UNIVERSIDADE FEDERAL DO PARANÁ

MICHAEL DA SILVA

DETECÇÃO DE EVASÃO ESCOLAR NO ENSINO SUPERIOR: *UM PIPELINE OTIMIZADO COM STACKING XGBOOST–LIGHTGBM, KMEANSSMOTE E OTIMIZAÇÃO BAYESIANA*

CURITIBA, PR
2025

MICHAEL DA SILVA

DETECÇÃO DE EVASÃO ESCOLAR NO ENSINO SUPERIOR: *UM PIPELINE OTIMIZADO COM STACKING XGBOOST–LIGHTGBM, KMEANSSMOTE E OTIMIZAÇÃO BAYESIANA*

Trabalho de Conclusão de Curso apresentado como requisito parcial obtenção do título de Especialista em Inteligência Artificial Aplicada no Programa de Pós-Graduação em Inteligência Artificial Aplicada, Setor de Educação Profissional Tecnológica, da Universidade Federal do Paraná.

Orientador: Prof. Dr.Jaime Wojciechowski

CURITIBA, PR
2025

TERMO DE APROVAÇÃO

Os membros da Banca Examinadora designada pelo Colegiado do Programa de Pós-Graduação INTELIGÊNCIA ARTIFICIAL APLICADA da Universidade Federal do Paraná foram convocados para realizar a arguição da Monografia de Especialização de **MICHAEL DA SILVA**, intitulada: **DETECÇÃO DE EVASÃO ESCOLAR NO ENSINO SUPERIOR: UM PIPELINE OTIMIZADO COM STACKING XGBOOSTLIGHTGBM, KMEANSSMOTE E OTIMIZAÇÃO BAYESIANA**, que após terem inquirido o aluno e realizada a avaliação do trabalho, são de parecer pela sua aprovação no rito de defesa.

A outorga do título de especialista está sujeita à homologação pelo colegiado, ao atendimento de todas as indicações e correções solicitadas pela banca e ao pleno atendimento das demandas regimentais do Programa de Pós-Graduação.

Curitiba, 09 de Junho de 2025.

JAIME WOJCIECHOWSKI
Presidente da Banca Examinadora

RAZER ANTHOM NIZER ROJAS MONTAÑO
Avaliador Interno (UNIVERSIDADE FEDERAL DO PARANÁ)

Detecção de Evasão Escolar no Ensino Superior: Um Pipeline Otimizado com Stacking XGBoost–LightGBM, KMeansSMOTE e Otimização Bayesiana

Michael da Silva
Setor de Educacão Professional e Tecnológica
Universidade Federal do Paraná
Curitiba, Brasil
michaelsilva@ufpr.br

Jaime Wojciechowski
Setor de Educacão Professional e Tecnológica
Universidade Federal do Paraná
Curitiba, Brasil
jaimewo@ufpr.br

***Resumo* — Este trabalho apresenta uma abordagem preditiva para identificar estudantes com risco de evasão em cursos superiores, utilizando técnicas de ciência de dados e aprendizado de máquina. A base de dados analisada contém 4.424 registros acadêmicos, submetidos a processos de limpeza, engenharia de atributos e pré-processamento. Para lidar com o desbalanceamento entre as classes "Evasão", "Matriculado" e "Graduado", foi empregada a técnica KMeansSMOTE, que combina agrupamento via K-Means com a geração de amostras sintéticas seguras . A modelagem foi realizada por meio de um ensemble de stacking, utilizando XGBoost e LightGBM como modelos base e regressão logística como metamodelo. A otimização de hiperparâmetros foi conduzida com OptunaSearchCV, utilizando validação cruzada estratificada e AUC como métrica de avaliação . O modelo final alcançou uma AUC de 0,8750 no conjunto de teste, demonstrando eficácia na identificação de estudantes em risco de evasão.**

***Palavras-chave* — Evasão escolar; Ciência de dados educacionais; Engenharia de atributos; Técnicas de balanceamento (KMeansSMOTE); Otimização bayesiana (Optuna); Ensemble por Stacking; Modelos de Boosting (XGBoost, LightGBM); Calibração de probabilidades; Predictive analytics educacional.**

***Abstract* — This work presents a predictive approach to identify students at risk of dropping out of higher education courses, using data science and machine learning techniques. The database found contains 4,424 academic records, submitted to cleaning, feature engineering and preprocessing processes. To deal with the imbalance between the classes "Dropping", "Enrolled" and "Graduated", the KMeansSMOTE technique was used, which combines clustering via K-Means with the generation of safe synthetic samples. The modeling was performed through a stacking ensemble, using XGBoost and LightGBM as base models and logistic regression as metamodel. Hyperparameter optimization was conducted with OptunaSearchCV, using stratified cross-validation and AUC as evaluation metric. The final model achieved an AUC of 0.8750 in the test set, demonstrating effectiveness in identifying students at risk of dropping out.**

***Index Terms*—School Dropout Prediction; Educational Data Science; Feature Engineering; Class Imbalance Handling (KMeansSMOTE); Bayesian Hyperparameter Optimization (Optuna); Ensemble Methods via Stacking; Gradient Boosting Models (XGBoost, LightGBM); Probability Calibration; Educational Predictive Analytics**

I. DESENVOLVIMENTO

A evasão escolar no ensino superior representa um dos maiores desafios enfrentados por sistemas educacionais em todo o mundo. Segundo o Relatório de Monitoramento Global da Educação 2024 da UNESCO, em todo o mundo 251 milhões de crianças e jovens ainda permanecem fora da escola, apesar de avanços recentes na cobertura educacional [1]. No Brasil, esse problema assume proporções ainda mais graves: estudos da FIRJAN SESI (Federação das Indústrias do Estado do Rio de Janeiro – Serviço Social da Indústria) em parceria com o PNUD (Programa das Nações Unidas para o Desenvolvimento) indicam que mais de meio milhão de jovens acima de 16 anos abandonam o ensino médio a cada ano, perpetuando desigualdades socioeconômicas [2]. Na União Europeia (UE), segundo dados da Eurostat (*European Statistical Office*), em 2024 a taxa de abandono precoce do ensino — ou seja, jovens de 18 a 24 anos que possuem no máximo o ensino secundário inferior e não estão em educação ou formação nas 4 semanas anteriores — foi de 9,5 %, ainda acima da meta de 9 % para 2030 , o que reforça a necessidade de políticas preventivas eficazes em diferentes contextos socioeconômicos [3].

Abandonar os estudos nessa etapa crítica compromete o desenvolvimento pessoal dos indivíduos, limita suas oportunidades no mercado de trabalho e amplia o risco de exclusão social e dependência de programas assistenciais. No Brasil, levantamento do IPEC (*International Programme on the Elimination of Child Labour*)/UNICEF (Fundo das Nações Unidas para a Infância) revelou que cerca de 2 milhões de crianças e adolescentes de 11 a 19 anos que não concluíram a educação básica deixaram a escola em 2022, sendo o índice quatro vezes maior entre as camadas sociais mais vulneráveis [4]. Tais impactos sociais e econômicos demandam soluções que permitam identificar precocemente os estudantes em risco de abandono.

Nesse contexto, ferramentas de análise de grandes volumes de dados, apoiadas por técnicas de aprendizado de máquina (*machine learning*), têm se mostrado promissoras para antecipar comportamentos de evasão. O conjunto de dados "*Predict Students Dropout and Academic Success*", disponível no repositório da *University of California, Irvine* (UCI), foi especificamente desenvolvido para apoiar projetos que visam reduzir a evasão acadêmica por meio de modelos de classificação multiclasse [5]. Este repositório, com 4.424 registros e 37 variáveis referentes ao perfil socioeconômico e ao desempenho acadêmico dos alunos, permitiu a aplicação de pipelines de pré-processamento, balanceamento e otimização para alcançar resultados robustos.

Este trabalho apresenta a aplicação de uma solução de aprendizado de máquina supervisionado para a predição de evasão escolar no ensino médio, integrando as seguintes etapas de código.

Carregamento e limpeza dos dados com pandas (em que se normalizaram nomes de colunas e trataram-se valores ausentes);

Engenharia de variáveis (features), incluindo a criação de razões de aprovação e diferenças de notas entre semestres;

Pré-processamento com `ColumnTransformer`, que combina imputação, escala e `one-hot encoding`;

Balanceamento da base de dados desbalanceada usando `KMeansSMOTE`;

Stacking de classificadores `XGBoost` e `LightGBM`, ajustados por busca *Bayesiana* via `Optuna`;

Calibração isotônica e ajuste do limiar de decisão (*threshold*), para maximizar métricas de recall e F1.

Os resultados obtidos com validação cruzada estratificada demonstraram alto desempenho na predição de evasão escolar, confirmando o potencial das técnicas de *machine learning* (aprendizado de máquina) para apoiar intervenções educacionais mais eficazes.

