I-AutoML: Uma Abordagem Iterativa e Automatizada para Classificação de Dados Usando um Auto Aprendiz

José E. S. Júnior
Departamento de Informática e Matemática Aplicada
Universidade Federal do Rio Grande do Norte
Natal, Brasil
jose.edivandro.104@ufrn.edu.br

Anne M. P. Canuto
Departamento de Informática e Matemática Aplicada
Universidade Federal do Rio Grande do Norte
Natal, Brasil
anne.canuto@ufrn.br

***Abstract*—The growing demand for machine learning (ML) solutions across diverse domains has driven the development of tools that automate complex modeling workflows. This paper introduces "I-AutoM", an interactive and automated system for supervised classification tasks. Built using Python and Streamlit, the system streamlines the entire ML pipeline—from data preprocessing and model training to evaluation and visualization—without requiring user intervention. It integrates eleven popular classifiers, including Random Forest, XGBoost, LightGBM, and CatBoost, and evaluates their performance using five metrics: accuracy, precision, recall, F1-score, and ROC-AUC. Tested on ten real-world datasets from Kaggle, the tool demonstrated its ability to adapt to different data structures, offering accessible, explainable, and efficient model comparison. Results highlight how model performance varies across datasets and emphasize the tool's value for educational, exploratory, and production environments.**

***Index Terms*—component, formatting, style, styling, insert**

I. Introdução

A crescente demanda por soluções de aprendizado de máquina (ML) em diversos setores da indústria e também no âmbito acadêmico tem evidenciado a necessidade de ferramentas que simplifiquem e automatizem o processo de desenvolvimento de modelos preditivos. Tradicionalmente, a construção de modelos de ML envolve etapas complexas, como pré-processamento de dados, seleção de algoritmos, ajuste de hiperparâmetros e validação de desempenho, exigindo conhecimento técnico especializado e demandando um tempo considerável para a maturação da implementação.

O I-AutoML"Auto Aprendiz" surge como uma proposta para democratizar o acesso ao ML, oferecendo uma plataforma que automatiza integralmente o pipeline de classificação de dados. Desenvolvido em Python, com interface interativa via Streamlit, o sistema integra técnicas de pré-processamento, seleção e avaliação de modelos, proporcionando uma experiência acessível tanto para especialistas quanto para usuários com menor familiaridade técnica.

Este artigo apresenta a arquitetura e funcionalidades do I-AutoML, destacando sua capacidade de adaptar-se a diferentes conjuntos de dados e oferecer recomendações de modelos baseadas em métricas de desempenho. A proposta visa contribuir para a ampliação do uso de ML em contextos diversos, reduzindo barreiras técnicas e promovendo a eficiência no desenvolvimento de soluções baseadas em dados. Para tal, os seguintes objetivos específicos são definidos:

- Padronizar o pré-processamento de dados heterogêneos por meio de um pipeline automático;
- Integrar diferentes algoritmos de classificação supervisionada;
- Implementar uma interface gráfica com visualização dos resultados;
- Avaliar os modelos com base em múltiplas métricas;
- Permitir sugestões de modelagem com auxílio de modelos de linguagem (LLMs).

Este artigo está organizado em sete seções e dividido da seguinte forma: A Seção 2 descreve o referencial teórico relacionado ao artigo, assim como os principais trabalhos relacionados ao tema. A Seção 3 apresenta a descrição do I-AutoML, enquanto a Seção 4 descreve a metodologia utilizada na análise empírica. Os resultados obtidos estão apresentados na Seção 5 e discutidos com mais detalhes na Seção 6. Finalmente, a Seção 7 apresenta as considerações finais do artigo.

II. Referencial teórico e trabalhos relacionados

A. Referencial teórico

O campo de *Automated Machine Learning* (AutoML) surgiu com o objetivo de automatizar etapas essenciais do processo de construção de modelos preditivos, como a seleção de algoritmos, pré-processamento dos dados, extração de atributos, ajuste de hiperparâmetros e validação cruzada [7]. Essa abordagem busca reduzir a necessidade de expertise técnica, tornando o uso de aprendizado de máquina mais acessível para analistas, profissionais de negócios e pesquisadores não especializados. Hutter, Kotthoff e Vanschoren [7] formalizaram um dos problemas centrais do AutoML como o CASH (Combined Algorithm Selection and Hyperparameter Optimization), no qual o sistema busca simultaneamente o melhor

algoritmo e a melhor configuração de parâmetros para os dados disponíveis. Soluções como Auto-sklearn, Auto-WEKA e TPOT baseiam-se nesse princípio, adotando estratégias como meta-aprendizado, validação cruzada e algoritmos genéticos para otimização. O Auto-sklearn, por exemplo, combina histórico de desempenho em outras tarefas com otimização bayesiana, criando pipelines automáticos e ensembles de classificadores com bom desempenho em benchmarks [5], [6]. Já o TPOT utiliza programação genética para construir sequências otimizadas de transformações e modelos, gerando soluções interpretáveis, mas computacionalmente mais pesadas.

