

UNIVERSIDADE FEDERAL DE SANTA CATARINA
CENTRO TECNOLÓGICO
DEPARTAMENTO DE INFORMÁTICA E ESTATÍSTICA
CIÊNCIA DA COMPUTAÇÃO

William Silveira Figueiredo

**Um estudo sobre a Aplicabilidade de Modelos de Aprendizado de Máquina para Previsão
de Absenteísmo em Serviços de Saúde**

Florianópolis
2026

William Silveira Figueiredo

**Um estudo sobre a Aplicabilidade de Modelos de Aprendizado de
Máquina para Previsão de Absenteísmo em Serviços de Saúde**

Trabalho de Conclusão de Curso submetido ao
Curso de Graduação em Ciência da Computa-
ção do Centro Tecnológico da Universidade
Federal de Santa Catarina como requisito para
obtenção do título de Bacharel em Ciência da
Computação.

Orientador: Prof. Eduardo Camilo Inacio, Dr.

Florianópolis

2026

William Silveira Figueiredo

Um estudo sobre a Aplicabilidade de Modelos de Aprendizado de Máquina para Previsão de Absenteísmo em Serviços de Saúde

Este Trabalho de Conclusão de Curso foi julgado adequado para obtenção do Título de Bacharel em Ciência da Computação e aprovado em sua forma final pelo curso de Graduação em Ciência da Computação.

Florianópolis, 15 de junho de 2026.

Banca Examinadora:

Prof. Eduardo Camilo Inacio, Dr.
Orientador
Universidade Federal de Santa Catarina

Profa. Jerusa Marchi, Dra.
Avaliadora
Universidade Federal de Santa Catarina

Prof. Douglas Dyllon Jeronimo de Macedo,
Dr.
Avaliador
Universidade Federal de Santa Catarina

AGRADECIMENTOS

Agradeço, em primeiro lugar, à minha família. À minha mãe, Maria, ao meu pai, Wellington, e à minha irmã, Williane, pelo apoio incondicional, pelo incentivo nos momentos difíceis e por serem a base de tudo o que construí até aqui.

Agradeço também ao meu orientador, Eduardo Camilo, pela orientação, pela paciência e pelos ensinamentos que tornaram este trabalho possível.

A todos, o meu sincero obrigado.

LISTA DE FIGURAS

Figura 1 – Esquema geral de uma árvore de decisão.	22
Figura 2 – Trecho da planilha de resultados da regressão logística.	46
Figura 3 – Arquitetura de execução dos experimentos no Azure.	48

LISTA DE TABELAS

Tabela 1 – <i>One-Hot Encoding</i> da variável <i>periodo_dia</i> com <i>manha</i> como categoria de referência.	25
Tabela 2 – <i>Label Encoding</i> da variável <i>periodo_dia</i>	25
Tabela 3 – Consequências de trocar a codificação entre os modelos.	26
Tabela 4 – Amostra de cinco agendamentos do conjunto de dados (subconjunto das variáveis, com a lista completa na Tabela 5).	31
Tabela 5 – Dicionário de variáveis do conjunto de dados.	32
Tabela 6 – Distribuição das classes no conjunto de dados.	32
Tabela 7 – Divisão dos dados em treino, validação e teste.	33
Tabela 8 – Tratamento aplicado a cada variável do conjunto de dados.	34
Tabela 9 – Tamanho e proporção de faltas de cada conjunto, após o pré-processamento.	35
Tabela 10 – Variáveis derivadas a partir das colunas originais.	35
Tabela 11 – Granularidades testadas para a categorização da idade.	36
Tabela 12 – Granularidades testadas para a categorização da faixa de espera.	37
Tabela 13 – Estratégias de balanceamento testadas.	37
Tabela 14 – Limiares mínimos testados para o agrupamento de bairros.	38
Tabela 15 – Conjuntos de <i>features</i> de interação testados.	38
Tabela 16 – Combinação dos cinco fatores de pré-processamento.	39
Tabela 17 – Codificação aplicada às variáveis categóricas em cada modelo.	40
Tabela 18 – As 32 combinações de hiperparâmetros da regressão logística.	43
Tabela 19 – As 14 combinações de hiperparâmetros da árvore de decisão.	45
Tabela 20 – Permutação total de experimentos por modelo.	46
Tabela 21 – Panorama do AUC nas duas famílias de modelos.	51
Tabela 22 – Proporção de configurações de cada família que ultrapassam um dado nível de AUC no teste.	52
Tabela 23 – Estabilidade dos cinco modelos de regressão logística selecionados: AUC nos três conjuntos e diferenças absolutas entre validação e teste.	53
Tabela 24 – Estabilidade dos cinco modelos de árvore de decisão selecionados: AUC nos três conjuntos e diferenças absolutas entre validação e teste.	53
Tabela 25 – Métricas dos melhores modelos de cada família, no teste.	54
Tabela 26 – Matrizes de confusão no teste (18.593 agendamentos, dos quais 5.208 faltas).	55
Tabela 27 – Estratégias simples comparadas aos melhores modelos, no teste.	56
Tabela 28 – Resumo da comparação: a estratégia que aparece como melhor em cada métrica e a interpretação correspondente.	57
Tabela 29 – Variação das métricas do modelo v77 (regressão logística) por limiar de decisão, no teste.	58

Tabela 30 – Variação das métricas do modelo v304 (árvore de decisão) por limiar de
decisão, no teste. 59

RESUMO

O absenteísmo é uma das principais métricas que impactam o custo em instituições nos mais diversos setores. Na saúde em específico, onde há uma alta taxa de não comparecimento de pacientes, existem grandes filas para realização de consultas, exames e procedimentos. Nesse contexto, esse trabalho propõe uma comparação entre modelos de aprendizado de máquina, com foco em regressão logística e árvore de decisão, visando prever o absenteísmo e, assim, fornecer subsídios para desenvolvedores de modelos preditivos que visam resolver problemas do setor de saúde com aprendizado de máquina. Para isso, foram geradas 243 versões do conjunto de dados a partir de diferentes decisões de pré-processamento, combinadas com as configurações de cada modelo, totalizando mais de onze mil experimentos, avaliados por métricas apropriadas a dados desbalanceados, como AUC, precisão, *recall* e F1, além da comparação com estratégias simples de referência e da análise do limiar de decisão, de modo a identificar o que cada abordagem efetivamente oferece para mitigar o problema do absenteísmo. Os resultados mostraram que a regressão logística superou a árvore de decisão, alcançando um desempenho modesto, porém estável, com AUC em torno de 0,60 em dados nunca vistos, enquanto a árvore de decisão apresentou forte sobreajuste e desempenho próximo ao acaso fora do conjunto de treino. Mais do que um modelo pronto para uso, o trabalho entrega uma avaliação controlada e reprodutível do que esses modelos conseguem, e do que não conseguem, oferecer como apoio à gestão do absenteísmo.

Palavras-chave: Inteligência Artificial. Modelos preditivos. Absenteísmo. Saúde. Aprendizado de Máquina.

ABSTRACT

Absenteeism is one of the main metrics that impact costs in institutions in the most diverse sectors. In healthcare specifically, where there is a high rate of patient non-attendance, there are long queues for appointments, exams and procedures. In this context, this work proposes a comparison between machine learning models, focusing on logistic regression and decision tree, aiming to predict absenteeism and, thus, provide support for developers of predictive models that aim to solve problems in the health sector with machine learning. To this end, 243 versions of the dataset were generated from different pre-processing decisions, combined with the configurations of each model, totaling more than eleven thousand experiments, evaluated with metrics suited to imbalanced data, such as AUC, precision, recall and F1, in addition to a comparison with simple baseline strategies and an analysis of the decision threshold, in order to identify what each approach effectively offers to mitigate the problem of absenteeism. The results showed that logistic regression outperformed the decision tree, achieving a modest but stable performance, with an AUC of around 0.60 on unseen data, while the decision tree showed strong overfitting and performance close to chance outside the training set. More than a ready-to-use model, this work delivers a controlled and reproducible assessment of what these models can, and cannot, offer as support for managing absenteeism.

Keywords: Artificial Intelligence. Predictive models. Absenteeism. Health. Machine Learning.

SUMÁRIO

1	INTRODUÇÃO	12
1.1	OBJETIVOS	13
1.1.1	Objetivo Geral	13
1.1.2	Objetivos Específicos	13
1.2	JUSTIFICATIVA	14
1.3	ESCOPO	14
2	FUNDAMENTAÇÃO TEÓRICA	16
2.1	ABSENTEÍSMO NA SAÚDE	16
2.1.1	Histórico de Gestão de Absenteísmo na Saúde	16
2.1.2	Conceito Geral de Absenteísmo	16
2.2	APRENDIZADO SUPERVISIONADO	17
2.2.1	Subáreas da Inteligência Artificial	18
2.2.2	Histórico de Desenvolvimento da Inteligência Artificial	19
2.3	MODELOS DE CLASSIFICAÇÃO	20
2.3.1	Regressão Logística	20
2.3.2	Árvore de Decisão	21
2.4	PRÉ-PROCESSAMENTO DE DADOS	23
2.4.1	Codificação de Variáveis Categóricas	23
2.4.2	One-Hot Encoding	24
2.4.3	Label Encoding	25
2.4.4	Diferenças entre One-Hot e Label Encoding	25
2.4.5	Normalização de Variáveis Numéricas	26
2.5	MÉTRICAS	27
2.5.1	Acurácia	28
2.5.2	Precisão	28
2.5.3	Recall	28
2.5.4	F1-Score	29
2.5.5	AUC	29
2.6	TRABALHOS RELACIONADOS	29
3	METODOLOGIA	31
3.1	CONJUNTO DE DADOS	31
3.1.1	Descrição do Conjunto de Dados	31
3.1.2	Dicionário de Dados	31
3.1.3	Conjunto de treino, conjunto de validação e conjunto de teste	33
3.1.4	Limpeza de Dados	34

3.1.5	Variáveis derivadas	35
3.1.6	Categorização por idade	36
3.1.7	Categorização por faixa de espera	36
3.1.8	Balanceamento de classes	37
3.1.9	Limiar de bairros	37
3.1.10	<i>Features</i> de interação	38
3.1.11	Geração dos 243 conjuntos de dados base	39
3.2	PRÉ-PROCESSAMENTO E ENGENHARIA DE <i>FEATURES</i>	39
3.2.1	Codificação de Variáveis Categóricas	39
3.2.2	Normalização de Variáveis	40
3.3	MODELOS: REGRESSÃO LOGÍSTICA E ÁRVORE DE DECISÃO	40
3.3.1	Regressão Logística	41
3.3.1.1	<i>Solver</i>	41
3.3.1.2	<i>Penalty</i>	41
3.3.1.3	<i>Parâmetro C</i>	42
3.3.1.4	<i>L1 Ratio</i>	42
3.3.1.5	<i>Total de combinações</i>	42
3.3.2	Árvore de Decisão	44
3.3.2.1	<i>Criterion</i>	44
3.3.2.2	<i>Max Depth</i>	44
3.3.2.3	<i>Total de combinações</i>	45
3.4	PERMUTAÇÃO TOTAL DE EXPERIMENTOS	45
3.5	ARQUITETURA DE EXECUÇÃO	47
3.6	MÉTRICAS NO CONTEXTO DO ABSENTEÍSMO	49
3.6.1	Estratégias de referência	49
4	RESULTADOS	51
4.1	VISÃO GERAL DOS EXPERIMENTOS	51
4.1.1	Panorama geral do AUC	51
4.2	SELEÇÃO E ESTABILIDADE DOS MODELOS	52
4.2.1	Diferença entre estabilidade e qualidade	54
4.3	DESEMPENHO DOS MODELOS FINAIS NO CONJUNTO DE TESTE	54
4.4	COMPARAÇÃO COM ESTRATÉGIAS SIMPLES	55
4.5	ANÁLISE DE LIMIAR DE DECISÃO	57
5	CONCLUSÃO	60
5.1	LIMITAÇÕES DO CONJUNTO DE DADOS	60
5.2	RISCO ÉTICO EM APLICAÇÃO REAL	61
5.3	DA PREVISÃO À POLÍTICA DE INTERVENÇÃO	61
5.4	TRABALHOS FUTUROS	62

REFERÊNCIAS 64

APÊNDICE A – ARTIGO SBC 66

1 INTRODUÇÃO

O absenteísmo, definido por Goodman e Atkin (1984) como “o padrão habitual de ausência de um dever ou obrigação sem justa causa”, é um fenômeno antigo e amplamente estudado em diversos setores. Nos ambientes hospitalar e clínico, o não comparecimento do paciente a consultas, exames ou procedimentos agendados não compromete apenas a continuidade do cuidado: ele mantém ociosos recursos humanos e financeiros já reservados, prolonga as filas de espera e desorganiza o fluxo de atendimento, pois o horário perdido raramente é reaproveitado por outro paciente.

A magnitude do problema é expressiva e persistente. Uma revisão sistemática internacional estimou taxa média de não comparecimento em torno de 23%, variando conforme a região e a especialidade, e apontou o tempo de antecedência entre o agendamento e a consulta e o histórico de faltas anteriores como os determinantes mais consistentes, ao lado de fatores como idade, condição socioeconômica e distância até o serviço (DANTAS et al., 2018). No Brasil, o cenário é ainda mais crítico: Beltrame et al. (2019) encontraram taxas de 38,6% em consultas especializadas no Espírito Santo, associadas a um desperdício estimado em mais de R\$ 18 milhões no período analisado.

Diante desse quadro, diversas estratégias foram propostas para reduzir as faltas. As mais difundidas são os lembretes de consulta, por mensagens de texto (SMS) ou aplicativos como o WhatsApp (Medicina S/A, 2024), e as campanhas de conscientização dos pacientes (Agência Pará, 2024). Há ainda ajustes na própria organização da agenda, como o *overbooking* e a redução do tempo de espera (DANTAS et al., 2018). Os lembretes são a estratégia mais estudada: uma ampla revisão de estudos clínicos mostrou que enviar mensagens de texto aos pacientes de fato aumenta o comparecimento, embora o ganho seja pequeno (GUROL-URGANCI et al., 2013).

O problema dessas estratégias, porém, não é apenas o efeito reduzido. Como são aplicadas da mesma forma a todos os pacientes, sem distinguir quem realmente tende a faltar, acabam gastando esforço e recursos com quem compareceria de qualquer maneira. Por isso, conseguir identificar com antecedência os pacientes de maior risco seria um passo decisivo para direcionar essas ações e torná-las mais eficientes.

É nesse ponto que entram as técnicas de inteligência artificial e aprendizado de máquina, que têm emergido como um caminho promissor para compreender e prever eventos em larga escala no setor de saúde, como na previsão de mortalidade infantil (MBUNGE et al., 2023; SARAVANOU et al., 2021). Aplicadas ao absenteísmo, elas buscam aprender padrões de não comparecimento a partir de dados históricos de agendamento e do perfil dos pacientes. A literatura específica, no entanto, é menos otimista do que aparenta. Uma revisão sistemática de 50 trabalhos sobre predição de faltas reportou desempenho preditivo tipicamente moderado, com a AUC (do inglês *Area Under the Curve*) concentrada entre 0,70 e 0,86 e apenas um estudo acima de 0,90. Mais do que isso, a maioria dos modelos que reportaram acurácia não superava a taxa

de comparecimento de base, o que evidencia como essa métrica engana em dados desbalanceados (CARRERAS-GARCÍA et al., 2020). Os ganhos, quando existem, concentram-se em poucas variáveis estáveis, como o histórico de faltas, o tempo de antecedência do agendamento, a idade e a distância, e os desempenhos mais altos costumam depender de engenharia de atributos específica de cada serviço, o que limita sua generalização entre populações (CARRERAS-GARCÍA et al., 2020). Além disso, decisões automatizadas a partir dessas previsões exigem cautela, para não penalizar justamente os pacientes mais vulneráveis.

O uso desses dados também levanta questões de privacidade e de ética. Os registros de agendamento reúnem informações sensíveis dos pacientes, como condições de saúde, bairro de moradia e participação em programas sociais, cujo tratamento exige proteção adequada e está sujeito à legislação de proteção de dados pessoais. Para além do sigilo, apoiar decisões em variáveis socialmente sensíveis pode reproduzir desigualdades já existentes no acesso à saúde. Este trabalho reconhece esses riscos desde o início e os retoma em detalhe na Seção 5.2, ao tratar do risco ético em uma aplicação real.

Ainda assim, a crescente disponibilidade de dados sobre agendamentos, perfis de pacientes e indicadores institucionais abre espaço para o estudo de abordagens orientadas a dados como apoio à gestão (TOPOL, 2019). Mais do que maximizar a acurácia de um único modelo, o desafio que se coloca à área é avaliar o que essas abordagens de fato entregam e produzir soluções transparentes e reproduzíveis, que possam ser compreendidas e repetidas por outros, e aplicáveis à realidade dos serviços de saúde, sobretudo em cenários de grande volume de dados e recursos escassos, como o do sistema público brasileiro.

1.1 OBJETIVOS

Nas seções abaixo estão descritos o objetivo geral e os objetivos específicos deste Trabalho de Conclusão de Curso (TCC).

1.1.1 Objetivo Geral

Investigar a aplicabilidade de modelos de aprendizado de máquina para a previsão de absenteísmo em serviços de saúde, comparando o desempenho de modelos clássicos e interpretáveis aplicados a dados reais de agendamento, a fim de avaliar o que cada abordagem oferece como potencial apoio à gestão dos serviços de saúde, na etapa preditiva do problema.

1.1.2 Objetivos Específicos

Visando atingir o objetivo geral, foram definidos os seguintes objetivos específicos.

- Preparar os conjuntos de dados dos experimentos, a partir de diferentes decisões de pré-processamento sobre a base de agendamentos.

- Desenvolver modelos preditivos para absenteísmo.
- Validar os modelos preditivos desenvolvidos.
- Propor recomendações baseadas nos resultados obtidos para o desenvolvimento de modelos preditivos para solução de problemas na área da saúde.

1.2 JUSTIFICATIVA

O absenteísmo em consultas é um desafio crítico para o sistema público de saúde brasileiro, em que a alta demanda e a escassez de recursos tornam a eficiência operacional uma necessidade urgente: cada falta representa um horário ocioso que não retorna e uma fila que não anda (BELTRAME et al., 2019). Além do custo operacional, cada falta tem um custo humano: o paciente que não comparece adia o próprio cuidado, o que pode agravar seu quadro de saúde, enquanto o horário ocioso alonga a espera de quem aguarda atendimento. Em um sistema de recursos escassos como o público brasileiro, esse efeito recai com mais força sobre quem tem menos acesso ao serviço, o que reforça a importância de direcionar as ações de reforço para os pacientes com maior risco de faltar, em vez de aplicá-las de forma indiscriminada. Embora existam soluções consagradas, como lembretes e campanhas, elas atuam de forma indiscriminada e têm efeito limitado (GUROL-URGANCI et al., 2013), enquanto os modelos preditivos mais precisos da literatura costumam depender de engenharia específica de cada serviço, são de baixa interpretabilidade e de difícil reprodução em outros contextos (CARRERAS-GARCÍA et al., 2020). Há, portanto, uma lacuna entre o que é tecnicamente sofisticado e o que é efetivamente aplicável na rotina de um serviço de saúde.

É nessa lacuna que este estudo se justifica. Em vez de perseguir o maior desempenho preditivo possível, ele compara modelos clássicos e interpretáveis, a regressão logística e a árvore de decisão, sobre dados reais de agendamento, avaliando de forma transparente o que cada abordagem efetivamente entrega quando submetida a um protocolo experimental rigoroso. A relevância da pesquisa está menos em propor um modelo definitivo e mais em produzir resultados e recomendações reprodutíveis que orientem desenvolvedores e gestores na construção de soluções preditivas viáveis para o setor de saúde.

1.3 ESCOPO

Este trabalho delimita-se à etapa preditiva do problema do absenteísmo. Está dentro do seu escopo a comparação entre regressão logística e árvore de decisão, o *pipeline* experimental sobre os cinco fatores de pré-processamento, a análise do limiar de decisão e a documentação das limitações do conjunto de dados. A escolha por esses dois modelos reflete o tempo e o escopo disponíveis, com foco em dois algoritmos clássicos e interpretáveis, um linear e outro baseado em regras. A comparação com outras famílias de modelos, como redes Bayesianas

ou mapas auto-organizáveis de Kohonen, exigiria mais tempo e fica como trabalho futuro. Está fora do escopo a implementação de qualquer intervenção em ambiente real, a avaliação prospectiva por ensaio controlado, a mensuração do impacto operacional sobre a taxa de absenteísmo, a definição de uma política específica para uma unidade de saúde e a estimativa do custo total de adoção do sistema, incluindo a integração com o sistema de gestão da unidade e o treinamento dos atendentes. Também fica fora do escopo a identificação, a partir dos dados, dos fatores que mais influenciam o absenteísmo, o que exigiria uma análise de importância das variáveis. Fechar a lacuna entre previsão e política exigiria um segundo trabalho, com desenho experimental próprio e parceria com uma rede de saúde. Este trabalho entrega a parte preditiva da equação e mapeia explicitamente o que ainda falta.

