

# CreditAI: Sistema de Concessão de Crédito e Detecção de Risco com Técnicas de Inteligência Artificial

Diego Meinerz<sup>1</sup>, Elias A. G. Miranda<sup>1</sup>, Fernanda T. Wammes<sup>1</sup>, Matheus M. Burin<sup>1</sup>

<sup>1</sup>Universidade Regional do Noroeste do Estado do Rio Grande do Sul (UNIJUÍ)  
Santa Rosa – RS – Brazil, Departamento de Ciências da Computação

diego.meinerz@sou.unijui.edu.br, elias.miranda@sou.unijui.edu.br

fernanda.wammes@sou.unijui.edu.br, matheus.burin@sou.unijui.edu.br

**Abstract.** *This paper presents CreditAI, a decision-support system for credit granting and default risk detection in financial institutions. The proposed model automates the analysis of loan applications by combining four Artificial Intelligence techniques in a sequential pipeline: (1) Depth-First Search (DFS) for persona classification using decision trees, (2) Breadth-First Search (BFS) for credit limit calculation exploring amount-installment state spaces, (3) Fuzzy Logic with scikit-fuzzy for risk assessment, and (4) a Multi-Layer Perceptron (MLP) neural network implemented in PyTorch for final approval decision. The system receives customer profile data (income, credit score, employment status, debt ratio, among others) and returns the approved limit, risk level (low, medium or high) and the final status (approved, rejected or pending review). Results indicate that the approach allows more transparent, consistent and scalable credit analysis when compared to purely manual processes.*

**Resumo.** *Este artigo apresenta o CreditAI, um sistema de apoio à decisão para concessão de crédito e detecção de risco de inadimplência em instituições financeiras. O modelo proposto automatiza a análise de propostas de empréstimo combinando quatro técnicas de Inteligência Artificial em um pipeline sequencial: (1) Busca em Profundidade (DFS) para classificação de personas usando árvores de decisão, (2) Busca em Largura (BFS) para cálculo de limite explorando espaços de estados valor-parcela, (3) Lógica Fuzzy com scikit-fuzzy para avaliação de risco, e (4) uma Rede Neural Perceptron Multicamadas (MLP) implementada em PyTorch para decisão final de aprovação. A solução recebe dados do perfil do cliente (renda, score de crédito, status de emprego, taxa de endividamento, entre outros) e retorna o limite aprovado, o nível de risco (baixo, médio ou alto) e o status final (aprovado, reprovado ou em análise). Os resultados indicam maior transparência, padronização e escalabilidade em relação ao processo manual tradicional.*

## 1. Introdução

A concessão de crédito é uma atividade central para bancos, cooperativas e fintechs, mas envolve riscos relevantes de inadimplência e perda financeira. Tradicionalmente, a análise

de propostas era realizada por analistas de crédito, de forma manual, a partir de documentos, consultas a birôs e experiência prévia. Esse processo tende a ser lento, subjetivo, custoso e suscetível a erros humanos, especialmente em cenários de alto volume de solicitações.

Com a expansão do crédito digital e o surgimento de novas fintechs, há uma demanda crescente por sistemas automatizados que apoiem a tomada de decisão, reduzindo risco e aumentando a eficiência. Nesse contexto, propomos o *CreditAI*, um sistema que aplica quatro técnicas distintas de Inteligência Artificial organizadas em um pipeline sequencial para analisar pedidos de crédito, estimar limites e classificar o risco de inadimplência de forma estruturada.

## 2. Problema e Relevância

O problema tratado neste trabalho pode ser resumido da seguinte forma: dado o perfil de um cliente e os parâmetros de produtos de crédito (empréstimo pessoal, cartão de crédito, garantia de veículo ou imóvel), decidir se a proposta deve ser aprovada, reprovada ou encaminhada para análise manual, bem como calcular o limite sugerido e o risco associado.