Entretanto, diversos estudos apontam que o desempenho de sistemas AutoML varia de acordo com características do conjunto de dados, como desbalanceamento entre classes, presença de ruído, dados ausentes ou variáveis categóricas complexas [1], [15]. Em tarefas específicas, como séries temporais ou dados textuais, ferramentas generalistas podem não alcançar bons resultados, como observado por Sun et al. [13]. Isso torna desejável a existência de abordagens mais controláveis, como oI-AutoM, que automatiza os principais passos, mas mantém o usuário informado e no controle do processo de avaliação.

No I-AutoML, são avaliados modelos com diferentes naturezas e estratégias. O *Random Forest* [2] é amplamente usado por sua robustez, combinando várias árvores de decisão com amostragem aleatória e votação majoritária. Ele tem bom desempenho mesmo em bases desbalanceadas e ruidosas [3]. O XGBoost, criado por [4], é uma das variantes mais otimizadas do Gradient Boosting e ficou popular por seu sucesso em competições do Kaggle. Ele realiza adições sequenciais de árvores que corrigem erros anteriores e inclui regularização para evitar overfitting [8]. O LightGBM, por sua vez, é uma alternativa mais eficiente para grandes bases de dados, com tempo de treinamento reduzido, uso de histogramas e divisão das folhas em ordem baseada em ganho de informação [9]. O CatBoost, criado pela Yandex, é especializado no tratamento de variáveis categóricas e é eficaz em bases desbalanceadas, dispensando transformações manuais nos dados [10], [11].

O Gradient Boosting tradicional, base de todos os anteriores, é um método aditivo que minimiza os erros sequencialmente, com forte capacidade preditiva, mas alta sensibilidade aos hiperparâmetros. Outros modelos também são incluídos para representar técnicas clássicas. A Decision Tree é um classificador intuitivo e interpretável, mas propenso a overfitting quando não regularizada. O SVM (Support Vector Machine) busca encontrar hiperplanos que separem as classes com a maior margem possível, sendo eficiente em conjuntos com alta dimensionalidade, embora menos adequado para grandes volumes de dados ruidosos [14]. A Regressão Logística, mesmo sendo um modelo linear simples, é amplamente utilizada em tarefas de classificação binária e serve como referência de base. O MLPClassifier é uma rede neural multicamada capaz de capturar relações não lineares entre variáveis, embora demande mais dados e tempo de treinamento. Já o K-Nearest Neighbors classifica com base na proximidade entre instâncias, sendo sensível à escala dos dados. Por fim, o Naive Bayes, baseado em teorema de Bayes com suposição de independência entre atributos, é eficaz mesmo com pouco dado rotulado, desde que as distribuições se comportem de forma semelhante [10], [12].

A presença dessa diversidade de modelos no I-AutoM é estratégica. Como cada conjunto de dados apresenta padrões distintos, a comparação entre algoritmos com diferentes princípios permite identificar qual abordagem é mais adequada para o cenário em questão. Essa lógica está em linha com os estudos de benchmarking de AutoML [1], [6], que mostram que não existe um único modelo superior universalmente, reforçando a importância de comparações orientadas por múltiplas métricas

B. Trabalhos relacionados

Além das abordagens discutidas, é importante destacar algumas ferramentas consolidadas no ecossistema AutoML, como Auto-sklearn, TPOT, H2O AutoML e Google AutoML Tables. O Auto-sklearn é amplamente utilizado por sua eficiência na busca por hiperparâmetros e uso de meta-aprendizado, porém apresenta limitações em termos de interpretabilidade e controle pelo usuário final [5]. O TPOT, por sua vez, oferece geração automática de pipelines por meio de programação genética, mas seu custo computacional elevado pode inviabilizar sua aplicação em ambientes com recursos limitados [6]. Já o H2O AutoML combina múltiplos algoritmos em uma estrutura de stacking com validação cruzada, obtendo bons resultados mesmo sem ajuste manual, mas exige familiaridade com seu ecossistema proprietário. O Google AutoML Tables, embora forneça uma solução altamente automatizada e com bons índices de acerto, depende da infraestrutura da nuvem e do pagamento por uso, o que pode ser um entrave para aplicações acadêmicas ou institucionais de pequeno porte. Nesse cenário, o I-AutoML se destaca por ser uma ferramenta leve, gratuita, interpretável e de fácil implantação local. Sua proposta didática e interativa visa preencher a lacuna entre sistemas altamente automatizados e ambientes de aprendizado, proporcionando uma experiência acessível, transparente e ajustável para tarefas de classificação supervisionada.

