



UNIVERSIDADE FEDERAL RURAL DO SEMI-ÁRIDO
PRÓ-REITORIA DE PESQUISA E PÓS-GRADUAÇÃO
DEPARTAMENTO DE CIÊNCIAS EXATAS E TECNOLOGIA DA INFORMAÇÃO
BACHARELADO EM SISTEMAS DE INFORMAÇÃO

LUIZ FERNANDO DA CUNHA SILVA

**PREVISÃO DA DIABETES USANDO TÉCNICAS DE *MACHINE LEARNING*: UMA
ANÁLISE COMPARATIVA DE MODELOS E APLICAÇÃO PRÁTICA**

ANGICOS-RN

2025

LUIZ FERNANDO DA CUNHA SILVA

**PREVISÃO DA DIABETES USANDO TÉCNICAS DE *MACHINE LEARNING*: UMA
ANÁLISE COMPARATIVA DE MODELOS E APLICAÇÃO PRÁTICA**

Trabalho de Conclusão de Curso apresentado à
Universidade Federal Rural do Semi-Árido como
pré-requisito para cumprimento do componente
curricular Trabalho de Conclusão de Curso II.

Orientadora: Profa. Dra. Samara Martins
Nascimento Gonçalves

ANGICOS–RN

2025

© Todos os direitos estão reservados a Universidade Federal Rural do Semi-Árido. O conteúdo desta obra é de inteira responsabilidade do (a) autor (a), sendo o mesmo, passível de sanções administrativas ou penais, caso sejam infringidas as leis que regulamentam a Propriedade Intelectual, respectivamente, Patentes: Lei nº 9.279/1996 e Direitos Autorais: Lei nº 9.610/1998. O conteúdo desta obra tomar-se-á de domínio público após a data de defesa e homologação da sua respectiva ata. A mesma poderá servir de base literária para novas pesquisas, desde que a obra e seu (a) respectivo (a) autor (a) sejam devidamente citados e mencionados os seus créditos bibliográficos.

S586p Silva, Luiz Fernando da Cunha.
Previsão da diabetes usando técnicas de
machine learning: uma análise comparativa de
modelos e aplicação prática / Luiz Fernando da
Cunha Silva. - 2025.
102 f. : il.

Orientadora: Samara Martins Nascimento
Gonçalves.
Monografia (graduação) - Universidade Federal
Rural do Semi-árido, Curso de Sistemas de
Informação, 2025.

1. Inteligência Artificial. 2. Machine
Learning. 3. Diabetes Mellitus. 4. Predição. I.
Gonçalves, Samara Martins Nascimento, orient. II.
Título.

Ficha catalográfica elaborada por sistema gerador automático em conformidade
com AACR2 e os dados fornecidos pelo(a) autor(a).

Biblioteca Campus Angicos
Bibliotecário: Helder Romero Maia Duarte
CRB: 15/673

O serviço de Geração Automática de Ficha Catalográfica para Trabalhos de Conclusão de Curso (TCC's) foi desenvolvido pelo Instituto de Ciências Matemáticas e de Computação da Universidade de São Paulo (USP) e gentilmente cedido para o Sistema de Bibliotecas da Universidade Federal Rural do Semi-Árido (SISBI-UFERSA), sendo customizado pela Superintendência de Tecnologia da Informação e Comunicação (SUTIC) sob orientação dos bibliotecários da instituição para ser adaptado às necessidades dos alunos dos Cursos de Graduação e Programas de Pós-Graduação da Universidade.

LUIZ FERNANDO DA CUNHA SILVA

**PREVISÃO DA DIABETES USANDO TÉCNICAS DE *MACHINE LEARNING*: UMA
ANÁLISE COMPARATIVA DE MODELOS E APLICAÇÃO PRÁTICA**

Trabalho de Conclusão de Curso apresentado à
Universidade Federal Rural do Semi-Árido como
pré-requisito para cumprimento do componente
curricular Trabalho de Conclusão de Curso II.

Defendida em: 10/10/2025

BANCA EXAMINADORA

Prof. Dra. Samara Martins Nascimento Gonçalves (UFERSA)
Presidente

Prof. Dra. Adriana Mara Guimarães de Farias (UFERSA)
Membro Examinador

Prof. Dr. Reudismam Rolim de Sousa (UFERSA)
Membro Examinador

AGRADECIMENTOS

A jornada até a conclusão deste trabalho foi repleta de desafios, aprendizado e crescimento. Nada disso teria sido possível sem o apoio, a orientação e a inspiração de muitas pessoas, às quais expresso minha mais profunda gratidão. Primeiro, à minha família pelo incentivo e compreensão ao longo de toda minha trajetória acadêmica. Em especial, agradeço aos meus pais, por sempre acreditarem em mim e me apoiarem em cada etapa desse caminho. Aos meus amigos, que estiveram ao meu lado, oferecendo apoio. A minha orientadora, Dra. Samara Martins Nascimento Gonçalves, pela paciência, pelo conhecimento compartilhado e pelos conselhos valiosos que tornaram este trabalho possível. Aos colegas e professores que contribuíram com ideias, discussões e ensinamentos que ampliaram minha visão e me ajudaram a chegar até aqui. E, por fim, a todos que, de alguma forma, fizeram parte dessa caminhada, meu mais sincero agradecimento. Este trabalho é fruto de um esforço coletivo, e levo comigo cada aprendizado e cada momento vivido nessa trajetória.

“Não é meu objetivo surpreendê-los ou chocá-los, mas o modo mais simples de resumir tudo isso é dizer que agora existem no mundo máquinas que pensam, aprendem e criam. Além disso, sua capacidade de realizar essas atividades está crescendo rapidamente até o ponto – em um futuro visível – no qual a variedade de problemas com que elas poderão lidar será correspondente à variedade de problemas com os quais lida a mente humana.”

Herbert Simon, 1957.

RESUMO

O uso de técnicas de *Machine Learning* para a previsão de doenças, como a diabetes, tem se destacado devido à sua capacidade de identificar padrões em grandes volumes de dados médicos e auxiliar na tomada de decisões clínicas. A diabetes é uma condição crônica que, se não for diagnosticada e tratada precocemente, pode levar a complicações graves. Dessa forma, prever o seu desenvolvimento em estágios iniciais, com base em fatores como: idade, histórico familiar, hábitos de vida e exames laboratoriais, pode contribuir significativamente para a gestão da saúde e a personalização do tratamento. Neste contexto, este trabalho investiga a aplicação de algoritmos de *Machine Learning* na análise de dados médicos para prever a ocorrência da doença. Além disso, esta pesquisa enfatiza a importância da qualidade dos dados, da seleção de características relevantes e da interpretabilidade dos modelos, fatores essenciais para a confiabilidade das previsões e seu impacto nas decisões médicas. De forma geral, a pesquisa compara o desempenho de diferentes algoritmos, analisando a contribuição de cada variável utilizada e destacando os fatores mais influentes na previsão da diabetes. Dessa forma, foram desenvolvidos modelos que, além de eficazes, fornecem *insights* que podem aprimorar abordagens preditivas no contexto da saúde. Seguindo esse norte, o estudo alcançou, como resultado, uma acurácia de 99% para o modelo *Random Forest*. Por fim, foi desenvolvida uma API, a Diagly, que integra o modelo *Random Forest* a sistemas de saúde, permitindo que profissionais da área utilizem a ferramenta para auxiliar no diagnóstico e monitoramento da diabetes. Essa API facilitará, então, o acesso aos resultados das previsões, tornando o uso dos modelos mais prático e acessível.

Palavras – chave: Inteligência Artificial, *Machine Learning*; Diabetes Mellitus; Predição.

ABSTRACT

The use of Machine Learning techniques for predicting diseases such as diabetes has gained prominence due to their ability to identify patterns in large volumes of medical data and support clinical decision-making. Diabetes is a chronic condition that, if not diagnosed and treated early, can lead to severe complications. Therefore, predicting its development at early stages based on factors such as age, family history, lifestyle habits, and laboratory tests can significantly contribute to health management and treatment personalization. In this context, this study investigates the application of Machine Learning algorithms in analyzing medical data to predict the occurrence of the disease. Furthermore, this research emphasizes the importance of data quality, the selection of relevant features, and model interpretability, essential factors for ensuring the reliability of predictions and their impact on medical decisions. Additionally, the study compares the performance of different algorithms, analyzing the contribution of each variable used and highlighting the most influential factors in diabetes prediction. As a result, models were developed that are not only effective but also provide insights that can enhance predictive approaches in healthcare. Following this direction, the present study achieved an accuracy of 99% with the Random Forest model. Finally, an API, called Diagly, was developed to integrate the Random Forest model into healthcare systems, enabling professionals to use this tool to assist in the diagnosis and monitoring of diabetes. This API will facilitate access to prediction results, making the use of these models more practical and accessible.

Keywords: Artificial Intelligence; Machine Learning; Diabetes *Mellitus*; Prediction.

LISTA DE ABREVIATURAS E SIGLAS

AB	Ada Boost
ADO	Árvore de Decisão Otimizada
AMIG	API do Modelo de IA Generativa
ANN	Artificial Neural Networks
API	Application Programming Interface
AUC	Área Sob a Curva ROC
BRFSS	Behavioral Risk Factor Surveillance System
C	Parâmetro de Regularização
CCM	Coefficiente de Correlação de Matthews
CDC	Centers for Disease Control and Prevention
DJ	Decision Jungle
DMG	Diabetes <i>Mellitus</i> Gestacional
DM1	Diabetes <i>Mellitus</i> Tipo 1
DM2	Diabetes <i>Mellitus</i> Tipo 2
DT	Decision Tree
GB	Gradient Boosting
IA	Inteligência Artificial
IDF	Federação Internacional de Diabetes
IQR	Intervalo Interquartil
IMC	Índice de Massa Corporal
JWT	JSON Web Tokens
LLM	Large Language Model
LB	LogitBoost
LDD	LMCH Diabetes Dataset
LightGBM	Light Gradient Boosting Machine
LR	Logistic Regression
MDP	Processo de Decisão de Markov
ML	Machine Learning
NB	Naive Bayes
NIDDK	National Institute of Diabetes and Digestive and Kidney Diseases
NN	Neural Networks

KNN	K-Nearest Neighbors
O2GDNN	Twice-Growth Deep Neural Network
ORM	Object-Relational Mapper
PIDD	Pima Indian Database
RF	Random Forest
SBD	Sociedade Brasileira de Diabetes
SMML	Serviço do Modelo de ML
SMOTE	Synthetic Minority Over-sampling Technique
SUS	Sistema Único de Saúde
SVM	Support Vector Machine
SMOTE	Synthetic Minority Oversampling Technique
TP	Transformação de Potência
VC	Voting Classifier

LISTA DE FIGURAS

Figura 1 - Hierarquia dos Subcampos da Inteligência Artificial.....	22
Figura 2 - Processo de Treinamento e Teste de um Modelo de Machine Learning.....	23
Figura 3 - Conjunto de Treinamento Rotulado para uma Tarefa de Classificação.....	25
Figura 4 - Exemplo de Regressão Baseada em Aprendizado Supervisionado.....	25
Figura 5 - Exemplo de Clusterização em Espaço Bidimensional.....	26
Figura 6 - Exemplo de Detecção de Anomalias em Dados Bidimensionais.....	27
Figura 7 - Aprendizado Semissupervisionado com Duas Classes.....	27
Figura 8 - Interação Entre o Agente e o Ambiente no Aprendizado por Reforço.....	28
Figura 9 - Planejamento da Pesquisa.....	37
Figura 10 - Tecnologias utilizadas.....	38
Figura 11 - Mapa de Calor de Correlação.....	44
Figura 12 - Distribuição das Variáveis Numéricas.....	45
Figura 13 - Distribuição da variável Urea.....	46
Figura 14 - Distribuição da variável Creatinine Ratio.....	46
Figura 15 - Distribuição da variável Triglycerides.....	47
Figura 16 - Distribuição da variável VLDL.....	47
Figura 17 - Otimização do Número de Características no KNN.....	49
Figura 18 - Matriz de Confusão para o modelo KNN Total.....	52
Figura 19 - Matriz de Confusão para o modelo KNN Otimizado.....	53
Figura 20 - Otimização do Número de Características na SVM.....	54
Figura 21 - Matriz de Confusão para o modelo SVM Total.....	56
Figura 22 - Matriz de Confusão para o modelo SVM Otimizado.....	57
Figura 23 - Otimização do Número de Características na LR.....	58
Figura 24 - Matriz de Confusão para o modelo LR Total.....	60
Figura 25 - Matriz de Confusão para o modelo LR Otimizado.....	61
Figura 26 - Otimização do Número de Características no NB.....	62
Figura 27 - Matriz de Confusão para o modelo NB Total.....	64
Figura 28 - Matriz de Confusão para o modelo NB Otimizado.....	65

Figura 29 - Otimização do Número de Características no GB.....	66
Figura 30 - Matriz de Confusão para o modelo GB.....	68
Figura 31 - Otimização do Número de Características na DT.....	69
Figura 32 - Matriz de Confusão para o modelo DT.....	71
Figura 33 - Estrutura da DT treinada.....	72
Figura 34 - Otimização do Número de Características na RF.....	73
Figura 35 - Matriz de Confusão para o modelo RF.....	76
Figura 36 - Arquitetura da API Diagly.....	81
Figura 37 - Conselho Retornado pelo AMIG Considerando um Diagnóstico Negativo.....	82
Figura 38 - Diagrama de Caso de Uso da API Diagly.....	83
Figura 39 - Diagrama de Sequência Simplificado da API Diagly.....	84

LISTA DE TABELAS

Tabela 1 - Comparação de Estudos Relacionados.....	35
Tabela 2 - Resultados do Treinamento do Algoritmo KNN.....	50
Tabela 3 - Resultados do Treinamento do Algoritmo SVM.....	55
Tabela 4 - Resultados do Treinamento do Algoritmo LR.....	59
Tabela 5 - Resultados do Treinamento do Algoritmo NB.....	63
Tabela 6 - Resultados do Treinamento do Algoritmo GB.....	67
Tabela 7 - Resultados do Treinamento do Algoritmo DT.....	70
Tabela 8 - Resultados do Treinamento do Algoritmo RF.....	74
Tabela 9 - Comparação do desempenho dos modelos de ML treinados com todas as características e com as melhores características (SelectKBest).....	77
Tabela 10 - Comparação entre acurácia obtida e valores reportados em trabalhos anteriores..	78

SUMÁRIO

1. INTRODUÇÃO.....	15
1.1. Justificativa e Problemática.....	15
1.2. Objetivos.....	17
1.2.1 Objetivos Gerais.....	17
1.2.2 Objetivos Específicos.....	17
1.3. Organização do trabalho.....	17
2. FUNDAMENTAÇÃO TEÓRICA.....	19
2.1. Diabetes.....	19
2.2 Inteligência Artificial.....	20
2.3. Machine Learning.....	22
2.3.1 Aprendizado Supervisionado.....	24
2.3.2 Aprendizado Não Supervisionado.....	26
2.3.3 Aprendizado Semissupervisionado.....	27
2.3.4 Aprendizado Por Reforço.....	28
2.4. Trabalhos Relacionados.....	28
3. METODOLOGIA DE PESQUISA.....	36
3.1 Tecnologias utilizadas.....	38
3.1.1 Machine Learning com Python e IA Generativa com Cohere.....	39
3.1.2 API RESTful com Node, TypeScript, Express, PostgreSQL e TypeORM.....	40
4. RESULTADOS.....	43
4.1 Pré-Processamento dos dados para o treinamento dos modelos de ML.....	43
4.2 Treinando modelos de ML.....	48
4.2.1 Treinamento do K-Nearest Neighbors (KNN).....	48
4.2.2 Treinamento da Support Vector Machine (SVM).....	53
4.2.3 Treinamento da Logistic Regression (LR).....	57
4.2.4 Treinamento do Naive Bayers (NB).....	61
4.2.5 Treinamento do Gradient Boosting (GB).....	65
4.2.6 Treinamento da Decission Tree (DT).....	68

4.2.7 Treinamento da Random Forest (RF).....	72
4.3 Discussão dos Resultados.....	76
4.4 Desenvolvimento da API Diagly.....	80
5. CONCLUSÃO E TRABALHOS FUTUROS.....	85
REFERÊNCIAS BIBLIOGRÁFICAS.....	87
APÊNDICE A - HIPERPARÂMETROS CONSIDERADOS PARA OS MODELOS.....	94
APÊNDICE B - MÉTRICAS CONSIDERADAS PARA AVALIAÇÃO DOS MODELOS...	96
APÊNDICE C - REGRAS DE NEGÓCIOS DA API DIAGLY.....	97
APÊNDICE D - REQUISITOS DA API DIAGLY.....	98
APÊNDICE E - PROMPT ENVIADO PARA O LLM.....	100
APÊNDICE F - TEMPOS DE RESPOSTA DOS MÓDULO DA API DIAGLY.....	101

1. INTRODUÇÃO

A diabetes *mellitus* é uma doença crônica ocasionada pela produção insuficiente de insulina pelo pâncreas, resultando em um estado de hiperglicemia que, ao longo do tempo, causa danos graves aos sistemas humanos, como os nervos e vasos sanguíneos (Who, 2024). Além disso, essa condição pode ser classificada em diferentes tipos, dos quais incluem pré-diabetes, Diabetes *Mellitus* Tipo 1 (DM1), Diabetes *Mellitus* Tipo 2 (DM2) e Gestacional, sendo cada uma dessas associadas a diferentes complicações. Nesse contexto, diante da gravidade da doença, é perceptível que a realização de um diagnóstico precoce seja de extrema importância, para que se possa evitar futuras complicações, além de permitir que sejam adotadas medidas preventivas eficientes (Bhat *et al.* 2023).

Conforme exposto por Olisah, Smith e Smith (2022), os métodos tradicionais de diagnóstico da diabetes são demorados, sendo necessária a realização de múltiplos testes que não garantem precisão suficiente. Além disso, segundo Khanam e Foo (2021), a identificação manual da diabetes, técnica comumente realizada, depende da experiência do médico, podendo, então, ser suscetível a erros, em especial, em decorrência da complexidade vista nos padrões dos dados médicos relacionados à doença. Ainda segundo os autores, esse diagnóstico é frequentemente realizado em estágios já avançados, limitando as opções de tratamento preventivo para os pacientes.

Com base nos desafios e limitações apresentados a partir da utilização das técnicas tradicionais para o diagnóstico da diabetes, surge a necessidade da exploração de soluções tecnológicas mais eficazes para a realização dessa tarefa (Olishah, Smith e Smith, 2022). Dentro desse cenário, o uso de técnicas de *Machine Learning* (ML) mostra-se como uma alternativa promissora, uma vez que estas permitem construir modelos preditivos que podem agir como uma ferramenta capaz de auxiliar no diagnóstico precoce. No entanto, para que esses modelos possam ser amplamente utilizados na prática clínica e em sistemas de suporte à decisão médica, é importante que sua disponibilização seja eficiente e acessível. Assim, o desenvolvimento de uma *Application Programming Interface* (API) que permita o acesso ao modelo mais eficaz para a predição da doença é essencial para a melhoria do manejo da doença.

1.1. Justificativa e Problemática

Segundo Zhou *et al.* (2024), a prevalência global da diabetes *mellitus* apresentou um aumento significativo entre 1990 e 2022, atingindo um total estimado de 828 milhões de

adultos com a doença em 2022, representando um crescimento de 630 milhões em relação a 1990. Ademais, os autores também destacam que as prevalências padronizadas por idade aumentaram em 131 países para mulheres e em 155 países para homens nesse período, sendo esse crescimento particularmente expressivo em países de baixa e média renda, com destaque para regiões do sudeste e sul da Ásia, Oriente Médio, norte da África, América Latina e Caribe. Dessa forma, considerando o pensamento de Kahn *et al.* (2009), torna-se perceptível que os países com maiores índices de pobreza não apresentam recursos suficientes para a prestação de cuidados adequados para pacientes com diabetes.

No Brasil, a Sociedade Brasileira de Diabetes (SBD) estima que cerca de 20 milhões de pessoas são portadoras de algum dos tipos de diabetes *mellitus* (Brasil, 2025). Além disso, de acordo com dados da Federação Internacional de Diabetes (IDF), o Brasil ocupa a sexta posição dos casos gerais de diabetes no mundo, demonstrando a alta incidência da doença no contexto nacional (Brasil, 2021). Desse modo, esse cenário reforça a necessidade de estratégias eficazes para o diagnóstico precoce, visando um tratamento adequado da doença, minimizando suas complicações associadas. Ainda nesse panorama, Souza (2020) apresenta que cerca de oito milhões de brasileiros não sabem que são portadores de algum tipo de diabetes, ou seja, muitos dos casos permanecem sem diagnóstico, o que demonstra que o diagnóstico da doença é um dos principais desafios enfrentados pelo sistema de saúde.

Com base no exposto, entende-se que a aplicação de técnicas de ML para a predição da diabetes surge como uma abordagem promissora, possibilitando diagnósticos rápidos e precisos com base em dados clínicos e características de pacientes já diagnosticados (Zou *et al.*, 2018). Sendo assim, embora pesquisas presentes na literatura já tenham abordado o uso de técnicas de ML como ferramenta preditiva para a doença, observa-se uma oportunidade de estudo direcionado a construção de uma API que permita a integração dessas técnicas a sistemas de saúde e aplicações médicas, contribuindo para redução dos casos não diagnosticados, além de auxiliar médicos no processo de tomada de decisão. A partir disso, este trabalho apresenta, como problemática central, a seguinte questão: “*A utilização de técnicas de Machine Learning pode contribuir para a predição mais eficaz da diabetes? Além disso, quais modelos apresentam melhor desempenho nessa tarefa e como a integração do modelo mais eficiente em uma API pode minimizar erros de diagnóstico e facilitar o manejo da doença?*”.

1.2. Objetivos

A seção a seguir apresenta o objetivo geral deste trabalho, além de seus objetivos específicos.

1.2.1 Objetivos Gerais

Como forma de responder à questão central deste trabalho, foi definido um objetivo geral: realizar um estudo comparativo acerca dos modelos de ML, para previsão da diabetes, identificando suas características e desempenho; além de desenvolver uma API, que integre o modelo com melhor desempenho, oferecendo uma solução para a predição da doença, com o intuito de contribuir para a ampliação do uso dessa tecnologia no apoio ao diagnóstico.

1.2.2 Objetivos Específicos

Para atingir o objetivo geral deste trabalho, este foi fragmentado em diferentes objetivos específicos, sendo esses:

- Identificar quais e como as técnicas de ML, no contexto de aprendizado supervisionado por classificação, são utilizadas atualmente para a previsão de diabetes;
- Identificar bases de dados disponíveis de forma pública relacionadas a diabetes;
- Entender o comportamento das variáveis e sua relação com a predição da doença;
- Implementar diferentes algoritmos de ML para classificação dos casos de diabetes;
- Comparar o desempenho dos modelos em termos de acurácia, *recall*, precisão e *F1-Score*¹;
- Desenvolver uma API para integrar o modelo com melhor desempenho treinado, facilitando seu uso em aplicações práticas.

1.3. Organização do trabalho

De modo geral, este trabalho está estruturado em cinco capítulos, incluindo a introdução. O segundo capítulo aborda os principais conceitos relacionados ao tema de pesquisa, incluindo uma revisão sobre diabetes *mellitus*, inteligência artificial, *Machine Learning*, IA Generativa e a apresentação de trabalhos correlatos que exploram a aplicação de algoritmos para a previsão da doença. Já o terceiro capítulo detalha a metodologia adotada para a realização do estudo, incluindo as abordagens científicas e tecnologias utilizadas. O

¹ Visite o Apêndice B para entender as métricas de desempenhos consideradas para estes estudo.

quarto capítulo apresenta os resultados dos treinamentos dos modelos de ML, além da API desenvolvida. Por fim, o quinto capítulo apresenta uma conclusão a respeito deste estudo.

2. FUNDAMENTAÇÃO TEÓRICA

Neste capítulo, serão tratados conceitos utilizados para a construção desta proposta de pesquisa. Sendo assim, nas próximas seções serão elencadas as definições acerca da Diabetes e Técnicas de *Machine Learning*. Além disso, serão apresentados os Trabalhos Relacionados a esta pesquisa, para futuras comparações de resultados.

2.1. Diabetes

A diabetes *mellitus* é uma doença reconhecida pelo aumento dos níveis de glicose no sangue, sendo associada a diferentes alterações metabólicas, as quais podem afetar carboidratos, lipídios e proteínas, além de resultar em diversas complicações vasculares, tanto ao nível micro quanto macro (Inzucchi, 2007). Segundo Kahn *et al.* (2009), a diabetes pode apresentar sintomas típicos, como sede intensa, poliúria, visão turva, emagrecimento, fome excessiva, e, em sua forma mais grave, podem ocorrer cetoacidose ou hiperosmolaridade não-cetótica, que, caso não haja tratamento, podem evoluir para coma e óbito.

A diabetes pode ser classificada com base em seus dois tipos principais, a DM1 e a DM2, além da pré-diabetes e da Diabetes *Mellitus* Gestacional (DMG). Nesse contexto, a DM1, segundo Inzucchi (2007), é determinada pela deficiência decorrente da destruição das células β , responsáveis pela produção de insulina², produzidas no pâncreas, atingindo níveis insignificantes no organismo, podendo resultar na ausência desse hormônio. Já com relação a DM2, Inzucchi (2007) aponta que esta é a manifestação mais comum da doença, representando cerca de 90% dos casos em escala global, apresentando o histórico familiar como uma peça fundamental para o surgimento da doença. Ademais, Inzucchi (2007) caracteriza a DM2 com base em sua resistência à insulina, resultando em um quadro de redução da captação periférica de glicose pelos músculos, além do aumento da produção hepática do açúcar no fígado.

Com relação a pré-diabetes, Milech *et al.* (2014) aponta que esse termo se relaciona aos indivíduos que apresentam alto risco em desenvolver a doença, sendo importante que esse estágio não se refere a doença em si, mas sim a uma condição pré-clínica da diabetes *mellitus*, a qual pode aumentar o risco de alguma complicação da doença, em especial, o de desenvolvimento do DM2. Por fim, Kahn *et al.* (2009) aborda o DMG como sendo a intolerância a carboidratos, a qual está relacionada a uma condição hiperglicêmica, que tem

² A insulina é o hormônio responsável pela regulação da quantidade de glicose no sangue. De modo geral, quando a produção desse hormônio é baixa, o açúcar se acumula no sangue e em outras partes do corpo, provocando a diabetes.

início durante a gravidez. Ademais, o autor apresenta que a diabetes durante a gestação pode acarretar consequências tanto para o feto quanto para a mãe, incluindo o risco de morte fetal intra-uterina, além de anormalidades congênitas.

Com relação à faixa etária relacionada à diabetes, percebe-se que a DM1 ocorre, geralmente, em indivíduos com idade inferior a 30 ou 40 anos, sendo essa tipologia da doença mais comum durante a infância ou adolescência (Inzucchi, 2007). Entretanto, é importante destacar que a doença pode ocorrer em qualquer idade, como apontado por Inzucchi (2007), podendo ser erroneamente diagnosticada como DM2. Em contrapartida, os pacientes que apresentam a DM2 estão, na maior parte dos casos, acima dos 40 anos, porém, devido o aumento das taxas de obesidade e sedentarismo, tem-se observado um aumento nos casos de DM2 em grupos mais jovens (Inzucchi, 2007).

De acordo com Milech *et al.* (2014), o controle glicêmico é um dos principais objetivos do tratamento da diabetes *mellitus*. No entanto, para definir a melhor abordagem terapêutica, é essencial considerar diversos fatores individuais, como o tipo de diabetes, nível de escolaridade, condições socioeconômicas e emocionais, idade do paciente, tempo de evolução da doença e possíveis complicações associadas. Além disso, os autores destacam que, por ir além do uso de medicamentos, o manejo da diabetes é um desafio, exigindo mudanças no estilo de vida que devem ser incorporadas de forma contínua. Como resultado, a adesão ao tratamento muitas vezes é baixa, reforçando a necessidade de estratégias personalizadas e embasadas em evidências científicas.

Ainda em relação ao tratamento da diabetes, o Sistema Único de Saúde (SUS) oferece, gratuitamente, uma série de recursos que auxiliam no controle e no tratamento da doença. Nesse contexto, Enfermeira *et al.* (2024) apontam que o SUS fornece medicamentos de insulinas, como metformina e glibenclamida, e diversos insumos, incluindo seringas, agulhas, aparelhos de glicemia, tiras reagentes, entre outros. Desse modo, para acessar esses serviços, é necessário que o paciente esteja cadastrado em uma Unidade Básica de Saúde (UBS) e siga os protocolos estabelecidos pelo SUS.

2.2 Inteligência Artificial

Bertozzo (2019) aponta a inteligência como uma característica essencial do ser humano, fundamental para a sobrevivência e desenvolvimento da espécie ao longo da história. O autor apresenta, ainda, que essa capacidade tem sido objeto de estudo há séculos, visando a compreensão do processo de pensamento e da forma como o conhecimento é construído.