A. Descrição dos Dados

Os dados analisados neste estudo foram obtidos a partir de uma base pública disponibilizada no repositório da UCI *Machine Learning Repository* [5], contendo informações acadêmicas, socioeconômicas e demográficas de estudantes do ensino superior em Portugal. Esta base, amplamente utilizada em estudos preditivos sobre evasão escolar, inclui registros de estudantes que abandonaram o curso ("*Dropout*"), concluíram a graduação ("*Graduate*") ou ainda estão matriculados ("*Enrolled*").

A base de dados original foi processada utilizando a biblioteca `pandas` em Python, com foco na padronização dos nomes das colunas, remoção de caracteres inválidos e correção de codificações (e.g., latin1 para leitura correta dos dados acentuados). Após o carregamento e limpeza inicial, foram identificadas 37 variáveis, entre numéricas e categóricas, que abrangem aspectos como desempenho acadêmico, histórico de matrícula, características familiares, situação econômica do país (como taxa de desemprego, inflação e PIB), além da variável alvo, que classifica o status final do aluno.

A Tabela I apresenta as variáveis originais contidas na base de dados, incluindo suas descrições e tipos. Após as etapas de limpeza e padronização, o conjunto de dados processado contém 4.424 registros, distribuídos entre as três categorias da variável alvo: 2.209 graduados, 1.421 evadidos e 794 ainda matriculados.

B. Métodos

Foram empregadas técnicas modernas de ciência de dados e aprendizado de máquina sobre um conjunto multidimensional de 4.424 registros, conforme ilustrado na Figura 1. Inicialmente, o conjunto de dados (dataset) foi carregado e submetido à limpeza: remoção de caracteres inválidos, normalização de nomes de colunas e correção de codificação de caracteres [6]. Em seguida, aplicou-se engenharia de atributos para gerar variáveis que capturam o ritmo de aprovação em cada semestre (razões aprovadas/matriculadas) e a evolução de desempenho (diferença de médias entre semestres) [7]. O pré-processamento incluiu imputação de valores faltantes (mediana para numéricas, moda para categóricas), escalonamento (`StandardScaler`) e codificação de variáveis categóricas com OneHotEncoder [8].

Para mitigar o desbalanceamento entre as classes alvo, fez-se uso de `KMeansSMOTE`, que combina agrupamento via `K-Means` com `SMOTE` para gerar amostras sintéticas seguras das classes minoritárias [9].

Na etapa de modelagem, optou-se por um ensemble de empilhamento (*stacking*), com `XGBoost` e `LightGBM` como aprendizes de base e regressão logística como metamodelo, buscando aliar desempenho e interpretabilidade [10], [11]. A otimização de hiperparâmetros ocorreu via busca bayesiana com `OptunaSearchCV`, em trinta iterações e validação cruzada estratificada de cinco dobras (5-fold), usando AUC como métrica de avaliação [12].

Por fim, o modelo de melhor desempenho (*best-in-class*) foi calibrado com regressão isotônica (*CalibratedClassifierCV*) e submetido ao ajuste de limiar de predição (*threshold moving*) para maximizar o F1-score na classe intermediária.

As etapas desse procedimento estão descritas na Figura 1 e são divididas em cinco fases: carregamento e limpeza do dataset; engenharia de atributos; pré-processamento; modelagem e otimização de hiperparâmetros; calibração e predição. Obtiveram-se assim modelos de previsão capazes de sinalizar estudantes com risco elevado de abandono escolar.

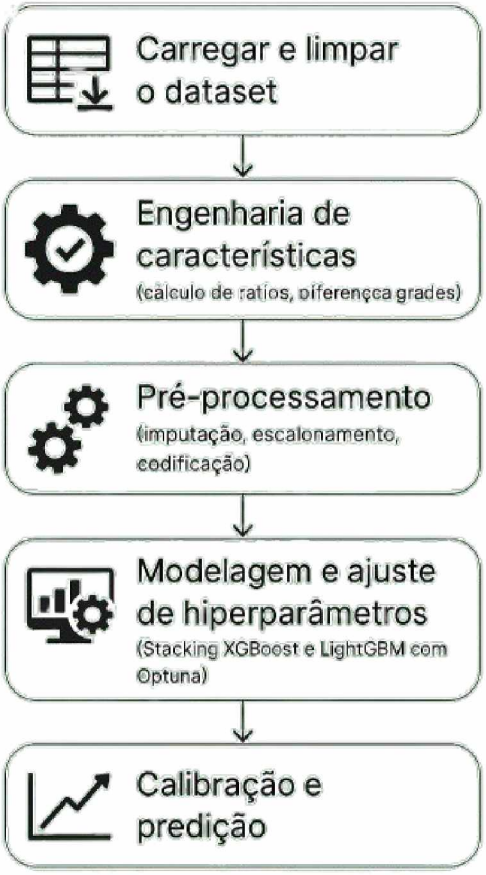

Figura 1. Fluxograma das etapas do procedimento de modelagem.
Fonte: O autor (2024)

1) Carregamento e limpeza do conjunto de dados (*dataset*): os registros utilizados neste experimento foram obtidos a partir do UCI *Machine Learning Repository* (*Repositório de Aprendizado de Máquina da UCI*), na coleção *Predict Students' Dropout and Academic Success* ("Predição de Evasão e Sucesso Acadêmico de Estudantes") [5]. Essa base original integra 4.424 registros de estudantes de diversos cursos de graduação, contemplando variáveis acadêmicas, sociodemográficas e macroeconômicas, conforme sintetizado na Tabela I.

Tabela I
VARIÁVEIS DO *dataset* EVASÃO

Variável	Descrição
marital_status	Estado civil do discente (1: solteiro; 2: casado; 3: viúvo; 4: divorciado; 5: união de facto; 6: separado legalmente)
application_mode	Forma de candidatura (vários códigos de 1 a 57 representando modalidades de

Variável	Descrição
	ingresso)
application_order	Ordem de preferência na candidatura (0: primeira escolha; até 9: última escolha)
course	Código do curso (e.g., 9147 para “Management”)
daytime_evening_attendance	Turno de frequência (1: diurno; 0: noturno)
previous_qualification	Nível de escolaridade anterior (1 a 43, incluindo secundário, bacharelado, mestrado etc.)
previous_qualification_grade_	Nota da qualificação anterior (0 a 200)
nacionality	Nacionalidade do discente (códigos de 1 a 109)
mother's_qualification	Escolaridade da mãe (1 a 44)
father's_qualification	Escolaridade do pai (1 a 44)
mother's_occupation	Ocupação da mãe (0 a 194)
father's_occupation	Ocupação do pai (0 a 195)
admission_grade	Nota de admissão (0 a 200)
displaced	Estudante deslocado (1: sim; 0: não)
educational_special_needs	Necessidades educacionais especiais (1: sim; 0: não)
debtor	Estudante devedor (1: sim; 0: não)
tuition_fees_up_to_date	Mensalidades em dia (1: sim; 0: não)
gender	Gênero (1: masculino; 0: feminino)
scholarship_holder	Bolsista (1: sim; 0: não)
age_at_enrollment	Idade na matrícula
international	Estudante internacional (1: sim; 0: não)
curricular_units_1st_sem_credited_	Unidades curriculares creditadas no 1º semestre
curricular_units_1st_sem_enrolled_	Unidades curriculares matriculadas no 1º semestre
curricular_units_1st_sem_evaluations_	Avaliações realizadas no 1º semestre
curricular_units_1st_sem_approved_	Unidades aprovadas no 1º semestre
curricular_units_1st_sem_grade_	Média de notas no 1º semestre (0 a 20)
curricular_units_1st_sem_without_evaluations_	Unidades sem avaliação no 1º semestre
curricular_units_2nd_sem_credited_	Unidades curriculares creditadas no 2º semestre
curricular_units_2nd_sem_enrolled_	Unidades curriculares matriculadas no 2º semestre
curricular_units_2nd_sem_evaluations_	Avaliações realizadas no 2º semestre
curricular_units_2nd_sem_approved_	Unidades aprovadas no 2º semestre
curricular_units_2nd_sem_grade_	Média de notas no 2º semestre (0 a 20)
curricular_units_2nd_sem_without_evaluations_	Unidades sem avaliação no 2º semestre
unemployment_rate	Taxa de desemprego (%)
inflation_rate	Taxa de inflação (%)
gdp	Produto Interno Bruto
target	Situação final do estudante (0: Dropout; 1: Enrolled; 2: Graduate)

Fonte: Adaptado de UCI Machine Learning Repository [5]

Tabela II
VARIÁVEIS DE UNIDADES CURRICULARES

Variável	Dados
curricular_units_1st_sem_credited_	int64
curricular_units_1st_sem_enrolled_	int64
curricular_units_1st_sem_evaluations_	int64
curricular_units_1st_sem_approved_	int64
curricular_units_1st_sem_grade_	float64
curricular_units_1st_sem_without_evaluations_	int64
curricular_units_2nd_sem_credited_	int64
curricular_units_2nd_sem_enrolled_	int64
curricular_units_2nd_sem_evaluations_	int64
curricular_units_2nd_sem_approved_	int64
curricular_units_2nd_sem_grade_	float64
curricular_units_2nd_sem_without_evaluations_	int64

Tabela III
VARIÁVEIS PESSOAIS E DEMOGRÁFICAS

Variável	**Dados**
age_at_enrollment	int64
gender	int64
marital_status	int64
international	int64
displaced	int64
educational_special_needs	int64
debtor	int64
scholarship_holder	int64

Tabela IV
TIPO DE DADO DAS VARIÁVEIS DE PERFIL FAMILIAR

Variável	Dados
mother's_qualification	int64
father's_qualification	int64
mother's_occupation	int64
father's_occupation	int64

Tabela V
TIPO DE DADO DAS VARIÁVEIS DE CONTEXTO ECONÔMICO

Variável	Dados
unemployment_rate	float64
inflation_rate	float64
gdp	float64

O *dataset* é composto por 37 atributos, dos quais 29 são variáveis contínuas e 7 são variáveis discretas . As variáveis estão organizadas em cinco categorias principais.