III. I-AutoML - Uma ferramenta iterativa de AutoML

A ferramenta desenvolvida, chamada I-AutoML, tem como objetivo facilitar o processo de classificação de dados supervisionados por meio da automação das principais etapas de um pipeline de aprendizado de máquina. Sua proposta é oferecer uma interface interativa que guie o usuário na construção de modelos de forma iterativa, permitindo ajustes e visualizações em tempo real dos resultados.

A I-AutoML foi pensada para usuários com diferentes níveis de experiência em ciência de dados, possibilitando desde análises rápidas até testes mais elaborados de modelos preditivos. A Figura 1 ilustra o fluxo geral de funcionamento da ferramenta, desde o carregamento dos dados até a apresentação dos resultados finais, com destaque para os momentos de interação com o usuário.

Fig. 1. Fluxo de utilização do I-AutoML.

A. Fluxo da ferramenta

O sistema segue um fluxo estruturado, composto pelas seguintes etapas:

1) Carregamento dos dados: o usuário envia um arquivo .csv contendo a base de dados.
2) Seleção da variável alvo: a aplicação exibe as colunas disponíveis e sugere como alvo apenas aquelas com mais de uma classe e variabilidade suficiente.
3) Pré-processamento automatizado: inclui:
   - exclusão de valores nulos na variável alvo;
   - codificação de variáveis categóricas com One-Hot Encoding;
   - imputação de valores numéricos ausentes pela média;
   - padronização com StandardScaler;
   - codificação do alvo com LabelEncoder.
4) Treinamento dos modelos: onze classificadores são treinados:
   - Regressão Logística, Árvore de Decisão, Florestas Aleatórias, Gradient Boosting, XGBoost, LightGBM, CatBoost, k-NN, SVM, Naive Bayes e MLPClassifier.
5) Divisão treino-teste: os dados são divididos entre treinamento e testes, onde na interface do I-AutoM poderá variar entre 10% até 50%
6) Avaliação dos modelos: são calculadas cinco métricas: Acurácia, Precisão, Recall, F1-score e ROC-AUC, de forma automatizada para todos os modelos.
7) Visualização e comparação: os resultados são apresentados em uma tabela interativa e gráficos de barras, com ordenação por métrica selecionada.
8) Sugestão com LLM: opcionalmente, o usuário pode fornecer uma chave da API OpenAI e receber sugestões de modelos adequados via GPT-3.5, com base nas características da base.

B. Ferramentas utilizadas

As seguintes bibliotecas foram empregadas:

- Scikit-learn: para pré-processamento, pipelines, classificadores e métricas (Pedregosa et al., 2011);
- Pandas e NumPy: para manipulação de dados;
- Plotly: para visualização gráfica interativa;
- Streamlit: para interface de usuário (Nápoles-Duarte et al., 2022);
- XGBoost, LightGBM e CatBoost: como algoritmos de boosting;
- OpenAI API: para integração com modelo de linguagem GPT.

A implementação completa foi feita em Python 3.10 e validada localmente com execução via streamlit

IV. Metodologia Experimental

Para avaliar a eficácia do I-AutoM, foi conduzido um conjunto de experimentos utilizando dez bases de dados reais, selecionadas da plataforma Kaggle (www.kaggle.com). A diversidade dessas bases teve como objetivo testar o sistema em cenários próximos da prática, abrangendo distintos domínios, estruturas e características estatísticas. A validação foi planejada para verificar o comportamento do sistema em tarefas de classificação com diferentes níveis de complexidade e distribuição das classes.