Nesse contexto, percebe-se que a conceituação do termo inteligência remonta a diferentes épocas e perspectivas filosóficas, sociológicas ou, até mesmo, biológicas; entretanto, todas essas abordagens apontam para a ideia de que a inteligência se relaciona com a capacidade adaptativa, e de resolução de problemas, imposta pelo ambiente.

[A inteligência é] uma capacidade mental bastante geral que, entre outras coisas, envolve a habilidade de raciocinar, planejar, resolver problemas, pensar de forma abstrata, compreender ideias complexas, aprender rápido e aprender com a experiência. [...] o conceito refere-se a uma capacidade mais ampla e mais profunda de compreensão do mundo à sua volta - “pegar no ar”, “pegar” o sentido das coisas ou “perceber” uma coisa (Gottfredson, 1994).

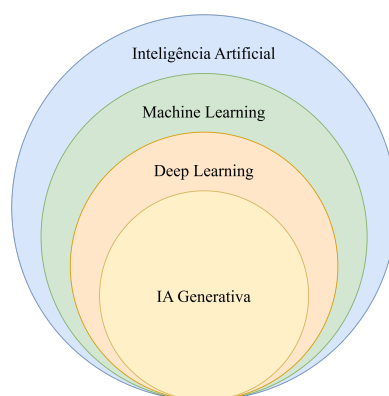
Com base no exposto, é possível definir a Inteligência Artificial (IA) como uma área da ciência da computação que busca desenvolver sistemas computacionais capazes de executar tarefas que necessitam da inteligência humana (Luger, 2013). Nesse contexto, Russell e Norvig (2022) apresentam a IA como o estudo de agentes inteligentes que percebem o seu ambiente e tomam ações, ou decisões, que maximizem suas chances de sucesso. Desse modo, esses agentes conseguem operar em diversos cenários, a partir da aplicação de técnicas computacionais que possam lidar com problemas complexos.

De acordo com Lima (2014), os primeiros conhecimentos a respeito da IA surgiram na década de 1940, durante a Segunda Guerra Mundial. Nesse período, foram notados o surgimento dos primeiros computadores digitais, desenvolvidos inicialmente para questões bélicas. Já em 1950, Alan Turing apresentou a ideia de que uma máquina poderia ser considerada inteligente caso pudesse enganar um humano, fazendo-o acreditar que estava interagindo com outro indivíduo (Lima, 2014). Dessa forma, esse conceito ficou conhecido como Teste de Turing e foi fundamental para o estabelecimento das bases filosóficas e teóricas relacionadas à IA.

Em 1956, John McCarthy, Marvin Minsky, Nathaniel Rochester e Claude Shannon organizaram o *Dartmouth Conference*, um *workshop* considerado o marco inicial dos estudos científicos da IA (Russell e Norvig, 2022). Durante esse evento, o termo “inteligência artificial” foi apresentado pela primeira vez, e os pesquisadores apontaram a possibilidade de criar máquinas que simulassem o raciocínio humano para exercer tarefas como jogos, raciocínio lógico e resolução de problemas. Entretanto, durante as décadas de 1960 e 1970, o progresso foi mais lento, em especial devido a limitações tecnológicas e expectativas fora da realidade, além da ausência de dados suficientes para o treinamento de tecnologias (Russell e Norvig, 2022).

A partir da década de 1990, com o aumento do poder computacional, da disponibilidade de grandes conjuntos de dados e o surgimento do *Machine Learning* (ML), a IA conseguiu restabelecer o seu crescimento. Assim, nas décadas mais recentes, impulsionada por avanços técnicos, como o *Deep Learning* (DL), que permite a criação de modelos capazes de processar dados complexos, a IA atingiu diversas aplicações, as quais incluem desde assistentes virtuais até diagnósticos médicos (Ludermir, 2021). Nesse contexto, possibilita-se abordar a IA em diferentes campos de pesquisa, como demonstrado na Figura 1, gerando uma estrutura hierárquica interligada, apresentando esta tecnologia como um escopo abrangente que inclui várias abordagens que buscam simular a inteligência humana. Dentro desse domínio, apresenta-se o *Machine Learning* (ML), o qual se concentra no aprendizado por meio de dados, a fim de resolver problemas de forma automática. Por sua vez, o DL é uma subárea do ML que utiliza redes neurais profundas para abordar problemas complexos, como a visão computacional. Por fim, a IA Generativa representa o subcampo do DL que emprega modelos, como Redes Geradoras Adversárias, para gerar novo conteúdo a partir de padrões aprendidos previamente (AWS, 2025). Dessa forma, essa estruturação hierárquica demonstra a interligação e o propósito de cada subcampo para a contribuição da resolução de problemas específicos no vasto campo da IA.

Figura 1 - Hierarquia dos Subcampos da Inteligência Artificial



Fonte: Adaptado de AWS (2025)

2.3. *Machine Learning*

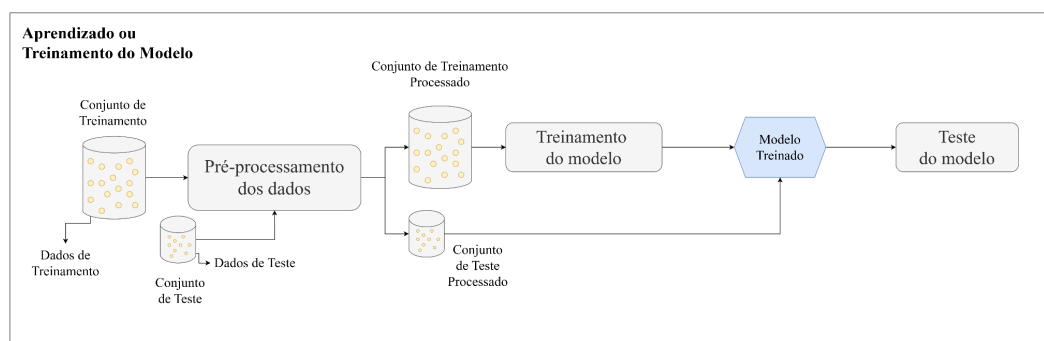
De acordo com Lima (2014), o campo de pesquisa do ML vem sofrendo um grande impulso, em especial, em decorrência do aumento do poder computacional e da difusão das técnicas de IA. Essa área busca, então, desenvolver algoritmos que possam conseguir

aprender determinados padrões a partir de dados, construindo modelos que possam realizar previsões ou classificações sem a necessidade de programação explícita. Assim, o objetivo dos modelos do ML é generalizar bem para novas amostras nunca vistas por estes (Zhou, 2021).

Segundo Zhou (2021), a construção de modelos de ML a partir de dados é denominada como aprendizado ou treinamento, e, durante esse processo, utiliza-se um conjunto específico de informações conhecido como dados de treinamento, onde cada registro representa um exemplo individual que, ao serem agrupados, formam o conjunto de treinamento. Ainda conforme Zhou (2021), após o treinamento de um modelo, é possível avaliar este por meio de testes, para isso, é utilizado um conjunto específico de dados chamado conjunto de teste, composto por amostras destinadas à validação do desempenho do modelo.

O funcionamento do processo de aprendizado de um modelo de ML é descrito na Figura 2. Inicialmente, os dados coletados, que estão divididos em conjunto de treinamento e de teste, são submetidos a uma etapa de pré-processamento, garantindo a qualidade e adequação destes ao modelo. Em seguida, os dados de treinamento já processados são utilizados para alimentar o algoritmo e permitir que ele aprenda os padrões e as relações presentes nesse conjunto, gerando, assim, um modelo treinado. Por fim, esse modelo é avaliado utilizando o conjunto de teste, permitindo a validação da capacidade preditiva deste (Murphy, 2012).

Figura 2 - Processo de Treinamento e Teste de um Modelo de *Machine Learning*



Fonte: Autoria própria

É importante notar que a qualidade dos dados é relevante para a realização do treinamento de um modelo de ML. Géron (2021) aponta que dados que não condizem com a realidade ou que apresentem qualidade inferior podem findar no comprometimento da

capacidade de generalização do modelo para novas predições. Desse modo, aplicar técnicas de limpeza de dados e *feature engineering*³, como apontado por Géron (2021), na etapa de processamento de dados é essencial para garantir que o modelo seja representativo e livre de viés.

Assim, ao garantir a qualidade dos dados utilizados para o treinamento dos modelos, é possível, além da melhora na sua capacidade de generalização, ampliar as aplicações de ML para diversos contextos. De acordo com Müller e Guido (2016), modelos treinados a partir de dados limpos e representativos tendem a ser ferramentas poderosas para problemas complexos, os quais exigem uma maior precisão e múltiplos ajustes. Desse modo, é possível afirmar que a escolha e tratamento adequado dos dados é um parâmetro essencial para a construção de um modelo eficaz.

A partir do exposto e conforme Géron (2021), é possível afirmar que o ML se mostra altamente eficaz em tarefas que envolvem complexidade, ajustes precisos ou uma sequência longa de etapas a serem executadas. Assim, percebe-se que o ML é uma abordagem flexível e poderosa para a resolução de problemas custosos de serem solucionados usando técnicas convencionais de programação, sendo alguns desses tarefas de detecção de padrões, análise preditiva, e, até mesmo, para a automação de processos que envolvam um grande volume de dados. Nesse contexto, o ML pode, então, ser dividido em diferentes domínios, incluindo o aprendizado supervisionado, não supervisionado, semissupervisionado e por reforço (Géron, 2021). As seções a seguir apresentam cada uma dessas abordagens, destacando suas características e aplicações.

2.3.1 Aprendizado Supervisionado

O aprendizado supervisionado, segundo Müller e Guido (2016), é aquele utilizado para prever uma determinada saída a partir de uma entrada fornecida, ou seja, são necessários pares de entrada/saída para a predição. Em outras palavras, aprendizado supervisionado é aquele no qual dados rotulados⁴ são utilizados para o treinamento do modelo de ML, visando que este aprenda a mapear as entradas e as saídas. Ademais, esse aprendizado pode ser dividido em duas categorias: classificação, quando o objetivo é prever uma categoria, ou

³ *Feature engineering* é o método de transformar dados brutos em características que melhorem o desempenho dos modelos de ML. Seu escopo inclui a criação, transformação, seleção e eliminação de *features* para melhorar a capacidade preditiva do modelo (Nargesian *et al.*, 2017).

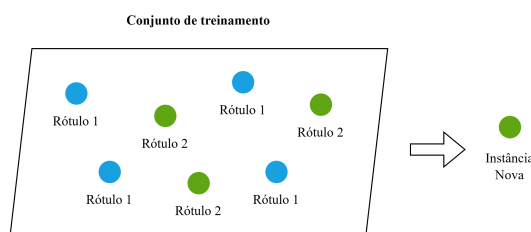
⁴ Dados rotulados são aqueles nos quais cada entrada possui uma saída associada, conhecida como rótulo (Géron, 2021).

classe, para cada entrada; ou regressão, quando se busca prever um valor numérico que seja contínuo (Géron, 2021).

O comportamento de tarefas de classificação, no contexto do aprendizado supervisionado, é mostrado na Figura 3, onde o modelo utiliza dados com rótulos para aprender e separar diferentes classes em um determinado espaço de características. Dessa forma, o modelo consegue prever corretamente as classes de novos dados que não foram vistos durante a etapa de treinamento. Além disso, segundo Zhou (2021), os problemas de classificação podem ser diferenciados com base na quantidade de categorias que precisam ser previstas. Assim, segundo o autor, quando há somente duas possíveis classes, trata-se de uma classificação binária, já nos casos em que existem mais de duas categorias possíveis, o problema é denominado classificação multiclasse.

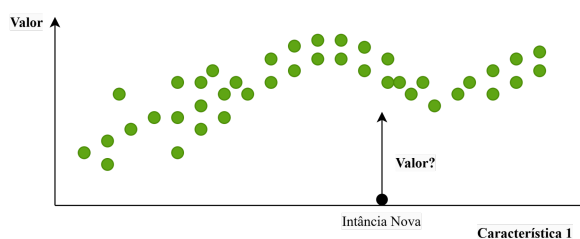
Já o comportamento de tarefas de regressão é baseado na previsão de um valor contínuo para uma nova instância, o qual é realizado a partir da relação encontrada nos dados rotulados de treino. Dessa forma, ao se realizar uma inferência, o ponto “Instância Nova”, o modelo irá estimar um valor esperado com base na tendência dos dados que já foram observados, conforme observado na Figura 4.

Figura 3 - Conjunto de Treinamento Rotulado para uma Tarefa de Classificação



Fonte: Adaptado de Géron (2021)

Figura 4 - Exemplo de Regressão Baseada em Aprendizado Supervisionado



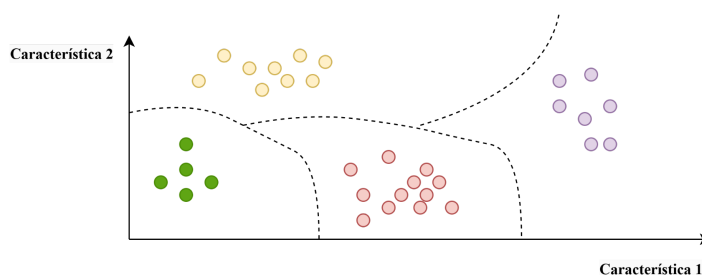
Fonte: Adaptado de Géron (2021)

2.3.2 Aprendizado Não Supervisionado

Segundo Müller e Guido (2016), no aprendizado não supervisionado, o algoritmo analisa somente os dados de entrada, sem informações prévias sobre as respostas esperadas, buscando identificar padrões e estruturas ocultas nos dados. Ou seja, para esse aprendizado, são utilizados dados não rotulados, sem considerar nenhuma orientação externa.

Géron (2021) aponta que as principais aplicações do aprendizado não supervisionado incluem a clusterização, a visualização e redução de dimensionalidade, o aprendizado de regras por associação e a detecção de anomalias. Conforme Géron (2021), a clusterização pretende identificar agrupamentos de dados com características semelhantes, permitindo que o modelo reconheça padrões sem a necessidade de rótulos pré-definidos. O funcionamento dessa técnica é demonstrado na Figura 5, na qual os pontos são agrupados em *clusters* diferentes, representando grupos de dados que compartilham características semelhantes.

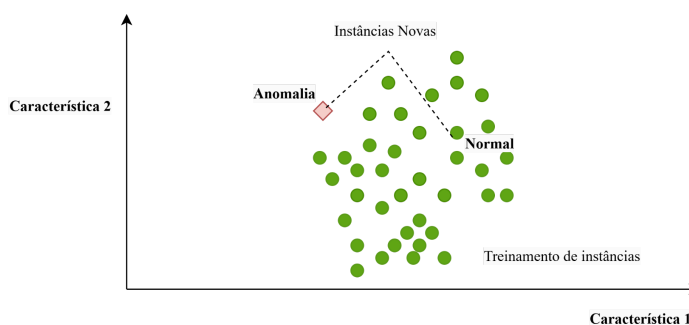
Figura 5 - Exemplo de Clusterização em Espaço Bidimensional



Fonte: Adaptado de Géron (2021)

Os algoritmos de visualização e redução de dimensionalidade buscam simplificar conjuntos de dados complexos, buscando uma representação mais concisa desses dados através da redução de atributos, e sem que haja uma grande perda de informação (Müller e Guido, 2016). Já o aprendizado de regras por associação busca realizar a identificação de relações entre variáveis em grandes conjuntos de dados (Géron, 2021). Por fim, algoritmos de detecção de anomalias buscam identificar padrões incomuns, como *outliers*, que se desviem do comportamento esperado (Géron, 2021). O funcionamento desse método é descrito na Figura 6, que representa um exemplo de detecção de anomalia em um espaço bidimensional, no qual uma instância nova, que não se assemelha aos padrões aprendidos pelo modelo, é detectada como uma anomalia, ou seja, tem um comportamento fora do esperado.

Figura 6 - Exemplo de Detecção de Anomalias em Dados Bidimensionais



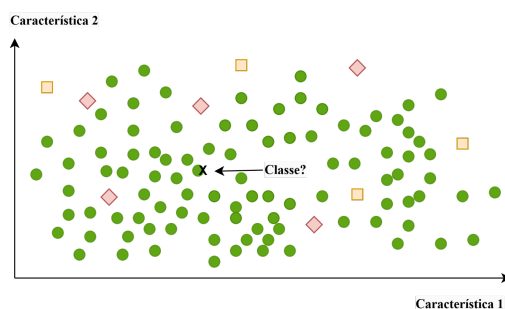
Fonte: Adaptado de Géron (2021)

2.3.3 Aprendizado Semissupervisionado

Conforme Géron (2021), o aprendizado semissupervisionado é uma abordagem utilizada quando somente uma parte dos dados possui rótulos, permitindo que o modelo aproveite tanto amostras rotuladas quanto não rotuladas no processo de aprendizado. Dessa forma, percebe-se que tal estratégia de aprendizado é extremamente útil em situações nas quais a obtenção de dados rotulados é custosa, porém, existe uma abundância de dados não rotulados disponíveis.

O funcionamento desse tipo de aprendizado é descrito na Figura 7, na qual são apresentadas duas classes, a de quadrado e losango, e exemplos não rotulados, os círculos, são utilizados para classificação de uma nova instância (Géron, 2021). Assim, no aprendizado semissupervisionado, o modelo é treinado a partir de uma pequena quantidade de dados rotulados, os quais fornecem informações sobre as saídas esperadas. Em seguida, dados não rotulados são usados para refinar o modelo, buscando padrões e estruturas que podem não ser evidentes nos dados rotulados.

Figura 7 - Aprendizado Semissupervisionado com Duas Classes



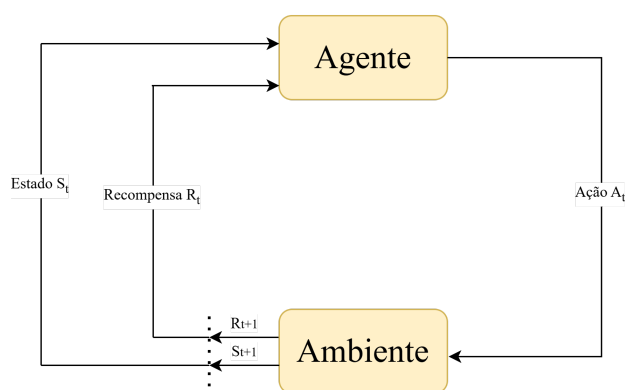
Fonte: Adaptado de Géron (2021)

2.3.4 Aprendizado Por Reforço

Segundo Faceli *et al.* (2021), o aprendizado por reforço é baseado na interação de um agente com o ambiente, no qual ele recebe recompensas quando toma decisões corretas e penalidades quando escolhe ações inadequadas. Ainda conforme os autores, esse mecanismo de *feedback* permite que o agente aprenda a otimizar seu comportamento ao longo do tempo. Desse modo, esse processo de tentativa e erro possibilita que o agente maximize a sua recompensa, findando no ajuste de sua estratégia de forma autônoma. Ademais, a diferenciação dessa aprendizagem para com as demais está no fato de que essa não depende de um conjunto de dados pré-definido, já que o agente aprende diretamente com o ambiente (Sutton e Barto, 2018).

A Figura 8 ilustra o Processo de Decisão de Markov (MDP), um modelo matemático utilizado como base para o aprendizado por reforço que representa a interação entre um agente e um ambiente (Sutton e Barto, 2018). No MDP, o ambiente é configurado como um estado (S_t) no qual o agente pode se encontrar a cada momento. Assim, o agente toma uma ação (A_t) com base no seu estado atual, o que pode resultar em uma transição para um novo estado (S_{t+1}). Após essa execução, o agente recebe uma recompensa (R_t), a qual reflete o sucesso da ação que foi tomada. Nesse contexto, o processo iterativo do agente, maximizando sua recompensa acumulada ao longo do tempo, resulta em uma política que mapeia os melhores resultados para uma tomada de decisão mais eficiente.

Figura 8 - Interação Entre o Agente e o Ambiente no Aprendizado por Reforço



Fonte: Adaptado de Sutton e Barto (2018)

2.4. Trabalhos Relacionados

A utilização de algoritmos de ML para predição de diabetes é um campo bastante amplo da IA, uma vez que é considerado de grande relevância para a saúde pública global

(Santos, 2024). Nesse contexto, diversos trabalhos surgem na literatura explorando diferentes abordagens e técnicas para melhorar a precisão e a eficiência na detecção dessa doença. Sendo assim, esta seção apresenta uma breve revisão de trabalhos relacionados a esta pesquisa, destacando metodologias, bases de dados utilizadas e os resultados obtidos.

A busca por trabalhos relacionados a essa pesquisa foi realizada de forma sistemática, priorizando a seleção de estudos atualizados e relevantes presentes na literatura científica. Para isso, foram utilizadas bases de dados acadêmicas reconhecidas, como a *ScienceDirect*, *IEEE Xplore* e *Google Scholar*. A partir disso, os critérios de inclusão foram definidos, os quais envolvem artigos publicados entre 2020 e 2024, a fim de garantir que os modelos e técnicas utilizadas estivessem alinhados com os avanços recentes da área de pesquisa. Além disso, a estratégia de busca envolveu a formulação de palavras-chave, sendo essas: *Machine Learning*, *Diabetes* e *Predição*, combinadas por operadores booleanos para aumentar a precisão da pesquisa.

Após a recuperação dos artigos, foi realizada uma etapa de triagem com base na leitura de títulos e resumos. Com base nesse critério, foi definido que trabalhos que não estivessem diretamente relacionados ao escopo desse projeto fossem excluídos. A partir disso, os artigos selecionados passaram a ser analisados, considerando suas metodologias, bases de dados e resultados obtidos. Dessa forma, permitiu-se identificar quais os padrões nas abordagens utilizadas, comparar diferentes algoritmos empregados e extrair informações para a condução da presente pesquisa. Por fim, os estudos selecionados foram organizados e sintetizados, destacando os principais achados em relação às bases de dados, métodos de pré-processamento, algoritmos de ML utilizados e desempenho dos modelos treinados. Os trabalhos selecionados são apresentados a seguir.

Soni e Varma (2020), Khanam e Foo (2021), Olisah, Smith e Smith (2022), Febrian *et al.* (2023), Bhat *et al.* (2023) e Hossain *et al.* (2022) utilizaram, em seus trabalhos, a base de dados *Pima Indian Database* (PIDD)⁵, disponibilizada via Kaggle pelo *UCI Machine Learning Repository*, para o treinamento e a avaliação dos modelos de ML. O uso abrangente dessa base de dados se dá pela sua disponibilidade pública, simplicidade e relevância clínica, além de possuir atributos médicos bem definidos, os quais permitem uma análise variada da doença.

O PIDD é originário do *National Institute of Diabetes and Digestive and Kidney Diseases* (NIDDK) dos Estados Unidos e contém informações a respeito de 768 pacientes, de

⁵ <https://www.kaggle.com/datasets/uciml/pima-indians-diabetes-database>

no mínimo 21 anos e do sexo feminino do grupo étnico Pima, com atributos clínicos, como: gravidez, idade, glicose, pressão sanguínea, insulina, índice de massa corporal (IMC), espessura da pele e a variável-alvo *outcome*. Além disso, o conjunto de dados é levemente desbalanceado, com 500 amostras negativas e 268 positivas em relação à doença.

Olisah, Smith e Smith (2022) e Abdelhafez e Amer (2024) utilizaram a base de dados *LMCH Diabetes Dataset* (LDD)⁶, a qual inclui o registro de 1.000 pacientes do *Hospital Medical City*, no Iraque. O diferencial dessa base de dados se dá pela presença de uma classe relacionada ao diagnóstico de pré-diabetes em sua variável-alvo, permitindo uma abordagem mais ampla em relação à identificação da doença. Além disso, o conjunto de dados apresenta 14 atributos, sendo esses: código do paciente, nível de açúcar no sangue, idade, gênero, creatina, IMC, ureia, colesterol, LDL, VLDL, triglicerídeos, colesterol HDL, HBA1C, além da variável-alvo, com as classes: normal, pré-diabetes e diabetes. Essa base de dados se demonstra desbalanceada, com 844 pacientes diabéticos, 103 pré-diabéticos e 53 sem a doença.

Soni e Varma (2020), motivados pela necessidade de prever a diabetes precocemente, propuseram um estudo que utilizou modelos de ML, os quais foram treinados com a base de dados PIDD, para a previsão da doença. As técnicas utilizadas no trabalho incluíram: classificação supervisionada, a partir dos algoritmos *Support Vector Machine* (SVM), *K-Nearest Neighbors* (KNN), *Logistic Regression* (LR), e *Decision Tree* (DT); e métodos de *ensemble*⁷, com *Random Forest* (RF) e *Gradient Boosting* (GB), para melhorar a acurácia combinando modelos mais simples. A principal métrica utilizada foi a acurácia, a qual foi comparada para a identificação do modelo que apresentasse o melhor desempenho. Assim, o modelo baseado em RF foi o de maior destaque, com acurácia de 77%, demonstrando a eficácia das técnicas de *ensemble* em melhorar a previsão de diabetes. Ademais, foi apontado que atributos como glicose e IMC foram os mais relevantes na previsão.

Serra e Nascimento (2022), a partir da percepção do aumento da incidência de DM2 no Brasil e no mundo, associada, principalmente, a maus hábitos de vida, propuseram a construção de um modelo de ML com foco em redes neurais artificiais para a previsão de diabetes. Para isso, o algoritmo *MLPClassifier*, ou Perceptron Multicamadas, foi treinado com dados balanceados para prever casos positivos e negativos da doença com base em características como pressão alta, IMC, nível de atividade física, entre outras. Para esse

⁶ <https://data.mendeley.com/datasets/wj9rwkp9c2/1>

⁷ Aprendizado por *ensemble*, em ML, é aquele que busca combinar diferentes modelos para melhorar o desempenho geral e reduzir erros de predição.

trabalho, os autores utilizaram um recorte da base de dados *Behavioral Risk Factor Surveillance System* (BRFSS), disponível de forma pública via Kaggle, o *Diabetes Health Indicators Dataset*⁸. Essa amostragem apresenta 441.455 registros balanceados, ou seja, 50% são de casos positivos e os outros 50% são negativos, coletados pelo *Centers for Disease Control and Prevention* (CDC) nos Estados Unidos.

A principal métrica utilizada por Serra e Nascimento (2022) foi a acurácia, complementada por precisão, *recall* e *F1-Score*. O tempo de execução do treinamento e teste do modelo também foi registrado. A partir disso, o modelo apresentou, como resultado, uma acurácia média de 75% em 80 segundos de processamento, demonstrando potencial para auxiliar no diagnóstico precoce de diabetes. É importante destacar, ainda, que o conjunto de dados balanceado contribuiu para a consistência dos resultados, com precisão similar para casos positivos e negativos da doença.

Visando uma detecção precoce da diabetes como mecanismo essencial para o controle eficaz da doença, Khanam e Foo (2021) utilizaram um conjunto de sete algoritmos de ML para comparar a eficácia desses na predição de diabetes, incluindo: LR, SVM, *Naive Bayes* (NB), RF, DT, KNN e *AdaBoost* (AB). Além disso, foram implementados modelos de *Artificial Neural Networks* (ANN) com uma, duas e três camadas ocultas para explorar o impacto da arquitetura e dos parâmetros no desempenho. Em relação à base de dados, os modelos foram construídos com base na PIDD, na qual os dados foram normalizados e tratados para remoção de *outliers* e valores ausentes. Foram, ainda, extraídas as melhores características do conjunto de dados, a partir da utilização do método de correlação de Pearson, resultando que a glicose, IMC, nível de insulina, gravidez e idade são as características mais relevantes. Foram empregadas métricas de avaliação baseadas na matriz de confusão, as quais incluem acurácia, precisão, *recall* e *F1-Score*. Desse modo, os melhores desempenhos foram obtidos pelo LR e SVM, com acurácia de aproximadamente 77% a 79%, respectivamente. O ANN com duas camadas ocultas alcançou o melhor desempenho geral, com acurácia de 88,6%.

Olisah, Smith e Smith (2022) buscam melhorar a predição e o diagnóstico de diabetes *mellitus* por meio de ML, dada a alta prevalência da doença e seu grave impacto na saúde pública. Nesse sentido, os autores buscaram desenvolver um *framework* robusto que integrasse técnicas de pré-processamento de dados e modelos de classificação para obter resultados mais precisos e generalizáveis. Nessa perspectiva, o estudo propôs um *framework*

⁸ <https://www.kaggle.com/datasets/alexteboul/diabetes-health-indicators-dataset>

para a aplicação de técnicas de ML para a predição de diabetes com as seguintes etapas: 1) de pré-processamento de dados, através da Correlação de Spearman, utilizada para seleção de características relevantes, e da Regressão polinomial, empregada para imputação de valores ausentes, especialmente em distribuições não normais; 2) de aplicação de modelos de classificação, os quais, no estudo, foram usados o RF, SVM e um modelo baseado em redes neurais profundas, desenvolvido pelos próprios autores, chamado *Twice-Growth Deep Neural Network* (O2GDNN), com múltiplas camadas ocultas e otimizações; e 3) de otimização de hiperparâmetros, implementada via *Grid Search* e validação cruzada estratificada repetida. Em relação às métricas de avaliação, foram utilizadas acurácia, precisão, *recall*, *F1-Score* e especificidade. Assim sendo, e realizando uma comparação dos modelos treinados a partir de duas diferentes bases de dados, a PIDD e a LDD, os autores destacaram o resultado do modelo O2GDNN, o qual obteve uma acurácia de 97.24% no PIDD e 97.33% no LDD.

