Relatório Final - Aplicação de Inteligência Artificial para Previsão de Evasão no Ensino Superior

Antônio Henrique Carlos Clístenes Erasmo Alves
Everton Barbosa Jônatas Henrique Pedro Bullé

24 de novembro de 2025

Resumo

A evasão no ensino superior é um fenômeno complexo que gera impactos acadêmicos, sociais e orçamentários significativos. Este trabalho propõe uma abordagem baseada em Inteligência Artificial para identificar estudantes em risco de abandono, utilizando uma base de dados com 4.424 registros. Foram desenvolvidos e comparados modelos baseados em *Multi-Layer Perceptron* (MLP) e *Random Forest* em dois cenários distintos: classificação multiclasse (três estados) e binária (focada no desfecho final). A metodologia incluiu uma etapa rigorosa de pré-processamento, com normalização estatística via *StandardScaler*, e validação cruzada com 30 execuções independentes. Optou-se por manter o desbalanceamento original dos dados para avaliar a robustez dos classificadores em cenários reais. Os resultados demonstraram a superioridade do *Random Forest*, alcançando 91,04% de acurácia no cenário binário com alta estabilidade. Como produto final, desenvolveu-se uma aplicação *web* em *Streamlit*, permitindo a utilização prática dos modelos para suporte à tomada de decisão institucional.

1 Introdução e Motivação

O abandono escolar no nível superior representa um dos desafios mais críticos para a gestão educacional contemporânea. A evasão não apenas reflete a perda de investimento público e privado, mas também sinaliza falhas no processo de integração e suporte ao estudante. Antecipar quais alunos possuem maior probabilidade de abandonar o curso é fundamental para a implementação de políticas de retenção proativas, permitindo intervenções pedagógicas antes que o desligamento ocorra.

A motivação deste trabalho reside na capacidade dos algoritmos de Aprendizado de Máquina (*Machine Learning*) de processar grandes volumes de dados heterogêneos — como notas, dados demográficos e socioeconômicos — para encontrar padrões não triviais que precedem a evasão. Ao contrário de métodos estatísticos tradicionais, modelos como Redes Neurais e Florestas Aleatórias podem capturar interações complexas e não-lineares entre as variáveis.

O objetivo deste projeto é duplo:

1. **Modelagem Preditiva:** Comparar o desempenho de duas famílias distintas de algoritmos (Redes Neurais vs. Árvores de Decisão) na tarefa de predição de status acadêmico, avaliando precisão e estabilidade.

2. **Aplicação Prática:** Desenvolver uma ferramenta de software interativa que abstraia a complexidade matemática, tornando os modelos acessíveis a gestores educacionais.

2 Base de Dados

2.1 Origem e Descrição

Os dados utilizados provêm do *Higher Education Students Performance Dataset*. O conjunto total contém **4.424 registros** de estudantes, abrangendo variáveis de diferentes naturezas:

- **Dados Demográficos:** Idade na matrícula, gênero, estado civil e nacionalidade.
- **Dados Socioeconômicos:** Escolaridade e ocupação dos pais, condição de deslocado, status de pagamento de mensalidades e bolsa de estudos.
- **Dados Acadêmicos:** Unidades curriculares aprovadas, notas do 1º e 2º semestres e unidades creditadas.
- **Dados Macroeconômicos:** Taxa de desemprego, taxa de inflação e PIB.

Foi adicionada uma coluna `ID` sequencial ao *dataset* para viabilizar a funcionalidade de busca individual na aplicação final.

2.2 Distribuição das Classes e Desbalanceamento

A variável alvo (*Target*) apresenta três categorias. A análise da distribuição revelou um desbalanceamento significativo na base de dados original:

- **Graduate (Graduado):** 2.209 alunos (Classe majoritária).
- **Dropout (Evadido):** 1.421 alunos.
- **Enrolled (Matriculado):** 794 alunos (Classe minoritária).

Neste trabalho, optou-se deliberadamente por **não aplicar técnicas de balanceamento artificial** (como SMOTE). O objetivo desta decisão metodológica é avaliar o desempenho dos modelos frente à distribuição natural e realista do problema, evitando a introdução de ruídos sintéticos.

2.3 Análise Exploratória e Correlação

A análise exploratória permitiu identificar quais fatores possuem maior peso na decisão do modelo. O gráfico de importância abaixo (Figura 1) evidencia as relações lineares entre as variáveis numéricas e o alvo.