2 FUNDAMENTAÇÃO TEÓRICA

Este capítulo reúne os conceitos que fundamentam o trabalho. São apresentados o problema do absenteísmo na saúde, as ideias centrais do aprendizado de máquina supervisionado, os dois métodos comparados, regressão logística e árvore de decisão, as técnicas de preparação dos dados (codificação de variáveis categóricas e normalização) e as métricas usadas para avaliar os modelos em dados desbalanceados.

2.1 ABSENTEÍSMO NA SAÚDE

2.1.1 Histórico de Gestão de Absenteísmo na Saúde

O absenteísmo é um fenômeno antigo que afeta diversos setores, sendo especialmente crítico na saúde, onde impacta diretamente a eficiência operacional das entidades e a qualidade dos serviços prestados. No setor de saúde, esse problema tem sido enfrentado ao longo dos anos com diferentes abordagens, porém, a solução definitiva ainda representa um desafio significativo (Secretaria de Estado da Saúde de Alagoas, 2024).

Estudos recentes reforçam a gravidade do problema. Por exemplo, pesquisa realizada pela revista *Medicina S/A* apontou que clínicas que não utilizam tecnologias avançadas de controle de absenteísmo com modelos preditivos entre os clientes de um grande prontuário brasileiro registraram uma diferença de 8,5% no índice de absenteísmo (Medicina S/A, 2024).

2.1.2 Conceito Geral de Absenteísmo

O absenteísmo é um tema amplamente estudado em diferentes áreas do conhecimento, sendo definido por Goodman e Atkin (1984) como “o padrão habitual de ausência de um dever ou obrigação sem justa causa”. Embora essa definição seja genérica, ele abrange tantas ausências voluntárias quanto involuntárias, fenômeno que pode ser observado em diversos setores, como educação, indústria e, de forma mais crítica, na saúde. O conceito foi expandido ao longo dos anos para incluir dimensões como frequência, duração e causas das ausências, tornando-se um tema central em estudos de gestão e produtividade (GOODMAN; ATKIN, 1984).

De acordo com Fiocruz (2015), o absenteísmo pode ser entendido como um reflexo de diferentes fatores, incluindo condições de trabalho, características individuais e organizacionais, além de aspectos externos, como problemas de saúde ou dificuldades pessoais. Em ambientes hospitalares e clínicos, ele assume uma relevância particular, pois as ausências não afetam apenas o desempenho individual, mas comprometem diretamente a prestação de serviços essenciais.

No setor de saúde, o absenteísmo de pacientes está frequentemente relacionado a barreiras logísticas, falta de recursos financeiros e baixa conscientização sobre a importância do

acompanhamento médico regular (DANTAS et al., 2018). Em clínicas e hospitais, a ausência de pacientes impacta diretamente o planejamento de consultas, a alocação de equipe e o uso de instalações (BELTRAME et al., 2019).

Historicamente, uma das práticas comuns para reduzir o absenteísmo tem sido a utilização de sistemas de confirmação de consultas. O uso de ferramentas simples, como mensagens via WhatsApp, tem mostrado resultados positivos em contextos específicos. Um exemplo relevante é a Santa Casa de Curitiba, que conseguiu reduzir significativamente suas taxas de absenteísmo no ambulatório por meio desse método (Santa Casa de Curitiba, 2024). No entanto, apesar de eficazes em cenários pontuais, essas soluções apresentam limitação de escalabilidade, especialmente em sistemas de saúde complexos, como grandes operadoras ou o Sistema Único de Saúde (SUS) brasileiro.

Campanhas de conscientização também têm sido realizadas como uma estratégia tradicional para combater o absenteísmo. No setor público, iniciativas como a campanha “absenteísmo Zero” realizada no Complexo Hospitalar de Caetés, em Capanema-PA, demonstram o esforço contínuo para sensibilizar a população sobre o impacto das faltas. Sob o tema “A consulta que você falta, faz falta para alguém”, a campanha buscou destacar a importância da presença dos pacientes nas consultas agendadas, evidenciando o impacto coletivo do problema (Agência Pará, 2024).

Apesar dessas abordagens tradicionais tenham obtido resultados pontuais, elas enfrentam limitações significativas para lidar com a complexidade do absenteísmo em larga escala. Métodos manuais ou campanhas de conscientização frequentemente carecem de eficiência operacional e precisão para antecipar padrões e comportamentos dos pacientes (CARRERAS-GARCÍA et al., 2020). Nesse contexto, o avanço de tecnologias, como ferramentas baseadas em aprendizado de máquina, surge como uma alternativa para gestão mais precisa e escalável a nível nacional.

2.2 APRENDIZADO SUPERVISIONADO

A inteligência artificial (IA) é um campo da Ciência da Computação que busca desenvolver sistemas capazes de realizar tarefas que, tradicionalmente, requerem inteligência humana. Essas tarefas incluem reconhecimento de padrões, tomada de decisões, compreensão de linguagem natural, entre outras (RUSSELL; NORVIG, 2016). A IA engloba diversas subáreas, como aprendizado de máquina, Processamento de Linguagem Natural, Visão Computacional e Robótica, cada uma focando em aspectos específicos da simulação da inteligência humana.

Segundo McCarthy et al. (1955), a IA pode ser definida como “a ciência e a engenharia de fazer máquinas inteligentes”. Desde a sua concepção, a Inteligência Artificial tem evoluído significativamente, passando de abordagens simbólicas e baseadas em regras para métodos mais sofisticados, como redes neurais e algoritmos de aprendizado profundo. Essa evolução foi impulsionada pelo aumento da capacidade computacional, disponibilidade de grandes volumes de

dados e avanços nos algoritmos de aprendizagem (GOODFELLOW; BENGIO; COURVILLE, 2016).

2.2.1 Subáreas da Inteligência Artificial

Dentro do vasto campo da inteligência artificial, uma das principais subáreas é o aprendizado de máquina (AM). O AM concentra-se no desenvolvimento de algoritmos capazes de aprender a partir de dados, realizando previsões ou tomando decisões sem a necessidade de programação explícita para cada tarefa específica (BISHOP, 2006). Essa subárea se divide em diversas abordagens, incluindo aprendizado supervisionado, não supervisionado e por reforço, cada uma resolvendo diferentes tipos de problemas e usando diferentes estrutura de dados.

No aprendizado supervisionado, os modelos são treinados utilizando conjuntos de dados rotulados, onde cada exemplo do treinamento inclui uma entrada e saída desejada correspondente. Técnicas comuns são regressão linear, regressão logística, máquinas de vetores de suporte (SVM) e árvores de decisão (BISHOP, 2006). Esses métodos são utilizados em tarefas de classificação e regressão, permitindo a previsão de categorias ou valores contínuos com base em novas entradas de dados.

No aprendizado não supervisionado, os modelos são treinados utilizando um conjunto de dados não rotulados, focando na identificação de padrões e estruturas adjacentes nos dados. O agrupamento (*clustering*) e a análise de componentes principais (PCA) são exemplos dessa abordagem: o primeiro reúne dados semelhantes, revelando agrupamentos naturais, enquanto o segundo reduz a dimensionalidade dos dados (GÉRON, 2019). Essas técnicas são essenciais para explorar e entender a estrutura dos dados sem a necessidade de informações prévias sobre as classes ou categorias presentes.

No aprendizado por reforço, os modelos envolvem agentes que aprendem a tomar decisões sequenciais através de interações com o ambiente, recebendo recompensas ou penalidades com base em suas ações. Essa abordagem é particularmente útil em problemas onde a decisão contínua e a adaptação são cruciais, como em jogos, controle de sistemas dinâmicos e robótica (GÉRON, 2019). O aprendizado por reforço permite que os agentes desenvolvam estratégias otimizadas para maximizar recompensas acumuladas ao longo do tempo, adaptando-se a mudanças e incertezas no ambiente.

As Redes Neurais Artificiais são uma classe de métodos do aprendizado de máquina, e o *Deep Learning* corresponde a uma de suas subáreas. Inspiradas no funcionamento do cérebro humano, as redes neurais são compostas por camadas de nós interconectados, ou neurônios, que processam as informações de forma hierárquica. O *Deep Learning* refere-se ao uso de redes neurais com múltiplas camadas ocultas, capazes de modelar relações complexas e abstrações de alto nível nos dados (GÉRON, 2019). As aplicações de *Deep Learning* são vastas e incluem reconhecimento de imagem, processamento de linguagem natural, tradução automática e previsão de séries temporais, demonstrando sua capacidade de capturar nuances e padrão em grandes

volumes de dados.

Essas subáreas da inteligência artificial não apenas se complementam, mas também fornecem métodos eficazes para resolver problemas complexos em diversos setores. No contexto da previsão de absenteísmo, por exemplo, a combinação de técnicas de aprendizado supervisionado para classificação pode resultar em modelos preditivos robustos e precisos. Essas subáreas impulsionam o aumento da capacidade computacional ao passo que um elevado volume de dados é gerado pelos sistemas a cada dia, o que permite prever novos avanços na aplicação prática de IA.

2.2.2 Histórico de Desenvolvimento da Inteligência Artificial

A trajetória da inteligência artificial é marcada por períodos de inovação, expectativas elevadas e desafios técnicos que refletem a complexidade de criar sistemas capazes de simular a inteligência humana. Desde a sua concepção nas décadas de 1950 e 1960 até os avanços recentes em *deep learning*, a evolução da IA pode ser dividida em várias fases distintas que demonstram as etapas necessárias para superar os desafios tecnológicos sobre esse tema.

A fundação da IA como disciplina acadêmica ocorreu na década de 1950, com a organização da conferência de Darmouth em 1956, que formalizou o estudo da IA e estabeleceu as bases para pesquisas futuras (RUSSELL; NORVIG, 2016). Inicialmente, as abordagens simbólicas predominavam, focando na manipulação de símbolos e na lógica formal para resolver problemas complexos. Durante as décadas de 1960 e 1970, essa abordagem simbólica avançou significativamente com o desenvolvimento de linguagens de programação como o *Lisp* e a criação de sistemas que imitavam a tomada de decisão humana. Projetos pioneiros, como o ELIZA, um dos primeiros programas de processamento de linguagem natural demonstraram o potencial das máquinas em interagir com humanos de maneira rudimentar, embora enfrentassem limitações em termos de escalabilidade e adaptabilidade a contexto mais amplos (RUSSELL; NORVIG, 2016).

O período conhecido com Inverno da IA que se estendeu pelas décadas de 1970 e 1980, foi caracterizado por uma redução no financiamento e no interesse pela pesquisa em IA. As expectativas excessivamente otimistas não foram atendidas, e os sistemas de IA da época mostraram-se incapazes de lidar com a ambiguidade e a complexidade do mundo real, levando a desilusão temporária na comunidade científica e na indústria (RUSSELL; NORVIG, 2016).

A partir dos anos 1980, a IA começou a ressurgir graças aos avanços na aprendizagem de máquina e nas Redes Neurais Artificiais. A introdução de algoritmos mais eficientes e a exploração técnicas de estatísticas permitiram que os sistemas de IA aprendessem a partir de dados, superando algumas das limitações das abordagens simbólicas. Modelos como as máquinas de vetores de suporte (SVM) e as Redes Neurais Multicamadas (MLP) ganham destaque, proporcionando melhorias significativas em tarefas de classificação e reconhecimento de padrões (BISHOP, 2006).

No século XXI, a combinação de grandes volumes de dados (*Big Data*), aumento da capacidade computacional e avanços em algoritmos de *Deep Learning* impulsionou uma nova era de inovação na IA. As redes neurais permitiram o desenvolvimento de modelos altamente eficazes em diversas aplicações, como reconhecimento de imagem, processamento de linguagem natural e previsão de séries temporais (GÉRON, 2019; RUSSELL; NORVIG, 2016). Ferramentas e bibliotecas como *TensorFlow* e *Scikit-Learn* facilitaram a implementação e a experimentação de modelos de aprendizagem de máquina, democratizando assim o acesso à inteligência artificial e acelerando a pesquisa e o desenvolvimento do tema (GÉRON, 2019).

Atualmente, a IA está profundamente integrada em uma ampla gama de setores, incluindo saúde, mercado corporativo, automobilismo e entretenimento. No contexto da saúde, por exemplo, técnicas de aprendizagem supervisionada e redes neurais são utilizadas para prever absenteísmo, otimizar o agendamento de consultas e personalizar tratamentos médicos (TOPOL, 2019). A capacidade da IA em analisar grandes conjuntos de dados e identificar padrões complexos contribui significativamente para a eficiência operacional e a melhoria da qualidade dos serviços prestados.

A IA moderna enfrenta desafios contínuos relacionados à ética e privacidade dos dados, exigindo esforços contínuos para desenvolver algoritmos mais transparentes e justos, garantindo a confiança e a adoção responsável das tecnologias (RUSSELL; NORVIG, 2016). Apesar desses desafios, o futuro da IA promete avanços ainda mais significativos, impulsionados pela pesquisa contínua em áreas como aprendizagem por reforço e aprendizagem não supervisionada. Essas áreas emergentes expandem as capacidades da IA, permitindo que os sistemas aprendam de forma mais autônoma e adaptativa, abrindo novas possibilidades para aplicações inovadoras e impactantes em diversos domínios. A evolução contínua da IA, aliada ao aumento do poder computacional e à disponibilidade de grandes volumes de dados, assegura que as tecnologias inteligentes continuarão a transformar o panorama tecnológico e organizacional nos próximos anos (RUSSELL; NORVIG, 2016; GÉRON, 2019).

2.3 MODELOS DE CLASSIFICAÇÃO

Esta seção apresenta o funcionamento dos métodos de aprendizado supervisionado escolhidos para este trabalho. Ambos resolvem o mesmo tipo de problema (classificação binária, ou seja, separar exemplos em duas classes) mas funcionam de formas distintas. A escolha dos dois modelos permite comparar abordagens diferentes em interpretabilidade e capacidade de capturar relações não-lineares.

2.3.1 Regressão Logística

A regressão logística é um método de classificação que estima a probabilidade de um evento binário acontecer. Ele separa os exemplos em duas categorias, que neste trabalho são

o comparecimento e a ausência do paciente. O modelo recebe uma combinação linear das variáveis de entrada e passa esse valor pela função logística, também chamada de sigmoide, que sempre devolve um número entre 0 e 1. Esse número é interpretado como a probabilidade da classe positiva (BISHOP, 2006).

A probabilidade prevista pelo modelo é dada pela Equação 2.1.

$$P(Y = 1 | X) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \dots + \beta_n X_n)}} \quad (2.1)$$

Nessa equação, Y é a variável-alvo (0 ou 1), $X_1, X_2, \dots, X_n$ são as variáveis de entrada (*features*) e $\beta_0, \beta_1, \dots, \beta_n$ são os coeficientes que o modelo aprende durante o treinamento. Cada coeficiente β_i mede o impacto da *feature* X_i na probabilidade prevista. Quando β_i é positivo, a *feature* aumenta a probabilidade da classe 1. Quando é negativo, ela diminui (BISHOP, 2006).

O treinamento consiste em encontrar o conjunto de coeficientes que melhor ajusta o modelo aos dados de treino. Isso é feito por um algoritmo iterativo, chamado de *solver*, que minimiza uma função de custo conhecida como *log loss*. A biblioteca *scikit-learn* oferece diferentes *solvers*, cada um adequado a um tamanho de dados e a uma estratégia de regularização específica (GÉRON, 2019).

A regressão logística tem três vantagens principais (BISHOP, 2006). A primeira é a interpretabilidade: os coeficientes β_i têm significado direto e indicam quais *features* pesam mais na previsão. A segunda é a saída probabilística, pois o modelo devolve uma probabilidade, e não apenas um rótulo, o que permite ajustar o limiar de decisão conforme a aplicação. A terceira é a eficiência computacional, já que o treino é rápido mesmo em grandes conjuntos de dados.

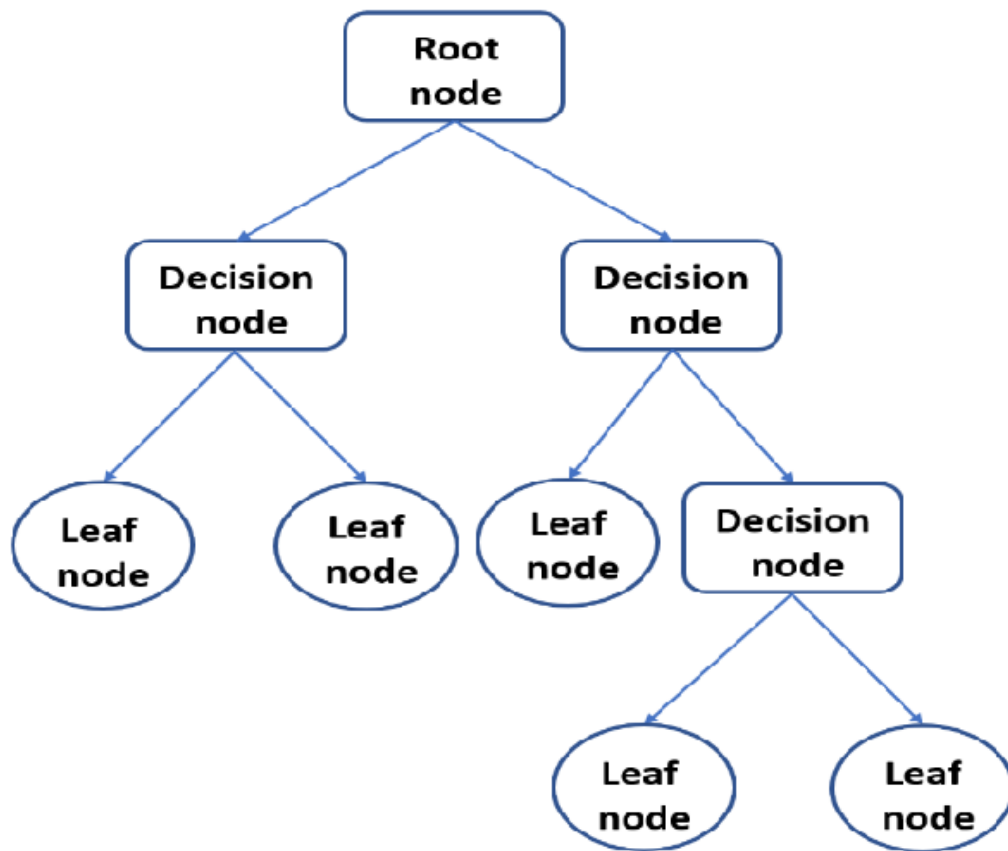
As limitações também são conhecidas. A primeira é a suposição de linearidade: o modelo assume relação linear entre as *features* e o *logit* da probabilidade e, quando a relação real é não-linear, o desempenho fica abaixo do que modelos mais flexíveis atingem. A segunda é a sensibilidade a *features* correlacionadas, pois colunas linearmente dependentes geram coeficientes instáveis, problema conhecido como multicolinearidade. A terceira é a necessidade de pré-processamento, uma vez que variáveis categóricas precisam ser codificadas e variáveis numéricas costumam ser normalizadas antes do treino.

Por ser simples, rápida de treinar e fácil de explicar, a regressão logística costuma servir como modelo de referência em comparações com algoritmos mais complexos. Em aplicações de saúde, finanças e ciências sociais, ela é amplamente utilizada quando a interpretação dos resultados é tão importante quanto a previsão em si (GÉRON, 2019).

2.3.2 Árvore de Decisão

A árvore de decisão é um modelo de classificação baseado em uma estrutura em forma de árvore, ilustrada na Figura 1. Cada nó interno faz uma pergunta sobre uma *feature*, do tipo “ X é maior ou menor que t ?”. Conforme a resposta, o exemplo segue por um dos ramos até chegar a um nó-folha, que contém a previsão da classe (BISHOP, 2006).

Figura 1 – Esquema geral de uma árvore de decisão.



Fonte: https://www.researchgate.net/figure/The-general-scheme-of-a-Decision-Tree-25_fig5_366892408

O treinamento funciona por divisão recursiva. A partir do conjunto de treino completo, o algoritmo avalia todas as *features* e todos os pontos de corte possíveis, e escolhe aquele que melhor separa as classes. O processo se repete em cada subconjunto resultante até atingir um critério de parada, como profundidade máxima da árvore ou ausência de ganho relevante na separação (GÉRON, 2019).