Visando explorar algoritmos de aprendizado supervisionado como uma alternativa eficiente e precisa para o diagnóstico de diabetes, reduzindo complicações graves associadas à detecção tardia, Febrian *et al.* (2023) realizaram um estudo comparativo entre dois modelos de ML, sendo esses o KNN e o NB, e utilizando a base de dados PIDD. O processo de treinamento abordado no trabalho incluiu o pré-processamento de dados, através da normalização dos valores e preenchimento de dados ausentes, uma validação cruzada *K-Fold*, em 10 partes, para garantir a confiabilidade dos resultados, e a divisão de dados em conjuntos de treinamento e teste em proporções variadas (80:20, 70:30, etc.) até que as 10 partes do *K-Fold* fossem atingidas.

Com base nas métricas acurácia, precisão, *recall* e matriz de confusão, Febrian *et al.* (2023) observaram que o algoritmo NB superou o KNN em todos os experimentos, alcançando uma acurácia média de 76,07%, precisão de 73,37% e *recall* de 71,37%. Em contrapartida, o KNN apresentou acurácia média de 73,33%, com desempenho inferior nas métricas de precisão e *recall*. Além disso, também percebeu-se que o melhor resultado foi obtido quando 80% dos dados foram usados para treinamento.

Chou, Hsu e Chou (2023), fundamentados no aumento do impacto da diabetes em Taiwan, buscaram comparar uma série de algoritmos de ML visando uma detecção precoce da doença como alternativa essencial para mitigar complicações severas, reduzir custos de saúde e melhorar a qualidade de vida. Para isso, o estudo utilizou um conjunto de dados com o registro de 15.000 mulheres, entre 20 e 80 anos, atendidas em um centro médico municipal em Taipei, capital de Taiwan, entre 2018 e 2022. As variáveis incluíram: número de

gestações, glicose plasmática, pressão arterial diastólica, espessura da pele, nível de insulina, IMC, função de pedigree de diabetes e idade. A partir disso, Chou, Hsu e Chou (2023) utilizaram quatro modelos de ML: LR, ANN, *Decision Jungle* (DJ) e Árvore de Decisão Otimizada (ADO). Ademais, as etapas relacionadas ao treinamento dos modelos incluíram o pré-processamento de dados, através da normalização de variáveis contínuas, como glicose plasmática e IMC, a divisão de Dados, com 80% para treinamento e 20% para teste, e uma validação cruzada para garantir a confiabilidade dos resultados. Foram, também, empregadas as seguintes métricas: acurácia, precisão, *recall*, F1-Score e Área Sob a Curva ROC (AUC). Como resultado, observou-se que o modelo ADO apresentou o melhor desempenho, com uma acurácia de 95,3%, F1-Score de 0,929 e AUC com valor de 0,991. O DJ foi o segundo mais eficaz, com AUC de 0,976. Os modelos ANN e LR também obtiveram resultados satisfatórios, mas inferiores.

Bhat *et al.* (2023), motivados pelo aumento global da prevalência de diabetes *mellitus*, comparam três algoritmos supervisionados para predição da doença: LR, GB e DT. Para isso, foi utilizado o dataset PIDD, o qual passou por uma etapa de pré-processamento, a qual incluiu a imputação de valores ausentes, usando a média de atributos, e a detecção e tratamento de outliers com base no intervalo interquartil. Ademais, também foi realizado o processo de balanceamento de classes, implementado por meio da técnica *Synthetic Minority Oversampling Technique* (SMOTE). Para validação, foi utilizada a técnica de Validação Cruzada *K-Fold*, com 10 partes, a fim de reduzir vieses e avaliar a robustez dos modelos. As métricas utilizadas para avaliação dos modelos incluíram: acurácia, precisão, *recall*, F1-Score e AUC. A partir disso, observou-se que o modelo DT apresentou o melhor desempenho, alcançando uma acurácia de 91%, precisão de 96%, *recall* de 92%, F1-Score de 94% e AUC com valor igual a 0,99. Ademais, os algoritmos GB e LR tiveram desempenhos ligeiramente inferiores, com acurácia de 89% e 81%, respectivamente.

Já Abdelhafez e Amer (2024) estiveram fundamentados na necessidade de prever a diabetes em estágios iniciais. Sendo assim, os autores utilizaram técnicas de ML visando melhorar a precisão do diagnóstico da doença, facilitando intervenções precoces que possam reduzir custos e melhorar a qualidade de vida dos pacientes. Foram, então, analisados oito algoritmos de aprendizado supervisionado: NB, DT, LR, RF, SVM, NN, *LogitBoost* (LB) e *Voting Classifier* (VC). Como pré-processamento, foram realizadas as seguintes etapas: tratamento de outliers, usando o método do intervalo interquartil (IQR); e seleção de características, através dos métodos *CfsSubsetEval* e *WrapperSubsetEval* para reduzir a

dimensionalidade. As métricas de avaliação incluíram acurácia, *recall*, especificidade, F1-Score e Coeficiente de Correlação de Matthews (CCM). Com base nisso, os modelos RF e DT apresentaram os melhores desempenhos, com acurácia de 99,67% (RF) e 99,33% (DT) usando sete características selecionadas, F1-Score de 99,7% (RF) e 99,3% (DT) e MCC de 0,988 (RF). Além disso, destacam-se nos resultados o trabalho de remoção dos *outliers*, o que melhorou o desempenho de todos os modelos, especialmente do SVM.

Visando reduzir a necessidade de exames invasivos, Hossain *et al.* (2022) propuseram um aplicativo móvel inteligente baseado em ML para diagnosticar diabetes e realizar o monitoramento diário da doença. Para isso, os autores realizaram um estudo comparativo de dois algoritmos de ML, incluindo *Light Gradient Boosting Machine (LightGBM)* e KNN, treinados a partir da base de dados PIDD. Com base nas métricas acurácia, precisão e AUC, o modelo *LightGBM* obteve um desempenho superior, com uma precisão de 90% e uma área sob a curva ROC (AUC) de 0,948, além de uma acurácia de 92% em uma validação cruzada com 5 *folds*. Além disso, uma análise comparativa evidenciou que os métodos propostos superaram estudos anteriores sobre o diagnóstico de diabetes utilizando o mesmo conjunto de dados. Após o treinamento com o *LightGBM*, esse foi integrado a um aplicativo Android com *backend* em Flask e Heroku, para, então, ser proposto um chatbot para perguntas relacionadas à saúde, módulo de recomendações alimentares, lembretes de medicamentos e monitoramento de atividades físicas utilizando APIs como Google Fit e *Dialogflow*. Em seu estudo Hossain (2022), destaca, então, a importância de integrar ML às aplicações práticas, para melhorar o diagnóstico precoce e o monitoramento contínuo de doenças, com potencial de impacto significativo na saúde pública global.

A Tabela 1 demonstra, de forma geral, os melhores resultados dos trabalhos elencados anteriormente, tomando como base o valor da acurácia. Com esse levantamento, é possível destinar quais serão os algoritmos a serem implementados nesta pesquisa e verificar, usando uma mesma base de dados, quais modelos têm melhor desempenho. Além disso, será possível confrontar os resultados alcançados com os dados já validados nos trabalhos relacionados a este.

Tabela 1 - Comparação de Estudos Relacionados

Referência	Modelo de Destaque	Acurácia (em %)
Abdelhafez e Amer (2024)	<i>Random Forest</i>	99,67
Soni e Varma (2020)	<i>Random Forest</i>	77,00
Olisah, Smith e Smith (2022)	O2GDNN	97.33
Chou, Hsu e Chou (2023)	Árvore de Decisão Otimizada	95,30
Bhat <i>et al.</i> (2023)	<i>Decision Tree</i>	91,00
Febrian <i>et al.</i> (2023)	<i>Naive Bayes</i>	76,07
Hossain <i>et al.</i> (2022)	<i>LightGBM</i>	92,00
Khanam e Foo (2021)	<i>Support Vector Machine</i>	79,00
Serra e Nascimento (2022)	<i>MLPClassifier</i>	75,00

Fonte: Autoria própria.

3. METODOLOGIA DE PESQUISA

Este estudo caracteriza-se como uma pesquisa quantitativa e aplicada, com abordagem experimental. Conforme o exposto por Creswell (2014), a abordagem quantitativa é adotada devido à natureza dos dados analisados para esse estudo, além da necessidade de mensuração das variáveis envolvidas na predição da diabetes. Assim, a pesquisa quantitativa permite, ao se considerar o escopo desta pesquisa, a aplicação de técnicas estatísticas e de ML para a extração de padrões e avaliação do desempenho dos modelos preditivos.

Com relação à classificação aplicada para este trabalho, destacam-se as questões práticas dos conhecimentos abordados e suas consequências, que podem ser justificadas a partir da busca pela geração de conhecimento (Gil, 2008). Nesse sentido, considera-se a aplicação prática na área da saúde, em específico na predição da diabetes. Com base nisso, é perceptível que, diferente de pesquisas puramente teóricas, este estudo busca contribuir para a construção de modelos computacionais que possam ser utilizados em contextos reais, auxiliando profissionais de saúde na tomada de decisão e na identificação precoce de indivíduos em risco.

Também é possível designar um caráter experimental para esta pesquisa, já que envolve a implementação, treinamento e avaliação de modelos de ML utilizando dados históricos de pacientes. Nesse contexto, e com base nos conceitos abordados por Montgomery (2017), percebe-se que os experimentos são conduzidos para testar diferentes algoritmos e configurações, comparando métricas de desempenho, como acurácia, precisão e *recall*. Desse modo, esse tipo de abordagem experimental permite validar a eficácia dos modelos e identificar quais técnicas são mais adequadas para a predição da diabetes.

A metodologia adotada durante as etapas iniciais dessa pesquisa concentrou-se na revisão de trabalhos relacionados existentes na literatura, além da construção de uma fundamentação teórica a respeito de conceitos importantes voltados para a área de pesquisa deste estudo, conforme mostrado na Figura 9. Com base nisso, é possível afirmar que essa etapa é essencial para a compreensão dos desafios, metodologias e técnicas previamente utilizadas na literatura, bem como para embasar a escolha dos métodos a serem empregados nos experimentos futuros.

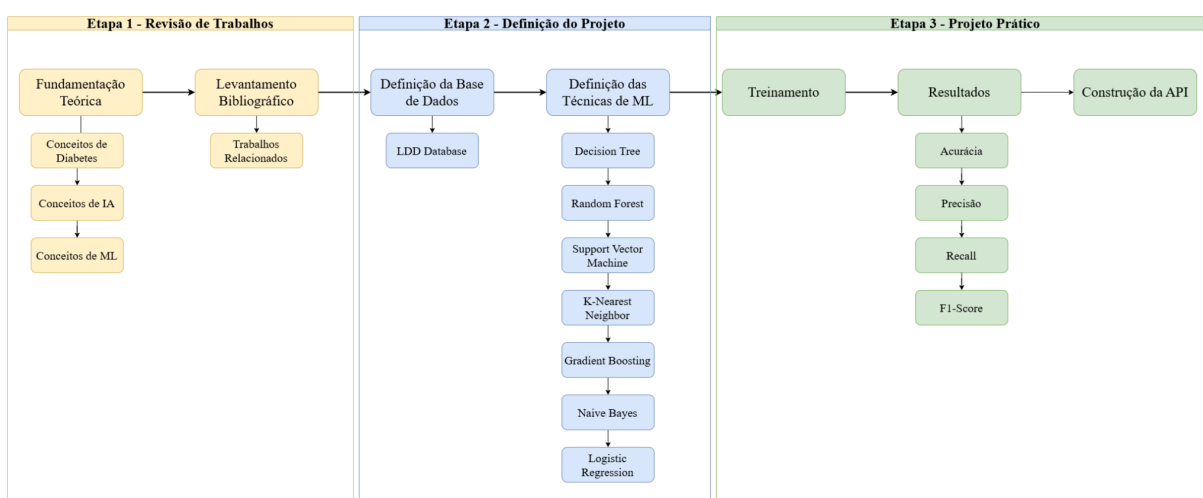
Após a realização da primeira etapa de planejamento para execução deste trabalho (i.e. a realização do levantamento bibliográfico), foi possível identificar os estudos já existentes na literatura, que aplicaram algoritmos de ML para a previsão de diabetes. Essa pesquisa permitiu destacar as abordagens mais eficazes, além das bases de dados mais relevantes.

Nesse contexto, torna-se importante apresentar que, ao decorrer da etapa de levantamento de dados, observou-se uma limitação em relação à disponibilidade de conjuntos de dados públicos que apresentassem atributos clínicos voltados ao diagnóstico da diabetes no contexto nacional, já que as fontes acessíveis apresentam, em sua maioria, somente indicadores estatísticos e epidemiológicos relacionados à doença, sem informações individuais de pacientes, que seria o ideal para o treinamento de modelos de ML.

A partir disso, foi definido o LDD⁹ como o conjunto de dados a ser utilizado nesta pesquisa, sua escolha é justificada devido a sua aplicabilidade em uma ampla variedade de pesquisas semelhantes a esta, além de apresentar resultados expressivos, como os publicados por Abdelhafez e Amer (2024) e Olisah, Smith e Smith (2022), com acurácia acima dos 90% em seus melhores modelos. Ademais, ainda com base na revisão, foram selecionados os algoritmos de ML mais promissores, no contexto de classificação, para os experimentos práticos desta pesquisa, resultando nas futuras implementações dos seguintes modelos: DT, RF, SVM, KNN, GB, NB e LR.

Com base no exposto na Figura 9, a terceira etapa desta pesquisa envolveu a aplicação prática dos modelos definidos, a partir do treinamento e validação, com base em métricas de desempenho como acurácia, precisão, *recall* e F1-Score, visando a identificação do algoritmo com o melhor desempenho. Por fim, o modelo de maior destaque foi disponibilizado em uma API, possibilitando a automatização das predições, além de facilitar a aplicação prática dos resultados em sistemas médicos.

Figura 9 - Planejamento da Pesquisa



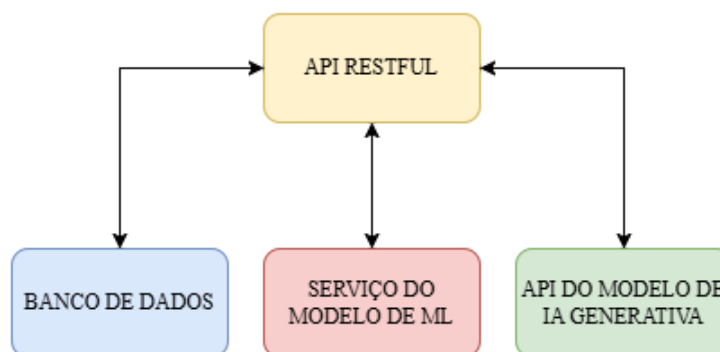
Fonte: Autoria própria

⁹ <https://data.mendeley.com/datasets/wj9rwkp9c2/1>

3.1 Tecnologias utilizadas

A seção a seguir descreve as tecnologias utilizadas no desenvolvimento deste trabalho, incluindo uma API *RESTful*, um serviço responsável pela disponibilização de um modelo de aprendizado de máquina, uma API de modelo de IA Generativa e um banco de dados. De maneira geral, a arquitetura do sistema desenvolvido está representada na Figura 10, a qual ilustra a integração entre os principais componentes da aplicação.

Figura 10 - Tecnologias Utilizadas



Fonte: Autoria própria

O núcleo da aplicação é composto por uma API *RESTful* responsável por orquestrar e intermediar a comunicação entre os componentes, oferecendo uma interface padronizada para o consumo de serviços e promovendo a modularização do sistema. Conectado a essa API, o banco de dados é responsável por armazenar e persistir as informações essenciais ao funcionamento da aplicação. Ademais, a interação com a base de dados ocorre exclusivamente por meio da API *RESTful*, assegurando a segurança dos dados e garante a coesão no controle de acesso.

O serviço responsável pelo modelo de ML constitui um componente dedicado à inferência do algoritmo preditivo de melhor desempenho abordado neste trabalho. Dessa forma, a API *RESTful* encaminha os dados dos pacientes, armazenados no banco de dados, para esse serviço, que, por sua vez, processa as informações e retorna a predição gerada. Por fim, a API do modelo de IA generativa integra ao sistema um *Large Language Model* (LLM), capaz de fornecer uma explicação interpretável acerca da predição realizada pelo modelo de ML. As seções a seguir detalham as escolhas tecnológicas adotadas na composição dessa arquitetura.

3.1.1 *Machine Learning* com Python e IA Generativa com Cohere

Lutz e Ascher (2007) apresentam o Python como uma linguagem de programação de propósito geral, que combina características de orientação a objetos com funcionalidades típicas de *scripts*. VanderPlas (2017) aponta ainda que devido à sua versatilidade, simplicidade sintática e a vasta disponibilidade de bibliotecas, o Python se tornou uma das linguagens mais utilizadas na área de ciência de dados. Nesse contexto, e considerando o desenvolvimento de soluções baseadas em ML, destacam-se três ferramentas baseadas em Python e empregadas neste trabalho: *Scikit-learn*, utilizada para o treinamento dos modelos; *Joblib*, responsável pela persistência dos modelos; e *Flask*, adotada para expor o modelo de melhor desempenho como um serviço via API.

De acordo com McKinney (2023), o *Scikit-learn*¹⁰ configura-se como um conjunto de ferramentas de uso geral para ML, oferecendo submódulos voltados ao desenvolvimento de modelos de classificação, regressão e clusterização. Além disso, disponibiliza recursos para redução de dimensionalidade, seleção e avaliação de modelos, e técnicas de pré-processamento de dados, facilitando a construção de *pipelines* completos de ML (McKinney, 2023).

VanderPlas (2017) destaca que a API do *Scikit-learn* é fundamentada em cinco princípios que definem o seu uso em projetos de ML: consistência na interface entre os objetos, transparência no acesso aos parâmetros, simplicidade na hierarquia de classes, utilização de estruturas de dados padronizadas, composição modular de algoritmos e definição de valores padrão que favorecem a usabilidade¹¹. Desse modo, conforme explica VanderPlas (2017), o processo de treinamento inicia-se com a escolha da classe de modelo adequada ao problema, por meio da importação do estimador correspondente, seguida da instanciação do modelo com os hiperparâmetros desejados. O autor aponta ainda que os dados devem ser organizados em uma matriz de atributos e um vetor de rótulos¹². Após essas etapas, o modelo é, então, ajustado aos dados para que, por fim, possa ser aplicado a novas instâncias.

A biblioteca *Joblib*¹³ destaca-se no ecossistema Python por sua simplicidade e eficiência, especialmente em tarefas de ML. Segundo Zerbo (2024), sua aplicação é particularmente vantajosa em contextos que demandam alto desempenho computacional, devido à facilidade de execução de processos em paralelo com baixa complexidade de

¹⁰ <https://scikit-learn.org/>

¹¹ Na obra original: *consistency, inspection, limited object hierarchy, composition* e *sensible defaults*.

¹² Esses termos também podem ser chamados de *feature* ou *target*, respectivamente.

¹³ <https://joblib.readthedocs.io/>

implementação. Além disso, a *Joblib* é amplamente empregada para a persistência de modelos treinados, permitindo que estes sejam salvos e reutilizados, sem a necessidade de reprocessamento (Pedregosa *et al.*, 2011). Desse modo, essa funcionalidade é essencial para aplicações que requerem a reutilização de modelos em diferentes ambientes ou momentos, promovendo eficiência e escalabilidade no desenvolvimento de soluções baseadas em ML.

Visando a disponibilização do modelo de ML como um serviço disponível via rede, o *Flask*¹⁴ foi adotado e aplicado ao algoritmo de melhor desempenho descrito neste trabalho. Grinberg (2018) apresenta o *Flask* como um *microframework* que simplifica o desenvolvimento de serviços web, permitindo a definição de rotas e o gerenciamento de URLs expostas pela aplicação. No contexto deste estudo, o *Flask* configura-se como uma ferramenta essencial para a construção de uma API capaz de integrar um modelo de ML como um serviço. Assim, essa API possibilita a criação de *endpoints* responsáveis por receber os dados de entrada, realizar a inferência por meio de um modelo previamente salvo com a *Joblib*, e retornar os resultados de forma estruturada e acessível ao usuário final.

Por fim, para explicar as predições realizadas pelo modelo de ML, foi utilizado o LLM *Command R+*, disponibilizado via API pela empresa Cohere. Segundo a documentação oficial¹⁵, esse LLM apresenta um alto desempenho para tarefas de interação conversacional e tarefas de *long-context*¹⁶, o que o torna capaz de responder perguntas, fornecer explicações e apoiar processos de tomada de decisão a partir de entradas estruturadas ou contextuais (Cohere, 2025). Neste trabalho, o *Command R+* foi utilizado para gerar explicações textuais baseadas no resultado da inferência do algoritmo de ML e em informações médicas dos pacientes mantidos no banco de dados da aplicação, o que possibilitou a geração de explicações detalhadas acerca das predições realizadas.

3.1.2 API *RESTful* com Node, TypeScript, *Express*, PostgreSQL e TypeORM

Flanagan (2013) descreve o JavaScript como uma linguagem de programação de alto nível, dinâmica, interpretada e não tipada, sendo adequada tanto para os paradigmas de orientação a objetos quanto ao funcional. O autor destaca, ainda, o JavaScript como um dos pilares do desenvolvimento de aplicações web, especialmente devido à sua presença nativa na maioria dos navegadores modernos, o que o torna amplamente utilizado no lado do cliente.

¹⁴ <https://flask.palletsprojects.com/>

¹⁵ Disponível em: <<https://docs.cohere.com/docs/command-r-plus>>

¹⁶ Tarefas de *long-context* permitem que modelos de IA Generativa consigam processar uma quantidade maior de informações em uma única janela de processamento.

Além disso, por se tratar de uma linguagem de uso geral, foram desenvolvidos ambientes de execução que possibilitam sua aplicação também no lado do servidor. Nesse contexto, o Node surge como uma plataforma que permite a execução do código JavaScript fora do navegador, permitindo sua integração a diferentes contextos, incluindo clientes HTTP e APIs de servidores (Flanagan, 2013). Entretanto, o uso do JavaScript pode acarretar falhas ou erros durante o desenvolvimento de aplicações, especialmente devido à sua baixa restrição estrutural, resultante principalmente da tipagem dinâmica; a documentação pouco rigorosa sobre valores e tipos; e a limitação de suas ferramentas de desenvolvimento (Goldberg, 2022).

Diante das limitações apresentadas, o TypeScript surge como uma alternativa que, mantendo a sintaxe base do JavaScript, introduz recursos próprios para o uso da tipagem estática (Goldberg, 2022). Assim, embora o Node ainda não ofereça suporte nativo à execução de código em TypeScript, a integração entre ambos ocorre por meio de transpiladores, em especial, o *tsc*¹⁷, que permite a conversão do código TypeScript em JavaScript (Cherny, 2019).

Powers (2017) destaca que um dos principais recursos do Node é o uso de pacotes, sejam eles nativos ou disponibilizados por terceiros, que permitem a instalação de conjuntos reutilizáveis de código em um projeto, para adicionar funcionalidades específicas, como o gerenciamento de rotas e a conexão com bancos de dados. Para este trabalho, foram destacados dois pacotes: o *Express*, utilizado para o desenvolvimento de aplicações web, e o *TypeORM*, que abstrai a camada de acesso ao banco de dados.

O *Express*¹⁸ foi empregado para o desenvolvimento de uma API que integra o banco de dados, o serviço responsável pela disponibilização do modelo de ML e as requisições para a API do *Command R+*, consolidando tudo em uma única aplicação. Esse pacote oferece suporte, por exemplo, ao gerenciamento das rotas responsáveis pela autenticação de usuários, criação e consulta de registros no banco de dados, além do envio de respostas às requisições. Dessa forma, conforme exposto por Brown (2019), o *Express* é um *framework* que fornece mecanismos eficientes para definição de *middlewares* e para o controle do fluxo de requisições HTTP, sendo ideal para arquiteturas baseadas em *RESTful APIs*.

Por fim, para o gerenciamento dos dados, optou-se pela utilização do PostgreSQL, um sistema gerenciador de banco de dados relacional de código aberto, amplamente utilizado pela sua robustez, escalabilidade e aderência ao padrão SQL (Obe, Hsu, 2014). Para simplificar a

¹⁷ O *tsc* é o compilador oficial do TypeScript que traduz o código com tipagem estática para JavaScript, padrão compatível com a especificação ECMAScript suportada pelo ambiente de execução.

¹⁸ <https://expressjs.com/pt-br/5x/api.html>

interação entre a aplicação e o banco de dados, foi utilizado o *TypeORM*, um *Object-Relational Mapper* (ORM). Oliveira, *et al.* (2024), conceituam ORM como uma abordagem orientada a objetos que, por meio de uma camada de abstração, simplifica a manipulação de dados relacionais, facilitando operações como consultas e atualizações no banco. Além disso, o *TypeORM* oferece, segundo a sua documentação¹⁹, suporte a migrações²⁰, facilitando o versionamento e a estrutura do banco de dados ao longo do ciclo de desenvolvimento da aplicação.

¹⁹Disponível em: < <https://typeorm.io/> >.

²⁰ Migração, em um ORM, é o processo de transferência dos dados, e de seus esquemas, de uma aplicação para um novo esquema ou uma nova aplicação.

4. RESULTADOS

A seção a seguir apresenta os resultados obtidos durante o desenvolvimento deste estudo, incluindo a etapas de pré-processamento dos dados e de treinamento dos algoritmos, além de uma discussão dos modelos, em termos de métricas de desempenho, sobre os valores resultantes, comparando-os com os apresentados em estudos anteriores, e, por fim, a apresentação da API desenvolvida.

4.1 Pré-Processamento dos dados para o treinamento dos modelos de ML

Para esse trabalho, o *dataset* LDD foi utilizado. Esse conjunto de dados é composto por um total de 1000 registros desbalanceados, extraídos de um hospital no Iraque, de 844 pacientes classificados como diabéticos, 103 não diabéticos e 53 pré-diabéticos. Além disso, as variáveis que compõem essa base de dados incluem informações médicas e demográficas, como: *Id, No. of Patient, Sugar Level Blood, Age, Gender, Creatinine Ratio, Body Mass Index, Urea, Cholesterol, Fasting lipid profile, LDL, VLDL, Triglycerides, HDL Cholesterol, HbA1c* e o alvo *Class*²¹ (Rashid, 2020). A seção a seguir apresenta a etapa de exploração e pré-processamento dos dados presentes no LDD.

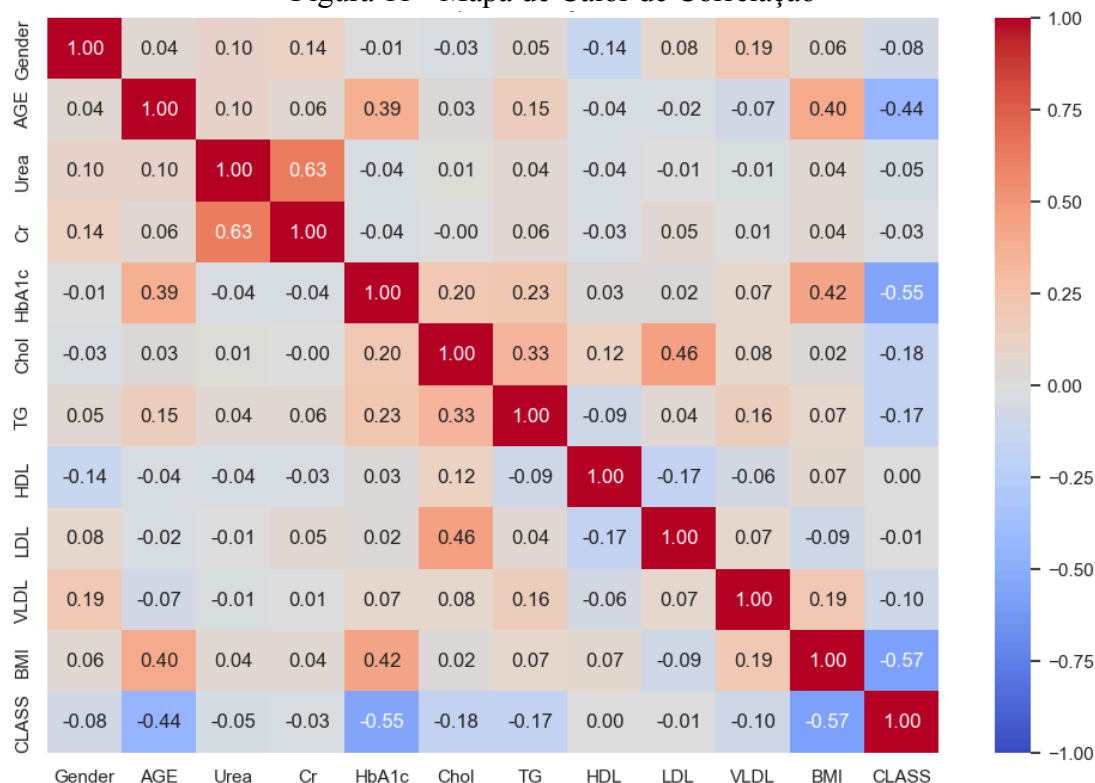
A etapa de pré-processamento dos dados foi fundamental para garantir a qualidade das informações utilizadas para o treinamento dos modelos de ML. Inicialmente, as colunas que não agregavam valor informativo foram removidas do conjunto de dados. Dessa forma, as variáveis *Id* e *No. of Patient* foram excluídas, pois serviam somente para identificação e não possuíam relação direta com o diagnóstico de diabetes. Além disso, foram identificadas variáveis categóricas, *Class* e *Gender*, que passaram por um processo de tratamento: primeiro, foram corrigidas inconsistências de formatação, com a conversão de todos os valores para letras maiúsculas e a remoção de espaços em branco; em seguida, essas variáveis foram convertidas para valores numéricos, visando maior consistência e adequação ao processo de treinamento.