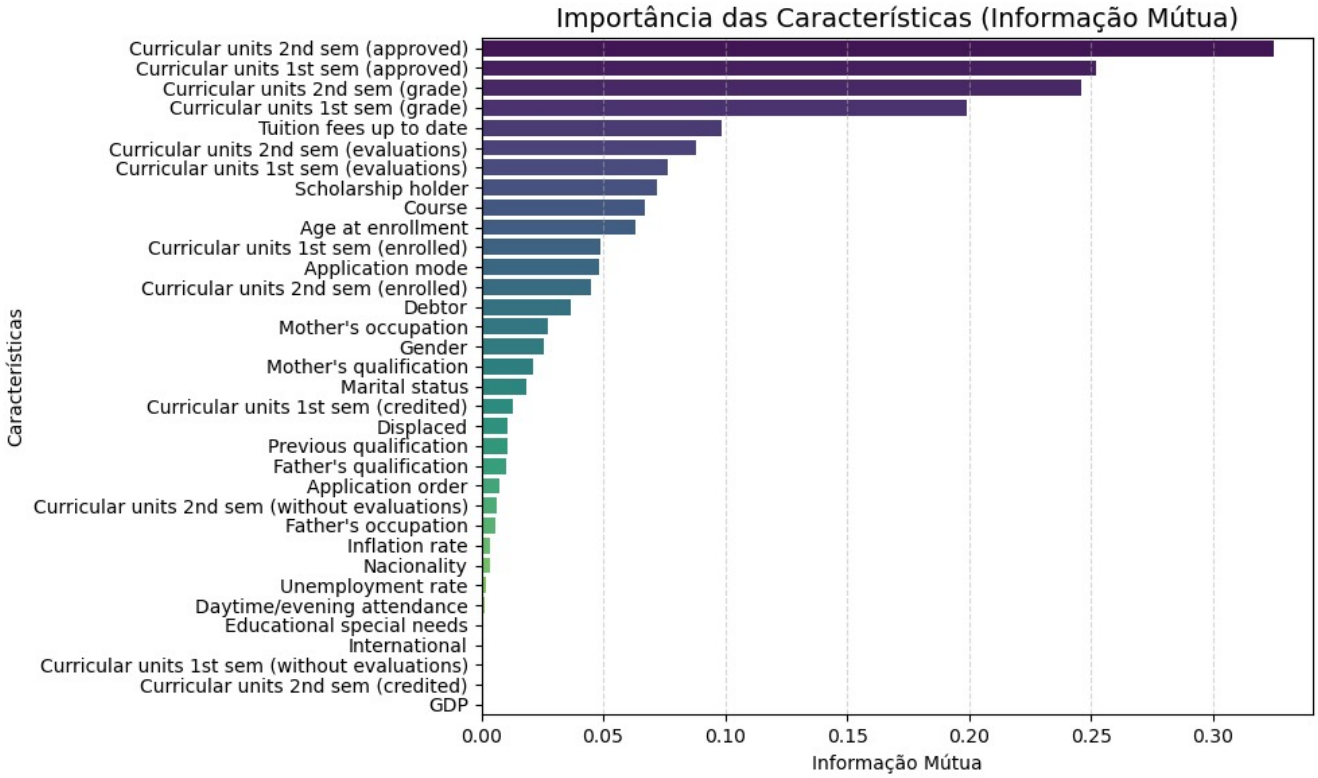


Figura 1: Gráfico das correlações entre variáveis. Nota-se a forte influência positiva do desempenho acadêmico e negativa da idade/débitos.

Conforme observado na Figura 1, as variáveis de desempenho acadêmico imediato apresentaram as correlações positivas mais fortes com a permanência do aluno, destacando-se as *Unidades Curriculares Aprovadas no 2º Semestre* (0.62) e a *Nota do 2º Semestre* (0.57). A situação financeira também se mostrou um preditor chave, evidenciada pela correlação da variável *Tuition fees up to date* (Mensalidades em dia) com índice de 0.41.

Em contrapartida, variáveis como a Idade na matrícula (-0.24) e a condição de Devedor (-0.24) apresentaram correlação negativa, indicando serem fatores de risco para a evasão.