A. Critérios de seleção dos dados

As bases escolhidas pertencem a três grandes categorias temáticas, refletindo contextos em que tarefas de classificação supervisionada são frequentemente aplicadas:

- Cidades e dados demográficos: Incluem atributos sobre população, infraestrutura, localização geográfica, indicadores socioeconômicos, mobilidade urbana, entre outros. Essas bases geralmente apresentam variáveis mistas (numéricas e categóricas), o que representa um bom teste para a robustez do pipeline de pré-processamento automático.
- Negócios e produtos: Relacionadas a empresas, cadastros de clientes, produtos de e-commerce e comportamento de consumo. Nessas bases, a tarefa de classificação envolve prever, por exemplo, categorias de clientes, faixa de faturamento ou decisão de compra. São cenários típicos de uso empresarial da ciência de dados.
- Ações jurídicas e administrativas: Conjuntos com registros de decisões, manifestações ou documentos jurídicos e administrativos. Além da variabilidade estrutural, essas bases frequentemente apresentam variáveis desbalanceadas, como decisões majoritariamente favoráveis ou contrárias, exigindo cuidado na avaliação por métricas além da acurácia.

A seleção foi feita de forma a incluir bases balanceadas e desbalanceadas, permitindo avaliar o desempenho do sistema em contextos com diferentes proporções de classes. Em alguns

casos, como decisões binárias jurídicas ou categorias raras em setores comerciais, as classes minoritárias representavam menos de 10

B. Configuração dos testes

Os testes foram executados em ambiente local, com Python 3.10, Streamlit 1.32, e pacotes atualizados das bibliotecas utilizadas. Cada base foi submetida ao I-AutoM sem qualquer ajuste manual nos dados. O processo seguiu a lógica proposta pelo sistema:

- O arquivo .csv era carregado diretamente no sistema;
- O usuário selecionava a variável alvo proposta automaticamente (com mais de uma classe e proporção aceitável);
- As etapas de pré-processamento, modelagem e avaliação eram executadas de forma automatizada, conforme descrito na metodologia.

C. Observações sobre a execução

Durante os testes, o I-AutoM demonstrou boa adaptabilidade a formatos variados de dados. Bases com muitas variáveis categóricas foram corretamente tratadas pelo pipeline automático com One-Hot Encoding, enquanto valores numéricos ausentes foram imputados pela média de forma eficaz. Bases com grande número de colunas foram processadas sem erros, e em alguns casos, foi possível identificar rapidamente a presença de colunas pouco informativas (como IDs ou códigos únicos) por meio da visualização dos resultados dos modelos.

Além disso, os gráficos interativos e as tabelas ordenáveis contribuíram para a análise qualitativa dos resultados, facilitando a comparação entre modelos em diferentes cenários. Em bases desbalanceadas, pôde-se observar como métricas como F1-score e ROC-AUC forneciam diagnósticos mais precisos do que a acurácia simples, o que está em conformidade com a literatura sobre avaliação de classificadores [1]

As execuções também serviram para validar a capacidade do sistema de lidar com conjuntos de dados reais sem exigir ajustes técnicos por parte do usuário, reforçando seu caráter educativo e exploratório.

V. Resultados

Os experimentos conduzidos com o sistema I-AutoM utilizaram 10 conjuntos de dados obtidos de fontes públicas, com foco inicial em domínios como cidades, negócios e processos jurídicos. A ferramenta foi configurada para treinar, testar e comparar automaticamente os principais classificadores de machine learning disponíveis na biblioteca scikit-learn e extensões modernas, avaliando seu desempenho com base em cinco métricas: acurácia, precisão, recall, F1-score e ROC-AUC (curva Característica operacional do receptor).

Os gráficos a seguir representam os resultados obtidos em um dos conjuntos de dados avaliados, mais especificamente o conjunto demonimado ”Crime data”, este dataset contém registros detalhados sobre incidentes criminais, com informações sobre os infratores e vítimas. As variáveis incluem dados sobre a natureza dos crimes, localizações, e categorias de crimes, com o objetivo de realizar tarefas de classificação preditiva.” ilustrando a capacidade da ferramenta de destacar diferenças significativas entre os modelos em diferentes métricas:

A. Acurácia

A Figura 2 mostra que os modelos Random Forest, XGBoost e LightGBM apresentaram os melhores desempenhos em termos de acurácia, com valores acima de 77%. O pior desempenho foi do Naive Bayes, com apenas 25.68%, destacando sua limitação frente aos demais modelos para essa base.

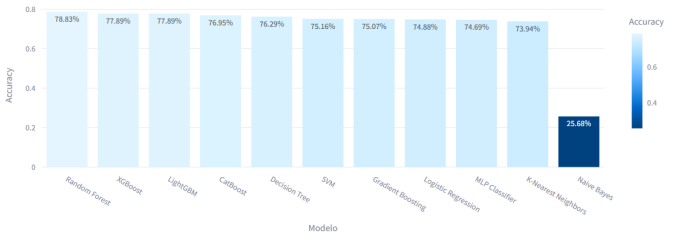

Fig. 2. Desempenho dos modelos em termos de acurácia para uma das bases avaliadas.

