modelo AdaBoost	62
5.14 Modelo Árvore de Decisão	63
5.15 Cálculo da Importância de Gini	63
5.16 Modelo Regressão Logística	64
5.17 Modelo XGBoost	65
5.18 Métricas de Desempenho	66
5.19 SHAP no XGBoost para preparação dos dados e bar plot de importâncias	67

LISTA DE ABREVIATURAS E SIGLAS

AUC	Área Sob a Curva
BPN	Baixo Peso ao Nascer
CGIAE	Coordenação-Geral de Informações e Análises Epidemiológicas
DN	Declaração de Nascidos Vivos
FN	Falso Negativo
FP	Falso Positivo
IA	Inteligência Artificial
ML	<i>Machine Learning</i>
MLP	<i>Multilayer Perceptron</i>
OMS	Organização Mundial da Saúde
ROC	<i>Receiver Operating Characteristic</i>
SHAP	<i>Shapley Additive Explanations</i>
SIM	Sistema de Informação sobre Mortalidade
SINASC	Sistema de Informações sobre Nascidos Vivos
SVSA	Secretaria de Vigilância em Saúde e Ambiente
TPU	Unidade de Processamento Tensorial
VN	Verdadeiro Negativo
VP	Verdadeiro Positivo

SUMÁRIO

1	INTRODUÇÃO	14
1.1	Justificativa	15
1.2	Objetivos	16
1.2.1	Objetivos Gerais	16
1.2.2	Objetivos Específicos	16
2	FUNDAMENTAÇÃO TEÓRICA	17
2.1	Definição do Problema	17
2.2	<i>Machine Learning</i> (Aprendizado de Máquina)	18
2.2.1	Técnicas de <i>Machine Learning</i>	19
2.3	Análise dos Dados	19
2.3.1	Base de Dados SINASC	20
2.3.2	Pré-processamento dos Dados	21
2.3.2.1	Limpeza dos Dados	21
2.3.2.2	Transformação dos Dados	22
2.3.2.3	Matriz de Correlação	22
2.4	Desbalanceamento de Classes	23
2.4.1	Métodos de Amostragem	24
2.5	Algoritmos de ML	24
2.5.1	<i>Adaptive Boosting</i>	25
2.5.2	Árvores de Decisão	26
2.5.3	Regressão Logística	28
2.5.4	XGBoost	30
2.6	Treinamento dos Modelos	31
2.7	Avaliação do Desempenho dos Modelos	32
2.8	Definição das Variáveis com Mais Impacto na Predição dos Modelos	34
2.8.1	<i>Shapley Additive Explanations</i>	34
3	TRABALHOS RELACIONADOS	37

3.1	Trabalhos Selecionados	37
4	METODOLOGIA	40
4.1	Revisão da Literatura	40
4.2	Definição das Ferramentas e Tecnologias	41
4.2.1	Seleção da Base de Dados	41
4.2.2	Ambiente de Desenvolvimento, Linguagem de Programação e Bibliotecas	41
4.3	Seleção dos Dados	43
4.4	Seleção das Variáveis	44
4.4.1	Variáveis Socioeconômicas e Demográficas	44
4.4.2	Variáveis sobre a Mãe	45
4.4.3	Variáveis sobre a Gestação e o Parto	45
4.4.4	Variáveis sobre o Bebê Nascido	45
4.5	Codificação das Variáveis Categóricas	46
4.6	Classificação Binária da Variável de Interesse	47
4.7	Exclusão de valores preenchidos como "Ignorado"	48
4.8	Tratamento de Valores Ausentes	50
4.9	Tratamento do Desbalanceamento de Classes	51
4.10	Seleção dos Algoritmos de <i>Machine Learning</i>	52
4.11	Avaliação das Métricas e Identificação das Variáveis Relevantes	53
5	DESENVOLVIMENTO	54
5.1	Ambiente de Desenvolvimento e Pré-Configurações	54
5.2	Importação das Bibliotecas	54
5.3	Configuração da Base de Dados	55
5.4	Seleção dos Dados	56
5.5	Seleção das Variáveis	57
5.6	Codificação das Variáveis Categóricas	57
5.7	Classificação Binária da Variável de Interesse	58
5.8	Exclusão de valores preenchidos como "Ignorado"	58
5.9	Remoção de Linhas com Valores Ausentes	59
5.10	Preenchimento dos Campos Ausentes com a Mediana	60

5.11	Tratamento do Desbalanceamento de Classes com a Subamostra- gem Aleatória	60
5.12	Divisão da Base em Dados de Treino e Teste	61
5.13	Criação das Variáveis de Entrada	61
5.14	Criação dos Modelos	62
5.14.1	AdaBoost	62
5.14.2	Árvores de Decisão	63
5.14.3	Regressão Logística	63
5.14.4	XGBoost	65
5.15	Cálculo das Métricas de Desempenho	66
5.16	Execução do SHAP	67
6	RESULTADOS E DISCUSSÕES	69
6.1	Definição do Modelo que Obteve Melhor Desempenho	69
6.2	Interpretação dos Resultados para Identificação das Variáveis mais Importantes para a Predição	71
7	CONCLUSÕES E TRABALHOS FUTUROS	76
7.1	Conclusões	76
7.2	Trabalhos Futuros	77
	REFERÊNCIAS	78

1 INTRODUÇÃO

A sobrevivência infantil, o desenvolvimento físico e mental, assim como o estado de saúde e o histórico da mãe, estão ligados a um importante indicador de saúde: o peso ao nascer (ARAYESHGARI et al., 2023). O baixo peso ao nascer (BPN) é determinado pela Organização Mundial da Saúde (OMS) por valores abaixo de 2.500g e é considerado um dos principais fatores associados à mortalidade neonatal. É importante destacar que o BPN é definido por outros critérios além do peso, como a idade gestacional no momento do parto e a taxa de crescimento fetal. Isso se deve ao fato de que recém-nascidos prematuros, em geral, apresentam baixo peso decorrente da falta de desenvolvimento adequado, tal como é esperado do recém-nascido a termo – nascidos a partir da 37^a semana de gestação (SANTOS et al., 2021).

No mundo, um em cada sete recém-nascidos sofre de baixo peso ao nascer (UNICEF, 2023). Nestes, é possível observar o aumento na probabilidade de óbito em seu primeiro mês de vida, e em caso de sobrevivência, sofrem com condições crônicas, como obesidade, diabetes e doenças cardíacas (MOREIRA; SOUSA; SARNO, 2018).

A OMS estabeleceu como meta reduzir o número de bebês com baixo peso em 30% até 2025. Estima-se que as taxas de BPN sejam de cerca de 7% em países desenvolvidos, 16,5% em países em desenvolvimento, e 18,6% nos países menos desenvolvidos. Dessa forma, a atenção à saúde materna, incluindo nutrição adequada, acesso a cuidados pré-natais e monitoramento contínuo, é essencial para prevenir o nascimento de bebês com baixo peso (ARAYESHGARI et al., 2023).

De acordo com dados do Sistema de Informações sobre Nascidos Vivos (SINASC), principal fonte de dados do país no tocante à saúde neonatal, o Brasil apresentou no ano de 2022 cerca de 9% dos recém-nascidos diagnosticados com BPN. A região sudeste apresentou a maior incidência de casos, com o estado de São Paulo liderando com 51.742. No que se refere aos dados disponibilizados de 2019 a 2022, também é notada a presença da região Sudeste ocupando a região de maiores ocorrências, com o Rio de Janeiro sendo o terceiro estado com maiores números (BRASIL, 2020). Esses números são resultantes principalmente das más condições que a mãe enfrenta durante o período de gestação,

como falta de assistência e dificuldades socioeconômicas (SANTOS et al., 2021).

Diante desse cenário, novas técnicas têm possibilitado avanços na análise de dados em saúde, principalmente com a aplicação de Inteligência Artificial (IA). *Machine Learning* (ML), ou Aprendizagem de Máquina, é uma dessas tecnologias que vêm permitindo a identificação de padrões em grandes volumes de dados e oferecendo novas formas de predição de resultados com maior precisão (COLLIN et al., 2022). Na área da saúde, algoritmos de ML podem ser usados como ferramentas para realizar a prévia de riscos de doenças ou diagnósticos a partir de dados informativos do paciente, auxiliando profissionais da saúde nas tomadas de decisões (FERNANDES; FILHO, 2019).

Machine Learning é uma subárea da IA que busca, por meio de algoritmos computacionais e técnicas matemáticas, identificar padrões em conjuntos de dados. Seu objetivo é a previsão de um resultado de interesse, que pode ser gerado pela construção de um sistema orientado e capacitado. A partir de conjuntos de dados pré-definidos, é capaz de gerar modelos de predição, classificação ou identificação. Algoritmos de ML têm sido amplamente utilizados em áreas como logística, robótica, e sistemas de detecção de fraudes bancárias, entre outros (PAIXAO et al., 2022).

Assim, reconhecendo os potenciais riscos do BPN e o papel que o ML pode desempenhar nesse contexto, este estudo busca analisar os principais indicadores que influenciam o baixo peso ao nascer, aplicando técnicas de ML para prever os casos de BPN (COLLIN et al., 2022).

1.1 JUSTIFICATIVA

A implementação de modelos de *Machine Learning* para a predição de baixo peso ao nascer se justifica pela importância da identificação dos seus principais fatores influenciadores. Utilizar dados reais da saúde pública do Brasil permitirá o desenvolvimento de soluções computacionais robustas, eficientes e adaptadas ao contexto local, refletindo a realidade de nosso país em um dos responsáveis pela taxa de mortalidade infantil.

O uso dessas ferramentas computacionais permitirá um apoio mais preciso ao planejamento e à gestão de recursos na saúde pública, proporcionando maior eficiência no acompanhamento pré-natal e na identificação de gestantes em situação de risco. Para o

setor de tecnologia, os resultados poderão servir como base para o desenvolvimento de novas ferramentas e plataformas que utilizem modelos preditivos, ampliando o uso de inteligência artificial em diferentes áreas da saúde.

1.2 OBJETIVOS

1.2.1 Objetivos Gerais

Este trabalho tem como objetivo desenvolver, aplicar e avaliar modelos preditivos baseados em técnicas de *Machine Learning* a partir de dados públicos de nascimentos e variáveis socioeconômicas para prever o risco de baixo peso ao nascer no município de Campos dos Goytacazes, no estado do Rio de Janeiro.

1.2.2 Objetivos Específicos

- Implementar diferentes modelos de ML para prever o risco de baixo peso ao nascer.
- Comparar e avaliar a performance dos modelos de ML implementados, identificando aquele com melhor desempenho.
- Determinar os principais fatores que influenciam no baixo peso ao nascer, por meio de técnicas de análise estatística e da interpretação dos modelos utilizando o algoritmo SHAP.

2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo, abordam-se os conceitos fundamentais relacionados ao baixo peso ao nascer e às metodologias de aprendizado de máquina aplicáveis a essa temática. A definição do problema é apresentada, destacando a relevância do baixo peso ao nascer como um desafio de saúde pública.

Em seguida, discute-se o aprendizado de máquina, suas técnicas e a relação com a análise de dados, especificamente na base do Sistema de Informações sobre Nascidos Vivos (SINASC). Também explora-se o desbalanceamento de classes, apresentando métodos de amostragem, como subamostragem e sobreamostragem, para lidar com essa questão.

Por fim, serão apresentados os algoritmos de aprendizado de máquina selecionados para o estudo, além das etapas de treinamento, avaliação de desempenho dos modelos e definição das variáveis mais impactantes na predição. Essa fundamentação teórica servirá como base para as análises e discussões subsequentes.

2.1 DEFINIÇÃO DO PROBLEMA

O baixo peso ao nascer (BPN) é uma condição caracterizada pelo peso inferior a 2.500 gramas ao nascimento, sendo um indicador crítico da saúde neonatal e um fator de risco para complicações a curto e longo prazo. O BPN não apenas afeta a saúde imediata dos recém-nascidos, aumentando o risco de mortalidade e morbidade, mas também pode ter consequências duradouras, como dificuldades no desenvolvimento cognitivo e motor, predisposição a doenças crônicas e redução da qualidade de vida (MOREIRA; SOUSA; SARNO, 2018).

Diversos fatores contribuem para o BPN, incluindo condições socioeconômicas, saúde materna e comportamentos de risco durante a gestação. Mulheres que enfrentam pobreza, baixa escolaridade e falta de acesso a cuidados de saúde adequados têm maior probabilidade de dar à luz bebês com baixo peso. Além disso, a saúde materna, que inclui a presença de doenças crônicas e complicações durante a gestação, também desempenha um papel fundamental. A nutrição inadequada, a ingestão de substâncias como álcool e

tabaco, e a ausência de acompanhamento médico regular aumentam a probabilidade do BPN (ARAYESHGARI et al., 2023).

O fenômeno do BPN não é apenas um problema de saúde individual, mas também um desafio que afeta comunidades inteiras, pois está ligado a questões sociais e políticas mais amplas. Portanto, entender o BPN requer uma abordagem multidisciplinar que considere a interconexão entre fatores sociais, econômicos e de saúde, além de demandar a identificação de fatores chave que contribuem para essa condição (SANTOS et al., 2021).

2.2 MACHINE LEARNING (APRENDIZADO DE MÁQUINA)

Pode-se definir o *Machine Learning* (ML) como um conjunto de técnicas, ferramentas e métodos que têm como objetivo o reconhecimento de padrões e a classificação de informações em grandes volumes de dados (SANTOS, 2021). De acordo com Sarker (2021), o ML é uma subárea da Inteligência Artificial (IA), permitindo que sistemas aprendam e se aprimorem a partir de experiências prévias, sem a necessidade de uma programação explícita que defina cada passo do processo. Nos últimos anos, o ML vem ganhando relevância significativa, sendo frequentemente citado como uma das tecnologias mais influentes da quarta revolução industrial (FREITAS, 2023).

Segundo Moreira (2023), essa crescente relevância está diretamente relacionada ao surgimento e à disponibilidade de grandes conjuntos de dados, como no caso do Brasil, onde sistemas como o SINASC e o Sistema de Informação sobre Mortalidade (SIM) fornecem uma quantidade massiva de informações (BRASIL, 2020). Esses grandes volumes de dados tornam viáveis o treinamento, a validação e o teste de modelos de ML, permitindo previsões e classificações de alta precisão e eficiência.

Dessa forma, o ML é tido como um método eficiente de obter informações e reconhecer padrões importantes a partir de grandes volumes de dados, tornando-se cada vez mais notável, especialmente no campo da saúde. Os algoritmos de ML orientam o processo de aprendizado dos modelos, definindo como os dados são processados, transformados e ajustados para gerar previsões (BORBA, 2023).

2.2.1 Técnicas de *Machine Learning*

O ML pode ser classificado em três categorias principais: aprendizado supervisionado, não-supervisionado e por reforço. No aprendizado supervisionado, os algoritmos utilizam dados rotulados, ou seja, com respostas pré-definidas, para treinar o modelo. Esse método inclui tanto classificação, que agrupa os dados em categorias discretas, quanto regressão, que prevê valores contínuos. No aprendizado não-supervisionado, os dados fornecidos não possuem rótulos, e o algoritmo busca padrões ou agrupamentos dentro dos dados (GREENER et al. (2022 *apud* PEIXOTO, 2023)). Por fim, o aprendizado por reforço envolve um agente que aprende por tentativa e erro, recebendo recompensas ou penalidades conforme toma decisões, com o objetivo de maximizar a recompensa ao longo do tempo (MOREIRA, 2023).

Neste trabalho, será utilizado a abordagem via aprendizado supervisionado voltado à classificação, com o objetivo de prever quais casos apresentam maior risco de baixo peso ao nascer. A classificação em ML é um método utilizado para categorizar dados dentro de um conjunto, auxiliando na tomada de decisões diante das questões que surgem após a análise do conteúdo. Nesse contexto, a classificação binária se refere à simplificação do problema de classificação ao pertencimento a um de dois conjuntos distintos (SOARES et al., 2021). Para isso, métodos de classificação binária serão utilizados a fim de descrever a variável dependente em dois rótulos: zero para identificar bebês com peso normal e um para os nascidos com baixo peso.

2.3 ANÁLISE DOS DADOS

A qualidade e o tamanho da base de dados são aspectos cruciais para o sucesso da aplicação de modelos de ML, já que é fundamental que os dados contenham variáveis que expliquem corretamente a variável de interesse. Dessa forma, a base utilizada deve ser obtida de uma fonte confiável, de forma que os dados presentes nela estejam de acordo com a realidade. Para mitigar possíveis problemas que podem vir a existir em uma base, como dados faltantes, incorretos, quantidade de dados disponíveis, desbalanceamento de classes, dentre outros, é necessário aplicar técnicas específicas que garantam a eficácia do modelo (BORGES, 2020). O enriquecimento e ajuste da base proporciona um modelo mais coerente com a realidade, ajudando na melhoria dos resultados e possibilitando

descobrir variáveis de alta relevância para a predição proposta (MOREIRA, 2023).

2.3.1 Base de Dados SINASC

O Sistema de Informações sobre Nascidos Vivos (SINASC) teve sua implementação realizada no ano de 1990, tendo a finalidade de registrar os dados sobre os nascimentos ocorridos por todo o ambiente nacional, fornecendo dados relevantes para todos os níveis do Sistema de Saúde. Trata-se de uma base de acesso público, sem a possibilidade de identificação individual do nascido ou dos pais (Brasil, 2024).

Ao que se refere à coleta e processamento dos dados do SINASC, seu processo é essencialmente baseado na Declaração de Nascidos Vivos (DN), um documento fundamental utilizado em todo o Brasil para a coleta de informações sobre nascimentos, sendo indispensável para a emissão da Certidão de Nascimento pelos Cartórios de Registro Civil (Brasil, 2024).

Esse documento é emitido em três vias, numeradas sequencialmente, com sua produção e distribuição para os estados sendo de responsabilidade do Ministério da Saúde. As Secretarias Estaduais de Saúde, por sua vez, são encarregadas de repassar as DN aos municípios. As Secretarias Municipais de Saúde ficam responsáveis por controlar a distribuição das DN entre as unidades de saúde e os Cartórios de Registro Civil. Além disso, formulários também são disponibilizados para profissionais de saúde e parteiras tradicionais, desde que vinculadas a uma unidade de saúde legal, para atender a partos domiciliares, com o controle realizado pelas Secretarias Municipais de Saúde (Brasil, 2024).

Os dados do SINASC são coletados por meio das DN, preenchidas por profissionais de saúde ou parteiras tradicionais após o parto. Essas declarações são recolhidas pelas Secretarias Municipais de Saúde, onde são digitadas, processadas, verificadas e consolidadas. Em seguida, os dados são transferidos para a base estadual, onde é enriquecida e, posteriormente, para o nível federal, através da internet, de forma simultânea entre os três níveis de gestão (municipal, estadual e federal). No nível federal, a Secretaria de Vigilância em Saúde e Ambiente (SVSA), gestora do SINASC, é responsável pela análise e disseminação das informações, agregando os dados por estado e elaborando relatórios e painéis de indicadores sobre natalidade. A Coordenação-Geral de Informações e Análises

Epidemiológicas (CGIAE), pertencente à estrutura funcional do SINASC, trata da análise, avaliação e distribuição das informações sobre o sistema, agregando-as por Estado, e elaborando relatórios analíticos, painéis de indicadores e outros instrumentos estatísticos de informações sobre natalidade que são disseminados para todo o país (Brasil, 2024).

