Projeto Prático de Aprendizado de Máquina com *Streamlit*
2COP507 - Reconhecimento de Padrões
Prof. Dr. Bruno B. Zarpelão / Prof. Dr. Sylvio Barbon Jr.

Eduardo Fernandes Bueno[1]

[1]Instituto de Computação - Universidade Estadual de Londrina (UEL)
Rod. Celso Garcia Cid, S/N - Campus Universitário, Londrina - PR, 86057-970

bueno.fernandes.eduardo@uel.br
github.com/EddieFerb/streamlit-evasao-inep
youtube.com/@EddieFerb

Abstract. *This work presents a Streamlit-based application designed to predict dropout rates in undergraduate programs offered by Brazilian Higher Education Institutions, using official and public INEP microdata (2009–2024). The system integrates a complete Machine Learning pipeline. The main algorithm was* ***Random Forest****, which achieved* $R^2 \approx 0.9999$ *(regression), MSE near zero, accuracy* $\approx 99.97\%$ *(binary classification). The interface enables interactive hyperparameter tuning and displays metrics, charts, and individual predictions.*

Resumo. *Este trabalho desenvolveu aplicação em* ***Streamlit*** *que prevê a taxa de evasão em cursos de graduação de Instituições de Ensino Superior, a partir de microdados oficiais e públicos do INEP (2009-2024). O sistema integra pipeline completo de* ***Aprendizado de Máquina****. O algoritmo principal foi o* ***Random Forest****, com desempenho* $R^2 \approx 0.9999$ *(regressão), MSE próximo de zero acurácia* $\approx 99.97\%$ *(classificação binária). A interface permite ajuste interativo de hiperparâmetros e exibe métricas, gráficos e previsões individuais.*

1. Contexto & Modelo de Aprendizado de Máquina: *Random Forest*

O problema tratado neste projeto é a predição da *taxa de evasão* em cursos de graduação das Instituições de Ensino Superior (IES) brasileiras, a partir de dados oficiais e públicos do INEP/MEC no período de 2009 a 2024. Para tal, foi escolhido o algoritmo ***Random Forest*** (RF) como modelo de *Aprendizado de Máquina* (AM), com `RandomForestRegressor` para prever taxa de evasão. Justificativas principais:

Capacidade de modelar relações não lineares: a evasão de alunos tem combinações complexas de variáveis de oferta e demanda (cursos, vagas, matriculados, concluintes);

Robustez à ruído e heterogeneidade: a base combina cursos de diferentes áreas, modalidades (presencial/EaD) e redes (pública/privada). O *ensemble* de árvores reduz a variância em relação a uma árvore de decisão isolada, gera modelo mais robusto à *outliers* e à heterogeneidade entre cursos;

Histórico positivo em Mineração de Dados Educacionais (MDE): estudos sobre predição de evasão em ensino superior reportam bom desempenho do RF em comparação com outros algoritmos como regressão, SVM, RNAs etc. [Andrade-Girón et al. 2023]

ensinam que RF é o modelo mais usado, presente 21,73% dos estudos com 99% de precisão na previsão de evasão de alunos e os [Silvestri et al. 2025] obtiveram desempenho de 78% de acurácia com RF na base de dados completa em suas MDE.

2. Configuração do modelo e hiperparâmetros

O alvo de regressão foi a variável contínua `taxa_evasao`, calculada no *pipeline* de processamento de dados a partir de ingressantes, matriculados e concluintes, gerando um valor entre 0 e 1 para cada par (curso, ano). Na etapa de avaliação classificatória, a taxa de evasão contínua binarizada em tempo de execução, ou seja, cursos com taxa_evasao acima de um limiar (padrão 0,5) são rotulados como (1) '*evasão alta*' e os demais como (0) '*evasão baixa*'. Esse limiar também é utilizado para binarizar as predições do modelo, gerando a matriz de confusão e as métricas de classificação, abordagens semelhantes às usadas por trabalhos (inter)nacionais, p. ex. [Manhães et al. 2011] e [Deb 2024].

Para evitar o vazamento de informação *leakage*, problema identificado em versões anteriores do projeto, quando taxas derivadas eram usadas simultaneamente como atributos e alvo, o *pipeline* final exclui as métricas derivadas (`taxa_ingresso`, `taxa_conclusao`, `taxa_evasao`) do conjunto de entrada. As variáveis utilizadas como *features* foram: `vagas_totais`, `inscritos_totais`, `ingressantes`, `matriculados` e `concluintes`, conforme a abordagem de pré-processamento discutida por [Deb 2024] e por estudos europeus recentes de predição de evasão em IES.

