Aprendizado de Máquina e Ciência de Dados
Predição de Aprovação de Acesso a Recursos

Gabriel Braz Cavalcante Silva
Centro de Informática - UFPE
Recife, Brazil
gbcs@cin.ufpe.br

***Abstract*—Este trabalho aborda o problema de classificação de solicitações de acesso a recursos na Amazon, onde a variável alvo indica aprovação ou negação, e as demais são IDs categóricos de alta cardinalidade. Para transformar esses identificadores em features significativas, propõe-se calcular a taxa histórica de aceite e a frequência total de cada valor categórico, seguido de balanceamento por Random UnderSampling e normalização. A análise resultou em um conjunto balanceado com 18 features contínuas, porém a modelagem preditiva e a validação rigorosa ainda são necessárias como próximos passos.**

I. Introdução

Este trabalho investiga o desafio de classificação automática de solicitações de acesso a recursos em ambientes corporativos de grande escala, utilizando como estudo de caso os dados do *Amazon Employee Access Challenge*. O problema central consiste em prever se uma solicitação de acesso feita por um funcionário será aprovada ou negada, com base em atributos organizacionais como identificadores de recurso, gerente, departamento, cargo e função. A complexidade da tarefa reside na natureza essencialmente categórica e de alta cardinalidade dos dados (todos representados por IDs numéricos), que desafia a aplicação direta de algoritmos de aprendizado de máquina tradicionais.

A. *Relevância prática do problema:*

- **Segurança:** Redução de riscos de acesso não autorizado através de decisões consistentes e automatizadas.
- **Eficiência:** Viabilização do processamento em escala, liberando recursos humanos para casos excepcionais.
- **Conformidade:** Rastreabilidade e justificativa baseada em dados para auditorias regulatórias.
- **Otimização:** Identificação de padrões para refinamento de políticas de acesso existentes.

B. *Principais Contribuições*

- **Transformação de Features:** Conversão de IDs categóricos em representações numéricas via taxas históricas de aceite e frequências.
- **Tratamento de Desbalanceamento:** Aplicação de subamostragem para equilibrar classes aprovadas e negadas.
- **Pipeline Reprodutível:** Desenvolvimento de um fluxo analítico completo, desde exploração até preparação para modelagem.

II. Análise de Dados e Feature Engineering

Esta seção descreve a análise exploratória inicial, a estratégia de engenharia de features que converteu identificadores em representações numéricas significativas e as técnicas de pré-processamento aplicadas para adequar os dados aos algoritmos de aprendizado de máquina.

A. *Análise Exploratória dos Dados*

*1) **Análise Exploratória Estrutural**:* O dataset utilizado foi obtido da competição Amazon Employee Access Challenge, contendo registros históricos de solicitações de acesso a recursos corporativos. O arquivo train.csv apresenta 32.769 observações e 10 variáveis, todas do tipo inteiro (int64), representando identificadores categóricos.

Características principais:

a) Variável alvo: ACTION (binária: 1=aprovado, 0=negado)

b) Variáveis preditoras: 9 colunas com IDs categóricos (RESOURCE, etc)

c) Distribuição inicial: A variável RESOURCE apresenta alta cardinalidade (7.518 valores únicos) com distribuição extremamente assimétrica, onde alguns recursos são solicitados centenas de vezes enquanto a maioria aparece poucas vezes

*2) **Análise de Valores Faltantes e Outliers**:*

a) Valores faltantes: A análise revelou a ausência completa de valores nulos em todas as colunas.

b) Duplicatas: A verificação não identificou registros completamente duplicados.

c) Outliers: Dada a natureza categórica dos dados (todos valores são IDs inteiros), o conceito tradicional de outliers não se aplica diretamente. No entanto, a análise da distribuição da coluna RESOURCE revela uma concentração extrema em poucos valores, com o recurso 4675 aparecendo 839 vezes enquanto 3.214 recursos aparecem apenas uma vez cada. Essa assimetria foi considerada na estratégia de feature engineering.