B. Precisão

A Figura 2 mostra a métrica de precisão. O Random Forest obteve o melhor valor (77.61%), enquanto o Logistic Regression teve o pior desempenho, com 67.41%. É interessante notar que o Naive Bayes, embora tenha ido mal em acurácia, ficou em terceiro lugar em precisão (74.54%), o que indica que seus acertos foram mais específicos, ainda que escassos.

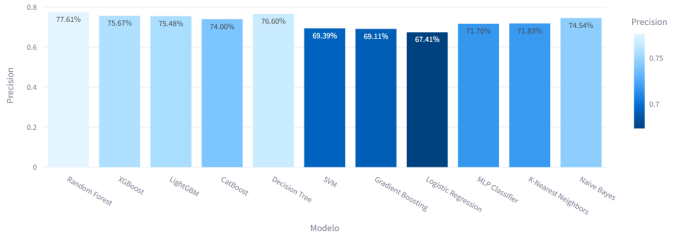

Fig. 3. Desempenho dos modelos em termos de precisão para uma das bases avaliadas.

.

C. Recall

Na Figura 3, observamos que Random Forest, XGBoost e LightGBM lideram novamente, com recall de 77.89% e 78.83%. Já o Naive Bayes ficou em último lugar, com recall de 25.68%, reforçando sua instabilidade nesta base.

.

D. F1-Score

A Figura 4 apresenta os valores de F1-score, uma métrica importante para conjuntos desbalanceados. O melhor modelo foi o Random Forest (77.99%), seguido por Decision Tree. O pior desempenho novamente foi do Naive Bayes, com apenas 11.78%, indicando falta de equilíbrio entre precisão e recall.

.

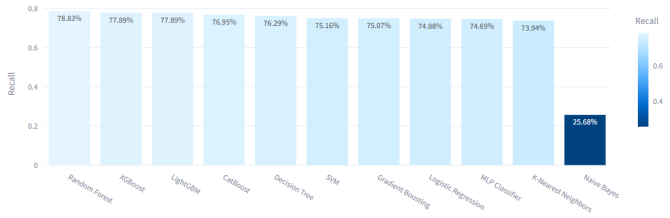

Fig. 4. Desempenho dos modelos em termos de recall para uma das bases avaliadas.

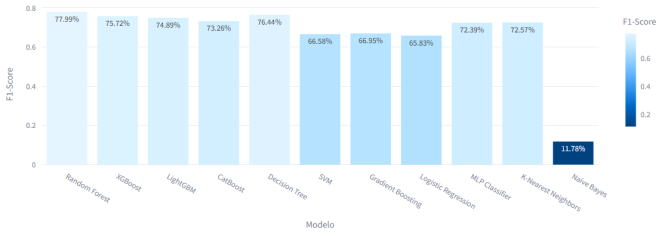

Fig. 5. Desempenho dos modelos em termos de F1-score para uma das bases avaliadas.

E. ROC-AUC

A Figura 5 mostra que o Decision Tree obteve a maior pontuação em ROC-AUC (68.89%), seguido de perto por Random Forest. O pior desempenho foi do Logistic Regression (50.95%), valor próximo ao aleatório, sugerindo que esse modelo não foi eficiente em discriminar entre as classes.

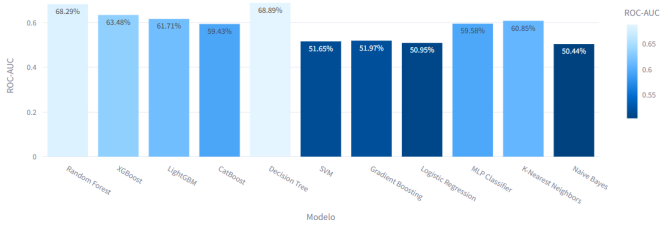

Fig. 6. Desempenho dos modelos em termos de ROC-AUC para uma das bases avaliadas.

.

F. Resumo das diretrizes

Por fim, o I-AutoML gera uma tabela resumo do desempenho de todos os modelos avaliados, considerando as cinco métricas mais comuns de avaliação preditiva: acurácia, precisão, recall, F1-score e ROC-AUC. Observa-se que o modelo XGBoost apresentou o melhor equilíbrio geral entre as métricas, seguido de perto pelo LightGBM. Já o modelo Naive Bayes obteve o pior desempenho em quase todos os indicadores, especialmente no F1-score, onde atingiu apenas 11.78%. Essa visão tabular permite comparar rapidamente o desempenho dos modelos, sendo particularmente útil quando o critério de escolha não se resume à acurácia, mas sim à robustez em diferentes métricas.