2.3.2 Pré-processamento dos Dados

A fase de processamento de dados tem início na coleta dos dados, partindo para a compreensão das informações e resolução dos problemas identificados sobre o conjunto obtido. Essas práticas adequam os dados para a fase de extração de conhecimento sobre os mesmos, além de preparar o terreno para a construção dos modelos de ML (Batista, 2003). Segundo Freitas (2023), esse período é essencial para o preparo dos dados para o desenvolvimento do projeto ao qual eles foram propostos.

Conforme Batista (2003), o pré-processamento de dados é um processo que é inteiramente dependente da capacidade do autor de realizar a identificação dos problemas e de sua natureza nos dados, sendo também necessário que ele tenha a capacidade de discernir qual a melhor abordagem para solucionar os obstáculos que surgirem. Ainda em consoante com o citado, é possível classificar o problema abordado neste trabalho como uma tarefa fracamente dependente de conhecimento de domínio, visto que é factível que os métodos selecionados para o tratamento dos casos de BPN sejam capazes de realizar a extração de conhecimento suficiente para tratar os empecilhos que surgem durante o pré-processamento.

Esse processo engloba as fases de limpeza, integração, transformação e redução dos dados, de modo a permitir a padronização do conjunto utilizado na construção dos modelos (SIVAKUMAR; GUNASUNDARI, 2017).

2.3.2.1 Limpeza dos Dados

Os dados encontrados em bases reais costumam ser incompletos e inconsistentes, o que incide na necessidade de realizar a fase de limpeza. Essa etapa se propõe a repor valores ausentes, identificar dados atípicos (*outliers*) e corrigir inconsistências encontradas no conjunto (SIVAKUMAR; GUNASUNDARI, 2017).

2.3.2.2 Transformação dos Dados

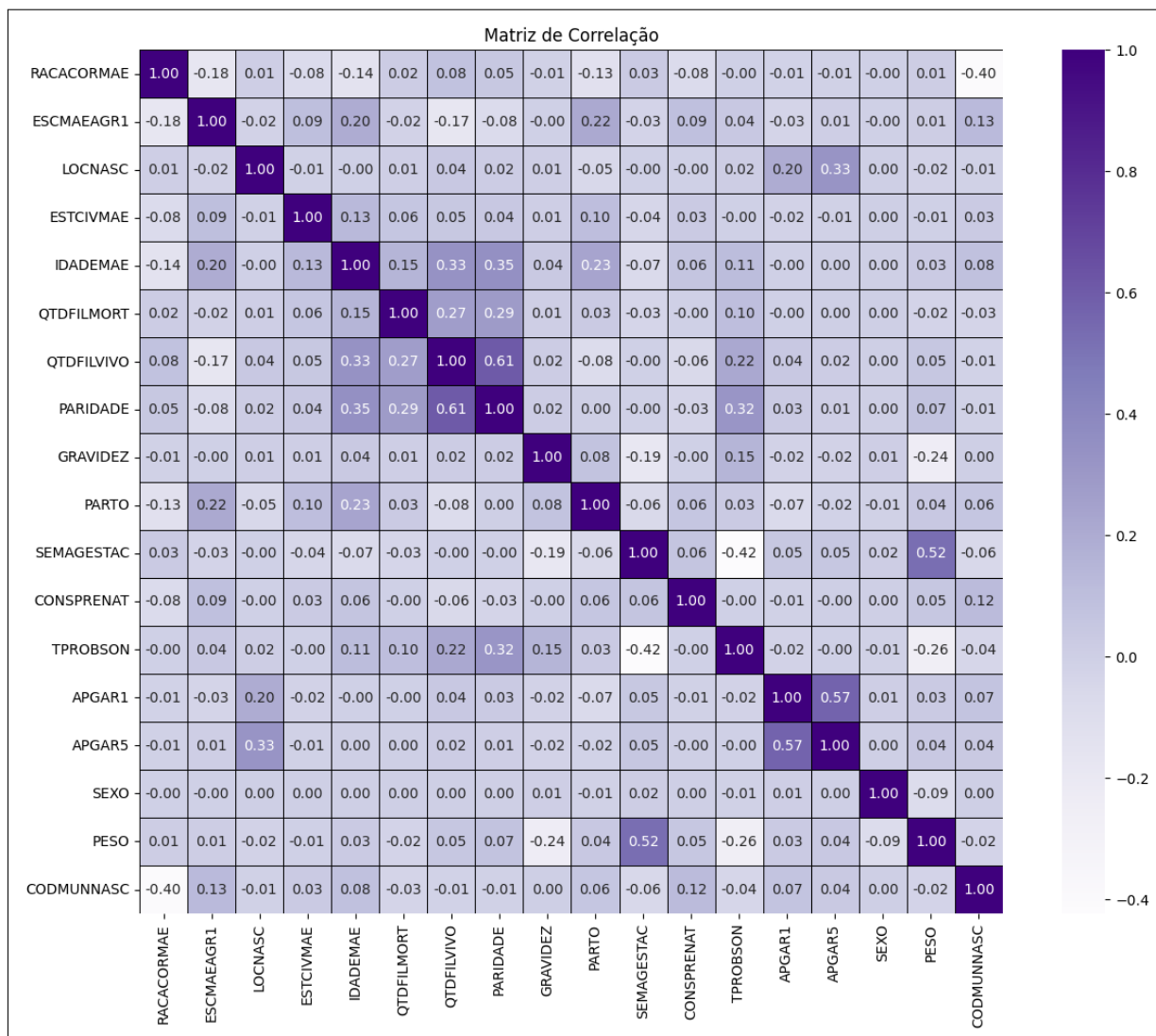
A fase de transformação compreende a necessidade de tornar o conjunto de dados numa representação que transponha as limitações que surjam durante a aplicação do algoritmo. Isso pode ocorrer nos casos em que, por exemplo, o algoritmo apenas trate intervalos numéricos, porém o modelo é alimentado por um conjunto que possui formatos textuais, de data, dentre outros. Uma das principais formas de transformação é a de codificação dos dados, a qual será utilizada no decorrer deste trabalho (BATISTA, 2003).

A codificação dos dados se refere ao processo de adequação dos valores de uma variável, transformando os valores categóricos em valores numéricos distintos. Por exemplo, dada uma variável de raça com os valores "amarelo", "pardo", "negro", "branco" e "indígena", com a aplicação da codificação uma possível resultante seriam os valores 1, 2, 3, 4 e 5 para representação numérica das categorias supracitadas.

2.3.2.3 Matriz de Correlação

A matriz de correlação é a representação gráfica do cálculo do coeficiente de correlação, o qual quantifica a associação dentre as variáveis encontradas num conjunto de dados. Essa correlação é calculada individualmente entre duas variáveis, até que iterativamente o cálculo tenha sido feito por todas os pares. Como exemplo, na Figura 1 é possível analisar essa representação, em que as diagonais sempre têm valor 1 na interseção de x e y , pois esses casos em específico se referem à correlação da variável com ela própria. Nos demais, quanto mais próximo de 1 o valor, mais forte é a associação das duas variáveis. Caso contrário e portanto mais perto de -1, a correlação é fortemente negativa (SILVA, 2021).

Figura 1 – Matriz de Correlação



Fonte: Autores, 2024.

2.4 DESBALANCEAMENTO DE CLASSES

Pode-se definir o desbalanceamento de classes como a amostragem irrisória da variável de interesse (classe minoritária) em comparação com as demais categorias (classes majoritárias) encontradas dentro do conjunto de dados (PEIXOTO, 2023).

Conforme destacado por Moreira (2023), essa característica possui uma forte inclinação a produzir modelos que favorecem a classe majoritária, o que compromete a capacidade de identificar corretamente os casos minoritários. Além disso, Machado (2009) ressalta que classificadores treinados com dados desbalanceados apresentam uma alta incidência de falsos negativos nas classes sub-representadas.

2.4.1 Métodos de Amostragem

O tratamento do problema elucidado acima é compreendido pela aplicação dos ditos métodos de amostragem, cujo objetivo é balancear a distribuição dos dados de modo a equiparar a quantidade de amostras entre as classes minoritárias e majoritárias.

Eles se dividem em três técnicas de amostragem (MACHADO, 2009):

- **Subamostragem (*undersampling*):** técnica que retira aleatoriamente amostras da classe majoritária até que o conjunto de dados esteja satisfatoriamente balanceado (ALBISUA et al. (2013) *apud* PEIXOTO, 2023).
- **Sobreamostragem (*oversampling*):** técnica que realiza a replicação de casos da classe minoritária de modo a balancear a distribuição de amostras entre ambas as classes (ALBISUA et al., 2013). Um exemplo é o SMOTE (*Synthetic Minority Oversampling Technique*).
- **Combinação de subamostragem com sobreamostragem:** combina ambas as técnicas citadas acima.

2.5 ALGORITMOS DE ML

De acordo com Borba (2023), um algoritmo de ML tem por função determinar como um modelo aprende a partir dos dados, abrangendo a forma como ele processa, transforma e ajusta os dados para fazer realizar suas previsões. Sendo assim responsáveis pela modelagem dos dados que consome, e o teste que irá avaliar sua performance a partir da predição em novos dados.

Paixao et al. (2022) afirmam que o desenvolvimento de um algoritmo de ML é feito por um processo de três partes: pré-processamento, treinamento e avaliação do modelo. A etapa inicial envolve a organização do banco de dados, definição da questão de pesquisa e separação dos dados em conjuntos de treinamento e teste. Durante o treinamento, o aprendizado pode ser feito a partir das duas abordagens: supervisionado ou não supervisionado. Já na fase de avaliação, o modelo é testado a partir de um conjunto de dados separado, e os seus resultados são comparados com os valores reais. Dessa forma, os algoritmos de aprendizado de máquina aprendem por meio de observações repetidas,

construindo um mapeamento que permite rotular com precisão e confiança novos dados que não foram apresentados em nenhum momento antes ao algoritmo.

Paixao et al. (2022) também concluem que o processo para o desenvolvimento de um algoritmo de ML ser eficiente, ele deve ser realizado a partir de uma base de dados consolidada e validada, pois podem gerar resultados deturpados caso contrário.

2.5.1 Adaptive Boosting

Um modelo de *ensemble* consiste em uma técnica de *Machine Learning* que combina as previsões de múltiplos modelos base para produzir uma previsão única, geralmente mais precisa do que a obtida por qualquer modelo individual (MUREL; KAVLAKOGLU, 2024). Um dos algoritmos mais conhecidos dessa família é o *Adaptive Boosting*, ou simplesmente AdaBoost. Ele combina diversos classificadores, que iniciam com pesos iguais, com o objetivo de melhorar a precisão do modelo em treinamento (MOREIRA, 2023). À medida que o algoritmo é executado, os classificadores que cometem mais erros recebem um peso maior e são priorizados nas próximas iterações, até que os erros sejam minimizados e o limite de erro residual seja atingido (PEIXOTO, 2023).

Assim, o AdaBoost é capaz de criar um classificador forte a partir da combinação de classificadores fracos. O nome "classificador adaptável" surge justamente porque o algoritmo ajusta e aprimora seu desempenho conforme os erros são identificados e corrigidos ao longo do processo. Dessa forma, os classificadores fracos subsequentes se concentram nessas instâncias mais difíceis. Assim, o treinamento gera um modelo clusterizado mais complexo, ou *strong learners*, geralmente com preditores sequenciais corrigindo os predecessores (SARKER, 2021). A cada iteração, o algoritmo ajusta os pesos das amostras de treinamento, dando mais ênfase às instâncias que foram classificadas incorretamente (FREUND; SCHAPIRE, 1999).

Silva (2022) descreve o processo do AdaBoost, e explica que o funcionamento dele se divide em etapas as quais serão explicadas a seguir. A primeira delas é o treinamento. Nela, a entrada para o AdaBoost é dado por um conjunto de exemplos descrito na forma abaixo.

$$S = \{(x_1, y_1), (x_2, y_2), \dots, (x_m, y_m)\}, \quad i = 1, 2, \dots, m. \quad (2.1)$$

Onde, x_i se refere a um vetor de atributos dentro do sistema, ou seja, um conjunto

de informações relacionadas aos parâmetros ou características analisadas, que representam um determinado momento do sistema. O y_i representa o grupo de classificação associado a cada x_i dentre todos grupos de classificação possíveis dentro do sistema, e o número total das amostras do exemplo é definido por m .

O AdaBoost faz várias chamadas consecutivas ao classificador base, realizando isso em diversas etapas. A cada etapa, o algoritmo recebe uma distribuição específica de pesos associada a cada ponto de dados no conjunto de treinamento. Inicialmente, esses pesos são distribuídos de forma homogênea, garantindo que todos os exemplos tenham a mesma relevância no começo do processo. A distribuição inicial de pesos D_0 no conjunto de treinamento com m número de exemplos é descrita por:

$$D_0 = \frac{1}{m} \quad (2.2)$$

Durante cada iteração do ciclo de aprendizado, o algoritmo base cria uma hipótese h_t , levando em consideração os pesos atuais, de modo a dar prioridade à correta classificação dos dados que possuem maior peso relativo, com y_t representando o valor verdadeiro para o índice i . Assim, o propósito do algoritmo base é gerar uma hipótese que diminua o erro de ϵ_t , sendo:

$$\epsilon_t = \sum_{i:h_t(x_i) \neq y_t} D_t(i) \quad (2.3)$$

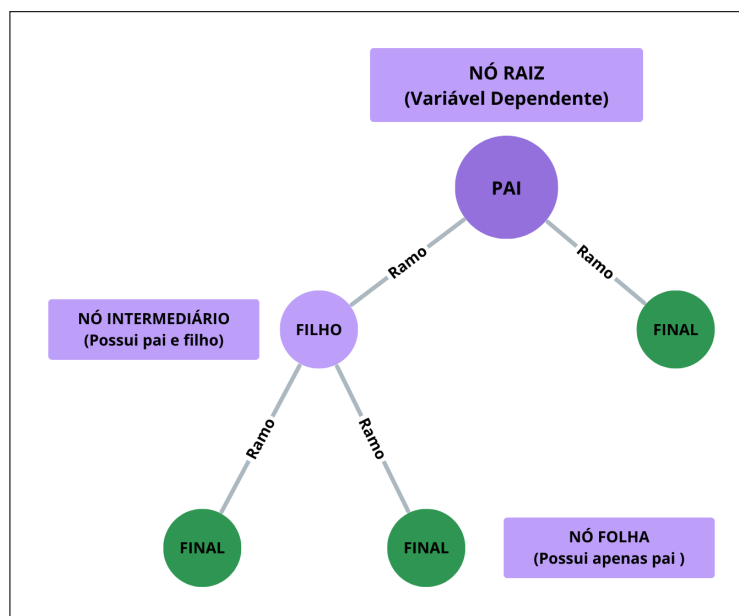
Após cada ciclo de aprendizado, os pesos são reajustados, aumentando o valor para os dados que foram classificados incorretamente. Em seguida, um novo ciclo tem início. Ao término de todas as T iterações predefinidas, o AdaBoost integra todas as hipóteses intermediárias h_t para gerar uma hipótese final única $H(X)$.

2.5.2 Árvores de Decisão

As árvores de decisão são modelos supervisionados de aprendizado de máquina que tem por objetivo dividir os dados de entrada de forma hierárquica, utilizando testes baseados em seus atributos. Essa divisão ocorre a partir de dois critérios mais comuns: impureza de Gini e ganho de informação, ambos com o objetivo de reduzir o número de divisões até a determinação de uma classe. O algoritmo de aprendizado seleciona, em cada etapa, o atributo que mais contribui para diminuir a incerteza quanto à classificação.

Na estrutura da árvore, as folhas representam os rótulos das classes, e os ramos refletem as combinações de atributos que levam a esses rótulos (PEIXOTO, 2023). Na Figura 2 abaixo é representada um exemplo de árvore de decisão.

Figura 2 – Representação de uma árvore de decisão



Fonte: Autores, 2024.

Sendo considerado uma técnica simples de aprendizado de máquina, o aprendizado por meio de árvores de decisão gera uma estrutura semelhante a uma árvore ao dividir repetidamente o conjunto de dados com base em um critério que maximize a separação dos dados. Essa técnica é aplicada inicialmente nas partes mais distantes da árvore e, em seguida, retorna ao início, conforme um método denominado retorno retrógrado. As árvores de decisão têm um papel importante no diagnóstico médico, sendo uma opção interessante dentro de problemas nesse contexto (ARAYESHGARI et al., 2023).

Peixoto (2023) descreve que a impureza de Gini pode ser determinada como um tipo de entropia de um conjunto de dados. Ela representa um número indicativo que descreve a probabilidade de novos dados aleatórios serem classificados erroneamente caso recebam um rótulo de classe aleatório de acordo com a distribuição de classe no conjunto de dados. Nesse contexto, um atributo com menor impureza é escolhido como divisor do nó. A impureza de Gini pode ser descrita matematicamente da forma:

$$gini(p) = 1 - \sum_{i=1}^n p_i^2 \quad (2.4)$$

Sendo p a distribuição de proporções (ou probabilidades) de diferentes elementos em um conjunto de dados, n o número de classes pertencentes ao o conjunto de dados e p_i a proporção de itens do elemento i na distribuição.

O ganho de informação avalia a mudança na impureza dos dados ao realizar a divisão de um nó na árvore de decisão. Em outras palavras, dado um atributo x no nó atual e um conjunto S , com $s_i \in S$, que representa os atributos potenciais dos nós filhos x , o cálculo do ganho de informação é feito comparando a impureza dos dados antes e depois da divisão, sendo descrito pela equação (PEIXOTO, 2023):

$$g(S, x) = gini(x) - \sum_{i=1}^S gini(s_i) \quad (2.5)$$

Moreira (2023) conclui que esse modelo vem sendo utilizado em várias áreas do conhecimento por conta de ser facilmente aplicável, livre de ambiguidade e robusto mesmo em casos de valores ausentes. Outra característica que fomenta esse ponto é a possibilidade de ser aplicado tanto com dados categóricos, quanto numéricos.

2.5.3 Regressão Logística

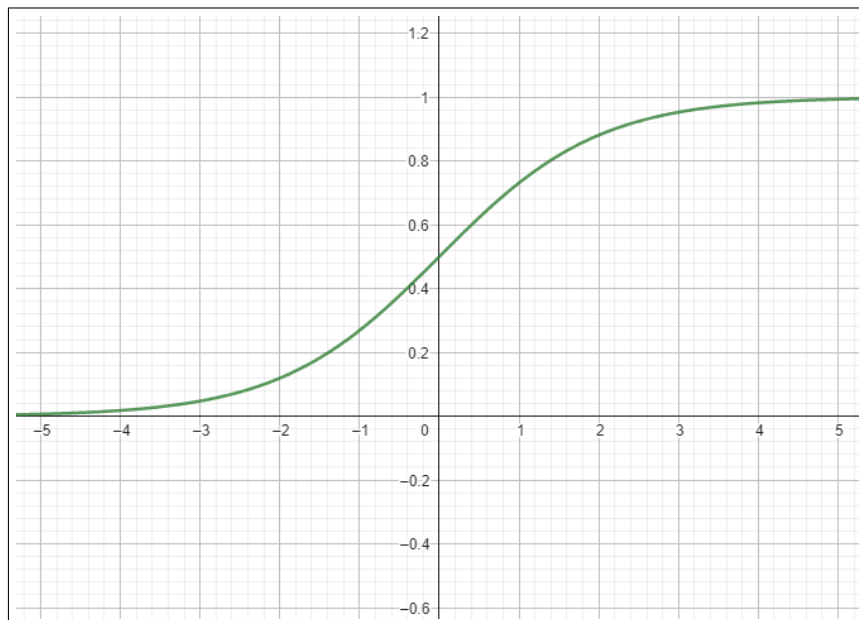
A regressão logística é uma técnica de aprendizagem supervisionada usada para problemas de classificação. Ela se baseia na transformação de uma regressão linear utilizando a função logística, também conhecida como função sigmoide, a qual é construída a partir da regressão linear, que mapeia qualquer valor real para o intervalo entre zero e um. Isso a torna ideal para lidar com problemas em que o objetivo é prever rótulos binários (REMIGIO, 2020).