Os principais desafios são: (i) grande volume de solicitações; (ii) dados por vezes incompletos ou inconsistentes; (iii) necessidade de respostas rápidas; (iv) cumprimento de políticas internas e exigências regulatórias; (v) balanceamento entre inclusão financeira e gestão de risco.

A relevância do problema está na redução de perdas financeiras, aumento da competitividade, maior inclusão financeira e melhoria da rastreabilidade e explicabilidade das decisões de crédito.

## 3. Dados Utilizados

O *CreditAI* utiliza dados estruturados do perfil do cliente, organizados nos seguintes grupos:

- **Dados demográficos:** idade, gênero e estado civil.
- **Dados financeiros:** renda mensal (`income`), score de crédito em escala de 0 a 1000 (`credit_score`), e taxa de endividamento como proporção da renda (`debt_to_income_ratio`).
- **Dados de emprego:** status de emprego (`employment_status`: empregado, autônomo, aposentado ou desempregado) e tempo no emprego atual em meses (`time_at_job_months`).
- **Histórico bancário e de crédito:** existência de conta bancária (`has_bank_account`), restrições no BACEN (`has_bacen_restriction`), número de consultas de crédito recentes (`num_credit_inquiries`) e número de empréstimos ativos (`num_existing_loans`).
- **Dados da solicitação:** valor solicitado (`requested_amount`), número de parcelas desejado (`requested_installments`) e tipo de produto (`product_type`).

Cada produto possui parâmetros específicos de valor mínimo/máximo, número máximo de parcelas e taxa de juros base.

## 4. Modelo Proposto

O *CreditAI* organiza o processo de análise em quatro etapas sequenciais, cada uma utilizando uma técnica de Inteligência Artificial específica:

### 4.1. Etapa 1: Filtro por Persona com Busca em Profundidade (DFS)

A primeira etapa classifica o cliente em uma de três personas de crédito – *Premium*, *Standard* ou *Basic* – utilizando uma árvore de decisão percorrida por **Busca em Profundidade (Depth-First Search – DFS)**.

A árvore é construída com nós de decisão baseados em condições sobre os atributos do cliente. Cada nó interno representa uma condição (por exemplo, “Renda  $\geq$  R\$ 10.000?”), com ramificações para respostas verdadeiras e falsas. Os nós folha representam as personas finais ou rejeição quando o cliente não se enquadra em nenhuma categoria.

A navegação na árvore inicia pela raiz e segue recursivamente pelos ramos até alcançar uma folha:

1. **Persona Premium:** Renda  $\geq$  R\$ 10.000, emprego qualificado (empregado ou autônomo) e score  $\geq$  750. Limite máximo: R\$ 100.000, multiplicador de renda: 5x.
2. **Persona Standard:** Renda  $\geq$  R\$ 2.000, emprego qualificado e score  $\geq$  550. Limite máximo: R\$ 50.000, multiplicador de renda: 3x.
3. **Persona Basic:** Renda  $\geq$  R\$ 0.00, emprego qualificado ou aposentado, e score  $\geq$  0. Limite máximo: R\$ 20.000, multiplicador de renda: 2x.

Se o cliente não atende aos critérios mínimos de nenhuma persona, a proposta é rejeitada automaticamente nesta etapa.

### 4.2. Etapa 2: Cálculo do Limite com Busca em Largura (BFS)

O objetivo desta etapa é determinar o *limite de crédito aprovado* e o número de parcelas viável utilizando **Busca em Largura (Breadth-First Search – BFS)** para explorar o espaço de estados (valor, parcelas).