2.4 Pré-processamento

2.4.1 Definição de Cenários

- **Cenário Multiclasse:** Mantém a integridade dos dados (4.424 amostras). Útil para monitorar alunos estagnados (*Enrolled*).
- **Cenário Binário:** Remove a classe *Enrolled*, resultando em 3.630 amostras. Foca na distinção crítica entre sucesso (*Graduate*) e fracasso (*Dropout*).

2.4.2 Normalização (StandardScaler)

Aplicou-se a padronização estatística nas variáveis numéricas utilizando a classe `StandardScaler` da biblioteca *Scikit-learn*. Esta etapa é fundamental para garantir que variáveis com diferentes magnitudes (ex: Idade vs. PIB) contribuam equitativamente para o modelo, especialmente no caso da rede neural.

O escalador foi configurado com seus parâmetros padrão:

- `with_mean = True`: Determina que a média (μ) das amostras deve ser calculada e subtraída (centralização em zero).
- `with_std = True`: Indica que o desvio padrão (σ) deve ser calculado para dividir os dados, escalonando a variância para a unidade.

A transformação matemática aplicada a cada amostra x segue a fórmula do Z-score:

$$z = \frac{x - \mu}{\sigma} \tag{1}$$

Onde:

x = valor original

μ = média da feature no conjunto de treino

σ = desvio padrão da feature no conjunto de treino

z = valor padronizado

3 Metodologia

3.1 Modelos Selecionados

A seleção dos algoritmos visou contrapor duas abordagens distintas de aprendizado supervisionado:

1. **Multi-Layer Perceptron (MLP):** Uma rede neural artificial do tipo *feedforward*. Este modelo foi escolhido por sua propriedade de aproximador universal de funções, capaz de mapear relações não-lineares complexas entre os atributos de entrada e a classe de saída, através do ajuste de pesos via *backpropagation*.

2. **Random Forest Classifier:** Um método de *ensemble* baseado na técnica de *Bagging*. O modelo constrói múltiplas árvores de decisão durante o treinamento e produz a classe que é a moda das classes das árvores individuais. A justificativa para seu uso reside na sua alta capacidade de generalização, resistência ao *overfitting* e eficiência em dados tabulares.

3.2 Protocolo Experimental

Para assegurar a validade estatística e mitigar o viés estocástico, estabeleceu-se o seguinte protocolo rigoroso:

- **30 Execuções Independentes:** O experimento completo foi repetido 30 vezes. Em cada iteração, uma nova semente aleatória (*seed*) foi gerada no `numpy` e `random`, garantindo embaralhamentos e divisões de dados únicos a cada rodada.

- **Divisão Estratificada:** Os dados foram divididos em 80% para treinamento e 20% para teste, utilizando amostragem estratificada para preservar a proporção original das classes.

- **Otimização Aninhada (GridSearchCV):** Em **cada uma das 30 execuções**, aplicou-se um *GridSearchCV* com validação cruzada ($k = 5$) nos dados de treino. Isso assegura que os hiperparâmetros sejam otimizados especificamente para a distribuição de treino daquela iteração.

4 Resultados

Os resultados apresentados refletem a média das 30 execuções independentes.

4.1 Desempenho Geral

A Tabela 1 apresenta a acurácia média. O **Random Forest** superou a MLP em ambos os cenários, demonstrando também menor desvio padrão.

Tabela 1: Comparação de Acurácia Média e Desvio Padrão (30 Execuções)

Cenário	**Modelo**	**Acurácia Média**	**Desvio Padrão**
Multiclasse	MLP	0.7473 (74,73%)	0.0093
Multiclasse	Random Forest	**0.7762 (77,62%)**	**0.0047**
Binário	MLP	0.9050 (90,50%)	0.0061
Binário	Random Forest	**0.9104 (91,04%)**	**0.0033**