Modelo	Accuracy	Precision	Recall	F1-Score	ROC-AUC
XGBoost	77.89%	75.67%	77.89%	75.72%	63.48%
LightGBM	77.89%	75.48%	77.89%	74.89%	61.71%
CatBoost	76.95%	74.00%	76.95%	73.26%	59.43%
Decision Tree	76.29%	76.60%	76.29%	76.44%	68.89%
SVM	75.16%	69.39%	75.16%	66.58%	51.65%
Gradient Boosting	75.07%	69.11%	75.07%	66.95%	51.97%
Logistic Regression	74.88%	67.41%	74.88%	65.83%	50.95%
MLP Classifier	74.69%	71.70%	74.69%	72.39%	59.58%
K-Nearest Neighbors	73.94%	71.83%	73.94%	72.57%	60.85%
Naive Bayes	25.68%	74.54%	25.68%	11.78%	50.44%

Fig. 7. Resumo dos resultados dos modelos utilizados pelo I-AutoM.

Na Figura 7, apresentamos um resumo do desempenho de todos os modelos avaliados, considerando as métricas acurácia, precisão, recall, F1-score e ROC-AUC. O XGBoost se destacou em termos de equilíbrio entre essas métricas, enquanto o Naive Bayes teve um desempenho inferior em todas elas. A análise dessa figura ilustra o valor de se usar múltiplas métricas para a avaliação de modelos em tarefas de classificação, destacando os pontos fortes e fracos de cada abordagem.

VI. Discussão dos Resultados

A análise dos resultados obtidos a partir da aplicação do I-AutoM sobre diferentes bases de dados permite observar padrões recorrentes e diferenças importantes entre os algoritmos testados. De modo geral, modelos baseados em técnicas de ensemble — como Random Forest, XGBoost, LightGBM e CatBoost — apresentaram desempenho superior nas cinco métricas avaliadas (Accuracy, Precision, Recall, F1-score e ROC-AUC), com destaque para a robustez em bases com desbalanceamento de classes ou presença de ruído. Em particular, o XGBoost demonstrou consistência em quase todas as métricas, reforçando sua posição já consolidada na literatura como um dos algoritmos mais eficazes em tarefas supervisionadas [8].

Os gráficos evidenciam que modelos mais simples, como Naive Bayes e Regressão Logística, embora mais rápidos de treinar, apresentaram limitações em termos de recall e F1-score, especialmente em bases com variáveis categóricas não balanceadas ou atributos com alta correlação. O Naive Bayes, por exemplo, teve desempenho satisfatório em precisão, mas falhou ao capturar as classes menos representadas, refletido em F1-scores muito baixos, o que confirma as observações de Sokolova e Lapalme [12]. Já a Regressão Logística, embora competitiva em bases com separação linear clara, demonstrou

limitações em conjuntos com relações não lineares entre atributos.

Modelos baseados em vizinhança, como o K-Nearest Neighbors, obtiveram bons resultados em precisão e recall quando os dados estavam normalizados e com baixa dimensionalidade, mas perderam desempenho em bases com ruído ou alta dispersão. O SVM mostrou desempenho intermediário, com bom recall, mas variando bastante conforme a estrutura da base, reforçando seu conhecido desafio de escalabilidade em grandes conjuntos de dados [14].

Outro ponto observado foi que os modelos de rede neural, como o MLP Classifier, mostraram desempenho competitivo em bases mais complexas, mas com maior tempo de processamento e maior sensibilidade a ajustes nos dados e à normalização. Isso sugere que, para usuários menos experientes, modelos como Random Forest e XGBoost tendem a oferecer um melhor equilíbrio entre performance e facilidade de uso, principalmente quando não há necessidade de interpretar detalhadamente os coeficientes internos.

A apresentação gráfica dos resultados por métrica reforça a importância de avaliar os modelos com múltiplos critérios. Enquanto alguns algoritmos se destacaram em acurácia, outros tiveram melhor recall ou área sob a curva ROC, mostrando que a escolha do modelo ideal depende não apenas da base de dados, mas também da prioridade da tarefa. Por exemplo, em problemas onde a identificação correta de todas as classes é mais crítica do que a média global de acertos, métricas como recall e F1-score devem ser priorizadas em detrimento da acurácia pura.