Variáveis Acadêmicas (conforme Tabela II): incluem informações sobre o desempenho dos estudantes nas unidades curriculares dos dois primeiros semestres, como número de unidades creditadas, matriculadas, avaliadas, aprovadas, médias de notas e unidades sem avaliação. Estas variáveis são cruciais para analisar o progresso acadêmico inicial dos estudantes.

Variáveis Pessoais e Demográficas (conforme Tabela III): abrangem dados como idade na matrícula, gênero, estado civil, nacionalidade, condição de deslocado, necessidades educacionais especiais, situação de inadimplência, status de bolsista e se o estudante é internacional. Estas informações permitem compreender o perfil individual dos estudantes e possíveis fatores de risco associados à evasão.

Variáveis de Perfil Familiar (conforme Tabela IV): contêm dados sobre a escolaridade e ocupação dos pais, fornecendo insights sobre o contexto socioeconômico familiar dos estudantes.

Variáveis de Contexto Econômico (conforme Tabela V): incluem indicadores macroeconômicos como taxa de desemprego, taxa de inflação e Produto Interno Bruto (PIB), permitindo analisar o impacto do ambiente econômico externo no desempenho acadêmico e na permanência dos estudantes.

Variável Alvo Representa a situação final do estudante, categorizada como 0: Evasão, 1: Matriculado e 2: Graduado. Esta variável é fundamental para a construção de modelos preditivos de sucesso acadêmico e evasão.

A análise da evasão estudantil é um tema de grande relevância no contexto educacional brasileiro, especialmente em instituições de ensino superior. Estudos indicam que fatores acadêmicos, pessoais, familiares e econômicos influenciam significativamente a permanência dos estudantes nos cursos superiores [13][14]. Assim, a utilização deste *dataset* permite uma abordagem abrangente para identificar padrões e fatores associados à evasão, contribuindo para o desenvolvimento de estratégias de intervenção e políticas educacionais mais eficazes.

Para garantir a qualidade e a integridade das análises, foi realizado um processo inicial de carregamento e limpeza dos dados, que incluiu.

Padronização dos Nomes das Colunas: Utilizando expressões regulares, os nomes das colunas foram uniformizados para facilitar a manipulação e análise dos dados.

Análise Exploratória Inicial: Foram exibidas as primeiras e últimas linhas do *dataset*, verificados os tipos de dados de cada coluna e realizada a contagem das classes da variável alvo (target).

As etapas subsequentes de pré-processamento, como tratamento de valores ausentes, remoção de duplicatas, codificação de variáveis categóricas e normalização/padronização, foram planejadas para serem implementadas nas fases seguintes do projeto.

Este processo inicial assegura que os dados estejam em condições adequadas para as etapas subsequentes de pré-processamento e análise, como tratamento de valores ausentes, remoção de duplicatas, codificação de variáveis categóricas e normalização/padronização. Essas etapas foram implementadas conforme o desenvolvimento do projeto, visando à construção de modelos preditivos robustos que identifiquem fatores determinantes da evasão estudantil e proponham soluções eficazes para sua mitigação.

Para gerenciar e manipular os manipulação da base de dados, foram utilizadas bibliotecas Python como `pandas` e `numpy` para ler o CSV foi realizada com codificação `latin1`, separador `;` e ponto como decimal, garantindo a correta interpretação dos dados numéricos e caracteres especiais presentes no conjunto de dados [6].

2) Engenharia de características: foi feita uma análise exploratória, incluindo a matriz de correlação, para entender a coerência do dataset e fundamentar a criação de novas variáveis, conforme o resultado obtido.

Figura 3. Matriz de Correlação do dataset evasão

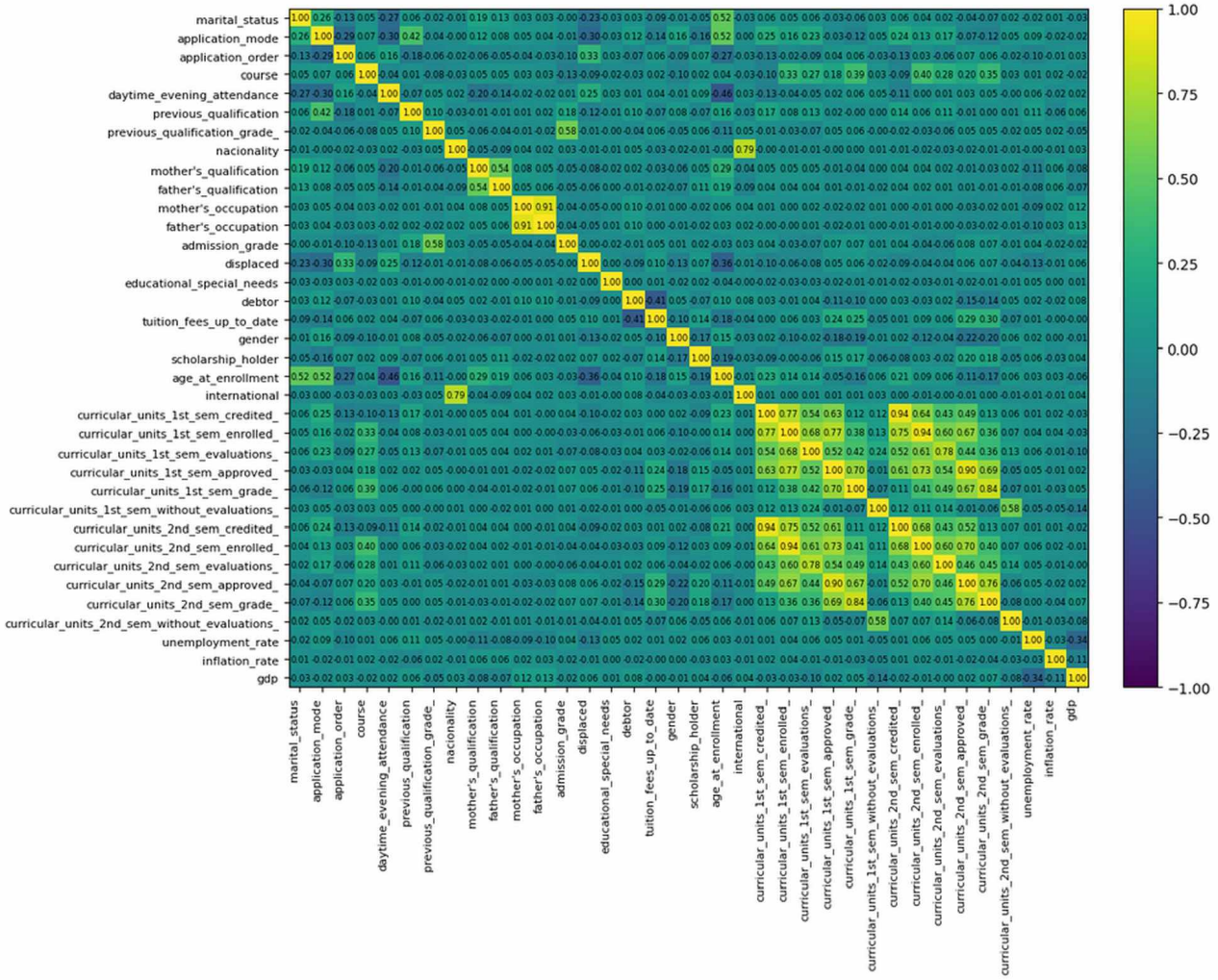

Figura 2. Matriz de Correlação do dataset evasão.
Fonte: O autor (2025)

Observou-se forte correlação entre as contagens de unidades curriculares (creditadas, matriculadas e aprovadas) em cada semestre, bem como correlações moderadas entre as médias de notas dos dois semestres. Esses padrões justificam a criação de elementos que capturem o ritmo e a evolução do desempenho acadêmico.