O pré-processamento foi realizado em *Python* com a biblioteca *Pandas*, gerando conjunto tabular consolidado pronto para ser consumido pelos modelos de regressão e classificação.

Esse formato garante que todas as etapas de transformação sejam aplicadas de forma consistente em treino, validação e teste, reduz o risco de *data leakage* manual e facilita a serialização do modelo final em arquivo `.pkl` para consumo pela aplicação *Streamlit*.

Na **estratégia de validação**, optou-se por um *split temporal* entre treino e teste, por ser um problema temporal (series anuais de 2009 a 2024): **Treino:** anos de 2009 a 2018; **Teste:** anos de 2019 a 2024. Essa estratégia segue recomendações presentes em estudos de predição de evasão que utilizam divisão cronológica entre treino e teste, evitando o uso de dados futuros no treino do modelo de acordo com [Andrade-Girón et al. 2023].

O modelo aprende apenas com dados históricos anteriores (treino, 2009 - 2018) e é avaliado em anos futuros (teste, 2019 - 2024), simulando cenário realista de previsão , evita o uso de informações do "futuro" no treino. No conjunto de treino o modelo foi ajustado diretamente com uma divisão temporal simples. Estratégia análoga usada por [Deb 2024] para excluir centenas de alunos matriculados com status acadêmico pouco claro, para se concentrar em classificação binária.

Como **hiperparâmetros e justificativas**, o modelo utilizado neste trabalho foi o `RandomForestRegressor`, da biblioteca *scikit-learn*, configurado no padrão, exceto pelo número de estimadores, definido como `n_estimators = 100`. Esse valor fornece um conjunto suficientemente grande de árvores para capturar relações não lineares sem impor custo computacional elevado. O `max_depth = None` permite que as árvores cresçam até o critério natural de pureza, e `min_samples_leaf = 1` que preserva a granularidade das divisões. Alcançou desempenho superior à *Regressão Linear* no conjunto de teste temporal (2019 - 2024). Na etapa classificatória, foi uti-

lizado `RandomForestRegressor` para binarizar a partir do limiar 0,50, gerando duas classes (”alta evasão” e ”baixa evasão”) com os hiperparâmetros padrão, alcançando acurácia de 0.9997, precisão de 0.9999 e F1-score de 0.9998, indicando desempenho robusto mesmo sem otimização formal de hiperparâmetros.

3. *Streamlit*: App Evasão

A aplicação foi estruturada em quatro abas: (i) **Sobre**, com síntese do problema e instruções de uso; (ii) **Predição Individual**, que permite ao usuário informar manualmente as variáveis estruturais de um curso e obter a taxa de evasão predita com interpretação textual; (iii) **Avaliação do Modelo**, em que é possível alternar entre o modelo de *pipeline* fixo e um modelo customizado treinado em tempo de execução, ajustar hiperparâmetros do RF e visualizar métricas (MSE, R², acurácia, precisão, recall, F1) e gráficos (matriz de confusão, dispersão real vs. predito, importância de atributos); e (iv) **Upload CSV**, que realiza predições em lote e disponibiliza os resultados para *download*.

4. Resultados e Discussão

Desempenho em Regressão para a taxa de evasão (`RandomForestRegressor`) foi avaliado no conjunto de teste (2019 - 2024), obteve: MSE (Erro Quadrático Médio): aproximadamente 0,0000; R^2 (coeficiente de determinação): 0,9999. Esses valores indicam que o erro médio ao estimar a taxa de evasão na escala [0, 1] é praticamente nulo e que o modelo explica cerca de 99,99% da variância no período de teste. Logo, as curvas de evasão real e predita são quase sobrepostas para a maioria dos cursos e anos, o que reforça a forte capacidade preditiva das variáveis estruturais selecionadas.

No **desempenho e classificação binária**, a taxa de evasão contínua foi convertida em duas classes a partir do limiar padrão de 0,50, classifica como “alta evasão” ($\geq 0{,}50$) ou “baixa evasão” ($< 0{,}50$), apresentou os resultados no conjunto de teste (2019 - 2024): **Acurácia:** 0,9997; **Precisão:** 0,9999; **Revocação (Recall):** 0,9996; **F1-score:** 0,9998.

A matriz de confusão [[44588, 5], [39, 99058]] composta por 44.588 cursos de baixa evasão foram corretamente classificados (TN) e 99.058 cursos de alta evasão foram identificados corretamente (TP) como classe 1, com 5 falsos positivos e 39 falsos negativos.