A busca por correlações permite identificar a intensidade e a direção das relações entre os atributos do LDD. Para isso, foi analisada a Figura 11, que apresenta um mapa de calor das relações estabelecidas entre todas as variáveis do conjunto de dados. De modo geral, os coeficientes de correlação variam de -1 a +1, indicando, respectivamente, uma correlação negativa ou positiva. Esse tipo de análise possibilita identificar quais variáveis possuem maior

²¹ Traduzindo para o português, N° do Paciente, Nível de Açúcar no Sangue, Idade, Sexo, Creatinina, Índice de Massa Corporal, Ureia, Colesterol, Perfil Lipídico em Jejum, LDL, VLDL, Triglicerídeos, Colesterol HDL, HbA1c e Classe.

associação com a variável alvo *Class*, contribuindo para a compreensão dos fatores mais influentes na predição do diabetes. Assim, observa-se que as variáveis *Body Mass Index*, *Age* e *HbA1c* apresentam as correlações negativas mais elevadas com o alvo, sugerindo que o aumento de seus valores tende a impactar diretamente na classificação, sendo, portanto, as características mais relevantes para os modelos preditivos.

Figura 11 - Mapa de Calor de Correlação



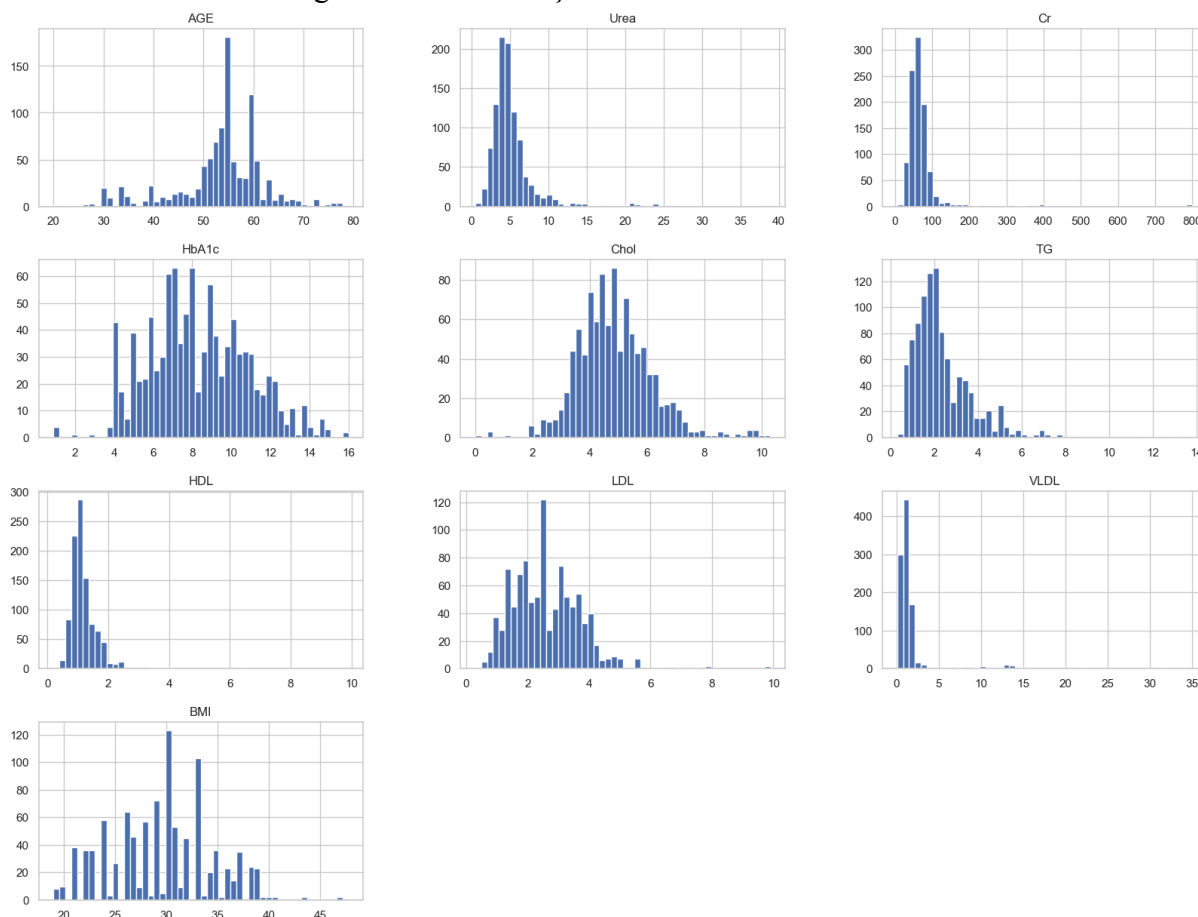
Fonte: Autoria própria

Buscando uma visualização mais ampla a respeito das variáveis presentes no conjunto de dados, foram plotados histogramas para cada atributo numérico presente no LDD, resultando na Figura 12. Assim, tal representação permite a compreensão do formato e dispersão dos dados, resultando na identificação de assimetrias e potenciais *outliers*. Conforme apresentado na Figura 12, percebe-se que algumas variáveis, como *Urea*, *Triglycerides*, *Creatinine Ratio* e *VLDL*, exibem uma distribuição assimétrica com caudas longas, indicando a presença de valores extremos que podem influenciar negativamente o desempenho de modelos de ML.

Desse modo, foram realizados tratamentos para a remoção de *outliers* e normalização das distribuições. Inicialmente, o método do Intervalo Interquartil (IQR) foi empregado para detectar, de forma mais acurada, *outliers* nas colunas *Urea*, *Triglycerides*, *Creatinine Ratio* e

VLDL. Assim, como resultado, foram removidas 148 linhas com valores identificados como *outliers*. Ademais, o método da Transformação de Potência (TP), ou *Yeo-Johnson*, foi utilizado para abordar a assimetria, ou *skewness*, das distribuições. Essa transformação, segundo Yeo e Johnson (2000), consegue tornar as distribuições mais simétricas e próximas de uma distribuição normal, características ideais para o treinamento de modelos de ML. Então, para cada uma das variáveis que indicaram assimetria em sua distribuição, foi aplicada a TP. Doane e Seward (2011), apontam que um valor assimétrico positivo indica que a cauda da distribuição é mais longa no lado direito, enquanto um valor negativo indica uma cauda longa no lado esquerdo. Considerando o contexto do ML, distribuições simétricas, ou seja, próximas de zero, são preferíveis por resultar em um desempenho mais preciso.

Figura 12 - Distribuição das Variáveis Numéricas

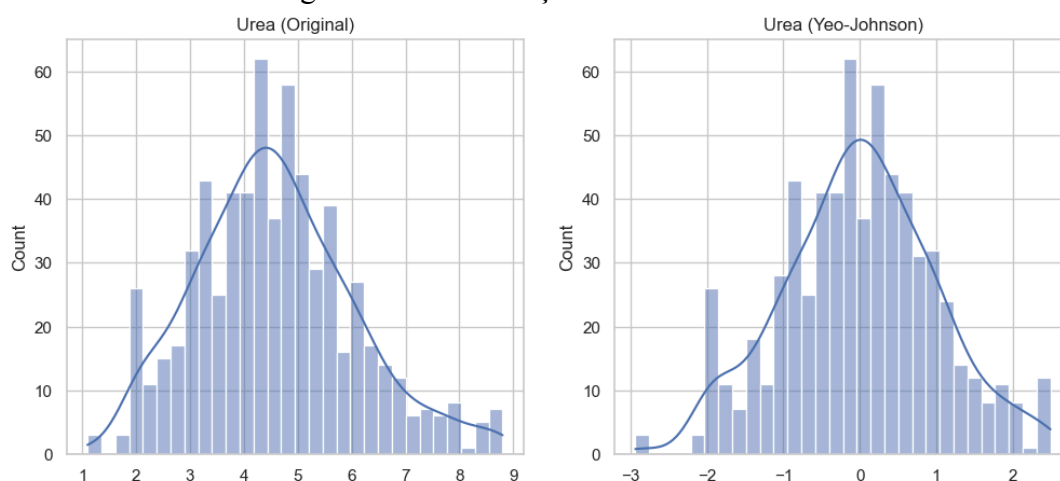


Fonte: Autoria própria

A Figura 13 representa a distribuição da variável *Urea*, que inicialmente apresentava uma assimetria positiva moderada, com valor de 0,42, indicando uma concentração de valores menores e uma cauda estendida para valores mais elevados. Após a aplicação da TP, a

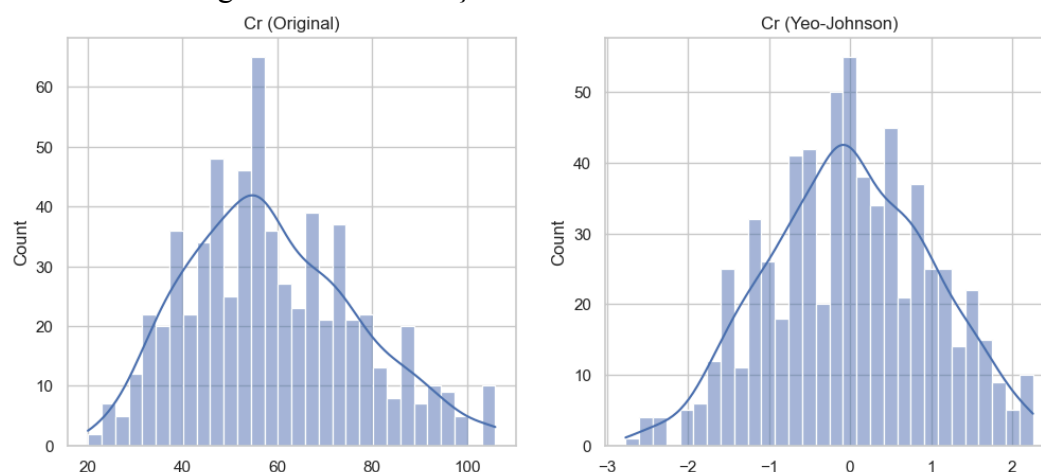
assimetria foi reduzida para 0, tornando a distribuição simétrica. Similarmente, a distribuição da variável *Creatinine Ratio*, representada na Figura 14, possuía inicialmente uma assimetria de 0,41; após a aplicação da TP, esse valor foi reduzido para -0,02, evidenciando uma distribuição ainda mais próxima da simetria.

Figura 13 - Distribuição da Variável *Urea*



Fonte: Autoria própria

Figura 14 - Distribuição da Variável *Creatinine Ratio*

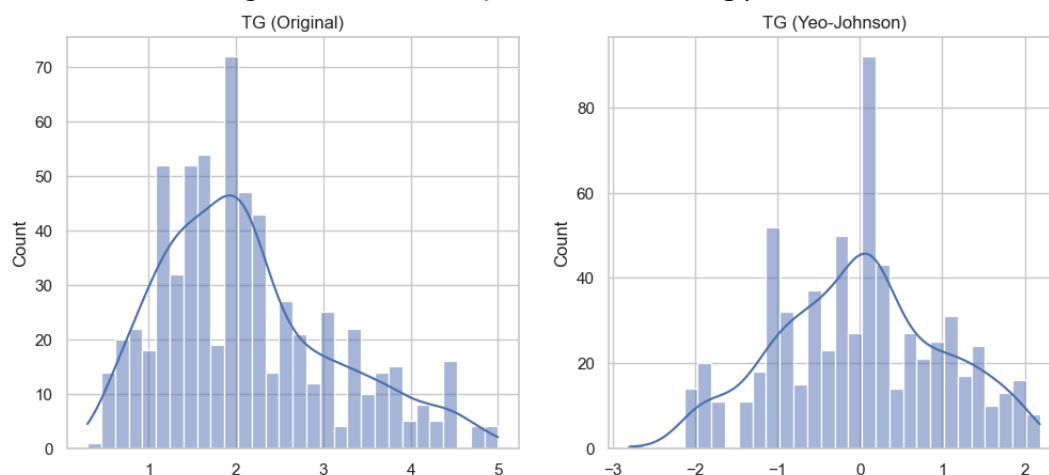


Fonte: Autoria própria

A Figura 15 representa a distribuição da variável *Triglycerides*, que originalmente apresentava uma assimetria positiva mais acentuada, com valor de 0,75. A aplicação da transformação reduziu a assimetria para 0,01, indicando uma distribuição praticamente simétrica. Por fim, a Figura 16 mostra a distribuição da variável *VLDL*, que, assim como a

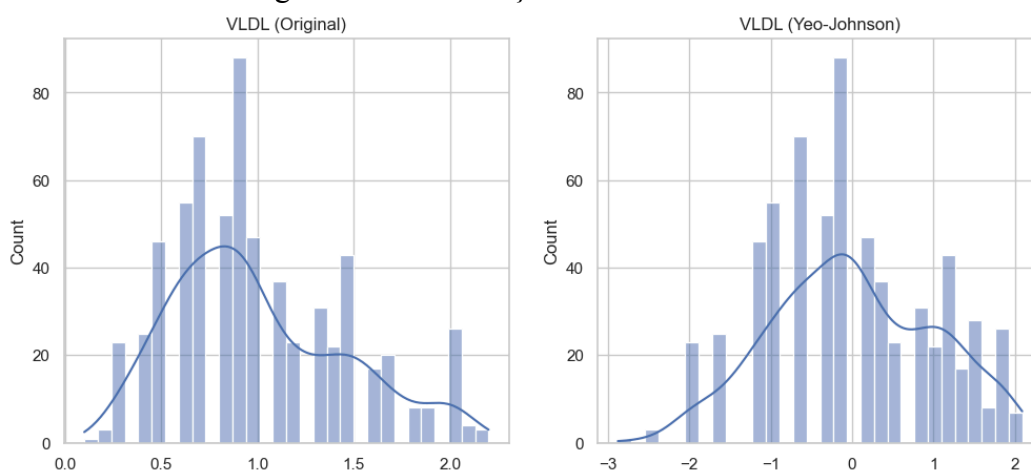
anterior, exibiu uma assimetria positiva significativa, com valor de 0,62, ajustada para 0,02 após a transformação.

Figura 15 - Distribuição da Variável *Triglycerides*



Fonte: Autoria própria

Figura 16 - Distribuição da Variável VLDL



Fonte: Autoria própria

Visando uma avaliação eficiente do modelo, o conjunto de dados foi dividido em treino e teste estratificadamente, com uma proporção de 80% dos dados alocados para a etapa de treinamento e 20% para a validação. De acordo com Géron (2021), a amostragem estratificada consiste em segmentar a população em subgrupos homogêneos, chamados de estratos, e, em seguida, selecionar amostras de cada um deles em proporções adequadas, assegurando que o conjunto de teste reflita a diversidade e as características da população na totalidade. Ou seja, ao utilizar, neste trabalho, uma abordagem estratificada para a divisão dos

conjuntos, busca-se garantir que as classes mantivessem a proporção tanto no treinamento, quanto no teste.

Por fim, visando balancear o conjunto de dados, a técnica *Synthetic Minority Over-sampling Technique* (SMOTE)²² foi utilizada. Segundo Chawla (2002), essa estratégia busca gerar amostras sintéticas para as classes minoritárias presentes na base de dados, equilibrando a distribuição dessas e reduzindo o viés em relação à classe majoritária. Neste trabalho, o SMOTE foi aplicado buscando melhorar a disposição das classes pré-diabético e não diabéticos, que se apresentavam em menor quantidade em relação à classe diabético. Após a aplicação do SMOTE, todas as classes ficaram com a mesma distribuição, com um valor de 675 instâncias para cada grupo.

4.2 Treinando modelos de ML

O treinamento dos algoritmos de ML nesta pesquisa foi conduzido em duas etapas: a primeira considerou todas as características presentes na base de dados; e a segunda considerou somente os atributos mais relevantes, selecionados por meio da técnica *SelectKBest*. De acordo com George (2017), o *SelectKBest*²³ realiza testes estatísticos em cada variável do conjunto de dados, buscando identificar aquelas que mais influenciam a variável-alvo. Além disso, em ambas as etapas foi empregada a técnica *Grid Search*²⁴, que, conforme apontado por Géron (2021), avalia todas as combinações possíveis dos valores de hiperparâmetros fornecidos aos algoritmos, utilizando validação cruzada para encontrar a configuração mais eficaz e, assim, aprimorar o desempenho dos modelos. As seções a seguir detalham o processo de treinamento de cada um dos algoritmos utilizados nesta pesquisa.

4.2.1 Treinamento do *K-Nearest Neighbors* (KNN)

De acordo com Faceli *et al.* (2021), o KNN é um algoritmo de ML que realiza a classificação de objetos a partir da sua proximidade com outros elementos no espaço de atributos. Ou seja, a classe atribuída a uma nova instância é inferida considerando as características dos seus vizinhos mais próximos, assumindo que objetos semelhantes tendem a pertencer à mesma categoria. Nesse contexto, para avaliar o desempenho do modelo em diferentes cenários de entrada e seleção de atributos, o algoritmo KNN foi treinado²⁵,

²² https://imbalanced-learn.org/stable/references/generated/imblearn.over_sampling.SMOTE.html

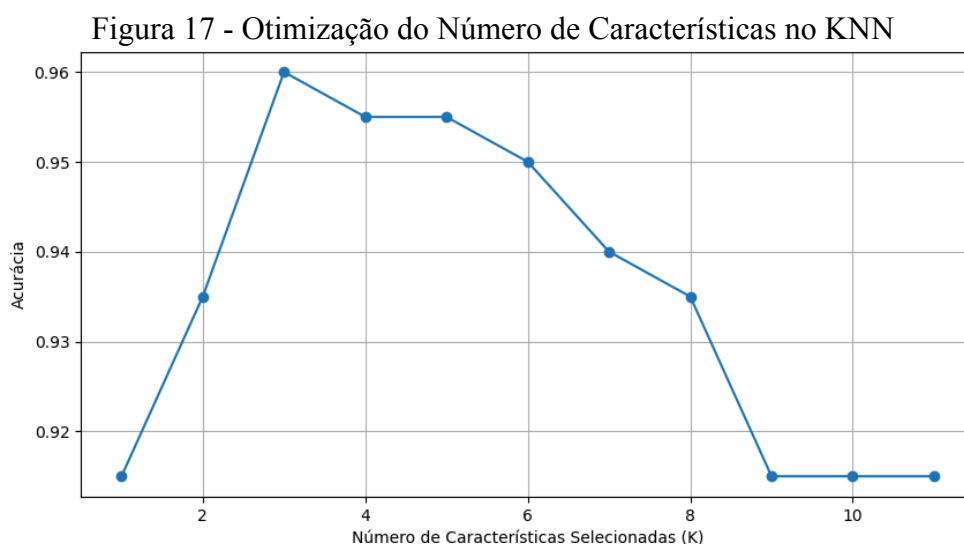
²³ <https://tinyurl.com/SciKitSelectKBest>

²⁴ <https://tinyurl.com/SciKitGridSearchCV>

²⁵ <https://tinyurl.com/knn-training>

utilizando a classe *KNeighborsClassifier*²⁶ da biblioteca *Scikit-learn*, em duas abordagens distintas: a primeira considerou todas as variáveis presentes no conjunto de dados, e a segunda utilizou somente as características mais relevantes, selecionadas com a técnica *SelectKBest*. Na etapa inicial, buscou-se explorar o máximo de informações possíveis fornecidas pelo *dataset*, sem aplicar nenhum tipo de filtragem ou redução de dimensionalidade nos dados. Já na segunda, foram identificadas as variáveis fortemente relacionadas à variável-alvo, contribuindo para a construção de um modelo mais otimizado e, possivelmente, mais eficaz.

A Figura 17 ilustra o processo de otimização do número de atributos a partir da aplicação da técnica *SelectKBest*. No contexto do KNN, observa-se que a acurácia do modelo aumenta conforme são adicionadas mais características, alcançando um valor máximo para um total de três variáveis selecionadas. A partir desse ponto, a inclusão de novas informações não contribui para a melhora do desempenho, resultando, até mesmo, na queda da acurácia. Esse resultado indica que um subconjunto reduzido dos dados, composto por *Age*, *HbA1c* e *BMI*, possui uma alta relevância para a tarefa de predição. Por outro lado, os demais elementos que compõem o conjunto de dados podem ser fracos em termos de poder preditivo, resultando em uma baixa contribuição para o modelo.



Fonte: Autoria própria

Foi empregado, em ambas as etapas de treinamento, o método *Grid Search* para a otimização dos hiperparâmetros do modelo, como o número de vizinhos e o tipo de métrica de

²⁶ <https://tinyurl.com/SciKitKNN>

distância. Em ambos os cenários, considerando o treinamento com todas as variáveis e com as selecionadas, o valor ótimo para o parâmetro que controla a quantidade de vizinhos, o *n_neighbor*, foi igual a 1. Esse resultado indica que o classificador apresentou o melhor desempenho ao considerar somente o vizinho mais próximo para a tomada de decisão. Além disso, o valor obtido para o parâmetro da métrica que mede a distância entre os vizinhos foi a *Manhattan*, sugerindo que o modelo se sobressai ao considerar a soma das distâncias absolutas entre as características²⁷.

A Tabela 2 apresenta os resultados do treinamento do algoritmo KNN nos dois cenários descritos, considerando, inicialmente, todas as variáveis disponíveis no conjunto de dados, e, depois, somente as selecionadas, ou seja, o modelo otimizado. Em termos gerais, a acurácia do modelo treinado com todas as características atingiu um valor de 94%, indicando que o modelo pôde classificar corretamente a maioria das instâncias avaliadas. Já com relação ao cenário otimizado, o KNN obteve uma acurácia de 97%, ou seja, um leve aumento em relação ao contexto anterior, indicando que a remoção de atributos menos informativos pode aumentar o desempenho do classificador.

À vista disso, é possível afirmar que os modelos são eficazes para a classificação correta das instâncias. Entretanto, considerando que o problema apresentado é uma classificação multiclasse, a acurácia não é suficiente para avaliar o desempenho dos algoritmos, tornando-se necessário a análise de métricas que detalham cada uma das classes consideradas pela variável-alvo: predição positiva da diabetes, pré-disposição e predição negativa²⁸. Assim, foram consideradas as métricas *recall*, precisão, *F1-Score*, além da matriz de confusão para auxiliar no entendimento do modelo e de sua capacidade de predição.

Tabela 2 - Resultados do Treinamento do Algoritmo KNN

Atributos usados	Acurácia	Precisão			<i>Recall</i>			<i>F1-Score</i>		
		Classe 0	Classe 1	Classe 2	Classe 0	Classe 1	Classe 2	Classe 0	Classe 1	Classe 2
Total	94%	0.98	0.82	0.70	0.94	0.90	0.90	0.96	0.86	0.79
Otimizado	97%	1.00	0.91	0.83	0.98	1.00	0.95	0.99	0.95	0.89

Fonte: Autoria própria

²⁷ Conferir o Apêndice A com as explicações dos hiperparâmetros considerados para cada um dos modelos treinados neste estudo.

²⁸ Considere durante o trabalho: Classe 0 para a predição positiva, Classe 1 para pré-disposição e Classe 2 para predição negativa da diabetes.

Em relação ao *recall* para o modelo treinado com todas as variáveis, foram obtidos os seguintes valores: 94% para a Classe 0 e 90% para as Classes 1 e 2. Esses resultados indicam, então, que o KNN teve um bom desempenho em todas as classes, especialmente no cenário de predição positiva da diabetes. Entretanto, ao se considerar a precisão, percebe-se uma maior variação nos valores de cada uma das classes, nas quais os resultados foram mais elevados para a Classe 0, com 98%, e menores para as Classes 1 (82%) e 2 (70%). Dessa forma, tais resultados evidenciam que, embora o modelo preveja os casos positivos da diabetes eficientemente, sua precisão reduz para os cenários de casos negativos e de predisposição à doença. Em especial, o resultado para a Classe 2 indica uma taxa elevada de falsos negativos, o que pode ser interpretado como uma tendência do modelo em superestimar essa classe.

No cenário no qual o KNN foi treinado com o conjunto reduzido de características, observou-se uma melhora nos valores da precisão, com a Classe 0 atingindo 100%, enquanto as Classes 1 e 2 avançaram para 91% e 73%, respectivamente. Tais resultados demonstram que, ao focar em variáveis mais informativas, o modelo conseguiu se tornar mais eficiente na tarefa de classificação correta dos dados para cada uma das classes, reduzindo a taxa de falsos positivos. Da mesma forma, o *recall* também apresentou um aumento no seu desempenho, considerando o cenário otimizado. Nesse contexto, a Classe 0 apresentou 98%, e a Classe 1 atingiu 100%, indicando que todos os indivíduos dessa categoria foram corretamente identificados. Além disso, a Classe 2 também apresentou melhoria, com *recall* de 95%. Assim, os resultados refletem que o modelo treinado com a redução do número de atributos apresenta maior sensibilidade, conseguindo identificar com mais precisão os casos verdadeiros de cada classe.

O *F1-Score*, caracterizado por representar a média harmônica entre precisão e *recall*, apresentou uma alta variação para os resultados de cada classe, no contexto do modelo treinado com todas as variáveis, com 96% para a Classe 0, 86% para a Classe 1 e 79% para a Classe 2. Esse comportamento reforça a tendência observada nas métricas anteriores: um desempenho geral robusto, porém com limitações na predição das classes relacionadas a pré-condição e, em especial, ao diagnóstico negativo da doença. Para o cenário otimizado, o *F1-Score* apresentou melhorias em todas as classes, com 99% para a Classe 0, 95% para a Classe 1 e 89% para a Classe 2. Tais avanços mostram que o modelo passou a ser mais equilibrado, reduzindo as diferenças entre as classes e garantindo um desempenho mais uniforme.

A matriz de confusão, apresentada na Figura 18, ilustra o desempenho do KNN treinado com todas as variáveis disponíveis no conjunto de dados. Para essa situação, percebe-se que a Classe 0 foi bem classificada para a maioria dos casos, apresentando somente 10 erros de classificação, sendo dois confundidos como pré-diabéticos e oito como não diabéticos. Além disso, a Classe 1 obteve um desempenho satisfatório, com somente um erro de predição. Da mesma forma, a Classe 2 também apresentou um bom desempenho, com duas instâncias classificadas incorretamente. Esses resultados evidenciam, então, que, embora o modelo tenha desempenho robusto, ainda apresenta confusões nas fronteiras entre os grupos.

Figura 18 - Matriz de Confusão Para o Modelo KNN Total

Real	0	159	2	8
	1	1	9	0
	2	2	0	19
		0	1	2
		Predito		

Fonte: Autoria própria

Por fim, a matriz de confusão para o contexto de treinamento do KNN com a otimização do número de variáveis é ilustrada na Figura 19. Nesse cenário, percebe-se que o desempenho do modelo foi bastante equilibrado, com uma boa capacidade de discriminar entre as três classes. Com relação à Classe 0, o modelo identificou corretamente 165 casos, cometendo somente 4 falsos positivos preditos como pertencentes a Classe 2. Dessa forma, tal comportamento indica uma leve tendência à subestimação do diagnóstico positivo, o que pode resultar no risco de não identificação da doença. Em contrapartida, o desempenho da classe de pré-disposição foi ótimo, acertando todas as instâncias previstas para essa classe. Ademais, a Classe 2 classificou somente uma instância errada, enquanto acertou 20. No geral, é possível afirmar que o modelo foi eficaz para a tarefa de classificação da diabetes.