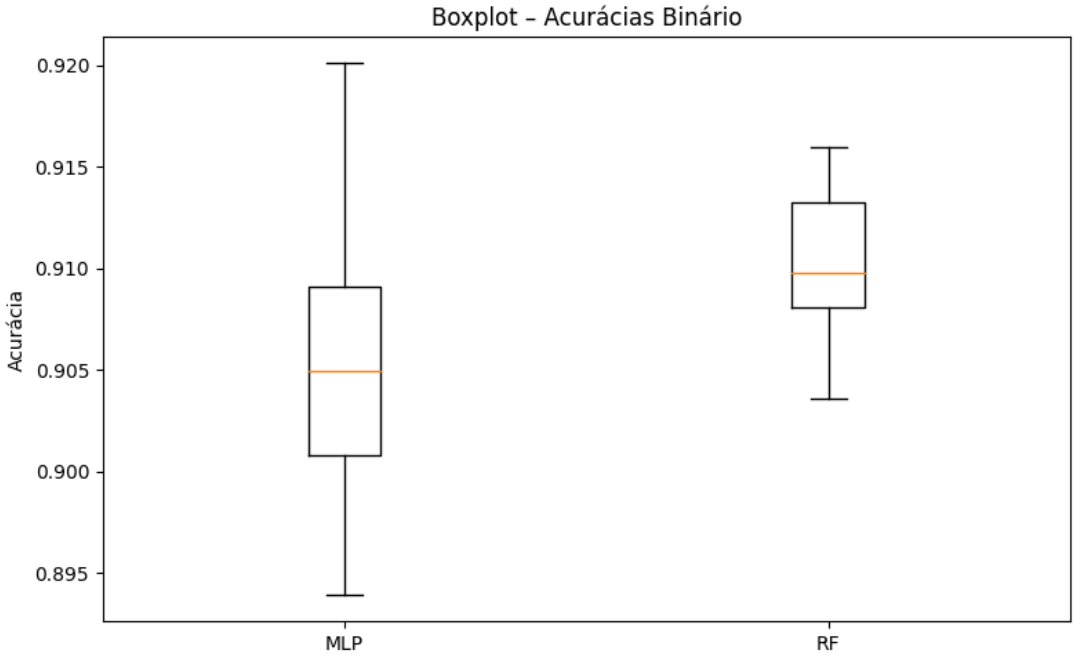
4.2 Análise de Hiperparâmetros

A análise de frequência dos parâmetros selecionados pelo *GridSearch* nas 30 execuções revelou a convergência dos modelos:

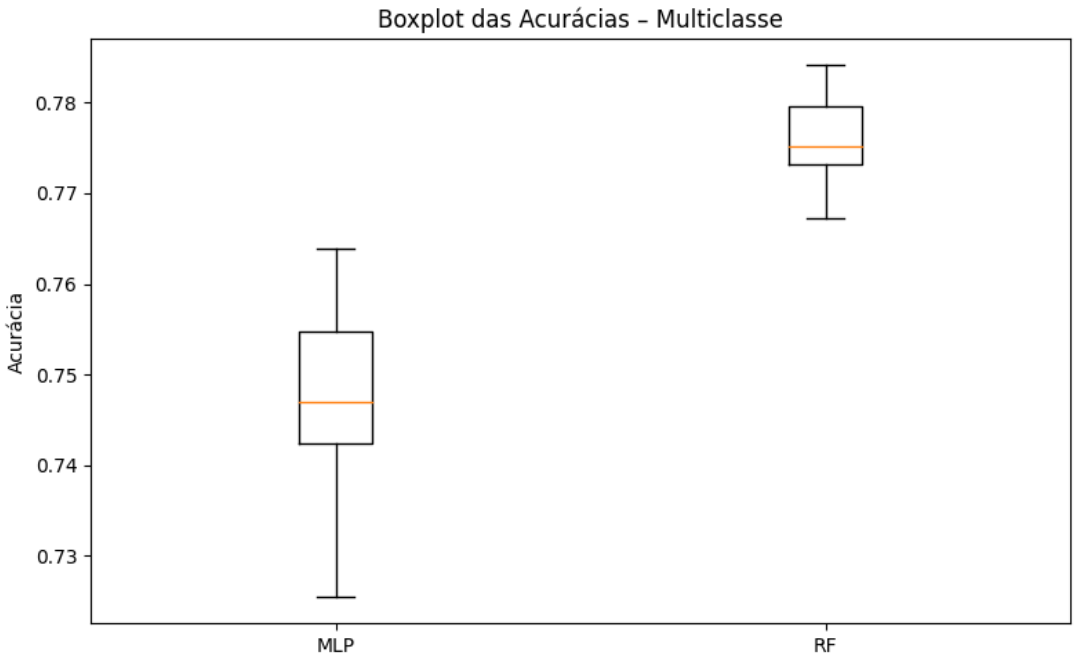
- **MLP (Binário):** Em 27 das 30 execuções, a configuração vencedora foi: `activation='logisti` `hidden_layer_sizes=(10,)` e `learning_rate_init=0.001`.
- **Random Forest (Multiclasse):** O modelo tendeu a configurações mais robustas para lidar com a complexidade das três classes, selecionando frequentemente `n_estimators=200` e `max_depth=20`.