Enquanto a regressão linear prevê valores em um intervalo infinito de números reais, a classificação requer a previsão de rótulos discretos, como zero ou um. Como o objetivo é prever problemas em que a saída só pode assumir valores de zero e um, a função sigmoide resolve esse problema, convertendo o valor real gerado pela combinação linear dos atributos em uma probabilidade entre 0 e 1, ou seja, a saída admitida estará dentro do intervalo $[0, 1]$, permitindo a rotulação de um dado (REMIGIO, 2020).

Freitas (2023) descreve que para cada preditor, um coeficiente cujo valor trata-se de uma medida para a relação entre a variável dependente e independente é aplicado. Dessa

maneira, a variável dependente receberá dois tipos de valores para indicar se determinado evento ocorrerá ou não. A Figura 3 abaixo representa uma função sigmoide.

Figura 3 – Função sigmoide



Fonte: Autores, 2024.

Matematicamente, a função sigmoide é descrita por:

$$p = \frac{1}{1 + e^{-y}} \quad (2.6)$$

Onde p descreve a probabilidade de uma instância qualquer pertencer à classe analisada: evento (geralmente é definido como positivo/verdadeiro, sendo atribuído o valor um). Já y é um número real dado pela combinação linear dos atributos utilizados na predição, derivado da regressão linear.

Em suma, a fórmula matemática da função sigmoide é utilizada para transformar a saída linear em uma probabilidade, e a classificação final é obtida a partir de um limiar de decisão (geralmente 0,5). Se a probabilidade for maior que o limiar, o registro é classificado como "1", caso contrário, como "0", representando a probabilidade do registro de entrada pertencer à classe evento (REMIGIO, 2020).

O modelo de regressão logística oferece vantagens significativas, como facilidade de implementação e interpretação, além de fornecer bons resultados quando os dados são linearmente separáveis. No entanto, em contextos com alta dimensionalidade, ele pode

sofrer de *overfitting*, o que pode ser mitigado com técnicas de regularização (REMIGIO, 2020). No caso da regressão logística, adicionar mais variáveis pode inicialmente melhorar o desempenho no conjunto de treinamento, mas, em certo ponto, isso começa a prejudicar o modelo, resultando em baixo desempenho quando testado com novos dados (SILVA JUNIOR, 2016).

2.5.4 XGBoost

XGBoost (*Extreme Gradient Boosting Trees*) é um algoritmo baseado em árvores de decisões e gradiente descendente. Isso significa que ele consiste em múltiplas árvores de decisão, onde cada uma delas aplica o aprendizado via gradiente descendente, refinando o que as árvores anteriores já aprenderam. No final, a classificação é determinada a partir da combinação das saídas de todas as árvores. Entre os principais diferenciais do XGBoost estão sua eficiente gestão de dados esparsos e a capacidade de executar de forma paralela ou distribuída (CHANG; CHANG; WU (2018) *apud* PEIXOTO, 2023).

A pontuação de uma amostra é prevista após a criação de k árvores durante o treinamento. Na prática, cada árvore terá um nó folha correspondente, determinado pelas características dessa amostra, e cada nó folha está relacionado a uma pontuação (BEKELE, 2022).

O maior benefício de se trabalhar com o XGBoost é a sua alta escalabilidade para tratar dados em diferentes cenários. A computação distribuída ou paralela presente no algoritmo possibilita o processo de aprendizado ocorrer de uma forma mais rápida do que os demais classificadores (CHEN; GUESTRIN, 2016).

De acordo com a documentação XGBoost Developers (*apud* (TRAN; LE; SHI, 2022)), a função objetivo do modelo pode ser expressa pela equação:

$$L = \sum_{i=1}^n l(y_i, \hat{y}_i) + \sum_{k=1}^K \Omega(f_k), \quad (2.7)$$

Sendo n o número de exemplos de treinamento, y_i e \hat{y}_i os rótulos reais e previstos, respectivamente; K representa o número de árvores do modelo, e f uma função no espaço funcional F que inclui todas as possíveis árvores de classificação e regressão. A primeira parte da equação representa a perda de treinamento, que reflete o quão precisa a predição do modelo é, enquanto a segunda representa o termo de regularização, que controla

a complexidade do modelo para evitar o *overfitting*, o qual ocorre quando um modelo apresenta bom desempenho no conjunto de dados de treinamento, mas tem performance insatisfatória em novos dados não utilizados durante o ajuste. Isso acontece, em geral, porque o modelo se torna excessivamente complexo, com muitos parâmetros, e passa a capturar particularidades do conjunto de treinamento, em vez de generalizar.

2.6 TREINAMENTO DOS MODELOS

Conforme descrito na Seção 4.5, o treinamento dos dados consiste na etapa em que os algoritmos de aprendizado de máquina são alimentados com um conjunto de dados, conhecidos como dados de treino, previamente processados. Cada um desses algoritmos aplica suas próprias técnicas para otimizar o ajuste do modelo. Por exemplo, o XGBoost utiliza árvores de decisão de forma iterativa e ajustada por gradiente descendente, enquanto o AdaBoost reavalia os pesos dos dados mal classificados a cada iteração para gerar hipóteses mais assertivas, enquanto a Regressão Logística cria uma função sigmoide para modelar a probabilidade de uma instância pertencer a uma classe específica. Durante o processo de treinamento, o objetivo comum de todos eles é minimizar os erros de classificação no conjunto de treino e gerar um modelo que seja capaz de generalizar corretamente em novos dados, evitando problemas como o *overfitting*.

Uma abordagem amplamente utilizada para aprimorar a performance de modelos de ML, especialmente em relação à sua capacidade de prever corretamente em dados fora do conjunto de treinamento, é a técnica de aumento de dados. Esse método é implementado como uma etapa opcional de pré-processamento, antes do início do treinamento do modelo. Consiste em expandir o conjunto de treinamento original, D_{train} , com novos pontos de dados gerados por meio de transformações que podem ser determinísticas ou aleatórias, aumentando assim a diversidade dos dados disponíveis para o modelo aprender.

Quando um modelo treinado apresenta um desempenho muito bom nas amostras utilizadas no treinamento, mas tem um desempenho ruim ao lidar com novas amostras, ou seja, apresentando *overfitting* e não conseguindo generalizar adequadamente, é necessário encontrar um conjunto ideal de parâmetros que equilibre bem esses dois aspectos. Isso se deve ser feito dividindo os dados em conjuntos de treinamento e validação, onde o conjunto de treinamento é utilizado para construir o modelo com diferentes configurações

de parâmetros, e cada modelo treinado é testado com o conjunto de validação, que por sua vez contém amostras conhecidas que não são utilizadas no treinamento dele, permitindo avaliar a precisão das previsões feitas [XU; GOODACRE \(2018\)](#) *apud* [OLIVEIRA, 2021](#).

2.7 AVALIAÇÃO DO DESEMPENHO DOS MODELOS

Uma modelagem que tem por objetivo a tomada de decisões deve ser acompanhada de métricas de avaliações desses modelos. Esses dados permitem entender o quanto um modelo acertou ou errou com relação à realidade a partir de valores numéricos obtidos da avaliação [\(SILVA, 2022\)](#).

[Silva \(2022\)](#) também explica que para entender a assertividade de uma predição, a matriz de confusão (ou matriz de erro) pode ser utilizada para descrever cada caso. A matriz de confusão é uma tabela que descreve os erros e acertos de um modelo conforme os valores reais. A seguir, a Tabela 1 descreve o funcionamento de uma matriz de confusão.

Tabela 1 – Matriz de Confusão

Real/Predito	Sim	Não
Sim	Verdadeiro Positivo (VP)	Falso Negativo (FN)
Não	Falso Positivo (FP)	Verdadeiro Negativo (VN)

Fonte: adaptada de [Silva \(2022\)](#).

- **Verdadeiro Positivo (VP):** ocorre quando a classe Positivo é corretamente classificada; ou seja, o modelo estimou Positivo e o valor real também pertence à classe Positivo.
- **Falso Positivo (FP):** trata-se de um erro em que o modelo previu a classe Positivo, mas o valor real era, na verdade, da classe Negativo.
- **Verdadeiro Negativo (VN):** acontece quando a classe Negativo é corretamente identificada; ou seja, o modelo estimou Negativo e o valor real corresponde à classe Negativo.
- **Falso Negativo (FN):** erro em que o modelo classificou como Negativo, mas o valor real pertencia à classe Positivo.

Para realizar essas validações, alguns classificadores são utilizados para observar as métricas de desempenho, avaliando a qualidade do modelo nas tarefas de classificação. Essas métricas fornecem uma visão do desempenho do classificador, auxiliando na comparação entre diferentes modelos (PAIXAO et al., 2022). Algumas dessas métricas são descritas por Silva (2022) e por Oliveira (2021), por exemplo: acurácia, precisão, *recall*, especificidade, *f1-score* e a Área Sob a Curva (AUC).

- **Acurácia:** conforme descrito anteriormente, a acurácia mede o quão correto o modelo esta de acordo com os dados reais, podendo ser descrito por:

$$AC = \frac{VP + VN}{VP + VN + FP + FN} \quad (2.8)$$

- **Precisão:** a precisão é uma métrica que avalia a proporção de padrões positivos corretamente previstos em relação ao total de padrões previstos como pertencentes à classe positiva.

$$P = \frac{VP}{VP + FP} \quad (2.9)$$

- **Recall:** o *recall* (ou sensibilidade) é calculado dividindo o número de exemplos corretamente classificados como pertencentes a uma determinada classe (Verdadeiros Positivos) pelo total de exemplos que realmente pertencem a essa classe.

$$R = \frac{VP}{VP + FN} \quad (2.10)$$

- **F1-Score:** é a média harmônica entre precisão e *recall*, permitindo avaliar ambas as métricas de forma combinada em uma única medida. Um valor baixo de F1-Score indica que uma ou ambas as métricas (precisão ou *recall*) provavelmente estão com desempenho inferior. A fórmula que descreve essa métrica é descrita por:

$$F1-Score = \frac{2 * P * RC}{P + RC} \quad (2.11)$$

- **Especificidade:** enquanto o *recall* mede a taxa de verdadeiro positivo, a especificidade mede a taxa de verdadeiro negativo.

$$E = \frac{VN}{VN + FP} \quad (2.12)$$

- **Área sob a curva (AUC):** conforme descrito por Oliveira (2021), a análise da curva ROC (*Receiver Operating Characteristic*) é uma técnica amplamente utilizada

para avaliar o desempenho de classificadores em conjuntos de dados desbalanceados. O gráfico ROC ilustra a relação entre a taxa de verdadeiros positivos e a taxa de falsos positivos. Com essa abordagem, é possível escolher classificadores com base no equilíbrio entre verdadeiros e falsos positivos. Para evitar a comparação visual das curvas, a área sob a curva (AUC - *Area Under the Curve*) sintetiza o desempenho de um modelo em um único valor, facilitando a comparação entre diferentes modelos de classificação. Um classificador aleatório apresenta uma AUC de 0.5, enquanto um classificador ideal tem AUC de 1.

- **Validação Cruzada:** de acordo com [Medeiros \(2023\)](#), a validação cruzada é uma técnica utilizada para avaliar o desempenho de um modelo de aprendizado de máquina de forma mais robusta, reduzindo a chance de que os resultados estejam enviesados para um subconjunto específico dos dados. O procedimento mais comum é o *k-fold cross-validation*, no qual o conjunto de dados é dividido em k subconjuntos (ou *folds*) aproximadamente do mesmo tamanho. O modelo é treinado k vezes, em cada iteração utilizando-se $k - 1$ partes para treino e a parte restante para teste. Ao final, é calculada a média das métricas obtidas em cada iteração, o que fornece uma estimativa mais estável do desempenho real do modelo. A fórmula genérica para o cálculo da validação cruzada pode ser representada por:

$$CV = \frac{1}{k} \sum_{i=1}^k M_i \quad (2.13)$$

onde M_i representa o valor da métrica de desempenho (como acurácia, precisão ou F1-Score) obtida na i -ésima iteração. Assim, a validação cruzada permite avaliar a capacidade de generalização do modelo, mitigando problemas de sobreajuste (*overfitting*) e subajuste (*underfitting*).

2.8 DEFINIÇÃO DAS VARIÁVEIS COM MAIS IMPACTO NA PRE-DIÇÃO DOS MODELOS

2.8.1 *Shapley Additive Explanations*

A interpretabilidade de modelos de aprendizado de máquina é um aspecto essencial quando tais modelos são utilizados em processos de tomada de decisão. Nesse contexto,

o método *Shapley Additive Explanations* (SHAP) foi desenvolvido com base na teoria dos jogos cooperativos, com o objetivo de atribuir de forma justa a contribuição de cada variável para o resultado do modelo (GOPINATH, 2021).

A ideia central do método está na utilização dos valores de *Shapley*, originalmente desenvolvidos na década de 1950, para distribuir ganhos em jogos cooperativos entre os participantes. Adaptado ao aprendizado de máquina, cada variável é tratada como um “jogador” que colabora para o valor final da predição, de modo que os valores de *Shapley* representem a contribuição marginal de cada atributo ao resultado (GOPINATH, 2021).

De forma geral, o valor de *Shapley* para uma variável i é definido como a média ponderada de suas contribuições marginais em todos os subconjuntos possíveis de variáveis. A fórmula matemática é dada por:

$$\phi_i(v) = \sum_{S \subseteq N \setminus \{i\}} \frac{|S|! (|N| - |S| - 1)!}{|N|!} [v(S \cup \{i\}) - v(S)] \quad (2.14)$$

Em que:

- N representa o conjunto total de variáveis (jogadores);
- S é um subconjunto de N que não contém i ;
- $v(S)$ indica o valor da função de característica (no contexto de ML, a predição do modelo) para o subconjunto S ;
- $\phi_i(v)$ é o valor de *Shapley* associado à variável i .

A interpretação dessa fórmula é que a contribuição de cada variável é calculada considerando todas as possíveis coalizões entre variáveis e, em seguida, ponderada de maneira a refletir sua importância relativa em diferentes combinações (GOPINATH, 2021).

Portanto, para interpretar os resultados e identificar as variáveis que mais contribuíram para a classificação, a técnica de SHAP calcula o valor de contribuição de cada variável preditora para a decisão do modelo, considerando todas as possíveis combinações entre variáveis. Dessa forma, cada atributo recebe um valor de SHAP que representa sua importância relativa no processo de predição. Além de facilitar a interpretação dos

modelos de aprendizado de máquina, o SHAP contribui para aumentar a transparência e a confiabilidade das decisões algorítmicas (DELPINO et al., 2025).

O SHAP é considerado um método agnóstico ao modelo, ou seja, pode ser aplicado independentemente do algoritmo utilizado. Ele permite tanto explicações locais, que descrevem a contribuição das variáveis em uma predição individual, quanto explicações globais, que descrevem a importância das variáveis para o comportamento geral do modelo (PAVANYA et al., 2025).

A abordagem é calculada através da perturbação dos valores de entrada e da observação de como as alterações afetam a predição. Frequentemente, utiliza-se um grupo de comparação (ou *baseline*), que serve como referência para definir contra qual cenário a decisão está sendo explicada. Dessa forma, o SHAP fornece uma estrutura teórica para avaliar a relevância das variáveis em modelos de aprendizado de máquina, permitindo identificar quais atributos exercem maior impacto sobre a predição (GOPINATH, 2021).

3 TRABALHOS RELACIONADOS

Esse capítulo aborda a busca e seleção de trabalhos relevantes para o tema versado neste estudo. Essa seleção é pertinente para fornecer uma base teórica sólida e situar a pesquisa dentro do contexto atual de estudos relacionados à predição de BPN utilizando técnicas de ML. Serão destacados, a seguir, os estudos que ofereceram as contribuições mais relevantes para esta área de pesquisa e que têm uma relação direta com o desenvolvimento deste trabalho.

3.1 TRABALHOS SELECIONADOS

O estudo de [Borba \(2023\)](#) discute a generalização de algoritmos de ML em saúde e destaca a importância de validar esses modelos em diferentes cenários para garantir sua eficácia, especialmente em contextos de saúde pública. Um ponto relevante do trabalho é a utilização de modelos preditivos para prever mortalidade neonatal a partir de dados tabulares de saúde coletados em São Paulo e em outras localidades do Brasil, incluindo algoritmos como XGBoost, CatBoost e LightGBM, além de uma rede neural do tipo *Multilayer Perceptron* (MLP). Entre esses, o XGBoost apresentou o melhor desempenho geral. Além disso, o estudo explora a técnica de *transfer learning* (aprendizado por transferência), mostrando que, embora promissora, essa abordagem não superou os métodos tradicionais de *boosting* em alguns casos. Esses aspectos destacam o valor da generalização e da adaptação de modelos em diferentes contextos geográficos e temporais, algo fundamental em saúde pública para melhorar a qualidade das decisões clínicas e de gestão. A análise de importância de variáveis destacou fatores clínicos e sociodemográficos como relevantes para a predição da mortalidade neonatal, embora sem detalhar especificamente a variável mais determinante.

O trabalho de [Moreira \(2023\)](#) discute a aplicação de técnicas de ML para a classificação de risco neonatal, utilizando as bases de dados de saúde pública no Brasil, como o SINASC e o Sistema de Informação sobre Mortalidade (SIM). O autor integra dados de diferentes sistemas e aplica algoritmos de ML, como AdaBoost e XGBoost, para prever o risco de mortalidade neonatal, com o objetivo de gerar classificadores que auxiliem na

tomada de decisão de gestores de saúde e profissionais na prevenção de óbitos neonatais. A relevância do trabalho está em seu foco na saúde pública, na exploração de grandes bases de dados e no uso de técnicas de mineração de dados para oferecer *insights* que podem ajudar a melhorar os cuidados com a saúde neonatal. Entre os classificadores avaliados, o AdaBoost foi destacado como o mais adequado por aliar bom desempenho com menor tempo de execução. A análise de importância de variáveis, realizada por meio do algoritmo SHAP, apontou como fatores mais influentes os índices de Apgar de um e cinco minutos, o peso ao nascer, a prematuridade, a idade gestacional, a presença de anomalias congênitas, a raça materna e o número de filhos vivos.