Figura 19 - Matriz de Confusão Para o Modelo KNN Otimizado

Real \ Predito	0	1	2
0	165	0	4
1	0	10	0
2	0	1	20

Fonte: Autoria própria

4.2.2 Treinamento da *Support Vector Machine* (SVM)

Conforme exposto por Smola *et al.* (2000, *apud* Faceli *et al.*, 2021), os *Large Margin Classifiers* (LMC) constituem um conjunto de técnicas que buscam, de forma implícita ou explícita, maximizar a distância de separação entre amostras pertencentes a classes distintas, promovendo, assim, uma melhor generalização dos modelos. Com base nesse conceito, a SVM destaca-se como um algoritmo maximizador de margens, sendo aplicável a tarefas de classificação linear e não linear, regressão, bem como à detecção de *outliers* (Géron, 2021).

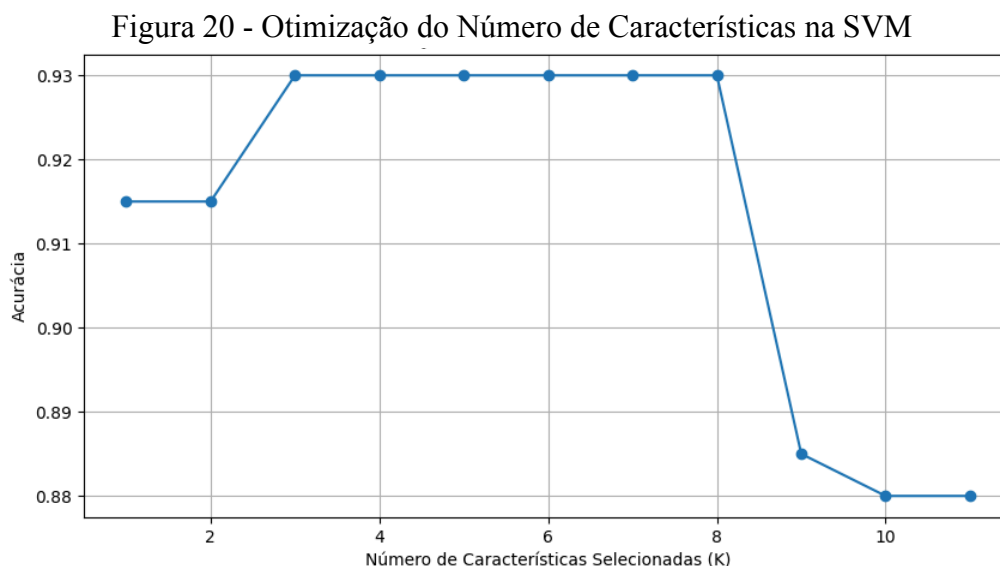
O modelo SVM foi treinado²⁹ utilizando a classe *SVC*³⁰ da biblioteca *Scikit-learn*, buscando realizar uma tarefa de classificação multiclasse, e integrando abordagens de seleção de atributos e ajuste de hiperparâmetros. Assim como no KNN, a metodologia de treinamento foi dividida em duas etapas. Inicialmente, o modelo foi ajustado utilizando todas as características presentes no conjunto de dados, sem qualquer filtragem ou redução dimensional. Já na segunda etapa, a abordagem de seleção das variáveis mais relevantes, a partir da técnica *SelectKBest*, foi aplicada, visando melhorar a eficiência e a capacidade preditiva do classificador.

O processo de seleção de características com o *SelectKBest* é ilustrado na Figura 20, que mostra como a acurácia do classificador varia conforme diferentes quantidades de atributos são considerados. Dessa forma, observa-se que, ao selecionar entre 3 e 8 variáveis, o modelo atinge seu melhor desempenho, atingindo uma acurácia acima dos 90%. Em contrapartida, a inclusão de mais atributos resulta em uma queda acentuada na acurácia, indicando que atributos adicionais podem trazer ruído ou redundância, o que compromete a

²⁹ <https://tinyurl.com/svm-training>

³⁰ <https://tinyurl.com/SciKitSVC>

capacidade preditiva do modelo. Além disso, assim como observado no treinamento do KNN, as três variáveis selecionadas foram *Age*, *HbA1c* e *BMI*, evidenciando a forte relevância preditiva desses atributos no contexto da classificação da diabetes.



Fonte: Autoria própria

O *Grid Search*, em conjunto da validação cruzada, foi utilizado para a identificação da combinação dos melhores hiperparâmetros do modelo. No contexto da SVM, foram considerados o parâmetro de regularização (C), o tipo de *kernel* e, quando aplicável, o parâmetro *gamma*. Como resultado, em ambos os cenários, com todas as variáveis e com os atributos selecionados, identificou-se o valor ideal de C como 10, sugerindo que o modelo deve ser ajustado para penalizar com mais rigor os erros de classificação, priorizando uma separação mais precisa das classes, mesmo que a margem entre eles seja reduzida. Além disso, o *kernel* selecionado foi o linear, indicando que os dados são linearmente separáveis em seu espaço original, dispensando o mapeamento para um espaço de maior dimensionalidade. Por fim, o parâmetro *gamma* foi definido como *scale*, que ajusta automaticamente o valor com base na variância dos dados.

Com relação aos resultados obtidos com seu treinamento, o modelo SVM apresentou um desempenho considerável para ambos os cenários. A Tabela 3 apresenta esses resultados e, ao observar o contexto geral dos modelos treinados, a acurácia apresentou um resultado de 96% para o modelo que considerou todas as variáveis disponíveis no conjunto de dados, e 97% para o cenário de seleção de características. Desse modo, esses resultados indicam que o

SVM pôde classificar, em ambos os contextos, corretamente a maioria das instâncias, refletindo uma alta capacidade de generalização.

Tabela 3 - Resultados do Treinamento do Algoritmo SVM

Atributos usados	Acurácia	Precisão			<i>Recall</i>			<i>F1-Score</i>		
		Classe 0	Classe 1	Classe 2	Classe 0	Classe 1	Classe 2	Classe 0	Classe 1	Classe 2
Total	96%	1.00	0.77	0.84	0.96	1.00	1.00	0.98	0.87	0.91
Otimizado	97%	1.00	0.91	0.81	0.96	1.00	1.00	0.98	0.95	0.89

Fonte: Autoria própria

No que diz respeito à precisão, observou-se que o modelo treinado com todas as variáveis apresentou um desempenho máximo para a Classe 0 (100%), indicando que todas as instâncias classificadas como diabéticos estavam corretas. Entretanto, ao observar os casos de pré-diabéticos e negativos, os resultados sofrem uma queda, com 77% e 84% para cada uma das classes, respectivamente. Tais valores sugerem que o modelo tende a gerar falsos positivos para esse casos, em especial no contexto de predisposição à doença. A partir da aplicação da seleção de características, a precisão da Classe 0 permaneceu com 100%, enquanto a Classe 1 aumentou para 91%, indicando que o modelo passou a errar menos ao classificar instâncias de pré-diabéticos. Por outro lado, a Classe 2 apresentou uma leve queda, apresentando uma precisão de 81%, o que pode indicar que, ao favorecer uma melhor distinção da Classe 1, houve uma pequena queda na separabilidade da Classe 2.

Com relação ao *recall*, os resultados obtidos para o cenário total demonstram uma eficiência em todas as classes, com 96% para os casos positivos e 100% para os casos de pré-diabete e negativos. Esses valores apontam, então, que o modelo não deixou de detectar nenhum paciente para as Classes 1 e 2 e classificou a maioria dos pacientes da Classe 0. Ademais, considerando o contexto de cenários clínicos, esse comportamento é essencial para a garantia de que nenhum caso deixe de ser identificado. Após a otimização, o *recall* continuou estável para todas as classes, com 100% para as Classes 1 e 2, e 96% para a Classe 0. Isso demonstra que o modelo otimizou sua capacidade preditiva sem comprometer a sua sensibilidade.

O *F1-Score* obtido para o modelo total manteve-se elevado para todas as classes, com 98% para a Classe 0, 87% para a Classe 1 e 91% para a classe 2. Tais resultados refletem a eficiência do modelo em sua capacidade de generalização. Entretanto, é importante ressaltar

que a ligeira redução do *F1-Score* para a classe de pré-diabete é resultante da menor precisão obtida, refletindo uma dificuldade do modelo de distinguir corretamente os casos de predisposição. Ainda assim, os resultados consistentes indicam que o modelo SVM apresenta equilíbrio entre sensibilidade e especificidade mesmo com um conjunto reduzido de atributos. Após o cenário de otimização, o *F1-Score* aumentou para a Classes 1 (95%), refletindo uma melhora na relação entre a precisão e o *recall* dessa categoria. As demais classes mantiveram-se estáveis, com 98% para a Classe 0 e 89% para a Classe 2, indicando que a otimização equilibrou as métricas de todas as classes.

Figura 21 - Matriz de Confusão Para o Modelo SVM Total

Real	0	162	3	4
	1	0	10	0
	2	1	0	20
		0	1	2
		Predito		

Fonte: Autoria própria

A matriz de confusão ilustrada na Figura 21 apresenta o desempenho do modelo SVM com todas as características. Nesse contexto, o modelo conseguiu identificar corretamente 162 casos positivos da diabetes, com somente 3 instâncias confundidas como pré-diabete e 4 com o diagnóstico negativo, indicando, apesar da leve taxa de erro, uma alta sensibilidade para a Classe 0. Ademais, a Classe 1 não sofreu nenhum erro de classificação, com todas as instâncias sendo corretamente identificadas como pré-disposição a diabetes, refletindo o resultado obtido pelo *recall*. Já a Classe 2 obteve somente um erro, no qual o indivíduo foi identificado como diabético, representando um falso positivo. De modo geral, a matriz revela que o modelo SVM completo conseguiu capturar adequadamente os padrões das três classes, com baixo viés extremo para uma única classe. Por fim, a Figura 22 representa a matriz de confusão para modelo SVM otimizado com a seleção de características mais relevantes. Nesse cenário, observa-se um leve aumento no desempenho com relação ao treinamento com todas as características, em particular para a classificação positiva da diabetes, com 163 acertos e 6

erros. Ademais, com relação às Classes 1 e 2, percebe-se que ambas apresentaram desempenho satisfatório, com todos os casos classificados corretamente.

Figura 22 - Matriz de Confusão Para o Modelo SVM Otimizado

	0	1	2
Real 0	163	1	5
1	0	10	0
2	0	0	21
	0	1	2
	Predito		

Fonte: Autoria própria

4.2.3 Treinamento da *Logistic Regression* (LR)

Segundo Das (2021), a LR é um método estatístico que consegue gerar um modelo de ML capaz de representar a relação entre uma variável binária e um conjunto de valores preditivos. Ou seja, como apontado por Géron (2021), esse algoritmo busca estimar a probabilidade de uma instância ser pertencente a uma classe específica, para isso, o modelo “calcula a soma ponderada de das características de entrada [...], gerando a saída logística desse resultado”. Ademais, a LR consegue, a partir da aplicação de uma função exponencial normalizada, oferece suporte a problemas de classificação multiclasse sem a necessidade de combinar diferentes classificadores binários (Géron, 2021).

Assim como nos treinamentos anteriores, a LR foi abordada em dois cenários diferentes, o primeiro considerando todas as variáveis disponíveis no conjunto de dados, e o segundo considerando somente as selecionadas pelo *SelectKBest*. Desse modo, para o treinamento desse algoritmo³¹, foi utilizada a classe *LogisticRegression*³² da biblioteca *Scikit-learn*, que disponibiliza uma implementação do algoritmo LR, visando realizar uma tarefa de classificação multiclasse.

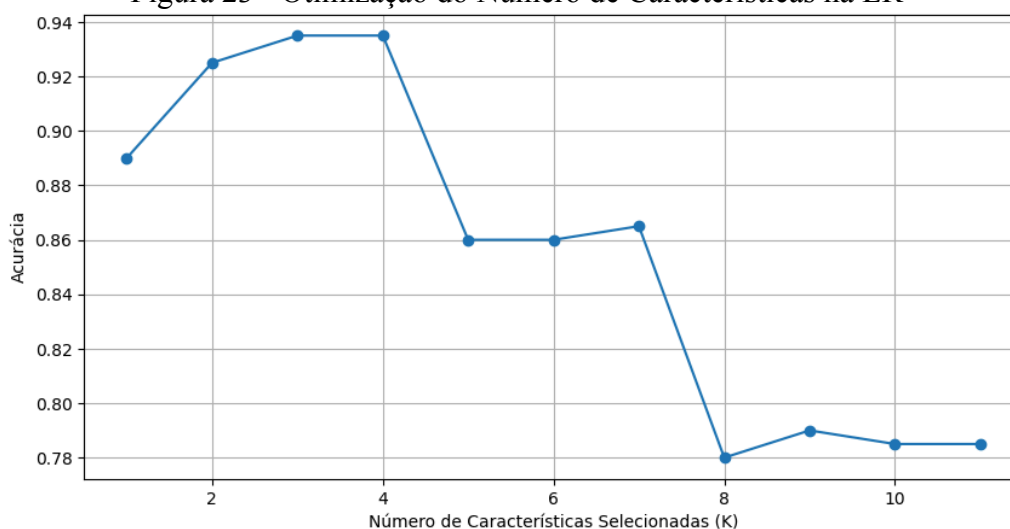
A Figura 23 apresenta o processo de otimização do modelo LR com base na quantidade de variáveis utilizadas em seu treinamento. Com base no gráfico, percebe-se que o algoritmo atinge o seu desempenho máximo ao considerar um total de três características e, a

³¹ <https://tinyurl.com/lr-training>

³² <https://tinyurl.com/SciKitLogisticRegression>

partir desse ponto, a inclusão de variáveis adicionais resulta em uma queda no desempenho, possivelmente devido à introdução de ruído ou redundância. Ademais, assim como nos modelos KNN e SVM, as características selecionadas foram *Age*, *HbA1c* e *BMI*, reforçando a relevância preditiva dessas variáveis no contexto da detecção de diabetes.

Figura 23 - Otimização do Número de Características na LR



Fonte: Autoria própria

Em ambas as abordagens, com todas as variáveis e com os atributos selecionados, a técnica *Grid Search* foi abordada para a otimização dos parâmetros do modelo. No cenário da LR, o processo explorou diferentes combinações, incluindo o *C*, o tipo de penalização, o solucionador numérico e o número máximo de iterações. Com relação aos resultados, foi observado que o *C* obteve o valor de 10 e a penalidade selecionada foi a *L1*, que tende a gerar modelos mais esparsos, ou seja, o modelo obteve um melhor desempenho ao aplicar uma penalização mais intensa. Ademais, a seleção do *solver liblinear*, mostra-se adequada para problemas de pequena e média complexidade e compatível com a penalização *L1*, reforçando a robustez do modelo. Por fim, o valor máximo de iterações foi definido como 200, indicando que o modelo precisou de um número maior de atualizações nos parâmetros para alcançar uma solução ideal.

A Tabela 4 apresenta os resultados obtidos com o treinamento do modelo LR com todas as características e com as selecionadas. Em termos gerais, a acurácia do modelo treinado no primeiro cenário obteve um valor de 95%, enquanto a do modelo otimizado com a seleção de atributos resultou em 94%, apresentando uma queda de um ponto percentual. Tais resultados evidenciam, dessa forma, a consistência da LR na tarefa de classificação

multiclasse. Entretanto, é importante analisar métricas de desempenho que especifiquem os resultados de cada uma das classes que compõem o problema.

Tabela 4 - Resultados do Treinamento do Algoritmo LR

Atributos usados	Acurácia	Precisão			<i>Recall</i>			<i>F1-Score</i>		
		Classe 0	Classe 1	Classe 2	Classe 0	Classe 1	Classe 2	Classe 0	Classe 1	Classe 2
Total	95%	0.99	0.64	0.87	0.96	0.90	0.95	0.98	0.75	0.91
Otimizado	94%	1.00	0.62	0.77	0.93	1.00	0.95	0.97	0.77	0.85

Fonte: Autoria própria

Com relação à precisão do modelo treinado com todas as variáveis, destaca-se o desempenho para a predição positiva da diabete, com valor de 99%, indicando que o modelo quase não cometeu falsos positivos para essa classe. Além disso, a Classe 2 também apresentou uma boa precisão, com 87%. Entretanto, o modelo refletiu incerteza na predição dos casos de pré-diabete, já que o valor da precisão para a Classe 1 foi de 64%. Com a aplicação da seleção de características, observou-se um leve aumento na precisão da Classe 0, que atingiu o valor de 100%, indicando que todas as instâncias classificadas como diabéticos estavam corretas. Entretanto, a precisão da Classe 1 caiu para 62%, mantendo-se como o ponto fraco do modelo. Da mesma forma, a Classe 2 apresentou queda, indo para 77%. Esses resultados sugerem que a otimização não contribuiu para um avanço expressivo na precisão das classes de predição da diabete.

Por outro lado, na métrica *recall*, para o contexto de treinamento com todas as características, observa-se um cenário bastante balanceado, com 96% para a Classe 0, 90% para a Classe 1 e 95% para a Classe 2. Tal resultado demonstra que o modelo conseguiu identificar corretamente a maioria dos indivíduos nessas classes. O *F1-Score*, que representa o equilíbrio entre as métricas anteriores, confirmou a robustez do modelo em relação às Classes 0 e 2. Entretanto, para a Classe 1 foi observado o valor de 75%, revelando um desempenho moderado, o qual foi impactado pela menor precisão. Considerando o cenário de otimização, houve uma melhoria para o *recall* da Classe 1, apresentando valor de 100%. As demais classes mantiveram-se estáveis, o que demonstra que o modelo ajustado conseguiu melhorar na predição correta dos casos de pré-diabete. Além disso, o *F1-Score* da Classe 1 melhorou ligeiramente, atingindo 77%, demonstrando um pequeno ganho de equilíbrio para essa

categoria. Já as demais classes sofreram uma leve queda em seus desempenhos, com 97% para a Classe 0 e 85% para a Classe 1.

A Figura 24 ilustra a matriz de confusão do modelo LR que considerou todas as variáveis em seu treinamento. Com relação à classe de predição positiva da diabetes, observou-se que 162 casos foram classificados corretamente, porém observou-se sete erros de predição, representando um número de falsos negativos que pode impactar o contexto clínico. Além disso, a Classe 1 obteve um bom desempenho com nove acertos e somente um erro. Da mesma forma, a Classe 2 apresentou 20 classificações corretas, com um erro. Dessa forma, esse resultado indica que, embora eficaz para a maioria dos casos, o modelo apresenta desafios na diferenciação precisa entre as Classes 0 e 1, o que pode ser reflexo da similaridade clínica entre essas.

Figura 24 - Matriz de Confusão Para o Modelo LR Total

Real	0	162	4	3
	1	1	9	0
	2	0	1	20
		0	1	2
		Predito		

Fonte: Autoria própria

Por fim, a matriz de confusão apresentada na Figura 25 evidencia o desempenho da LR otimizada a partir da seleção de características. Nesse contexto, observa-se uma leve queda em relação ao cenário anterior, com um total de 158 instâncias classificadas corretamente para a Classe 0, com 11 erros distribuídos entre as Classes 1 e 2. Além disso, para a Classe 1, todas as instâncias foram classificadas corretamente, indicando um *recall* máximo para essa classe. Já a Classe 2 apresentou 20 acertos, com somente um erro, apresentando um bom desempenho geral.

Figura 25 - Matriz de Confusão Para o Modelo LR Otimizado

Real	0	158	5	6
	1	0	10	0
	2	0	1	20
		0	1	2
		Predito		

Fonte: Autoria própria

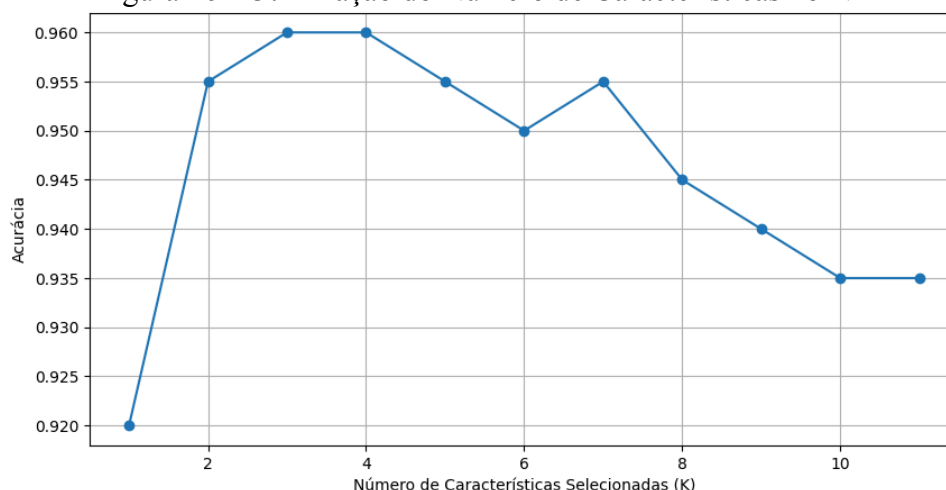
4.2.4 Treinamento do *Naive Bayes* (NB)

De acordo com Webb (2017), o NB é um algoritmo de ML baseado no teorema de Bayes, assumindo que os atributos são independentes em termos de relação à classe. Dessa forma, o NB oferece um método que, a partir dos dados amostrais, estima a probabilidade a posteriori de uma classe específica para um determinado objeto (Webb, 2017). Com base nesse princípio, e assim como nos algoritmos anteriores, o NB foi treinado³³ considerando dois cenários distintos quanto à quantidade de variáveis utilizadas: o primeiro com todas as características disponíveis no conjunto de dados e o segundo com um subconjunto selecionado por meio da técnica de seleção de atributos *SelectKBest*. Para isso, foi utilizada a função *GaussianNB*³⁴ da biblioteca *Scikit-learn*, que disponibiliza uma implementação desse algoritmo baseada na suposição de que as variáveis preditoras seguem uma distribuição normal, característica comum em casos de classificação multiclasse. Os resultados obtidos com o processo de seleção de características utilizando o método *SelectKBest* são descritos na Figura 33.

³³ <https://tinyurl.com/training-nb>

³⁴ <https://tinyurl.com/SciKit-GaussianNB>

Figura 26 - Otimização do Número de Características no NB



Fonte: Autoria própria

Com base no observado, percebe-se que a acurácia do modelo aumenta ao considerar as três primeiras variáveis mais relevantes, sendo essas *Age*, *HbA1c* e *BMI*, atingindo o valor máximo com a seleção de três ou quatro variáveis. A partir desse ponto, o desempenho do NB começa a decair, indicando que a inclusão de variáveis adicionais pode introduzir ruído ou redundância, prejudicando a capacidade de generalização do classificador. Dessa forma, esse resultado reforça a importância da etapa de seleção de atributos, já que um conjunto de valores enxuto e informativo pode potencializar a capacidade de predição sem comprometer a simplicidade computacional do modelo.

Para otimizar o NB com relação a seu hiperparâmetro *var_smoothing*, responsável por controlar a estabilidade numérica durante o cálculo das probabilidades, a técnica *Grid Search* foi conduzida. Dessa forma, após a execução do processo em ambos os cenários de treinamento, o valor otimizado encontrado foi o mesmo: $1e^{-09}$. Esse resultado indica, então, que o modelo se beneficia de um pequeno grau de suavização nas variâncias, contribuindo para mitigar problemas causados por valores pequenos na estimação das distribuições.

Os resultados do treinamento do NB em cada um dos cenários propostos, inicialmente com todas as variáveis disponíveis e, depois, com o modelo otimizado com a seleção de características, estão descritos na Tabela 8. De modo geral, observa-se que o desempenho do NB melhorou com a otimização, em especial em métricas voltadas para a quantificação do desempenho das classes, indicando que a seleção de atributos contribuiu para o ajuste do classificador. Dessa forma, em relação à acurácia, observa-se um aumento de 94% para 96% após a otimização. Esse ganho evidencia que a filtragem de atributos impactou positivamente

o desempenho geral do modelo, permitindo que ele acertasse mais classificações no conjunto de testes.

Tabela 5 - Resultados do Treinamento do Algoritmo NB

Modelo NB	Acurácia	Precisão			<i>Recall</i>			<i>F1-Score</i>		
		Classe 0	Classe 1	Classe 2	Classe 0	Classe 1	Classe 2	Classe 0	Classe 1	Classe 2
Total	94%	0.99	0.64	0.75	0.93	0.90	1.00	0.96	0.75	0.86
Otimizado	96%	0.99	0.91	0.77	0.96	1.00	0.95	0.98	0.95	0.85

Fonte: Autoria própria

Considerando o cenário de treinamento com todas as variáveis disponíveis, a precisão do NB apresentou um desempenho estável para a Classe 0, atingindo um total de 99%. Entretanto, as demais classes resultaram menores, com 64% para a Classe 1 e 75% para a Classe 2. Esses valores indicam, então, que o modelo demonstrou uma maior dificuldade em prever corretamente as instâncias dessas duas classes sem incorrer em falsos positivos. Já no contexto de seleção de atributos, houve uma melhora na precisão das Classes 1 e 2, com valores de 91% e 77%, respectivamente, além de manter os 99% para a Classe 0. Com base nesses resultados, observa-se que a redução do conjunto de atributos permitiu uma tomada de decisão mais assertiva por parte do modelo, reduzindo a incidência de falsos positivos.

Com relação ao *recall* para o modelo total, o NB atingiu 93% para a classe de predição positiva da doença, 90% para a de predisposição e 100% para a de predição negativa. Esses resultados indicam, dessa forma, que o algoritmo foi eficaz em identificar corretamente todas as instâncias para a Classe 2, porém obteve um desempenho um pouco inferior para as demais classes, em especial na Classe 1, em que parte das instâncias reais não foram classificadas corretamente. Com relação ao modelo otimizado, os valores do recall aumentaram para as Classes 0 e 1, agora com 96% e 100%, respectivamente. Entretanto, a Classe 2 apresentou uma leve queda, de 100% para 95%. Ainda assim, o NB treinado no contexto de seleção de atributos demonstrou maior equilíbrio na capacidade de recuperação das classes, resultando em um modelo com uma maior capacidade de generalização.

Ao considerar o *F1-Score* para o modelo total, observam-se os seguintes valores: 96% para Classe 0, 75% para a Classe 1 e 86% para a Classe 2. Nessa situação, o valor elevado para a classe de predição positiva da diabetes demonstra uma harmonia entre a precisão e o *recall*, porém os valores mais baixos para as demais classes apontam um certo desequilíbrio

para esses grupos, em especial para a Classe 1. Em contrapartida, o modelo otimizado apresentou uma melhoria significativa para a Classe 1, cujo *F1-Score* passou de 75% para 95%. Ademais, as Classes 0 e 2 se mantiveram estáveis, com 98% e 85%, respectivamente. Isso indica que, apesar de uma leve redução para a classe de predição negativa, a redução do número de variáveis contribuiu para um desempenho mais otimizado para todas as classes.

A Figura 34 ilustra a matriz de confusão do NB treinado com todas as características do conjunto de dados. Nesse contexto, observa-se que a Classe 0, apesar de demonstrar um bom desempenho geral, foi a mais impactada com erros de classificação, com 12 amostras incorretamente atribuídas a outras classes. Além disso, a Classe 1 também apresentou um pequeno número de erros, com uma instância sendo classificada incorretamente como diabético. Por outro lado, a Classe 2 teve todas as instâncias classificadas corretamente. Dessa forma, tais resultados refletem o padrão identificado nas métricas descritas anteriormente, indicando que o modelo possui maior dificuldade em distinguir entre as classes de predição positiva e de predisposição a diabetes.

Figura 27 - Matriz de Confusão Para o Modelo NB Total

Real	0	157	5	7
	1	1	9	0
	2	0	0	21
		0	1	2
		Predito		

Fonte: Autoria própria

Por fim, a Figura 35 apresenta a matriz de confusão do modelo NB otimizado em relação à seleção de atributos. Em comparação ao cenário anterior, nota-se uma melhora na capacidade de classificação dos indivíduos com predição positiva da diabetes, o qual apresentou somente 7 amostras classificadas incorretamente. Ademais, a Classe 1 sofreu um leve aumento em seu desempenho, agora com todas as amostras sendo corretamente

classificadas. Por outro lado, a Classe 2 apresentou uma leve queda, com um erro de predição. De forma geral, o modelo otimizado conseguiu reduzir os erros de classificação, especialmente para a Classe 0, evidenciando um ganho de desempenho.

Figura 28 - Matriz de Confusão Para o Modelo NB Otimizado

Real	0	162	1	6
	1	0	10	0
	2	1	0	20
		0	1	2
		Predito		

Fonte: Autoria própria

4.2.5 Treinamento do *Gradient Boosting* (GB)

Segundo Natekin e Knoll (2013), o GB é um algoritmo *ensemble* de ML que, a partir da iteração de modelos simples, consegue construir um modelo preditivo forte orientado pelo gradiente da função de perda, reduzindo progressivamente o erro. Dessa forma, a cada repetição, um novo modelo é treinado para ser o mais correlacionado possível com os erros resultantes, fazendo com que o algoritmo corrija as falhas dos modelos anteriores (Natekin e Knoll, 2013).