O algoritmo funciona da seguinte forma:

1. Calcula o limite base a partir da renda e do multiplicador da persona:  
`income_limit = income * multiplier.`
2. Aplica fatores de ajuste baseados em:
  - Score de crédito (`score_factor`): varia de 0.7 a 1.2 dependendo da faixa de score.
  - Status de emprego (`employment_factor`): empregado (1.0), autônomo (0.95), aposentado (0.85), desempregado (0.5).
  - Histórico de crédito (`history_factor`): bonificação por empréstimos anteriores e penalização por alto endividamento.
3. Define o teto de busca (`search_cap`) como o mínimo entre o limite da persona, limite do produto e limite ajustado pelos fatores.
4. Inicia a BFS com estado inicial (`min_amount`, `start_installments`) e explora vizinhos:

- Incrementa o valor em passos fixos (ex: R\$ 500).
  - Aumenta o número de parcelas para reduzir valor mensal.
5. Para cada estado, calcula a parcela mensal usando a fórmula Price:

$$PMT = P \cdot \frac{r \cdot (1 + r)^n}{(1 + r)^n - 1}$$

onde  $P$  é o principal,  $r$  a taxa mensal e  $n$  o número de parcelas.

6. Valida se a parcela respeita a regra de comprometimento máximo de renda (30%). Se válido, atualiza o melhor valor encontrado.
7. Continua a busca até esgotar os estados viáveis ou atingir o teto de busca.

A BFS garante que o sistema explore sistematicamente as combinações de valor e prazo, encontrando o maior limite aprovável que respeita as restrições de capacidade de pagamento.

### 4.3. Etapa 3: Avaliação de Risco com Lógica Fuzzy

A terceira etapa estima o risco de inadimplência usando **Lógica Fuzzy** implementada com a biblioteca `scikit-fuzzy`. Esta abordagem permite lidar com a incerteza e subjetividade inerentes à avaliação de risco.

O sistema fuzzy é composto por:

- **Variáveis de entrada** (antecedentes):
  - `credit_score`: score de crédito (0–1000), com conjuntos fuzzy *low*, *med* e *high*.
  - `income`: renda mensal (0–50.000), com conjuntos *low*, *med* e *high*.
  - `debt_ratio`: taxa de endividamento (0–1), com conjuntos *low*, *med* e *high*.
  - `employment_time`: tempo de emprego em meses (0–120), com conjuntos *short*, *med* e *long*.
  - `inquiries`: consultas de crédito (0–20), com conjuntos *few* e *many*.
  - `limit_ratio`: razão entre valor solicitado e limite aprovado (0–1), com conjuntos *low*, *med* e *high*.
- **Variável de saída** (consequente): `risk`: score de risco (0–1), com conjuntos *low*, *med* e *high*.
- **Funções de pertinência**: funções trapezoidais e triangulares modelam a transição gradual entre conceitos (ex: score “baixo” vs “médio”). O sistema fuzzy utiliza funções trapezoidais nos extremos das variáveis (garantindo região de pertinência máxima seguida de transição gradual) e funções triangulares para valores intermediários (com pico único no protótipo ideal). Essa combinação permite que um valor possua graus de pertinência em múltiplos conjuntos simultaneamente, ativando várias regras em paralelo e gerando decisões ponderadas que refletem a incerteza natural dos critérios de crédito.
- **Regras fuzzy**: 15 regras do tipo SE-ENTÃO capturam o conhecimento especialista, por exemplo:
  - SE score é *high* E dívida é *low* ENTÃO risco é *low*.
  - SE score é *low* E dívida é *med* ENTÃO risco é *high*.
  - SE dívida é *high* E renda é *low* ENTÃO risco é *high*.

O processo de inferência fuzzy funciona como segue:

1. **Fuzzificação:** os valores numéricos das entradas são convertidos em graus de pertinência nos conjuntos fuzzy.
2. **Inferência:** as regras são avaliadas usando operadores fuzzy (AND = mínimo, OR = máximo).
3. **Agregação:** os resultados das regras ativadas são combinados.
4. **Defuzzificação:** o conjunto fuzzy de saída é convertido em um valor numérico.