4.3 Estabilidade dos Modelos (Boxplots)

A Figura 2 ilustra a distribuição das acurácias ao longo das 30 execuções. A visualização confirma a robustez do Random Forest, representado por caixas mais compactas (menor variância).



(a) Cenário Multiclasse



(b) Cenário Binário

Figura 2: Boxplots comparativos da acurácia (30 execuções). O Random Forest apresenta menor dispersão em ambos os cenários.

4.4 Análise Detalhada das Matrizes de Confusão

As matrizes de confusão, apresentadas na Figura 3, permitem uma inspeção granular dos erros cometidos pelos classificadores, cruzando as classes reais (linhas) com as classes

preditas (colunas).

4.4.1 Cenário Multiclasse

No cenário com três classes, observa-se que a diagonal principal — que representa os acertos — é predominante para as classes *Graduate* e *Dropout*. No entanto, a classe *Enrolled* (Matriculado) revelou-se o principal fator de degradação do desempenho, atuando como uma zona de incerteza entre o sucesso e a evasão.

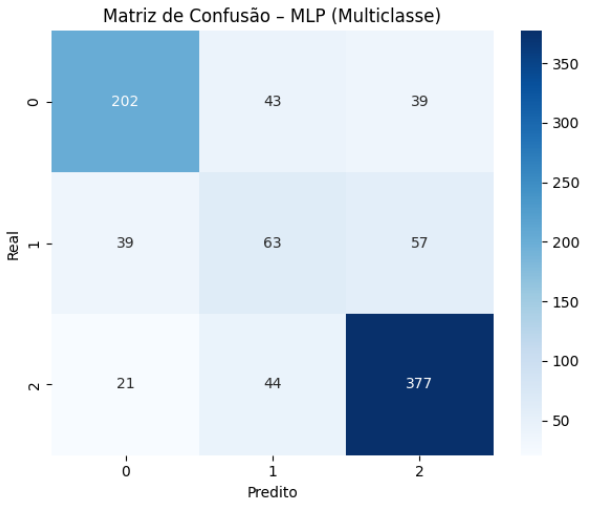
Especificamente para o modelo **MLP**, nota-se uma dificuldade acentuada em delinear a fronteira de decisão da classe *Enrolled*. O modelo tendeu a classificar erroneamente um número significativo de alunos matriculados como evadidos (*Dropout*), o que, na prática, geraria falsos alertas de risco. O **Random Forest**, por sua natureza de particionamento do espaço de características, conseguiu isolar melhor os padrões exclusivos dos alunos matriculados, embora ainda apresente confusão residual com a classe *Dropout*.

Esta confusão é esperada, visto que o estado de "matrícula ativa"é transitório e pode conter alunos com perfis de notas e frequência muito similares aos que acabam evadindo (ex: alunos retidos) ou aos que se graduam (ex: alunos apenas atrasados no fluxo).

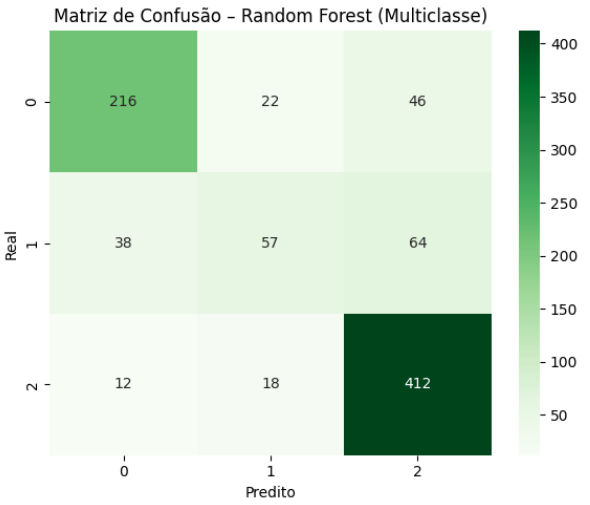
4.4.2 Cenário Binário

Ao remover a classe ruidosa *Enrolled*, a tarefa de classificação tornou-se significativamente mais precisa. Ambos os modelos apresentaram diagonais principais robustas.