A pesquisa realizada por [Neri et al. \(2023\)](#) aplicou e comparou diferentes algoritmos de ML para a predição de BPN, um importante indicador de saúde neonatal associado a maior mortalidade infantil e complicações na vida adulta. Considerando a relevância do BPN como um fator de risco para doenças crônicas e comprometimento no desenvolvimento, os autores utilizaram modelos como AdaBoost e XGBoost em dados do SINASC para avaliar a precisão na predição desses casos. A comparação foi conduzida com o objetivo de identificar o melhor desempenho em termos de precisão e especificidade, utilizando técnicas de pré-processamento e balanceamento de classes para otimizar os resultados dos modelos. Entre os algoritmos avaliados, o AdaBoost apresentou o melhor desempenho global, enquanto o XGBoost se destacou pela maior sensibilidade na identificação dos casos de BPN.

O estudo de [Peixoto \(2023\)](#) aborda diferentes estratégias para lidar com o desbalanceamento de classes, um problema recorrente em dados de saúde, especialmente em eventos raros como a mortalidade neonatal. Essa questão é de grande relevância para este trabalho, que também enfrenta o desafio do desbalanceamento ao prever o BPN. O estudo aplica técnicas como SMOTE-ENN e SMOTE-Tomek, além de algoritmos como XGBoost e AdaBoost, que são eficazes na correção do desbalanceamento de classes, melhorando a performance dos modelos preditivos. A sensibilidade é destacada como uma métrica crítica, já que em contextos de saúde é essencial minimizar falsos negativos. Entre os modelos avaliados, o XGBoost combinado ao método SMOTE-ENN apresentou o melhor desempenho. A análise de importância de variáveis, realizada por meio do algoritmo SHAP, apontou como mais relevantes a quantidade de filhos nascidos mortos da mãe, a escala de Apgar no quinto minuto, o mês de início do pré-natal, a quantidade de partos normais,

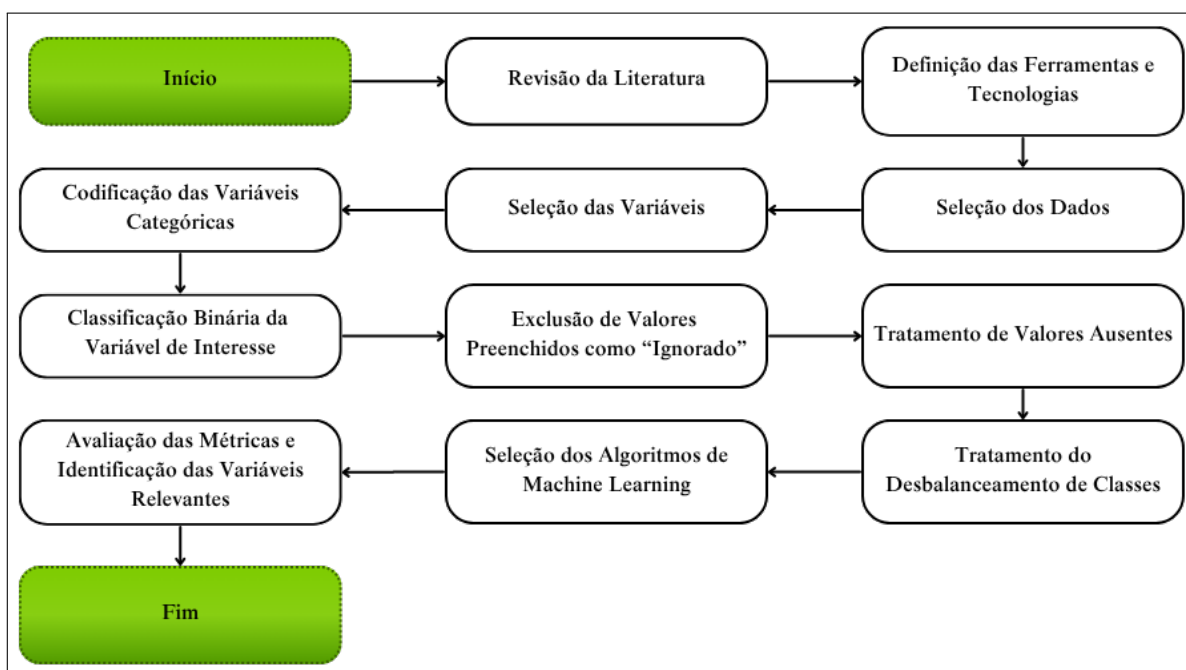
a indução do trabalho de parto e o tipo de apresentação do recém-nascido. A utilização de algoritmos baseados em árvores também foi eficiente, demonstrando a importância de escolher métodos robustos para tratar variáveis categóricas e classes desbalanceadas, o que é diretamente aplicável ao cenário de predição de BPN.

O estudo conduzido por [Victor et al. \(2025\)](#) analisou o desempenho de diferentes algoritmos de ML na predição de baixo peso ao nascer, utilizando dados de uma coorte de gestantes em Araraquara (SP). Os autores compararam modelos tradicionais, como a regressão logística, com algoritmos de *boosting*, incluindo o XGBoost e o CatBoost, identificando que estes últimos apresentaram resultados superiores em termos de AUROC, atingindo valores próximos de 0,94. Entre os modelos avaliados, o XGBoost obteve o melhor desempenho, alcançando AUROC de 0,941. A análise de importância de variáveis, realizada por meio do algoritmo SHAP, apontou a idade gestacional como o fator mais determinante, seguida do estado civil materno e da frequência de consultas pré-natais, além de variáveis como atividade física, raça materna e paridade. O trabalho destaca que métodos baseados em *boosting* são particularmente eficazes para bases tabulares, por sua capacidade de capturar relações complexas e não lineares entre os preditores. Esses achados reforçam a tendência de que algoritmos mais robustos e modernos superam abordagens convencionais, fomentando a utilização de algoritmos de *boosting* também neste estudo.

4 METODOLOGIA

A metodologia deste trabalho será estruturada de forma a explorar a predição de baixo peso ao nascer (BPN) por meio de técnicas de *Machine Learning* (ML). Nesta abordagem, será realizada a coleta e exploração dos dados disponíveis, seguida pela seleção e aplicação de diversos algoritmos de aprendizado de máquina. A análise focará em identificar as variáveis que influenciam o BPN, além de comparar o desempenho dos modelos implementados. O fluxograma presente na Figura 4 descreve os passos metodológicos envolvidos nos processos que serão desenvolvidos ao decorrer desse trabalho.

Figura 4 – Fluxograma



Fonte: Autores, 2025.

4.1 REVISÃO DA LITERATURA

Foi realizada uma revisão da literatura que objetivou a expansão do conhecimento sobre os tópicos necessários para o desenvolvimento da pesquisa, com ênfase em trabalhos que interligam os temas de ML, agrupamento e classificação de dados, baixo peso neonatal, e modelos baseados em árvores de decisão.

4.2 DEFINIÇÃO DAS FERRAMENTAS E TECNOLOGIAS

Nesta etapa foram selecionadas as ferramentas que estruturaram o desenvolvimento do presente estudo. Essa seleção foi realizada baseando-se na literatura relevante e nos desafios que surgiram a medida que o trabalho foi produzido.

4.2.1 Seleção da Base de Dados

Com o entendimento do processamento e preenchimento das fontes dos dados, foram selecionadas as bases do SINASC fornecidas pelo DATASUS, referentes ao período de 2012 a 2023. O período a partir de 2012 ofereceu registros mais consistentes para a predição do baixo peso ao nascer, assegurando a presença de variáveis relevantes como peso ao nascer, idade gestacional e fatores maternos, evitando limitações observadas em anos anteriores (MOREIRA, 2023).

Com as bases devidamente selecionadas, foi realizado a combinação em uma base unificada, que armazena todos os dados dos anos selecionados. Essa união resultou, ao todo, em 33.887.605 registros distribuídos em 68 colunas.

4.2.2 Ambiente de Desenvolvimento, Linguagem de Programação e Bibliotecas

Na definição das ferramentas para a predição, optou-se pelo ambiente de computação em nuvem Google Colab, ou *Colaboratory*, devido à sua simplicidade de uso, facilidade de colaboração em tempo real e acesso gratuito a altas capacidades de memória e Unidade de Processamento Tensorial (TPU), indispensáveis para o manuseio de grandes volumes de dados e o desenvolvimento de modelos de ML.

Dada a dimensão da base unificada e a alta demanda computacional para as etapas de pré-processamento foi necessário recorrer à versão paga do ambiente, o *Colab Pro*. Essa escolha permitiu o uso de recursos ampliados de memória e processamento, garantindo estabilidade durante a execução de operações intensivas e evitando interrupções frequentes que ocorrem na versão gratuita do *Colab*. Assim, o ambiente do *Colab* foi configurado para o tratamento dos dados, utilizando uma TPU de *back-end* do *Google Compute Engine* em *Python 3*. Com uma capacidade de 334,6 GB de RAM e 225,3 GB de armazenamento em disco, o sistema possui os recursos necessários para manipular os grandes volumes de

dados presentes na tabela.

Foi feita a escolha do *Python* como a linguagem de programação, uma ferramenta *open source* que têm uma comunidade grande e ativa de programadores e cientistas, o que implica na abundância de estudos na área e na criação de bibliotecas e pacotes relevantes na área de ML (FILHO, 2015). Além disso, foram considerados a relevância da linguagem na literatura relacionada e o fato do ambiente da *Google* ser estruturado em torno dessa linguagem.

Ao longo do desenvolvimento do trabalho, foram utilizadas diversas bibliotecas da linguagem Python, cada uma desempenhando um papel específico no processo de análise, modelagem e visualização dos dados. A seleção dessas bibliotecas ocorreu de forma iterativa, à medida que a complexidade das tarefas se apresentava, permitindo um melhor ajuste às exigências do projeto. Essas ferramentas foram fundamentais para garantir eficiência, reprodutibilidade e clareza nos resultados obtidos. A seguir, destacam-se as bibliotecas empregadas:

- **pandas** e **NumPy**: utilizadas para manipulação, limpeza e transformação dos dados, permitindo a criação e alteração de tabelas, assim como cálculos e operações matriciais.
- **math**: utilizada para operações matemáticas auxiliares e mais simples.
- **Matplotlib** e **Seaborn**: aplicadas na geração de gráficos e visualizações estatísticas, possibilitando uma melhor análise visual dos resultados.
- **Scikit-Learn**: principal biblioteca de aprendizado de máquina utilizada para este estudo. Utilizada para construção e avaliação dos modelos preditivos, incluindo *Decision Tree*, *Logistic Regression*, *Random Forest* e *HistGradientBoosting*. Também foram utilizadas ferramentas de pré-processamento, como *LabelEncoder*, *OneHotEncoder*, *StandardScaler*, *SimpleImputer*, além de recursos de validação como o *train_test_split*, *StratifiedKFold*, *cross_val_score* e *cross_validate*.
- **XGBoost**: biblioteca especializada para a construção de modelos de gradiente otimizado, responsável pela aplicação do algoritmo de XGBoost e suas dependências.

- **SHAP**: biblioteca aplicada para interpretação dos modelos através do *Shapley Value*, fornecendo explicações sobre a contribuição de cada variável na previsão.
- **Métricas do Scikit-Learn**: como *accuracy*, *precision*, *recall*, *f1-score*, *AUC*, *confusion matrix*, *classification report* e *mean squared error*, utilizadas para a avaliação quantitativa dos modelos.

4.3 SELEÇÃO DOS DADOS

Nesse processo, os dados consolidados em uma base única foram filtrados apenas para as localizações de Campos dos Goytacazes (RJ), Mossoró (RN), Volta Redonda (RJ), Uberaba (MG) e Imperatriz (MA) a partir da informação da variável CODMUNNASC, que corresponde ao código do município de nascimento da criança definido pelo Instituto Brasileiro de Geografia e Estatística (IBGE). Essa escolha foi baseada na semelhança de IDH e população com Campos dos Goytacazes, como visto na Tabela 2 (ATLASBR, 2010).

Tabela 2 – IDH e população das cidades de acordo com o último Censo disponível

Cidade (Estado)	IDH	População
Campos dos Goytacazes (RJ)	0,716	483.540
Mossoró (RN)	0,720	264.577
Imperatriz (MA)	0,731	273.110
Volta Redonda (RJ)	0,770	261.563
Uberaba (MG)	0,772	337.836

Fonte: Instituto Brasileiro de Geografia e Estatística (IBGE) (2022).

A variável CODMUNNASC foi utilizada apenas para a seleção das cidades e para a divisão do conjunto em treino e teste, não sendo considerada como preditora nos modelos desenvolvidos. Essa seleção resultou na distribuição de dados observada na Tabela 3.

Tabela 3 – Quantidade de Nascidos Vivos por Cidade Seleccionada

Cidade (Estado)	CODMUNNASC	Qtd. Nascidos Vivos
Campos dos Goytacazes (RJ)	330100	99.386
Mossoró (RN)	240800	84.417
Imperatriz (MA)	210530	108.001
Volta Redonda (RJ)	330630	46.497
Uberaba (MG)	317010	61.587

Fonte: Autores, 2025.

4.4 SELEÇÃO DAS VARIÁVEIS

A seleção das variáveis utilizadas no modelo considerou as evidências disponíveis na literatura sobre os principais determinantes do baixo peso ao nascer (BPN), mortalidade e da prematuridade no Brasil. Estudos baseados no SINASC indicam que características maternas, da gestação, do parto e do recém-nascido estão fortemente associadas a esses desfechos, justificando sua inclusão no presente trabalho (HENRIQUES et al., 2019; PEDRAZA, 2014; MAIA; SOUZA; MENDES, 2020). Em especial, Pedraza (2014) apresenta uma análise composta por 23 artigos que identificam um conjunto recorrente de variáveis relacionadas ao BPN, como por exemplo, idade materna, número de consultas pré-natais, escolaridade da mãe, sexo da criança e duração da gestação, entre outras. Henriques et al. (2019) destacam a relevância da determinação da idade gestacional para a acurácia das análises. Maia, Souza e Mendes (2020) reforçam, em estudos baseados em cinco cidades brasileiras, o papel de determinantes socioeconômicos, maternos e das condições do parto na mortalidade infantil.

Dessa forma, a escolha das variáveis neste trabalho buscou contemplar os determinantes mais recorrentes e consistentes apontados pela literatura, equilibrando aspectos sociais, maternos, obstétricos e neonatais. Esta abordagem garante maior direcionamento ao modelo proposto, possibilitando o descarte controlado das variáveis restantes, além de permitir o entendimento deste estudo com os resultados dos artigos citados.

Portanto, foram selecionados os dados relevantes para o estudo a partir da revisão feita na literatura de referência, mantendo-se apenas as colunas e registros que contribuem diretamente para a predição.

Esse processo de escolha, a partir das cinco localidades, resultou em um total de 399.888 registros, distribuídos entre 17 colunas, detalhadas a seguir.

4.4.1 Variáveis Socioeconômicas e Demográficas

- **RACACORMAE**: cor ou raça da mãe.
- **ESMAEAGR1**: escolaridade da mãe em categorias agregadas derivadas do Censo de 2010.
- **LOCNASC**: local de nascimento (hospital, domicílio, etc.).

- **ESTCIVMAE**: estado civil da mãe (solteira, casada, viuva, etc).

4.4.2 Variáveis sobre a Mãe

- **IDADEMAE**: idade da mãe no momento do parto.
- **QTDFILMORT**: quantidade de filhos mortos.
- **QTDFILVIVO**: quantidade de filhos vivos.
- **PARIDADE**: quantidade de partos que a gestante teve com ≥ 20 semanas de gestação (idade gestacional viável).

4.4.3 Variáveis sobre a Gestação e o Parto

- **GRAVIDEZ**: tipo de gravidez (única, gemelar, etc.).
- **PARTO**: tipo de parto (normal, cesárea, etc.).
- **SEMAGESTAC**: duração da gestação em semanas.
- **CONSPRENAT**: número de consultas pré-natais.
- **TPROBSON**: grupo de Robson.
- **APGAR1**: índice de Apgar medido no 1^o minuto de vida.
- **APGAR5**: índice de Apgar medido no 5^o minuto de vida.

4.4.4 Variáveis sobre o Bebê Nascido

- **SEXO**: sexo do bebê (masculino ou feminino).
- **PESO**: peso do bebê ao nascer, em gramas.

Por não serem de compreensão imediata, algumas variáveis apresentadas acima necessitam de contexto adicional, que serão detalhados a seguir.

O Grupo de Robson é uma classificação criada em 2001 por Michael Robson, que agrupa as gestantes em dez categorias distintas, considerando características como a paridade, idade gestacional, apresentação fetal e histórico de cesárea anterior, diferentemente

de classificações baseadas apenas na indicação da cesariana (ROBSON; HARTIGAN; MURPHY, 2013)).

Quanto ao índice de Apgar, Leitão et al. (2023) o descreve como uma escala clínica utilizada para avaliar rapidamente a condição do recém-nascido logo após o parto e orientar eventuais intervenções necessárias na recepção do bebê. São analisados cinco sinais vitais: cor da pele, frequência cardíaca, irritabilidade reflexa, tônus muscular e esforço respiratório, resultando em uma pontuação de 0 a 10 pontos. Valores entre 7 e 10 indicam boas condições gerais de vitalidade, pontuações de 4 a 6 sugerem dificuldade leve a moderada associada à asfixia, enquanto escores entre 0 e 3 correspondem a quadros de asfixia grave, demandando cuidados imediatos.

Optou-se pela utilização da variável ESCMAEAGR1 em relação às outras variáveis que também representam a escolaridade materna, como ESCMAE2010 (escolaridade da mãe conforme padrão de categorias do Censo de 2010) e ESCMAE (escolaridade da mãe em anos de estudo concluído). A escolha se justifica por essa variável apresentar uma categorização mais detalhada, permitindo distinguir os níveis de ensino de forma mais clara. Além disso, as categorias presentes em ESCMAEAGR1 permitem reduzir possíveis ambiguidades presentes em outras variáveis, oferecendo uma classificação mais padronizada, o que facilita a análise e interpretação dos resultados.

4.5 CODIFICAÇÃO DAS VARIÁVEIS CATEGÓRICAS

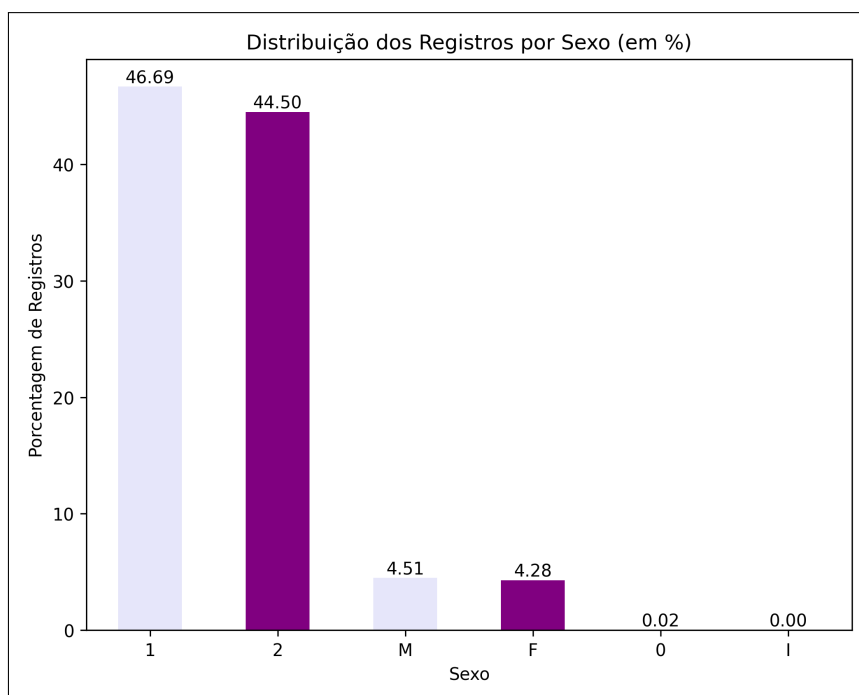
Segundo Potdar, Pardawala e Pai (2017), algoritmos de ML aceitam como entrada apenas variáveis numéricas. Desse modo, é preciso converter variáveis categóricas, como por exemplo raça (amarelo, branco, negro) e gênero (feminino, masculino), em valores numéricos para que suas categorias sejam devidamente representadas no modelo.