O score de risco resultante (0–1) é mapeado em três níveis:

- **Baixo risco:**  $\text{risco} < 0.40$
- **Médio risco:**  $0.40 \leq \text{risco} < 0.70$
- **Alto risco:**  $\text{risco} \geq 0.70$

O sistema também gera gráficos das curvas de pertinência com o score calculado marcado, facilitando a interpretação e auditoria das decisões.

#### 4.4. Etapa 4: Decisão Final com Rede Neural (PyTorch)

A etapa final utiliza uma **Rede Neural Perceptron Multicamadas (MLP)** implementada em **PyTorch** para determinar a decisão final: *approved*, *pending\_review* ou *rejected*.

A arquitetura da rede é:

- **Camada de entrada:** 10 neurônios correspondentes aos atributos normalizados:
  - Idade normalizada:  $(idade - 18)/(75 - 18)$
  - Score normalizado:  $score/1000$
  - Renda normalizada:  $\min(1, \ln(1 + renda)/\ln(50001))$
  - Taxa de endividamento (0–1)
  - Emprego (binário: 1 se empregado/autônomo, 0 caso contrário)
  - Conta bancária (binário)
  - Consultas normalizadas:  $\min(1, consultas/10)$
  - Empréstimos normalizados:  $\min(1, empréstimos/5)$
  - Score de risco fuzzy (0–1)
  - Razão limite:  $\min(1, valor\_solicitado/limite)$
- **Camada oculta:** 16 neurônios com função de ativação ReLU.
- **Camada de saída:** 3 neurônios (logits) correspondentes às classes *approved*, *pending* e *rejected*.

Utilizou-se duas funções de ativação:

ReLU na camada oculta: evita saturação de gradientes, treina rápido em dados tabulares e lida bem com valores não normalizados simétricos.

Softmax na saída: transforma os logits em probabilidades somando 1 para as três classes.

Os pesos da rede são inicializados com heurísticas de negócio alinhadas às regras de rotulagem dos dados sintéticos, garantindo convergência estável durante o treinamento. A decisão final é obtida aplicando `softmax` aos logits e selecionando a classe com maior probabilidade.

#### 4.4.1. Treinamento da Rede

O sistema implementa um pipeline completo de treinamento:

1. **Geração de dados sintéticos:** método `generate_dataset_jsonl` cria exemplos de treino com 1000 amostras por padrão, usando regras de negócio para rotulação:
  - *Rejected* se:  $\text{risco fuzzy} > 0.75$  OU  $\text{score} < 500$  OU  $\text{endividamento} > 0.55$ .
  - *Pending* se:  $\text{risco} \geq 0.45$  OU  $\text{razão limite} > 0.95$  OU *desempregado*.
  - *Approved* caso contrário.
2. **Treinamento:** método `train_from_jsonl` realiza o treinamento supervisionado usando:
  - Otimizador: Adam com learning rate de  $10^{-3}$  e weight decay de  $10^{-4}$ .
  - Função de perda: Cross-Entropy Loss.
  - Batch size: 64.
  - Épocas: 30 (padrão).
3. **Rastreamento com MLflow:** o sistema integra-se com MLflow para registrar:
  - Hiperparâmetros (learning rate, epochs, batch size).
  - Métricas de perda por época.
  - Artefatos (pesos do modelo, dados de treino).
4. **Salvamento e carregamento:** os pesos treinados são salvos em `models/approval_mlp.pt` e carregados automaticamente na próxima inicialização.

A rede neural permite capturar interações não-lineares complexas entre os atributos que regras manuais dificilmente modelariam, além de fornecer probabilidades calibradas para cada decisão.