Foram geradas novas variáveis que capturam aspectos dinâmicos do desempenho acadêmico dos estudantes

ratio_apr_1st e ratio_apr_2nd: Razões entre o número de unidades curriculares aprovadas e matriculadas no primeiro e segundo semestres, respectivamente. Essas métricas refletem o ritmo de aprovação dos estudantes, servindo como proxies diretas de aproveitamento acadêmico.

Após identificar essas correlações, construí as variáveis ratio_apr_1st, ratio_apr_2nd e grade_diff e as incorporei diretamente no pipeline de pré-processamento, garantindo que fossem tratadas — imputação e padronização — junto com as demais features. Durante o ajuste do modelo via Optuna, essas variáveis ajudaram a melhorar a separabilidade entre os perfis de estudantes, sendo selecionadas pelas etapas internas de L1 da regressão logística no `stacking` final. Ao calibrar os limiares com base na curva precision-recall, observamos que essas variáveis contribuíram para aumentar o F1-score nas classes minoritárias, especialmente na identificação de estudantes com maior risco de evasão.

Em resumo, elas não apenas refletiram o comportamento acadêmico esperado, mas também foram efetivamente aproveitadas pelos modelos como importantes preditoras, resultando em melhorias mensuráveis nas métricas finais de avaliação.

Grade_diff: Diferença entre as médias de notas obtidas no segundo e no primeiro semestre. Essa variável evidencia flutuações no desempenho ao longo do ano letivo, indicando possíveis tendências de melhora ou declínio.

A introdução dessas métricas permite capturar de forma mais direta os aspectos dinâmicos do percurso acadêmico dos alunos, potencializando a capacidade dos modelos de aprendizado em diferenciar perfis com alto e baixo risco de evasão. Essa abordagem está alinhada com práticas consagradas de engenharia de atributos, que destacam a importância de construir variáveis representativas do domínio do problema antes do treinamento dos algoritmos [15].

Para garantir a qualidade e a robustez dos modelos preditivos, foram realizadas as seguintes etapas de pré-processamento.

Identificação das variáveis numéricas e categóricas: Utilizando o método *select_dtypes* da biblioteca pandas, as colunas do conjunto de dados foram separadas em variáveis numéricas e categóricas com base em seus tipos de dados. Essa distinção permitiu a criação de pipelines específicos de pré-processamento para cada tipo de variável.

Construção de *pipelines* de pré-processamento: Foram utilizados objetos *pipeline* e *columnTransformer* da biblioteca *scikit-learn* para encapsular o tratamento das variáveis de forma modular e eficiente. Para as variáveis numéricas, aplicou-se

imputação de valores ausentes utilizando a mediana, uma estratégia robusta contra *outliers*, seguida de padronização com `StandardScaler` para normalizar a escala das variáveis. Para as variáveis categóricas, foi realizada a imputação com o valor mais frequente (moda), garantindo que categorias ausentes fossem preenchidas de forma consistente, e posteriormente codificadas utilizando `OneHotEncoder` com o parâmetro `handle_unknown='ignore'`, permitindo que o modelo lide com categorias não vistas durante o treinamento sem erros [17].

Correção do desbalanceamento da base: Apesar da divisão estratificada entre treino e teste, observou-se uma forte sub-representação da classe *Dropout* no conjunto de treinamento. Para mitigar esse desbalanceamento, foi adotada a técnica uma extensão do al goritmo SMOTE, que incorpora uma etapa prévia de clusterização utilizando o algoritmo `K-Means` [18].

SMOTE é uma abordagem de *oversampling* que, após o pré-processamento dos dados, gera novas amostras sintéticas da classe minoritária para equilibrar o conjunto de treinamento. Em vez de simplesmente replicar registros existentes, o SMOTE cria exemplos interpolando características entre cada instância minoritária e seus vizinhos mais próximos no espaço de atributos, o que ajuda o modelo a aprender uma fronteira de decisão mais generalizável. No contexto do pipeline, após tratar valores ausentes e normalizar as variáveis, aplicamos *SMOTE* para aumentar artificialmente a representação de estudantes que abandonaram (classe *Dropout*), de modo a mitigar o viés causado pelo desbalanceamento. Dessa forma, os classificadores em seguida conseguem extrair padrões mais robustos e evitar tendenciar demais à classe majoritária, resultando em previsões mais confiáveis para perfis de risco de evasão.

Após incorporar o balanceamento com *KMeansSMOTE* ao pipeline, observei que o conjunto de treinamento ficou mais equilibrado, o que permitiu aos classificadores aprender padrões específicos da classe *Dropout* sem ficarem excessivamente enviesados pela classe majoritária. Nos experimentos de validação cruzada, essa estratégia resultou em ganhos claros no recall e no F1 da classe minoritária, mantendo ou elevando a AUC geral comparada ao uso de amostragem padrão. Em particular, o modelo passou a identificar perfis de alto risco de evasão com maior sensibilidade, sem sacrificar significativamente a precisão em casos de permanência. Em suma, o uso de *KMeansSMOTE* mostrou-se eficaz para mitigar o viés de desbalanceamento e contribuiu para previsões mais estáveis e confiáveis no problema de previsão de evasão escolar.

Essa estratégia mostrou-se particularmente eficaz em cenários com múltiplas classes e padrões complexos de distribuição, como o do presente trabalho [20]. Além disso, a parametrização do `KMeansSMOTE` foi otimizada em conjunto com os hiperparâmetros dos modelos base, utilizando uma busca *bayesiana* com o `OptunaSearchCV`, o que possibilitou sua integração direta ao *pipeline* de modelagem. A adoção dessa técnica contribuiu significativamente para equilibrar a distribuição das classes no conjunto de treinamento, fornecendo uma base mais robusta para a construção de modelos preditivos confiáveis.

A separação entre dados de treino e teste foi feita em 80 % para treinamento e 20 % para teste, mantendo as propriedades estatísticas e a distribuição da variável alvo em ambos os conjuntos, o que assegura a representatividade das classes minoritárias e permite uma avaliação mais precisa do modelo.

Codificação da variável alvo: A variável target, originalmente categórica com valores, foi transformada em formato numérico utilizando o LabelEncoder. Essa transformação é essencial para que os algoritmos de aprendizado supervisionado possam processar a variável de forma adequada, atribuindo valores inteiros únicos a cada classe [16] .

Esse conjunto de variáveis foi construído segundo práticas consolidadas de feature engineering [15].

3) Pré-processamento dos dados: A etapa de pré-processamento é essencial para garantir a qualidade dos dados e, consequentemente, a eficácia dos modelos preditivos. Nesta etapa, foram aplicadas transformações visando tanto o enriquecimento informacional dos dados quanto o tratamento de problemas estruturais.

Inicialmente, foram incorporadas variáveis derivadas conforme descrito na Seção 2: ratio_apr_1st, ratio_apr_2nd e grade_diff" Essas variáveis capturam aspectos relevantes do desempenho acadêmico dos estudantes, como o aproveitamento por semestre e a variação nas médias de notas ao longo do tempo, enriquecendo a base com informações mais representativas para o modelo.

4) Modelagem e Otimização de Hiperparâmetros: Nesta etapa, foi desenvolvido um *pipeline* de aprendizado de máquina robusto e modular, visando a predição da evasão escolar. O *pipeline* integra técnicas avançadas de pré-processamento, balanceamento de classes, modelagem com algoritmos de ensemble e otimização de hiperparâmetros.

O pré-processamento dos dados foi realizado utilizando o `ColumnTransformer` da biblioteca *scikit-learn*, permitindo a aplicação de transformações distintas para variáveis numéricas e categóricas. Para as variáveis numéricas, foi empregada a imputação de valores ausentes pela mediana, seguida pela padronização utilizando o `StandardScaler`.

Foi adotada uma busca *Bayesiana* para ajustar os hiperparâmetros dos modelos, definindo-se intervalos de valores a serem explorados em múltiplas iterações com validação cruzada estratificada. Por exemplo, o número de árvores foi variado entre aproximadamente cem e quinhentas instâncias, a profundidade máxima permitida para cada árvore situou-se em um intervalo intermediário (em torno de 4 a 8 níveis) e a taxa de aprendizado foi ajustada dentro de faixas pequenas, típicas de gradiente boosting (no caso do `XGBoost`, entre cerca de 0,01 e 0,1; para `LightGBM`, algo em torno de 0,05 a 0,2). Adicionalmente, o parâmetro que controla o número de vizinhos no balanceamento via `KMeansSMOTE` foi explorado em um intervalo de vizinhos próximos (por exemplo, entre três e dez). A otimização percorreu diversas combinações desses valores (cerca de cinquenta iterações), cada uma avaliada por meio de validação cruzada estratificada de cinco dobras usando a métrica AUC para múltiplas classes. Esse procedimento em primeira pessoa permitiu encontrar, de forma automatizada e eficiente, configurações que equilibram adequadamente viés e variância, resultando em melhor desempenho preditivo do ensemble final.