Entre as técnicas disponíveis para essa conversão, destaca-se o *Label Encoding*, que, conforme apontam Kumar e Bhardwaj (2025), consiste em atribuir um número inteiro a cada categoria distinta. Essa abordagem garante que os rótulos textuais ou simbólicos passem a ser representados de forma numérica, permitindo que sejam processados pelos algoritmos.

No conjunto de dados utilizado neste trabalho, verificou-se a existência de diferentes

formas de rotulação em uma mesma variável. Em específico a variável *SEXO* apresentava múltiplos valores para uma mesma categoria: 1 e M para masculino, 2 e F para feminino, e 0 e I para ignorado. Para uniformizar os registros, aplicou-se o *Label Encoding*, codificando I, M e F em valores numéricos equivalentes a 0, 1 e 2, respectivamente. A Figura 5 ilustra a distribuição antes da codificação.

Figura 5 – Distribuição dos Registros por Sexo



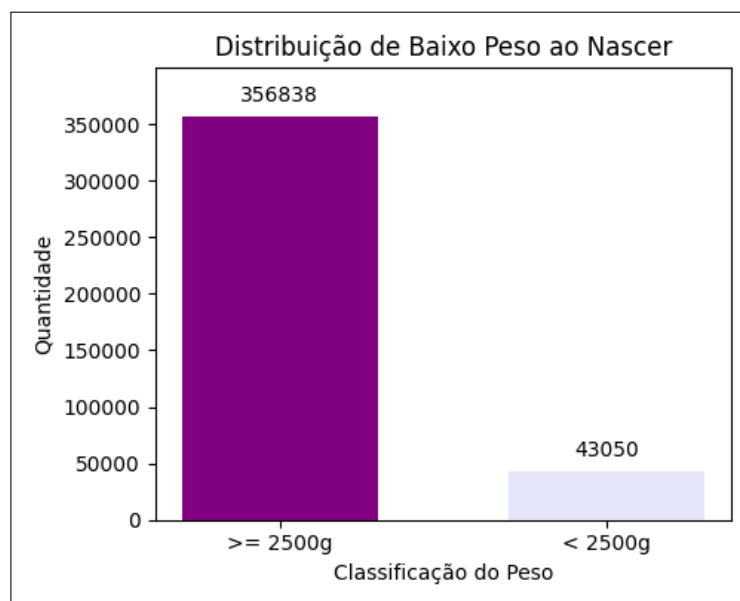
Fonte: Autores, 2025.

4.6 CLASSIFICAÇÃO BINÁRIA DA VARIÁVEL DE INTERESSE

A classificação binária se refere à simplificação do problema de classificação ao pertencimento a um de dois conjuntos distintos. No presente trabalho, o problema do BPN foi rotulado em duas categorias: baixo peso e peso normal (OLIVEIRA, 2023).

Foi criada uma nova coluna na base de dados que classifica os registros de recém-nascidos com peso inferior a 2.500g como 1 (baixo peso) e os demais como 0 (peso normal) (ARAYESHGARI et al., 2023). Essa coluna foi denominada PESO_BINARIO e passou a representar a variável de interesse para a predição. A Figura 6 apresenta a distribuição resultante desse processo de binarização.

Figura 6 – Distribuição de Recém-nascidos por Peso (Normal vs Baixo Peso)



Fonte: Autores, 2024.

4.7 EXCLUSÃO DE VALORES PREENCHIDOS COMO "IGNORADO"

As variáveis RACACORMAE, ESCMAEAGR1, GRAVIDEZ, PARTO, SEXO e LOCNASC apresentavam a categoria "Ignorado/Não Informado". Embora esses campos estivessem preenchidos, esse tipo de categoria não representa uma informação real sobre a variável e, portanto, não possui relevância preditiva. A Tabela 4 mostra a quantidade de ocorrências identificadas em cada coluna.

Tabela 4 – Quantidade de Valores Ignorados nas Variáveis Seleccionadas

Variável	Qtd. Valores Ignorados
RACACORMAE	64
ESMAEAGR1	6118
GRAVIDEZ	4
PARTO	4
SEXO	53
LOCNASC	10

Fonte: Autores, 2025.

Assim, optou-se pela remoção dos registros em que esse valor ocorria, garantindo que apenas dados efetivamente informativos fossem considerados nos modelos. Essa etapa

resultou na exclusão de 6.238 registros, correspondendo a aproximadamente 1,5% do total inicial. Após o tratamento, obteve-se um conjunto final de 393.650 registros, distribuídos em 17 variáveis, conforme detalhado na Tabela 5 e Tabela 6.

Tabela 5 – Dicionário das Variáveis Seleccionadas (Parte 1)

Variável	Descrição	Valores
RACACORMAE	Cor ou raça da mãe	1: Branca 2: Preta 3: Amarela 4: Parda 5: Indígena
ESMAEAGR1	Escolaridade 2010 agregada	0: Sem Escolaridade 1: Fundamental I Incompleto 2: Fundamental I Completo 3: Fundamental II Incompleto 4: Fundamental II Completo 5: Ensino Médio Incompleto 6: Ensino Médio Completo 7: Superior Incompleto 8: Superior Completo 10: Fundamental I Inespecífico 11: Fundamental II Inespecífico 12: Ensino Médio Inespecífico
LOCNASC	Local de nascimento	1: Hospital 2: Domicílio 3: Outro
ESTCIVMAE	Estado civil da mãe	1: Solteira 2: Casada 3: Viúva 4: Separada judicialmente 5: União consensual
IDADEMAE	Idade da mãe	Número absoluto
QTDFILMORT	Filhos mortos	Número absoluto
QTDFILVIVO	Filhos vivos	Número absoluto
PARIDADE	Quantidade de partos ≥ 20 semanas de gestação	Número absoluto
GRAVIDEZ	Tipo de gravidez	1: Única 2: Dupla 3: Tripla ou mais
PARTO	Tipo de parto	1: Vaginal 2: Cesáreo
SEMAGESTAC	Duração da gestação	Número de semanas

Fonte: Autores, 2025.

Tabela 6 – Dicionário das Variáveis Seleccionadas (Parte 2)

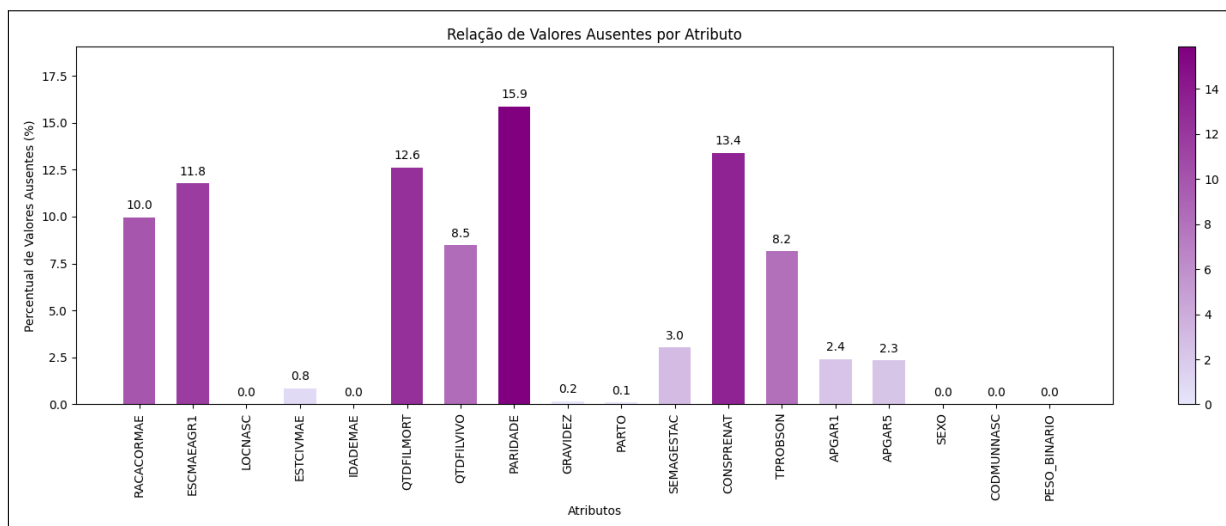
Variável	Descrição	Valores
CONSPRENAT	Número de consultas pré-natais	Número absoluto
TPROBSON	Grupo de Robson	1 a 10
APGAR1	Índice de Apgar no 1º minuto	0–10
APGAR5	Índice de Apgar no 5º minuto	0–10
SEXO	Sexo do bebê	1: Masculino 2: Feminino
PESO_BINARIO	Peso ao nascer classificado	0: Peso normal 1: Baixo peso

Fonte: Autores, 2025.

4.8 TRATAMENTO DE VALORES AUSENTES

Posteriormente, foi realizada uma nova etapa de tratamento dos dados na base unificada, com o objetivo de remover elementos nulos, mal formatados ou inconsistentes, que poderiam comprometer a precisão e a confiabilidade das previsões do modelo. Na Figura 7 é possível observar a distribuição dos valores ausentes por atributos na base.

Figura 7 – Relação de Valores Ausentes por Atributo



Fonte: Autores, 2025.

A remoção de todas as linhas com dados ausentes ou anômalos resultaria em uma redução drástica de aproximadamente 33% dos registros na base, passando de 393.650

para 263.158 casos. Para evitar essa perda significativa de dados, será adotada uma estratégia mais refinada, composta pelos seguintes passos:

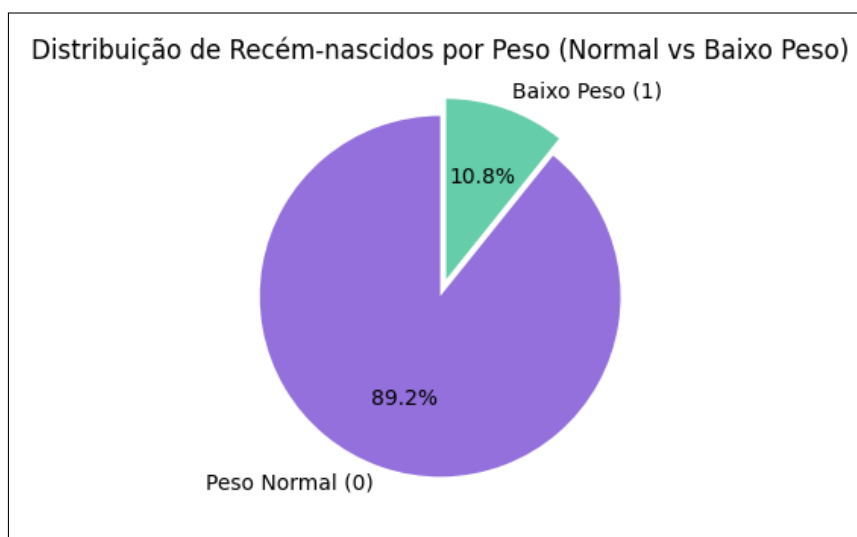
1. Remoção das linhas com ausência de valores em mais de duas colunas da classe majoritária (peso normal).
2. Preenchimento dos campos ausentes remanescentes do passo anterior utilizando o valor da mediana, calculada a partir dos valores preenchidos em cada campo específico. Esse procedimento foi aplicado em ambas as classes.

Ao término das etapas de tratamento descritas, obteve-se um total de 333.388 registros.

4.9 TRATAMENTO DO DESBALANCEAMENTO DE CLASSES

Na análise da distribuição da variável de interesse PESO_BINARIO, apresentada na Figura 6, observou-se uma grande discrepância entre as classes, onde a classe de interesse “baixo peso” representou cerca de apenas 10,8% das amostras totais preenchidas, o que caracteriza o desbalanceamento das classes, como é possível observar na Figura 8

Figura 8 – Gráfico de Setores da Distribuição de Peso



Fonte: Autores, 2025.

Esse problema foi tratado com a técnica de subamostragem aleatória, na qual parte dos registros da classe majoritária é removida de forma controlada. Para isso,

utilizou-se como referência a quantidade de casos de baixo peso (42.511), e selecionou-se aleatoriamente o mesmo número de registros da classe de peso normal. Dessa forma, as duas classes passaram a ter quantidades equivalentes de amostras, garantindo equilíbrio na base de treinamento e contribuindo para melhorar a performance do modelo.

Inicialmente, foi avaliado o uso do método SMOTE para realização dessa etapa. Entretanto, os resultados obtidos não foram satisfatórios, possivelmente porque a criação de muitos registros artificiais introduziu padrões que não representavam fielmente a realidade, o que pode ter levado à redução da capacidade dos modelos. Além disso, outro aspecto a considerar é que o pré-processamento realizado nesta etapa ainda não estava totalmente consolidado, o que pode ter potencializado essas perdas ao introduzir inconsistências que se refletiram na geração desses novos dados.

Nesse sentido, optou-se pelo uso da subamostragem, por trabalhar exclusivamente com registros reais. Apesar da perda de parte dos dados da classe majoritária, optou-se por esse método para preservar a consistência dos dados e produzir métricas de desempenho mais equilibradas. Após o tratamento, obteve-se um conjunto final de 85.022 registros.

4.10 SELEÇÃO DOS ALGORITMOS DE *MACHINE LEARNING*

A seleção dos algoritmos foi pautada na literatura relacionada, considerando a relevância e o bom desempenho apresentado nos resultados dos trabalhos consultados. [Arayeshgari et al. \(2023\)](#) apontam alguns algoritmos de ML para a predição do baixo peso ao nascer, levando em consideração suas características, capacidade de generalização e desempenho comprovado em problemas de classificação. Esses algoritmos foram escolhidos com base em sua eficiência em lidar com dados desbalanceados, flexibilidade em ajustar hiperparâmetros e robustez ao enfrentar cenários com variabilidade dos dados, dessa forma, direcionando as escolhas dos algoritmos, sendo eles:

- **AdaBoost:** algoritmo de *boosting* que combinará múltiplos classificadores fracos para formar um classificador forte, aumentando iterativamente a precisão. A cada iteração, o algoritmo ajusta os pesos dos exemplos incorretamente classificados, concentrando-se mais nesses exemplos para melhorar o desempenho. Ele é eficaz em reduzir o viés e costuma ser utilizado com árvores de decisão como classificadores

fracos.

- **Árvores de Decisão:** algoritmo de aprendizado supervisionado que dividirá os dados em subconjuntos com base em variáveis explicativas, formando uma estrutura hierárquica que facilitará a interpretação dos resultados.
- **Regressão Logística:** algoritmo clássico de classificação binária que estimará a probabilidade de um evento com base em uma função logística, sendo eficiente em problemas de classificação simples.
- **XGBoost:** algoritmo de *boosting* baseado em árvores de decisão, que otimiza a técnica de *gradient boosting*, utilizando métodos como regularização e paralelização para melhorar o desempenho e reduzir o risco de *overfitting*.

A escolha desses algoritmos visará a exploração de diferentes abordagens para maximizar a precisão na predição de baixo peso ao nascer, utilizando tanto técnicas de *boosting* quanto de modelos lineares e baseados em árvores.

4.11 AVALIAÇÃO DAS MÉTRICAS E IDENTIFICAÇÃO DAS VARIÁVEIS RELEVANTES

A análise do desempenho dos modelos foi realizada a partir de métricas de avaliação de classificação binária: acurácia, precisão, *recall*, *F1-Score*, especificidade e área sob a curva ROC (AUC). Além disso, foi aplicada a técnica de validação cruzada, fornecendo estimativas mais robustas sobre a capacidade de generalização dos algoritmos. A análise conjunta dessas métricas permitiu comparar o desempenho dos diferentes modelos, permitindo a identificação de cada métrica individualmente para os mesmos.

Complementarmente, foi realizada a identificação das variáveis mais relevantes para a predição do baixo peso ao nascer por meio do algoritmo SHAP. Essa técnica quantifica a contribuição de cada variável no resultado do modelo, atribuindo valores médios absolutos de importância. Dessa forma, é possível interpretar o impacto individual de cada atributo na classificação, fomentando o entendimento das conclusões obtidas.

5 DESENVOLVIMENTO

5.1 AMBIENTE DE DESENVOLVIMENTO E PRÉ-CONFIGURAÇÕES

No ambiente Google Colab criou-se um *notebook* para a execução dos passos necessários para o desenvolvimento do projeto.

5.2 IMPORTAÇÃO DAS BIBLIOTECAS

No Código [5.1](#), nas linhas 1 a 6, foram importadas bibliotecas de manipulação e visualização de dados, como *pandas*, *numpy*, *seaborn*, *matplotlib* e *shap*. A linha 7 trouxe o módulo *tree* do *scikit-learn*. Nas linhas 8 e 9, foram importados métodos para divisão da base e validação cruzada. Nas linhas 10 a 13, adicionaram-se métricas para avaliação de desempenho, incluindo medidas de classificação e de regressão. Na linha 14, foi carregado o *DecisionTreeClassifier*. Nas linhas 15 e 16, foram selecionados recursos de pré-processamento, como codificação, padronização e criação de pipelines. A linha 17 trouxe o *SimpleImputer*, responsável pela imputação de valores. Já na linha 18, foi importado o modelo de Regressão Logística. A linha 19 adicionou o *ColumnTransformer*, enquanto a linha 20 trouxe o *HistGradientBoostingClassifier*. Na linha 21 foram importadas técnicas de validação estratificada. Por fim, a linha 22 incorporou o *XGBClassifier*, ampliando as opções de algoritmos de classificação.

Código 5.1 – Seleção das Bibliotecas

```
1 import pandas as pd
2 import numpy as np
3 import seaborn as sns
4 import matplotlib.pyplot as plt
5 import matplotlib.colors as mcolors
6 import shap
7 from sklearn import tree
8 from sklearn.model_selection import train_test_split,
   cross_val_score
```

```
9 from sklearn.metrics import (  
10     confusion_matrix, accuracy_score, precision_score,  
11     recall_score, f1_score, roc_auc_score,  
12     classification_report, mean_squared_error,  
13     mean_absolute_error, r2_score)  
14 from sklearn.tree import DecisionTreeClassifier  
15 from sklearn.preprocessing import LabelEncoder, OneHotEncoder,  
    StandardScaler  
16 from sklearn.pipeline import Pipeline  
17 from sklearn.impute import SimpleImputer  
18 from sklearn.linear_model import LogisticRegression  
19 from sklearn.compose import ColumnTransformer  
20 from sklearn.ensemble import HistGradientBoostingClassifier  
21 from sklearn.model_selection import StratifiedKFold, cross_validate  
22 from xgboost import XGBClassifier
```

5.3 CONFIGURAÇÃO DA BASE DE DADOS

No Código [5.2](#), nas linhas 1 a 12, foram carregados individualmente os arquivos do SINASC correspondentes aos anos de 2012 a 2023, utilizando o método `read_csv` com delimitador “;”. Em seguida, na linha 13, foi criada a lista `junction_dados`, reunindo todos os `dataframes` anuais. Por fim, na linha 14, aplicou-se o método `concat` para concatenar os conjuntos de dados, resultando em uma base única consolidada para análise.