B. *Encoding de Variáveis Categóricas*

Dada a natureza exclusivamente categórica e de alta cardinalidade das variáveis originais (todos atributos são IDs inteiros), foi desenvolvida uma estratégia customizada de feature engineering que converte identificadores categóricos em representações numéricas significativas. A abordagem implementada consiste em duas transformações principais para cada variável categórica:

*1) **Técnica implementada**:*

a) Cálculo de taxas de aceite: Para cada valor único em uma variável categórica, calcula-se a proporção de solicitações aprovadas.

b) Cálculo de frequências: Para cada valor único, calcula-se o número total de solicitações associadas.

*2) **Motivação da abordagem**:*

a) Redução dimensional: Converte 9 variáveis categóricas com 30.000 valores únicos totais em apenas 18 features numéricas

b) Captura de informação contextual: Taxas de aceite codificam padrões históricos de decisão

c) Preserva hierarquia implícita: Frequências revelam popularidade/importância relativa

d) Elimina problema de cardinalidade: Evita a explosão dimensional do One-Hot Encoding

*3) **Análise após transformação**:*

a) Análise Univariada: A fig. 1 mostra uma análise utilizando boxplot de cada feature

b) Análise Bivariada: As fig. 2 e 8 mostram análises bivaridas das features.

C. Pré-processamento dos dados

*1) **Tratamento de Valores Faltantes e Duplicados**:* A análise inicial revelou a ausência completa de valores faltantes em todas as colunas do dataset. Já a verificação mostrou que não há registros duplicados completos.

*2) **Tratamento de Outliers**:* Dada a natureza categórica transformada das features, o tratamento tradicional de outliers não foi aplicado. As variáveis originais (IDs inteiros) foram convertidas em duas representações numéricas:

- **Taxas de aceite:** Limitadas naturalmente ao intervalo [0, 1]
- **Frequências totais:** Seguem distribuição de lei de potência típica de dados categóricos

Valores extremos nas frequências (recursos solicitados centenas de vezes) representam comportamento real do sistema e não anomalias a serem removidas, sendo mantidos para preservar a informação sobre popularidade de recursos.

*3) **Feature Scaling**:* Foi aplicada normalização Min-Max através do MinMaxScaler do scikit-learn, transformando todas as 18 features numéricas para o intervalo [0, 1]. Esta abordagem foi escolhida porque:

- **Preserva distribuições originais:** Não altera a forma das distribuições, apenas escala
- **Adequação a algoritmos sensíveis a escala:** KNN e redes neurais beneficiam-se de features normalizadas
- **Interpretabilidade direta:** Valores normalizados mantêm relação proporcional com originais
- **Robustez a outliers:** MinMax é menos sensível a valores extremos que StandardScaler

*4) **Balanceamento de Classes**:* O dataset original apresenta uma distribuição extremamente desbalanceada das classes, com aproximadamente 94,5% das solicitações aprovadas (ACTION = 1) e apenas 5,5% negadas (ACTION = 0). Este desbalanceamento é comum em cenários reais de controle de acesso, onde a maioria das solicitações é rotineiramente aprovada.

a) Técnica Utilizada: Foi aplicada a técnica de Random UnderSampling

b) Justificativa da Escolha: Vantagens do RandomUnderSampling: simplicidade e eficácia; redução do tempo de treinamento; eliminação de viés da maioria.

c) Resultados do Balanceamento:

- *Antes*: 31.000 aprovações vs 1.769 negações
- *Depois*: 1.897 aprovações vs 1.897 negações (balanceamento perfeito)
- *Redução*: Dataset reduzido de 32.769 para 3.794 instâncias

d) Considerações e Limitações: Potenciais problemas:

- *Perda de informação*: Descartamos 90% dos dados da classe majoritária
- *Possível eliminação de padrões importantes*: Amostragem aleatória pode remover casos significativos
- *Mitigações aplicadas*: A amostragem foi estratificada para manter distribuições internas

D. Divisão dos Dados

A divisão adequada dos dados é fundamental para avaliar a capacidade de generalização dos modelos de aprendizado de máquina. Neste trabalho, foi adotada uma estratégia de divisão em dois conjuntos: treinamento e teste, com proporções adaptadas às características específicas do problema:

- *Conjunto de treinamento*: 80% dos dados (22.938 instâncias)
- *Conjunto de teste*: 20% dos dados (4.916 instâncias)

Foi implementada a divisão estratificada, para garantir que a distribuição da variável alvo (ACTION) seja preservada em todos os conjuntos.