A qualidade de cada corte é medida por uma métrica de impureza, e as duas mais comuns em problemas de classificação são a impureza de Gini e a entropia. A impureza de Gini mede a probabilidade de classificar de forma incorreta um exemplo do nó, caso ele fosse rotulado segundo a distribuição atual de classes naquele nó. Quanto menor o valor, mais puro o nó. A entropia, vinda da Teoria da Informação, mede a desordem da distribuição de classes no nó. Quanto menor o valor, mais homogêneo o nó. Independentemente da métrica usada, o algoritmo procura em cada divisão o corte que mais reduz a impureza, e essa redução é chamada de ganho de informação (GÉRON, 2019).

A árvore de decisão tem três vantagens principais (BISHOP, 2006). A primeira é a interpretabilidade: a estrutura da árvore pode ser visualizada, e cada previsão segue uma sequência clara de regras, o que facilita explicar a decisão. A segunda é a captura de relações não-lineares, pois, como cada nó faz cortes independentes, a árvore consegue modelar interações complexas entre *features* sem precisar defini-las previamente. A terceira é a tolerância

a tipos diferentes de *feature*, já que a árvore opera diretamente sobre variáveis numéricas e categóricas codificadas, sem exigir normalização das escalas.

As principais limitações são duas. A primeira é o *overfitting*: árvores muito profundas memorizam o ruído do conjunto de treino em vez de aprender padrões generalizáveis, razão pela qual é comum limitar a profundidade máxima ou exigir um número mínimo de exemplos por nó. A segunda é a instabilidade, pois pequenas mudanças no conjunto de treino podem gerar árvores com estruturas bem diferentes, o que reduz a confiança na interpretação dos cortes. *Ensembles* como *Random Forest* e *Gradient Boosting* mitigam esse problema combinando muitas árvores, mas saem do escopo deste trabalho (GÉRON, 2019).

Em problemas onde uma das classes é muito mais frequente que a outra, a árvore tende a privilegiar a classe majoritária na escolha dos cortes, ignorando os padrões da minoritária. Há duas formas de contornar isso. A primeira é reamostrar o conjunto de treino, gerando exemplos sintéticos da classe minoritária (*Synthetic Minority Over-sampling Technique*, SMOTE) ou removendo aleatoriamente exemplos da majoritária (*undersampling*). A segunda é a ponderação de classes, que atribui um peso maior aos erros cometidos sobre a classe minoritária ao avaliar cada corte, tornando o treino sensível ao desbalanceamento. As duas estratégias podem ser combinadas. Em ambos os casos, o algoritmo de indução da árvore continua o mesmo. O que muda são os dados ou o critério de custo usados no treino. Como consequência, a árvore aprendida é diferente da que seria obtida sem esses ajustes, o que é justamente o efeito desejado, pois ela passa a representar melhor a classe minoritária (GÉRON, 2019).

2.4 PRÉ-PROCESSAMENTO DE DADOS

Os modelos de aprendizado supervisionado não operam diretamente sobre variáveis categóricas em formato de texto nem lidam bem com variáveis numéricas em escalas muito diferentes. Antes do treino, os dados passam por etapas de preparação que convertem cada variável em uma representação numérica adequada ao modelo. Duas dessas etapas são centrais para os métodos comparados neste trabalho: a codificação de variáveis categóricas e a normalização de variáveis numéricas. A escolha da codificação não é arbitrária, pois depende da forma como o modelo combina as *features*, e por isso modelos lineares e modelos baseados em árvore exigem estratégias distintas.

2.4.1 Codificação de Variáveis Categóricas

Uma variável categórica representa um atributo qualitativo, cujos valores são rótulos (por exemplo, o período do dia de uma consulta: *manhã*, *tarde* ou *noite*). Como os modelos calculam com números, cada categoria precisa ser convertida em uma forma numérica antes do treino. Duas estratégias são amplamente usadas para essa conversão: o *One-Hot Encoding* e o *Label Encoding*. As duas produzem representações com formato e suposições diferentes, e

cada uma é apropriada a uma família de modelos. O *One-Hot Encoding* é adequado a modelos lineares, como a regressão logística, enquanto o *Label Encoding* é adequado a modelos baseados em árvore, como a árvore de decisão. As subseções a seguir descrevem cada estratégia e, em seguida, a Seção 2.4.4 compara as duas e explica por que não é viável trocá-las entre os modelos.

2.4.2 One-Hot Encoding

No *One-Hot Encoding*, cada categoria de uma variável passa a ser representada por uma coluna binária independente, que assume o valor 1 quando a categoria está presente na linha e 0 caso contrário. Uma variável com N categorias gera, a princípio, N colunas binárias. Para cada variável, porém, uma das categorias é descartada e tratada como categoria de referência, o que evita a redundância entre as colunas resultantes (GÉRON, 2019).

Essa codificação é coerente com a natureza linear da regressão logística. O modelo calcula uma soma ponderada das *features* e aplica a função logística sobre o resultado. Cada coluna binária recebe um coeficiente próprio, então o efeito de cada categoria sobre a probabilidade prevista é estimado de forma independente, sem suposições sobre a relação entre categorias.

Há um motivo direto para descartar uma categoria. Uma variável categórica com N categorias, codificada sem esse descarte, gera N colunas binárias que somam 1 em toda linha (exatamente uma vale 1 e as outras valem 0). Isso significa que qualquer uma das colunas pode ser deduzida das outras, ou seja, não acrescenta informação nova ao conjunto. Esse problema de colunas linearmente dependentes é chamado de multicolinearidade.

A regressão logística estima seus coeficientes resolvendo um sistema de equações que envolve inverter uma matriz construída a partir das *features*. Quando há colunas redundantes, essa matriz fica próxima de singular: os coeficientes ficam instáveis (valores muito altos, sinais sem sentido) e, em casos extremos, o algoritmo de otimização não converge. Descartar uma categoria por variável resolve o problema. A categoria descartada vira a referência, e os coeficientes das colunas restantes passam a representar o efeito de cada categoria em comparação a essa categoria-base. O ajuste é padrão em modelos lineares e não se aplica a árvores de decisão, que dividem o espaço por cortes em vez de combinar *features* linearmente (GÉRON, 2019).

A Tabela 1 ilustra o mecanismo para a variável *periodo_dia*, com três categorias (*manha*, *tarde* e *noite*), adotando *manha* como categoria de referência. Sem o descarte, haveria uma terceira coluna *periodo_dia_manha*, totalmente determinada pelas outras duas (a soma das três colunas seria sempre 1), o que caracteriza a multicolinearidade. Com *manha* como referência, a linha correspondente a essa categoria apresenta zeros em todas as colunas, configuração que a distingue das demais e permite ao modelo estimar coeficientes válidos.

Tabela 1 – *One-Hot Encoding* da variável *periodo_dia* com *manha* como categoria de referência.

Categoria original	<i>periodo_dia_tarde</i>	<i>periodo_dia_noite</i>
<i>manha</i> (referência)	0	0
<i>tarde</i>	1	0
<i>noite</i>	0	1

2.4.3 Label Encoding

No *Label Encoding*, cada categoria recebe um inteiro distinto ($0, 1, 2, \dots$), mantendo a variável em uma única coluna numérica. Diferentemente do *One-Hot*, a dimensionalidade não cresce com o número de categorias, pois a variável permanece em uma só coluna. A Tabela 2 ilustra essa codificação para a variável *periodo_dia*, a mesma usada no exemplo de *One-Hot* da Seção 2.4.2: as três categorias passam a ocupar uma única coluna de inteiros.

Tabela 2 – *Label Encoding* da variável *periodo_dia*.

Categoria original	<i>periodo_dia</i> (codificada)
<i>manha</i>	0
<i>tarde</i>	1
<i>noite</i>	2

Essa codificação é coerente com a forma como a árvore decide cortes. A árvore de decisão não combina os valores das *features* por soma ou produto. Ela divide o espaço de dados por meio de comparações do tipo “a *feature X* é maior ou menor que o limiar t ?”. Como o algoritmo é capaz de escolher qualquer limiar entre dois valores quaisquer, a ordem específica dos inteiros atribuídos às categorias não compromete a qualidade das divisões: a árvore identifica naturalmente os pontos de corte que separam melhor a variável-alvo (GÉRON, 2019).

A atribuição dos inteiros é determinística e definida a partir do conjunto de treino. O mesmo mapeamento é reaplicado em validação e teste, garantindo que cada categoria receba sempre o mesmo inteiro em todos os conjuntos e evitando vazamento de informação.

2.4.4 Diferenças entre One-Hot e Label Encoding

As duas codificações produzem representações com formato e suposições muito diferentes. Aplicar a codificação errada a um modelo não gera um erro de execução, mas degrada o desempenho ou viola pressupostos (GÉRON, 2019). As diferenças concentram-se em quatro aspectos. Quanto à saída, o *One-Hot Encoding* gera uma coluna binária por categoria, enquanto o *Label Encoding* mantém a variável em uma única coluna inteira. Como consequência, a dimensionalidade do *One-Hot* cresce com o número de categorias, ao passo que a do *Label Encoding*

permanece constante, com apenas uma coluna por variável. Quanto à suposição de ordem, o *One-Hot* não impõe nenhuma, pois trata as categorias como independentes, enquanto o *Label Encoding* embute uma ordem implícita, dada pela sequência dos inteiros. Por essas razões, cada codificação é apropriada a uma família de modelos: o *One-Hot Encoding* aos modelos lineares (como a regressão logística), e o *Label Encoding* aos modelos baseados em árvore. A Tabela 3 resume o que acontece quando se troca a codificação entre os modelos.

O *Label Encoding* não é adequado para a regressão logística, que estima um único coeficiente para cada *feature* numérica. Se uma variável como *periodo_dia* fosse codificada como inteiros (*manha*=0, *tarde*=1, *noite*=2), o modelo aprenderia um único peso β que multiplicaria esse valor. Isso impõe uma escala linear artificial: o efeito de *noite* seria forçosamente o dobro do efeito de *tarde*, e ambos seriam alinhados em uma reta a partir de *manha*. Como a ordem dos inteiros é arbitrária (depende apenas da ordem em que as categorias foram codificadas), essa restrição não tem fundamento nos dados. O resultado é um modelo com coeficientes enviesados e poder preditivo reduzido. O *One-Hot Encoding* contorna o problema dando um coeficiente independente para cada categoria.

O *One-Hot Encoding* não é adequado para a árvore de decisão, que poderia tecnicamente operar sobre colunas *One-Hot*, mas com vários efeitos colaterais. Cada coluna binária só permite o corte trivial $X < 0,5$, ou seja, separa uma única categoria do resto. Para reproduzir um agrupamento como “*manha* ou *tarde*” versus “*noite*”, a árvore precisa de dois nós em vez de um, gerando estruturas mais profundas e fragmentadas. Quando a variável tem muitas categorias, como o bairro de residência do paciente, o *One-Hot* multiplica o número de *features* e dilui o sinal: cada coluna individual cobre poucos exemplos, o que reduz a estabilidade do ganho de informação e tende a privilegiar variáveis numéricas no critério de divisão. Em conjunto, isso piora a qualidade dos cortes, aumenta o tempo de treino e torna a árvore mais difícil de interpretar. O *Label Encoding* resolve essa rigidez ao manter a variável em uma única coluna inteira, deixando a árvore livre para escolher qualquer limiar.

Tabela 3 – Consequências de trocar a codificação entre os modelos.

Combinação inadequada	Problema
<i>Label Encoding</i> na regressão logística	Impõe uma escala linear artificial entre categorias (<i>noite</i> valeria o dobro de <i>tarde</i>), com coeficientes enviesados e poder preditivo reduzido.
<i>One-Hot Encoding</i> na árvore de decisão	Restringe cada <i>split</i> ao corte trivial $X < 0,5$, gerando árvores mais profundas e fragmentadas. Com variáveis de muitas categorias, dilui o sinal e desestabiliza o ganho de informação.

2.4.5 Normalização de Variáveis Numéricas

A normalização padroniza as *features* numéricas para que todas estejam na mesma escala. Sem ela, *features* com valores grandes (como a idade do paciente, que vai de 0 a 100) dominam o treinamento em comparação com *features* de valores pequenos (como um indicador

binário, que vale 0 ou 1). Uma técnica comum é a padronização (*z-score*), que transforma cada coluna conforme a Equação 2.2 (GÉRON, 2019).

$$z = \frac{x - \mu}{\sigma} \quad (2.2)$$

Nessa equação, x é o valor original, μ é a média da coluna e σ é o desvio padrão. O resultado z tem média 0 e desvio padrão 1 (GÉRON, 2019). A transformação não altera a forma da distribuição de cada variável, apenas a recentra e a reescala. Assim, variáveis originalmente em escalas muito distintas (por exemplo, uma idade que varia de 0 a 100 e um tempo de espera de poucos dias) passam a ocupar a mesma faixa de valores depois da padronização, que é justamente o objetivo da técnica.

A padronização interessa sobretudo a modelos lineares. A regressão logística calcula uma soma ponderada das *features*, e o peso (coeficiente β_i) de cada *feature* depende da escala da variável de entrada. Se uma *feature* tem valores muito maiores que outra, seu coeficiente fica artificialmente pequeno para compensar, o que dificulta a otimização e atrapalha a interpretação dos coeficientes (GÉRON, 2019). A árvore de decisão, por outro lado, não precisa de normalização. Como os cortes são baseados em limiares (“a *feature* é maior que t ?”), a magnitude absoluta da *feature* não muda a decisão do algoritmo.

Em qualquer caso, a padronização é ajustada apenas no conjunto de treino. A mesma média e desvio padrão calculados ali são reaplicados aos conjuntos de validação e teste, sem reajuste. Esse cuidado evita o vazamento de informação, pois, se a média do teste influenciasse o ajuste, o modelo estaria indiretamente vendo dados do teste durante o treino, o que inflaria artificialmente o desempenho final (GÉRON, 2019).

2.5 MÉTRICAS

A avaliação de modelos de classificação é feita por meio de métricas que medem o quanto as previsões se aproximam das classes reais. Cada métrica olha para um aspecto diferente do desempenho, e a escolha da métrica adequada depende do contexto do problema. Em conjuntos com classes desbalanceadas, como o conjunto de dados de absentismo utilizado neste trabalho, em que cerca de 30% dos agendamentos são faltas e 70% comparecimentos (Tabela 6), algumas métricas podem dar uma impressão enganosa de bom desempenho. Esta seção apresenta as cinco métricas usadas neste trabalho.

Todas elas são derivadas dos quatro resultados possíveis de uma classificação binária. Um verdadeiro positivo (VP) ocorre quando o modelo prevê a classe 1 e a classe real é 1, e um verdadeiro negativo (VN), quando prevê a classe 0 e a classe real é 0. Já um falso positivo (FP) ocorre quando o modelo prevê a classe 1, mas a classe real é 0, e um falso negativo (FN), quando prevê a classe 0, mas a classe real é 1.

2.5.1 Acurácia

A acurácia mede a fração de previsões corretas sobre o total de previsões feitas. É a métrica mais intuitiva e funciona bem em problemas balanceados.

$$\text{Acurácia} = \frac{VP + VN}{VP + VN + FP + FN} \quad (2.3)$$

Em conjuntos desbalanceados, a acurácia pode enganar. Em um problema com 90% de exemplos de uma classe (por exemplo, comparecimentos) e 10% da outra (faltas), um modelo que sempre prevê a classe majoritária, ou seja, comparecimento, atinge 90% de acurácia sem aprender nada útil. Por isso, na previsão de absenteísmo, a acurácia precisa ser interpretada junto com outras métricas mais sensíveis à classe minoritária (GÉRON, 2019).

2.5.2 Precisão

A precisão mede, entre os exemplos classificados como positivos, quantos eram de fato positivos. Ou seja, dentre os pacientes que o modelo aponta como prováveis faltas, ela indica com que frequência ele acerta.

$$\text{Precisão} = \frac{VP}{VP + FP} \quad (2.4)$$

Precisão alta significa que o modelo gera poucos falsos alarmes. Em aplicações onde acionar um falso positivo tem custo, por exemplo ligar para um paciente que vai comparecer, a precisão é uma métrica importante (GÉRON, 2019).

2.5.3 Recall

O *recall*, também chamado de sensibilidade, mede, entre os exemplos que de fato pertencem à classe positiva, quantos o modelo conseguiu identificar. Ou seja, de todos os pacientes que de fato vão faltar, ele indica quantos o modelo consegue prever.

$$\text{Recall} = \frac{VP}{VP + FN} \quad (2.5)$$

Recall alto significa que o modelo deixa passar poucos casos da classe positiva. A escolha entre priorizar o *recall* ou a precisão depende do custo relativo de cada tipo de erro: quando deixar de identificar um exemplo positivo (falso negativo) é mais custoso que um falso alarme (falso positivo), prioriza-se o *recall*. No caso contrário, prioriza-se a precisão (GÉRON, 2019).

2.5.4 F1-Score

O *F1-score* combina precisão e *recall* em um único valor, calculado como a média harmônica das duas métricas.

$$F1 = 2 \cdot \frac{\text{Precisão} \cdot \text{Recall}}{\text{Precisão} + \text{Recall}} \quad (2.6)$$

O *F1-score* só é alto quando precisão e *recall* são altas ao mesmo tempo. A média harmônica penaliza valores muito desbalanceados. Por exemplo, precisão de 0,9 e *recall* de 0,1 dão F1 de aproximadamente 0,18, enquanto a média aritmética seria 0,5. Por isso, o *F1-score* é útil quando se quer um único indicador que reflita o equilíbrio entre precisão e *recall* (GÉRON, 2019).

2.5.5 AUC

A AUC é a área sob a curva ROC (*Receiver Operating Characteristic*). A curva ROC mostra como o modelo se comporta em diferentes limiares de decisão, traçando o *recall*, ou taxa de verdadeiros positivos (TPR), contra a taxa de falsos positivos (FPR). A AUC resume essa curva em um único valor entre 0 e 1, calculado como a área sob ela (Equação 2.7).

$$AUC = \int_0^1 TPR d(FPR) \quad (2.7)$$

Uma AUC de 0,5 indica desempenho equivalente ao acaso, ou seja, o modelo não distingue as classes. Uma AUC de 1 indica separação perfeita. Valores entre 0,7 e 0,9 são considerados bons (GÉRON, 2019). A principal vantagem da AUC é ser robusta ao desbalanceamento e independente do limiar de decisão. Ela mede a capacidade do modelo de separar as classes, atribuindo probabilidade prevista mais alta aos exemplos positivos do que aos negativos (GÉRON, 2019).

2.6 TRABALHOS RELACIONADOS

A predição de absenteísmo por aprendizado de máquina já foi objeto de revisões sistemáticas. Carreras-García et al. (2020) analisaram 50 estudos e observaram desempenho tipicamente moderado, com AUC concentrada entre 0,70 e 0,86 e apenas um trabalho acima de 0,90. A revisão também alertou que a maioria dos estudos que reportou acurácia não superava a taxa de comparecimento de base, o que reforça o quanto essa métrica engana em dados desbalanceados. Os melhores desempenhos, quando ocorrem, dependem de engenharia de atributos específica de cada serviço, o que limita a reprodução em outros contextos, e os preditores mais estáveis entre os estudos são poucos: o histórico de faltas, o tempo de antecedência do agendamento, a idade e a distância até o serviço.

Quanto aos fatores associados à falta, Dantas et al. (2018) revisaram a literatura sobre o não comparecimento a consultas e apontaram como determinantes mais consistentes o tempo de antecedência do agendamento, o histórico de faltas anteriores, a idade e a condição socioeconômica. Do lado das intervenções, a revisão Cochrane de Gurol-Urganci et al. (2013) mostrou que lembretes por mensagem aumentam o comparecimento, embora com ganho pequeno e de forma indiscriminada, sem distinguir quem de fato tende a faltar.

No contexto brasileiro, Beltrame et al. (2019) mediram o impacto do absenteísmo como desperdício em um sistema público de saúde, com taxas de 38,6% em consultas especializadas no Espírito Santo. Para além do absenteísmo, o aprendizado de máquina tem sido aplicado a outros problemas de predição em saúde, como a mortalidade infantil (MBUNGE et al., 2023; SARAVANOU et al., 2021), o que evidencia a tendência de abordagens orientadas a dados no setor.

Este trabalho se distingue dessa literatura menos pelo desempenho e mais pela abordagem. Em vez de perseguir o maior AUC possível com engenharia específica de um serviço, que é justamente o que dificulta a reprodução em outros contextos, ele compara dois modelos clássicos e interpretáveis sob um protocolo amplo e transparente, com 243 versões do conjunto de dados e mais de onze mil experimentos, e reporta de forma honesta o desempenho modesto obtido e os seus limites. A contribuição está em produzir uma comparação reprodutível, e não um modelo de alto desempenho preso ao contexto em que foi treinado.