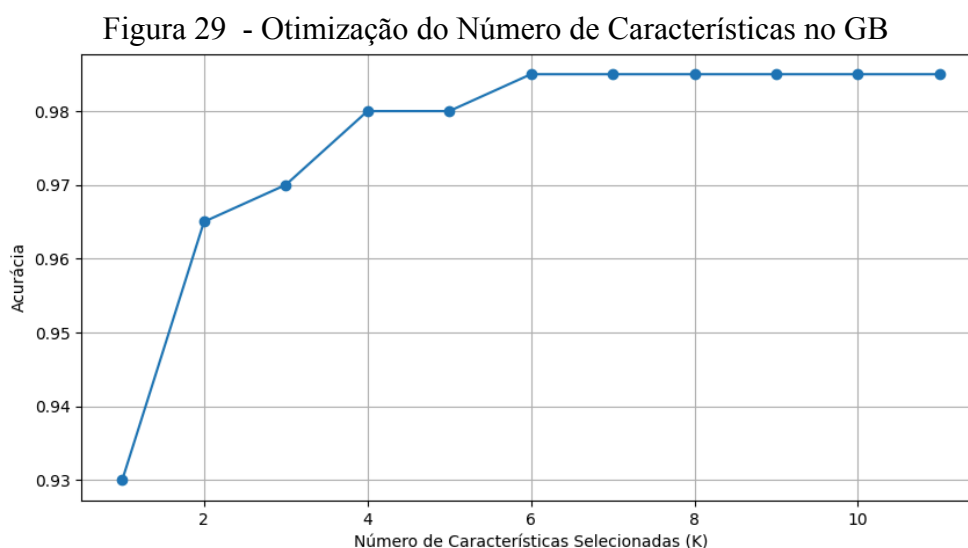
Assim sendo, o GB também foi treinado conforme o cenário descrito nos algoritmos anteriores: inicialmente foram consideradas todas as variáveis presentes no conjunto de dados; e, depois, somente as características com maior relação com o alvo, as quais foram selecionadas a partir da técnica *SelectKBest*. No contexto de treinamento do GB³⁵, foi utilizada a função *GradientBoostingClassifier*³⁶ da biblioteca *Scikit-learn*, que disponibiliza uma implementação do algoritmo GB voltada para tarefas de classificação.

A Figura 26 ilustra o processo de seleção das características que apresentam uma maior relação com o alvo, no contexto do GB. A partir disso, observa-se que a acurácia do

³⁵ <https://tinyurl.com/gb-training>

³⁶ <https://tinyurl.com/SciKitGradientBoostin>

modelo aumenta à medida que o número de variáveis cresce, estabilizando-se com um valor de 6. Assim, a partir desse ponto, a inclusão de novos atributos não gera ganhos de desempenho, indicando que o modelo alcança seu melhor resultado com somente um subconjunto dos dados disponíveis, o qual é composto por *Gender*, *Age*, *HbA1c*, *Cholesterol*, *Triglycerides* e *BMI*. Ademais, é importante destacar que, ao comparar o resultado dessa seleção com as dos algoritmos anteriores, somente três variáveis se repetem, *Age*, *HbA1c* e *BMI*, sugerindo, então, que, embora o GB extraia valor preditivo adicional de outros atributos, existe um núcleo de características fortemente associados ao diagnóstico da diabetes, o qual se mantém consistente independente do modelo adotado.



Fonte: Autoria própria

Assim como nos modelos anteriores, o *Grid Search* foi utilizado para otimizar os parâmetros do algoritmo. Para o GB, foram considerados os seguintes padrões: número de estimadores, taxa de aprendizado e profundidade máxima das árvores. Assim, após a avaliação de todas as combinações possíveis, o melhor desempenho foi proporcionado por um total de 100 estimadores, uma taxa de 0,1 para o aprendizado e profundidade máxima igual a 3 para as árvores. Desse modo, esses valores indicam que o modelo se saiu melhor com valores moderados para cada um dos parâmetros, sugerindo um bom equilíbrio entre complexidade e capacidade de generalização.

A Tabela 5 apresenta os resultados obtidos com o treinamento do GB, sendo possível observar que os valores das métricas foram iguais para o modelo que considerou todas as características e o que considerou somente as seleccionadas pelo *SelectKBest*, indicando que a

força preditiva está concentrada em um conjunto reduzido dos dados. Considerando o contexto geral, a acurácia do modelo atingiu o valor de 98%, indicando que o GB pôde classificar corretamente a maioria das instâncias do conjunto de dados. Entretanto, por se tratar de um problema de classificação multiclasse, é importante observar outras métricas baseadas nos resultados de cada uma das classes.

Tabela 6 - Resultados do Treinamento do Algoritmo GB

Atributos usados	Acurácia	Precisão			<i>Recall</i>			<i>F1-Score</i>		
		Classe 0	Classe 1	Classe 2	Classe 0	Classe 1	Classe 2	Classe 0	Classe 1	Classe 2
Total	98%	1.00	0.91	0.91	0.98	1.00	1.00	0.99	0.95	0.95
Otimizado	98%	1.00	0.91	0.91	0.98	1.00	1.00	0.99	0.95	0.95

Fonte: Autoria própria

No que se refere a precisão, o modelo apresentou um alto desempenho para todas as classes: a classe de predição positiva da diabetes atingiu 100%, seguido de 91% para as classes de pré-diabetes e não diabetes. Nesse contexto, destaca-se o resultado da Classe 0, o qual indicou a ausência de falsos positivos, ou seja, todas as previsões positivas para essa classe estavam corretas. Ademais, os resultados das demais classes apontam um leve grau de incerteza, porém ainda em um intervalo que garanta um alto desempenho preditivo. Em termos gerais, os resultados para essa métrica demonstram que o modelo consegue, além de identificar corretamente os casos positivos, evitar atribuições incorretas de indivíduos saudáveis em categorias incorretas, o que é essencial para aplicações clínicas.

Com relação ao *recall*, observa-se, novamente, um alto desempenho para todas as classes, com valores de 98% para a Classe 0 e 100% para as Classes 1 e 2. Nesse cenário, o GB se mostrou sensível para todas as categorias, identificando praticamente todos os casos verdadeiros presentes no conjunto de testes. Assim, esse desempenho é especialmente importante para o contexto de diagnóstico médico, já que minimiza o risco de subdiagnóstico, ou seja, a possibilidade de que pacientes com diabetes não sejam identificados corretamente pelo modelo. Além disso, o *F1-Score* também apresentou resultados elevados e balanceados, com 99% para a Classe 0 e 95% para as Classes 1 e 2. Isso evidencia, então, que o modelo mantém um equilíbrio entre sua capacidade de identificar corretamente os casos e de evitar classificações incorretas.

Por fim, a Figura 27 ilustra a matriz de confusão obtida no treinamento do GB, a qual reforça o desempenho eficiente do algoritmo na tarefa de classificação multiclasse da diabetes. Com relação à Classe 0, percebe-se que foram identificados corretamente 166 casos como diabéticos, e somente três erros de classificação foram apontados, dos quais dois foram atribuídos como não diabéticos e um como pré-diabético. Já a Classe 1 apresentou um desempenho sem erros de classificação, com todos os 10 indivíduos sendo corretamente apontados como pré-dispostos a doença. Similar a anterior, a Classe 2 também classificou corretamente todas as 21 instâncias como não diabéticos. Dessa forma, o resultado observado não aponta somente um bom desempenho geral para o modelo, mas também um equilíbrio entre as predições das diferentes categorias, característica essencial para o contexto clínico, no qual a diferenciação dos estágios da doença é crucial.

Figura 30 - Matriz de Confusão Para o Modelo GB

Real	0	1	2
	166	1	2
	0	10	0
2	0	0	21
	0	1	2
	Predito		

Fonte: Autoria própria

4.2.6 Treinamento da *Decision Tree* (DT)

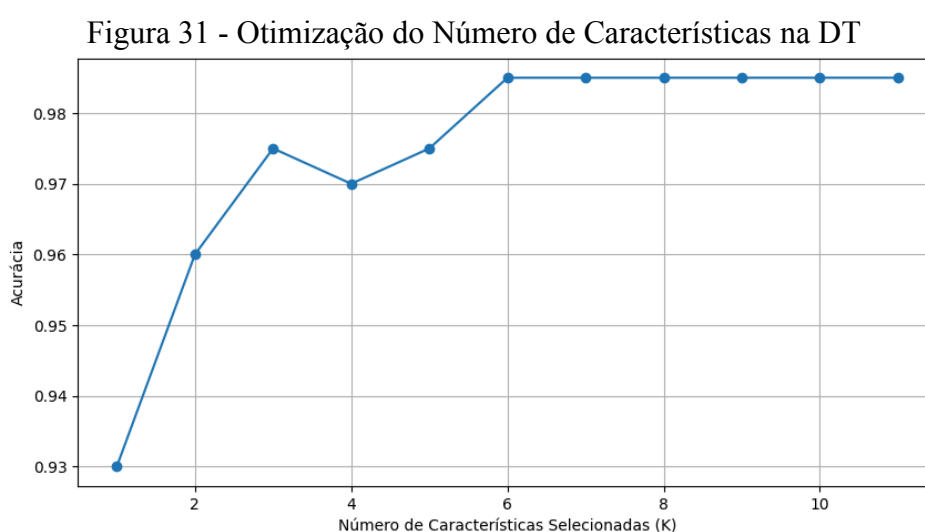
Segundo Faceli *et al.* (2021), a DT é baseada no conceito de “*dividir para conquistar*”, ou seja, no contexto do ML, um problema complexo é decomposto em subproblemas simples, aos quais se aplica, recursivamente, a mesma estratégia de solução. Desse modo, ainda segundo os autores, os resultados combinados desses subproblemas são representados em uma árvore, que expressa a decisão, ou resultado, para o problema original. Géron (2021), aponta a DT como um algoritmo de ML polivalente, podendo ser aplicado para tarefas de regressão, classificação, ou, até mesmo, de *multioutput*³⁷. Além disso, é importante

³⁷ *Multioutput*, no contexto do ML, se refere a algoritmos capazes de abordar problemas com múltiplas entradas e múltiplas saídas.

apontar que uma das principais vantagens da DT é sua interpretabilidade, uma vez que seus resultados são intuitivos e fáceis de serem entendidos (Géron, 2021).

Uma DT é composta, normalmente, por um nó raiz, diversos nós internos e vários nós folha (Zhou, 2021). Segundo Zhou (2021), os nós folhas representam os resultados de uma decisão, enquanto os demais correspondem a testes de atributo. Nesse contexto, e seguindo a mesma abordagem adotada nos demais algoritmos, a DT foi treinada³⁸ em dois cenários distintos: utilizando todas as características disponíveis no conjunto de dados e, posteriormente, considerando somente as variáveis selecionadas, a partir da técnica *SelectKBest*, por apresentarem maior relevância em relação à variável-alvo. Para isso, foi utilizada a classe *DecisionTreeClassifier*³⁹ da biblioteca *Scikit-learn*, que disponibiliza uma implementação do algoritmo DT para o cenário de classificação multiclasse.

A Figura 28 representa o processo de otimização do modelo DT com relação à quantidade de atributos selecionados a partir da técnica *SelectKBest*. Sendo assim, observa-se que a acurácia do modelo aumenta à medida que mais variáveis são incorporadas no processo de treinamento, apresentando um aumento expressivo entre os valores um e três. Além disso, o ápice do desempenho é atingido ao selecionar um total de seis características, incluindo: *Gender*, *Age*, *HbA1c*, *Cholesterol*, *TG* e *BMI*, se mantendo estável a partir desse ponto. Desse modo, tal comportamento indica que o modelo alcança seu melhor desempenho com um conjunto reduzido de características, sendo que a inclusão das outras variáveis não resulta em ganho significativo de desempenho.



Fonte: Autoria própria

³⁸ <https://tinyurl.com/dt-training>

³⁹ <https://tinyurl.com/SciKitDecisionTree>

A técnica *Grid Search* foi utilizada para otimizar os parâmetros do modelo treinado. No contexto da DT, o espaço de busca foi definido com base em quatro critérios: profundidade máxima da árvore, número mínimo de amostras para a divisão de um nó, número mínimo de amostras em uma folha e a função de avaliação da qualidade da divisão. Após a avaliação de todas as combinações possíveis, os resultados obtidos foram entropia para o critério de avaliação, 5 para a profundidade e 2 para o mínimo de amostras em uma folha e para o mínimo de divisão de um nó. Com base nessa configuração, percebe-se que o modelo obteve um melhor desempenho com uma profundidade controlada para a árvore, além de critérios de divisão mais conservadores, favorecendo a generalização sem comprometer a capacidade de aprendizado.

A Tabela 6 apresenta os resultados do treinamento do algoritmo DT para os cenários com todas as características e o de seleção de atributos. Assim como no GB, os valores das métricas foram iguais em ambos os casos, consequência da distribuição da força preditiva do modelo em um conjunto reduzido de variáveis. Em termos gerais, a acurácia da DT atingiu o valor de 98%, refletindo um desempenho estável e capaz de classificar corretamente a maioria das instâncias do conjunto de testes. Com relação à redução do número de atributos, permite-se afirmar que o processo de seleção consegue alcançar um alto desempenho mesmo com um conjunto reduzido e informativo de características, favorecendo a simplicidade e a interpretabilidade do modelo.

Tabela 7 - Resultados do Treinamento do Algoritmo DT

Modelo DT	Acurácia	Precisão			Recall			F1-Score		
		Classe 0	Classe 1	Classe 2	Classe 0	Classe 1	Classe 2	Classe 0	Classe 1	Classe 2
Total	98%	1.00	0.91	0.88	0.98	1.00	1.00	0.99	0.95	0.93
Otimizado	98%	1.00	0.91	0.88	0.98	1.00	1.00	0.99	0.95	0.93

Fonte: Autoria própria

Entretanto, ao se observar o contexto de classificação multiclasse presente na problemática abordada neste trabalho, é importante considerar outras métricas de desempenho voltadas para cada uma das classes que compõem a variável alvo. Nesse contexto, a precisão obteve valores satisfatórios em seus resultados, com 100% para a classe de predição positiva da diabetes, 91% para a de pré-disposição a doença e 88% para a classificação negativa. Os

números das classes 1 e 2 demonstram uma leve tendência em classificar incorretamente instâncias destas categorias, porém ainda em um limite aceitável.

Com relação ao *recall*, evidenciou-se um cenário de excelente desempenho para todas as classes. Nessa situação, as Classes 1 e 2 obtiveram um valor de 100%, enquanto a Classe 0 obteve 98%, demonstrando que quase todos os pacientes diabéticos foram classificados corretamente. Esse desempenho é, então, relevante para o cenário médico, já que a não identificação de casos positivos pode acabar gerando consequências clínicas para o paciente. Assim, um valor elevado de *recall* em todas as categorias reforça a sensibilidade do modelo, minimizando falsos negativos. O *F1-Score* apresentou valores elevados para todas as classes, com 99% para a Classe 0, 95% para a Classe 1 e 93% para a Classe 2. Tais resultados indicam que a DT treinada apresenta um balanceamento entre a capacidade de identificar corretamente casos da doença e de evitar classificações equivocadas.

A Figura 29 ilustra a matriz de confusão do modelo DT tanto no cenário com todas as variáveis, quanto no modelo otimizado com seleção de características, sendo os resultados iguais para ambos os casos. Com relação à Classe 0, foram identificados corretamente 165 instâncias, com somente quatro erros, três classificados com não diabéticos e um como pré-diabético. A Classe 1 teve todas as suas instâncias classificadas corretamente, com todos os 10 indivíduos identificados como pré-diabéticos. Do mesmo modo, a Classe 2 também classificou corretamente todas as suas instâncias, totalizando 21 acertos. Esses resultados evidenciam a alta capacidade do modelo em distinguir corretamente as diferenças clínicas entre cada uma das classes.

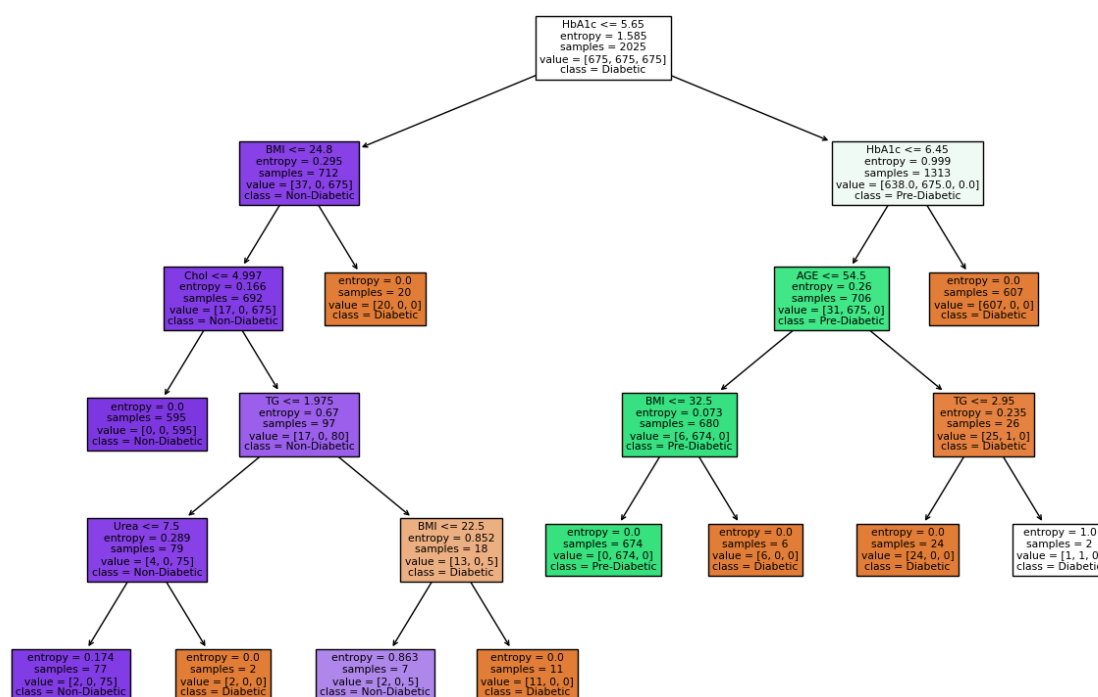
Figura 32 - Matriz de Confusão Para o Modelo DT

Real	0	165	1	3
	1	0	10	0
	2	0	0	21
		0	1	2
		Predito		

Fonte: Autoria própria

Por fim, a Figura 30 apresenta a estrutura da árvore de decisão gerada pelo modelo DT, que foi idêntica para ambos os cenários de treinamento descritos anteriormente. A raiz da árvore é formada pela variável HbA1c, indicando sua forte relevância na distinção entre os grupos, especialmente na separação inicial entre pacientes diabéticos, pré-diabéticos e não diabéticos. Em seguida, outras variáveis clínicas aparecem ao longo dos ramos, incluindo BMI, Age, Triglycerides e Urea, demonstrando sua contribuição para a segmentação dos diferentes perfis relacionados à doença. Em termos gerais, a árvore mostra-se bem balanceada e com baixa profundidade, sugerindo boa capacidade de generalização e interpretabilidade. Além disso, as regiões terminais com entropia igual a 0 indicam nós puros, ou seja, classificações realizadas com alto grau de confiança.

Figura 33 - Estrutura da DT treinada



Fonte: Autoria própria

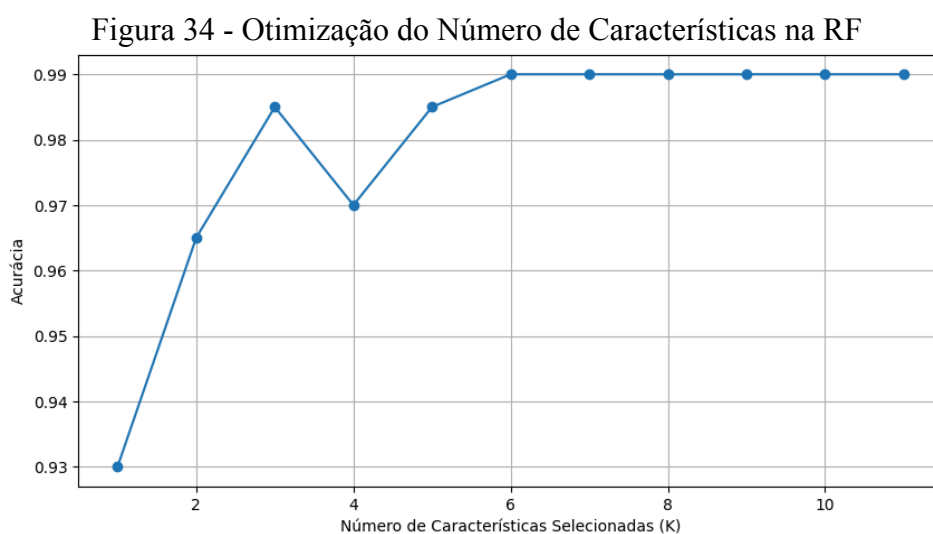
4.2.7 Treinamento da *Random Forest* (RF)

O aprendizado por *ensemble*, ou aprendizado por agrupamento, é baseado no conceito de sabedoria das multidões. Segundo Zhou (2021), esse tipo de abordagem consiste em treinar e combinar múltiplos modelos de ML, para solucionar um mesmo problema de forma mais eficiente e robusta. Um exemplo de método *ensemble* envolve, de acordo com Géron (2021),

a combinação de várias DT, onde cada uma é treinada com uma amostra aleatória do conjunto de treinamento. Assim, a partir das previsões individuais de cada árvore, a classe final é definida com base no voto majoritário. Esse algoritmo é conhecido como RF, uma técnica que associa diversas DTs para reduzir o risco de *overfitting*, aprimorando, assim, a capacidade de generalização do modelo (Ho, 1995).

Nesta proposta, o RF foi treinado⁴⁰ utilizando a classe *RandomForestClassifier*⁴¹, da biblioteca *Scikit-learn*, que disponibiliza uma implementação desse algoritmo, baseada em conjuntos de árvores de decisão, para tarefas de classificação multiclasse. Além disso, assim como nas propostas anteriores, o RF foi implementado em dois contextos distintos: inicialmente com todas as variáveis disponíveis no conjunto de dados e, posteriormente, utilizando somente as características mais relevantes selecionadas a partir da técnica *SelectKBest*.

A Figura 31 apresenta o processo de otimização do número de variáveis para o modelo RF, utilizando o método de seleção univariada *SelectKBest*. Ao observar os resultados expressos na imagem, percebe-se que a acurácia do modelo aumenta significativamente até o ponto em que seis características são selecionadas, atingindo uma acurácia máxima de aproximadamente 99%. A partir desse ponto, a inclusão de novos atributos não resulta em melhorias adicionais no desempenho, o que indica que o modelo atinge o seu potencial utilizando somente um conjunto reduzido, composto por: *Gender*, *Age*, *HbA1c*, *Cholesterol*, *TG*, *BMI*, das variáveis disponíveis no conjunto de dados.



Fonte: Autoria própria

⁴⁰ <https://tinyurl.com/training-rf>

⁴¹ <https://tinyurl.com/SciKit-RandomForest>

Com base na análise da Figura 31, percebe-se que a estabilidade da acurácia nos valores mais altos de K sugere que as variáveis mais relevantes concentram grande parte da capacidade preditiva do modelo, ou seja, o RF tende a ser eficiente mesmo com um conjunto de entrada mais enxuto.

Visando a otimização dos hiperparâmetros do modelo RF, foi aplicada a técnica *Grid Search* com validação cruzada, a fim de identificar a configuração que maximiza o desempenho preditivo do algoritmo. Para o RF, o espaço de busca considerou os seguintes parâmetros: número de estimadores, profundidade máxima das árvores, número mínimo de amostras para realizar uma divisão e número de amostras em uma folha. Desse modo, após a avaliação de todas as combinações, a melhor configuração encontrada foi: 100 para a quantidade de estimadores; *None* para a profundidade das árvores; 2 para o valor mínimo de amostras para divisão; e 1 para o valor de amostras em uma folha. Esses valores indicam, então, que o RF obteve melhor desempenho com árvores totalmente expandidas e sem restrições de profundidade, condizente com a natureza do algoritmo, que mitiga o risco de *overfitting* ao combinar múltiplas árvores profundas de forma agregada e controlada.

Os resultados obtidos com o treinamento do modelo RF, tanto no cenário com todas as variáveis disponíveis, quanto na versão otimizada com seleção de características, estão apresentados na Tabela 7. Assim como observado nos demais algoritmos avaliados, as métricas de desempenho foram idênticas em ambos os contextos, reforçando a evidência de que a capacidade preditiva do modelo está concentrada em um subconjunto específico das variáveis do conjunto de dados. Em termos gerais, a acurácia do RF foi de 99% para ambos os cenários. Esse resultado, reflete o desempenho elevado do modelo em tarefas de classificação multiclasse, podendo prever corretamente quase todas as instâncias. Ademais, a persistência do valor da métrica mesmo após a redução do número de atributos confirma que o desempenho do RF é sustentado por um conjunto reduzido, porém informativo, de variáveis.

Tabela 8 - Resultados do Treinamento do Algoritmo RF

Modelo RF	Acurácia	Precisão			Recall			F1-Score		
		Classe 0	Classe 1	Classe 2	Classe 0	Classe 1	Classe 2	Classe 0	Classe 1	Classe 2
Total	99%	1.00	1.00	0.91	0.99	1.00	1.00	0.99	1.00	0.95
Otimizado	99%	1.00	1.00	0.91	0.99	1.00	1.00	0.99	1.00	0.95

Fonte: Autoria própria

No que diz respeito à precisão, os resultados obtidos também foram elevados. Para essa métrica, as Classes 0 e 1 conseguiram atingir um total de 100%, indicando que todas as instâncias previstas como pertencentes a essas classes estavam corretas, ou seja, não houve a ocorrência de falsos positivos. Ademais, a Classe 2 obteve uma precisão de 91%, sugerindo uma leve tendência do modelo em superestimar essa classificação. De forma geral, a alta precisão para todas as classes reforça a confiabilidade do modelo, especialmente para contextos clínicos, nos quais decisões equivocadas podem comprometer o diagnóstico.

O *recall* também apresentou valores elevados, com 100% para as Classes 1 e 2, indicando que todas as instâncias reais dessas categorias foram corretamente identificadas. Além disso, a Classe 0 apresentou *recall* de 99%, sinalizando que uma pequena quantidade de indivíduos diabéticos não foram detectados pelo modelo. Esse desempenho evidencia, dessa forma, a alta sensibilidade do RF para o contexto de predição da diabetes, fundamental para minimizar o número de falsos negativos, especialmente no diagnóstico de condições de saúde, no qual a falha na identificação pode representar riscos importantes ao paciente.

Os resultados do *F1-Score* confirmam a robustez do modelo. Nesse contexto, a Classe 0 apresentou um valor de 99%, enquanto as Classes 1 e 2 registraram *F1-scores* de 100% e 95%, respectivamente. Tais resultados indicam que o modelo mantém um desempenho equilibrado para todas as suas classes, com destaque para a estabilidade entre os dois cenários de treinamento.

Por fim, a Figura 32 ilustra a matriz de correlação para o modelo RF, na qual a Classe 0 foi corretamente classificada em 167 ocorrências, com somente duas instâncias sendo classificadas erroneamente como não diabéticos. Ademais, as Classes 1 e 2 foram previstas corretamente em todas as instâncias. Dessa forma, observa-se uma alta capacidade discriminativa por parte do modelo, principalmente ao diferenciar os diferentes estágios clínicos da diabetes, uma vez que a ausência de erros em classes com características próximas, como pré-diabéticos e diabéticos, indica que o modelo capturou padrões relevantes nos dados.

Figura 35 - Matriz de Confusão Para o Modelo RF

Real	0	1	2
0	167	0	2
1	0	10	0
2	0	0	21
	0	1	2
	Predito		

Fonte: Autoria própria

4.3 Discussão dos Resultados

Esta seção apresenta uma análise crítica a respeito dos resultados da aplicação dos modelos de ML no contexto preditivo da diabetes, considerando as métricas de avaliação: acurácia, *recall*, precisão e *F1-Score*, além dos dois cenários distintos de treinamento apresentados no estudo: utilizando todas as características disponíveis e empregando somente os melhores atributos selecionados. Além disso, é realizada uma comparação entre os resultados alcançados neste estudo e os obtidos em trabalhos anteriores, permitindo verificar avanços e contribuições desta pesquisa com relação ao estado da arte.

A Tabela 9 apresenta os resultados do desempenho dos modelos de ML treinados no contexto de predição da diabetes. Assim, observa-se que, ao considerar o contexto de treinamento que envolveu todas as variáveis disponíveis, os modelos apresentaram desempenhos elevados de acurácia, variando de 94%, para o KNN e o NB, a 99%, para o RF. Nessa situação, entre as técnicas analisadas, o RF destacou-se ao apresentar o melhor desempenho geral, atingindo a acurácia mais elevada entre os modelos treinados, aproximadamente 99%, além dos valores de 97% para a precisão, 99% para o *recall* e 98% para o *F1-Score*.

Tabela 9 - Comparação do Desempenho dos Modelos de ML Treinados Com Todas as Características e Com as Melhores Características (*SelectKBest*).