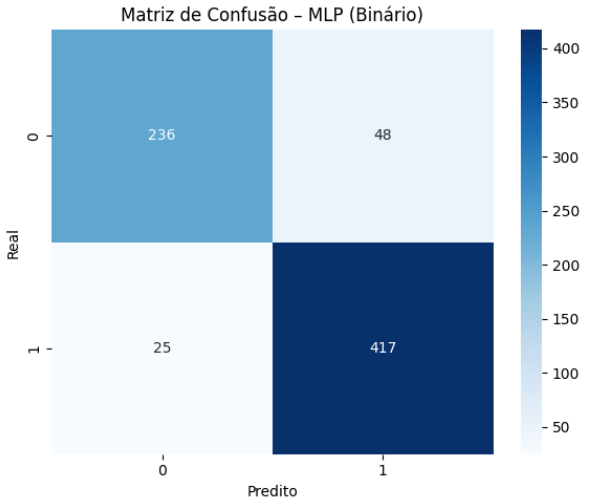
O destaque vai para o desempenho do **Random Forest** na classe *Dropout*. A minimização de falsos negativos nesta classe (alunos que evadiram, mas que o modelo previu que graduariam) é crítica para o objetivo do projeto. O modelo demonstrou alta sensibilidade (*recall*) para detectar a evasão, o que é fundamental para garantir que as intervenções pedagógicas alcancem o público-alvo correto. A MLP, embora competitiva, apresentou uma taxa ligeiramente maior de erros fora da diagonal, confirmando a superioridade das árvores de decisão para este conjunto de dados tabular.



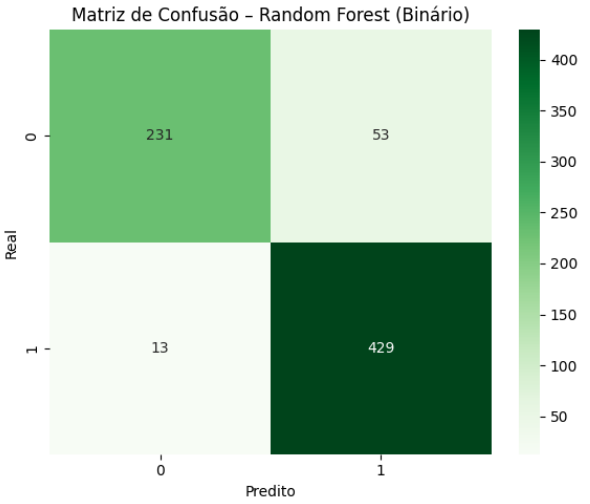
(a) MLP (Multiclasse)



(b) RF (Multiclasse)



(c) MLP (Binário)



(d) RF (Binário)

Figura 3: Matrizes de Confusão comparativas da 30ª execução. A remoção da classe *Enrolled* no cenário binário elimina a maior fonte de ruído, resultando em uma diagonal principal quase perfeita para o Random Forest.

5 Aplicação Desenvolvida (Streamlit)

Para operacionalizar os modelos, desenvolveu-se uma aplicação *web* utilizando o framework **Streamlit**. A ferramenta carrega os artefatos gerados no treinamento (arquivos `.pkl`) e oferece quatro módulos:

1. **Visão Geral:** *Dashboard* com estatísticas descritivas, histogramas interativos das distribuições e mapas de correlação.

2. **Prever por ID:** Permite buscar um aluno pelo seu ID na base de dados e visualizar a predição de risco em tempo real, aplicando o *StandardScaler* automaticamente

antes da inferência.

3. **Prever Novo Aluno:** Formulário para simulação de cenários (ex: "E se a nota do 2º semestre cair?"), essencial para análise de sensibilidade e apoio à decisão.

4. **Comparar Modelos:** Exibe os resultados técnicos (boxplots e matrizes) para transparência sobre a confiabilidade da IA.

6 Conclusões e Trabalhos Futuros

Este trabalho validou o uso de IA para prevenção da evasão. O **Random Forest** mostrou-se o modelo mais robusto, atingindo 91,04% de acurácia no cenário binário. A decisão de manter os dados desbalanceados permitiu observar a dificuldade real dos modelos em classificar a categoria minoritária *Enrolled* no cenário multiclasse, onde a acurácia caiu para 77,62%.

Como trabalhos futuros, sugere-se testar técnicas de balanceamento (SMOTE) especificamente para melhorar a detecção da classe *Enrolled* e explorar algoritmos de *Gradient Boosting* (XGBoost).