Código 5.2 – Seleção dos Dados

```
1 b2012 = pd.read_csv('/content/drive/MyDrive/base_TCC_HM/SINASC_2012  
    .csv', delimiter=';')  
2 b2013 = pd.read_csv('/content/drive/MyDrive/base_TCC_HM/SINASC_2013  
    .csv', delimiter=';')  
3 b2014 = pd.read_csv('/content/drive/MyDrive/base_TCC_HM/SINASC_2014  
    .csv', delimiter=';')  
4 b2015 = pd.read_csv('/content/drive/MyDrive/base_TCC_HM/SINASC_2015  
    .csv', delimiter=';')
```

```
5 b2016 = pd.read_csv('/content/drive/MyDrive/base_TCC_HM/SINASC_2016
    .csv', delimiter=';')
6 b2017 = pd.read_csv('/content/drive/MyDrive/base_TCC_HM/SINASC_2017
    .csv', delimiter=';')
7 b2018 = pd.read_csv('/content/drive/MyDrive/base_TCC_HM/SINASC_2018
    .csv', delimiter=';')
8 b2019 = pd.read_csv('/content/drive/MyDrive/base_TCC_HM/SINASC_2019
    .csv', delimiter=';')
9 b2020 = pd.read_csv('/content/drive/MyDrive/base_TCC_HM/SINASC_2020
    .csv', delimiter=';')
10 b2021 = pd.read_csv('/content/drive/MyDrive/base_TCC_HM/SINASC_2021
    .csv', delimiter=';')
11 b2022 = pd.read_csv('/content/drive/MyDrive/base_TCC_HM/SINASC_2022
    .csv', delimiter=';')
12 b2023 = pd.read_csv('/content/drive/MyDrive/base_TCC_HM/SINASC_2023
    .csv', delimiter=';')
13 junction_dados = [b2012, b2013, b2014, b2015, b2016, b2017, b2018,
    b2019, b2020, b2021, b2022, b2023]
14 base = pd.concat(junction_dados)
```

Esse processo foi o que teve o maior tempo de execução e, também, o maior consumo de memória. Por vezes foi necessário reiniciar o ambiente de execução.

5.4 SELEÇÃO DOS DADOS

No Código [5.3](#), na linha 1, foi criada a lista *valores_codmunnasc* contendo os códigos dos municípios de interesse. Em seguida, na linha 2, a base de dados foi filtrada por meio do método *isin*, de modo a manter apenas os registros cujo campo *CODMUNNASC* corresponde a um dos valores presentes na lista definida.

Código 5.3 – Seleção dos Dados

```
1 valores_codmunnasc = [210530, 240800, 317010, 330100, 330630]
2 base = base[base['CODMUNNASC'].isin(valores_codmunnasc)]
```

5.5 SELEÇÃO DAS VARIÁVEIS

Como visto abaixo no Código 5.4, estabeleceram-se as colunas desejadas no filtro `colunas_selecionadas_lit`, de acordo com o apresentado na Seção 4.4. Em seguida o filtro foi usado para criação de uma nova base de nome `base_colunas_lit`.

Código 5.4 – Criação da base com as colunas selecionadas

```
1 colunas_selecionadas_lit = [  
2     'RACACORMAE',  
3     'ESCMAEAGR1',  
4     'LOCNASC',  
5     'ESTCIVMAE',  
6     'IDADEMAE',  
7     'QTDFILMORT',  
8     'QTDFILVIVO',  
9     'PARIDADE',  
10    'GRAVIDEZ',  
11    'PARTO',  
12    'SEMAGESTAC',  
13    'CONSPRENAT',  
14    'TPROBSON',  
15    'APGAR1',  
16    'APGAR5',  
17    'SEXO',  
18    'PESO',  
19    'CODMUNNASC']  
20 base_colunas_lit = base[colunas_selecionadas_lit]
```

5.6 CODIFICAÇÃO DAS VARIÁVEIS CATEGÓRICAS

No Código 5.5, nas linhas 1 a 7, foi definido um dicionário de mapeamento que associa diferentes formas de representação da variável categórica SEXO a valores numéricos padronizados. Em seguida, na linha 8, aplicou-se o método `replace` para substituir os valores originais da coluna por aqueles definidos no dicionário, garantindo uniformidade

na codificação dessa variável.

Código 5.5 – Codificação das Variáveis Categóricas

```
1 mapeamento = {  
2     1: 1,  
3     'M': 1,  
4     2: 2,  
5     'F': 2,  
6     0: 0,  
7     'I': 0}  
8 base['SEXO'] = base['SEXO'].replace(mapeamento)
```

5.7 CLASSIFICAÇÃO BINÁRIA DA VARIÁVEL DE INTERESSE

No Código [5.6](#) realizou-se a transformação da variável contínua PESO em uma variável binária de classificação. Para isso, na linha 1 criou-se a coluna PESO_BINARIO, na qual foi atribuída a regra de valor 1 para casos em que o peso ao nascer fosse inferior a 2.500 gramas e valor 0 para os demais registros. Na linha 2, a coluna original PESO foi removida do conjunto de dados, preservando apenas a versão binária utilizada nas análises posteriores.

Código 5.6 – Binarização da variável de interesse

```
1 base['PESO_BINARIO'] = base['PESO'].apply(lambda x: 1 if x < 2500  
     else 0)  
2 base = base.drop(columns=['PESO'])
```

5.8 EXCLUSÃO DE VALORES PREENCHIDOS COMO "IGNORADO"

O Código [5.7](#) realiza a exclusão de registros considerados "não informados" dentro do conjunto de dados. Para isso, define-se inicialmente a variável *condicoes_exclusao*, que corresponde a uma expressão lógica combinada com operadores "OU" (|). Cada condição

verifica se determinadas variáveis categóricas possuem valores reservados que representam dados ignorados ou inconsistentes.

Após a definição dessas condições, o código aplica a filtragem ao objeto `base`, removendo todos os registros que atendam a qualquer uma delas. Isso é feito pelo uso de `condicoes_exclusao`, que inverte a seleção e mantém apenas os registros válidos. Por fim, o método `.copy()` assegura a criação de uma nova cópia da base filtrada, evitando possíveis problemas de referência ao conjunto de dados original.

Código 5.7 – Exclusão dos valores ignorados

```

1 condicoes_exclusao = (
2     (base["RACACORMAE"] == 9) |
3     (base["ESCMAEAGR1"] == 9) |
4     (base["GRAVIDEZ"] == 9) |
5     (base["PARTO"] == 9) |
6     (base["SEXO"] == 0) |
7     (base["LOCNASC"] == 9))
8 base = base[~condicoes_exclusao].copy()

```

5.9 REMOÇÃO DE LINHAS COM VALORES AUSENTES

No Código 5.8, nas linhas 1 e 2, a base de dados foi dividida em dois subconjuntos de acordo com a variável `PESO_BINARIO`, separando os registros normais dos de baixo peso. Nas linhas 4 a 6, criou-se a coluna auxiliar `valores_faltantes`, responsável por contabilizar a quantidade de campos nulos em cada registro, tanto na base completa quanto nos subconjuntos. Em seguida, na linha 8, a filtragem restringiu os dados da classe majoritária a registros com no máximo dois valores ausentes. Por fim, na linha 10, a coluna auxiliar foi removida, mantendo apenas as variáveis de interesse para a análise.

Código 5.8 – Remoção das linhas com valores ausentes

```

1 base_normal = base[base['PESO_BINARIO'] == 0].copy()
2 base_baixo = base[base['PESO_BINARIO'] == 1].copy()
3
4 base['valores_faltantes'] = base.isnull().sum(axis=1)
5 base_normal['valores_faltantes'] = base_normal.isnull().sum(axis=1)

```

```
6 base_baixo['valores_faltantes'] = base_baixo.isnull().sum(axis=1)
7
8 base_normal_filtrado = base_normal[base_normal['valores_faltantes']
   ].isin([0, 1, 2])]
9
10 base_normal_filtrado = base_normal_filtrado.drop(columns='
   valores_faltantes')
```

5.10 PREENCHIMENTO DOS CAMPOS AUSENTES COM A MEDIANA

No Código [5.9](#) realizou-se o preenchimento dos valores ausentes na base de dados balanceada. Para isso, aplicou-se o comando da linha 1, que substitui cada campo nulo pela mediana calculada a partir dos valores existentes em sua respectiva coluna.

Código 5.9 – Preenchimento com a mediana

```
1 base_balanceada = base_balanceada.fillna(base_balanceada.median(
   numeric_only=True))
```

5.11 TRATAMENTO DO DESBALANCEAMENTO DE CLASSES COM A SUBAMOSTRAGEM ALEATÓRIA

No Código [5.10](#), na linha 1, foi realizada a subamostragem aleatória da classe majoritária, de forma a igualar a quantidade de registros em relação à classe minoritária. Na linha 2, ambos os subconjuntos foram concatenados, originando a base balanceada. Em seguida, na linha 3, aplicou-se o embaralhamento aleatório das observações com redefinição dos índices. Por fim, na linha 4, removeu-se a coluna auxiliar *valores_faltantes*, resultando em um conjunto de dados pronto para as etapas de modelagem.

Código 5.10 – Subamostragem Aleatória

```
1 base_normal_filtrado = base_normal_filtrado.sample(n=len(base_baixo
   ), random_state=42)
```

```
2 base = pd.concat([base_normal_filtrado, base_baixo], axis=0)
3 base = base.sample(frac=1, random_state=42).reset_index(drop=True)
4 base = base.drop(columns='valores_faltantes')
```

5.12 DIVISÃO DA BASE EM DADOS DE TREINO E TESTE

No Código [5.11](#), nas linhas 1 e 2, foram definidos os códigos de municípios utilizados como critério de divisão da base de dados, distinguindo os pertencentes ao conjunto de treino (Mossoró (RN), Imperatriz (MA), Volta Redonda (RJ) e Uberaba (MG)) daqueles destinados ao conjunto de teste (Campos dos Goytacazes (RJ)). Nas linhas 3 e 4, foram criadas cópias da base original para manter a integridade dos dados. Em seguida, nas linhas 5 e 6, aplicou-se o filtro condicional sobre a variável CODMUNNASC, direcionando os registros de acordo com os valores estabelecidos. Por fim, nas linhas 7 e 8, a variável CODMUNNASC foi removida de ambos os subconjuntos, uma vez que já havia cumprido sua função de particionamento.

Código 5.11 – Separação da base entre conjuntos de treino e teste

```
1 valores_codmunnasc_idh = [240800, 210530, 330630, 317010]
2 valores_codmunnasc_campos = 330100
3 train = base.copy()
4 test = base.copy()
5 train = train[train['CODMUNNASC'].isin(valores_codmunnasc_idh)]
6 test = test[test['CODMUNNASC'] == valores_codmunnasc_campos]
7 train = train.drop(columns=['CODMUNNASC'])
8 test = test.drop(columns=['CODMUNNASC'])
```

5.13 CRIAÇÃO DAS VARIÁVEIS DE ENTRADA

Nessa etapa é realizada a organização da base de dados de modo a definir claramente as variáveis de entrada e a variável de saída. Esse procedimento garante que todos os algoritmos recebam a mesma estrutura de dados, assegurando consistência na comparação dos resultados. No Código [5.12](#), nas linhas 1 e 2, o conjunto de treino foi dividido em variáveis independentes (X_{train}) e variável alvo (y_{train}), sendo esta última a coluna

PESO_BINARIO. De forma análoga, nas linhas 3 e 4, o mesmo procedimento foi aplicado ao conjunto de teste, originando X_{test} e y_{test} . Esse processo garante a separação entre preditores e variável de resposta, etapa essencial para a modelagem supervisionada e que servirá de entrada para todos os modelos.

Código 5.12 – Criação das Variáveis de Entrada

```
1 X_train = train.drop(columns=['PESO_BINARIO'])
2 y_train = train['PESO_BINARIO']
3 X_test = test.drop(columns=['PESO_BINARIO'])
4 y_test = test['PESO_BINARIO']
```

5.14 CRIAÇÃO DOS MODELOS

Os modelos preditivos foram construídos de acordo com os algoritmos escolhidos, utilizando a linguagem de programação previamente definida. Abaixo são descritos, em ordem de execução, os códigos respectivos a cada algoritmo.

5.14.1 AdaBoost

No Código [5.13](#), nas linhas 1 a 3, foi instanciado o modelo *AdaboostClassifier* com taxa de aprendizado de 0,5. Esse valor busca equilibrar velocidade e estabilidade no processo de treinamento. O ajuste ocorreu de forma gradual: o treinamento iniciou-se com uma taxa mais elevada e, a cada iteração, esse valor foi reduzido quando se constatava melhora nas métricas de desempenho. Também foi estabelecida a semente aleatória como 42 para assegurar a reprodutibilidade. Na linha 4, o modelo foi treinado a partir dos conjuntos de treino X_{train} e y_{train} .

Código 5.13 – Modelo AdaBoost

```
1 model_ada = AdaboostClassifier(
2     learning_rate=0.5,
3     random_state=42)
4 model_ada.fit(X_train, y_train)
```

5.14.2 Árvores de Decisão

No Código 5.14, nas linhas 1 a 3, foi instanciado o modelo *DecisionTreeClassifier* com profundidade máxima igual a 5 para limitar a complexidade do modelo e reduzir o risco de *overfitting*. A semente aleatória foi fixada em 42 para garantir reprodutibilidade. Em seguida, na linha 4, o modelo foi ajustado aos dados de treino *X_train* e *y_train*, permitindo a construção da árvore de decisão com base nos padrões identificados no conjunto de treinamento.

Código 5.14 – Modelo Árvore de Decisão

```
1 model_arvore = DecisionTreeClassifier(  
2     max_depth=5,  
3     random_state=42)  
4 model_arvore.fit(X_train, y_train)
```

No Código 5.15, nas linhas 1 a 4, foi construída uma estrutura em *dataframe* para armazenar as variáveis preditoras (*feature*) e seus respectivos valores de importância de Gini (*gini_importance*), obtidos diretamente do modelo treinado. Em seguida, os resultados foram ordenados em ordem decrescente de relevância, destacando as variáveis que mais contribuíram para as divisões internas da árvore de decisão. Por fim, na linha 5, o dicionário de resultados foi atualizado para incluir a tabela de importâncias, garantindo que essa informação ficasse registrada junto às demais métricas de desempenho.

Código 5.15 – Cálculo da Importância de Gini

```
1 importancias_gini = pd.DataFrame({  
2     "feature": X_train.columns,  
3     "gini_importance": model.feature_importances_  
4 }).sort_values(by="gini_importance", ascending=False)  
5 resultados["gini_importances"] = importancias_gini
```

5.14.3 Regressão Logística

No Código 5.16, nas linhas 1 e 2, foram identificadas as variáveis numéricas e categóricas do conjunto de treino, separando-as em listas distintas. Nas linhas 3 e 4, foi definido o pipeline *numeric_tf*, responsável por aplicar padronização às variáveis numé-

ricas por meio do *StandardScaler*. De forma complementar, nas linhas 5 e 6, o pipeline *categoric_tf* preparou as variáveis categóricas utilizando a técnica *OneHotEncoder*, que gera codificação binária mesmo para categorias não vistas previamente. Na sequência, nas linhas 7 a 10, o objeto *ColumnTransformer* consolidou ambos os pipelines, aplicando-os de maneira específica conforme o tipo de variável. Posteriormente, nas linhas 11 a 24, foi criado o pipeline *pipe_log*, que integra o pré-processamento definido e o modelo de *LogisticRegression*. Dentre os parâmetros, foi configurado o *max_iter* (número máximo de iterações) para 2.000 após diversos testes. Inicialmente, utilizou-se o valor padrão da biblioteca *Scikit-learn* (100), aumentando-se gradualmente conforme houve melhora no desempenho. O parâmetro *solver* (algoritmo de otimização do problema) foi definido como *liblinear*, recomendado para problemas de classificação binária. A semente aleatória foi estabelecida como 42. Por fim, na linha 15, o pipeline foi ajustado com os dados de treino, realizando a etapa de aprendizado supervisionado.

Código 5.16 – Modelo Regressão Logística

```
1 num_cols = X_train.select_dtypes(include=[np.number]).columns.  
    tolist()  
2 cat_cols = X_train.columns.difference(num_cols).tolist()  
3 numeric_tf = Pipeline(steps=[('scaler',  
4     StandardScaler(with_mean=False))])  
5 categoric_tf = Pipeline(steps=[('onehot',  
6     OneHotEncoder(handle_unknown='ignore'))])  
7 preprocess = ColumnTransformer(transformers=[  
8     ('num', numeric_tf, num_cols),  
9     ('cat', categoric_tf, cat_cols),],  
10    remainder='drop')  
11 pipe_log = Pipeline(steps=[('prep', preprocess), ('model',  
    LogisticRegression(  
12        max_iter=2000,  
13        solver='liblinear',  
14        random_state=42))])  
15 pipe_log.fit(X_train, y_train)
```

5.14.4 XGBoost

No Código 5.17, nas linhas 1 a 3, foram calculadas as quantidades de exemplos positivos e negativos no conjunto de treino, permitindo a definição do parâmetro *scale_pos_weight*, que ajusta o balanceamento da classe minoritária. Nas linhas 5 a 15, foi instanciado o modelo *XGBClassifier* com parâmetros configurados para classificação binária logística, incluindo 400 estimadores, profundidade máxima igual a 4, taxa de aprendizado de 0,05 e amostragem de 80% das instâncias e colunas em cada árvore, além de regularização L2, definição de semente aleatória, utilização de múltiplos núcleos de processamento e métrica de avaliação *logloss*. Nas linhas 16 a 18, foi criado o pipeline *pipe_xgb*, integrando o pré-processamento estabelecido anteriormente ao modelo XGBoost. Por fim, na linha 19, o pipeline foi ajustado com os dados de treino, concluindo a etapa de aprendizado supervisionado.

Código 5.17 – Modelo XGBoost

```
1 pos = y_train.sum()
2 neg = (y_train == 0).sum()
3 scale_pos_weight = float(neg) / float(pos) if pos > 0 else 1.0
4 xgb = XGBClassifier(
5     objective='binary:logistic',
6     n_estimators=400,
7     max_depth=4,
8     learning_rate=0.05,
9     subsample=0.8,
10    colsample_bytree=0.8,
11    reg_lambda=1.0,
12    random_state=42,
13    n_jobs=-1,
14    scale_pos_weight=scale_pos_weight,
15    eval_metric='logloss')
16 pipe_xgb = Pipeline(steps=[
17     ('prep', preprocess),
18     ('model', xgb)])
19 pipe_xgb.fit(X_train, y_train)
```

5.15 CÁLCULO DAS MÉTRICAS DE DESEMPENHO

Para avaliar o desempenho dos modelos construídos, foram adotadas métricas padronizadas que se aplicam de forma idêntica a todos eles. O que varia em cada caso é apenas a entrada utilizada nos cálculos, uma vez que se consideram as variáveis de saída geradas por cada algoritmo durante a fase de predição. Essa padronização assegura comparabilidade direta entre os resultados obtidos, possibilitando identificar com clareza quais modelos apresentam melhor desempenho na tarefa de classificação do BPN.