Modelo	Todas características				Melhores características			
	Acurácia	Precisão	<i>Recall</i>	F1-Score	Acurácia	Precisão	<i>Recall</i>	F1-Score
KNN	94%	≈ 83%	≈ 91%	≈ 87%	97%	≈ 91%	≈ 97%	≈ 94%
SVM	96%	≈ 87%	≈ 98%	≈ 92%	97%	≈ 90%	≈ 98%	≈ 94%
LR	95%	≈ 83%	≈ 93%	≈ 88%	94%	≈ 79%	≈ 96%	≈ 86%
GB	98%	≈ 94%	≈ 99%	≈ 96%	98%	≈ 94%	≈ 99%	≈ 96%
DT	98%	≈ 94%	≈ 99%	≈ 95%	98%	≈ 94%	≈ 99%	≈ 95%
RF	99%	≈ 97%	≈ 99%	≈ 98%	99%	≈ 97%	≈ 99%	≈ 98%
NB	94%	≈ 79%	≈ 94%	≈ 85%	96%	≈ 89%	≈ 97%	≈ 92%

Fonte: Autoria própria.

Conforme visto na Tabela 9, os modelos GB e DT também apresentaram resultados robustos para esse cenário, ambos apresentando 98% de acurácia e F1-Score próximo aos 96%, reforçando a eficácia dos métodos baseados em árvores para a predição da diabetes. Por outro lado, algoritmos lineares como a LR e o NB apresentaram resultados inferiores, especialmente em termos de precisão, apresentando valores próximos a 83% (LR) e 79% (NB), indicando uma maior ocorrência de falsos positivos. Por fim, o KNN e o SVM apresentaram desempenhos favoráveis, com 94% de acurácia para o KNN e 96% para o SVM.

A partir da redução do número de características, em decorrência do processo de seleção de atributos mais relevantes para a variável-alvo, observou-se uma melhora sutil em boa parte dos modelos. Para esse contexto de treinamento, destaca-se o desempenho do KNN, cujo F1-Score aumentou de 87% para 94%, indicando que a seleção de características contribuiu para a redução de ruídos no espaço de vizinhança. Além disso, para o NB, também foi visível uma melhora significativa: de 85% para 92%, evidenciando que modelos probabilísticos se beneficiam de dados mais informativos e enxutos. Por fim, ao considerar os modelos baseados em árvores, GB, DT e RF, percebe-se que o impacto da seleção de atributos foi mínimo, mantendo seus valores semelhantes ao cenário de treinamento com todas as variáveis, comportamento resultante do fato desses algoritmos apresentarem mecanismos internos de avaliação de importância de atributos.

De modo geral, os resultados confirmaram que modelos *ensemble*, como o RF e o GB, são altamente eficazes para a predição da diabetes, mesmo sem redução de dimensionalidade. Ademais, a seleção de características mostrou-se benéfica para modelos mais sensíveis a dimensionalidade, como KNN e NB, os quais apresentaram ganhos significativos em métricas de equilíbrio entre precisão e *recall*. Além disso, o alto valor da métrica *recall* em quase todos os algoritmos treinados, em especial para a DT, o RF e o GB, com 99%, indica que a maioria dos pacientes diabéticos foram corretamente identificados, o que é essencial para aplicações na área médica, em que falsos positivos podem ter graves consequências. Por outro lado, a precisão mais baixa em alguns modelos, como o NB e o LR, sugere mais falsos positivos, o que pode implicar em uma sobrecarga de recursos em cenários clínicos reais.

Tabela 10 - Comparação Entre Acurácia e Valores Reportados em Trabalhos Anteriores

Modelo	Acurácia (Todas as características)	Acurácia (Melhores as características)	Estudos anteriores (Acurácia)
KNN	94%	97%	Soni e Varma (2020): $\approx 69\%$
			Febrian <i>et al.</i> (2023): $\approx 73\%$
			Khanam e Foo (2021): $\approx 79\%$
SVM	96%	97%	Soni e Varma (2020): $\approx 73\%$
			Olisah, Smith e Smith (2022): $\approx 96\%$
			Abdelhafez e Amer (2024): $\approx 87\%$
			Khanam e Foo (2021): $\approx 77\%$
LR	95%	94%	Soni e Varma (2020): $\approx 75\%$
			Chou, Hsu e Chou (2023): $\approx 80\%$
			Bhat <i>et al.</i> (2023): $\approx 81\%$
			Abdelhafez e Amer (2024): $\approx 91\%$
			Khanam e Foo (2021): $\approx 78\%$
GB	98%	98%	Soni e Varma (2020): $\approx 75\%$
			Bhat <i>et al.</i> (2023): $\approx 89\%$
DT	98%	98%	Soni e Varma (2020): $\approx 75\%$
			Chou, Hsu e Chou (2023): $\approx 95\%$
			Bhat <i>et al.</i> (2023): $\approx 91\%$

			Abdelhafez e Amer (2024): $\approx 99\%$
			Khanam e Foo (2021): $\approx 73\%$
RF	99%	99%	Soni e Varma (2020): $\approx 77\%$
			Abdelhafez e Amer (2024): $\approx 99\%$
			Olisah, Smith e Smith (2022): $\approx 98\%$
			Khanam e Foo (2021): $\approx 77\%$
NB	94%	96%	Khanam e Foo (2021): $\approx 78\%$
			Abdelhafez e Amer (2024): $\approx 91\%$
			Febrian <i>et al.</i> (2023): $\approx 76\%$

Fonte: Autoria própria

A Tabela 10 apresenta a comparação entre as acurácias obtidas neste estudo e os valores reportados nos trabalhos relacionados, para fins de comparação. De forma geral, observa-se que os modelos treinados nesta pesquisa alcançaram desempenhos superiores ou similares aos encontrados na literatura, demonstrando a eficácia da abordagem adotada para o treinamento, a qual considerou a seleção de atributos, o balanceamento dos dados e o ajuste dos hiperparâmetros.

Com relação ao KNN, que obteve uma acurácia de 94% no contexto de treinamento com todas as características e 97% após a seleção dos melhores atributos, percebe-se valores acima dos reportados por Soni e Varma (2020), com cerca de 69%, e por Febrian *et al.* (2023), com aproximadamente 73%. Além disso, para o SVM, este estudo obteve os valores de 96% e 97% em termos de acurácia, superando estudos anteriores como os de Soni e Varma (2020), Abdelhafez e Amer (2024), Khanam e Foo (2021), que obtiveram valores entre 73% e 87%, e se assimilando aos resultados apresentados por Olisah, Smith e Smith (2022), que obtiveram uma acurácia de 96%.

Considerando os algoritmos lineares abordados no estudo, percebe-se que a LR, com os valores de 95% e 96% de acurácia, apresentou um desempenho superior a trabalhos anteriores, como os de Soni e Varma (2020), Chou, Hsu e Chou (2023), Bhat *et al.* (2023), Abdelhafez e Amer (2024) e Khanam e Foo (2021), que apresentaram valores entre 73% e 87%. Além disso, o NB apresentou 94% e 96% de acurácia, sendo também superior aos trabalhos anteriores, como os de Khanam e Foo (2021), Abdelhafez e Amer (2024) e Febrian *et al.* (2023), que apresentaram acurácia entre 71% e 78%. Por fim, com relação aos

algoritmos baseados em árvores, observou-se que a DT, com 98% de acurácia, obteve resultados similares aos estudos anteriores, sendo, inclusive, superada no estudo proposto por Abdelhafez e Amer (2024), no qual foi obtido o valor de 99%. Tal comportamento também foi similar para o RF, o qual obteve o valor de 99% de acurácia, se assimilando aos estudos de Abdelhafez e Amer (2024) e Olisah, Smith e Smith (2022). Ademais, o modelo GB atingiu 98% de acurácia neste estudo, superando os valores obtidos por Soni e Varma (2020), com 75%, e Bhat *et al.* (2023), com 89%.

Com base no apresentado, e dentre todos os modelos avaliados, o RF treinado no contexto de seleção de atributos foi o algoritmo selecionado para a construção da API Diagly devido ao seu desempenho superior em todas as métricas avaliadas. Em termos gerais, esse modelo apresentou 99% de acurácia, 97% de precisão, 99% de *recall* e 98% de F1-Score, mantendo esses valores tanto com todas as variáveis quanto após a aplicação do *SelectKBest*. Dessa forma, tal consistência aponta não somente para uma robustez, mas também sua capacidade de lidar com a correlação entre as características presentes na base de dados. Além disso, a utilização do cenário reduzido de instâncias contribui para a redução da complexidade computacional, aumentando sua interpretabilidade e tornando o modelo mais eficiente para a integração em aplicações práticas, sem comprometer sua capacidade preditiva.

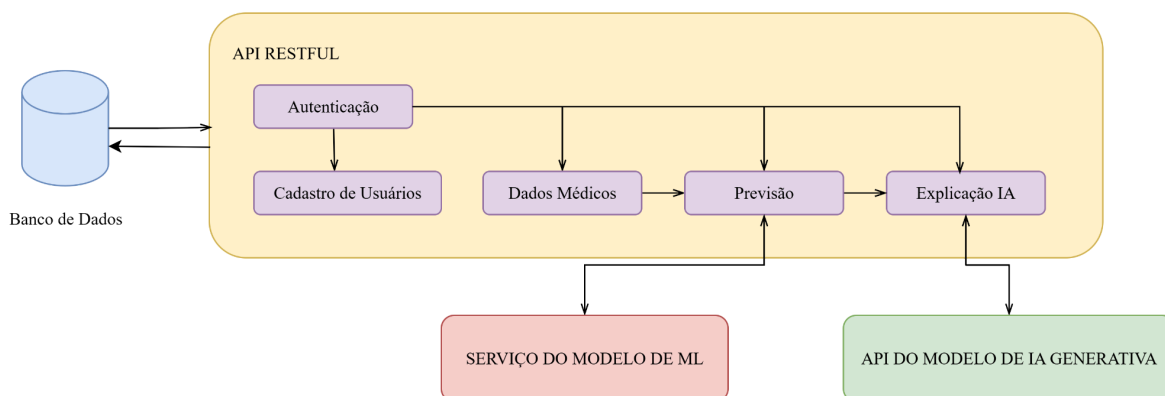
4.4 Desenvolvimento da API Diagly

A partir da seleção do modelo de melhor desempenho, a API Diagly foi desenvolvida com o propósito de disponibilizar uma ferramenta capaz de realizar previsões eficientes da ocorrência de diabetes, além de gerar explicações interpretáveis sobre os diagnósticos, por meio de técnicas de ML e IA. Ou seja, de modo geral, o objetivo da API é oferecer um serviço capaz de gerenciar informações clínicas, processar dados médicos e disponibilizar resultados preditivos de forma rápida e confiável. Ademais, a API obteve o Certificado de Registro de Programa de Computador junto ao Instituto Nacional da Propriedade Industrial (INPI), sob o Processo N°: BR512025004089-0.

A Figura 36 ilustra a arquitetura proposta para a API, seguindo o padrão *RESTful*, que possibilita a comunicação padronizada entre a aplicação cliente, os serviços internos e os externos. Em geral, o núcleo da aplicação é composto por cinco módulos principais: Autenticação, Cadastro de Usuário, Dados Médicos, Previsão e Explicação IA. Dessa forma, cada uma dessas camadas são responsáveis por gerenciar desde a segurança e controle de acesso até o processamento e disponibilização das previsões. Além disso, a Diagly apresenta

um banco de dados conectado diretamente ao núcleo *RESTful* para persistência dos dados, garantindo consistência e confiabilidade das informações.

Figura 36 - Arquitetura da API Diagly



Fonte: Autoria própria

No que se refere aos serviços, o módulo de Autenticação assegura, através do uso de *JSON Web Tokens* (JWT), que somente usuários autorizados possam interagir com as camadas da API. Além disso, o Cadastro de Usuário possibilita a criação e o gerenciamento de perfis distintos, os quais compõem as entidades que interagem com a aplicação: administradores, médicos e pacientes. Já o módulo de Dados Médicos é responsável pelo armazenamento e consulta das informações clínicas de cada paciente, enquanto a etapa de Predição realiza a comunicação com o Serviço do Modelo de ML (SMML), que contém o algoritmo RF otimizado com a seleção de atributos para predição da ocorrência da diabetes. O SMML é um serviço independente do núcleo *RESTful*, garantindo flexibilidade para atualização ou substituição do modelo sem impacto direto na API principal.

A arquitetura incorpora, também, a API do Modelo de IA Generativa (AMIG), mais especificamente a API da Cohere, que disponibiliza o LLM *Command R+*, destinada à geração de explicações interpretáveis sobre os diagnósticos. Nesse cenário, o Apêndice E apresenta o *prompt*⁴² utilizado para se comunicar com o LLM, definindo as regras para o modelo explicar de forma simplificada a justificativa do diagnóstico, destacando variáveis laboratoriais relevantes, como HbA1c, colesterol e IMC. Além disso, para a construção do *prompt* foi utilizada a técnica *Few-Shot Prompting*, a qual, segundo Ahmed *et al.* (2023, *apud* Marvin *et al.* 2024), consiste em utilizar um conjunto reduzido de *prompts* elaborados para

⁴² No contexto de IA Generativa, um *prompt* se refere a instrução dada ao modelo para que ele consiga realizar uma tarefa ou gerar um conteúdo específico.

capacitar o LLM a executar uma nova tarefa ou a adaptar-se a um novo domínio. Dessa forma, permite-se que o modelo produza respostas mais alinhadas ao contexto do problema investigado, oferecendo explicações interpretáveis que apoiam tanto profissionais de saúde quanto pacientes na compreensão do diagnóstico, como as apresentadas na Figura 37.

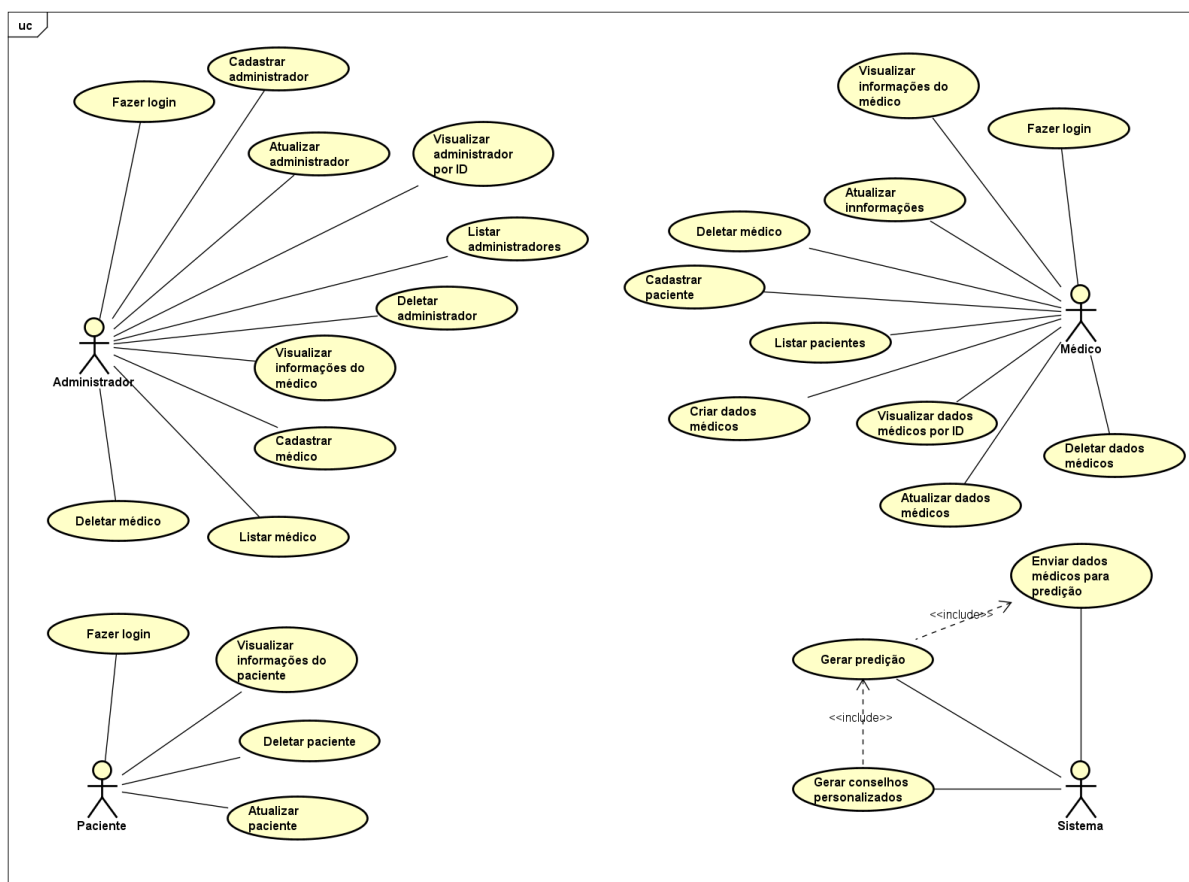
Figura 37 - Conselho Retornado Pelo AMIG Considerando um Diagnóstico Negativo

```
1 [{"tips": "**Diagnosis Explanation:**\nVincenzo's lab results indicate a non-diabetic condition, which is a reassuring finding. The HbA1c value of 4.9% is well within the healthy range, typically considered normal if below 5.7%. This measurement reflects the average blood sugar levels over the past 2-3 months, suggesting that Vincenzo's body is effectively regulating glucose. His cholesterol and triglyceride levels are also optimal, and his BMI is in the healthy weight category. These factors collectively contribute to a low risk of diabetes.\n\n**Lifestyle Tip:**\nMaintain a balanced diet rich in whole grains, lean proteins, and plenty of fruits and vegetables to keep your blood sugar and overall health in check."}]
```

Fonte: Autoria própria

A Figura 38 apresenta o diagrama de caso de uso da API Diagly, representando os principais atores do sistema e suas respectivas interações. Em geral, a modelagem contempla quatro atores distintos: Administrador, Médico, Paciente e Sistema, sendo cada um responsável por funcionalidades distintas que refletem as regras de negócio da aplicação. Com relação ao Administrador, este possui a função de gerenciar o sistema ao nível estrutural, a partir do cadastro, atualização, listagem e exclusão de novos administradores e médicos. Além disso, o Administrador também consegue visualizar informações específicas desses usuários, garantindo a manutenção e controle do ambiente de modo centralizado. Já com relação ao Médico, este é responsável por interagir com os dados clínicos. Assim, suas principais funções incluem o cadastro, atualização, listagem e exclusão de pacientes e dados médicos associados a estes. Ademais, este ator também acessa os módulos de predição e explicação, permitindo a utilização dos modelos de ML e IA que apoiam o processo de diagnóstico da diabetes.

Figura 38 - Diagrama de Caso de Uso da API Diagly



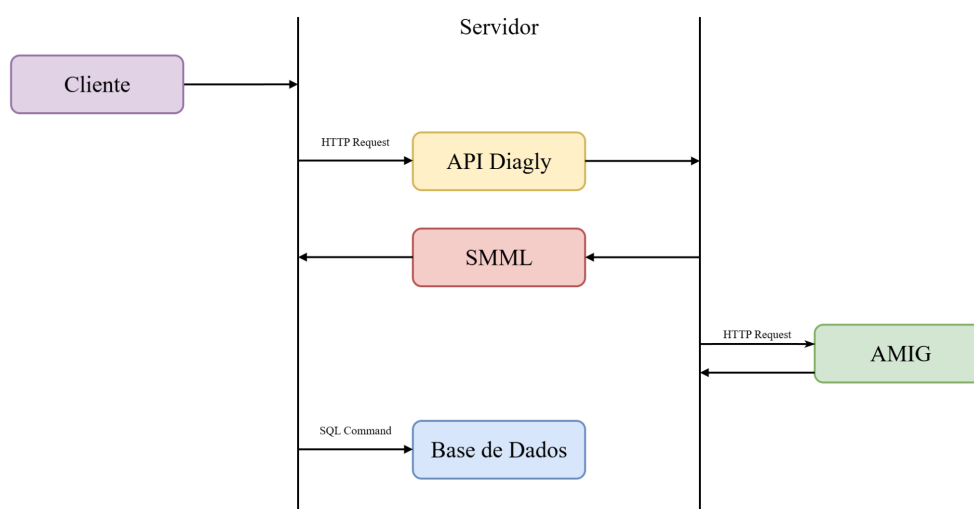
Fonte: Autoria própria

Conforme mostrado na Figura 38, os pacientes possuem acesso restrito a suas próprias informações, podendo realizar login, visualizar, atualizar e excluir seus dados pessoais. Com isso, essa limitação garante a segurança e privacidade das informações inseridas no sistema. Por fim, o Sistema é representado como um ator autônomo, o qual é encarregado de realizar operações internas relacionadas à análise preditiva. Dessa forma, esse ator envia dados médicos para o SMML, gera resultados probabilísticos da ocorrência da doença e, em seguida, utiliza a AMIG para produzir explicações e conselhos personalizados.

Um diagrama de sequência simplificado da API Diagly é representado na Figura 39, apresentando o fluxo de comunicação entre os principais componentes do sistema. De modo geral, o Cliente realiza requisições HTTP para a API, que atua como uma camada intermediária responsável por orquestrar as interações. Assim, a partir dessas solicitações, a Diagly pode consultar a base de dados por meio de comando SQL, acessar o SMML para obtenção da predição e, ainda, interagir com o AMIG para geração de explicações interpretáveis. Tal fluxo evidencia, então, a modularidade da aplicação, destacando a

separação de responsabilidades entre persistência de dados, predição e explicação, além de demonstrar a flexibilidade para futuras atualizações ou substituições de serviços sem impacto direto na API principal.

Figura 39 - Diagrama de Sequência Simplificado da API Diagly



Fonte: Autoria própria

Por fim, em termos de desempenho, os testes realizados com a API Diagly e utilizando a aplicação Postman, considerando um ambiente de desenvolvimento local e um total de 50 requisições por rota, demonstraram que o tempo médio de respostas para operações de cadastro, autenticação e consultas de dados manteve-se entre 90 e 300 ms, garantindo uma boa fluidez em seu uso em ambiente web. Além disso, para as requisições envolvendo o SSML, responsáveis pela predição da diabetes, o tempo médio foi de 300 ms, sendo este eficiente ao considerar o volume de dados utilizado. Já com relação às chamadas ao AMIG, utilizadas para a geração das explicações personalizadas a partir do uso da API da Cohere, o tempo médio foi de 2,5 segundos, o que se torna justificável pelo processamento acontecer em um serviço externo ao ambiente da aplicação⁴³. Dessa forma, em termos gerais, o tempo das requisições permaneceu em uma faixa aceitável para o contexto clínico, atendendo de forma satisfatória às necessidades de um sistema de apoio ao diagnóstico.

⁴³ Conferir o Apêndice F para os tempos médios de respostas da API em cada um de seus módulos.

5. CONCLUSÃO E TRABALHOS FUTUROS

Este trabalho teve como objetivo investigar o uso de técnicas de ML para a predição da diabetes, a partir da comparação de diferentes algoritmos de classificação e, além disso, buscou-se construir uma API, a Diagly, para disponibilizar o modelo de melhor desempenho obtido. A motivação para esse trabalho surge da relevância do diagnóstico precoce da doença e das limitações dos métodos tradicionais, os quais requerem tempo, múltiplos exames e, em alguns casos, sem apresentar uma precisão satisfatória.

A partir da revisão bibliográfica, foi possível compreender o estado da arte atual em relação à aplicação de aprendizado supervisionado para o contexto da diabetes, além de identificar bases de dados públicas que possibilitasse a aplicação de experimentos de ML. Em termos de metodologia, a pesquisa foi realizada de forma quantitativa, aplicada e experimental, sendo, então, estruturada nas etapas de levantamento dos dados, pré-processamento, treinamento dos modelos e desenvolvimento da API. Ademais, foram utilizados algoritmos de classificação clássicos, identificados como os de melhores resultados em trabalhos anteriores, incluindo: KNN, SVM, LR, GB, DT, RF e NB.

No contexto dos resultados, observou-se que todos os modelos alcançaram desempenhos satisfatórios, superando ou se aproximando, em diversos casos, os valores reportados nos estudos anteriores. Dentre eles, a RF apresentou o maior desempenho, com uma acurácia de 99%, além de métricas de precisão, *recall* e *F1-Score* também elevadas, o que motivou a sua seleção para compor a API Diagly. Além da etapa de treinamento, a pesquisa abordou a implementação de uma API *RESTfull*, a qual possibilita o uso prática do modelo em aplicações reais, integrando módulos de autenticação, cadastro de paciente, armazenamento de dados clínicos e geração de explicações interpretáveis.

No geral, o trabalho atingiu os objetivos propostos: (i) realizar uma análise comparativa entre algoritmos de ML na predição de diabetes, (ii) identificar o modelo de melhor desempenho e (iii) disponibilizá-lo por meio de uma solução tecnológica de acesso simplificado. Dessa forma, a pesquisa contribui tanto para o avanço acadêmico na área de inteligência artificial aplicada à saúde quanto para o desenvolvimento de ferramentas que podem apoiar médicos e profissionais na tomada de decisão clínica.

Embora os resultados tenham sido expressivos, algumas limitações foram identificadas e podem direcionar pesquisas futuras na área. Inicialmente, o conjunto de dados pode ser ampliado, já que a base utilizada para o treinamento do modelo não apresentou uma quantidade muito expressiva de registros. Além disso, incluir informações clínicas nacionais

pode melhorar a capacidade de generalização do modelo. Outro ponto de destaque é realizar uma validação clínica do modelo treinado com dados reais de pacientes, buscando validar a sua eficácia prática e aceitabilidade pelos profissionais da área de saúde. Nesse contexto, aplicar técnicas avançadas de explicabilidade do modelo, como o *Explainable AI*, pode aumentar a interpretabilidade dos diagnósticos, podendo ajudar com o grau de aceite por parte dos profissionais. Ademais, expandir o uso da API para o contexto multiplataforma, atingindo aplicativos móveis e sistemas hospitalares, é uma forma de expandir a disponibilização do modelo treinado. Por fim, a exploração de modelos híbridos, a partir da combinação de ML e DL, pode potencializar os resultados preditivos.

Assim, a partir do exposto, este trabalho abre caminho para o desenvolvimento contínuo de soluções inteligentes voltadas à saúde, reforçando o papel do ML como aliado na prevenção e diagnóstico precoce da diabetes, com potencial de reduzir custos, aumentar a eficiência e, sobretudo, melhorar a qualidade de vida dos pacientes.

REFERÊNCIAS BIBLIOGRÁFICAS

ABDELHAFEZ, Hoda A.; AMER, Abeer A. Machine Learning Techniques for Diabetes Prediction: A Comparative Analysis. **Journal of Applied Data Sciences**, [s. l.], ano 2024, v. 5, ed. 2, p. 792-807, 6 jun. 2024. Disponível em: <https://doi.org/10.47738/jads.v5i2.219>. Acesso em: 24 dez. 2024.

AWS. Fundamentals of Machine Learning and Artificial Intelligence. *In*: AWS. **AWS Skill Builder**. [S. l.], 2025. Disponível em: <https://tinyurl.com/mrxrvytt>. Acesso em: 3 jul. 2025.

BERTOZZO, Richard Junior. **Aplicação de Machine Learning em dataset de consultas médicas do SUS**. Orientador: Jose Leomar Todesco. 2019. 100 f. Trabalho de conclusão de curso (Bacharelado em Sistemas de Informação) - Universidade Federal de Santa Catarina, Florianópolis, 2019. Disponível em: <https://repositorio.ufsc.br/handle/123456789/202663>. Acesso em: 19 maio 2025.

BHAT, Salliah Shaf; BANU, Madhina; ANSARI, Gufran Ahmad; SELVAM, Venkatesan. A risk assessment and prediction framework for diabetes mellitus using machine learning algorithms. **Healthcare Analytics**, [s. l.], ano 2023, ed. 4, 23 out. 2023. Disponível em: <https://doi.org/10.1016/j.health.2023.100273>. Acesso em: 24 dez. 2024.

Brazil Diabetes Report. *In*: INTERNATIONAL DIABETES FEDERATION (IDF). **IDF Diabetes Atlas**. 10. ed. [S. l.], 2021. Disponível em: <https://diabetesatlas.org/data/en/>. Acesso em: 14 fev. 2025.

Brasil já tem cerca de 20 milhões de pessoas com diabetes. *In*: **Sociedade Brasileira de Diabetes**. [S. l.], 31 jan. 2025. Disponível em: <https://diabetes.org.br/brasil-ja-tem-cerca-de-20-milhoes-de-pessoas-com-diabetes/>. Acesso em: 14 fev. 2025.

Brown, Ethan. **Web Development with Node and Express: Leveraging the JavaScript Stack**. 2. ed. atual. Sebastopol: O'Reilly Media, 2019. 343 p. ISBN 978-1492053514.

Chawla, N. V.; Bowyer, K. W.; Hall, L. O.; Kegelmeyer, W. P. SMOTE: Synthetic Minority Over-sampling Technique. **Journal of Artificial Intelligence Research**, [s. l.], v. 16, 1 jun. 2002. Disponível em: <https://doi.org/10.1613/jair.953>. Acesso em: 23 maio 2025.