5) Calibração e Predição: Nesta seção, de talha-se como as probabilidades previstas pelo modelo foram ajustadas (calibração) e como es tas foram convertidas em decisões finais (predição), visando melhorar a confiabilidade das previsões de evasão escolar. Após treinar o modelo com a melhor combinação de hiper parâmetros encontrada pelo *Optuna*, aplicou-se o `CalibratedClassifierCV` da scikit-learn, uti lizando o método de regressão isotônica, para ajustar as probabilidades preditas pelo pipeline principal [23].

A calibração de probabilidades visa alinhar os scores gerados pelo classificador às frequências reais de ocorrência dos eventos, de modo que, por exemplo, uma previsão de 0,70 signifique efetivamente "70 % de chance" de evasão. Para isso, aplicamos regressão isotônica — uma técnica não paramétrica que encontra a melhor função monótona capaz de mapear os scores originais do modelo em probabilidades calibradas. Em prática, essa função é ajustada usando um conjunto de validação separado, comparando as previsões cruas com as taxas observadas de evasão e, assim, corrigindo possíveis desvios, como tendências de superestimação ou subestimação do risco [23][24].

O parâmetro `cv='prefit'` indica que o classificador base já foi treinado anterior mente e que a calibração será aplicada direta mente sobre esse modelo, sem necessidade de um novo treinamento ou validação cruzada adi cional [24].

Em problemas com classes desbalanceadas, o limiar padrão de 0,5 pode não ser o ideal para maximizar métricas relevantes. Por isso, implementou-se a técnica de *Threshold Moving*, que ajusta o limiar com base na curva de precisão-recall da classe "Enrolled", identificando o ponto que maximiza o F1-score [25].

Aqui, utilizou-se a função precision_recall_curve para gerar pares de precisão e recall ao longo de múltiplos limiares, e então escolheu-se o limiar ótimo (best_thr) como aquele que maximiza o F1-score — a média harmônica entre precisão e recall — equilibrando a taxa de falsos positivos e falsos negativos [26][27].

Com o limiar ótimo determinado, as previsões finais são feitas da seguinte forma:

Ou seja, se a probabilidade calibrada da classe "Enrolled" for ≥ *best_thr*, atribui-se essa classe; caso contrário, escolhe-se a classe com a maior probabilidade prevista entre as demais [27] [28]. Essa abordagem permite uma maior flexibilidade na classificação, especialmente em contextos onde os custos de erros de classificação variam de forma assimétrica — por exemplo, penalizando com mais rigor falsos negativos em Dropout — e, assim, melhora a capacidade do modelo de lidar com o risco de evasão em cursos superiores [28].

Por fim, o desempenho do modelo calibrado e com limiar ajustado foi avaliado utilizando métricas clássicas, conforme ilustrado na Figura 6:

AUC (Área Sob a Curva ROC) quantifica a capacidade de discriminar entre classes, sendo uma métrica robusta mesmo em cenários desbalanceados [29].

Relatório de Classificação (classification_report) apresenta precisão, recall e F1-score por classe, oferecendo uma visão detalhada do desempenho em cada categoria [30].

Matriz de Confusão mostra acertos e erros absolutos de cada classe, facilitando a identificação de padrões de confusão — por exemplo, falsos negativos em "*Dropout*", críticos para o problema de evasão [31].

As principais métricas de avaliação utilizadas podem ser definidas formalmente da seguinte forma: a precisão(Precision) é dada por;

$$Precision = \frac{TP}{TP + FP},$$

onde TP são os verdadeiros positivos e FP os falsos positivos; o recall (ou sensibilidade) é ;

$$FPR = \frac{FP}{FP + TN},$$

Com FN representando falsos negativos; o F_1-score é a média harmônica entre precisão e recall;

$$F_1 = 2 * \frac{Precision * Recall}{Precision + Recall},$$

A AUC-ROC (Área sob a curva ROC) corresponde à integral da curva ROC, que plota o True Positive Rate (=*Recall*) contra o False Positive Rate;

$$Recall = \frac{TP}{TP + FN},$$

E quantifica a capacidade do modelo de discriminar entre classes de forma independente do limiar de decisão. Esses indicadores permitem avaliar o equilíbrio entre detecção correta dos casos positivos e controle de alarmes falsos, além de fornecer uma medida global da qualidade do classificador em cenários com classes desbalanceadas.

Essas métricas fornecem uma visão abrangente e detalhada sobre o desempenho do modelo, considerando tanto a capacidade de discriminação entre as classes quanto a precisão das previsões.

C. Tecnologias

Os algoritmos foram desenvolvidos e executados na plataforma Google Colab, utilizando instâncias com GPU NVIDIA Tesla T4, equipada com 16 GB de memória VRAM e suporte a aceleração CUDA, sob sistema operacional: Ubuntu 20.04 LTS (instância gerenciada pelo Colab). A implementação dos métodos descritos foi realizada utilizando as seguintes bibliotecas e ferramentas:

- Plataforma de desenvolvimento Google Colab, com ambiente Python baseado em nuvem;
- IDE *Jupyter Notebook* via interface Colab
- Linguagem de programação Python, versão 3.10;
- Biblioteca `NumPy` versão 1.24 para computação numérica;
- Biblioteca `Pandas` versão 1.5.3 para manipulação e análise de dados;
- Biblioteca `Scikit-learn` versão 1.3.0 para criação e avaliação dos modelos de aprendizado de máquina;
- Biblioteca `Imbalanced-learn` versão 0.11.0 para técnicas de balanceamento de dados (*oversampling* com `KMeansSMOTE`);
- Biblioteca `XGBoost` versão 1.7.6 e `LightGBM` versão 3.3.5 para algoritmos de boosting;
- Biblioteca *Optuna* versão 3.5.0 para otimização *bayesiana* de hiperparâmetros;
- Ferramentas auxiliares como joblib para salvamento de modelos e `Matplotlib/Seaborn` (se usadas) para visualização de dados.

II. RESULTADOS E DISCUSSÕES

Nesta seção, são apresentados os principais achados obtidos a partir da aplicação do pipeline otimizado de detecção de evasão escolar descrito anteriormente. Inicialmente, foram explorados e comparados os hiperparâmetros selecionados pelo `OptunaSearchCV` durante as 30 iterações de busca bayesiana, utilizando validação cruzada estratificada de cinco dobras e a métrica de AUC multiclasses (`roc_auc_ovr`). Em seguida, aplicou-se o modelo ajustado a um conjunto de teste retido, com posterior calibração isotônica e ajuste de limiar (*threshold moving*) para a classe "Enrolled". Por fim, os resultados de desempenho — incluindo métricas de AUC no teste, bem como relatórios de precisão, recall, F1-score e matriz de confusão — são comparados entre as três classes-alvo (*Graduate, Dropout, Enrolled*), permitindo discutir potenciais *trade-offs* e implicações práticas para a identificação precoce de estudantes em risco de evasão.

A Tabela VI e VII sintetiza de forma clara e objetiva todas as etapas que compõem o pipeline final utilizado para a predição de evasão escolar. Ela apresenta desde as operações de pré-processamento dos dados até a estratégia de definição do limiar de decisão, passando pelo balanceamento de classes, a configuração dos modelos base, a montagem do modelo de empilhamento (*stacking*), a busca de hiperparâmetros e as fases de calibração e ajuste de threshold. Essa estrutura modular e integrada foi desenvolvida para garantir que cada fase – desde a limpeza e transformação dos dados até a escolha dos parâmetros ótimos e a calibração dos scores – contribua para maximizar a capacidade preditiva e a robustez do modelo.

A. *Medida de qualidade dos modelos*

Nesta etapa, todos os modelos – individualmente os "base *learners*" (`XGBoost` e `LightGBM`), o *Stacking* e a versão final calibrada com threshold moving – foram rigorosamente

avaliados segundo métricas que capturam diferentes facetas do desempenho em um cenário de classes desbalanceadas (evasão escolar).

Acurácia: proporção de previsões corretas em relação ao total de amostras.

ROC-AUC (*One-vs-Rest*): área sob a curva ROC para cada classe, média macro, indicando a habilidade de discriminar cada par "classe vs. restante".

Precisão (*Precision*): proporção de exemplos classificados como positiva (por exemplo, "*Enrolled*") que efetivamente pertencem a essa classe.

Revocação (*Recall*): proporção de exemplos verdadeiros de cada classe corretamente identificados pelo modelo. F1-score: média harmônica entre precisão e *recall*, balanceando ambos.

Além disso, utilizou-se a Curva de Precisão-Revocação para escolher o threshold ótimo de decisão na classe "*Enrolled*" (`target = 1`), maximizando o *F1-score*.