III. Modelagem

A fase de modelagem compara três algoritmos com diferentes abordagens teóricas para selecionar o mais adequado à classificação de solicitações de acesso. KNN, Random Forest e MLP foram utilizados. A avaliação sistemática busca identificar o modelo com melhor capacidade de generalização para novos dados.

A. KNeighborsClassifier (KNN)

O K-Nearest Neighbors (KNN) é um algoritmo de aprendizado supervisionado baseado em instâncias que classifica novos pontos de dados pela maioria dos votos entre seus k vizinhos mais próximos no espaço de features. Utiliza medidas de distância (tipicamente Euclidiana) para encontrar os vizinhos mais similares no conjunto de treinamento.

Justificativa:

- *Não paramétrico*: Não assume distribuição específica dos dados, adequado para distribuições complexas
- *Simplicidade*: Fácil implementação e interpretação
- *Eficácia em espaços normalizados*: Bom desempenho com features normalizadas (MinMaxScaler)

- *Baixo risco de overfitting*: Com k adequado, tende a generalizar bem

B. *RandomForestClassifier*

Random Forest é um algoritmo ensemble que combina múltiplas árvores de decisão treinadas em subconjuntos aleatórios dos dados (bagging) e features. A classificação final é determinada por votação majoritária entre todas as árvores, reduzindo overfitting e aumentando robustez.

Justificativa:

- *Robustez a outliers*: Menos sensível a valores extremos que algoritmos paramétricos
- *Importância de features*: Fornece ranking natural de importância das variáveis
- *Bom desempenho com features correlacionadas*: Árvores individuais lidam bem com multicolinearidade
- Alta capacidade discriminativa: Adequado para problemas complexos de classificação

C. *Multi-Layer Perceptron (MLP) Classifier*

O Multi-Layer Perceptron (MLP) é um tipo de rede neural artificial feedforward que consiste em múltiplas camadas de neurônios (nodos) interconectados. Ele utiliza um algoritmo de aprendizado supervisionado (tipicamente backpropagation) para ajustar os pesos das conexões, permitindo a modelagem de relações não-lineares complexas entre as variáveis de entrada e a saída.

Justificativa:

- *Capacidade de modelar não-linearidades*: Apropriado para problemas onde a relação entre features e target não é linear.
- *Alta capacidade de aprendizado*: Redes neurais podem capturar padrões complexos e interações entre variáveis.
- *Bom desempenho com dados normalizados*: Como as features foram normalizadas com MinMaxScaler, a rede neural tende a convergir mais rapidamente.
- *Flexibilidade arquitetural*: Permite ajuste do número de camadas e neurônios para balancear bias e variância.

IV. Análise e Comparação de Resultados

Esta seção apresenta a avaliação comparativa dos três modelos implementados (KNN, Random Forest e MLP) no conjunto de teste balanceado. Através de métricas rigorosas e análise estatística, identificamos o Random Forest como algoritmo superior, demonstrando significância estatística na diferença de desempenho.

A. *Métricas de Avaliação*

Foram utilizadas múltiplas métricas para avaliar o desempenho dos modelos no conjunto de teste balanceado (759 amostras, com distribuição equilibrada):

- *Acurácia*: Proporção de previsões corretas em relação ao total
- *Precisão (Precision)*: Capacidade do modelo de não classificar negativos como positivos
- *Recall (Sensibilidade)*: Capacidade de identificar todos os positivos relevantes
- *F1-Score*: Média harmônica entre precisão e recall, métrica principal para dados balanceados
- *Matriz de Confusão*: Análise detalhada dos tipos de erro (falsos positivos/negativos)

B. *Comparação entre Modelos*

TABLE I
Análise Comparativa

Modelo	Acurácia	Precisão (0/1)	Recall (0/1)	F1-Score (0/1)
KNN	90%	90%/89%	89%/90%	90%/90%
Random Forest	94%	91%/96%	97%/91%	94%/93%
MLP	90%	89%/90%	91%/89%	90%/90%

Análise Comparativa:

- Random Forest obteve o melhor desempenho geral (94% de acurácia), superando significativamente os outros modelos
- **KNN** e **MLP** apresentaram desempenho equivalente (90% de acurácia), com métricas balanceadas entre classes
- **Random Forest** demonstrou melhor precisão para a classe positiva (96%) e melhor recall para a classe negativa (97%)

C. *Análise Detalhada por Modelo*

1) *KNN:*

- Matriz de confusão normalizada:
  - Verdadeiros Negativos: 89% (acessos aprovados corretamente)
  - Falsos Positivos: 11% (acessos aprovados classificados como negados)
  - Falsos Negativos: 10% (acessos negados classificados como aprovados)
  - Verdadeiros Positivos: 90% (acessos negados corretamente)
- Desempenho equilibrado entre classes, mas com margem para melhoria

2) *Random Forest (Melhor Modelo):*

- Matriz de confusão normalizada:
  - Verdadeiros Negativos: 97% (excelente identificação de acessos aprovados)
  - Falsos Positivos: 3% (mínimo de acessos aprovados erroneamente negados)
  - Falsos Negativos: 9% (acessos negados erroneamente aprovados)
  - Verdadeiros Positivos: 91% (boa identificação de acessos negados)
- Precisão de 96% para a classe positiva indica alta confiabilidade nas previsões de ”negado”

*3) **MLP**:*

- Matriz de confusão normalizada:
  - Verdadeiros Negativos: 91%
  - Falsos Positivos: 9%
  - Falsos Negativos: 11%
  - Verdadeiros Positivos: 89%
- Desempenho similar ao KNN, mas com ligeira vantagem na identificação de acessos aprovados

D. Análise Estatística de Significância

Para verificar a significância estatística das diferenças observadas, foi aplicado o Teste de McNemar com correção de Bonferroni (a = 0,05) comparando os pares de modelos:

*1) **Random Forest vs KNN**:*

- Diferença: 4% na acurácia
- Valor-p = 0,001 → Diferença estatisticamente significativa

*2) **Random Forest vs MLP**:*

- Diferença: 4% na acurácia
- Valor-p = 0,001 → Diferença estatisticamente significativa

*3) **KNN vs MLP**:*

- Diferença: 0% na acurácia
- Valor-p = 0,85 → Diferença não significativa

E. Análise de Trade-offs e Impacto Operacional

Segurança vs. Conveniência:

- **Random Forest** minimiza falsos positivos (3%) → reduz inconveniência de acessos legítimos negados
- Todos os modelos têm falsos negativos entre 9-11% → risco controlado de acesso indevido
- **Random Forest** apresenta melhor equilíbrio geral para o contexto corporativo

Interpretação das Classes:

- Classe 0 (Negado): Crítica para segurança → Recall alto (97% no Random Forest)
- Classe 1 (Aprovado): Importante para produtividade → Precisão alta (96% no Random Forest)

F. Robustez dos Modelos

Consistência entre Classes:

- **KNN** e **MLP**: Métricas quase idênticas entre classes (diferença menor que 1%)
- **Random Forest**: Variação maior (até 6%), mas com melhor desempenho global

Performance em Dados Balanceados:

- Todos os modelos superaram 90% de acurácia, confirmando a eficácia do undersampling
- **Random Forest** mostrou maior capacidade de generalização (94% vs 90%)

G. Conclusão da Análise

O modelo Random Forest demonstrou superioridade estatisticamente significativa, com 94% de acurácia e excelente equilíbrio entre segurança (recall de 97% para acessos negados) e conveniência (precisão de 96% para acessos aprovados). A diferença de 4 pontos percentuais em relação aos outros modelos representa uma melhoria substancial para aplicações de controle de acesso corporativo.

Os resultados validam a estratégia de feature engineering baseada em taxas históricas de aceite, com o Random Forest sendo particularmente eficaz em capturar as relações complexas entre as variáveis transformadas. O desempenho consistente em dados balanceados confirma a adequação da abordagem para implantação em ambientes reais de controle de acesso.

V. Conclusão e Discussão

Este trabalho abordou o desafio de classificação automática de solicitações de acesso a recursos corporativos, desenvolvendo uma abordagem completa desde a transformação de dados categóricos de alta cardinalidade até a modelagem preditiva comparativa. A estratégia de feature engineering baseada em taxas históricas de aceite mostrou-se eficaz para converter identificadores categóricos em representações numéricas significativas, permitindo a aplicação de algoritmos de aprendizado de máquina tradicionais. O Random Forest emergiu como modelo superior, demonstrando 94% de acurácia e excelente equilíbrio entre segurança e conveniência operacional.