No Código [5.18](#), nas linhas 1 e 2, foram geradas as predições de classe (y_pred) e as probabilidades associadas à classe positiva (y_pred_proba). Nas linhas 3 a 7, calcularam-se métricas de avaliação fundamentais: acurácia, precisão, revocação, *F1-score* e área sob a curva ROC (AUC). Em seguida, na linha 8, a matriz de confusão foi decomposta em verdadeiros negativos, falsos positivos, falsos negativos e verdadeiros positivos, permitindo o cálculo da especificidade (linha 9). Na linha 10, aplicou-se a validação cruzada com cinco divisões ($cv=5$) para estimar a robustez do modelo. A matriz de confusão completa foi novamente gerada na linha 11. Por fim, nas linhas 12 a 22, todas as métricas e resultados foram organizados em um dicionário, facilitando a análise comparativa entre os diferentes modelos testados.

Código 5.18 – Métricas de Desempenho

```
1 y_pred = model.predict(X_test)
2 y_pred_proba = model.predict_proba(X_test)[: , 1]
3 accuracy = accuracy_score(y_test, y_pred)
4 precision = precision_score(y_test, y_pred)
5 recall = recall_score(y_test, y_pred)
6 f1 = f1_score(y_test, y_pred)
7 auc = roc_auc_score(y_test, y_pred_proba)
8 tn, fp, fn, tp = confusion_matrix(y_test, y_pred).ravel()
9 especificidade = tn / (tn + fp)
10 scores = cross_val_score(model, X_train, y_train, cv=5, scoring="
    accuracy")
11 cm = confusion_matrix(y_test, y_pred)
12 resultados = {
13     "model": model,
```

```
14     "y_pred": y_pred,
15     "accuracy": accuracy,
16     "f1_score": f1,
17     "precision": precision,
18     "recall": recall,
19     "especificidade": especificidade,
20     "auc": auc,
21     "cv_scores": scores,
22     "confusion_matrix": cm}
```

5.16 EXECUÇÃO DO SHAP

Após o treinamento dos modelos e a obtenção das métricas de desempenho, foi realizada a análise das variáveis mais relevantes para a predição por meio da técnica SHAP. Para fins de exemplificação, será apresentada a execução aplicada ao modelo XGBoost, uma vez que o procedimento adotado é semelhante para os demais algoritmos.

O Código [5.19](#) descreve o processo de execução do SHAP Values para o modelo XGBoost, utilizado para identificar a importância de cada variável preditora. O procedimento envolve a extração do modelo do pipeline obtido na etapa de treinamento, a transformação dos conjuntos de treino e teste com o mesmo pré-processamento aplicado durante a modelagem e a definição dos nomes das variáveis transformadas. Em seguida, é criado o *explainer* SHAP, calculam-se os valores no conjunto de teste e, por fim, o gráfico de barras que resume a contribuição média absoluta das variáveis.

Código 5.19 – SHAP no XGBoost para preparação dos dados e bar plot de importâncias

```
1 xgb_model = pipe_xgb.named_steps['model']
2 X_test_transformed = preprocess.transform(X_test)
3 X_train_transformed = preprocess.transform(X_train)
4 feature_names = preprocess.get_feature_names_out()
5 explainer = shap.Explainer(xgb_model, X_train_transformed,
6                             feature_names=feature_names)
7 shap_values = explainer(X_test_transformed)
8 shap.plots.bar(shap_values)
```

```
8 plt.tight_layout()
9 plt.show()
```

O fluxo inicia na linha 1, em que o modelo treinado é extraído do pipeline por meio do comando `pipe_xgb.named_steps['model']`. Nas linhas 2 e 3, os conjuntos de teste e treino passam pelo mesmo pré-processamento utilizado na etapa de modelagem, assegurando que os dados estejam na mesma forma esperada pelo classificador. Em seguida, na linha 4, são recuperados os nomes das variáveis já transformadas pelo pipeline, o que permite que os gráficos gerados identifiquem corretamente cada atributo.

Na linha 5, é criado o *explainer* SHAP a partir do modelo XGBoost e dos dados de treino transformados, utilizando também os nomes das variáveis como parâmetro. Esse passo define a base para calcular a contribuição de cada preditor. A linha 6 executa o cálculo dos valores de SHAP no conjunto de teste, retornando a contribuição individual de cada variável para cada instância avaliada. Por fim, nas linhas 7 a 9, é gerado o gráfico de barras que resume a importância média absoluta dos atributos.

Cabe ressaltar que, durante as tentativas iniciais de aplicação do SHAP, o processo apresentava longa duração de execução que resultava em erros durante o cálculo. Esses problemas possivelmente estavam relacionados ao uso de um conjunto de dados mais extenso e ainda sem o devido tratamento, o que aumentava a complexidade computacional e a ocorrência de inconsistências. Após a redução do volume de dados e a realização do pré-processamento adequado, a execução passou a ocorrer sem erros e permitiu a obtenção dos valores de forma adequada para todos os modelos.

6 RESULTADOS E DISCUSSÕES

6.1 DEFINIÇÃO DO MODELO QUE OBTEVE MELHOR DESEMPENHO

Após a execução dos quatro algoritmos, realizou-se a avaliação de desempenho com base nas métricas acurácia, precisão, *recall*, F1-Score, especificidade, AUC e Validação Cruzada. Os resultados de todos os modelos foram comparados para identificar aquele que apresentou os melhores valores. Também foi calculada uma média geral, obtida pela média aritmética simples das métricas. A Tabela 7 apresenta os valores de desempenho e suas médias, permitindo uma análise comparativa mais clara entre os algoritmos.

Tabela 7 – Métricas principais dos modelos de classificação

Métrica	Modelos de Classificação			
	Árv. de Decisão	AdaBoost	Regr. Logística	XGBoost
Acurácia	0,8010	0,8152	0,7919	0,8183
F1-Score	0,7744	0,7982	0,7596	0,8016
Precisão	0,8448	0,8346	0,8479	0,8378
Recall	0,7149	0,7648	0,6880	0,7685
Especificidade	0,8798	0,8613	0,8870	0,8638
AUC	0,8616	0,8837	0,8480	0,8866
Cross-Val Média	0,7959	0,8051	0,7668	0,8053
Média Geral	0,8103	0,8233	0,7985	0,8260

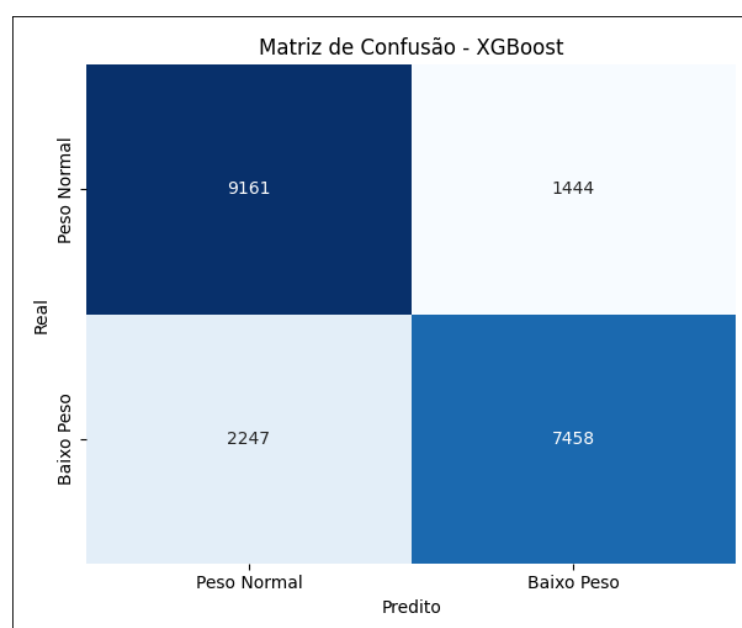
Fonte: Autores, 2025.

Observando individualmente as métricas, nota-se que a Regressão Logística apresentou o maior valor de especificidade (0,8870), porém com o menor *recall* (0,6880). A Árvore de Decisão também obteve boa especificidade (0,8798) e se destacou em Precisão (0,8448), mas apresentou desempenho inferior em *recall* (0,7149). O AdaBoost, por sua vez, mostrou resultados equilibrados, com valores consistentes em todas as métricas, ainda que não tenha liderado em nenhuma delas. Já o XGBoost alcançou os melhores resultados de forma mais abrangente, destacando-se na AUC (0,8866) e obtendo a maior média geral (0,8260), o que evidencia maior equilíbrio e robustez quanto as métricas avaliadas.

A matriz de confusão apresentada na Figura 9 permite visualizar a distribuição

das classificações realizadas pelo modelo XGBoost, o qual obteve melhor média entre as métricas. Por meio dela, é possível observar a quantidade de acertos e erros em cada classe, distinguindo os verdadeiros positivos, verdadeiros negativos, falsos positivos e falsos negativos. O algoritmo classificou corretamente 9.161 casos de recém-nascidos com peso normal e 7.458 casos de baixo peso. Em contrapartida, ocorreram 1.444 falsos positivos (recém-nascidos de peso normal classificados como baixo peso) e 2.247 falsos negativos (recém-nascidos de baixo peso classificados como peso normal).

Figura 9 – Matriz de Confusão do XGBoost (Normal vs Baixo Peso)



Fonte: Autores, 2025.

Esses resultados mostram que, embora ainda existam erros de classificação, o modelo apresenta bom desempenho geral, principalmente por manter elevado o número de verdadeiros positivos na classe de interesse (baixo peso), que é a mais relevante para esse estudo.

6.2 INTERPRETAÇÃO DOS RESULTADOS PARA IDENTIFICAÇÃO DAS VARIÁVEIS MAIS IMPORTANTES PARA A PREDIÇÃO

Os resultados obtidos após a execução dos modelos foram analisados pelo algoritmo *Shapley Values* para identificação das variáveis com maior peso na predição, auxiliando na compreensão do que mais influenciou os resultados.

A Figura 10 apresenta os valores médios absolutos de Shapley para o XGBoost, que indicam a relevância de cada variável para o processo de classificação. Observa-se que a variável mais influente foi SEMAGESTAC¹ com valor médio de aproximadamente 1,33, evidenciando sua forte associação com o peso ao nascer. Em seguida, destacam-se TPROBSON² (0,35), CONSPRENAT³ (0,24) e GRAVIDEZ⁴ (0,22), reforçando a importância de fatores relacionados ao acompanhamento pré-natal e às condições obstétricas no momento do parto.

Outras variáveis que também apresentaram relevância foram SEXO⁵ (0,16), APGAR1⁶ (0,15), APGAR5⁷ (0,13), QTDFILVIVO⁸ (0,08) e ESCMAEAGR1⁹ (0,08). Em contrapartida, variáveis socioeconômicas e demográficas, como a raça/cor e o estado civil materno, apresentaram impacto ainda mais reduzido, indicando que, embora tenham alguma relevância, não se mostraram tão expressivas para a tomada de decisão do modelo quanto as demais métricas.

¹SEMAGESTAC: duração da gestação em semanas.

²TPROBSON: grupo de Robson.

³CONSPRENAT: número de consultas pré-natais.

⁴GRAVIDEZ: tipo de gravidez (única, gemelar, etc.).

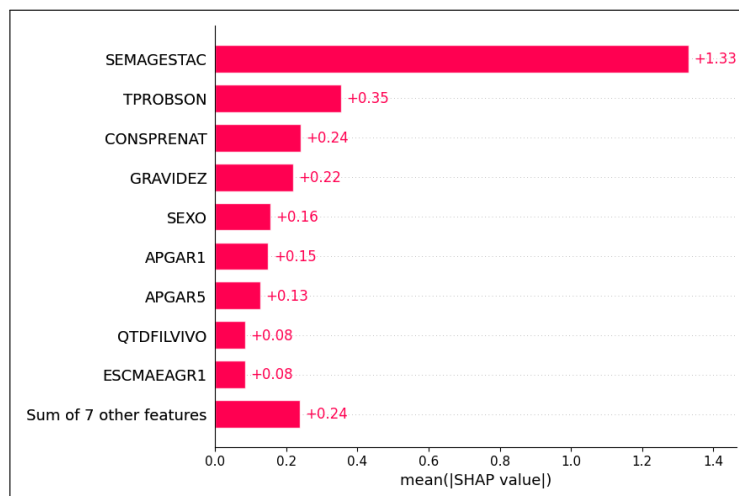
⁵SEXO: sexo do bebê (masculino ou feminino).

⁶APGAR1: índice de Apgar medido no 1º minuto de vida.

⁷APGAR5: índice de Apgar medido no 5º minuto de vida.

⁸QTDFILVIVO: quantidade de filhos vivos.

⁹ESMAEAGR1: escolaridade da mãe em categorias agregadas derivadas do Censo de 2010.

Figura 10 – Valores médios de importância dos *Shapley Values* para o XGBoost

Fonte: Autores, 2025.

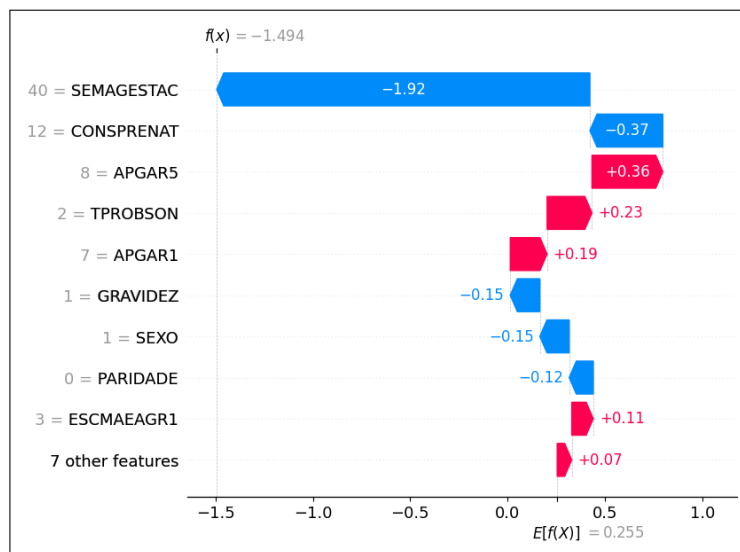
A Figura 11 ilustra a contribuição individual de cada variável para uma previsão específica do modelo XGBoost. Diferente da análise global, este gráfico mostra como os valores de cada atributo atuam sobre uma única amostra, deslocando a predição final em direção à classe de “baixo peso” (1) ou “peso normal” (0).

As barras que apontam para a esquerda representam variáveis que reduziram a probabilidade de baixo peso, enquanto as barras que apontam para a direita indicam aquelas que aumentaram essa probabilidade. Observa-se que SEMAGESTAC exerceu a maior influência negativa, reduzindo fortemente a tendência para baixo peso, seguida de CONSPRENAT, que também apresentou efeito redutor. Em contrapartida, variáveis como APGAR5, TPROBSON e APGAR1 contribuíram positivamente, elevando a chance prevista de baixo peso.

Nesse contexto, o valor $E[f(x)]$ representa a previsão média do modelo na escala *log-odds*, que corresponde a uma probabilidade aproximada de 56% de ocorrência de baixo peso no modelo em um caso qualquer sem a consideração das variáveis. Já o $f(x)$ indica a previsão final para a amostra específica, também expressa em *log-odds*, resultando em cerca de 18% de chance de baixo peso após o ajuste das contribuições individuais das variáveis.

Esse comportamento mostra como, em cada caso individual, o modelo pondera o impacto de cada variável com base em seus valores específicos, permitindo compreender de forma detalhada quais fatores levaram à decisão final do algoritmo.

Figura 11 – Valores locais de importância dos *Shapley Values* para o XGBoost



Fonte: Autores, 2025.

Na Tabela 8 abaixo observam-se os resultados do *Shapley Values* obtidos, evidenciando as três variáveis mais importantes na predição de cada modelo aplicado no estudo.

Tabela 8 – Três principais variáveis por modelo segundo os valores médios de SHAP

Modelo	1º	2º	3º
Árvore de Decisão	SEMAGESTAC	TPROBSON	APGAR5
AdaBoost	SEMAGESTAC	TPROBSON	CONSPRENAT
Regressão Logística	SEMAGESTAC	TPROBSON	APGAR1
XGBoost	SEMAGESTAC	TPROBSON	CONSPRENAT

Fonte: Autores, 2025.

Além dos valores de SHAP, também foi analisada a importância das variáveis por meio do índice de Gini obtido a partir da Árvore de Decisão. Os resultados de Gini apontaram SEMAGESTAC como a variável de maior relevância (0,8367), seguida por TPROBSON (0,1441) e APGAR5 (0,0085), com as demais variáveis apresentando valores inferiores de importância.

A análise comparativa entre os valores de SHAP e a importância de Gini na Árvore de Decisão evidencia consistência quanto à variável mais determinante: SEMAGESTAC, que ocupa a primeira posição em todos os modelos. Na sequência, tanto a métrica de Gini para a Árvore de Decisão quanto o SHAP desta e das demais técnicas destacaram TPROBSON como a segunda variável mais relevante. Já na terceira posição, observa-se uma variação: a Árvore de Decisão indicou APGAR5, enquanto AdaBoost e XGBoost

destacaram CONSPRENAT. Em contrapartida, Regressão Logística destacou APGAR1. Esses resultados mostram que, embora haja pequenas diferenças na ordenação das variáveis terciárias, existe forte convergência sobre os fatores mais determinantes para a classificação, o que reforça a solidez das conclusões.

Além disso, os resultados obtidos neste estudo vão ao encontro das evidências recentes da literatura. [Victor et al. \(2025\)](#), também identificaram o XGBoost como o algoritmo de melhor desempenho na predição de baixo peso ao nascer, com resultados superiores ao observado em métodos tradicionais como a regressão logística. Os autores destacam que algoritmos de *boosting*, como XGBoost e CatBoost, são particularmente poderosos para bases tabulares, pois capturam de maneira mais eficiente relações complexas e não lineares entre os preditores. Dessa forma, os resultados do trabalho apontam a tendência de que técnicas baseadas em *boosting* contribuem para problemas de classificação em saúde pública.

Os estudos de [Victor et al. \(2025\)](#) também apontam a duração da gestação como principal determinante do baixo peso ao nascer, enquanto características socioeconômicas possuem menor influência direta nas predições. Portanto, com os resultados obtidos referentes às métricas e variáveis preditoras, conclui-se que o algoritmo que apresentou melhor desempenho foi o XGBoost, alcançando o melhor resultado em cinco métricas (Acurácia, F1-Score, Recall, AUC e Validação Cruzada). Em seguida, a Regressão Logística destacou-se em duas métricas (Precisão e Especificidade), enquanto a Árvore de Decisão e o AdaBoost apresentaram desempenho consistente, mas sem se sobressair como os melhores em nenhuma métrica específica.