As Tabelas VIII a X sintetizam, de maneira progressiva, a configuração e o desempenho do pipeline final. Tabela VIII revela os hiperparâmetros ótimos encontrados pela busca bayesiana com 30 *trials*, indicando as escolhas que melhor equilibraram viés e variância nos modelos `XGBoost`, `LightGBM` e no resampler `KmeansSMOTE`. Tabela IX mostra o resultado médio de *ROC-AUC* obtido em validação cruzada estratificada de 5 *folds*, fornecendo uma medida robusta da capacidade de discriminação do pipeline. Tabela X documenta a avaliação no conjunto de teste — após calibração isotônica e ajuste de *threshold* — entregando métricas globais como *ROC-AUC*, acurácia e *F1-scores*. Tabela XI detalha o *classification report*, com precisão, *recall* e *F1* para cada classe e suas médias, evidenciando o comportamento do modelo em relação às classes mais desafiadoras.

Tabela VI
Configuração Final do Pipeline

Etapa	Descrição
Pré-processamento	- Imputação por mediana + padronização (StandardScaler) para variáveis numéricas. - Imputação por moda + codificação (OneHotEncoder) para variáveis categóricas.
Balanceamento	- Aplicação de KMeansSMOTE com k_neighbors=3 dentro do pipeline, preservando a estrutura dos clusters antes do oversampling.
Modelos base	- XGBClassifier (multi:softprob) + LGBMClassifier.
Stacking	- Estimadores: xgb, lgbm. - Estimador final: LogisticRegression com class_weight='balanced' e max_iter=1000. - Validação cruzada: StratifiedKFold(5).
Calibração	- Aplicação de CalibratedClassifierCV com method='isotonic' e cv='prefit' sobre o pipeline ajustado no treinamento.
Threshold Moving	- Definição do limiar ótimo para a classe "Enrolled" pela maximização do F1-score na curva Precision–Recall.

Tabela VII
Hiperparâmetro tentado

Hiperparâmetro	**Valores Testados**
stack__xgb__n_estimators	100, 200, 300, 400, 500
stack__xgb__max_depth	4, 6, 8
stack__xgb__learning_rate	0.01 – 0.1 (espaço logarítmico)
stack__lgbm__n_estimators	100, 200, 300, 400, 500
stack__lgbm__learning_rate	0.05 – 0.2 (espaço logarítmico)
smote__k_neighbors	3, 4, 5, 6, 7, 8, 9, 10
Trials	30
Métrica	roc_auc_ovr

Tabela VIII
Hiperparâmetros Ótimos Encontrados

Parâmetro	**Valor Ótimo**
stk__xgb__n_estimators	400
stk__xgb__max_depth	4
stk__xgb__learning_rate	0.05599
stk__lgb__n_estimators	400
stk__lgb__learning_rate	0.01429
smote__k_neighbors	3

Número de árvores (400) em ambos *XGBoost* e *LightGBM* demonstra que uma quantidade moderada de estimadores é suficiente para capturar padrões complexos de evasão sem incorrer em sobreajuste.

Profundidade de 4 no *XGBoost* indica que árvores relativamente rasas conseguem modelar bem as interações entre atributos acadêmicos, demográficos e socioeconômicos. Isso favorece a generalização, evitando "memorização" de ruídos pontuais.

Learning rates modestos (~0.05 no XGB e ~0.014 no LGB) sugerem que o modelo aprendeu de forma gradual e estável, o que costuma resultar em maior robustez contra variações nos dados.

`k_neighbors = 3` no *KMeansSMOTE* revela que, ao gerar amostras sintéticas para a classe "Dropout", cada nova instância se baseou apenas em 3 vizinhos próximos dentro do mesmo cluster, preservando a coesão local e minimizando introdução de ruído.

Além disso, a simetria nos valores escolhidos para XGBoost e LightGBM (ambos com 400 estimadores) sugere que os dois algoritmos têm capacidade de modelagem comparável neste contexto, o que reforça a importância do stacking para explorar diferentes vieses entre eles. A escolha de profundidades moderadas e *learning rates* baixos também contribui para um compromisso entre tempo de treinamento e performance, garantindo que o pipeline seja escalável para bases maiores sem perda significativa de acurácia. Por fim, a configuração de `k_neighbors=3` no `KMeansSMOTE` não apenas equilibra as classes, mas também mantém a interpretabilidade das amostras geradas, pois cada sintético está intimamente relacionado a um pequeno grupo de observações reais, o que facilita a análise de casos típicos de risco de evasão.

Tabela IV
Desempenho em Validação Cruzada

Métrica	Valor
CV ROC-AUC	0.8892

A ROC-AUC (Área sob a Curva ROC) é uma métrica que quantifica a capacidade do modelo de discriminar corretamente entre as classes, plotando a taxa de verdadeiros positivos contra a taxa de falsos positivos para diferentes limiares de decisão. Na validação cruzada estratificada de três dobras utilizada durante a busca *Bayesiana* com 50 trials, o melhor trial alcançou um valor médio macro de 0,8892, refletindo excelente performance na distinção de cada uma das três classes. Em cenários multiclasse, essa AUC agregada via esquema one-vs-rest demonstra que o modelo não apenas favorece a classe majoritária, mas mantém poder discriminativo consistente para as classes minoritárias. Esse resultado sugere que o conjunto de hiperparâmetros encontrado e o pré-processamento foram eficazes em tornar o pipeline robusto e equilibrado.

Tabela X
Avaliação Final no Conjunto de Teste

Métrica	Valor
Test ROC-AUC	0.8750
Acurácia	0.771
Macro F1	0.720
Weighted F1	0.770

A ROC-AUC de 0.8750 confirma boa capacidade de discriminação em dados não vistos, ainda que ligeiramente inferior ao CV, apontando para uma leve variação mas sem indícios de overfit ting. A acurácia de 77.1% indica que, de cada 10 estudantes, cerca de 8 foram corretamente clas sificados em suas respectivas categorias, o que é relevante em um contexto real onde inter venções erradas têm custo social e financeiro. O Macro F1 de 0.720 mostra equilíbrio entre precisão e recall nas três classes, reforçando que o modelo não negligencia as minoritárias. O Weighted F1 de 0.770, ponderado pelo su porte de cada classe, confirma que a alta perfor mance geral não é puxada apenas pela classe majoritária *Graduate*, mas reflete coerência em todo o espectro de classes.

Tabela XI
Relatório de Classificação

Classe	Precision	Recall	F1-Score	Support
0 (Dropout)	0.82	0.73	0.77	284
1 (Enrolled)	0.51	0.57	0.54	159
2 (Graduate)	0.85	0.87	0.86	442
Macro avg	0.72	0.72	0.72	885
Weighted avg	0.78	0.77	0.77	885

Dropout (0): *Precision* de 0,82 indica que a maioria dos alunos classificados como "Dropout" de fato evadiu; recall de 0,73 significa que 73 % dos casos reais de evasão foram detectados.

Enrolled (1): *Precision* e *recall* baixo (0.51 e 0.57) mostram que classificar correta mente alunos em situação de matrícula ainda é o maior desafio, provavelmente pela semelhança de padrões de engajamento entre "Enrolled" e outras classes. O threshold otimizado ajuda, mas futuras iterações podem explorar variáveis tem porais de frequência.

Graduate (2): Desempenho médio (preci sion 0.85, recall 0.87) indica que o modelo cap tura bem os perfis de alunos com alta probabili dade de conclusão, refletindo padrões claros de sucesso acadêmico.

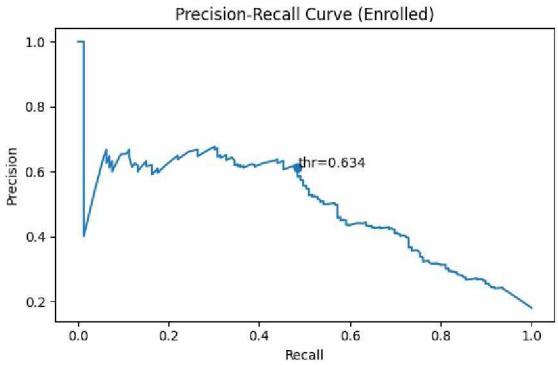

Figura 3. Curva Precision–Recall para a classe "Enrolled".
Fonte: O autor (2025)

A Figura 6 mostra a curva Precision–Recall para a classe 'Enrolled'. Observa-se que, a partir de *thresholds* muito baixos, a precisão é alta mas o recall é quase zero; à medida que aumen tamos o recall, a precisão oscila e depois de cresce, ilustrando o *trade-off* clássico entre sen sibilidade e especificidade positiva. O F1-score em cada ponto e selecionamos o threshold ≈ 0.634, que maximiza essa métrica.

B. *Matrizes de Confusão*

As matrizes de confusão oferecem uma visão aprofundada sobre como o modelo final (pipeline de pré-processamento + `KMeansSMOTE` + Stacking XGB+LGB + calibração isotônica + *threshold moving*) se comporta na classificação dos três possíveis desfechos estudantis: *Dropout* (0), *Enrolled* (1) e *Graduate* (2).