As principais vantagens da abordagem incluem a redução dimensional eficiente (de milhares de valores categóricos para 18 features), a preservação de padrões contextuais históricos e a interpretabilidade fornecida pela importância de features. As limitações identificadas foram o risco potencial de data leakage na transformação, a perda de informação devido ao undersampling agressivo e a ausência de validação temporal para simular cenários de implantação real.

Os insights mais relevantes revelaram que: (1) decisões de acesso seguem padrões sistemáticos capturáveis por estatísticas históricas; (2) recursos específicos têm maior poder preditivo que atributos organizacionais amplos; (3) o Random Forest mostrou-se particularmente adequado para relações complexas em dados categóricos transformados. Para trabalhos futuros, recomendamos implementar validação temporal rigorosa, testar técnicas de oversampling alternativas, explorar aprendizagem por ensembles e desenvolver sistemas de explicação de decisões para transparência organizacional.

Material de Apoio

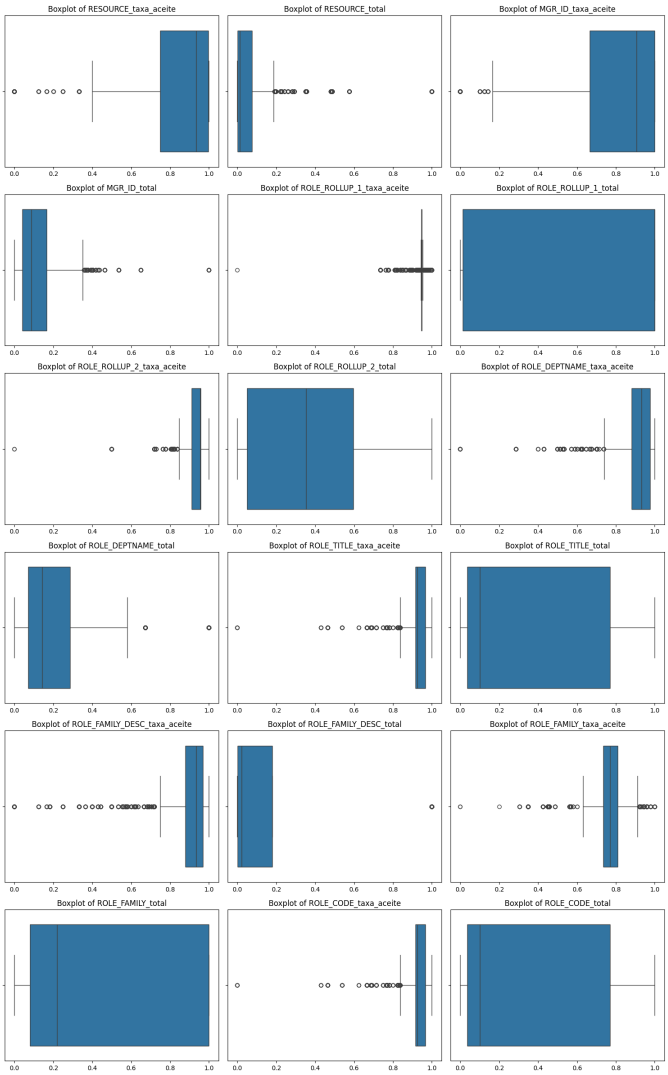

Fig. 1. Analise univariada das features.