Os resultados expressam que o modelo consegue identificar com alta precisão cursos com evasão elevada. Isso coaduna com trabalhos de [Manhães et al. 2011] onde constatou-se que RF tem acurácia entre 75% a 80%, e, nesse sentido [de Almeida Teodoro and Kappel 2020] usou dados INEP e obteve 80% de taxa de acerto nas previsões de evasão com uso de RF.

Para a **integração com a aplicação *Streamlit***, o modelo RF foi serializado em arquivo `.pkl` e integrado a um aplicação web desenvolvida em *Streamlit*. Nessa interface, o usuário pode: **a**justar valores das variáveis estruturais (vagas, inscritos, ingressantes, matriculados, concluintes, número de cursos, ano); **c**alcular a taxa de evasão predita para um curso específico; **v**isualizar a probabilidade de evasão ou permanência, bem como a interpretação do resultado na tela *”Predição Individual de Taxa de Evasão”*. O painel também permite testar diferentes configurações de hiperparâmetros, por exemplo, número de árvores e profundidade máxima, de forma interativa, reforçando o caráter didático da aplicação em relação ao conteúdo da disciplina de Reconhecimento de Padrões.

5. Possíveis melhorias

Redução de possíveis traços de *overfitting*: existe uma relação quase determinística entre as contagens (ingressantes, matriculados, concluintes) e a evasão, o que facilita que a RF "aprenda a fórmula" implícita. Isso explica, em parte, os valores R^2 próximos de 1. Explorar configurações mais restritivas, p. ex.: aumentar `min_samples_leaf` ≥ 5 ou 10; limitar `max_depth` a valores entre 5 e 10; reduzir `n_estimators` e comparar com modelos *Gradient Boosting*, *XGBoost*, *LightGBM* e *CatBoost*, para avaliar ganhos de generalização, como pode ser observados nas pesquisas de [Deb 2024] reveleram que *Logistic Regression* obteve 91,4% de acurácia e *Randm Forest* obteve acurácia de %85,4.

Aprimorar os modelos `feature_based` e `fine_tuning`: na etapa de comparação de modelos, foram executados três *scripts*: `randomforest.py`, `fine_tuning.py` e `feature_based.py`. Eles apresentaram desempenho elevado; mas, ao aplicar o *fine-tuning* e *feature-based* aos **subconjuntos** *final_medicina*, *final_direito*, *final_eng_civil* e *final_administracao* deram erros de amostragem, apenas 1 linha por subconjunto e ausência da `taxa_evasao` em *final_ingressantes*.

Reestruturar a geração desses subconjuntos para que cada um contenha uma série temporal longa, p. ex., vários anos por curso e a coluna `taxa_evasao` consolidada. Treinar e ajustar modelos específicos por área (Administração, Direito, etc.) na versão *feature-based* e na *fine-tuning*; comparar se modelos específicos superam o modelo global em termos de MSE, R^2 e F1-score; incorporar os modelos especializados na *app Streamlit*.

Criar novas variáveis derivadas com maior significado educacional, *v. g.*, idade do aluno, densidade regional de oferta, questões socioeconômicas da IES e testar validações (*rolling forecasting origin*, *nested cross-validation*) [Andrade-Girón et al. 2023] em blocos temporais para reduzir risco de superestimar o desempenho.

References

Andrade-Girón, D., Sandivar-Rosas, J., Marín-Rodriguez, W., Susanibar-Ramirez, E., Toro-Dextre, E., Ausejo-Sanchez, J., Villarreal-Torres, H., and Angeles-Morales, J. (2023). Predicting student dropout based on machine learning and deep learning: A systematic review. *EAI Endorsed Transactions on Scalable Information Systems*, 10(5).

de Almeida Teodoro, L. and Kappel, M. A. A. (2020). Aplicação de técnicas de aprendizado de máquina para predição de risco de evasão escolar em instituições públicas de ensino superior no brasil. *Revista Brasileira de Informática na Educação*, 28:838–863.

Deb, A. (2024). Unlocking student dropout patterns: A machine learning-based data analysis. *International Journal of Innovative Research in Technology (IJIRT)*, 11:555–560.

Manhães, L. M. B., Da Cruz, S. M. S., Costa, R. J. M., Zavaleta, J., and Zimbrão, G. (2011). Previsão de estudantes com risco de evasão utilizando técnicas de mineração de dados. In *Brazilian symposium on computers in education (simpósio brasileiro de informática na educação-sbie)*, volume 1.

Silvestri, F., de Souza, V. F., and Vieira, A. (2025). Mineração de dados educacionais: uma análise sobre os preditores da evasão no ensino superior. *Revista Educar Mais*, 9:1–23.