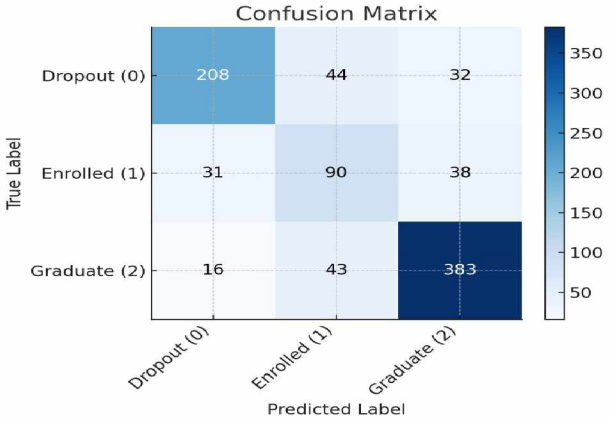

Figura 4. Matriz de Confusão.
Fonte: O autor (2025)

Classe *Dropout* (0) A taxa de recall dessa classe indica que uma parte significativa dos casos reais de evasão está sendo capturada pelo modelo. Já a precisão revela que, entre as previsões feitas como evasão, uma proporção considerável corresponde de fato a casos reais da classe. Esses dois indicadores sugerem que o modelo está sensível à identificação de estudantes evadidos, com relativamente poucos falsos positivos.

Classe *Enrolled* (1) Os valores de recall e precisão para essa classe são mais modestos, o que demonstra certa dificuldade do modelo em distinguir corretamente os estudantes que permanecem matriculados. Esse comportamento pode estar

relacionado à sobreposição de características entre essa classe e as demais, dificultando a separação clara nas decisões. A adoção de threshold específico para essa classe contribui para algum equilíbrio, mas ainda há margem para melhorias nas variáveis discriminativas.

Classe *Graduate* (2) Os índices de recall e precisão obtidos nessa classe estão entre os mais altos do modelo, sugerindo que ele é mais eficaz em reconhecer corretamente os estudantes que se formaram. Isso pode indicar que os padrões associados à graduação são mais marcantes em relação às demais classes, favorecendo a diferenciação pelo classificador.

Equilíbrio geral: A precisão e recall macro médios de 0,72 (ver Tabela XI) se confirmam nesta matriz: embora Graduate seja detectado com maior confiabilidade, as classes mi noritárias (*Dropout* e *Enrolled*) ainda apresen tam espaço para aperfeiçoamento. Custo de intervenção: Em cenários reais, fal sos positivos de Dropout (44) representam alunos que não precisariam de ação, mas recebe riam alertas de evasão; falsos negativos (32) representam evasões não previstas. Esse trade off de 73 % de recall com 82 % de precisão (conforme Tabela XI do Relatório de Classificação) pode ser considerado aceitável, mas políticas devem ponderar recursos disponíveis.

Trabalhos Futuros: Pretende-se investigar a incorporação de novas classes de variáveis, tais como indicadores comportamentais (logs de acesso, interações com tutores) e temporais (variações de desempenho ao longo dos semestres), com o objetivo de melhorar a separabilidade das classes. Além disso, será avaliado o uso de indicadores de engajamento precoce para otimizar métricas de recall sem comprometer a precisão do modelo.

C. *Discussões*

Os resultados demonstram que um pipeline integrado de pré-processamento, balanceamento com `KMeansSMOTE`, otimização de hiperparâmetros via Optuna e ensemble por stacking (`XGBoost` + `LightGBM`), complementado por calibragem isotônica e ajuste de limiares, alcançou desempenho preditivo robusto e interpretável na detecção de evasão escolar no ensino superior.

No conjunto de validação cruzada, o modelo obteve um valor médio de 0,8891, en quanto no conjunto de teste independente alcançou 0,8750, evidenciando consistência nos desempenhos e baixa variância entre treina mento e generalização.

A análise das predições revelou nu ances importantes sobre o comportamento do modelo. A classe *Graduate* (2) foi a mais bem identificada, com um recall de aproximada mente 87% e uma precisão de 85%, o que sug ere que os padrões que caracterizam o sucesso acadêmico são mais fáceis de aprender — possivelmente devido à consistência nas médias finais, progressão sem reprovações e alto engaja mento ao longo dos períodos. Este desempenho elevado indica que a ferramenta pode servir como um bom indicador de retenção e sucesso, auxiliando políticas de reconhecimento e está mulo. Por outro lado, a classe *Dropout* (0) também apresentou resultados encorajadores, com 73% de recall e 82% de precisão, demonstrando boa capacidade do modelo em detectar estudantes com risco de evasão. Este é um aspecto funda mental do trabalho, dado que o principal obje tivo é identificar precocemente os casos de risco e permitir intervenções pedagógicas ou psicos sociais. O observado — entre deixar de detectar 32 casos (falsos negativos) e acionar 44 alertas incorretos (falsos positivos).

A classe Enrolled (1), por sua vez, revelou ser a mais desafiadora, com métricas mais modestas (57% de recall e 51% de precisão). Isso pode ser atribuído ao fato de que os padrões comporta mentais dos alunos ainda matriculados tendem a sobrepor-se aos das demais classes — isto é, al guns alunos matriculados apresentam perfis semelhantes tanto aos que evadiram quanto aos que se graduaram. Este desafio sugere que atrib utos adicionais, particularmente de natureza temporal ou comportamental (por exemplo, fre quência em plataformas virtuais de aprendiza gem, número de acessos aos ambientes acadêmicos, participação em atividades ex tracurriculares ou interações com tutores), pode riam enriquecer significativamente a capacidade discriminativa do modelo.

O modelo proposto apresentou resultados satisfatórios e indicativos de aplicabilidade prática em instituições de ensino superior. No entanto, algumas observações merecem atenção para futuras investigações:

Aprimoramento da classe *Enrolled*: Como discutido, os alunos em curso tendem a exibir comportamentos ambíguos. Recomenda-se a incorporação de variáveis dinâmicas e longitudinais, como desempenho ao longo dos semestres, atraso na grade curricular, frequência em disciplinas e interações em plataformas digitais.

Análise de interpretabilidade: Apesar da boa performance preditiva, modelos ensemble como o stacking possuem menor interpretabilidade direta. A utilização de técnicas como *SHAP* ou *LIME* poderia ser empregada para explicar as decisões do modelo, o que é especialmente relevante em contextos educacionais, onde a transparência das decisões é fundamental.

Avaliação temporal: O presente estudo considerou os dados de forma agregada. Futuros trabalhos poderiam avaliar o modelo em um cenário de aprendizado contínuo ou janela temporal deslizante, testando sua robustez em diferentes coortes anuais de alunos.

Generalização para outras instituições: A base de dados analisada pertence a uma única instituição de ensino. Estudos futuros poderiam avaliar a capacidade de generalização do modelo em contextos distintos — universidades públicas, privadas, presenciais e a distância — a fim de validar sua eficácia em diferentes realidades acadêmicas.

Implantação prática e uso institucional: Finalmente, uma linha de pesquisa aplicada poderia se dedicar à construção de um sistema de alerta precoce baseado neste modelo, integrando-o aos sistemas acadêmicos das instituições para apoiar tutores, coordenadores e assistentes sociais na tomada de decisão.

Em suma, os resultados alcançados demonstram que o uso combinado de técnicas modernas de aprendizado de máquina, com atenção ao balanceamento dos dados, calibragem e customização dos *thresholds*, permite construir modelos preditivos eficazes para apoiar o combate à evasão escolar. Mais do que um exercício computacional, este trabalho contribui para a criação de ferramentas orientadas à equidade, permanência e sucesso estudantil, reforçando o papel da ciência de dados como aliada da educação.

A otimização de hiperparâmetros — conforme destacado no título — é realizada por meio do `OptunaSearchCV`, que executa uma busca bayesiana sobre distribuições específicas para os parâmetros dos classificadores `XGBoost`, `LightGBM`, `CatBoost` e do `SMOTE`. Esse processo de otimização bayesiana, orientado pelo critério de `roc_auc_ovr`, ajusta automaticamente as configurações de cada componente do ensemble antes do *stacking*, sendo peça-chave para alcançar os resultados de CV AUC 0,8891 e Test AUC 0,8750.

REFERÊNCIAS

[1] UNESCO. Relatório de Monitoramento Global da Educação 2024. Disponível em: https://www.unesco.org/pt/articles/251-milhoes-de-criancas-e-jovens-ainda-estao-fora-da-escola-apesar-de-decadas-de-progresso-relatorio. Acesso em: 15 de junho de 2025.

[2] FIRJAN SESI; PNUD. Combate à evasão no Ensino Médio: desafios e oportunidades. 2023. Disponível em: https://firjan.com.br/noticias/ifdm-2025-8AE4828D96AAF4370196AF8826102AF3-00.htm?&IdEditoriaPrincipal=4028818B46EEB3CD0146FD70E994340B. Acesso em: 15 de junho de 2025.