Essas observações destacam a utilidade do I-AutoM como ferramenta exploratória, oferecendo não apenas a seleção automática dos modelos, mas também a visualização clara das diferenças de desempenho. Isso permite que o usuário tome decisões informadas com base em múltiplas métricas e compreenda melhor os impactos de suas escolhas, mesmo sem conhecimento técnico aprofundado em machine learning.

A. Aplicações Potenciais

A proposta do Auto Aprendiz tem potencial de aplicação em uma variedade de contextos práticos, tanto no meio acadêmico quanto profissional. Seu principal diferencial está na capacidade de automatizar o pipeline de classificação supervisionada sem exigir conhecimento técnico prévio em aprendizado de máquina, o que o torna uma ferramenta acessível para públicos diversos e com níveis distintos de experiência.

No contexto educacional, o I-AutoM pode ser utilizado como instrumento didático em disciplinas de ciência de dados, estatística, inteligência artificial ou análise de dados. A ferramenta permite que alunos explorem conceitos como seleção de modelos, métricas de avaliação e interpretação de resultados de forma prática e interativa. Isso elimina a barreira inicial da programação, permitindo que o foco recaia na compreensão dos algoritmos e no comportamento das métricas. Professores podem utilizá-lo para demonstrar o impacto da escolha da métrica na avaliação de modelos, bem como a influência do balanceamento de classes nos resultados obtidos.

Na área jurídica e institucional, o sistema pode ser aplicado para classificar automaticamente tipos de ações judiciais, decisões, áreas de competência ou movimentações processuais com base em atributos categóricos e temporais. Muitas dessas bases apresentam dados ruidosos, ausentes ou desbalanceados, e o I-AutoM, ao realizar imputação automática e normalização, permite que modelos de boa performance sejam gerados com pouco esforço. Isso representa uma oportunidade concreta para órgãos públicos ou tribunais que desejam aplicar inteligência de dados sem necessariamente contar com uma equipe técnica especializada.

No setor empresarial, a ferramenta pode ser aplicada em tarefas como segmentação de clientes, classificação de leads, análise de comportamento de compra ou categorização de produtos com base em atributos comerciais. Em pequenas e médias empresas, onde o acesso a soluções robustas de AutoML é limitado por orçamento ou infraestrutura, o I-AutoM representa uma alternativa viável, gratuita e executável localmente. Por exemplo, ao importar um arquivo CSV com registros de vendas, o usuário pode rapidamente obter um modelo que identifica padrões de consumo ou classifica clientes com maior propensão de conversão.

Além disso, sua interface visual e interativa permite que analistas de negócio, gestores públicos e pesquisadores utilizem a ferramenta como apoio exploratório. A possibilidade de testar múltiplos algoritmos, com métricas padronizadas e gráficos de comparação automática, fornece insights que orientam tomadas de decisão baseadas em dados. Mesmo em contextos onde o objetivo final não seja necessariamente a classificação, o I-AutoM pode servir como ferramenta de triagem e análise preliminar da estrutura de uma base, contribuindo para estudos mais aprofundados ou para a geração de hipóteses iniciais.

Por fim, considerando sua estrutura leve e compatibilidade com plataformas como Streamlit e bibliotecas padrão do Python, o I-AutoM pode ser facilmente integrado a sistemas web, portais institucionais ou ambientes internos de análise, o que amplia ainda mais seu campo de aplicação prática.

VII. Considerações Finais

A proposta do Auto Aprendiz demonstrou-se eficaz como uma solução automatizada para tarefas de classificação, especialmente no contexto de usuários que não possuem domínio avançado de aprendizado de máquina. Com base na aplicação em múltiplos conjuntos de dados reais, o sistema foi capaz de comparar, de maneira transparente e interativa, diferentes algoritmos supervisionados, utilizando métricas amplamente aceitas pela literatura. O resultado foi um ambiente acessível e visual que não apenas entrega a previsão, mas permite a análise crítica do desempenho dos modelos. Mesmo em bases desbalanceadas ou com dados complexos, o Auto Aprendiz conseguiu apresentar uma visão clara dos pontos fortes e fracos de cada abordagem testada.

Apesar dos resultados positivos, é importante destacar que o sistema ainda está em processo de maturação. A versão

atual depende da integridade estrutural do conjunto de dados carregado pelo usuário e ainda não realiza tratamentos corretivos mais sofisticados de forma autônoma. Em alguns casos, determinadas colunas ou tipos de variáveis podem prejudicar a performance geral dos modelos, exigindo ajustes prévios por parte do usuário. Diante disso, uma das metas prioritárias nas próximas versões do I-AutoM é a adição de procedimentos automáticos de tratamento de dados — como eliminação de colunas problemáticas, tratamento inteligente de outliers, conversão contextualizada de variáveis categóricas e detecção de colunas com baixa variância — de forma que o próprio sistema consiga ajustar a base de dados internamente, sem a necessidade de intervenção manual.