3 METODOLOGIA

Nesta seção, são fornecidos detalhes sobre o conjunto de dados utilizado em nesse trabalho e sobre os algoritmos e parâmetros escolhidos.

3.1 CONJUNTO DE DADOS

Ao desenvolver ferramentas baseadas em aprendizado de máquina, a qualidade dos dados é crucial, pois está diretamente correlacionada com a confiabilidade e eficácia dos sistemas construídos. Para este estudo, utilizamos um conjunto de dados abrangente que contém informações detalhadas sobre 110.527 agendamentos de consultas médicas, com o objetivo principal de prever a ausência de pacientes (*no-shows*) às consultas agendadas. A Tabela 4 apresenta uma amostra desse conjunto de dados.

Tabela 4 – Amostra de cinco agendamentos do conjunto de dados (subconjunto das variáveis, com a lista completa na Tabela 5).

<i>Gender</i>	<i>Scheduled Day</i>	<i>Appointment Day</i>	<i>Age</i>	<i>Neighbourhood</i>	<i>Scholarship</i>	<i>Hipertension</i>	<i>SMS_received</i>	<i>No-show</i>
M	2016-04-26	2016-05-06	50	SANTO ANDRÉ	0	0	1	Yes
M	2016-04-06	2016-05-09	66	GURIGICA	0	1	0	Yes
F	2016-04-18	2016-05-30	35	MARIA ORTIZ	1	0	1	No
F	2016-05-03	2016-05-13	4	SÃO JOSÉ	0	0	0	No
M	2016-05-20	2016-05-30	70	MARIA ORTIZ	0	1	1	No

Fonte: elaborado pelo autor, a partir do conjunto de dados.

3.1.1 Descrição do Conjunto de Dados

O conjunto de dados utilizado neste trabalho é uma base pública de agendamentos de consultas médicas coletada em Vitória, no Espírito Santo, em 2016 (HOPPEN, 2016), que registra informações detalhadas sobre cada agendamento. Cada amostra do conjunto de dados contém 14 variáveis que descrevem diferentes aspectos do paciente, do agendamento e do contexto da consulta. A variável principal de interesse é a *No-show*, que indica se o paciente compareceu ou não à consulta agendada.

3.1.2 Dicionário de Dados

A Tabela 5 descreve detalhadamente as 14 variáveis presentes no conjunto de dados.

Tabela 5 – Dicionário de variáveis do conjunto de dados.

Nome da Variável	Descrição
<i>PatientId</i>	Identificação única de cada paciente.
<i>AppointmentID</i>	Identificação única de cada agendamento de consulta.
<i>Gender</i>	Gênero do paciente (Masculino ou Feminino). A maior proporção é de mulheres.
<i>AppointmentDay</i>	Data exata da consulta agendada, quando o paciente deve visitar o médico.
<i>ScheduledDay</i>	Data em que a consulta foi agendada, antecedendo a data da consulta propriamente dita.
<i>Age</i>	Idade do paciente.
<i>Neighbourhood</i>	Bairro onde a consulta ocorre.
<i>Scholarship</i>	Indicador se o paciente possui bolsa família (<i>True/False</i>). Este conceito está relacionado ao programa Bolsa Família, que fornece apoio financeiro a famílias em situação de vulnerabilidade social.
<i>Hipertension</i>	Indicador se o paciente possui hipertensão (<i>True/False</i>).
<i>Diabetes</i>	Indicador se o paciente possui diabetes (<i>True/False</i>).
<i>Alcoholism</i>	Indicador se o paciente possui alcoolismo (<i>True/False</i>).
<i>Handcap</i>	Indicador se o paciente possui alguma deficiência (<i>True/False</i>).
<i>SMS_received</i>	Indica se o paciente recebeu mensagem de lembrete por SMS da consulta (0 = não recebeu, 1 = recebeu).
<i>No-show</i>	Indicador se o paciente não compareceu à consulta agendada (<i>True/False</i>).

A Tabela 6 apresenta a distribuição das instâncias entre as classes *No-show* e *Show-up* no conjunto de dados utilizado.

Tabela 6 – Distribuição das classes no conjunto de dados.

Classe	Número de instâncias
<i>No-show</i>	33.158
<i>Show-up</i>	77.369
Total	110.527

A distribuição revela um conjunto de dados desbalanceado: as faltas (*No-show*) representam cerca de 30% das instâncias, contra aproximadamente 70% de comparecimentos (*Show-up*). Esse desbalanceamento é relevante porque a classe minoritária é justamente o alvo de interesse, ou seja, prever quem vai faltar. Em cenários assim, um modelo que simplesmente previsse o comparecimento de todos acertaria em torno de 70% dos casos sem nenhuma utilidade prática, o que torna a acurácia uma métrica enganosa. Por isso, a avaliação dos modelos prioriza métricas robustas ao desbalanceamento, como a AUC (Seção 2.5.5), e justifica as estratégias de balanceamento empregadas no pré-processamento.

3.1.3 Conjunto de treino, conjunto de validação e conjunto de teste

O conjunto de dados foi particionado em três subconjuntos com finalidades distintas no processo de modelagem: treino, validação e teste, cuja separação rigorosa é fundamental para que a estimativa final de desempenho não seja enviesada.

O conjunto de treino é utilizado exclusivamente para o ajuste dos modelos, ou seja, a estimação dos coeficientes da regressão logística e a definição das divisões da árvore de decisão. Ele não participa da comparação entre configurações nem da estimativa de desempenho final.

O conjunto de validação destina-se à comparação entre as diferentes configurações de hiperparâmetros e à seleção da mais adequada. Os modelos não são treinados sobre ele, apenas avaliados, o que orienta a escolha da configuração final. Esse subconjunto não é utilizado para treinamento nem para reportar o desempenho final do trabalho.

O conjunto de teste é reservado à avaliação final. Ele foi separado no início do processo, com 28.001 amostras, e não participou de nenhuma etapa de treinamento ou de seleção de modelos. Por não ter sido observado nas etapas anteriores, constitui um conjunto de dados inéditos, sobre o qual o desempenho é medido uma única vez, ao final, para a configuração previamente selecionada.

A divisão foi conduzida em duas etapas. Primeiro, o conjunto de teste, com 28.001 amostras, foi separado e mantido isolado das demais etapas. As 81.633 amostras restantes foram submetidas a uma segunda divisão, na proporção 80/20, com estratificação pela classe-alvo, resultando em aproximadamente 65.306 amostras de treino e 16.327 de validação. Dessa forma, treino e validação derivam do mesmo bloco de 81.633 amostras, enquanto o conjunto de teste é independente, por ter sido separado anteriormente. A Tabela 7 resume os três subconjuntos.

Tabela 7 – Divisão dos dados em treino, validação e teste.

Conjunto	Amostras	Proporção	Finalidade
Treino	65.306	60%	Ajuste dos modelos (coeficientes e divisões da árvore)
Validação	16.327	15%	Comparação de configurações e seleção da melhor, sem treinamento
Teste	28.001	25%	Avaliação final do desempenho, uma única vez, sobre dados inéditos

A reserva do conjunto de teste durante toda a etapa de comparação é o que confere confiabilidade à estimativa final de desempenho. Caso ele também fosse utilizado na seleção dos modelos, deixaria de representar dados inéditos e o desempenho reportado tenderia a ser superestimado. Por ser empregado apenas na avaliação final, a estimativa obtida aproxima-se do desempenho esperado sobre dados não observados. Ao longo do restante deste trabalho, o desempenho de teste refere-se sempre a esse conjunto.

3.1.4 Limpeza de Dados

O *pipeline* inclui uma camada de imputação para tratar eventuais valores faltantes nas colunas usadas pelos modelos. Para variáveis categóricas, como *Gender* e *Neighbourhood*, a imputação é feita pela moda. Para variáveis numéricas, como *Age* e *SMS_received*, é feita pela mediana. Essa etapa garante que o *pipeline* continue funcional mesmo diante de lacunas pontuais em novas versões do conjunto de dados. A Tabela 8 resume os tratamentos aplicados a cada uma das 14 variáveis do conjunto de dados.

Tabela 8 – Tratamento aplicado a cada variável do conjunto de dados.

Variável	Tipo	Tratamento
<i>PatientId</i> , <i>AppointmentID</i>	Identificador	Removidos do pipeline, pois não são utilizados como <i>features</i> .
<i>Gender</i>	Categórica	Imputação defensiva pela moda e codificação manual $M \rightarrow 0$, $F \rightarrow 1$.
<i>ScheduledDay</i>	Data	Convertida para <i>datetime</i> e usada como origem das variáveis derivadas <i>WaitingDays</i> , <i>scheduled_hour</i> e <i>periodo_dia</i> (Seção 3.1.5). A coluna original é removida ao final do <i>pipeline</i> .
<i>AppointmentDay</i>	Data	Convertida para <i>datetime</i> e usada como origem das variáveis derivadas <i>WaitingDays</i> e <i>weekday</i> (Seção 3.1.5). A coluna original é removida ao final do <i>pipeline</i> .
<i>Age</i>	Numérica	Imputação defensiva pela mediana e categorização em <i>faixa_idade</i> (Seção 3.1.6), mantendo-se também o valor numérico como <i>feature</i> .
<i>Neighbourhood</i>	Categórica	Os bairros abaixo do limiar de consultas são agrupados em <i>OUTROS_BAIRROS</i> (Seção 3.1.9) e a variável é codificada por <i>One-Hot Encoding</i> .
<i>Scholarship</i> , <i>Hipertension</i> , <i>Diabetes</i> , <i>Alcoholism</i> , <i>Handcap</i>	Binária	Imputação defensiva pela moda, mantidas como 0/1.
<i>SMS_received</i>	Numérica	Imputação defensiva pela mediana, mantida como valor numérico.
<i>No-show</i>	Alvo	Conversão $Yes \rightarrow 1$, $No \rightarrow 0$. É a variável-alvo do problema de classificação.

Além da imputação, o *pipeline* aplica um filtro sobre o tempo de espera. A variável *WaitingDays* é a diferença em dias entre a data da consulta e a data do agendamento (ver Seção 3.1.5). No conjunto de dados original, a data da consulta é registrada à meia-noite, enquanto a do agendamento traz a hora cheia, de modo que uma consulta marcada para o mesmo dia produz um tempo de espera negativo. Esses agendamentos de mesmo dia são removidos pelo filtro $WaitingDays \geq 0$.

Esse descarte ocorre durante o pré-processamento, ou seja, depois do particionamento dos dados (Tabela 7). O conjunto de teste foi separado antes de qualquer filtragem e é processado de forma independente, sem contato com os conjuntos de treino e validação, o que

evita vazamento de informação. O filtro é então aplicado a cada subconjunto separadamente: ao bloco de treino e validação, antes da divisão de 80/20, e ao conjunto de teste, na etapa de avaliação. Como os agendamentos de mesmo dia se distribuem de maneira semelhante entre os conjuntos, os três são reduzidos em proporção parecida, cerca de 35%: o treino passa de 65.306 para aproximadamente 42.212 instâncias, a validação de 16.327 para cerca de 10.553 e o teste de 28.001 para 18.593. A avaliação final, portanto, é feita apenas sobre os agendamentos com tempo de espera não negativo.

A Tabela 9 resume o tamanho e a proporção de faltas de cada conjunto após o pré-processamento. As três partições mantêm uma proporção de faltas semelhante, em torno de 28%, o que confirma que a estratificação pela classe-alvo foi preservada e faz com que a validação funcione como uma prévia confiável do desempenho no teste. Os tamanhos de treino e validação indicados referem-se às versões sem balanceamento, já que estratégias como *SMOTE* e *undersampling* alteram apenas o conjunto de treino (Seção 3.1.8).

Tabela 9 – Tamanho e proporção de faltas de cada conjunto, após o pré-processamento.

Conjunto	Agendamentos	Faltas
Treino	42.212	28,7%
Validação	10.553	28,7%
Teste	18.593	28,0%

3.1.5 Variáveis derivadas

Algumas das *features* usadas pelos modelos não vêm direto do conjunto de dados original, mas são calculadas a partir das colunas existentes. O objetivo é extrair sinais temporais que ficam escondidos dentro das datas brutas, e que costumam ser preditivos do comparecimento. A Tabela 10 resume as quatro variáveis derivadas geradas no *pipeline*.

Tabela 10 – Variáveis derivadas a partir das colunas originais.

Variável	Origem	Descrição
<i>WaitingDays</i>	<i>AppointmentDay</i> — <i>ScheduledDay</i>	Número de dias entre o agendamento e a data da consulta. Entra como valor numérico e também alimenta a categorização em <i>faixa_espera</i> (ver Seção 3.1.7).
<i>weekday</i>	<i>AppointmentDay.day_name()</i>	Dia da semana em que a consulta está marcada (<i>Monday</i> , <i>Tuesday</i> , ...). Captura padrões de comparecimento por dia da semana.
<i>scheduled_hour</i>	<i>ScheduledDay.hour</i>	Hora do dia (0 a 23) em que o agendamento foi feito. Entra como valor numérico nos dois modelos.
<i>periodo_dia</i>	<i>scheduled_hour</i>	Categorização grosseira de <i>scheduled_hour</i> em três faixas: <i>manha</i> (hora < 12), <i>tarde</i> (12 ≤ hora < 18) e <i>noite</i> (hora ≥ 18). Versão categórica do mesmo sinal de horário, mais robusta a ruído.

Todas essas variáveis derivadas entram como *features* nos modelos. A forma como cada uma é codificada (*One-Hot Encoding* na regressão logística, *LabelEncoder* na árvore de decisão) depende do tipo de modelo e está detalhada na Seção 2.4.1.

3.1.6 Categorização por idade

A idade do paciente foi tratada não só como número, mas também como faixa categórica. A motivação é que o efeito da idade sobre o comparecimento não é necessariamente linear. Uma criança pequena depende do responsável para ir ao posto, um adulto jovem tende a esquecer mais, e um idoso costuma ter rotina mais regular de cuidado com a saúde. Cortar a idade em faixas captura esse tipo de diferença sem assumir que a relação é uma reta.

Três níveis de granularidade foram testados, resumidos na Tabela 11.

Tabela 11 – Granularidades testadas para a categorização da idade.

Granularidade	Grupos	Nome da faixa	Intervalo de idade
Básica	3	criança	0–12 anos
		adulto	13–60 anos
		idoso	61+ anos
Detalhada	5	criança	0–12 anos
		jovem	13–19 anos
		adulto	20–35 anos
		meia-idade	36–60 anos
		idoso	61+ anos
Simples	2	jovem	0–35 anos
		adulto	36+ anos

A versão básica é o ponto de partida. A detalhada testa se separar jovens de meia-idade traz ganho. A simples testa o oposto: se um corte mínimo já é suficiente. Cada uma dessas três opções vira uma versão diferente do conjunto de dados, e o desempenho dos modelos é comparado entre elas.

3.1.7 Categorização por faixa de espera

A variável *WaitingDays* foi calculada como a diferença em dias entre a data do agendamento e a data da consulta. Quanto maior esse intervalo, maior é a chance de o paciente esquecer, mudar de planos ou simplesmente perder o interesse. Assim como na idade, esse efeito não é linear: faltar a uma consulta marcada para o mesmo dia é raro, mas faltar a uma marcada com mais de um mês de antecedência já é comum.

Três níveis de granularidade foram testados, resumidos na Tabela 12.

Tabela 12 – Granularidades testadas para a categorização da faixa de espera.

Granularidade	Grupos	Nome da faixa	Intervalo de espera
Básica	3	mesmo dia	0 dias
		até 7	1–7 dias
		acima de 7	8+ dias
Detalhada	7	mesmo dia	0 dias
		até 3	1–3 dias
		até 7	4–7 dias
		até 14	8–14 dias
		até 30	15–30 dias
		até 60	31–60 dias
Simples	2	acima de 60	61+ dias
		mesmo dia	0 dias
		até 14	1–14 dias

A lógica é a mesma da categorização por idade. Cada granularidade gera uma versão diferente do conjunto de dados, e o efeito sobre o desempenho dos modelos é comparado.

3.1.8 Balanceamento de classes

As classes do problema são desbalanceadas: cerca de 70% dos agendamentos resultam em comparecimento e apenas 30% em falta. Esse desbalanceamento pode levar o modelo a aprender a prever quase sempre a classe majoritária, ignorando os casos de falta, que são justamente os que interessam. Para testar o impacto disso, três estratégias foram comparadas, resumidas na Tabela 13.

Tabela 13 – Estratégias de balanceamento testadas.

Estratégia	Descrição
Sem balanceamento	Mantém a distribuição original do conjunto de dados. Serve como linha de base.
<i>SMOTE</i>	<i>Synthetic Minority Over-sampling Technique</i> . Gera amostras sintéticas da classe minoritária por interpolação entre vizinhos.
<i>Random Undersampler</i>	Remove aleatoriamente amostras da classe majoritária até que as duas classes tenham o mesmo tamanho.

Cada estratégia gera uma versão diferente do conjunto de dados de treino. O conjunto de validação e o conjunto de teste não são tocados pelo balanceamento, para que a avaliação seja feita sobre a distribuição real.

3.1.9 Limiar de bairros

O conjunto de dados contém pacientes de diversos bairros da cidade, mas a distribuição entre bairros é muito desigual. Alguns bairros aparecem em milhares de consultas, enquanto ou-

tros aparecem em apenas dezenas. Tratar todos como categorias individuais cria um problema: bairros raros geram colunas com muito ruído e pouca informação útil para o modelo.

Para lidar com isso, foi definido um limite mínimo de consultas para que um bairro entre como categoria própria. Bairros com menos consultas que o limite são agrupados em uma categoria geral chamada *OUTROS_BAIRROS*. Três limites foram testados, resumidos na Tabela 14.

Tabela 14 – Limiares mínimos testados para o agrupamento de bairros.

Limite	Mínimo	Efeito
Permissivo	≥ 100 consultas	Mais bairros entram como categoria. O modelo vê mais granularidade geográfica, mas com mais ruído.
Intermediário	≥ 300 consultas	Equilíbrio entre granularidade e ruído.
Restritivo	≥ 500 consultas	Apenas os bairros mais populares entram. Modelo mais simples, mas perde informação geográfica fina.

Cada limite gera uma versão diferente do conjunto de dados, pois altera quantos bairros são mantidos como categorias próprias e quantos são absorvidos por *OUTROS_BAIRROS*. Um limite mais permissivo preserva a granularidade geográfica, ao custo de mais colunas esparsas e ruidosas após a codificação por *One-Hot Encoding*. Um limite mais restritivo reduz a dimensionalidade e o ruído, mas descarta distinções entre os bairros menos frequentes. Como não há um valor ótimo conhecido de antemão, as três versões são tratadas como opções de pré-processamento e combinadas com as demais para compor o conjunto de experimentos (Seção 3.1.11), de modo que o efeito de cada limite sobre o desempenho dos modelos é avaliado empiricamente.

3.1.10 Features de interação

Variáveis isoladas capturam um sinal, mas a interação entre elas pode capturar outro. Por exemplo, receber um SMS pode ter um efeito diferente para um paciente que recebe Bolsa Família do que para um que não recebe. Para testar esse tipo de efeito, foram criadas *features* de interação. Três conjuntos foram testados, resumidos na Tabela 15.

Tabela 15 – Conjuntos de *features* de interação testados.

Conjunto	Quantidade	Features
Básico	2	<i>sms_bolsa</i> (recebeu SMS e tem Bolsa Família) e <i>idade_espera</i> (idade $\times$ dias de espera).
+ doenças	3	As duas anteriores mais <i>multi_doenca</i> (tem mais de uma doença simultânea).
+ sms $\times$ espera	4	As três anteriores mais <i>sms_espera</i> (SMS $\times$ dias de espera).

Cada conjunto define quantas e quais *features* extras entram no modelo. Conjuntos maiores permitem capturar relações mais complexas entre as variáveis, mas aumentam a di-

mensionalidade e o risco de sobreajuste, sobretudo nas interações pouco frequentes. Como não há garantia, de antemão, de que essas *features* melhorem a previsão, os três conjuntos são tratados como opções de pré-processamento e combinados com as demais para compor o conjunto de experimentos (Seção 3.1.11), de modo que o ganho efetivo de cada um é avaliado empiricamente.

3.1.11 Geração dos 243 conjuntos de dados base

Reunindo as cinco decisões anteriores, cada uma com três opções, chega-se a $3 \times 3 \times 3 \times 3 \times 3 = 243$ conjuntos de dados distintos de pré-processamento.

A Tabela 16 resume os cinco fatores e suas opções.

Tabela 16 – Combinação dos cinco fatores de pré-processamento.