Quanto aos fatores mais determinantes na predição, a partir dos resultados obtidos para esse estudo, percebe-se que o modelo demonstra maior sensibilidade a fatores clínicos e obstétricos do que a determinantes demográficos e sociais. Variáveis como a duração da gestação, o grupo de Robson, APGAR, o tipo de gravidez e o número de consultas pré-natais foram as que mais contribuíram para a capacidade preditiva, refletindo diretamente condições maternas e neonatais no momento do parto. Em contrapartida, características como a idade da mãe, estado civil e raça/cor, por exemplo, tiveram impacto reduzido, indicando que, embora ainda relevantes na análise, não foram tão expressivas para a tomada de decisão do modelo. Esse resultado sugere que a abordagem de aprendizado de

máquina utilizada neste estudo é particularmente eficaz na detecção de padrões clínicos objetivos, enquanto aspectos sociais tendem a exercer influência indireta sobre o baixo peso ao nascer.

7 CONCLUSÕES E TRABALHOS FUTUROS

7.1 CONCLUSÕES

Neste trabalho foi abordado o problema de classificação para identificação dos fatores mais impactantes no nascimento com baixo peso no município de Campos dos Goytacazes (RJ), utilizando técnicas de *Machine Learning* a partir da comparação de diferentes modelos preditivos a partir da base de dados do SINASC.

Os resultados evidenciaram que os fatores de risco mais relevantes estão associados a características clínicas e obstétricas, com destaque para as variáveis SEMAGESTAC (duração da gestação em semanas) e TPROBSON, que se apresentaram como as principais determinantes em todos os algoritmos avaliados. Outras variáveis como Apgar, número de consultas pré-natais e tipo de gravidez também apareceram entre os atributos mais importantes, o que fomenta a relação entre o acompanhamento gestacional adequado e a redução do risco de baixo peso ao nascer.

Adicionalmente, a análise comparativa entre os algoritmos mostrou que, embora todos tenham obtido desempenho satisfatório e relativamente semelhante quanto às métricas, o XGBoost se destacou por apresentar maior equilíbrio entre as métricas, alcançando a melhor média geral e confirmando sua capacidade em cenários de classificação. O AdaBoost também apresentou resultados consistentes, enquanto a Regressão Logística e a Árvore de Decisão, apesar de também úteis para interpretação, obtiveram métricas inferiores em relação aos algoritmos de *boosting*.

Dessa forma, a análise dos resultados permite concluir que os objetivos do estudo foram alcançados. Primeiramente, a comparação dos diferentes modelos de ML possibilitou avaliar suas vantagens e limitações, destacando o potencial de técnicas de *boosting*, como o XGBoost e AdaBoost, para aplicação em problemas de saúde pública, conforme apresentado na literatura de apoio. Em seguida, foi possível determinar os principais fatores que influenciam no baixo peso do recém-nascido, identificando as variáveis que também se alinham às evidências encontradas na literatura.

7.2 TRABALHOS FUTUROS

Como trabalhos futuros, sugere-se a ampliação da base de dados para incluir outras regiões do país, bem como a exploração de algoritmos adicionais, como *Random Forest*, *CatBoost* e redes neurais, que podem contribuir para aprimorar a acurácia preditiva. Além disso, sugere-se realizar uma revisão de literatura a partir de profissionais da área de saúde gestacional, incorporando outras áreas do conhecimento, e conseqüentemente contribuindo para um entendimento maior sobre o tema.

REFERÊNCIAS

- ALBISUA, I. et al. The quest for the optimal class distribution: an approach for enhancing the effectiveness of learning via resampling methods for imbalanced data sets. *Progress in Artificial Intelligence*, v. 2, n. 1, p. 45–63, 2013. Disponível em: <http://link.springer.com/10.1007/s13748-012-0034-6>.
- ARAYESHGARI, M. et al. Machine learning-based classifiers for the prediction of low birth weight. *Healthcare Informatics Research*, v. 29, n. 1, p. 54–63, 2023.
- ATLASBR. *Ranking de IDH*. 2010. Acesso em: 23 out. 2025. Disponível em: <http://www.atlasbrasil.org.br/ranking>.
- BATISTA, G. E. de A. P. A. *Pré-processamento de Dados em Aprendizado de Máquina Supervisionado*. Tese (Doutorado) — Universidade de São Paulo, 2003. São Paulo: São Carlos. Acesso em: 03 out. 2024. Disponível em: <http://www.teses.usp.br/teses/disponiveis/55/55134/tde-06102003-160219>.
- BEKELE, W. T. Machine learning algorithms for predicting low birth weight in ethiopia. *BMC Medical Informatics and Decision Making*, v. 22, n. 1, p. 232, 2022. Disponível em: <https://doi.org/10.1186/s12911-022-01981-9>.
- BORBA, M. F. *Análise da generalização de algoritmos de machine learning e suas aplicações na otimização de decisões em saúde*. Tese (Doutorado) — Universidade de São Paulo, São Paulo: São Paulo, 2023. Acesso em: 05 ago. 2024. Disponível em: <https://www.teses.usp.br/teses/disponiveis/6/6141/tde-05022024-163230>.
- BORGES, M. M. Monografia de Especialização, *Machine Learning como ferramenta gerencial para predição de indicadores e detecção de anomalias*. Porto Alegre: [s.n.], 2020. Acesso em: 03 out. 2024. Disponível em: <https://lume.ufrgs.br/handle/10183/218603>.
- BRASIL. Ministério da Saúde. *Sistema de Informações sobre Nascidos Vivos (SINASC) e Sistema de Informações sobre Mortalidade (SIM)*. 2020. Acesso em: 10 set. 2024. Disponível em: <https://svs.aids.gov.br/daent/cgiae>.
- Brasil. Ministério da Saúde. *Coordenação-Geral de Informações e Análises Epidemiológicas (CGIAE): Subordinada ao Departamento de Análise Epidemiológica e Vigilância de Doenças Não Transmissíveis (DAENT)*. 2024. Acesso em: 01 out. 2024. Disponível em: <https://svs.aids.gov.br/daent/cgiae/sinasc/apresentacao>.
- CHANG, Y.-C.; CHANG, K.-H.; WU, G.-J. Application of extreme gradient boosting trees in the construction of credit risk assessment models for financial institutions. *Applied Soft Computing Journal*, v. 73, p. 914–920, 2018.
- CHEN, T.; GUESTRIN, C. Xgboost: A scalable tree boosting system. In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. San Francisco, California, USA: ACM, 2016. (KDD '16), p. 785–794. ISBN 978-1-4503-4232-2. Disponível em: <https://dl.acm.org/doi/10.1145/2939672.2939785>.

COLLIN, C. B. et al. Computational models for clinical applications in personalized medicine—guidelines and recommendations for data integration and model validation. *Journal of Personalized Medicine*, v. 12, n. 2, p. 166, 2022.

DELPINO, F. M. et al. Predicting all-cause mortality with machine learning among brazilians aged 50 and over: results from the brazilian longitudinal study of ageing (elsi-brazil). *npj Aging*, v. 11, n. 22, 2025. Acesso em: 29 ago. 2025. Disponível em: <https://doi.org/10.1038/s41514-025-00210-7>.

FERNANDES, F. T.; FILHO, A. D. P. C. Perspectivas do uso de mineração de dados e aprendizado de máquina em saúde e segurança no trabalho. *Revista Brasileira de Saúde Ocupacional*, v. 44, p. e13, 2019. ISSN 2317-6369, 0303-7657.

FILHO, A. D. P. C. Uso de big data em saúde no brasil: perspectivas para um futuro próximo. *Epidemiologia e Serviços de Saúde*, v. 24, n. 2, p. 325–332, 2015. Acesso em: 05 out. 2024. Disponível em: http://www.iec.pa.gov.br/template_doi_ess.php?doi=10.5123/S1679-49742015000200015&scielo=S2237-96222015000200325.

FREITAS, D. d. S. Análise e predição de mortalidade infantil utilizando modelos de aprendizado de máquina. *Universidade Federal do Ceará*, 2023. Ceará: Quixadá. Acesso em: 21 set. 2024. Disponível em: https://repositorio.ufc.br/bitstream/riufc/76519/1/2023_tcc_dsfreitas.pdf.

FREUND, Y.; SCHAPIRE, R. E. A short introduction to boosting. *Journal of Japanese Society for Artificial Intelligence*, v. 14, n. 5, p. 771–780, 1999.

GOPINATH, D. The shapley value for ml models. *Towards Data Science*, 2021. Acesso em: 28 ago. 2025. Disponível em: <https://towardsdatascience.com/the-shapley-value-for-ml-models-f1100bff78d1/>.

GREENER, J. G. et al. A guide to machine learning for biologists. *Nature Reviews Molecular Cell Biology*, v. 23, n. 1, p. 40–55, 2022. Disponível em: <https://doi.org/10.1038/s41580-021-00407-0>.

HENRIQUES, L. B. et al. Acurácia da determinação da idade gestacional no sistema de informações sobre nascidos vivos (sinasc): um estudo de base populacional. *Cadernos de Saúde Pública*, v. 35, n. 3, p. e00098918, 2019. Acesso em: 27 ago. 2025. Disponível em: <https://www.scielo.org/article/csp/2019.v35n3/e00098918/>.

Instituto Brasileiro de Geografia e Estatística (IBGE). *Instituto Brasileiro de Geografia e Estatística*. 2022. Acesso em: 28 ago. 2025. Disponível em: <https://www.ibge.gov.br/>.

KUMAR, M.; BHARDWAJ, V. Evaluating label encoding and preprocessing techniques for breast cancer prediction using machine learning algorithms. *International Journal of Computational Intelligence Systems*, v. 18, n. 218, 2025. Acesso em: 31 out. 2025. Disponível em: <https://link.springer.com/article/10.1007/s44196-025-00957-7>.

LEITÃO, F. N. C. et al. Escala de apgar em recém-nascidos prematuros: revisão sistemática. *Revista Multidisciplinar em Saúde*, v. 4, n. 4, p. 59–73, 2023. Acesso em: 29 ago. 2025. Disponível em: <https://doi.org/10.51161/integrar/remis/3873>.

MACHADO, E. L. Um estudo de limpeza em base de dados desbalanceada e com sobreposição de classes. *Universidade de Brasília*, 2009. Brasília: Brasília. Acesso em: 29 set. 2024. Disponível em: <http://repositorio.umb.br/handle/10482/1397>.

MAIA, L. T. d. S.; SOUZA, W. V. d.; MENDES, A. d. C. G. Diferenciais nos fatores de risco para a mortalidade infantil em cinco cidades brasileiras: um estudo de caso-controle com base no sim e no sinasc. *Cadernos de Saúde Pública*, v. 36, n. 5, p. e00154919, 2020. Acesso em: 27 ago. 2025. Disponível em: <https://doi.org/10.1590/0102-311X00154919>.

MEDEIROS, G. S. d. Trabalho de Conclusão de Curso (Bacharelado em Estatística), *Associação entre as características maternas e do recém-nascido e a macrosomia fetal no Estado da Bahia: uma análise usando aprendizado de máquina*. Niterói, RJ: [s.n.], 2023. Acesso em: 27 ago. 2025. Disponível em: <https://estatistica.uff.br/wp-content/uploads/sites/33/2024/02/119054015PFII-Gabriel-Silva-de-Medeiros.pdf>.

MOREIRA, A.; SOUSA, P.; SARNO, F. Baixo peso ao nascer e seus fatores associados. *Instituto Insraelista de Ensino e Pesquisa Albert Einstein*, 2018.

MOREIRA, J. R. H. Classificação de risco neonatal usando aprendizado de máquina e dados dos sistemas de informação de saúde pública e de censo demográfico brasileiro. *Universidade Federal de Juiz de Fora, Instituto de Ciências Exatas*, 2023. Minas Gerais: Juiz de Fora.

MUREL, J.; KAVLAKOGLU, E. *What is ensemble learning?* 2024. Acesso em: 24 set. 2024. Disponível em: <https://www.ibm.com/topics/ensemble-learning>.

NERI, A. B. T. et al. Utilização de modelos de aprendizado de máquina baseados em árvore para predição de baixo peso ao nascer. *Universidade de Pernambuco (UPE)*, 2023. Pernambuco: Caruaru.

OLIVEIRA, A. L. Perceptron dilatação-erosão linear com treinamento baseado em otimização dc: aplicações em problemas de regressão e classificação. *Universidade Estadual de Campinas, Instituto de Matemática, Estatística e Computação Científica*, 2023. São Paulo: Campinas. Acesso em: 29 set. 2024. Disponível em: <https://hdl.handle.net/20.500.12733/9642>.

OLIVEIRA, B. *Algoritmos de Aprendizado de Máquina na Predição e Avaliação de Evasão de Clientes em Ambiente de Produção*. Dissertação (Dissertação (Mestrado)) — Universidade Federal de Goiás, Instituto de Informática, Goiânia, 2021. Orientador: Prof. Anderson da Silva Soares.

PAIXAO, G. M. D. M. et al. Machine learning na medicina: Revisão e aplicabilidade. *Arquivos Brasileiros de Cardiologia*, v. 118, n. 1, p. 95–102, 2022. ISSN 0066-782X, 1678-4170. Disponível em: <https://abccardiol.org/article/machine-learning-na-medicina-revisao-e-aplicabilidade/>.

PAVANYA, M. et al. Prediction of birthweight with early and mid-pregnancy antenatal markers utilising machine learning and explainable artificial intelligence. *Scientific Reports*, 2025. Acesso em: 28 ago. 2025.

PEDRAZA, D. F. Baixo peso ao nascer no brasil: revisão sistemática de estudos baseados no sistema de informações sobre nascidos vivos. *Revista de Atenção à Saúde*, v. 12, n. 41, p. 37–50, 2014. Acesso em: 27 ago. 2025. Disponível em: https://seer.uscs.edu.br/index.php/revista_ciencias_saude/article/view/2237.

PEIXOTO, T. d. O. *Comparação de estratégias para lidar com o desbalanceamento de classes: um estudo de caso com dados de mortalidade neonatal no Rio Grande do Sul*. 2023. Rio Grande do Sul: Porto Alegre. Acesso em: 10 set. 2024. Disponível em: <https://lume.ufrgs.br/>.

POTDAR, K.; PARDAWALA, T. S.; PAI, C. D. A comparative study of categorical variable encoding techniques for neural network classifiers. *International Journal of Computer Applications*, v. 175, n. 4, 2017. Acesso em: 31 ago. 2025. Disponível em: <https://www.ijcaonline.org/archives/volume175/number4/potdar-2017-ijca-915495.pdf>.

REMIGIO, M. *Regressão Logística — Logistic Regression*. 2020. Acesso em: 12 out. 2024. Disponível em: <https://medium.com/@msremigio/regress%C3%A3o-log%C3%ADstica-logistic-regression-997c6259ff9a>.

ROBSON, M.; HARTIGAN, L.; MURPHY, M. Methods of achieving and maintaining an appropriate caesarean section rate. *Best Practice Research Clinical Obstetrics Gynaecology*, v. 27, n. 2, p. 297–308, 2013. ISSN 1521-6934. Caesarean Section – Current Practice. Disponível em: <https://www.sciencedirect.com/science/article/pii/S1521693412001575>.

SANTOS, L. I. Adaptação de algoritmos híbridos baseados em aprendizagem de máquinas para aplicação em problemas na área de saúde com bases de dados desbalanceadas. *UNIVERSIDADE ESTADUAL DE MONTES CLAROS*, 2021. Minas Gerais: Montes Claros.

SANTOS, R. et al. Prevalence and factors associated with low birth weight in full-term newborns. *Rev Rene*, 2021.

SARKER, I. H. Machine learning: Algorithms, real-world applications and research directions. *SN Computer Science*, v. 2, n. 160, 2021.

SILVA, D. F. B. F. da. *Pré-processamento de Dados e Comparação entre Algoritmos de Machine Learning para a Análise Preditiva de Falhas em Linhas de Produção para o Controlo de Qualidade*. Tese (Doutorado) — Instituto Superior de Engenharia do Porto, 2021. Portugal: Porto. Acesso em: 05 out. 2024. Disponível em: <https://recipp.ipp.pt/handle/10400.22/18266>.

SILVA, J. M. M. d. *Impacto da pandemia da COVID-19 e modelos de aprendizagem de máquina para predição de prematuridade no Brasil*. Dissertação (Dissertação de Mestrado em Ciência da Computação) — Universidade Federal de Pernambuco, Recife, 2022.

SILVA JUNIOR, V. E. d. *Uma nova abordagem do Real AdaBoost resistente a overfitting para classificação de dados binários*. Dissertação (Dissertação de Mestrado) — Universidade Federal de Pernambuco, Recife, 2016.

SIVAKUMAR, A.; GUNASUNDARI, R. A survey on data preprocessing techniques for bioinformatics and web usage mining. *International Journal of Pure and Applied*

Mathematics, v. 117, n. 20, p. 785–794, 2017. Acesso em: 03 out. 2024. Disponível em: <https://acadpubl.eu/jsi/2017-117-20-22/articles/20/68.pdf>.

SOARES, W. L. G. et al. Caracterizando a mortalidade infantil utilizando técnicas de machine learning: um estudo de caso em dois estados brasileiros – santa catarina e amapá. *Brazilian Journal of Development*, v. 7, n. 5, p. 10360–10375, 2021. Recebido em: 07 abr. 2021; Aceito em: 03 maio 2021. Acesso em: 28 ago. 2025. Disponível em: <https://doi.org/10.34117/bjdv7n5-106>.

TRAN, T.; LE, U.; SHI, Y. An effective up-sampling approach for breast cancer prediction with imbalanced data: A machine learning model-based comparative analysis. *PLOS ONE*, Public Library of Science, v. 17, n. 5, p. e0269135, 2022. Disponível em: <https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0269135>.

UNICEF. *Low birthweight*. 2023. Acesso em: 10 set. 2024. Disponível em: <https://data.unicef.org/topic/nutrition/low-birthweight>.

VICTOR, A. et al. Predicting low birth weight risks in pregnant women in brazil using machine learning algorithms: data from the araraquara cohort study. *BMC Pregnancy and Childbirth*, v. 25, n. 320, 2025. Acesso em: 28 ago. 2025. Disponível em: <https://doi.org/10.1186/s12884-025-07351-3>.

XGBoost Developers. *XGBoost Documentation*. [S.l.], 2024. Acesso em: 04 out. 2024. Disponível em: <https://xgboost.readthedocs.io/en/latest/index.html>.

XU, Y.; GOODACRE, R. On splitting training and validation set: A comparative study of cross-validation, bootstrap and systematic sampling for estimating the generalization performance of supervised learning. *Journal of Analysis and Testing*, Springer, v. 2, n. 3, p. 249–262, 2018.

Predição e Classificação das Variáveis
Determinantes de Baixo Peso em Recém-Nascidos

UMA ABORDAGEM COM MACHINE LEARNING

🌐 www.atenaeditora.com.br

✉ contato@atenaeditora.com.br

📷 @atenaeditora

📘 www.facebook.com/atenaeditora.com.br



Atena
Editora
Ano 2026

Predição e Classificação das Variáveis
Determinantes de Baixo Peso em Recém-Nascidos

UMA ABORDAGEM COM MACHINE LEARNING

🌐 www.atenaeditora.com.br

✉ contato@atenaeditora.com.br

📷 @atenaeditora

📘 www.facebook.com/atenaeditora.com.br



Atena
Editora
Ano 2026