Essas melhorias têm como objetivo não apenas facilitar o uso, mas também elevar os índices de desempenho. Almeja-se alcançar resultados acima de 85% nas principais métricas (como acurácia e F1-score) para o melhor modelo em cada base, otimizando a entrega final sem comprometer a interpretabilidade. Para isso, serão incorporadas estratégias como ajustes automáticos de hiperparâmetros, escolha inteligente de pipelines de pré-processamento e refinamento das etapas de normalização, seleção de atributos e balanceamento de classes.

Como parte dos trabalhos futuros, também está prevista a integração com modelos de linguagem natural (LLMs), como o GPT, de forma a potencializar a autonomia do Auto Aprendiz. Em um primeiro momento, essa integração permitirá comparar diretamente os resultados do sistema com as sugestões fornecidas por uma LLM a partir da análise textual da base de dados, avaliando o grau de coerência e desempenho entre os dois caminhos. Em uma etapa mais avançada, a própria LLM poderá ser utilizada como apoio ao pipeline interno do Auto Aprendiz, auxiliando na análise semântica de variáveis, interpretação dos dados e até mesmo sugerindo transformações ou pré-ajustes baseados no contexto dos atributos. Esse tipo de colaboração entre um motor analítico e um modelo de linguagem representa um avanço significativo na direção de sistemas híbridos de inteligência artificial, mais adaptáveis, compreensivos e didáticos.

Como resultado deste artigo, pode-se concluir que o I-AutoML já se mostra uma ferramenta promissora para ambientes educacionais, exploratórios e até operacionais, e que, com as melhorias planejadas, tende a se consolidar como uma solução robusta e transparente para a classificação automatizada de dados. Ao equilibrar simplicidade, desempenho e explicabilidade, o sistema contribui com uma abordagem alternativa às plataformas de AutoML tradicionais, favorecendo o aprendizado e a confiança do usuário sobre os resultados obtidos.

REFERENCES

[1] M. Baratchi et al. Comparative analysis of automl systems for classification. *Artificial Intelligence Review*, 57, 2024.

[2] L. Breiman. Random forests. *Machine learning*, 2001.

[3] J.-S. Byeon et al. Application of svm and random forest for urban air quality prediction. *Sensors*, 21, 2021.

[4] T. Chen and C. Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 785–794, San Francisco, CA, USA, August 13–17 2016. ACM.

[5] M. Feurer, A. Klein, K. Eggensperger, J. Springenberg, M. Blum, and F. Hutter. Efficient and robust automated machine learning. *Advances in neural information processing systems*, 28, 2015.

[6] P. Gijsbers, E. LeDell, J. Thomas, and J. Vanschoren. An open source automl benchmark. In *Proceedings of the AutoML Conference*, 2022.

[7] F. Hutter, L. Kotthoff, and J. Vanschoren. *Automated Machine Learning: Methods, Systems, Challenges*. Springer, 2019.

[8] S. Ji et al. An application of a three-stage xgboost-based model to sales forecasting. *Mathematical Problems in Engineering*, 2019, 2019.

[9] T. Liang et al. Application of lightgbm to astronomical data classification. *The Astronomical Journal*, 163(153), 2022.

[10] P. Muthukkumar et al. Comparing the accuracy and developed models in machine learning approaches for detecting diabetes. *International Journal of Science and Research*, 11, 2023.

[11] L. Prokhorenkova, G. Gusev, A. Vorobev, A. Dorogush, and A. Gulin. Catboost: unbiased boosting with categorical features. *Advances in neural information processing systems*, 31, 2018.

[12] M. Sokolova and G. Lapalme. A systematic analysis of performance measures for classification tasks. *Information processing & management*, 45, 2009.

[13] Y. Sun, J. Wang, and X. Li. Challenges of applying automl in non-tabular data scenarios. *IEEE Access*, 2022.

[14] G. Varoquaux et al. Scikit-learn: Machine learning without the black box. *Communications of the ACM*, 61(11), 2018.

[15] J. Zhang, M. Zhang, Z. Ren, et al. Automl: A survey of the state-of-the-art. *International Journal of Artificial Intelligence*, 39(4):325–345, 2021.