Fator	Opção 1	Opção 2	Opção 3
Balanceamento	Sem	<i>SMOTE</i>	<i>Undersampler</i>
Limiar de bairros	≥ 100	≥ 300	≥ 500
Categ. de idade	Básica (3)	Detalhada (5)	Simples (2)
Categ. de espera	Básica (3)	Detalhada (7)	Simples (2)
<i>Features</i> de interação	Básica (2)	+ doenças (3)	+ sms $\times$ espera (4)

Cada um dos 243 conjuntos de dados é uma versão diferente da mesma base, com as mesmas amostras tratadas com escolhas diferentes em cinco pontos do pipeline. Isso permite comparar não só os modelos, mas também o efeito de cada escolha de pré-processamento sobre o desempenho final.

Esses 243 conjuntos de dados ainda serão cruzados com diferentes configurações de hiperparâmetros dos modelos. Como esse cruzamento se organiza, e o total de experimentos que ele gera, aparece na Seção 3.4.

3.2 PRÉ-PROCESSAMENTO E ENGENHARIA DE *FEATURES*

3.2.1 Codificação de Variáveis Categóricas

As variáveis categóricas do conjunto de dados (*Gender*, *Neighbourhood*, *weekday*, *periodo_dia*, *faixa_idade* e *faixa_espera*) foram convertidas em forma numérica antes do treino, segundo os mecanismos descritos na Seção 2.4.1. Como as duas famílias de modelos respondem de forma distinta à representação das categorias, cada uma recebeu uma codificação diferente. A regressão logística recebeu as variáveis codificadas por *One-Hot Encoding* (Seção 2.4.2), com descarte de uma categoria de referência por variável para evitar a multicolinearidade. A árvore de decisão recebeu as mesmas variáveis codificadas por *Label Encoding* (Seção 2.4.3). A variável *Gender*, por já ser binária, foi codificada manualmente como $M=0$ e $F=1$ em ambos os *pipelines*. A Tabela 17 resume a estratégia adotada para cada família.

Tabela 17 – Codificação aplicada às variáveis categóricas em cada modelo.

Modelo	Codificação	Mecanismo
Regressão Logística	<i>One-Hot Encoding</i> (com descarte de uma categoria de referência)	Cada categoria vira uma coluna binária independente (0/1). Uma coluna por variável é descartada para evitar multicolinearidade.
Árvore de Decisão	<i>Label Encoding</i>	Cada categoria recebe um inteiro distinto (0, 1, 2, ...) numa única coluna. A codificação é ajustada no treino e reaplicada em validação e teste.

A escolha de uma codificação específica para cada modelo não é intercambiável. Aplicar a codificação da regressão logística à árvore de decisão, ou vice-versa, degrada o desempenho ou viola pressupostos do modelo, pelas razões detalhadas na Seção 2.4.4. Em todos os casos, a codificação foi ajustada apenas no conjunto de treino e reaplicada aos conjuntos de validação e teste com o mesmo mapeamento, de modo a evitar o vazamento de informação.

3.2.2 Normalização de Variáveis

As *features* numéricas do *pipeline* da regressão logística foram padronizadas pelo *z-score* (Seção 2.4.5), que centraliza cada coluna em média 0 e desvio padrão 1. A padronização foi aplicada apenas à regressão logística, pois ela calcula uma soma ponderada das *features* e seus coeficientes dependem da escala das variáveis de entrada. A árvore de decisão não passou pela normalização, já que seus cortes baseiam-se em limiares e são invariantes à escala absoluta das *features*.

A padronização foi ajustada apenas no conjunto de treino. A mesma média e o mesmo desvio padrão calculados ali foram reaplicados aos conjuntos de validação e teste, sem reajuste, evitando o vazamento de informação do teste para o treino.

3.3 MODELOS: REGRESSÃO LOGÍSTICA E ÁRVORE DE DECISÃO

Nesta seção, são apresentados os modelos de regressão logística e árvore de decisão utilizados nesse trabalho. Esses modelos foram escolhidos devido à sua eficácia, interpretabilidade e capacidade de lidar com dados heterogêneos, características essenciais para a previsão do absentismo em instituições de saúde.

A ideia desta seção é fazer uma comparação entre os diferentes resultados obtidos entre regressão logística e árvores de decisão em diferentes contextos a partir do conjunto de dados apresentado nas seções anteriores.

3.3.1 Regressão Logística

A descrição teórica da regressão logística (definição, equação, vantagens e limitações) está na Seção 2.3.1. Esta subseção apresenta a configuração utilizada nos experimentos.

O modelo recebe como entrada cada um dos 243 conjuntos de dados gerados pelo *pipeline* de pré-processamento (Seção 3.1.11), com codificação por *One-Hot Encoding* (Seção 2.4.2) e normalização das *features* pelo *StandardScaler* (Seção 2.4.5). Quatro hiperparâmetros foram permutados durante a varredura: *solver*, *penalty*, *C* e *l1_ratio*. Cada um é detalhado a seguir.

3.3.1.1 Solver

O *solver* é o algoritmo numérico que ajusta os coeficientes da regressão logística durante o treino. Diferentes *solvers* têm comportamentos numéricos distintos e nem todos suportam as mesmas penalidades (GÉRON, 2019). Três opções foram testadas, oferecidas pelo *sklearn.linear_model.LogisticRegression*:

- *liblinear*: algoritmo de *coordinate descent*, recomendado para conjuntos de dados pequenos e médios. Suporta as penalidades *l1* e *l2*.
- *lbfgs*: método *quasi-Newton* baseado em *Limited-memory BFGS*. É o *solver* padrão do *scikit-learn*. Suporta apenas a penalidade *l2*.
- *saga*: variante do *stochastic average gradient*, recomendada para conjuntos de dados grandes. Suporta todas as penalidades, incluindo *elasticnet*.

3.3.1.2 Penalty

A penalidade, também chamada de regularização, controla a complexidade do modelo por meio da adição de um termo à função de custo durante o treino. Esse termo penaliza coeficientes grandes, o que reduz o risco de *overfitting* e ajuda na generalização (GÉRON, 2019). Três tipos foram testados, mas cada um é compatível apenas com certos *solvers*:

- *l1* (compatível com *liblinear* e *saga*): adiciona à função de custo a soma dos valores absolutos dos coeficientes. Tende a zerar coeficientes irrelevantes, gerando modelos mais esparsos.
- *l2* (compatível com *liblinear*, *lbfgs* e *saga*): adiciona à função de custo a soma dos quadrados dos coeficientes. Reduz a magnitude de todos os coeficientes, mas raramente os zera.
- *elasticnet* (compatível com *saga*): combinação linear de *l1* e *l2*, controlada pelo parâmetro *l1_ratio*.

3.3.1.3 Parâmetro C

O parâmetro C é o inverso da força de regularização. Valores baixos de C resultam em regularização forte (modelo mais simples, coeficientes pequenos), e valores altos resultam em regularização fraca (modelo mais flexível, ajuste mais próximo dos dados de treino) (GÉRON, 2019). Quatro valores foram testados:

- $C = 0,01$: regularização forte. Coeficientes muito pequenos, modelo simples. Risco de *underfitting*.
- $C = 0,1$: regularização moderada.
- $C = 1,0$: valor padrão do *scikit-learn*. Equilíbrio comum entre simplicidade e ajuste.
- $C = 10,0$: regularização fraca. Modelo se aproxima de uma regressão logística sem penalização.

3.3.1.4 $L1$ Ratio

O $l1_ratio$ só é usado quando a penalidade é *elasticnet*, e define a proporção entre $l1$ e $l2$ na composição (GÉRON, 2019). Três valores foram testados:

- $l1_ratio = 0,0$: 100% $l2$, 0% $l1$. Equivale à penalidade $l2$ pura.
- $l1_ratio = 0,5$: 50% $l1$, 50% $l2$. Equilíbrio entre esparsidade e suavização.
- $l1_ratio = 1,0$: 100% $l1$, 0% $l2$. Equivale à penalidade $l1$ pura.

3.3.1.5 Total de combinações

A combinação dos quatro hiperparâmetros não é um produto cartesiano simples, porque *penalty* e $l1_ratio$ dependem do *solver*. A Tabela 18 lista as 32 combinações testadas para cada um dos 243 conjuntos de dados.

Tabela 18 – As 32 combinações de hiperparâmetros da regressão logística.

#	<i>Solver</i>	<i>Penalty</i>	<i>C</i>	<i>ll_ratio</i>
1	<i>liblinear</i>	<i>l1</i>	0,01	N/A
2	<i>liblinear</i>	<i>l1</i>	0,1	N/A
3	<i>liblinear</i>	<i>l1</i>	1,0	N/A
4	<i>liblinear</i>	<i>l1</i>	10,0	N/A
5	<i>liblinear</i>	<i>l2</i>	0,01	N/A
6	<i>liblinear</i>	<i>l2</i>	0,1	N/A
7	<i>liblinear</i>	<i>l2</i>	1,0	N/A
8	<i>liblinear</i>	<i>l2</i>	10,0	N/A
9	<i>lbfgs</i>	<i>l2</i>	0,01	N/A
10	<i>lbfgs</i>	<i>l2</i>	0,1	N/A
11	<i>lbfgs</i>	<i>l2</i>	1,0	N/A
12	<i>lbfgs</i>	<i>l2</i>	10,0	N/A
13	<i>saga</i>	<i>l1</i>	0,01	N/A
14	<i>saga</i>	<i>l1</i>	0,1	N/A
15	<i>saga</i>	<i>l1</i>	1,0	N/A
16	<i>saga</i>	<i>l1</i>	10,0	N/A
17	<i>saga</i>	<i>l2</i>	0,01	N/A
18	<i>saga</i>	<i>l2</i>	0,1	N/A
19	<i>saga</i>	<i>l2</i>	1,0	N/A
20	<i>saga</i>	<i>l2</i>	10,0	N/A
21	<i>saga</i>	<i>elasticnet</i>	0,01	0,0
22	<i>saga</i>	<i>elasticnet</i>	0,01	0,5
23	<i>saga</i>	<i>elasticnet</i>	0,01	1,0
24	<i>saga</i>	<i>elasticnet</i>	0,1	0,0
25	<i>saga</i>	<i>elasticnet</i>	0,1	0,5
26	<i>saga</i>	<i>elasticnet</i>	0,1	1,0
27	<i>saga</i>	<i>elasticnet</i>	1,0	0,0
28	<i>saga</i>	<i>elasticnet</i>	1,0	0,5
29	<i>saga</i>	<i>elasticnet</i>	1,0	1,0
30	<i>saga</i>	<i>elasticnet</i>	10,0	0,0
31	<i>saga</i>	<i>elasticnet</i>	10,0	0,5
32	<i>saga</i>	<i>elasticnet</i>	10,0	1,0

Além dos quatro hiperparâmetros permutados, três parâmetros são fixos em todas as execuções:

- *random_state* = 42: garante reprodutibilidade dos resultados em execuções subsequentes.
- *max_iter* = 5000: número de iterações suficiente para os *solvers* convergirem mesmo nos cenários em que os conjuntos de dados foram rebalanceados por SMOTE.
- *class_weight* = 'balanced': atribui pesos inversamente proporcionais à frequência das classes durante a estimação dos coeficientes. Esse ajuste é mantido em todas as execuções, independentemente do fator de balanceamento aplicado no pré-processamento (Seção 3.1.8).

3.3.2 Árvore de Decisão

A descrição teórica da árvore de decisão (definição, treinamento, métricas de impureza, vantagens, limitações e tratamento de desbalanceamento) está na Seção 2.3.2. Esta subseção apresenta a configuração utilizada nos experimentos.

O modelo recebe como entrada cada um dos 243 conjuntos de dados gerados pelo *pipeline* de pré-processamento (Seção 3.1.11), com codificação por *Label Encoding* (Seção 2.4.3). Não há normalização das *features*, já que a árvore de decisão é invariante a transformações monotônicas de escala. Dois hiperparâmetros foram permutados durante a varredura: *criterion* e *max_depth*. O treinamento usa o *DecisionTreeClassifier* do *scikit-learn* (GÉRON, 2019).

Embora a árvore de decisão atribua a cada folha a classe majoritária como predição direta, a probabilidade de falta usada neste trabalho não é esse rótulo, e sim a proporção de amostras de treino da classe *no-show* na folha em que o paciente cai. Como a profundidade máxima é limitada, as folhas raramente são puras, de modo que essa proporção assume uma faixa de valores entre 0 e 1. É essa estimativa de probabilidade que permite aplicar e deslocar o limiar de decisão, da mesma forma que na regressão logística, em vez de usar apenas a regra da classe majoritária.

3.3.2.1 Criterion

O *criterion* é a métrica de impureza usada pela árvore para escolher o melhor corte em cada nó (GÉRON, 2019). Duas opções foram testadas, oferecidas pelo *sklearn.tree.DecisionTreeClassifier*:

- *gini*: impureza de Gini. Mede a probabilidade de classificar de forma incorreta um exemplo do nó. Valor padrão do *scikit-learn*.
- *entropy*: entropia (vinda da Teoria da Informação). Mede a desordem da distribuição de classes no nó.

3.3.2.2 Max Depth

O *max_depth* controla a profundidade máxima da árvore. Quanto maior a profundidade, mais cortes a árvore pode fazer e mais flexível ela fica. Limitar a profundidade é uma das técnicas mais comuns para reduzir o risco de *overfitting* (GÉRON, 2019). Sete valores foram testados:

- *max_depth = None*: sem limite de profundidade. A árvore cresce até cada folha conter uma única classe. Capacidade máxima, alto risco de *overfitting*.
- *max_depth = 5*: árvore rasa. Modelo simples e robusto, mas com risco de *underfitting*.
- *max_depth = 10*: profundidade baixa.

- $max_depth = 15$: profundidade média.
- $max_depth = 20$: profundidade alta.
- $max_depth = 25$: profundidade muito alta.
- $max_depth = 30$: maior valor finito de profundidade avaliado.

3.3.2.3 Total de combinações

Os dois hiperparâmetros são independentes, então as 14 combinações resultam do produto cartesiano simples ($2 \times 7 = 14$). A Tabela 19 lista as 14 combinações testadas para cada um dos 243 conjuntos de dados.

Tabela 19 – As 14 combinações de hiperparâmetros da árvore de decisão.

#	Criterion	max_depth
1	<i>gini</i>	<i>None</i>
2	<i>gini</i>	5
3	<i>gini</i>	10
4	<i>gini</i>	15
5	<i>gini</i>	20
6	<i>gini</i>	25
7	<i>gini</i>	30
8	<i>entropy</i>	<i>None</i>
9	<i>entropy</i>	5
10	<i>entropy</i>	10
11	<i>entropy</i>	15
12	<i>entropy</i>	20
13	<i>entropy</i>	25
14	<i>entropy</i>	30

Além dos dois hiperparâmetros permutados, dois parâmetros são fixos em todas as execuções:

- $random_state = 42$: garante reprodutibilidade dos resultados em execuções subsequentes.
- $class_weight = 'balanced'$: atribui pesos inversamente proporcionais à frequência das classes durante a escolha dos cortes. Esse ajuste é mantido em todas as execuções, independentemente do fator de balanceamento aplicado no pré-processamento (Seção 3.1.8).

3.4 PERMUTAÇÃO TOTAL DE EXPERIMENTOS

Esta seção apresenta o cálculo do número total de experimentos executados. O ponto de partida são os 243 conjuntos de dados gerados pelo *pipeline* de pré-processamento (Seção 3.1.11). Cada um deles é cruzado com todas as configurações de hiperparâmetros dos dois

modelos descritas nas seções anteriores, e cada combinação é registrada como um experimento independente.

A regressão logística tem 32 combinações de hiperparâmetros por conjunto de dados (Tabela 18). A árvore de decisão tem 14 combinações por conjunto de dados (Tabela 19). Multiplicando os 243 conjuntos de dados por essas configurações, chega-se a 7.776 experimentos de regressão logística e 3.402 de árvore de decisão. O total geral é 11.178 execuções, conforme resumido na Tabela 20.

Tabela 20 – Permutação total de experimentos por modelo.

Modelo	Conjuntos de dados	Combinações de hiperparâmetros	Experimentos
Regressão Logística	243	32	7.776
Árvore de Decisão	243	14	3.402
Total			11.178

Cada uma das 11.178 rodadas é registrada nas planilhas de resultados das duas famílias, com uma linha por configuração testada. Além do AUC no treino, na validação e no teste, cada linha guarda as contagens de acertos e erros do modelo (verdadeiros positivos, verdadeiros negativos, falsos positivos e falsos negativos), a partir das quais se calculam a acurácia, a precisão, o *recall* e o F1. A Figura 2 mostra um trecho dessa planilha.

Figura 2 – Trecho da planilha de resultados da regressão logística. Cada linha corresponde a um experimento, com a configuração testada, as contagens de acertos e erros (verdadeiros e falsos, positivos e negativos) e o AUC no treino, na validação e no teste.

K	L	M	N	O	P	Q	R	
Versão	VP_Treinamento	FP_Treinamento	FN_Treinamento	VN_Treinamento	AUC_Treino	VP_Validacao	FP_Validacao	FN_Val
v65	7360	13307	4772	16773	0,612574669	1832	3417	
v77	7226	12958	4906	17122	0,612584134	1793	3325	
v87	7226	12958	4906	17122	0,612584134	1793	3325	
v97	7464	13434	4668	16646	0,614134083	1839	3428	
v129	7464	13434	4668	16646	0,614133977	1839	3428	
v161	7445	13372	4687	16708	0,614291333	1837	3422	
v365	7040	12553	5092	17527	0,61312582	1748	3216	
v375	7040	12553	5092	17527	0,61312582	1748	3216	
v577	7035	12691	5097	17389	0,609437545	1748	3264	
v589	6917	12383	5215	17697	0,609443809	1720	3176	
v599	6917	12383	5215	17697	0,609443809	1720	3176	
v609	7035	12691	5097	17389	0,609437383	1748	3264	
v621	6917	12383	5215	17697	0,60944382	1720	3176	
v631	6917	12383	5215	17697	0,60944382	1720	3176	
v641	7028	12670	5104	17410	0,609698257	1744	3274	
v653	6904	12359	5228	17721	0,609706253	1714	3193	
v662	6860	12233	5272	17847	0,612312536	1715	3149	
v663	6904	12359	5228	17721	0,609706253	1714	3193	
v673	7102	12890	5030	17190	0,611206592	1772	3298	
v685	6992	12526	5150	17544	0,611110771	1740	3214	

Fonte: elaborado pelo autor.

Dessas rodadas, foram selecionadas dez configurações para análise mais detalhada nas seções seguintes: cinco da regressão logística (v77, v365, v161, v662 e v589) e cinco da árvore de decisão (v304, v325, v241, v346 e v367). O critério de seleção foi aplicado em duas

etapas: primeiro a estabilidade, mantendo apenas as configurações com diferenças pequenas entre validação e teste nas três métricas (acurácia, precisão e recall). Depois, entre as estáveis, o maior AUC no conjunto de teste. Esse critério é detalhado na Seção 4.2.

Para garantir reprodutibilidade, todos os pontos de aleatoriedade do experimento usam semente fixa em 42. Isso vale para o embaralhamento dentro do *train_test_split*, para o SMOTE, para o *RandomUnderSampler* e para o treinamento dos modelos que dependem de inicialização aleatória.

3.5 ARQUITETURA DE EXECUÇÃO

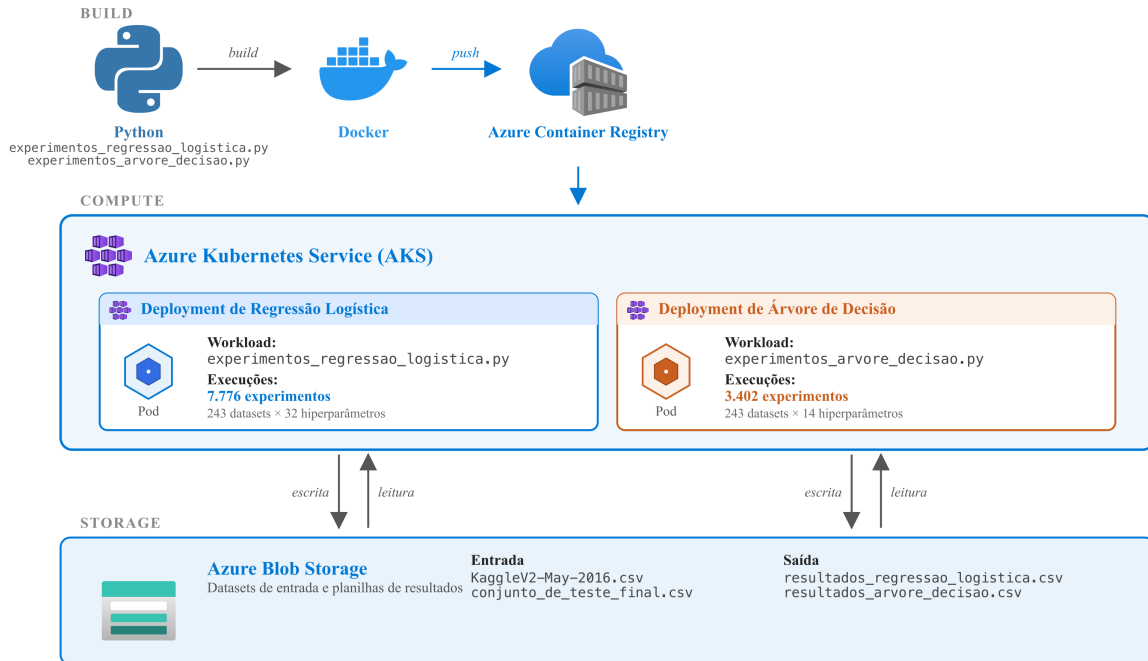
Os 11.178 experimentos descritos na Seção 3.4 foram executados em um *cluster Kubernetes* hospedado no Microsoft Azure. Esta seção descreve a arquitetura geral e o fluxo de execução.