CHERNY, Boris. **Programming TypeScript: Making Your JavaScript Applications Scale**. Sebastopol: O'Reilly Media, 2019. 302 p. ISBN 978-1-492-03765-1.

CHOU, Chun-Yang; HSU, Ding-Yang; CHOU, Chun-Hung. Predicting the Onset of Diabetes with Machine Learning Methods. **Journal of Personalized Medicine**, [s. l.], v. 13, n. 3, 24 fev. 2023. Disponível em: <https://doi.org/10.3390/jpm13030406>. Acesso em: 18 dez. 2024.

COHERE'S Command R+ Model: Details and Application). In: COHERE. **Cohere Docs**. V2. [S. l.], 2025. Disponível em: <https://docs.cohere.com/docs/command-r-plus>. Acesso em: 21 maio 2025.

CRESWELL, John W. **Research Design: Qualitative, Quantitative and Mixed Methods Approaches**. 4. ed. atual. Thousand Oaks: SAGE Publications, 2014. 273 p. ISBN 978-1452226101.

DAS, Abhik. Logistic Regression. In: MAGGINO, Filomena. **Encyclopedia of Quality of Life and Well-Being Research**. 2. ed. [S. l.]: Springer Cham, 2021. ISBN 978-3-319-69909-7. Disponível em: https://doi.org/10.1007/978-3-319-69909-7_1689-2. Acesso em: 20 ago. 2025.

DOANE, David P.; SEWARD, Lori E. Measuring Skewness: A Forgotten Statistic?. **Journal of Statistics Education**, [s. l.], v. 19, ed. 2, 2011. Disponível em: <https://doi.org/10.1080/10691898.2011.11889611>. Acesso em: 23 maio 2025.

ENFERMEIRA, Nilce Botto; ALIGIERI, Débora; ALMEIDA-PITITTO, Bianca; MELO, Karla FS; FORTI, Adriana Costa. Dispensação de medicamentos e insumos para o tratamento do diabetes mellitus no SUS. **Diretriz Oficial da Sociedade Brasileira de Diabetes**, [s. l.], 2024. Disponível em: <https://doi.org/10.29327/5412848.2024-2>. Acesso em: 13 fev. 2025.

FACELI, Katti; LORENA, Ana C.; GAMA, João; AL, *et al.* **Inteligência Artificial - Uma Abordagem de Aprendizado de Máquina**. 2. ed. Rio de Janeiro: LTC, 2021. *E-book*. p.4. ISBN 9788521637509. Disponível em: <https://app.minhabiblioteca.com.br/reader/books/9788521637509/>. Acesso em: 01 fev. 2025.

FEBRIAN, Muhammad Exell; FERDINAN, Fransiskus Xaverius; SENDANI, Gustian Paul; SURYANIGRUM, Kristien Margi; YUNANDA, Rezki. Diabetes prediction using supervised machine learning. **Procedia Computer Science**, [s. l.], v. 216, p. 21-30, 2023. Disponível em: <https://doi.org/10.1016/j.procs.2022.12.107>. Acesso em: 16 dez. 2024.

FLANAGAN, David. **JavaScript: O Guia Definitivo**. 6. ed. atual. São Paulo: Bookman, 2013. 1062 p. ISBN 978-8565837194.

GEORGE, Blesson. A Study of the Effect of Random Projection and other Dimensionality Reduction Techniques on Different Classification Methods. **Journal of Interdisciplinary**

Studies and Research, Kerala, ano 2017, v. 18, ed. 1, p. 69-75, 2017. Disponível em: <http://www.baseliuss.ac.in/wp-content/uploads/2021/01/Baseliuss-Journal-Jan-June-2017.pdf#page=68>. Acesso em: 30 maio 2025.

GÉRON, A. **Mãos à Obra: Aprendizado de Máquina com Scikit-Learn, Keras e TensorFlow**. 2. ed. atual. Rio de Janeiro: Alta Books, 2021. 614 p. ISBN 9788550815480.

GIL, Antonio C. **Métodos e Técnicas de Pesquisa Social**. 6. ed. atual. São Paulo: Atlas, 2008. 220 p. ISBN 978-85-224-5142-5.

GOTTFREDSON, Linda. Mainstream Science on Intelligence. **The Wall Street Journal**, Nova York, ano 1994, p. 8, 13 dez. 1994. Disponível em: <https://www1.udel.edu/educ/gottfredson/reprints/1994WSJmainstream.pdf>. Acesso em: 4 maio 2025.

GOLDBERG, Josh. **Aprendendo TypeScript: Melhore suas habilidades de desenvolvimento usando o JavaScript Type-Safe**. São Paulo: Novatec, 2022. 333 p. ISBN 978-8575228302.

GRINBERG, Miguel. **Flask Web Development: Developing Web Applications with Python**. 2. ed. atual. Sebastopol: O'Reilly Media, 2018. 312 p. ISBN 978-1491991732.

HO, Tin Kam. Random decision forests. **Proceedings of 3rd International Conference on Document Analysis and Recognition**, [S. l.], ano 1995, v. 2, p. 278-282, 1995. 3rd International Conference on Document Analysis and Recognition, 1995, Montreal.

HOSSAIN, Elias et al. Dm-Health App: Diabetes Diagnosis Using Machine Learning with Smartphone. **Computers, Materials & Continua**, [S. l.], v. 72, ed. 1, p. 1713-1746, 2022. Disponível em: <https://doi.org/10.32604/cmc.2022.024822>. Acesso em: 2 jan. 2025.

INZUCCHI, Silvio E. **Diabete Melito**. 6. ed. Porto Alegre: ArtMed, 2007. *E-book*. p.17. ISBN 9788536309743. Disponível em: <https://app.minhabiblioteca.com.br/reader/books/9788536309743/>. Acesso em: 10 fev. 2025.

KAHN, C R.; WEIR, Gordon C.; KING, George L.; et al. **Joslin: diabetes melito**. 14. ed. Porto Alegre: ArtMed, 2009. *E-book*. p.346. ISBN 9788536320304. Disponível em: <https://app.minhabiblioteca.com.br/reader/books/9788536320304/>. Acesso em: 10 fev. 2025.

KHANAM, Jobeda Jamal; FOO, Simon Y. A comparison of machine learning algorithms for diabetes prediction. **ICT Express**, Coreia do Sul, v. 7, ed. 4, p. 432-439, 20 fev. 2021. Disponível em: <https://doi.org/10.1016/j.ict.2021.02.004>. Acesso em: 11 dez. 2024.

LIMA, I. **Inteligência Artificial**. Rio de Janeiro: GEN LTC, 2014. *E-book*. p.1. ISBN 9788595152724. Disponível em: <https://app.minhabiblioteca.com.br/reader/books/9788595152724/>. Acesso em: 30 jan. 2025.

LUDERMIR, Teresa Bernarda. **Inteligência Artificial e Aprendizado de Máquina: estado atual e tendências**. Estudos Avançados, [s. l.], ed. 35, 2021. Disponível em: <https://doi.org/10.1590/s0103-4014.2021.35101.007>. Acesso em: 19 ago. 2025.

LUGER, G. F. **Inteligência artificial**. 6. ed. São Paulo: Pearson, 2013. *E-book*. Disponível em: <https://plataforma.bvirtual.com.br>. Acesso em: 14 fev. 2025.

LUTZ, Mark; ASCHER, David. **Aprendendo Python: Programação Orientada a Objetos**. 2. ed. atual. e aum. Porto Alegre: Bookman, 2007. 568 p. ISBN 978-8577800131.

MARVIN, Ggaliwango; HELLEN, Nakayiza; JJINGO, Daudi; NAKATUMBA-NABENDE, Joyce. **Prompt Engineering in Large Language Models. Data Intelligence and Cognitive Informatics**, [s. l.], p. 387–402, 7 jan. 2024. Disponível em: https://doi.org/10.1007/978-981-99-7962-2_30. Acesso em: 8 set. 2025.

MCKINNEY, Wes. **Python para Análise de Dados: Tratamento de Dados com pandas, NumPy & Jupyter**. 3. ed. atual. São Paulo: Novatec, 2023. 624 p. ISBN 978-8575228418.

MILECH, Adolpho; OLIVEIRA, José Egídio P.; ZAJDENVERG, Lenita; RODACKI, Melanie. **Rotinas de Diagnóstico e Tratamento do Diabetes Mellitus**. Rio de Janeiro: AC Farmacêutica, 2014. *E-book*. p.740. ISBN 978-85-8114-270-8. Disponível em: <https://app.minhabiblioteca.com.br/reader/books/978-85-8114-270-8/>. Acesso em: 13 fev. 2025.

MONTGOMERY, Douglas C. **Design and Analysis of Experiments**. 9. ed. Nova Jersey: Wiley, 2017. 752 p. ISBN 978-1119386100.

MÜLLER, A. C.; GUIDO, S. **Introduction to Machine Learning with Python: A guide for data scientists**. 1. ed. Sebastopol: O'Reilly, 2016. 376 p. ISBN 978144936941-5.

MURPHY, K. P. **Machine Learning: A Probabilistic Perspective**. 1. ed. Cambridge: MIT Press, 2012. 1104 p. ISBN 978-0262018029.

NARGESIAN, Fatemeh; SAMULOWITZ, Horst; KHURANA, Udayan; KHALIL, Elias B.; TURAGA, Deepak. Learning Feature Engineering for Classification. **Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence (IJCAI-17)**,

Melbourne, p. 2529-2535, 2017. Disponível em: <https://doi.org/10.24963/ijcai.2017/352>. Acesso em: 11 jul. 2025.

NATEKIN, Alexey; KNOLL, Alois. Gradient boosting machines, a tutorial. **METHODS article**, [s. l.], v. 7, 3 dez. 2013. Disponível em: <https://doi.org/10.3389/fnbot.2013.00021>. Acesso em: 23 ago. 2025.

OBE, Regina O.; HSU, Leo S. **PostgreSQL: Up and Running**. 2. ed. atual. Sebastopol: O'Reilly Media, 2014. 211 p. ISBN 978-1-449-37319-1

OLISAH, Chollette C.; SMITH, Lyndon; SMITH, Melvyn. Diabetes mellitus prediction and diagnosis from a data preprocessing and machine learning perspective. **Computer Methods and Programs in Biomedicine**, [s. l.], v. 220, 2022. Disponível em: <https://doi.org/10.1016/j.cmpb.2022.106773>. Acesso em: 12 dez. 2024.

OLIVEIRA, Eduardo Aparecido de; SOUZA, Vinicius Aparecido de; JUNGERS, Vinicius Cardoso; CODO, Fabio. Comparação de Desempenho entre os ORMs TypeORM, Prisma e Sequelize em Aplicações Node.js. **Revista Eletrônica e-Fatec**, [s. l.], v. 14, n. 2, 5 ago. 2024. Disponível em: <https://www.pesquisafatec.com.br/ojs/index.php/efatec/article/view/326>. Acesso em: 21 maio 2025.

PEDREGOSA, F.; VAROQUAUX, G.; GRAMFORT, A.; *et al.* *Scikit-learn: Machine learning in Python*. **Journal of Machine Learning Research**, v. 12, p. 2825-2830, 2011. Disponível em: <https://www.jmlr.org/papers/v12/pedregosa11a.html>. Acesso em: 15 mai. 2025.

POWERS, Shelley. **Aprendendo Node: Usando JavaScript no Servidor**. São Paulo: Novatec, 2017. 312 p. ISBN 978-8575225400.

RASHID, Ahlam. Diabetes Dataset. *In: Mendeley Data*. 1. [S. l.], 18 jul. 2020. Disponível em: <http://doi.org/10.17632/wj9rwkp9c2.1>. Acesso em: 23 maio 2025.

RUSSELL, Stuart J.; NORVIG, Peter. **Inteligência Artificial: Uma Abordagem Moderna**. 4. ed. Rio de Janeiro: GEN LTC, 2022. *E-book*. p.Capa. ISBN 9788595159495. Disponível em: <https://app.minhabiblioteca.com.br/reader/books/9788595159495/>. Acesso em: 29 jan. 2025.

SANTOS, V. de S. Comparison and selection of machine learning algorithms for diabetes prediction: An exploratory quantitative study based on medical data analysis. **Seven Editora**, [S. l.], p. 737-765, 2024. Disponível em: <https://sevenpublicacoes.com.br/editora/article/view/4102>. Acesso em: 31 jan. 2025.

SERRA, Henrique Oliveira; NASCIMENTO, Vinício da Silva. **Aprendizado de máquina para diagnóstico de diabetes mellitus**. 2022. Trabalho de Conclusão de Curso (Curso Superior de Tecnologia em Informática para Negócios) – Faculdade de Tecnologia de São José do Rio Preto, São José do Rio Preto, 2022.

SONI, Mitushi; VARMA, Dr. Sunita. Diabetes Prediction using Machine Learning Techniques. **International Journal of Engineering Research & Technology (IJERT)**, [s. l.], v. 9, ed. 9, p. 921-925, 2020. Disponível em: <https://www.ijert.org/research/diabetes-prediction-using-machine-learning-techniques-IJERT-V9IS090496.pdf>. Acesso em: 2 dez. 2024.

SOUZA, M. Diagnóstico precoce da diabetes pode evitar cegueira, amputações e infartos, dizem especialistas. **Agência Câmara de Notícias**, [S. l.], 19 nov. 2020. Disponível em: https://www.camara.leg.br/noticias/708896-diagnostico-precoce-da-diabetes-pode-evitar-cegueira-amputacoes-e-infartos-dizem-especialistas/?utm_source=chatgpt.com. Acesso em: 14 fev. 2025.

SUTTON, R. S.; BARTO, A. G. **Reinforcement Learning: An Introduction**. 2. ed. atual. Cambridge: Bradford Books, 2018. 552 p. ISBN 9780262039246.

VANDERPLAS, Jake. **Python Data Science Handbook: Essential Tools for Working with Data**. Sebastopol: O'Reilly Media, 2017. 529 p. ISBN 978-1-491-91205-8.

World Health Organization (WHO). Diabetes. [S. l.], 14 nov. 2024. Disponível em: <https://www.who.int/news-room/fact-sheets/detail/diabetes>. Acesso em: 16 jan. 2025.

WEBB, Geoffrey I. Naïve Bayes. In: SAMMUT, Claude; WEBB, Geoffrey I. **Encyclopedia of Machine Learning and Data Mining**. Boston: Springer, 2017. ISBN 978-1-4899-7687-1. Disponível em: https://doi.org/10.1007/978-1-4899-7687-1_581. Acesso em: 23 ago. 2025.

YEO, In-Kwon; JOHNSON, Richard A. A new family of power transformations to improve normality or symmetry. **Biometrika**, [s. l.], ano 2000, v. 87, ed. 4, p. 954–959, 1 dez. 2000. Disponível em: <https://doi.org/10.1093/biomet/87.4.954>. Acesso em: 23 maio 2025.

ZERBO, Ali; OUEDRAOGO, Moïse; SERE, Abdoulaye; DIARRA, Mamadou. Comparison of Joblib and Pymmp for Parallel Fingerprint Recognition. **Proceedings of the 6th Computer Science Research Days**, Ouagadougou, 2024. JRI 2023, 18-20 December 2023, Ouagadougou, Burkina Faso.

ZHOU, Bin *et al.* Worldwide trends in diabetes prevalence and treatment from 1990 to 2022: a pooled analysis of 1108 population-representative studies with 141 million participants. **The**

Lancet, [s. l.], v. 404, ed. 10467, p. 2077-2093, 2024. Disponível em: [https://doi.org/10.1016/S0140-6736\(24\)02317-1](https://doi.org/10.1016/S0140-6736(24)02317-1). Acesso em: 13 fev. 2025.

ZHOU, Zhi-Hua. **Machine Learning**. 1. ed. Singapura: Springer, 2021. 459 p. ISBN 9789811519666.

ZOU, Q.; QU, K.; LUO, Y.; YIN, D.; JU, Y.; TANG, H. Predicting Diabetes Mellitus With Machine Learning Techniques. **Frontiers in Genetics**, [s. l.], v. 9, n. 515, 2018. Disponível em: <https://doi.org/10.3389/fgene.2018.00515>. Acesso em: 14 fev. 2025.

APÊNDICE A - HIPERPARÂMETROS CONSIDERADOS PARA OS MODELOS

<i>K-Nearest Neighbors (KNN)</i>	
<i>n_neighbors</i> (k)	Número de vizinhos mais próximos a considerar para classificação ou regressão. Valores pequenos (ex.: 1, 3): modelo mais sensível a ruído, maior variância. Valores grandes (ex.: 15, 20): modelo mais suave, menor variância, mas maior viés.
<i>metric</i>	Métrica de distância usada para calcular a proximidade. Opções comuns: <i>minkowski</i> , <i>euclidean</i> , <i>manhattan</i> .
<i>Support Vector Machine (SVM)</i>	
C (parâmetro de regularização)	Controla o trade-off entre margem larga e erros de classificação. C alto: penaliza mais os erros, margem menor, modelo mais complexo (baixo viés, alta variância). C baixo: penaliza menos os erros, margem maior, modelo mais simples (alto viés, baixa variância).
<i>kernel</i>	Função usada para transformar os dados para espaço de alta dimensão. Opções principais: <i>linear</i> , <i>poly</i> , <i>rbf</i> , <i>sigmoid</i> .
<i>gamma</i>	Define a influência de um único ponto de treino. Gamma alto: cada ponto tem influência alta, modelo muito complexo (<i>overfitting</i>). Gamma baixo: cada ponto tem influência mais suave, modelo mais simples (<i>underfitting</i>).
<i>Logistic Regression (LR)</i>	
C (força da regularização)	Inverso da força de regularização. C alto: menos regularização, modelo mais complexo. C baixo: mais regularização, modelo mais simples.
<i>penalty</i>	Define a forma da penalização para evitar <i>overfitting</i> . Opções: L2, Regularização <i>Ridge</i> , L1, <i>elasticnet</i> , <i>none</i> .
<i>solver</i>	Método para encontrar os coeficientes. Opções comuns: <i>lbfgs</i> , <i>liblinear</i> , <i>saga</i> , <i>newton-cg</i> .
<i>max_iter</i>	Número máximo de iterações para convergência. Se o modelo não converge, aumentar esse valor.
<i>Gradient Boosting (GB)</i>	
<i>n_estimators</i>	Número de árvores (iterações do <i>boosting</i>). Valor alto: modelo mais complexo, risco de <i>overfitting</i> . Valor baixo: modelo simples, risco de <i>underfitting</i> .
<i>learning_rate</i>	Taxa de aprendizado, controla o peso de cada árvore adicionada. Menor taxa: treinamento mais lento, mas mais robusto (precisa aumentar <i>n_estimators</i>). Maior taxa: treinamento mais rápido, risco de <i>overfitting</i> .
<i>max_depth</i>	Profundidade máxima das árvores. Profundas (alto valor): captam relações complexas, mas podem <i>overfit</i> . Rasas (baixo valor): generalizam melhor, mas podem <i>underfit</i> .
<i>Decision Tree (DT)</i>	

<i>max_depth</i>	Profundidade máxima da árvore. Não definido (<i>None</i>), cresce até não poder mais, risco de <i>overfitting</i> . Valor pequeno, generaliza melhor (menos <i>overfitting</i>).
<i>min_samples_split</i>	Número mínimo de amostras para dividir um nó. Valor maior, árvores menos profundas (reduz <i>overfitting</i>).
<i>min_samples_leaf</i>	Número mínimo de amostras em uma folha. Valor maior, folhas com mais amostras e modelo mais estável.
<i>criterion</i>	Função usada para medir a qualidade da divisão. Opções (para classificação): <i>gini</i> , <i>entropy</i> , <i>log_loss</i> .
<i>Random Forest (RF)</i>	
<i>n_estimators</i>	Número de árvores na floresta. Mais árvores, modelo mais estável, menor variância, maior custo computacional.
<i>max_depth</i>	Profundidade máxima da árvore. Não definido (<i>None</i>), cresce até não poder mais, risco de <i>overfitting</i> . Valor pequeno, generaliza melhor (menos <i>overfitting</i>).
<i>min_samples_split</i>	Número mínimo de amostras para dividir um nó. Valor maior, árvores menos profundas (reduz <i>overfitting</i>).
<i>min_samples_leaf</i>	Número mínimo de amostras em uma folha. Valor maior, folhas com mais amostras e modelo mais estável.
<i>Naive Bayes (NB)</i>	
<i>var_smoothing</i>	Adiciona um valor pequeno à variância de cada <i>feature</i> para evitar divisão por zero ou valores extremamente pequenos.

APÊNDICE B - MÉTRICAS CONSIDERADAS PARA AVALIAÇÃO DOS MODELOS

Métrica		Representação Matemática ⁴⁴
Acurácia	Proporção de acertos entre todas as previsões.	$A = \frac{TP + TN}{TP + TN + FP + FN}$
Precisão	Percentual de previsões positivas que realmente são positivas.	$P = \frac{TP}{TP + FP}$
<i>Recall</i>	Percentual de casos positivos corretos identificados.	$R = \frac{TP}{TP + FN}$
F1-Score	Média harmônica entre Precisão e <i>Recall</i> .	$F1 = 2 \times \frac{P \times R}{P + R}$

⁴⁴ Considere: TP = Verdadeiro Positivo; TN = Verdadeiro Negativo; FP = Falso Positivo; FN = Falso Negativo.

APÊNDICE C - REGRAS DE NEGÓCIOS DA API DIAGLY

Identificador	Descrição
RN001	Todos os usuários (Administrador, Médico, Paciente) devem estar autenticados para acessar suas funcionalidades.
RN002	O login é feito via credenciais, via <i>JWT Token</i> .
RN003	Somente Administradores podem cadastrar, atualizar, listar e excluir outros administradores.
RN004	Somente Administradores podem cadastrar, listar e excluir médicos.
RN005	Somente Médicos podem cadastrar pacientes e manipular dados médicos (criar, atualizar, deletar, visualizar).
RN006	Pacientes só podem visualizar e atualizar os próprios dados.
RN007	Dados médicos só podem ser criados/atualizados/excluídos por médicos vinculados ao paciente.
RN008	A geração de predição depende do envio prévio de dados médicos.
RN009	Conselhos personalizados só são gerados após a predição
RN010	Alterações (atualizações e exclusões) devem manter rastreabilidade para auditoria.

APÊNDICE D - REQUISITOS DA API DIAGLY

Identificador	Descrição
Requisitos Funcionais	
RF001	O sistema deve permitir que Administradores, Médicos e Pacientes façam login.
RF002	O Administrador deve cadastrar novos administradores.
RF003	O Administrador deve atualizar informações de administradores.
RF004	O Administrador deve listar administradores.
RF005	O Administrador deve visualizar administrador por ID.
RF006	O Administrador deve excluir administradores.
RF007	O Administrador deve cadastrar médicos.
RF008	O Administrador deve listar médicos.
RF009	O Administrador deve excluir médicos.
RF010	O Administrador deve visualizar informações de médicos.
RF011	O Médico deve atualizar informações próprias.
RF012	O Médico deve visualizar informações próprias.
RF013	O Médico deve cadastrar pacientes.
RF014	O Médico deve listar pacientes.
RF015	O Médico deve visualizar dados médicos por ID
RF016	O Médico deve criar dados médicos de pacientes.
RF017	O Médico deve atualizar dados médicos.
RF018	O Médico deve excluir dados médicos.
RF019	O Paciente deve atualizar informações pessoais.
RF020	O Paciente deve visualizar informações pessoais.
RF021	O Paciente deve excluir a própria conta.
RF022	O sistema deve receber dados médicos para predição.
RF023	O sistema deve gerar predição com base nos dados enviados.
RF024	O sistema deve gerar conselhos personalizados após a predição.
Requisitos Não Funcionais	
RNF001	O sistema deve utilizar autenticação via JWT.

RNF002	Todas as rotas protegidas devem validar o <i>token</i> antes de conceder acesso.
RNF003	Dados sensíveis (senhas, tokens) devem ser armazenados com criptografia/ <i>Hash</i> .
RNF004	O sistema deve responder às requisições em até 2 segundos em condições normais de uso.
RNF005	A geração de predições deve ser processada em tempo hábil (máx. 5 segundos).
RNF006	Mensagens de erro devem ser claras e orientar o usuário sobre como proceder.
RNF007	O sistema deve suportar crescimento de usuários e pacientes sem perda significativa de desempenho.

APÊNDICE E - PROMPT ENVIADO PARA O LLM

You are a health and lifestyle specialist. Your task is to provide a clear, non-technical explanation of the patient's diabetes diagnosis based on the provided data, followed by one simple, practical tip for managing their health.

****Important Guidelines:****

- Do NOT reinterpret the diagnosis. Use the prediction as ground truth.
- Focus your response on:
 1. Explaining why the diagnosis makes sense based on the patient's lab values.
 2. Highlighting which values are abnormal and how they relate to diabetes.
 3. Ending with a short, simple lifestyle tip (1–2 sentences max).

****Output Format:****

- First: clear explanation of diagnosis.
- Then: one short actionable tip.

****Example 1:****

Diagnosis: Diabetic

Confidence: 93.2%

HbA1c: 8.2 | Cholesterol: 240 | BMI: 32

Explanation: The HbA1c is well above the normal threshold (usually <5.7), indicating chronic high blood sugar levels. Combined with high cholesterol and a BMI in the obese range, these values support the diabetic diagnosis.

Tip: Try taking a 20-minute walk after meals to help regulate blood sugar.

****Example 2:****

Diagnosis: Non-Diabetic

Confidence: 91.0%

HbA1c: 5.1 | BMI: 22

Explanation: The HbA1c level is within the normal range, and the BMI is healthy. These values do not indicate current diabetic risk.

Tip: Keep up your balanced routine and stay active daily.

****Patient Information:****

- Name: \${data.patientName}
- Age: \${data.age} years
- Gender: \${data.gender}

****Medical Data:****

- HbA1c: \${data.medicalData.hba1c}
- Cholesterol: \${data.medicalData.chol}
- Triglycerides: \${data.medicalData.tg}
- BMI: \${data.medicalData.bmi}

****Confirmed Diagnosis:**** \${data.prediction.result} (Confidence: \${data.prediction.confidenceScore} * 100}%).

Now explain the diagnosis and end with one brief tip.

max_tokens: 500,
temperature: 0.7,

APÊNDICE F - TEMPOS DE RESPOSTA DOS MÓDULO DA API DIAGLY

Tipo de Requisição	Descrição	Tempo Mínimo	Tempo Máximo	Tempo Médio
Módulo Administrador				
POST	Criar Administrador	≈ 267 ms	≈ 326 ms	≈ 284,4 ms
GET	Listar todos os Administradores	≈ 55 ms	≈ 91 ms	≈ 66,9 ms
GET	Retornar um Administrador	≈ 98 ms	≈ 129 ms	≈ 110,9 ms
PUT	Atualizar um Administrador	≈ 340 ms	≈ 415 ms	≈ 372,5 ms
DELETE	Excluir um Administrador	≈ 178 ms	≈ 214 ms	≈ 187,1 ms
POST	Login do Administrador	≈ 135 ms	≈ 220 ms	≈ 154,3 ms
Módulo Médico				
POST	Criar Médico	≈ 290 ms	≈ 359 ms	≈ 321,9 ms
GET	Listar todos os Médicos	≈ 55 ms	≈ 95 ms	≈ 66,2 ms
GET	Retornar um Médico	≈ 94 ms	≈ 132 ms	≈ 109,5 ms
PUT	Atualizar um Médico	≈ 325 ms	≈ 412 ms	≈ 369,2 ms
DELETE	Excluir um Médico	≈ 181 ms	≈ 219 ms	≈ 183,4 ms
POST	Login do Médico	≈ 139 ms	≈ 190 ms	≈ 163,4 ms
Módulo Paciente				
POST	Criar Paciente	≈ 312 ms	≈ 412 ms	≈ 363,2 ms
GET	Listar todos os Pacientes	≈ 55 ms	≈ 95 ms	≈ 61,4 ms
GET	Retornar um Paciente	≈ 98 ms	≈ 129 ms	≈ 111,3 ms
PUT	Atualizar um Paciente	≈ 345 ms	≈ 422 ms	≈ 383,4 ms
DELETE	Excluir um Paciente	≈ 189 ms	≈ 232 ms	≈ 193,2 ms
POST	Login do Paciente	≈ 135 ms	≈ 213 ms	≈ 146,7 ms
Módulo Dados Médicos				
POST	Criar Dado Médico	≈ 323 ms	≈ 417 ms	≈ 378,4 ms
GET	Retornar um Dado Médico	≈ 55 ms	≈ 87 ms	≈ 65,8 ms
PUT	Atualizar um Dado Médico	≈ 386 ms	≈ 446ms	≈ 341,6 ms
DELETE	Excluir um Dado Médico	≈ 148 ms	≈ 224ms	≈ 187,2 ms
Módulo Predição (SSML)				

POST	Gerar predição	≈ 286 ms	≈ 311 ms	$\approx 300,4$ ms
Módulo Conselho (AMIG)				
GET	Retornar um Conselho	$\approx 1,9$ s	$\approx 4,8$ ms	$\approx 2,5$ s