[3] Comissão Europeia. Panorama do abandono escolar precoce na União Europeia. Temático: Jovens que abandonam

precocemente a escola (dados de 2024). Disponível em: https://ec.europa.eu/eurostat/statistics-explained/index.php?title=Early_leavers_from_education_and_training. Acesso em: 15 de junho de 2025.
[4] IPEC; UNICEF. Evasão escolar no Brasil: 2 milhões de crianças e adolescentes fora da escola em 2022. Disponível em: https://pt.wikipedia.org/wiki/Evas%C3%A3o_escolar. Acesso em: 15 de junho de 2025.
[5] Realinho, V.; Martins, M. V.; Machado, J.; Baptista, L. (2021). Predict Students' Dropout and Academic Success [Dataset]. UCI Machine Learning Repository. Disponível em: https://archive.ics.uci.edu/dataset/697/predict+students+dropout+and+academic+success. Acesso em: 15 de junho de 2025.
[6] The Pandas Development Team. Python pandas 1.5 — documentação oficial. Disponível em: https://pandas.pydata.org/. Acesso em: 15 de junho de 2025.
[7] The Pandas Development Team. "10 Minutes to pandas" — guia rápido do pandas. Disponível em: https://pandas.pydata.org/docs/user_guide/10min.html. Acesso em: 15 de junho de 2025.
[8] Scikit-learn developers. OneHotEncoder — pré-processamento de variáveis categóricas. Disponível em: https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.OneHotEncoder.html. Acesso em: 15 de junho de 2025.
[9] Felix Last; Georgios Douzas; Fernando Bacao. imbalanced-learn KMeansSMOTE — oversampling combinado com clustering. Disponível em: https://imbalanced-learn.org/stable/references/generated/imblearn.over_sampling.KMeansSMOTE.html. Acesso em: 15 de junho de 2025.
[10] Chen, T.; Guestrin, C. XGBoost: A Scalable Tree Boosting System. KDD '16, 785–794, 2016. Disponível em: https://arxiv.org/abs/1603.02754. Acesso em: 15 de junho de 2025.
[11] Ke, G. et al. LightGBM: A Highly Efficient Gradient Boosting Decision Tree. NeurIPS '17, 3146–3154, 2017. Disponível em: https://papers.nips.cc/paper/6907-lightgbm-a-highly-efficient-gradient-boosting-decision-tree.pdf. Acesso em: 15 de junho de 2025.
[12] Akiba, T.; Sano, S.; Yanase, T.; Ohta, T.; Koyama, M. Optuna: A Next-Generation Hyperparameter Optimization Framework. KDD '19, 2623–2631, 2019. Disponível em: https://arxiv.org/abs/1907.10902. Acesso em: 15 de junho de 2025.
[13] Saccaro, A.; França, M. T. A.; Jacinto, P. A.
Fatores Associados à Evasão no Ensino Superior Brasileiro: um estudo de análise de sobrevivência.
Estudos Econômicos, São Paulo, vol. 49, n. 2, p. 337–373, abr.–jun. 2019. Disponível em: https://www.scielo.br/j/ee/a/9YxHxWkk6Dzy35CpgmxXbPt/?lang=pt. Acesso em: 15 de junho de 2025.
[14] Nierotka, R. L.; Salata, A.; Klitzke Martins, M. "Fatores Associados à Evasão no Ensino Superior: um estudo de sobrevivência". *Cadernos de Pesquisa*, vol. 53, n.º 1, 2023. Disponível em: https://educa.fcc.org.br/scielo.php?pid=S0100-15742023000100302&script=sci_arttext. Acesso em: 15 de junho de 2025.
[15] Al-Zawqari, A.; Peumans, D.; Vandersteen, G. A flexible feature selection approach for predicting students' academic performance. *Computers and Education: Artificial Intelligence*, 2022. Disponível em: https://www.sciencedirect.com/science/article/pii/S2666920X22000583. Acesso em: 15 de junho de 2025.
[16] Alakh Sethi. One Hot Encoding vs Label Encoding in Machine Learning. Analytics Vidhya. Disponível em: https://www.analyticsvidhya.com/blog/2020/03/one-hot-encoding-vs-label-encoding-using-scikit-learn/. Acesso em: 15 de junho de 2025.
[17] Scikit-learn Developers. OneHotEncoder — scikit-learn 1.6.1 documentation. Disponível em: https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.OneHotEncoder.html. Acesso em: 15 de junho de 2025.
[18] Chawla, N. V.; Bowyer, K. W.; Hall, L. O.; Kegelmeyer, W. P. SMOTE: Synthetic Minority Over-sampling Technique. Journal of Artificial Intelligence Research, 16, 321–357, 2002. Disponível em: https://www.jair.org/index.php/jair/article/view/10302. Acesso em: 15 de junho de 2025.
[19] Last, F.; Douzas, G.; Bacao, F. Oversampling for Imbalanced Learning Based on K-Means and SMOTE. arXiv preprint arXiv:1711.00837. Disponível em: https://arxiv.org/abs/1711.00837. Acesso em: 15 de junho de 2025.
[20] Hairani, H.; Saputro, K. E.; Fadli, S. K-means-SMOTE untuk menangani ketidakseimbangan kelas dalam klasifikasi penyakit diabetes com C4.5, SVM e naive Bayes. Jurnal Teknologi dan Sistem Komputer, 8(2), 89–93, 2020. Disponível em: https://doi.org/10.14710/jtsiskom.8.2.2020.89-93. Acesso em: 15 de junho de 2025.
[21] LAST, Felix. *KMeansSMOTE – biblioteca Python para balanceamento de dados*. Disponível em: https://pypi.org/project/kmeans-smote/. Acesso em: 15 jun. 2025.
[22] Lemaitre, G.; Nogueira, F.; Aridas, C. K. Imbalanced-learn: A Python Toolbox to Tackle the Curse of Imbalanced Datasets in Machine Learning. Journal of Machine Learning Research, 18(17), 1–5, 2017. Disponível em: https://arxiv.org/abs/1609.06570. Acesso em: 15 de junho de 2025.
[23] Scikit-learn Developers. CalibratedClassifierCV — scikit-learn 1.6.1 documentation. Disponível em: https://scikit-learn.org/stable/modules/generated/sklearn.calibration.CalibratedClassifierCV.html. Acesso em: 15 de junho de 2025.
[24] Scikit-learn Developers. Probability calibration — scikit-learn 1.6.1 User Guide. Disponível em: https://scikit-learn.org/stable/modules/calibration.html. Acesso em: 15 de junho de 2025.
[25] Scikit-learn Developers. Precision-Recall Curve — scikit-learn 1.6.1 documentation. Disponível em: https://scikit-learn.org/stable/modules/generated/sklearn.metrics.precision_recall_curve.html. Acesso em: 15 de junho de 2025.
[26] (Autor não identificado). Optimizing F1 Score with Threshold Tuning — GPTutorPro. Disponível em: https://gpttutorpro.com/f1-machine-learning-essentials-optimizing-f1-score-with-threshold-tuning/. Acesso em: 15 de junho de 2025.
[27] Brownlee, J. Threshold-Moving for Imbalanced Classification — Machine Learning Mastery. Disponível em: https://www.machinelearningmastery.com/threshold-moving-for-imbalanced-classification/. Acesso em: 15 de junho de 2025.
[28] W Brett Kennedy . Achieve Better Classification Results with ClassificationThresholdTuner — Medium. Disponível em: https://medium.com/data-science/achieve-better-classification-results-with-classificationthresholdtuner-39c5d454637e. Acesso em: 15 de junho de 2025.
[29] Scikit-learn Developers. ROC AUC — scikit-learn 1.6.1 User Guide: Model Evaluation. Disponível em: https://scikit-learn.org/stable/modules/model_evaluation.html. Acesso em: 15 de junho de 2025.
[30] Scikit-learn Developers. classification_report — scikit-learn 1.6.1 documentation. Disponível em: https://scikit-learn.org/stable/modules/generated/sklearn.metrics.classification_report.html. Acesso em: 15 de junho de 2025.
[31] Scikit-learn Developers. confusion_matrix — scikit-learn 1.6.1 documentation. Disponível em: https://scikit-learn.org/stable/modules/generated/sklearn.metrics.confusion_matrix.html. Acesso em: 15 de junho de 2025.
[32] Chen, T.; Guestrin, C. XGBoost: A Scalable Tree Boosting System. ArXiv:1603.02754. Disponível em: https://arxiv.org/abs/1603.02754. Acesso em: 15 de junho de 2025.

[33] Ke, G. et al. LightGBM: A Highly Efficient Gradient Boosting Decision Tree. NeurIPS 2017. Disponível em: https://papers.nips.cc/paper_files/paper/2017/hash/6449f44a102fde848669bdd9eb6b76fa-Abstract.html. Acesso em: 15 de junho de 2025.
[34] Sagi, O.; Rokach, L. Ensemble learning: A survey. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 2018. Disponível em: https://doi.org/10.1002/widm.1249. Acesso em: 15 de junho de 2025.