A escolha por uma infraestrutura em *cluster* foi motivada pelo volume de execuções e pela separação clara das duas varreduras de experimentos. Rodar os 11.178 experimentos sequencialmente em uma única máquina levaria várias horas, podendo chegar a dias dependendo do *hardware*. O *cluster Kubernetes* permite isolar a execução em dois *deployments* dedicados, um por modelo (regressão logística e árvore de decisão), o que facilita o controle de recursos, o monitoramento de cada varredura e a recuperação em caso de falha de qualquer um dos *workloads* sem reiniciar o outro.

A Figura 3 ilustra o fluxo geral da arquitetura, cujos principais componentes são:

- *Docker*: empacota os *scripts Python* (*experimentos_regressao_logistica.py* e *experimentos_arvore_decisao.py*) e suas dependências (*scikit-learn*, *pandas*, *imblearn*) em uma imagem reprodutível.
- Azure Container Registry (ACR): repositório privado de imagens *Docker*. Armazena a imagem usada pelos *pods* do *cluster* e a torna acessível ao AKS por *image pull*.
- Azure Kubernetes Service (AKS): *cluster Kubernetes* gerenciado. Hospeda os dois *deployments* responsáveis pela execução dos experimentos.
- *Deployments* dedicados: dois *deployments* independentes, um para a varredura da regressão logística (7.776 experimentos) e outro para a da árvore de decisão (3.402 experimentos). Cada *deployment* mantém seu próprio *pod*, com escopo isolado de recursos e *logs*.
- Azure Blob Storage: armazena os conjuntos de dados de entrada (*KaggleV2-May-2016.csv* e *conjunto_de_teste_final.csv*) e os arquivos de resultados (*resultados_regressao_logistica.csv* e *resultados_arvore_decisao.csv*).

Figura 3 – Arquitetura de execução dos 11.178 experimentos em cluster Kubernetes no Azure. Ícones dos serviços Azure (AKS, ACR e Blob Storage) reproduzidos de Microsoft Corporation (2025). Ícones do *Python* e *Docker* reproduzidos de Simple Icons (2025).



Fonte: elaborado pelo autor.

O fluxo de execução começa pelo empacotamento do código e das dependências em uma imagem *Docker*. Essa imagem é publicada no Azure Container Registry e fica disponível para o *cluster* por *image pull*. No AKS, dois *deployments* são aplicados: um executa o `experimentos_regressao_logistica.py` e produz `resultados_regressao_logistica.csv`. O outro executa o `experimentos_arvore_decisao.py` e produz `resultados_arvore_decisao.csv`. Cada *deployment* mantém um *pod* que baixa a imagem do ACR, lê os conjuntos de dados de entrada do Azure Blob Storage, executa a varredura completa do modelo correspondente e grava o arquivo CSV (*Comma-Separated Values*) de resultados de volta no Blob Storage. As duas planilhas finais servem de base para as análises do Capítulo 4.

A escolha do Azure como provedor de nuvem foi feita pela familiaridade do autor com o ambiente e pelos créditos disponíveis para uso acadêmico. A arquitetura é, no entanto, portátil: o mesmo padrão de execução pode ser aplicado em outros provedores que ofereçam *Kubernetes* gerenciado, como o *Google Kubernetes Engine* (GKE), no Google Cloud, ou o *Elastic Kubernetes Service* (EKS), na Amazon Web Services (AWS), sem alterações no código dos experimentos.

3.6 MÉTRICAS NO CONTEXTO DO ABSENTEÍSMO

A Seção 2.5 apresentou as métricas de avaliação em termos gerais. Esta seção aplica cada uma ao contexto específico do problema, em que a classe positiva é o *no-show* (paciente não comparece) e a classe negativa é o comparecimento.

A combinação entre o que o modelo prevê e o que acontece na realidade gera quatro resultados possíveis:

- VP (previu falta, paciente faltou): o modelo acertou. A unidade pode agir antes: lembrete por SMS, ligação de confirmação ou *double-booking* para preservar a vaga.
- VN (previu comparecimento, paciente compareceu): o modelo acertou. Nenhuma ação preventiva é necessária e o recurso operacional não é gasto à toa.
- FP (previu falta, paciente compareceu): alarme falso. A equipe gasta uma ligação ou SMS sem necessidade. Custo baixo, da ordem de poucos reais por contato.
- FN (previu comparecimento, paciente faltou): oportunidade perdida. A unidade não acionou nada e a vaga ficou ociosa. Custo alto, próximo de R\$ 139 por exame especializado desperdiçado Beltrame et al. (2019).

As três métricas-chave para acompanhar o desempenho do modelo no problema de absenteísmo, com foco no que cada uma representa para a operação da unidade de saúde, são:

- *Recall*: quantas faltas reais o modelo consegue antecipar. É a métrica prioritária no contexto do absenteísmo, porque deixar passar uma falta sem ação (FN) custa muito mais do que disparar um alarme falso (FP). *Recall* alto significa poucas faltas escapando do radar do modelo.
- *F1-Score*: indica se o modelo equilibra dois objetivos opostos: pegar o máximo de faltas (*recall*) sem cansar a equipe com excesso de alarmes falsos (precisão). F1 alto é evidência de que o modelo é útil operacionalmente, e não apenas estatisticamente.
- AUC: capacidade do modelo de distinguir, pela probabilidade estimada, os pacientes que faltam dos que comparecem. Útil em cenários de capacidade restrita: se a unidade só pode agir sobre os 10% com maior probabilidade estimada de falta (por limite de telefonemas, por exemplo), uma AUC alta garante que esses 10% concentram de fato a maioria das faltas.

3.6.1 Estratégias de referência

Para julgar se os modelos agregam valor, seu desempenho é comparado ao de três estratégias de referência que não usam aprendizado de máquina. Cada uma representa uma decisão trivial que a unidade de saúde poderia tomar sem nenhum modelo:

- Prever que todos comparecem: classifica todo paciente como presença, nunca sinalizando uma falta. Acerta sempre a classe majoritária, o que produz acurácia igual à taxa de comparecimento, mas recall nulo, já que não identifica nenhuma falta. Como não aponta nenhum paciente como falta, não há positivos previstos, e a precisão é convencionada como zero.
- Escolha aleatória: classifica cada paciente como falta ou comparecimento ao acaso, com probabilidade igual à taxa base de faltas. Serve de piso para qualquer método que pretenda extrair algum sinal dos dados.
- Ligar para todos: classifica todo paciente como provável falta, o oposto da primeira estratégia. Captura todas as faltas (recall de 100%), mas com precisão igual à taxa base de faltas, pois trata da mesma forma quem falta e quem comparece.

Nenhuma das três atribui graus distintos de risco aos pacientes, de modo que todas têm AUC de 0,5, equivalente ao acaso. Por isso, o AUC é a métrica que melhor revela se um modelo de fato supera essas referências. A comparação numérica completa é apresentada na Seção 4.4.

4 RESULTADOS

4.1 VISÃO GERAL DOS EXPERIMENTOS

Este capítulo compara o desempenho da regressão logística e da árvore de decisão na previsão de faltas. O foco está no conjunto de teste, formado por agendamentos que os modelos não viram durante o treino, por ser a medida mais realista de como cada um se comportaria na prática.

As opções de pré-processamento (agrupamento de bairros, faixas de idade, faixas de tempo de espera e variáveis de interação) geraram 243 versões do conjunto de dados, e cada versão foi treinada com as diferentes configurações de cada modelo. No total, foram 7.776 execuções da regressão logística e 3.402 da árvore de decisão. A análise começa por esse quadro geral e, nas seções seguintes, concentra-se nos melhores modelos.

4.1.1 Panorama geral do AUC

A Tabela 21 resume o AUC das duas famílias nos três conjuntos. No teste, além da mediana (o valor central que separa as configurações em duas metades, com 50% delas apresentando AUC maior ou igual a ela e 50% menor ou igual), aparece também o melhor resultado obtido por cada família.

Tabela 21 – Panorama do AUC nas duas famílias de modelos.

Família	Experimentos	AUC treino (mediana)	AUC validação (mediana)	AUC teste (mediana)	AUC teste (máximo)
Regressão Logística	7.776	0,617	0,613	0,595	0,604
Árvore de Decisão	3.402	0,963	0,576	0,522	0,588

A regressão logística mantém um desempenho parecido nos três conjuntos: cerca de 0,62 no treino e 0,60 no teste. Como a queda do treino para o teste é pequena, o padrão que ela aprende continua valendo em dados novos. Não é um modelo forte, já que 0,60 ainda está longe de um AUC alto, mas é consistente.

Com a árvore de decisão acontece o contrário. No treino, ela atinge um AUC mediano de 0,96, perto do acerto perfeito. No teste, cai para 0,52, quase o nível do acaso. Isso é um caso de *overfitting*: a árvore decora os dados de treino, inclusive seus detalhes e ruídos, e acaba não prevendo bem para pacientes que não viu. O elevado desempenho no treino, portanto, tem pouco significado prático, pois não se reflete no conjunto de teste.

Há várias formas de conter esse sobreajuste. A mais direta é limitar o crescimento da árvore, pela profundidade máxima ou pela exigência de um número mínimo de amostras por folha, para que ela não aprenda regras específicas demais para poucos exemplos. Outra é a poda, que remove ramos de pouca contribuição. Este trabalho já variou a profundidade máxima na

varredura de hiperparâmetros, o que ajuda a controlar o problema. Um caminho mais robusto, deixado como trabalho futuro, são os *ensembles* como o *Random Forest*, que combinam muitas árvores e reduzem a variância de uma árvore isolada (GÉRON, 2019).

Essa diferença também aparece na consistência das duas famílias de modelos, resumida na Tabela 22. Todas as 7.776 configurações de regressão logística superaram um AUC de 0,55 no teste, e uma em cada cinco passou de 0,60. Entre as 3.402 árvores, menos de um quarto chegou a 0,55 e nenhuma alcançou 0,60. Além de ter um teto mais baixo, a árvore de decisão é muito menos confiável: a maioria de suas configurações fica perto do acaso em dados novos.

Tabela 22 – Proporção de configurações de cada família que ultrapassam um dado nível de AUC no teste.

AUC de teste	regressão logística	árvore de decisão
acima de 0,55	100%	23%
acima de 0,58	68%	9%
acima de 0,60	20%	0%

O resultado é claro: nos experimentos realizados, a regressão logística obteve melhores resultados que a árvore de decisão. Não por ser um modelo excelente, mas por entregar um desempenho modesto e estável, enquanto a árvore acerta muito no treino e falha no teste. A próxima seção detalha como os melhores modelos de cada família foram escolhidos.

4.2 SELEÇÃO E ESTABILIDADE DOS MODELOS

Um AUC alto na validação não basta para confiar em um modelo. É preciso que esse desempenho se mantenha no teste. Quando o resultado cai muito da validação para o teste, o modelo não é confiável, ainda que a validação pareça boa. Por isso, a seleção dos modelos para análise detalhada incluiu um filtro de estabilidade, e não apenas o AUC.

A estabilidade foi medida pela diferença entre validação e teste em três métricas: acurácia, precisão e recall. Quanto mais perto de zero, melhor, porque significa que o que apareceu na validação se repetiu no teste. A seleção teve duas etapas: primeiro foram mantidas apenas as configurações estáveis. Depois, entre elas, foram escolhidas as cinco de cada família com maior AUC no teste. As Tabelas 23 e 24 mostram esses dez modelos, cinco de cada família: o AUC no treino, na validação e no teste, e as diferenças absolutas entre validação e teste em acurácia, precisão e *recall* (colunas Δ).

Tabela 23 – Estabilidade dos cinco modelos de regressão logística selecionados: AUC nos três conjuntos e diferenças absolutas entre validação e teste.

Modelo	AUC treino	AUC validação	AUC teste	Δ acurácia	Δ precisão	Δ <i>recall</i>
v77	0,613	0,608	0,603	0,003	0,009	0,009
v365	0,613	0,608	0,603	0,004	0,009	0,010
v161	0,614	0,609	0,601	0,003	0,008	0,010
v662	0,612	0,607	0,599	0,001	0,007	0,008
v589	0,609	0,605	0,599	0,002	0,008	0,007

Tabela 24 – Estabilidade dos cinco modelos de árvore de decisão selecionados: AUC nos três conjuntos e diferenças absolutas entre validação e teste.

Modelo	AUC treino	AUC validação	AUC teste	Δ acurácia	Δ precisão	Δ <i>recall</i>
v304	0,683	0,580	0,564	0,006	0,014	0,010
v325	0,683	0,579	0,563	0,005	0,013	0,008
v241	0,683	0,579	0,563	0,008	0,015	0,008
v346	0,683	0,579	0,563	0,006	0,013	0,010
v367	0,683	0,579	0,563	0,006	0,014	0,010

Os dez modelos são estáveis. Na regressão logística, as três diferenças entre validação e teste ficam todas abaixo de 0,01. Na árvore de decisão, ficam abaixo de 0,015, um pouco maiores, mas ainda pequenas. Nenhum modelo teve queda brusca no teste, ou seja, nas duas famílias a validação serviu como uma boa prévia do teste.

Entre treino e validação o quadro é diferente. As árvores selecionadas partem de um AUC de treino em torno de 0,68 e caem para cerca de 0,58 na validação, uma diferença de quase 0,10. Vale notar que esse 0,68 é bem menor do que a mediana de 0,96 do conjunto completo de árvores (Seção 4.1): o filtro de estabilidade acabou escolhendo justamente as árvores de profundidade limitada, que decoram menos o treino. Ainda assim, sobra alguma queda do treino para a validação, típica desse tipo de modelo. A regressão logística cai bem menos: o AUC médio dos cinco modelos selecionados vai de cerca de 0,612 no treino para 0,607 na validação, uma diferença de aproximadamente 0,005. Da validação para o teste, as duas famílias se mantêm estáveis, que é o que as diferenças da tabela medem.

A diferença que decide a comparação está no AUC de teste: a regressão logística fica em torno de 0,60 e a árvore de decisão em torno de 0,56. Como as duas são estáveis, a escolha entre elas não passa pela estabilidade, e sim por essa vantagem de desempenho da regressão logística.

4.2.1 Diferença entre estabilidade e qualidade

Estabilidade e qualidade são coisas diferentes, e é importante não confundir as duas. A estabilidade só diz que o modelo se comporta de forma parecida na validação e no teste. Ela não diz se o modelo acerta muito ou pouco.

O problema é que um modelo que erra de modo parecido na validação e no teste também tem diferença quase zero entre os dois, e isso não tem mérito nenhum. Um modelo pode ser, ao mesmo tempo, estável e ruim.

Foi o que apareceu no conjunto completo de árvores: várias configurações tinham diferença quase zero entre validação e teste, mas AUC em torno de 0,52, perto de um chute. Eram estáveis e, ainda assim, sem qualidade.

Por esse motivo, a estabilidade foi usada como um filtro, e não como a nota final. Primeiro descartam-se os modelos instáveis. Depois, entre os que sobraram, escolhem-se os melhores pelo AUC. Os dez modelos selecionados passam nas duas exigências: são estáveis e estão entre os de maior AUC nas respectivas famílias, ainda que o desempenho preditivo permaneça modesto.

4.3 DESEMPENHO DOS MODELOS FINAIS NO CONJUNTO DE TESTE

A Seção 4.2 selecionou os dez modelos mais confiáveis. Esta seção detalha o desempenho do melhor modelo de cada família no teste, usando, além do AUC, a acurácia, a precisão, o *recall* e o F1. O melhor modelo de regressão logística (v77) emprega *solver saga*, penalidade L1 e parâmetro de regularização $C = 0,01$. O melhor de árvore de decisão (v304) usa o critério de entropia e profundidade máxima de 10 níveis. Ambos foram treinados sobre a versão do conjunto de dados sem balanceamento de classes e com limiar de agrupamento de bairros de 100 consultas. A Tabela 25 compara os dois. Os outros quatro modelos de cada família têm valores quase iguais, como se vê nas Tabelas 23 e 24.

Tabela 25 – Métricas dos melhores modelos de cada família, no teste.

Métrica	Regressão Logística (v77)	Árvore de Decisão (v304)
AUC	0,603	0,564
Acurácia	0,564	0,544
Precisão	0,342	0,322
Recall	0,600	0,569
F1	0,435	0,411

Em todas as cinco métricas, a regressão logística fica à frente da árvore de decisão, embora por margens pequenas: a diferença de AUC é de cerca de 0,04 (0,603 contra 0,564) e, nas demais métricas, situa-se entre 0,02 e 0,03. Essa vantagem consistente, ainda que modesta, acompanha o que o AUC já indicava na visão geral (Seção 4.1) e sugere que a relação linear

capturada pela regressão logística generaliza um pouco melhor para dados novos do que os cortes da árvore. Mais relevante que a diferença entre as duas, porém, é o patamar absoluto em que ambas operam: nenhuma das duas famílias alcança um desempenho que possa ser considerado forte, com AUC pouco acima de 0,5 e precisão em torno de um terço.

A Tabela 26 mostra as matrizes de confusão dos dois modelos, que cruzam o que o paciente fez de verdade (faltou ou compareceu) com o que o modelo previu.

Regressão Logística (v77)

Tabela 26 – Matrizes de confusão no teste (18.593 agendamentos, dos quais 5.208 faltas).

	Previu falta	Previu comparecimento
Faltou (real)	3.125	2.083
Compareceu (real)	6.019	7.366

Árvore de Decisão (v304)		
	Previu falta	Previu comparecimento
Faltou (real)	2.963	2.245
Compareceu (real)	6.233	7.152

O modelo de regressão logística marcou 9.144 pacientes como prováveis faltas (3.125 mais 6.019). Desses, apenas 3.125 faltaram de fato, o que dá uma precisão de 34%: a cada três pacientes apontados como risco, só um realmente faltou. Dos 5.208 pacientes que faltaram, o modelo identificou 3.125, ou seja, 60% deles (o *recall*). Os outros 2.083 passaram despercebidos.

Como a taxa de faltas no teste é de 28%, uma precisão de 34% significa que o modelo acerta só um pouco mais do que apontar pacientes ao acaso. A acurácia de 56% pode parecer razoável, mas, como visto na Seção 4.1, ela engana num problema desbalanceado e não deve ser olhada sozinha.

A árvore de decisão segue o mesmo padrão, com números um pouco piores: precisão de 32% e *recall* de 57%. A regressão logística obteve o melhor desempenho entre as duas famílias nesses experimentos, mas nenhuma das duas prevê bem neste problema. A próxima seção compara esse desempenho com estratégias simples, que não usam modelo nenhum.

4.4 COMPARAÇÃO COM ESTRATÉGIAS SIMPLES

As seções anteriores mostraram que os modelos preveem pouco, mas esse desempenho precisa ser comparado com alguma referência para saber se eles agregam algo. Por isso, foram colocados lado a lado com três estratégias simples, que não usam modelo nenhum e estão detalhadas na Seção 3.6.1: prever que todos comparecem (nunca aciona nada), escolher pacientes

ao acaso, e ligar para todos (tratar todo paciente como provável falta). A Tabela 27 traz essas estratégias e os melhores modelos de cada família, sempre no teste.

Tabela 27 – Estratégias simples comparadas aos melhores modelos, no teste.

Estratégia	acurácia	precisão	<i>recall</i>	F1	AUC
Prever que todos comparecem	0,720	0,000	0,000	0,000	0,500
Escolha aleatória	0,500	0,280	0,500	0,359	0,500
Ligar para todos	0,280	0,280	1,000	0,438	0,500
Regressão Logística (v77)	0,564	0,342	0,600	0,435	0,603
Árvore de Decisão (v304)	0,544	0,322	0,569	0,411	0,564

Pela acurácia, prever que todos comparecem (72%) ganha dos dois modelos (56% e 54%). Mas essa estratégia nunca identifica uma falta: tem *recall* zero e seria inútil para a unidade de saúde. A acurácia alta apenas confirma que essa métrica não serve para avaliar este problema.