### 5. Pipeline de Execução

O *CreditAI* executa as quatro etapas sequencialmente, com saída antecipada em caso de rejeição:

1. Se a **Etapla 1** (DFS) não encontrar persona válida, a proposta é rejeitada com motivo `PERSONA_FILTER`.
2. A **Etapla 2** (BFS) calcula o limite aprovado e fatores relevantes.
3. A **Etapla 3** (Fuzzy) avalia o risco de inadimplência.
4. A **Etapla 4** (RNA) toma a decisão final considerando todos os dados anteriores:
  - Se *approved*: retorna valor aprovado, parcelas e valor mensal.
  - Se *rejected*: define o motivo como `HIGH_RISK` se  $\text{score fuzzy} > 0.65$ , caso contrário `OTHER`.
  - Se *pending*: encaminha para análise manual.

O sistema retorna um objeto `CreditAnalysisResult` completo com todos os detalhes das quatro etapas, permitindo auditoria e explicabilidade da decisão.

## 6. Resultados e Discussão

A implementação do *CreditAI* demonstra a viabilidade de integrar múltiplas técnicas de IA em um pipeline de análise de crédito. Cada técnica contribui com capacidades específicas:

- **DFS (Etapa 1):** Garante classificação eficiente em árvore de decisão com tempo  $O(h)$ , onde  $h$  é a altura da árvore. Permite segmentação clara de clientes e aplicação de políticas diferenciadas por persona.
- **BFS (Etapa 2):** Explora sistematicamente o espaço de soluções valor-parcela, garantindo que a melhor combinação viável seja encontrada. Tempo  $O(V)$  onde  $V$  é o número de estados visitados (limitado pelo teto de busca e passo).
- **Lógica Fuzzy (Etapa 3):** Trata incerteza e conceitos vagos (“score razoável”, “dívida alta”) de forma natural, produzindo scores contínuos mais informativos que classificações binárias. Facilita explicação das decisões através das regras ativas.
- **RNA/PyTorch (Etapa 4):** Captura padrões complexos e não-lineares, aprendendo com dados sintéticos que refletem políticas de negócio. O treinamento com back-propagation permite ajuste fino dos pesos. Fornece probabilidades calibradas para cada classe de decisão.

### 6.1. Vantagens da Abordagem

- **Modularidade:** cada etapa pode ser ajustada ou substituída independentemente.
- **Explicabilidade:** as três primeiras etapas são baseadas em regras interpretáveis; a RNA fornece probabilidades que auxiliam na explicação.
- **Escalabilidade:** o sistema processa requisições em tempo constante após carregamento dos modelos, adequado para alto volume.
- **Rastreabilidade:** integração com MLflow permite versionamento e auditoria completa dos modelos e decisões.

## 7. Considerações Finais

O *CreditAI* demonstra como técnicas clássicas de IA (busca em grafos, lógica fuzzy) podem ser integradas com aprendizado de máquina moderno (redes neurais com PyTorch) para criar um sistema de análise de crédito robusto e explicável. A organização em pipeline sequencial com quatro etapas distintas permite balancear eficiência computacional, interpretabilidade e capacidade preditiva.

O projeto serve como prova de conceito para instituições financeiras interessadas em automatizar e padronizar seus processos de concessão de crédito, reduzindo custos operacionais e tempo de resposta enquanto mantém controle sobre políticas de risco. A arquitetura modular facilita adaptações para diferentes produtos, segmentos de mercado e requisitos regulatórios.

## Referências

Cerf, S., Dayan, B., De Ambroggio, U., Kaufmann, M., Lengler, J., and Schaller, U. (2024). Balanced bidirectional breadth-first search on scale-free networks. Acesso em: 2025.

Chang, P. C. (2022). Multi-layer perceptron neural network for improving detection performance of malicious phishing urls without affecting other attack types classification. Acesso em: 2025.

Jurkovič, F. (2004). Stochastic fuzzy controller. Acesso em: 2025.

Tarjan, R. E. and Zwick, U. (2022). Finding strong components using depth-first search. Acesso em: 2025.

[Cerf et al. 2024] [Chang 2022] [Jurkovič 2004] [Tarjan and Zwick 2022]