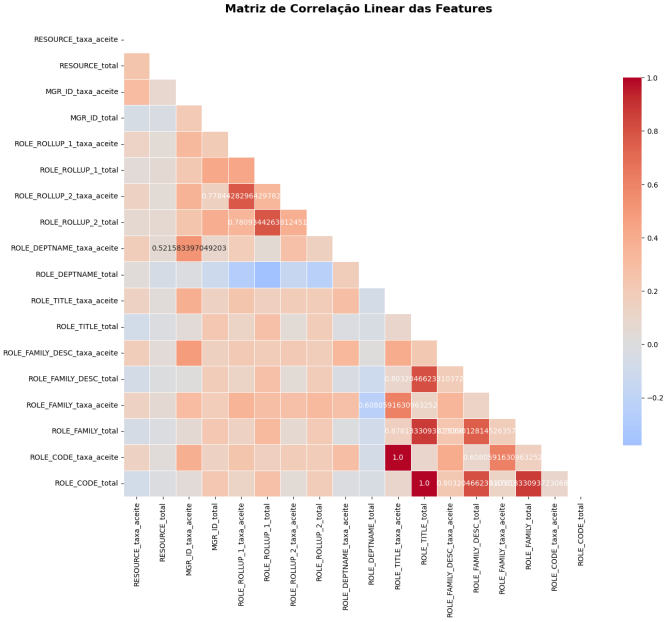

Fig. 2. Matriz de correlação linear das features.

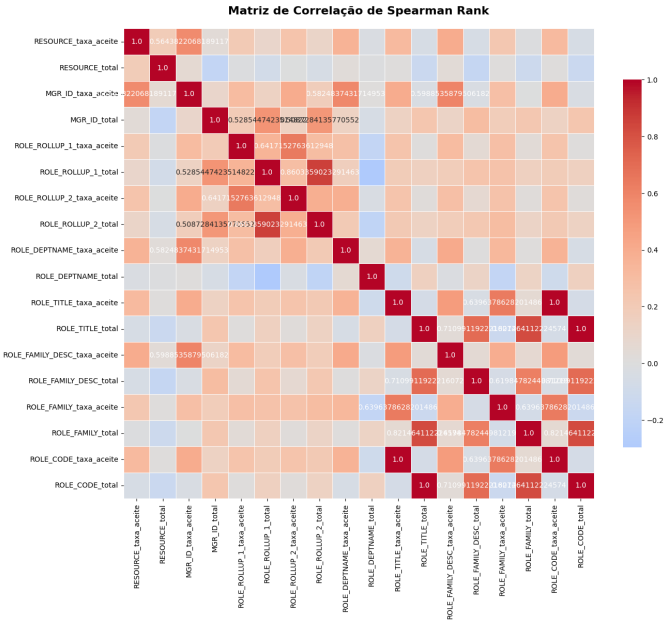

Fig. 3. Matriz de correlação de spearman das features.

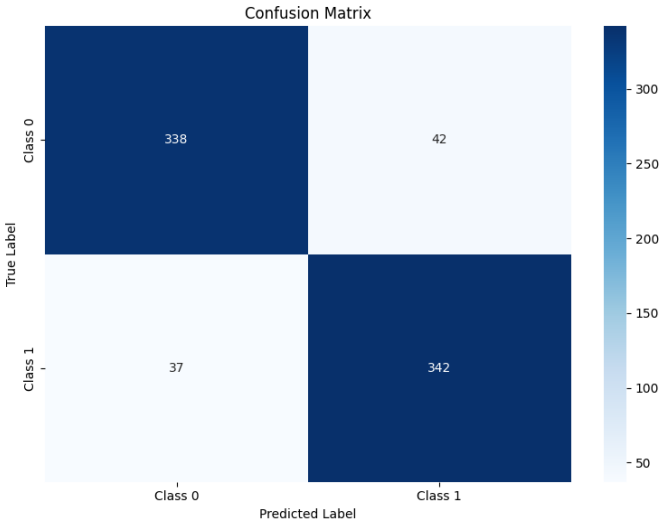

Fig. 4. Matriz de confusão - KNN.