Ligar para todos pega todas as faltas (*recall* de 100%), mas tem precisão de 28%, exatamente a taxa de faltas do conjunto, então não distingue um paciente do outro. E o F1 dessa estratégia (0,438) é praticamente igual ao da regressão logística (0,435). No corte padrão de decisão, o modelo não supera ligar para todos.

A diferença entre os modelos e as estratégias simples aparece no AUC. As estratégias triviais ficam todas em 0,500 porque nenhuma delas atribui um grau de risco aos pacientes: prever sempre a mesma classe, ou sortear ao acaso, não separa quem falta de quem comparece, e um classificador sem esse poder de distinção corresponde à diagonal da curva ROC, cujo AUC é exatamente 0,5. Os modelos, ao contrário, produzem uma probabilidade de falta para cada paciente, e com isso alcançam AUC de 0,603 (regressão logística) e 0,564 (árvore de decisão). Esse é o único aspecto em que superam as estratégias triviais, e ainda assim de forma modesta: um AUC de 0,603 significa que, tomados ao acaso um paciente que faltou e um que compareceu, o modelo atribui maior risco ao faltante em cerca de 60% das vezes, pouco acima dos 50% de um sorteio.

A Tabela 28 resume a comparação e a armadilha de cada métrica: em quase todas, uma estratégia trivial parece a melhor, e só o AUC realmente separa os modelos das alternativas que não usam aprendizado.

Tabela 28 – Resumo da comparação: a estratégia que aparece como melhor em cada métrica e a interpretação correspondente.

Métrica	Estratégia com melhor valor	Valor	Interpretação
Acurácia	Prever que todos comparecem	0,720	Engana: nunca identifica uma falta (<i>recall</i> zero), então a estratégia é inútil na prática.
<i>Recall</i>	Ligar para todos	1,000	Engana: trata todo paciente como falta. A precisão (0,280) é igual à taxa base, ou seja, não distingue ninguém.
AUC	regressão logística	0,603	Única métrica em que um modelo supera as estratégias triviais, ainda que de forma modesta (0,603 contra 0,500), por distinguir melhor faltantes de comparecentes.

É importante esclarecer por que esse pequeno ganho de AUC é o que carrega valor prático, apesar de o modelo ser inferior nas demais métricas. Acurácia, precisão e recall foram calculadas no corte fixo de 0,50 e, num problema desbalanceado, acabam dominadas por estratégias que sempre apostam na mesma classe, como já discutido. O AUC, por não depender de um corte específico, mede outra coisa: a qualidade com que o modelo separa faltantes de comparecentes pela probabilidade estimada, independentemente do limiar escolhido. Um AUC acima de 0,5 indica que o modelo, em média, atribui probabilidade mais alta a quem falta do que a quem comparece. Num cenário de capacidade restrita, em que a unidade só consegue contatar uma fração dos pacientes, agir sobre os de maior probabilidade estimada tende a alcançar mais faltantes do que sortear ou contatar todos indistintamente. Trata-se, porém, de um ganho modesto, condizente com um AUC de cerca de 0,60, e não de um modelo de alto poder preditivo.

O teto de desempenho observado, com AUC em torno de 0,60, indica que as variáveis disponíveis no conjunto de dados explicam apenas parte do comportamento de faltas. Essa constatação, aprofundada na Seção 5.1, é um dos resultados do trabalho, e não uma limitação assumida de antemão. Em síntese, a comparação evidenciou três pontos. Primeiro, a regressão logística obteve melhor desempenho que a árvore de decisão, embora ambas tenham desempenho modesto. Segundo, a acurácia, isoladamente, levaria a conclusões equivocadas num problema desbalanceado. Terceiro, o ganho dos modelos sobre as estratégias triviais concentra-se na capacidade de distinguir faltantes de comparecentes, captada pelo AUC.

4.5 ANÁLISE DE LIMIAR DE DECISÃO

Até aqui, os modelos foram avaliados no limiar padrão de 0,50: o paciente é classificado como provável falta quando a probabilidade prevista passa de 0,50. Esse limiar é uma convenção e pode ser ajustado para favorecer a precisão ou o recall. Esta seção usa o melhor modelo, a regressão logística v77, para mostrar como as métricas mudam ao longo dos limiares no teste.

A Tabela 29 traz essa variação. A coluna Marcados como falta é o número de pacientes que o modelo apontaria como risco em cada limiar.

Tabela 29 – Variação das métricas do modelo v77 (regressão logística) por limiar de decisão, no teste.

Limiar	Marcados como falta	precisão	<i>recall</i>	F1	acurácia
0,05	18.593	0,280	1,000	0,438	0,280
0,10	18.593	0,280	1,000	0,438	0,280
0,15	18.593	0,280	1,000	0,438	0,280
0,20	18.593	0,280	1,000	0,438	0,280
0,25	18.592	0,280	1,000	0,438	0,280
0,30	18.504	0,281	0,997	0,438	0,283
0,35	17.747	0,286	0,973	0,442	0,311
0,40	16.091	0,296	0,915	0,448	0,367
0,45	13.309	0,312	0,797	0,448	0,451
0,50	9.144	0,342	0,600	0,435	0,564
0,55	4.987	0,377	0,361	0,369	0,654
0,60	1.914	0,421	0,155	0,226	0,704
0,65	437	0,451	0,038	0,070	0,718
0,70	48	0,438	0,004	0,008	0,720
0,75	3	0,333	0,000	0,000	0,720
0,80	0	0,000	0,000	0,000	0,720
0,85	0	0,000	0,000	0,000	0,720
0,90	0	0,000	0,000	0,000	0,720
0,95	0	0,000	0,000	0,000	0,720

A tabela mostra o *trade-off* entre precisão e *recall*. Quando o limiar baixa, o modelo marca mais pacientes como falta: o *recall* sobe, mas a precisão cai. Quando o limiar sobe, acontece o contrário, pega menos faltas, porém com mais acertos entre os marcados.

Os limiares muito baixos (até 0,25) e muito altos (a partir de 0,80) não mudam nada. Isso ocorre porque o modelo, treinado com pesos de classe balanceados, quase nunca atribui probabilidade abaixo de 0,30 ou acima de 0,75. No limiar 0,25 ele marca praticamente todos os pacientes, o que equivale a ligar para todos. No limiar 0,80, não marca ninguém.

O F1 atinge o máximo de 0,448 no limiar 0,45. Mesmo nesse melhor ponto, o valor é quase igual ao da estratégia de ligar para todos (0,438), vista na Seção 4.4. Ou seja, nem ajustando o limiar o modelo se descola muito dessa estratégia trivial. A acurácia cresce com o limiar, mas só porque o modelo passa a prever comparecimento para quase todos, repetindo a armadilha discutida na Seção 4.1.

A árvore de decisão se comporta de forma parecida, embora valha uma ressalva: diferentemente da regressão logística, cuja saída já é uma probabilidade, a predição direta da árvore é a classe majoritária da folha. Ainda assim, a proporção de *no-shows* na folha em que cada paciente cai fornece, de forma natural, uma estimativa de probabilidade, e é ela que se usa aqui. Como a profundidade é limitada e as folhas não são puras, essas proporções variam o suficiente para que o limiar também possa ser deslocado, conforme detalhado na Seção 3.3.2. A Tabela 30 traz a variação do v304, o melhor modelo dessa família, e confirma que o padrão observado na regressão logística se repete.

Tabela 30 – Variação das métricas do modelo v304 (árvore de decisão) por limiar de decisão, no teste.

Limiar	Marcados como falta	precisão	<i>recall</i>	F1	acurácia
0,05	17.711	0,282	0,960	0,436	0,305
0,10	17.678	0,282	0,958	0,436	0,306
0,15	17.668	0,282	0,958	0,436	0,306
0,20	17.577	0,282	0,953	0,436	0,309
0,25	16.919	0,286	0,930	0,438	0,331
0,30	16.350	0,289	0,906	0,438	0,348
0,35	15.192	0,292	0,852	0,435	0,380
0,40	13.273	0,301	0,767	0,432	0,436
0,45	11.569	0,306	0,681	0,423	0,479
0,50	9.196	0,322	0,569	0,411	0,544
0,55	6.064	0,335	0,391	0,361	0,613
0,60	3.994	0,343	0,263	0,298	0,652
0,65	2.381	0,335	0,153	0,210	0,678
0,70	838	0,325	0,052	0,090	0,704
0,75	629	0,324	0,039	0,070	0,708
0,80	321	0,277	0,017	0,032	0,712
0,85	234	0,252	0,011	0,022	0,716
0,90	174	0,264	0,009	0,017	0,715
0,95	140	0,271	0,007	0,014	0,716

Os números são apenas um pouco piores: o F1 máximo fica em 0,438, nos limiares baixos, e a precisão não passa de 0,34 em nenhum ponto. A árvore espalha as probabilidades por uma faixa um pouco mais larga, mas isso não vira vantagem. Os demais modelos selecionados seguem o mesmo comportamento.

5 CONCLUSÃO

O objetivo geral deste trabalho foi investigar o uso de modelos de aprendizado de máquina para prever o absenteísmo em agendamentos de saúde. Para isso, foram comparadas duas famílias de modelos, a regressão logística e a árvore de decisão, sobre um conjunto de agendamentos médicos. A comparação seguiu um protocolo amplo: 243 versões diferentes do conjunto de dados, combinadas com várias configurações de cada modelo, totalizando mais de onze mil experimentos, seguidos de um filtro de estabilidade e da seleção de dez modelos para análise detalhada. Mais do que um modelo pronto para uso, o trabalho entregou uma avaliação controlada do que esses modelos conseguem, e do que não conseguem, fazer com os dados disponíveis.

Um achado central é o papel do limiar de decisão. O modelo não decide sozinho: cada um dos modelos produz, para cada agendamento, uma estimativa de probabilidade de falta, no caso da árvore a proporção da classe na folha (Seção 3.3.2), e é o limiar que transforma essa estimativa em ação. A análise mostrou que não existe um único melhor limiar: baixá-lo aumenta o *recall* à custa da precisão, e aumentá-lo faz o contrário. Onde colocar a linha de corte depende de quanto a unidade de saúde valoriza não deixar passar uma falta em comparação a evitar contatos desnecessários, sendo, portanto, uma decisão de gestão, e não uma escolha puramente técnica. Por isso este trabalho reporta a curva de limiares inteira, em vez de um número único por modelo.

A comparação também mostrou onde está a vantagem dos modelos sobre as estratégias triviais: a capacidade de ordenar os pacientes por risco de falta, que é o que o AUC mede. Essa vantagem é modesta, com AUC em torno de 0,60. Nos experimentos realizados, a regressão logística obteve melhores resultados que a árvore de decisão, e o valor prático dessa distinção depende de como a unidade de saúde a utiliza, por exemplo para priorizar contatos quando não é possível agir sobre todos os pacientes.

5.1 LIMITAÇÕES DO CONJUNTO DE DADOS

A investigação das limitações do conjunto de dados foi conduzida após a avaliação dos modelos, para compreender o desempenho observado. Duas características ajudam a explicar por que o teto de desempenho ficou em torno de 0,60 de AUC: o período de coleta e a cobertura geográfica dos dados, e o nível de detalhe das variáveis socioeconômicas.

Os 110.527 agendamentos cobrem apenas três meses, de abril a junho de 2016, em unidades básicas de saúde de Vitória, no Espírito Santo. A janela é curta e anterior à pandemia de COVID-19, período que acelerou mudanças no acesso aos serviços de saúde, com destaque para a expansão da telessaúde no país (CAETANO et al., 2020), fatores que podem ter alterado os padrões de comparecimento desde então. A restrição a um único município também limita a generalização, pois hábitos de uso do sistema de saúde, distâncias e o perfil dos bairros de uma

capital não correspondem necessariamente aos de municípios menores ou de outras regiões. Um modelo treinado nesse recorte aprende sinais específicos de Vitória em 2016, e aplicá-lo a outro contexto exigiria nova coleta e novo treinamento.

Além disso, das 14 variáveis do conjunto de dados, apenas duas carregam algum sinal socioeconômico: *Scholarship*, que indica beneficiários do Bolsa Família, e *Neighbourhood*, que registra o bairro do atendimento. Variáveis que a literatura sobre absenteísmo costuma associar à falta, como renda, escolaridade, distância até o serviço e meio de transporte, não estão disponíveis na base (DANTAS et al., 2018). Sem elas, o modelo apoia-se em substitutos: o bairro como aproximação de renda e distância, e *Scholarship* como único marcador individual de vulnerabilidade. Isso reduz o poder explicativo da análise e ajuda a explicar o desempenho preditivo modesto observado nos resultados.

5.2 RISCO ÉTICO EM APLICAÇÃO REAL

Tratar o absenteísmo como problema de classificação introduz riscos que não desaparecem com bons números de validação. Duas das poucas variáveis disponíveis são socialmente sensíveis: a bolsa família (*Scholarship*) identifica oficialmente famílias em situação de vulnerabilidade social, e o bairro (*Neighbourhood*) funciona como substituto aproximado de renda e infraestrutura. Caso um modelo se apoie fortemente nessas variáveis para prever a falta, ele pode reproduzir de forma automática a desigualdade estrutural que está por trás do não comparecimento. A esses fatores somam-se variáveis de saúde, como hipertensão, diabetes, alcoolismo e deficiência, e a variável derivada *multi_doenca*, que indica a presença de mais de uma condição. Por serem dados clínicos, são igualmente sensíveis e reforçam a necessidade de proteção e de cautela no uso preditivo dessas informações.

O risco fica mais concreto na política de *over-booking*, que consiste em marcar dois pacientes no mesmo horário quando um deles tem alta probabilidade prevista de faltar. Se os pacientes sinalizados como prováveis faltantes forem, em larga medida, os mais vulneráveis, uma política construída sobre essa previsão tenderia a alocar dois pacientes vulneráveis na mesma vaga, podendo deixar sem atendimento justamente quem mais depende do serviço público. Esse risco recomenda cautela em qualquer uso automatizado da previsão e reforça que a decisão final não deve ser delegada ao modelo.

5.3 DA PREVISÃO À POLÍTICA DE INTERVENÇÃO

O resultado técnico deste trabalho é um modelo que estima, para cada agendamento, a probabilidade de o paciente não comparecer. O resultado desejado pelo serviço de saúde é outro: a redução efetiva do absenteísmo. Entre um e outro há um espaço que este trabalho não preenche, o da política de intervenção. Sem uma ação acoplada à previsão, o modelo não altera o estado da agenda, pois um agendamento marcado como alto risco continua como estava, a

menos que alguma ação operacional seja disparada a partir dele.

Essa ação pode tomar várias formas, cada uma com custo, ganho esperado e risco distintos: a confirmação ativa por ligação ou SMS, o lembrete priorizado para os casos de maior risco, o *over-booking*, o reagendamento proativo ou o apoio direto, como transporte assistido ou vaga preferencial. A escolha entre elas não é uma decisão técnica, pois depende de quanto a unidade pode investir, de qual taxa de absenteísmo precisa atacar, de quanto incômodo é aceitável para o paciente e de quem assume o risco quando o modelo erra.

Dois limites de interpretação precisam ficar claros. Primeiro, o trabalho não faz nenhuma afirmação de causalidade: saber que uma variável ajuda a prever a falta não significa que ela a cause. Segundo, o trabalho não conclui que o modelo esteja pronto para uso direto em produção. Colocá-lo em operação exigiria, antes, uma política institucional de uso que definisse o peso de cada tipo de erro, a capacidade de atendimento disponível e o limiar adequado a essa realidade. O modelo informa a decisão, mas não decide sozinho.

Dentro desses limites, o trabalho cumpre o seu objetivo, ao mostrar o que modelos clássicos e interpretáveis efetivamente oferecem como apoio à gestão do absenteísmo, deixando claro que o ganho sobre estratégias simples é pequeno e que seu aproveitamento depende mais da política de uso do que do modelo em si.

5.4 TRABALHOS FUTUROS

Os resultados e os limites apontam direções de continuidade. A primeira é avaliar um conjunto de dados brasileiro atual. A base usada aqui é de 2016, anterior à pandemia, e repetir a análise sobre dados recentes, idealmente em parceria com uma rede municipal ou operadora de saúde, daria acesso a variáveis mais ricas e a um cenário mais próximo da aplicação real, além de permitir verificar se os padrões observados se mantêm em outros municípios e regiões, já que a base atual se restringe a um único município e a um período de três meses, o que limita a generalização dos resultados.

A segunda é testar outras famílias de modelos. Os dois modelos comparados são simples e de baixo poder preditivo, e um caminho natural é experimentar alternativas como o *Random Forest*, que combina muitas árvores de decisão e costuma generalizar melhor a casos não vistos no treino (GÉRON, 2019). Além dele, valeria comparar abordagens de natureza distinta, como as redes Bayesianas, que modelam de forma explícita as relações de probabilidade entre as variáveis, e os mapas auto-organizáveis de Kohonen, que exploram agrupamentos nos dados sem supervisão.

A terceira é prototipar e avaliar uma política de intervenção. O escopo deste trabalho encerra-se na etapa de previsão. O passo seguinte é acoplar à previsão uma ação concreta, como uma confirmação ativa ou um lembrete priorizado para os pacientes de maior risco, e medir o seu efeito real por meio de um experimento controlado, comparando um grupo que recebe a intervenção guiada pelo modelo com um grupo de controle. Apenas esse tipo de avaliação

mostra se a previsão, de fato, reduz as faltas na prática.

Por fim, vale discutir os limites das próprias técnicas. O AUC em torno de 0,60 vem de dois fatores: os dados e os modelos. A regressão logística capta apenas relações lineares, e a árvore de decisão, sozinha, tende ao sobreajuste. Modelos mais fortes podem melhorar esse resultado, mas só até onde os dados permitem, já que nenhum modelo encontra um padrão que não está nas variáveis. Saber quanto do limite vem do modelo e quanto vem dos dados exigiria refazer a comparação com uma base mais rica, o que já é uma pesquisa em si.

REFERÊNCIAS

- Agência Pará. **Hospital Regional dos Caetés lança campanha “Absentéismo Zero” para otimizar o atendimento aos usuários**. 2024. <https://agenciapara.com.br/noticia/52668/hospital-regional-dos-caetes-lanca-campanha-absenteismo-zero-para-otimizar-o-atendimento-aos-usuarios>. Acesso em: 23 mar. 2024.
- BELTRAME, S. M. et al. Absenteísmo de usuários como fator de desperdício: desafio para sustentabilidade em sistema universal de saúde. **Saúde em Debate**, v. 43, n. 123, p. 1015–1030, 2019. Acesso em: 26 maio 2026.
- BISHOP, C. M. **Pattern Recognition and Machine Learning**. New York: Springer, 2006.
- CAETANO, R. et al. Desafios e oportunidades para telessaúde em tempos da pandemia pela COVID-19: uma reflexão sobre os espaços e iniciativas no contexto brasileiro. **Cadernos de Saúde Pública**, v. 36, n. 5, p. e00088920, 2020.
- CARRERAS-GARCÍA, D. et al. Patient no-show prediction: A systematic literature review. **Entropy**, v. 22, n. 6, p. 675, 2020.
- DANTAS, L. F. et al. No-shows in appointment scheduling – a systematic literature review. **Health Policy**, v. 122, n. 4, p. 412–421, 2018.
- Fiocruz. **Fatores associados ao absenteísmo e possíveis intervenções - Lúcia Rotenberg**. 2015. <https://portal.fiocruz.br/video/fatores-associados-ao-absenteismo-e-possiveis-intervencoes-lucia-rotenberg>. Acesso em: 3 jun. 2026.
- GÉRON, A. **Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow**. 2. ed. Sebastopol: O'Reilly Media, 2019.
- GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. **Deep Learning**. Cambridge: MIT Press, 2016.
- GOODMAN, P. S.; ATKIN, R. S. **Absenteeism: New Approaches to Understanding, Measuring, and Managing Employee Absence**. San Francisco: Jossey-Bass, 1984.
- GUROL-URGANCI, I. et al. Mobile phone messaging reminders for attendance at healthcare appointments. **Cochrane Database of Systematic Reviews**, n. 12, p. CD007458, 2013.
- HOPPEN, J. **Medical Appointment No Shows**. 2016. <https://www.kaggle.com/datasets/joniarroba/noshowappointments>. Acesso em: 15 jun. 2026.
- MBUNGE, E. et al. Application of machine learning techniques for predicting child mortality and identifying associated risk factors. In: **Conference on Information Communication Technology and Society (ICTAS)**. [S.l.: s.n.], 2023. Acesso em: 10 jun. 2024.
- MCCARTHY, J. et al. **A proposal for the Dartmouth summer research project on artificial intelligence**. 1955.
- Medicina S/A. **Clínicas que não utilizam tecnologia têm crescimento de 8,5% em absenteísmo**. 2024. <https://medicinasa.com.br/confirmacao-consultas/>. Acesso em: 21 mar. 2024.