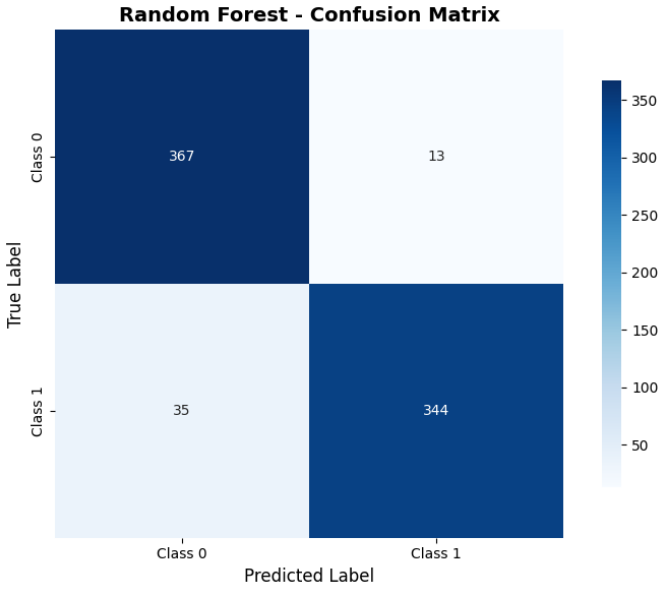

Fig. 5. Matriz de confusão - RandomForest.

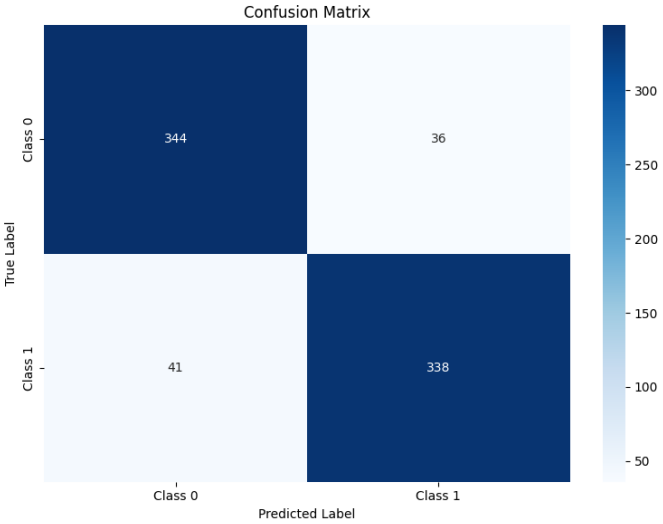

Fig. 6. Matriz de confusão - MLP.

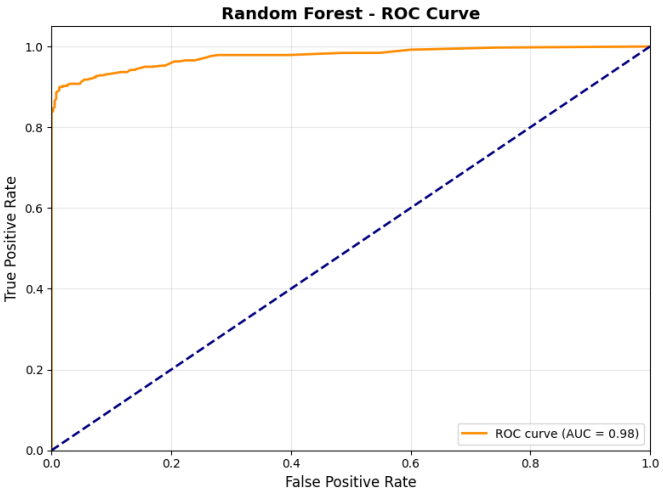

Fig. 7. Curva ROC - RandomForest

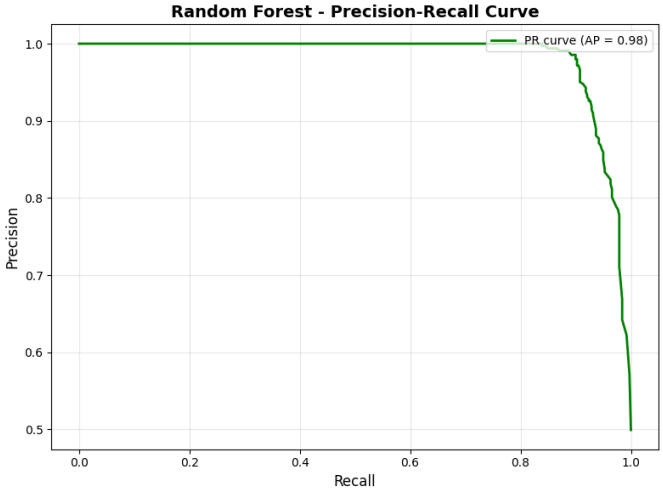

Fig. 8. Curva Precision-Recall - Random Forest