Microsoft Corporation. **Azure architecture icons**. 2025. <https://learn.microsoft.com/en-us/azure/architecture/icons/>. Versão 23. Acesso em: 29 maio 2026.

RUSSELL, S.; NORVIG, P. **Artificial Intelligence: A Modern Approach**. 3. ed. Harlow: Pearson, 2016.

Santa Casa de Curitiba. **Santa Casa de Curitiba reduz taxa de absenteísmo em consultas do ambulatório SAS**. 2024. <https://santacasacuritiba.com.br/saude-e-bem-estar/santa-casa-de-curitiba-reduz-taxa-de-absenteismo-em-consultas-do-ambulatorio-sas/>. Acesso em: 2 jul. 2024.

SARAVANOU, A. et al. Predictive modeling of infant mortality. **Data Mining and Knowledge Discovery**, 2021. Acesso em: 10 jun. 2024.

Secretaria de Estado da Saúde de Alagoas. **Absenteísmo: falta em consultas, exames e cirurgias marcadas compromete a saúde pública alagoana**. 2024. <https://www.saude.al.gov.br/absenteismo-falta-em-consultas-exames-e-cirurgias-marcadas-compromete-a-saude-publica-alagoana/>. Acesso em: 2 jun. 2026.

Simple Icons. **Simple Icons: free SVG icons for popular brands**. 2025. <https://simpleicons.org/>. Acesso em: 29 maio 2026.

TOPOL, E. J. High-performance medicine: the convergence of human and artificial intelligence. **Nature Medicine**, v. 25, p. 44–56, 2019.

APÊNDICE A – ARTIGO SBC

Aplicabilidade de Modelos de Aprendizado de Máquina para Previsão de Absenteísmo em Serviços de Saúde

William Silveira Figueiredo, Eduardo Camilo Inacio

Departamento de Informática e Estatística – Universidade Federal de Santa Catarina (UFSC)
Florianópolis – SC – Brasil

`william.silveira@grad.ufsc.br, eduardo.camilo@ufsc.br`

Abstract. Absenteeism is one of the main metrics that impact costs in institutions across sectors. In healthcare, where patient non-attendance is high, it produces long queues for appointments, exams and procedures. This work compares two machine learning models, logistic regression and decision tree, for predicting absenteeism, aiming to provide practical guidance for developers of predictive models in the health sector. From a public appointment dataset, 243 versions were generated from different preprocessing decisions and combined with each model's configurations, totaling more than eleven thousand experiments evaluated with metrics suited to imbalanced data (AUC, precision, recall and F1), together with a comparison against simple baselines and a decision-threshold analysis. Logistic regression outperformed the decision tree, reaching a modest but stable performance (AUC around 0.60 on unseen data), whereas the tree showed strong overfitting and near-chance performance outside the training set. More than a ready-to-use model, the work delivers a controlled and reproducible assessment of what these models can, and cannot, offer to support absenteeism management.

Resumo. O absenteísmo é uma das principais métricas que impactam o custo em instituições nos mais diversos setores. Na saúde, onde há alta taxa de não comparecimento, formam-se grandes filas para consultas, exames e procedimentos. Este trabalho compara dois modelos de aprendizado de máquina, regressão logística e árvore de decisão, para prever o absenteísmo, buscando fornecer subsídios para desenvolvedores de modelos preditivos no setor de saúde. A partir de uma base pública de agendamentos, foram geradas 243 versões do conjunto de dados a partir de diferentes decisões de pré-processamento, combinadas com as configurações de cada modelo, totalizando mais de onze mil experimentos, avaliados por métricas apropriadas a dados desbalanceados (AUC, precisão, recall e F1), além da comparação com estratégias simples de referência e da análise do limiar de decisão. A regressão logística superou a árvore de decisão, com desempenho modesto, porém estável (AUC em torno de 0,60 em dados nunca vistos), enquanto a árvore apresentou forte sobreajuste e desempenho próximo ao acaso fora do treino. Mais do que um modelo pronto para uso, o trabalho entrega uma avaliação controlada e reproduzível do que esses modelos conseguem, e do que não conseguem, oferecer como apoio à gestão do absenteísmo.

1 Introdução

O absenteísmo, entendido como o não comparecimento do paciente a uma consulta, exame ou procedimento previamente agendado, é um problema recorrente e custoso para os sistemas de saúde [Beltrame et al., 2019]. Cada horário reservado e não utilizado representa capacidade ociosa: profissionais, salas e equipamentos ficam parados enquanto outros pacientes

aguardam em filas [Dantas et al., 2018]. No sistema público brasileiro, onde a demanda supera com folga a oferta, esse desperdício se traduz em atraso no acesso e em elevação do custo por atendimento efetivamente realizado [Sec. Saúde Alagoas, 2024].

Diante da disponibilidade crescente de dados de agendamento e do amadurecimento das técnicas de aprendizado de máquina, surge a expectativa de que seja possível *antecipar* quais agendamentos têm maior risco de falta, permitindo que a gestão atue de forma direcionada, por exemplo com lembretes ou reorganização da agenda [Gurol-Urganci et al., 2013]. Este trabalho investiga, de forma controlada, até que ponto essa expectativa se sustenta com modelos clássicos e interpretáveis.

O objetivo geral é investigar a aplicabilidade de modelos de aprendizado de máquina para a previsão de absenteísmo em serviços de saúde, comparando o desempenho de dois modelos interpretáveis, regressão logística e árvore de decisão, aplicados a dados reais de agendamento, a fim de avaliar o que cada abordagem oferece como potencial apoio à gestão, na etapa preditiva do problema. Especificamente, busca-se: (i) preparar os conjuntos de dados dos experimentos a partir de diferentes decisões de pré-processamento; (ii) desenvolver os modelos preditivos; (iii) validá-los sob métricas apropriadas a dados desbalanceados; e (iv) propor recomendações para o desenvolvimento de modelos preditivos na área da saúde.

O restante deste artigo organiza-se como segue. A Seção 2 discute trabalhos relacionados. A Seção 3 apresenta a fundamentação teórica. A Seção 4 descreve a metodologia. A Seção 5 expõe e discute os resultados. A Seção 6 apresenta as conclusões, limitações e trabalhos futuros.

2 Trabalhos Relacionados

A previsão de não comparecimento (*no-show*) é um tema consolidado na literatura. Revisões sistemáticas mapeiam dezenas de estudos que aplicam desde regressão logística até métodos de *ensemble* para estimar o risco de falta, com desempenho preditivo que varia bastante conforme a base e o contexto [Dantas et al., 2018, Carreras-García et al., 2020]. De modo geral, esses trabalhos apontam que variáveis como o intervalo de espera entre o agendamento e a consulta, o histórico de faltas e o envio de lembretes estão entre os fatores mais associados ao absenteísmo.

No campo mais amplo do uso de aprendizado de máquina em saúde, há aplicações em predição de mortalidade e identificação de fatores de risco [Mbunge et al., 2023, Saravanou et al., 2021], refletindo a convergência entre inteligência artificial e medicina apontada na literatura [Topol, 2019]. Intervenções de baixo custo, como o envio de mensagens de texto de lembrete, também têm evidência de efeito sobre o comparecimento [Gurol-Urganci et al., 2013].

Este trabalho diferencia-se por não buscar o melhor modelo possível, e sim por conduzir uma avaliação ampla e reprodutível de dois modelos interpretáveis sobre uma mesma base pública, variando sistematicamente as decisões de pré-processamento e reportando explicitamente o que os modelos conseguem e o que não conseguem entregar frente a estratégias triviais.

3 Fundamentação Teórica

O aprendizado supervisionado busca, a partir de exemplos rotulados, uma função capaz de mapear atributos de entrada em uma saída de interesse [Russell and Norvig, 2016, Bishop, 2006]. Neste trabalho a saída é binária (o paciente falta ou comparece), caracterizando um problema de classificação.

A *regressão logística* modela a probabilidade da classe positiva como uma função linear dos atributos passada por uma sigmoide, resultando em coeficientes interpretáveis [Géron, 2019]. A *árvore de decisão* particiona recursivamente o espaço de atributos por meio de perguntas do tipo limiar, oferecendo regras legíveis, mas sujeita a sobreajuste (*overfitting*) quando cresce em profundidade [Géron, 2019, Goodfellow et al., 2016].

Em bases desbalanceadas, a acurácia é enganosa: um modelo que sempre prevê a classe majoritária pode ter acurácia alta e utilidade nula. Por isso adotam-se métricas como precisão, *recall*, F1 e, sobretudo, a área sob a curva ROC (AUC), que independe de um limiar específico e mede a capacidade de *ordenar* os casos por risco [Géron, 2019].

4 Metodologia

4.1 Base de dados e pré-processamento

Utilizou-se a base pública *Medical Appointment No Shows*, coletada em Vitória (ES) em 2016, com 110.527 registros de agendamentos e 14 variáveis [Hoppen, 2016]. A variável alvo indica se o paciente faltou. A base é desbalanceada: as faltas correspondem a cerca de 30% dos registros e os comparecimentos a cerca de 70%.

Os identificadores foram descartados e variáveis derivadas foram criadas a partir das datas de agendamento e de atendimento, com destaque para o intervalo de espera (*WaitingDays*), o dia da semana e o período do dia. Registros com intervalo de espera negativo foram removidos. Os dados foram divididos de forma estratificada, isolando primeiro o conjunto de teste e, em seguida, separando treino e validação, de modo a preservar a distribuição das classes e evitar vazamento de informação do teste.

Para investigar o efeito das decisões de pré-processamento, definiram-se cinco fatores com três opções cada: balanceamento (sem balanceamento, *SMOTE* ou *Random Undersampler*), limiar de agrupamento de bairros, categorização de idade, categorização da faixa de espera e conjunto de *features* de interação. A combinação exaustiva desses fatores gera $3^5 = 243$ versões do conjunto de dados.

4.2 Modelos e protocolo experimental

Cada família de modelo recebeu a codificação adequada: *One-Hot Encoding* e normalização por *z-score* para a regressão logística; *Label Encoding* para a árvore de decisão. Os hiperparâmetros foram varridos em 32 combinações para a regressão logística (*solver*, penalidade, *C* e *l1_ratio*) e 14 para a árvore (critério e profundidade máxima). O cruzamento das 243 versões com as combinações de hiperparâmetros resultou em $243 \times 32 = 7.776$ experimentos de regressão logística e $243 \times 14 = 3.402$ de árvore de decisão, totalizando 11.178 execuções. Toda a aleatoriedade foi fixada com semente 42, garantindo reprodutibilidade.

Pelo volume de execuções, os experimentos foram orquestrados em uma infraestrutura de contêineres sobre um *cluster* Kubernetes gerenciado, com *deployments* dedicados por modelo, conforme a Figura 1. As métricas de avaliação (acurácia, precisão, *recall*, F1 e AUC) foram escolhidas por sua adequação a dados desbalanceados, adotando-se a AUC como métrica principal por independe do limiar de decisão.

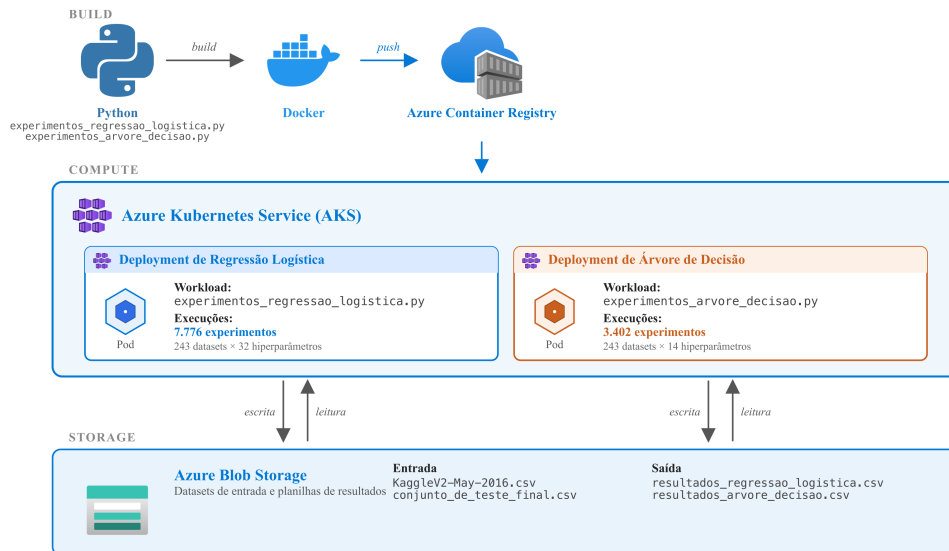


Figura 1. Arquitetura de execução dos experimentos em contêineres sobre *cluster* gerenciado

5 Resultados e Discussão

5.1 Desempenho geral

A Tabela 1 resume a AUC das duas famílias. A regressão logística manteve desempenho estável entre treino, validação e teste, com mediana em torno de 0,60 no conjunto de teste. A árvore de decisão, ao contrário, exibiu forte sobreajuste: mediana de AUC de 0,96 no treino, que desaba para cerca de 0,52 (próximo ao acaso) no teste.

Tabela 1. Visão geral da AUC por família de modelo (medianas e melhor valor no teste)

Família	Exper.	Treino	Validação	Teste (med.)	Teste (máx.)
Regressão Logística	7.776	0,617	0,613	0,595	0,604
Árvore de Decisão	3.402	0,963	0,576	0,522	0,588

A diferença de comportamento é nítida na distribuição sobre o teste: 100% das configurações de regressão logística superam AUC 0,55 e 20% superam 0,60, enquanto, entre as árvores, apenas 23% superam 0,55 e *nenhuma* alcança 0,60. Isso indica que o alto desempenho da árvore no treino tem pouco significado prático.

5.2 Comparação com estratégias simples

Para dimensionar o ganho real dos modelos, compararam-se três estratégias triviais com os melhores modelos de cada família (limiar 0,50), como mostra a Tabela 2. A estratégia de “prever que todos comparecem” atinge a maior acurácia (72%), mas é inútil, pois não identifica nenhuma falta (*recall* zero). A estratégia de “ligar para todos” obtém F1 (0,438) praticamente igual ao da regressão logística (0,435).

Tabela 2. Comparação entre estratégias simples e os melhores modelos no conjunto de teste

Estratégia	Acurácia	Precisão	Recall	F1	AUC
Prever que todos comparecem	0,720	0,000	0,000	0,000	0,500
Escolha aleatória	0,500	0,280	0,500	0,359	0,500
Ligar para todos	0,280	0,280	1,000	0,438	0,500
Regressão Logística (v77)	0,564	0,342	0,600	0,435	0,603
Árvore de Decisão (v304)	0,544	0,322	0,569	0,411	0,564

A única métrica em que os modelos superam de forma consistente as estratégias triviais é a AUC (0,603 e 0,564 contra 0,500). Ou seja, o real valor agregado dos modelos está na capacidade de *ordenar* os pacientes por risco: com AUC de 0,603, dado um par formado por um faltante e um comparecente, o modelo atribui maior risco ao faltante em cerca de 60% das vezes. É um ganho real, mas modesto.

5.3 Análise do limiar de decisão

Como a AUC independe do limiar, analisou-se o efeito de variá-lo entre 0,05 e 0,95 para o melhor modelo. Não existe um limiar universalmente ótimo: reduzi-lo aumenta o *recall* (captura mais faltas) à custa da precisão, e vice-versa. O F1 do melhor modelo atinge o máximo (0,448) em torno do limiar 0,45, valor essencialmente igual ao da estratégia de ligar para todos. A escolha do ponto de operação é, portanto, uma decisão de gestão, ligada ao custo relativo entre um falso negativo (uma falta não antecipada) e um falso positivo (uma ação desnecessária), e não uma questão puramente técnica.

5.4 Limitações e considerações éticas

Os resultados devem ser lidos à luz das limitações da base. Os dados cobrem apenas três meses de um único município, em período anterior à pandemia de COVID-19, o que restringe a generalização temporal e geográfica, sobretudo diante da expansão da telessaúde ocorrida desde então [Caetano et al., 2020]. Além disso, poucas variáveis carregam sinal socioeconômico, o que ajuda a explicar o teto de desempenho observado (AUC em torno de 0,60): boa parte do que determina a falta não está representada nos dados.

Há ainda um risco ético relevante. Variáveis sensíveis, como indicador de programa social e bairro de residência, funcionam como *proxy* de vulnerabilidade. Um modelo que se apoie nelas pode reproduzir desigualdades estruturais: em uma política de *overbooking*, os pacientes sinalizados como prováveis faltantes tendem a ser os mais vulneráveis, de modo que a decisão final não deveria ser delegada ao modelo.

6 Conclusão

Este trabalho conduziu uma avaliação ampla e reprodutível (243 versões de dados e 11.178 experimentos) da aplicabilidade de dois modelos interpretáveis à previsão de absentismo em saúde. A regressão logística superou a árvore de decisão, não por ser excelente, mas por ser modesta e estável (AUC em torno de 0,60 no teste), enquanto a árvore sobreajustou fortemente. O principal ganho dos modelos sobre estratégias triviais está na ordenação dos pacientes por risco, e é modesto; o limiar de decisão, por sua vez, é uma escolha de gestão.

Mais do que um modelo pronto para uso, o trabalho entrega um retrato honesto do que essas abordagens conseguem oferecer, e mapeia o que falta. Como trabalhos futuros, destacam-

se: avaliar uma base brasileira atual e mais rica em variáveis, idealmente em parceria com uma rede de saúde; testar famílias de modelos mais robustas, como *ensembles* do tipo *Random Forest*, além de redes Bayesianas e mapas auto-organizáveis de Kohonen; e acoplar o modelo a uma política de intervenção, avaliada por experimento controlado que meça a redução real do absenteísmo.

Referências

- [Beltrame et al., 2019] BELTRAME, S. M. et al. Absenteísmo de usuários como fator de desperdício: desafio para sustentabilidade em sistema universal de saúde. *Saúde em Debate*, v. 43, n. 123, p. 1015–1030, 2019.
- [Bishop, 2006] BISHOP, C. M. *Pattern Recognition and Machine Learning*. New York: Springer, 2006.
- [Caetano et al., 2020] CAETANO, R. et al. Desafios e oportunidades para telessaúde em tempos da pandemia pela COVID-19. *Cadernos de Saúde Pública*, v. 36, n. 5, e00088920, 2020.
- [Carreras-García et al., 2020] CARRERAS-GARCÍA, D. et al. Patient No-Show Prediction: A Systematic Literature Review. *Entropy*, v. 22, n. 6, p. 675, 2020.
- [Dantas et al., 2018] DANTAS, L. F. et al. No-shows in appointment scheduling – a systematic literature review. *Health Policy*, v. 122, n. 4, p. 412–421, 2018.
- [Géron, 2019] GÉRON, A. *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow*. 2. ed. Sebastopol: O’Reilly Media, 2019.
- [Goodfellow et al., 2016] GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. *Deep Learning*. Cambridge: MIT Press, 2016.
- [Gurol-Urganci et al., 2013] GUROL-URGANCİ, I. et al. Mobile phone messaging reminders for attendance at healthcare appointments. *Cochrane Database of Systematic Reviews*, n. 12, CD007458, 2013.
- [Hoppen, 2016] HOPPEN, J. *Medical Appointment No Shows*. Kaggle, 2016. Disponível em: <https://www.kaggle.com/datasets/joniarroba/noshowappointments>.
- [Mbunge et al., 2023] MBUNGE, E. et al. Application of machine learning techniques for predicting child mortality and identifying associated risk factors. In: *Conference on Information Communication Technology and Society (ICTAS)*, 2023.
- [Russell and Norvig, 2016] RUSSELL, S.; NORVIG, P. *Artificial Intelligence: A Modern Approach*. 3. ed. Harlow: Pearson, 2016.
- [Saravanou et al., 2021] SARAVANOU, A. et al. Predictive modeling of infant mortality. *Data Mining and Knowledge Discovery*, 2021.
- [Sec. Saúde Alagoas, 2024] SECRETARIA DE ESTADO DA SAÚDE DE ALAGOAS. Absenteísmo: falta em consultas, exames e cirurgias marcadas compromete a saúde pública alagoana. 2024.

[Topol, 2019] TOPOL, E. J. High-performance medicine: the convergence of human and artificial intelligence. *Nature Medicine*, v. 25, p. 44–56, 2019.